VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL REPORT**

**SOFTWARE ENGINEERING**

**FINAL REPORT**

**SOFTWARE ENGINEERING**

*Instructor:* **Master. Pham Thai Ky Trung**

*Student*: **Kieu Thanh Phat – 521H0125**

**Tran Dang Dang Khoa – 521H0086**

*Class* **: 21H50203**

Course **: 25**

**HO CHI MINH CITY, 2023**

VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL REPORT**

**SOFTWARE ENGINEERING**

**FINAL REPORT**

**SOFTWARE ENGINEERING**

*Instructor:* **Master. Pham Thai Ky Trung**

*Student*: **Kieu Thanh Phat – 521H0125**

**Tran Dang Dang Khoa – 521H0086**

*Class* **: 21H50203**

Course **: 25**

**HO CHI MINH CITY, 2023**

THANK YOU FOR TEACHER

Our group would like to express our gratitude to Master Pham Thai Ky Trung for creating favorable conditions for our group and other students in the class throughout the learning process to acquire and enhance our knowledge of the subject of software engineering. The procedures and issues, including numerous learning programs in the field of web, were taught by the teacher. We would like to extend our sincere thanks to the teacher for the support and dedicated teaching to our group. On behalf of the group, we sincerely thank you, Master, very much.

**PROJECT COMPLETED AT TON DUC THANG UNIVERSITY**

I hereby declare that this is my own project and is under the guidance of Dr. Mai Duy Tan. 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 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, 1st May, 2023*

*Author*

*(sign and write full name)*

**

*Trần Đặng Đăng Khoa*

CONFIRMATION AND EVALUATION SECTION OF THE LECTURER

MENTOR'S CONFIRMATION SECTION

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

Ho Chi Minh City, date month year

(signature and full name)

GRADER'S EVALUATION SECTION

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

Ho Chi Minh City, date month year

(signature and full name)

SUMMARISE

In this final semester report, our team aims to analyze and design software for a distributor selling Mobile Phones products to authorized agents. The software needs to have the following functions: Accountants shall be able to create Goods Received when the distributor imports goods (awarehouse receipt will include many items); Agents shall be able to place an order of items and choose a payment method (Cash, bank transfer, Momo...) and Agents would also Agents would also make an online payment and see the status of their orders; Accountants shall be able to create Goods Delivery Note to deliver goods to agents (print deliveryslips), update the status of orders as being transferred and update the payment status of agents; Accountants shall be able to view incoming/outgoing stock report, best-selling products and revenue report monthly using language C# to solution for questions. Specially, our teams are using three-tier Model MVC as well as many helpful frameworks will help accrlerate process.

TABLE OF CONTENTS

[THANK YOU FOR TEACHER iii](#_Toc134644847)

[CONFIRMATION AND EVALUATION SECTION OF THE LECTURER v](#_Toc134644848)

[SUMMARISE vi](#_Toc134644849)

[TABLE OF CONTENTS 1](#_Toc134644850)

[LIST OF SYMBOLS AND ABBREVIATIONS 3](#_Toc134644851)

[LIST OF TABLES, FIGURES, AND GRAPHS 4](#_Toc134644852)

[1. Introduction 5](#_Toc134644853)

[1.1 Purpose and Scope 5](#_Toc134644854)

[1.2 Product Overview 5](#_Toc134644855)

[1.3 Structure of the Document 6](#_Toc134644856)

[1.4 Terms, Acronyms, and Abbreviations 7](#_Toc134644857)

[2. Project Management Plan 7](#_Toc134644858)

[2.1 Project Organization 7](#_Toc134644859)

[2.2 Lifecycle Model Used 8](#_Toc134644860)

[2.3 Risk Analysis 8](#_Toc134644861)

[2.4 Hardware and Software Resource Requirements 9](#_Toc134644862)

[2.5 Deliverables and Schedule 9](#_Toc134644863)

[2.6 Monitoring, Reporting and Controlling Mechanisms 9](#_Toc134644864)

[2.7 Professional Standard 10](#_Toc134644865)

[2.8 Evidence all the artifacts have been placed under configuration management 10](#_Toc134644866)

[2.9 Impact of the project on individuals and organizations 11](#_Toc134644867)

[3. Requirement Specifications 12](#_Toc134644868)

[3.1 Stakeholders for the system 12](#_Toc134644869)

[3.2 Use case model 12](#_Toc134644870)

[3.3 Functional requirements 19](#_Toc134644871)

[3.4 Non-functional requirements 20](#_Toc134644872)

[4. Architecture 20](#_Toc134644873)

[4.1 Architecture style(s) used 20](#_Toc134644874)

[4.2 Architectural model 21](#_Toc134644875)

[4.3 Technology, software, and hardware used 21](#_Toc134644876)

[4.4 Rationable 22](#_Toc134644877)

[5. Design 22](#_Toc134644878)

[5.1 Database design 22](#_Toc134644879)

[5.2 Static model – class diagram 22](#_Toc134644880)

[5.3 Dynamic model – sequence diagram 23](#_Toc134644881)

[6. Test Plan 26](#_Toc134644882)

[6.1 Requirements/specifications-based system level test cases 26](#_Toc134644883)

[6.2 Traceability of test cases to use cases 28](#_Toc134644884)

[6.3 Techniques used for test generation 29](#_Toc134644885)

[6.4 Assessment of the goodness of your testsuite 29](#_Toc134644886)

[7. Demo 29](#_Toc134644887)

[7.1 Database 29](#_Toc134644888)

[7.2 Source code 29](#_Toc134644889)

LIST OF SYMBOLS AND ABBREVIATIONS

**SYMBOLS**

**ABBREVIATIONS**

LIST OF TABLES, FIGURES, AND GRAPHS

**LIST OF PICTURE**

[Picture 1.1: Use case Agents 14](#_Toc134644641)

[Picture 1.2: Use case Accountant with Goods 15](#_Toc134644642)

[Picture 1.3: Use case Accountant with Agents 16](#_Toc134644643)

[Picture 1.4: Use case Description for Agents 18](#_Toc134644644)

[Picture 1.5: Use case Description for Accountant with Goods 19](#_Toc134644645)

[Picture 1.6: Use case Description for Accountant with Agents 20](#_Toc134644646)

[Picture 1.7: Database 23](#_Toc134644647)

[Picture 1.8: Static model – Class diagram 24](#_Toc134644648)

[Picture 1.9: Dynamic model – Sequence - Agents diagram 25](#_Toc134644649)

[Picture 1.10: Dynamic model – Sequence – Accountant with Goods diagram 26](#_Toc134644650)

[Picture 1.11: Dynamic model – Sequence – Accountant with Agents diagram 27](#_Toc134644651)

**LIST OF TABLES**

1. Introduction

1.1 Purpose and Scope

**Purpose:** The purpose of the software requirements for a distributor selling Mobile Phones products to authorized resellers/agents is to provide the necessary functionality to support various aspects of the business operations.

**Scope:**

**- Accountants' Functions:**

+ Enable accountants to create Goods Received when the distributor imports goods. This includes the ability to generate warehouse receipts that encompass multiple items. The functionality should be accessible through both Win Form and Web Form interfaces.

+ Allow accountants to create Goods Delivery Notes for delivering goods to agents. This includes the ability to print delivery slips. Additionally, accountants should be able to update the status of orders as they are being transferred and update the payment status of agents. These functions should be available through both Win Form and Web Form interfaces.

+ Provide accountants with the ability to view incoming/outgoing stock reports, best-selling products, and monthly revenue reports. These reports should be accessible through Form or Report interfaces.

**- Reseller/Agent Functions:**

+ Enable resellers/agents to place orders for items and select a preferred payment method, such as cash, bank transfer, or Momo. They should also be able to make online payments and view the status of their orders. This functionality should be implemented through a Web form interface.

1.2 Product Overview

**Capabilities for Accountant’s:**

- Creating Goods Received: Accountants will be able to generate warehouse receipts when the distributor imports goods. The capability includes handling multiple items within a single warehouse receipt. This functionality can be accessed through both Win Form and Web Form interfaces.

- Managing Goods Delivery: Accountants will have the ability to create Goods Delivery Notes for delivering goods to agents. They can print delivery slips for the shipments. Additionally, accountants can update the status of orders during the transfer process and update the payment status of agents. These capabilities will be available through both Win Form and Web Form interfaces.

- Generating Reports: Accountants will be able to view incoming/outgoing stock reports, identify the best-selling products, and generate monthly revenue reports. These reports can be accessed through Form or Report interface.

**Scenarios for Resellers/Agents:**

**-** Placing Orders: Resellers/Agents can place orders for items they require. They will have the option to choose from different payment methods such as cash, bank transfer, or Momo. Additionally, they can make online payments and track the status of their orders. This functionality will be provided through a Web form interface.

1.3 Structure of the Document

In this document, we will walk through all the essential steps before implementing the system. First of all, we will discuss our plan for project management, then we will look into details of requirement specifications (draw models). In the last two steps, we will choose Architecture and design our database based on the plan and requirement analysis that we make at the beginning of this document.

1.4 Terms, Acronyms, and Abbreviations

|  |  |  |
| --- | --- | --- |
| STT | Item | Description |
| 1 | MVC | Model-View-Controller |
| 2 | HTTP | Hypertext-Transfer Protocol |
| 3 | HTML | File CSHTML in C# |
| 4 | EF6 | Entity framework |
| 5 | RDLC | Report Definition Language Client-side an extension of the report file created by using Microsoft reporting technology |
| 6 | ERD | Entity Relationship Diagram |

2. Project Management Plan

2.1 Project Organization

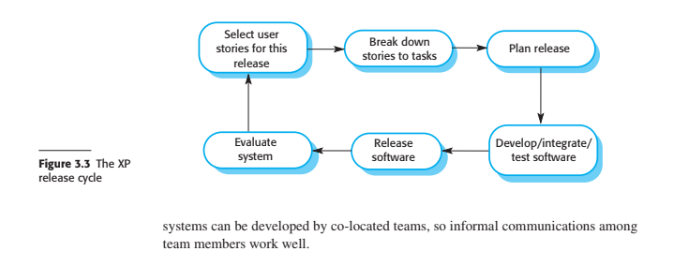
We discussed the assignment on the Discord application for analyzing the task and creating a plan. The first step is to analyze the requirements and summarize the key points. Then, the team will design the database based on the previously summarized content, including inserting data into tables and establishing relationships between them.

In the second step, we will design various diagrams (Sequence, Activity, Class) for the project. Next, we will implement the code in C# based on the designed diagrams to execute Winform and Webform. After completing the conceptualization and implementation of the code, we will test the functionalities using test cases and create an automated unit test. Finally, the product will be released.

Please note that the translation may not be a literal translation but aims to convey the overall meaning accurately.

2.2 Lifecycle Model Used

We use Agile because throughout the process of analysis and design implementation, there are always changes to the database, diagrammatic representations, and other aspects.

****

2.3 Risk Analysis

During the usage of our system, there are potential risks that need to be considered. While some of these risks can be identified and resolved promptly during the implementation and testing phase, there are also hidden or external risks that pose a significant threat to the system. We have encountered certain risks during testing, such as the possibility of Agents purchasing quantities that exceed the available stock in the warehouse, or instances where Accountants or Agents input invalid IDs or product codes into the system. Additionally, a common risk involves query errors in the database, which can cause the system to abruptly halt. These risks mainly stem from technical issues or a lack of user familiarity with the system. To mitigate these risks, we have implemented a testing approach during the implementation phase to proactively detect and address errors early on.

2.4 Hardware and Software Resource Requirements

For hardware, we use chip Intel ® Core(TM) i5 260GB storage RAM 8GB. For software, we use mostly Visual Studio 2022 C#, Microsoft Server Management Studio 2019.

Through this project, we learn more about three-tier of MVC Model and framework help to team solution for requirements.

2.5 Deliverables and Schedule

Use case model, Use case description, Graphical use case model, Architectural model: 18/04/2023 to 23/04/2023

Functional requirement , Non-functional requirements, Database design, Static-Model (Class digram), Dynamic-Model (Sequence diagram) 24/04/2023 to 26/04/2023

Implementation code C#: 27/04/2023 to 6/05/2023

Testing: 6/05/2023 to 7/05/2023

2.6 Monitoring, Reporting and Controlling Mechanisms

**Monitoring Goods Received:**

- Implement a monitoring system to track and record the goods received by the distributor when importing items.

- Create a mechanism for accountants to generate warehouse receipts that include multiple items.

- Ensure the functionality is available through both Win Form and Web Form interfaces.

**Monitoring Orders and Payments:**

- Develop a web-based feature for resellers/agents to place orders for items and choose payment methods such as cash, bank transfer, or Momo.

- Enable online payment options and provide a mechanism for resellers/agents to check the status of their orders.

- Implement a monitoring system to track and update the status of orders and payment information.

**Controlling Goods Delivery:**

- Enable accountants to create Goods Delivery Notes for delivering goods to agents.

- Implement the functionality to print delivery slips for the goods being transferred.

- Provide options to update the status of orders as they are being transferred and update the payment status of agents.

- Ensure this functionality is accessible through both Win Form and Web Form interfaces.

**Reporting and Analysis:**

- Develop reporting features for accountants to generate reports on incoming and outgoing stock.

- Implement functionality to generate reports on best-selling products and monthly revenue.

- Ensure the availability of forms and reports for reviewing stock statistics, top-selling items, and monthly revenue.

2.7 Professional Standard

-A meeting will be scheduled to assess the situation and find a solution. If a team member is dismissed, their grade will be reduced according to the number of weeks they were a part of the group.

- Maintain devotion and commitment through the whole project.

- Actively work on the project

2.8 Evidence all the artifacts have been placed under configuration management

- Creating folders to store important data

- Setting fixed deadlines for each team member and task

- Discussion to handle work

- Get the job done

- Impact on Accountants:

2.9 Impact of the project on individuals and organizations

- Impact on Accountants:

**+ Increased efficiency:** The software enables accountants to create Goods Received and Goods Delivery Note electronically, reducing manual paperwork and streamlining the process of importing and delivering goods.

**+ Enhanced accuracy:** With the system, accountants can accurately track and manage the inventory, ensuring that the quantities received, transferred, and delivered align with the records.

**+ Improved reporting:** The ability to generate reports on incoming/outgoing stock, best-selling products, and revenue on a monthly basis empowers accountants with valuable insights for decision-making and financial analysis.

- Impact on Resellers/Agents:

**+ Simplified ordering process:** Resellers/agents can easily place orders for items through the software, selecting their preferred payment methods such as cash, bank transfer, or online payment (e.g., Momo). This simplifies the ordering process and offers flexibility in payment options.

**+ Real-time order tracking:** Resellers/agents can view the status of their orders in real-time, allowing them to monitor the progress and take necessary actions accordingly.

**+ Improved transparency:** The software provides visibility into the availability of stock, enabling resellers/agents to make informed decisions when placing orders.

- Organizational Impac

+ **Streamlined operations:** The software centralizes and automates various processes related to inventory management, order placement, and financial transactions, leading to streamlined operations and reduced administrative overhead.

+ **Enhanced customer satisfaction:** With improved efficiency, accuracy, and transparency, the distributor can better serve their resellers/agents, resulting in improved customer satisfaction and loyalty.

+ **Data-driven decision-making:** The availability of comprehensive reports on stock, sales, and revenue empowers the organization to make data-driven decisions, identify trends, and optimize business strategies.

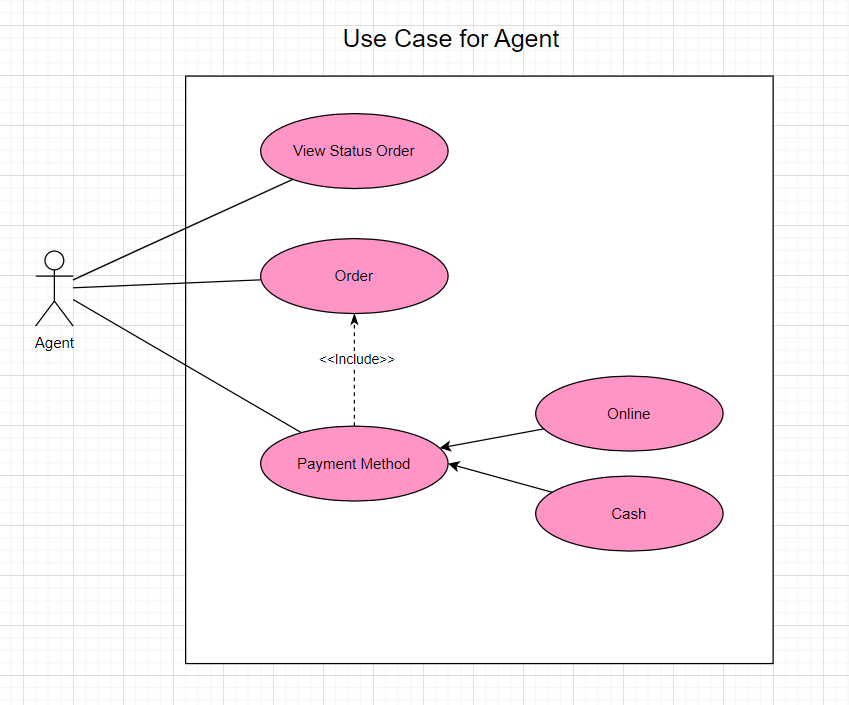
3. Requirement Specifications

3.1 Stakeholders for the system

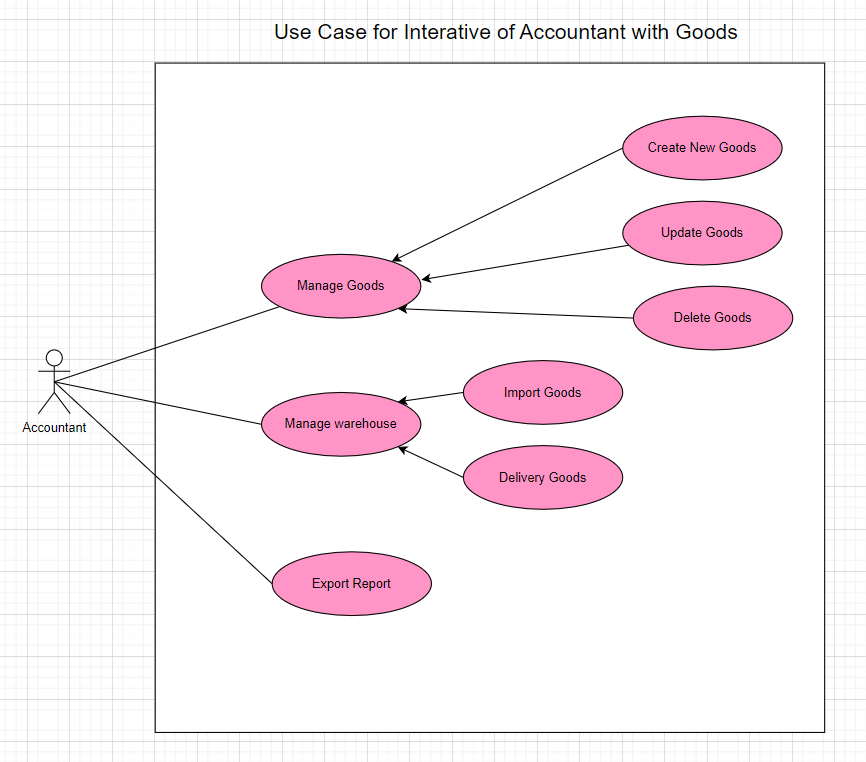
Distributor, Reseller/Agents, Accountants, Organization

3.2 Use case model

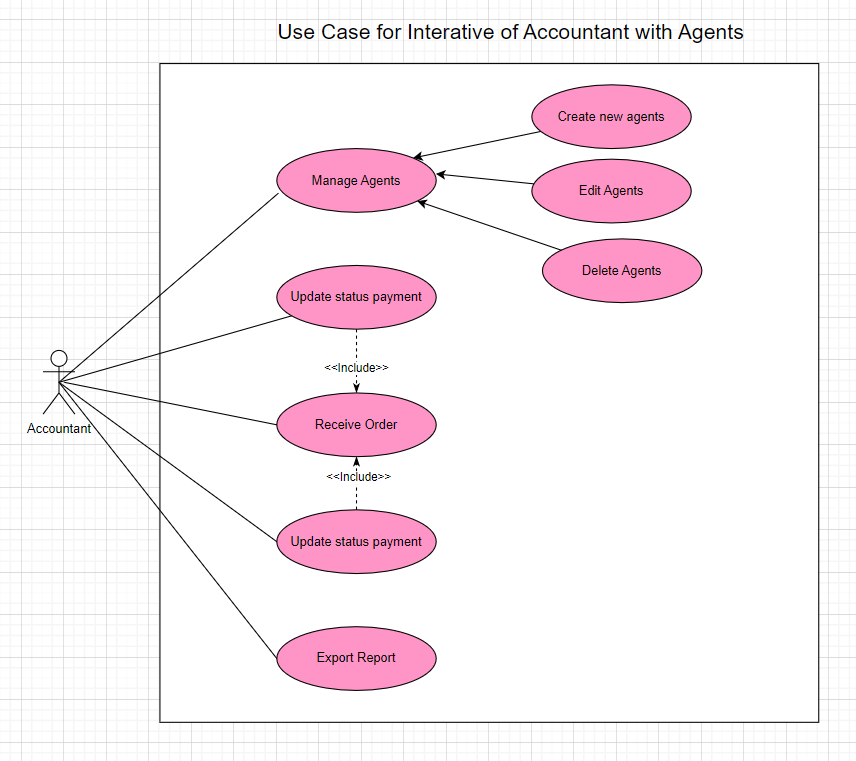
**3.2.1 Graphical use case model**

****

Picture 1.1: Use case Agents

****

Picture 1.2: Use case Accountant with Goods

****

Picture 1.3: Use case Accountant with Agents

**3.2.2 Textual Description for each use case**

**USE CASE DESCRIPTION: AGENTS**

|  |  |  |
| --- | --- | --- |
| **Use case name:** | Order Management | |
| **Scenario:** | A customer wants to view the status of their order, place a new order, or choose a payment method. | |
| **Triggering event:** | The customer wants to manage their order. | |
| **Brief description:** | This use case describes the process of managing orders, including viewing the status of existing orders, placing a new order, and choosing a payment method. | |
| **Actors:** | Customer: The person who wants to manage their order.  System: The software application that provides order management functionality. | |
| **Related use cases:** | Might be View Order Status, Place Order, Manage Payment | |
| **Stakeholders:** | Customer: The primary stakeholder who benefits from the successful management of their orders.  Business: The business that sells products and fulfills orders benefits from efficient and accurate order management. | |
| **Preconditions:** | The customer has an account with the business and has placed at least one order.  The system is operational and accessible to the customer. | |
| **Postconditions:** | The customer has successfully viewed the status of an existing order, placed a new order, or chosen a payment method.  The system has updated the order status or created a new order. | |
| **Flow of activities:** | **Actors** | **System** |
| 1. The customer logs into their account.  2. The customer selects the "Manage Orders" option from the menu.  3. If the customer selects "View Order Status,"  4. If the customer selects "Place Order,"  5. If the customer selects "Manage Payment," | 1.1 The system authenticates the customer and displays their account information.  2.1 The system presents the options to view the status of an existing order, place a new order, or choose a payment method.  3.1 The system displays a list of their existing orders and their current status.  4.1 The system guides them through the order placement process, including selecting products, entering payment and shipping information, and confirming the order.  5.1 the system presents payment options and processes the payment according to the selected method. |
| **Exception conditions:** | 1.1 If the customer enters incorrect login credentials, the system displays an error message and prompts the customer to try again.  2. 1 If the system is unavailable, the customer is unable to manage their orders and may receive an error message.  4. 1 If there are issues processing the payment, the system displays an error message and prompts the customer to try another payment method. | |

Picture 1.4: Use case Description for Agents

**USE CASE DESCRIPTION: ACCOUNTANT WITH GOODS**

|  |  |  |
| --- | --- | --- |
| **Use case name:** | Goods Management | |
| **Scenario:** | An accountant wants to manage the company's goods and warehouses. | |
| **Triggering event:** | The accountant selects the "Goods management" function on the system interface. | |
| **Brief description:** | This use case allows the accountant to manage information about goods, including creating, updating, and deleting items, as well as importing and exporting from warehouses. | |
| **Actors:** | Accountant, System | |
| **Related use cases:** | Might be Warehouse management, Export report | |
| **Stakeholders:** | Accountant, Warehouse manager, Company management | |
| **Preconditions:** | The accountant has logged into the system.  The system has stored information about goods and warehouses. | |
| **Postconditions:** | Information about goods and warehouses has been successfully managed and stored. | |
| **Flow of activities:** | **Actors** | **System** |
| 1. The accountant selects the "Goods management" function on the system interface.  2. The accountant can create, update, or delete goods as requested.  3. If the accountant wants to manage the warehouse, they select the "Warehouse management" function on the system interface.  4. The accountant can import or export goods from the warehouse as requested.  5. If the accountant wants to export a report about goods orwarehouses, they select the "Export report" function on the system interface. | 1.1 The system displays the list of current goods.  3.1 The system displays the list of current warehouses.  5.1 The system exports the report and provides it to the accountant. |
| **Exception conditions:** | 1.1 If the accountant cannot log into the system, the system will display an error message and require them to try again or contact support.  2.1 If the system cannot update or store information about goods or warehouses, the system will display an error message and require the accountant to try again or contact support. | |

Picture 1.5: Use case Description for Accountant with Goods

**USE CASE DESCRIPTION: ACCOUNTANT WITH AGENT**

|  |  |  |
| --- | --- | --- |
| **Use case name:** | Agent Management | |
| **Scenario:** | An accountant wants to manage information about agents, orders, and payments of agents. | |
| **Triggering event:** | The accountant selects the "Agent Management" function on the system interface. | |
| **Brief description:** | This use case allows the accountant to manage information about agents, including creating new agents, updating and deleting existing agents. It also enables the accountant to manage orders and payments of agents by updating the payment status of orders. | |
| **Actors:** | Accountant, System | |
| **Related use cases:** | Might be Order Management, Export Report | |
| **Stakeholders:** | Accountant, Agents, Company management | |
| **Preconditions:** | The accountant has logged into the system.  The system has stored information about agents. | |
| **Postconditions:** | The information about agents, orders, and payments of agents has been successfully managed and stored. | |
| **Flow of activities:** | **Actors** | **System** |
| 1. The accountant selects the "Agent Management" function on the system interface.  2. The accountant can create new, update or delete agents as required.  3. If the accountant wants to manage orders of an agent, they select the "Order Management" function on the system interface.  4. The accountant can update the payment status of orders as required.  5. If the accountant wants to export a report about agents or orders of agents, they select the "Export Report" function on the system interface. | 1.1 The system displays the list of existing agents.  3.1 The system displays the list of orders of the selected agent.  5.1 The system exports the report and provides it to the accountant. |
| **Exception conditions:** | 1.1 If the accountant cannot log in to the system, the system will display an error message and ask them to try again or contact the support department.  2.1.1 If the system cannot update or store information about agencies, orders, or payments, the system will display an error message and ask the accountant to try again or contact the support department.  2.1.2 If the accountant wants to create a new agency or order but has entered incorrect information, the system will display an error message and ask the accountant to update the information.  3. 1 If the accountant wants to view detailed information about an agency or order but does not have access rights, the system will display an error message and not allow them to view that information. | |

Picture 1.6: Use case Description for Accountant with Agents

3.3 Functional requirements

- The system allows users to easily change the server name and database. And the system has the ability to check the existence of the database before using it.

- The system has function Login for check account user.

- The system will show the detail of goods for all winform and webform(use for order goods or delivery).

- The function have check the validation of information before interacted with database.

**- When login as Admin system:**

+ The admin can manager the accountant (use for manage in company want use app manager).

**- When login as accountant:**

+ Can manage Goods (Add, edit, delete).

+Can manage Agent (Create new Agent, edit, delete).

+ Can manage order for delivery or goods received.

+ Can manage status for order of delivery and status for payment of agent .

+ Can manage the warehose: the quantity of each goods.

+ Can view the report for goods (warehouse), sales of month.

**- When login as Agent:**

+ Agent can create request order

+ Agent can select payment method

+ Views the list order

3.4 Non-functional requirements

- Response time: fastly, refresh the list of item when have event update occur

- Security: The system must ensure the safety and security of customer information and payment data during the transaction.

- Ease of use: The system must be designed to simplify the ticket purchase and payment process for users of all ages and levels.

- Scalability: The system must be scalable to be able to serve a large number of users at the same time.

- Maintainability: The system must be easy to maintain and repair when problems occur to ensure continuous operation.

4. Architecture

4.1 Architecture style(s) used

- In this project, we employ a data-centric architecture. The approach of this architecture is to design the data first and then build and utilize applications based on that data. Our system will access data directly from its source. We support a business model where application providers can be rewarded for adding value to the system.

4.2 Architectural model

The architectural model chosen to meet the requirements of a Distributor selling Mobile Phones products to authorized resellers/agents is the Client-Server model. This model divides the system into two main components: the client and the server.

The client component consists of the user interface and application used by the Accountants, Resellers/Agents, and other users. It allows the Accountants to create Goods Received by generating warehouse receipts for imported goods. This functionality can be developed using Win Form or Web Form technology. The Resellers/Agents can place orders for items, select payment methods (such as Cash, bank transfer, Momo), make online payments, and track the status of their orders. This functionality can be implemented using a web-based form (Webform).

The server component is responsible for data storage, processing, and business logic. The Accountants can create Goods Delivery Notes to deliver goods to agents, update the status of orders as they are being transferred, and update the payment status of agents. This functionality can be implemented using Win Form or Web Form technology. The Accountants can also view reports on incoming/outgoing stock, best-selling products, and monthly revenue. These reports can be generated using forms or reports.

Overall, the Client-Server architectural model allows for effective communication between the client-side and server-side components, enabling the required functions for the Distributor and its authorized resellers/agents.

4.3 Technology, software, and hardware used

For hardware, we use chip Intel ® Core(TM) i5 260GB storage RAM 8GB. For software, we use mostly Visual Studio 2022 C#, Microsoft Server Management Studio 2019.

4.4 Rationable

The mentioned functions are crucial for streamlining operations, enhancing communication, improving inventory management, and facilitating financial analysis in the distribution of mobile phone products to authorized resellers/agents.

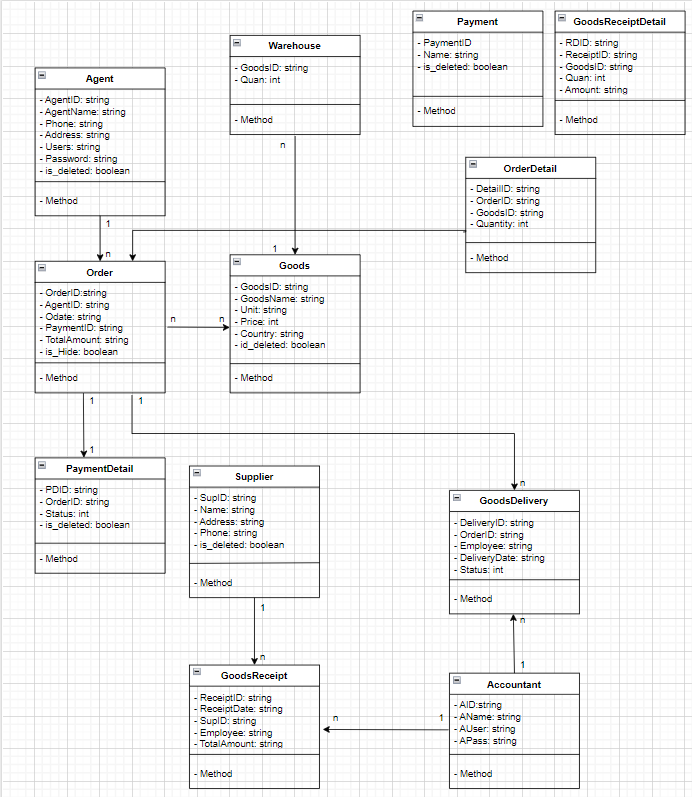
5. Design

5.1 Database design



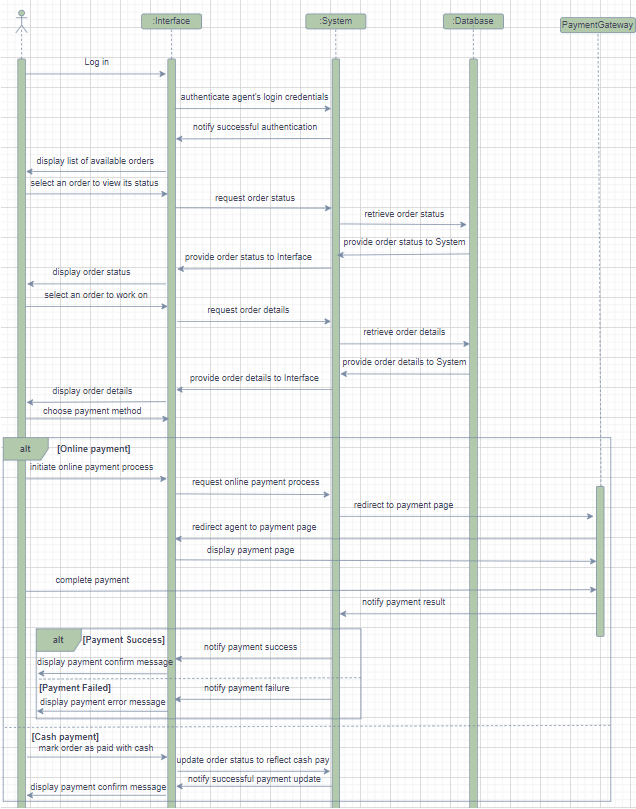
Picture 1.7: Database

5.2 Static model – class diagram



Picture 1.8: Static model – Class diagram

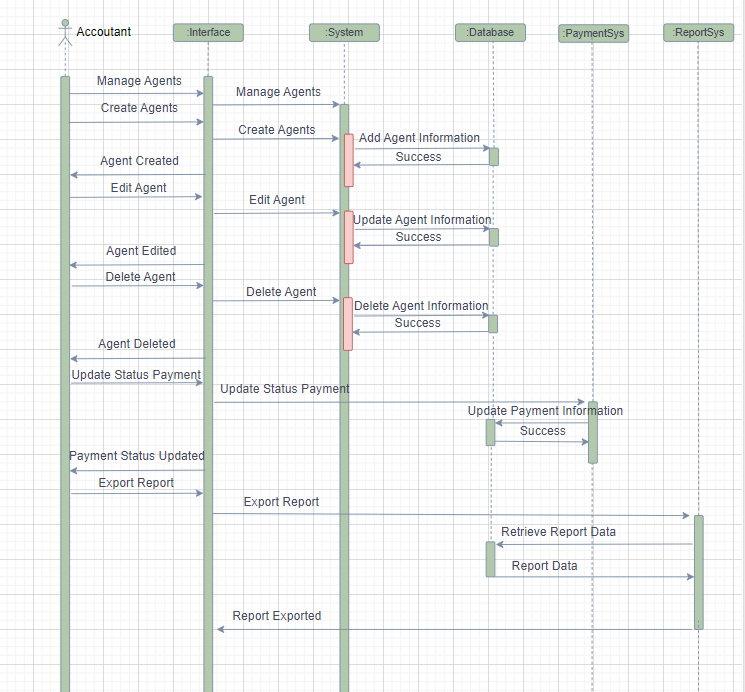
5.3 Dynamic model – sequence diagram



Picture 1.9: Dynamic model – Sequence - Agents diagram



Picture 1.10: Dynamic model – Sequence – Accountant with Goods diagram



Picture 1.11: Dynamic model – Sequence – Accountant with Agents diagram

6. Test Plan

6.1 Requirements/specifications-based system level test cases

**Test Case: Goods Received Creation**

- Description: Verify that Accountants can successfully create Goods Received when the distributor imports goods.

- Precondition: Accountant is logged into the system.

- Steps:

+ Accountant selects the option to create Goods Received.

+ Accountant enters the details of the imported items in the warehouse receipt.

+ Accountant submits the form.

- Expected Result: Goods Received record is successfully created with the specified items.

- Test Case: Order Placement and Payment

- Description: Verify that Resellers/Agents can place an order, choose a payment method, make online payment, and view order status.

- Precondition: Reseller/Agent is logged into the system.

- Steps:

+ Reseller/Agent selects the option to place an order.

+ Reseller/Agent adds items to the order list and selects a payment method (Cash, bank transfer, Momo, etc.).

+ Reseller/Agent proceeds to make an online payment.

+ Reseller/Agent checks the order status.

- Expected Result: Order is successfully placed, payment is made, and order status is visible to the Reseller/Agent.

**Test Case: Goods Delivery Note Creation**

- Description: Verify that Accountants can create Goods Delivery Note, update order status, and payment status for delivering goods to agents.

- Precondition: Accountant is logged into the system.

- Steps:

+ Accountant selects the option to create Goods Delivery Note.

+ Accountant enters the necessary details and prints the delivery slips.

+ Accountant updates the order status as "being transferred" and updates the payment status for agents.

Expected Result: Goods Delivery Note is created, order status is updated, and payment status is recorded accurately.

**Test Case: Reports Viewing**

- Description: Verify that Accountants can view incoming/outgoing stock report, best-selling products, and monthly revenue report.

- Precondition: Accountant is logged into the system.

- Steps:

+ Accountant selects the option to view reports.

+ Accountant selects the desired report type (incoming/outgoing stock, best-selling products, monthly revenue).

+ Accountant generates and views the report.

- Expected Result: The requested report is displayed correctly, providing accurate information on stock, sales, and revenue.

6.2 Traceability of test cases to use cases

- Creating Goods Receipt: Linked to the use case "Accountant creates Goods Receipt when the company imports goods." This test case ensures that the Goods Receipt creation function works correctly and meets the requirements of the accountants.

- Placing Orders and Payment: Linked to the use case "Reseller places an order, selects a payment method, and checks the order status." This test case ensures that the order placement, payment, and order status checking functions work accurately for the Reseller.

- Creating Goods Delivery Note: Connected to the use case "Accountant creates Goods Delivery Note, updates order status, and payment status." This test case ensures that the Goods Delivery Note creation function works properly and accurately updates the order status and payment status.

- Viewing Reports: Linked to the use case "Accountant views incoming/outgoing stock reports, best-selling products, and monthly revenue report." This test case ensures that the Accountant can generate and view the desired reports correctly, providing accurate information on stock, sales, and revenue.

6.3 Techniques used for test generation

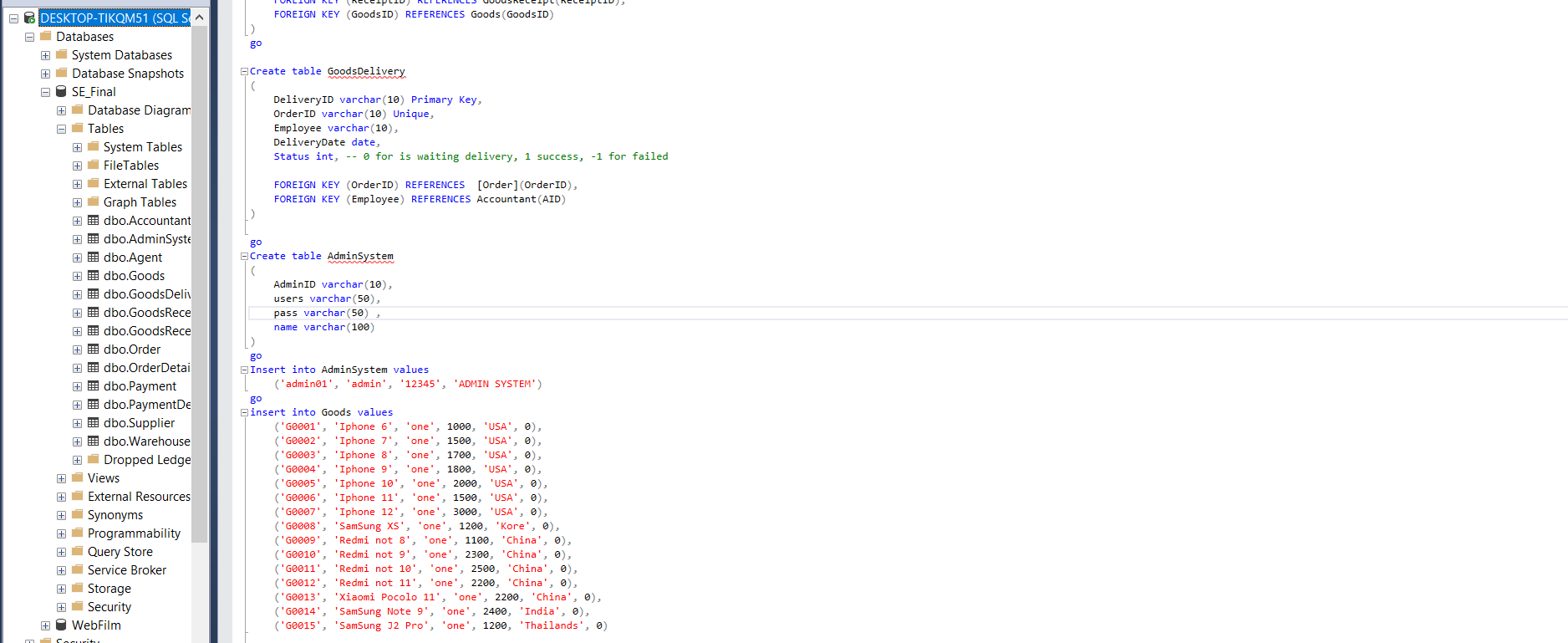
- To generate test cases, we have used techniques such as black-box testing, white-box testing, and integration testing. By employing these methods, we can test the correctness and performance of the software's functionalities.

6.4 Assessment of the goodness of your testsuite

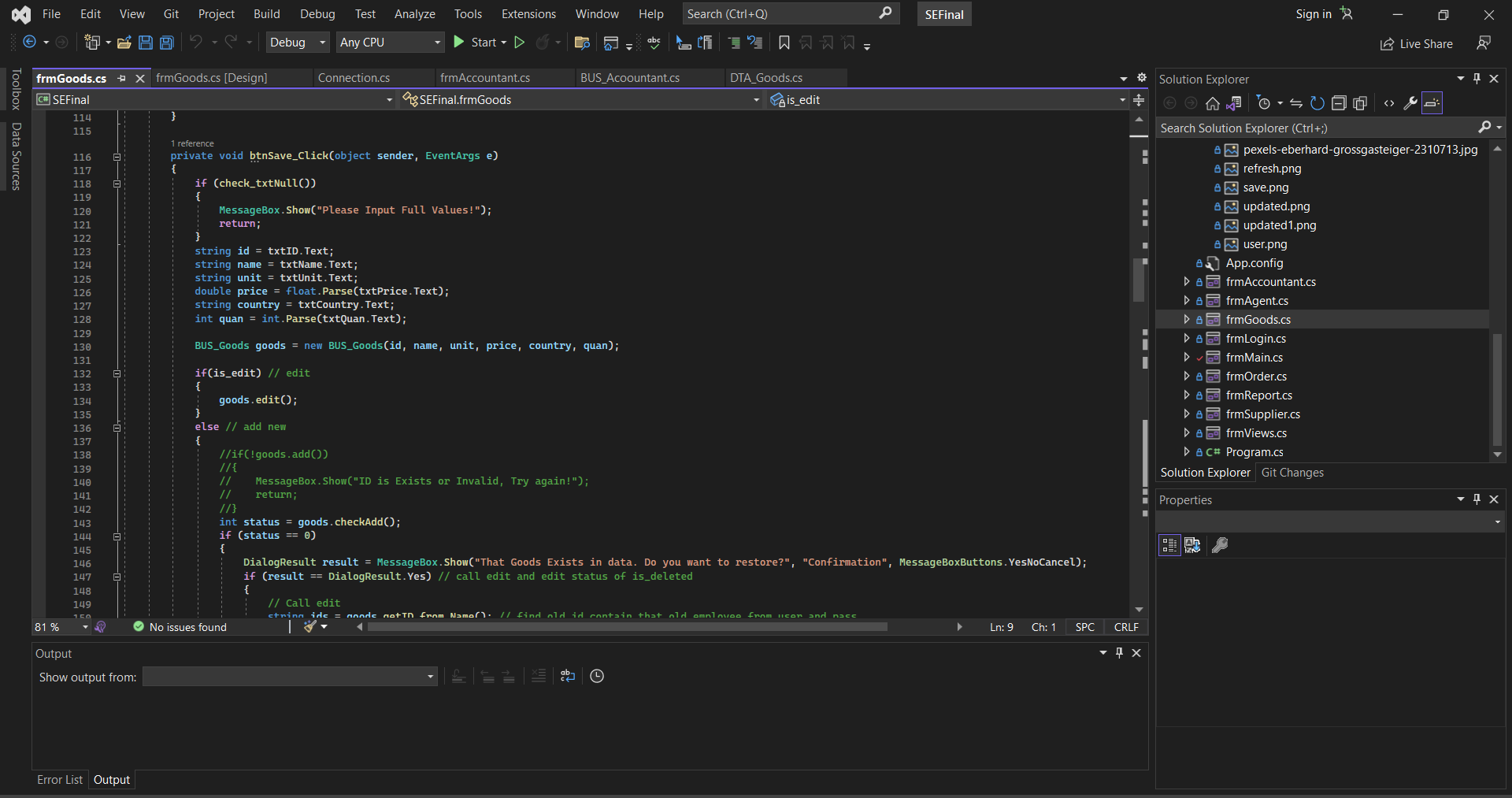
- To evaluate the quality of the test suite, we have used metrics such as code coverage and defect detection rate. Code coverage indicates the percentage of source code that has been tested by the test cases. Defect detection rate measures the test suite's ability to detect errors in the software

7. Demo

**7.1** Database



7.2 Source code



**REFERENCES**

**Vietnamese**

**English**

Introducing by Master. Pham Thai Ky Trung

1. Software Engineering – TENTH EDITION – Ian Sommerville
2. Introduction to Software Engineering – Second Edition
3. TDTU – Slides Chapter 2 – Software Process
4. TDTU – Slides Chapter 6 – Alige SW Dev

**PHỤ LỤC**