VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL PROJECT**

**SOFTWARE ENGINEERING**

*Student*: **NGUYỄN THẠCH TRƯỜNG LẠC –** **521H0462**

**TRẦN TRỌNG KHOA – 521H0458**

Class **: 21H50202**

Year  **: HK2 / 2022-2023**

*Supervisor*:  **Dr. PHẠM THÁI KỲ TRUNG**

**HO CHI MINH CITY, 2023**

VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL PROJECT**

**SOFTWARE ENGINEERING**

*Student*: **NGUYỄN THẠCH TRƯỜNG LẠC –** **521H0462**

**TRẦN TRỌNG KHOA – 521H0458**

Class **: 21H50202**

Year  **: HK2 / 2022-2023**

*Supervisor*: **Dr. PHẠM THÁI KỲ TRUNG**

**HO CHI MINH CITY, 2023**

THANK YOU

Dear Dr. Phạm Thái Kỳ Trung,

First and foremost, with utmost sincerity and deep gratitude, allow our team to express our appreciation to you for providing the support and assistance throughout our academic journey and research on this topic. The internship report was conducted over a period of nearly 2 weeks. At the initial stage of our practical experience, we were still limited and encountered some uncertainties, inevitably resulting in some shortcomings. We sincerely hope to receive your valuable feedback and suggestions to improve our knowledge in this field and enhance our awareness.

Please accept our sincere thanks and appreciation for your supporting work. You are a true professional, and it has been an honor to study with you.

With heartfelt thanks,

*Trần Trọng Khoa*

*Nguyễn Thạch Trường Lạc*

**PROJECT COMPLETED**

**AT TON DUC THONG UNIVERSITY**

I hereby declare that this is my/our own project and is under the guidance of Dr. Bui Thanh Hung. The research contents and results in this topic are honest and have not been published in any publication 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 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 (if any).

*Ho Chi Minh, 7/5/2023*

*Author*

*Nguyễn Thạch Trường Lạc*

*Trần Trọng Khoa*

LECTURER’S ASSESSMENT

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

*Ho Chi Minh, 7/5/2023*

*(sign)*

SUMMARY

PhoneHub is a software offers a comprehensive solution for businesses engaged in phone sales and distribution. The software provides a streamlined and efficient approach to managing inventory, orders, and agents.

Through PhoneHub, you can easily add products to your inventory, monitor stock levels, and handle product pricing. Additionally, you have the ability to create and track customer orders, manage order statuses, and monitor payment progress. Our software includes a comprehensive dashboard that provides detailed insights and data on sales trends, top-selling products, and other vital metrics.

One notable feature of our software is the agent management system, which simplifies the process of adding and managing agents. You can effortlessly track agent performance, including sales figures, commissions paid, and other significant metrics. Furthermore, you can provide secure logins to your agents, granting them access to performance updates, profile management, and relevant information on orders and products.

Overall, our mobile distribution software is an ideal tool for businesses seeking to optimize their distribution process, save time, and enhance productivity. With its user-friendly interface and extensive range of features, it ensures efficient management of sales and inventory.

CONTENT

Contents

[THANK YOU 1](#_Toc134654936)

[LECTURER’S ASSESSMENT 3](#_Toc134654937)

[SUMMARY 4](#_Toc134654938)

[CONTENT 5](#_Toc134654939)

[LIST OF TABLES, FIGURES, GRAPHS 8](#_Toc134654940)

[**1. INTRODUCTION** 9](#_Toc134654941)

[**1.1** **Purpose and Scope** 9](#_Toc134654942)

[**1.2** **Product Overview** 9](#_Toc134654943)

[**1.3** **Structure of the Document** 10](#_Toc134654944)

[**1.4** **Terms, Acronyms, and Abbreviations** 10](#_Toc134654945)

[**2** **Project Management Plan** 10](#_Toc134654946)

[**INTRODUCTION** 10](#_Toc134654947)

[**2.1** **Project Organization** 11](#_Toc134654948)

[**2.2** **Lifecycle Model Used** 11](#_Toc134654949)

[**2.3** **Risk Analysis** 11](#_Toc134654950)

[**2.4** **Hardware and Software Resource Requirements** 12](#_Toc134654951)

[**2.5** **Deliverables and Schedule** 12](#_Toc134654952)

[**2.6** **Monitoring, Reporting, and Controlling Mechanisms** 13](#_Toc134654953)

[**2.7** **Professional Standards** 13](#_Toc134654954)

[**2.8** **Evidence the Document has been placed under configuration management** 13](#_Toc134654955)

[**2.9** **Impact of the project on individuals and organizations** 13](#_Toc134654956)

[**3** **Requirement Specifications** 14](#_Toc134654957)

[**3.1** **Stakeholders for the system** 14](#_Toc134654958)

[**3.2** **Use case model** 14](#_Toc134654959)

[**3.2.1** **Graphical use case model** 14](#_Toc134654960)

[**3.2.2** **Textual Description for each use case** 17](#_Toc134654961)

[**3.3** **Functional requirements** 19](#_Toc134654962)

[**3.4** **Non-functional requirements** 20](#_Toc134654963)

[**4** **Architecture** 21](#_Toc134654964)

[**4.1** **Architectural style(s) used** 21](#_Toc134654965)

[**4.2** **Architectural model** 22](#_Toc134654966)

[**4.3** **Technology, software, and hardware used** 23](#_Toc134654967)

[**4.4** **Rationale for your architectural style and model** 23](#_Toc134654968)

[**5** **Design** 23](#_Toc134654969)

[**5.1** **Database design** 23](#_Toc134654970)

[**5.2** **Static model – class diagrams** 24](#_Toc134654971)

[**5.3** **Dynamic model – sequence diagrams** 25](#_Toc134654972)

[**5.4** **Rationale for your detailed design model** 25](#_Toc134654973)

[**5.5** **Traceability from requirements to detailed design model** 26](#_Toc134654974)

[**6** **Test Plan** 27](#_Toc134654975)

[**6.1** **Requirements/specifications-based system level test cases** 27](#_Toc134654976)

[**6.1.1** **Test Case: Goods Received Creation** 27](#_Toc134654977)

[**6.1.2** **Test Case: Order Placement and Payment** 27](#_Toc134654978)

[**6.1.3** **Test Case: Goods Delivery and Order Status Update** 28](#_Toc134654979)

[**6.1.4** **Test Case: Stock and Sales Reports** 28](#_Toc134654980)

[**6.2** **Traceability of test cases to use cases** 29](#_Toc134654981)

[**6.3** **Techniques used for test generation** 30](#_Toc134654982)

[**6.4** **Assessment of the goodness of your testsuite (Which metrics were used for such assessment?)** 31](#_Toc134654983)

[**7** **Demo** 31](#_Toc134654984)

[**7.1** **Database** 31](#_Toc134654985)

[**7.2** **Source code** 32](#_Toc134654986)

[**7.3** **Testing** 32](#_Toc134654987)

[REFERENCE 33](#_Toc134654988)

LIST OF TABLES, FIGURES, GRAPHS

TABLES

Table 1 Deliverables and Schedule

Table 2 Import goods Des.

Table 3 Export goods Des.

Table 4 Statistical goods Des.

Table 5 Technology, software, and hardware used.

**FIGURES**

Fig 1 Use case of Import goods

Fig 2 Use case of Export goods

Fig 3 Use case of Statistical

Fig 4 Class Diagram

Fig 5 Sequence Diagram

**1. INTRODUCTION**

* 1. **Purpose and Scope**
* PhoneHub is a software offers a comprehensive solution for businesses engaged in phone sales and distribution. The software provides a streamlined and efficient approach to managing inventory, orders, and agents.
* Through PhoneHub, you can easily add products to your inventory, monitor stock levels, and handle product pricing. Additionally, you have the ability to create and track customer orders, manage order statuses, and monitor payment progress. Our software includes a comprehensive dashboard that provides detailed insights and data on sales trends, top-selling products, and other vital metrics.
* One notable feature of our software is the agent management system, which simplifies the process of adding and managing agents. You can effortlessly track agent performance, including sales figures, commissions paid, and other significant metrics. Furthermore, you can provide secure logins to your agents, granting them access to performance updates, profile management, and relevant information on orders and products.
* Overall, our mobile distribution software is an ideal tool for businesses seeking to optimize their distribution process, save time, and enhance productivity. With its user-friendly interface and extensive range of features, it ensures efficient management of sales and inventory.
  1. **Product Overview**

- Currently, the demand for consumer electronics and the buying and selling of mobile phone products is very high. A software solution like PhoneHub is highly necessary. This software provides numerous intelligent features and utilities compared to similar software available in the market. To closely manage the supply chain and understand consumer needs, creating an interactive environment among retailers, distributors, and consumers, the emergence of this software is a natural progression.

* 1. **Structure of the Document**

- The structure of the document consists of 7 sections:

1. Introduction

2. Project Management Plan

3. Requirement Specifications

4. Architecture

5. Design

6. Test Plan

7. Demo

- *Note*: The details of each section are mentioned in the content.

* 1. **Terms, Acronyms, and Abbreviations**
* HTTP: Hypertext-Transfer Protocol
* HTML: Hypertext Mark-up Language
* MVC: Model – View – Control
* DAO: Data Access Object, this object is responsible for attaching to a

system, extracting some information, based on specific

requirements, and creating a value object.

* API: Application Programming Interface

1. **Project Management Plan**

**INTRODUCTION**

* The Project Management Plan section provides an overview of the project's management approach and outlines various aspects related to project organization, risk analysis, resource requirements, deliverables and schedule, monitoring and controlling mechanisms, professional standards, configuration management, and the project's impact on individuals and organizations.

1. **Project Organization**

* Project Manager/Team Lead: Lạc is team leader. He has responsible for overall project coordination, task assignment, and ensuring that the project stays on track.
* Development: Both two of us have responsible for designing, coding, and testing the software.
* Roles and Responsibilities: Lạc will primarily be responsible for the front-end portion, while Khoa will take responsibility for the back-end part. The functionalities will be discussed and allocated in a reasonable manner.
* Communication and Collaboration: We have regular meetings with each other through Discord. Occasionally, when we have free time, we will meet up at a coffee shop and have discussions.
* Time Management: The deadline we have set is 2 weeks. It may vary depending on various factors such as difficulty, complexity,...
* Documentation: We all contribute to the documentation based on the materials provided by the instructor on Classroom and other collected resources. This ensures knowledge transfer and facilitates software maintenance and updates in the future.

1. **Lifecycle Model Used**

* With a small team of 2 students and a limited timeframe of 2 weeks, it is important to choose a lifecycle model that allows for efficient and effective development within the given constraints. In this case, we choosed an Agile model (Scrum) because of its flexibility, adaptability, and focus on delivering value incrementally.

1. **Risk Analysis**

* We have identified that there is a risk that may affect our project. It is related to the Web-form design. We usually work on Win-form, so transitioning to Web-form may cause difficulties for us in its implementation and development.
* We have searched for relevant documentation to find the best possible solution. Additionally, we have contacted Professor Phạm Thái Kỳ Trung, who has helped us clarify some of the issues we encountered.

1. **Hardware and Software Resource Requirements**

**-** Hardware Requirements:

• We should have a personal laptop or computer with adequate processing power, memory, and storage capacity.

• The devices should be capable of running the necessary development tools and software smoothly.

• Reliable internet connectivity is essential for communication, research, and accessing online resources.

- Software Requirements:

• An Integrated Development Environment (IDE) suitable for the chosen programming language or framework is required. This will provide a convenient development environment with code editing, debugging, and compilation features. Both of used Visual Studio Code.

• A Database Management System (DBMS) is needed for storing and retrieving project data efficiently. Both of used SQL Server Management Studio Management Studio 19.

• Web development tools and frameworks are necessary for front-end and back-end development: HTML, CSS, JavaScript, php,….

• Testing tools: We used Test functions in Visual Studio 2022.

• Communication and collaboration tools: We used Facebook, Zalo, Discord and Google Drive to communicate or contact.

1. **Deliverables and Schedule**

|  |  |  |
| --- | --- | --- |
| Activities | Estimated time | Person responsible |
| Front-end of 2 Winform | 2 days | Lạc |
| Back-end of 2 Winform | 2 days | Khoa |
| Web-form | 5 days | Lạc, Khoa |
| Database | 1 days | Lạc |
| Report | 2 days | Khoa |
| Pptx file | 2 days | Lạc |

Table 1 Deliverables and Schedule

1. **Monitoring, Reporting, and Controlling Mechanisms**

* We have two management reports that need to be created: a development progress report and a Word report completion report. For the development progress report, we use Google Drive for storage. The report file will be accompanied by a Readme.txt file to update any edits and changes made to the software. As for the Word report, we use Google Docs for easy collaboration, control of edits, and monitoring of access to the file.

1. **Professional Standards**

* Lac is a very professional team leader. He is always on time for meetings as well as adhering to the deadline schedule. He is very good at motivating the team when we encounter difficulties. This helps to make the teamwork more effective and productive. Khoa is a very positive member who brings out many new ideas, even though most of them cannot be used to develop the project.

1. **Evidence the Document has been placed under configuration management**

* Version: 3.0 (last version)
* Date: 7/5/2023

1. **Impact of the project on individuals and organizations**

* Overall, this final report represents the culmination of our efforts in developing a practical software application based on theoretical concepts. Throughout this project, we have gained valuable insights and practical experience that have significantly enhanced our skills and deepened our understanding of the subject matter.
* For Lac, the project has provided an opportunity to delve into front-end programming. Through the completion of the final report, he has become proficient in utilizing PHP and HTML languages to build web forms. Additionally, the project has facilitated the improvement of his teamwork and interpersonal skills, as he successfully interacted with other team members and collaborated effectively.
* As for Khoa, the project has fostered his ability to conduct research and categorize relevant materials, which were subsequently synthesized into a comprehensive report. Furthermore, he has made significant progress in utilizing SQL Server for database management, thereby enhancing his technical proficiency in this area compared to his previous level of knowledge.
* In conclusion, this final report signifies the successful development of a practical software solution and showcases our individual growth and collective achievements. The knowledge and skills acquired throughout this project will undoubtedly contribute to our future endeavors and professional advancement.

1. **Requirement Specifications**
2. **Stakeholders for the system**

In this system, the stakeholders include:

* Distributors: They are the key suppliers of mobile phone products in the system. They play a crucial role in sourcing and distributing the latest mobile phone models to the agents. Distributors may include manufacturers, wholesalers, or authorized resellers who have established relationships with mobile phone brands.
* Agents: These are the intermediaries between the distributors and the end users. Agents typically operate as retailers or authorized dealers of mobile phones. They establish partnerships or agreements with distributors to acquire mobile phone inventory at wholesale prices. Agents are responsible for managing their inventory, marketing the products, and facilitating the sales process through the website.
* Users: Users are the ultimate consumers of the mobile phone products. They are individuals or groups of people who have the intention to purchase a mobile phone. Users interact with the agents through the website to explore available options, compare prices, and make a purchase decision. Users may have varying preferences, needs, and budgets when it comes to selecting a mobile phone model.

1. **Use case model**
2. **Graphical use case model**

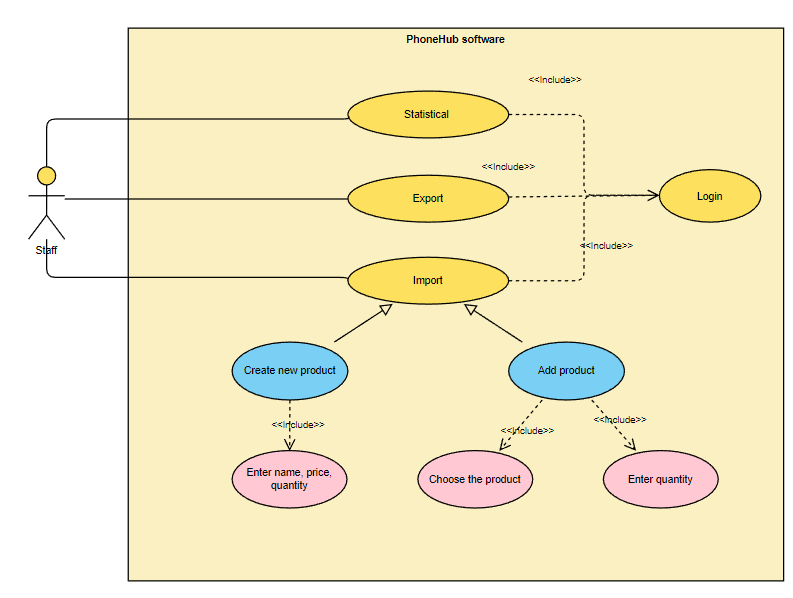
****

Fig 1 Use case of Import goods

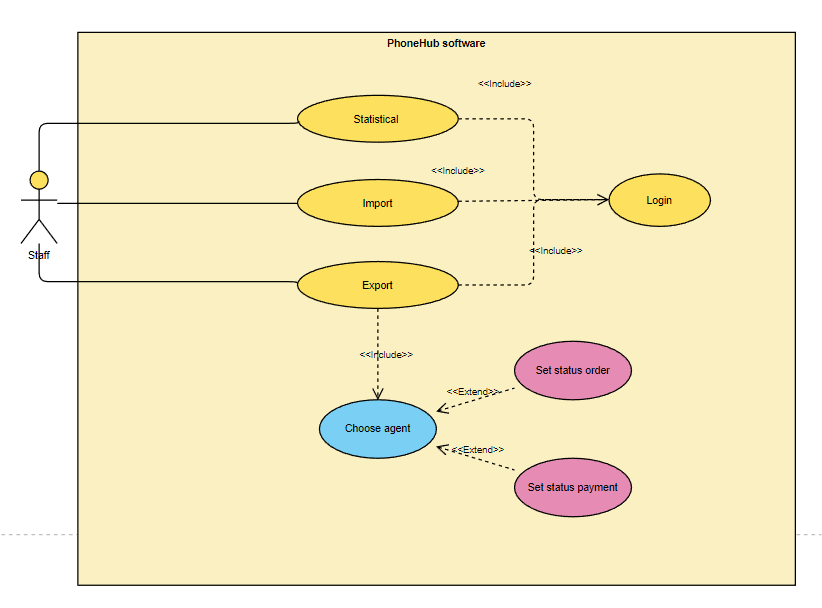
****

Fig 2 Use case of Export goods

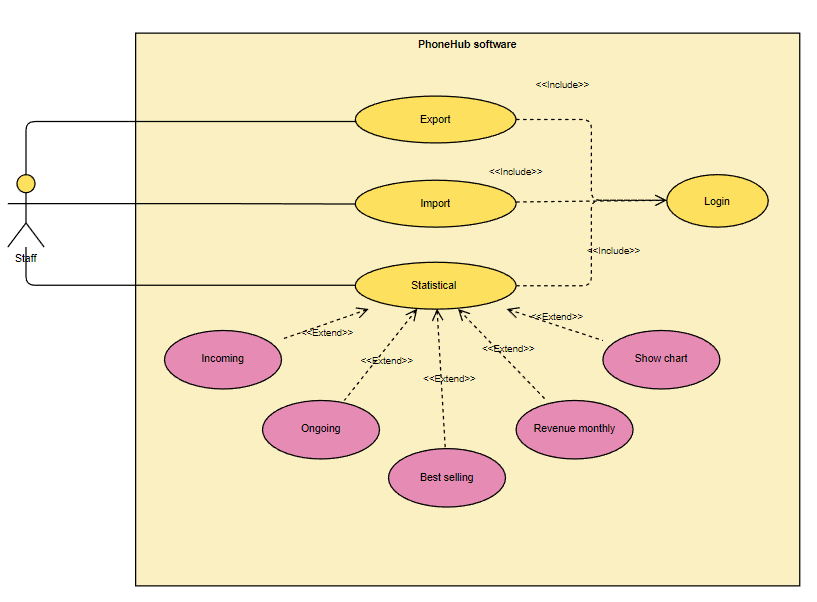
****

Fig 3 Use case of Statistical

1. **Textual Description for each use case**

|  |  |
| --- | --- |
| Use Case Name | Import goods |
| Participating Actors | Staff |
| Entry Condition(s) | Staff Account must be available |
| Normal Flow of Events | 1. Staff logins the form 2. Staff chooses Import tab 3. Staff choose a product and its quantity 4. Staff taps Import button |
| Exit Condition(s) | As a new product, its name must be available  The quantity and price must be available |
| Exceptions (Alternate Flow of Events) | If staff wants to create a new product,staff must follow this flow of events after:   1. Staff logins the form 2. Staff chooses Import tab 3. Staff type the name of new product, its quantity and price 4. Staff taps Create button |
| Special Requirements | The name of the new product must not duplicate an existing product  The number of quantity and price must be a positive whole number |

Table 2 Import goods Des.

|  |  |
| --- | --- |
| Use Case Name | Export goods |
| Participating Actors | Staff |
| Entry Condition(s) | Staff Account must be available |
| Normal Flow of Events | 1. Staff logins the form 2. Staff chooses Export tab 3. Staff chooses a agent 4. Staff chooses a order 5. Staff taps Export button |
| Exit Condition(s) | A new bill must be created |
| Exceptions (Alternate Flow of Events) | None |
| Special Requirements | The number of product in stock must be more than the number of product in the order |

Table 3 Export goods Des.

|  |  |
| --- | --- |
| Use Case Name | Statistical |
| Participating Actors | Staff |
| Entry Condition(s) | Staff Account must be available |
| Normal Flow of Events | 1. Staff logins the form 2. Staff chooses Statistical tab 3. Staff chooses a type of statistical |
| Exit Condition(s) | None |
| Exceptions (Alternate Flow of Events) | None |
| Special Requirements | None |

Table 4 Statistical goods Des.

1. **Functional requirements**

* The system should authenticate staff members' login credentials.
* The system should provide a user-friendly form for staff members to input import details.
* The system should display a list of available products for staff members to choose from.
* The system should allow staff members to enter the quantity of the selected product.
* The system should validate that the entered quantity is a positive whole number.
* The system should verify that the product being imported is not already existing in the system.
* The system should update the inventory with the imported product and its quantity.
* The system should record the import transaction with relevant details, such as the staff member's information and timestamp.
* The system should authenticate staff members' login credentials.
* The system should provide a user-friendly form for staff members to input export details.
* The system should display a list of available agents for staff members to choose from.
* The system should allow staff members to select an order for export.
* The system should verify that the selected order has sufficient stock available for export.
* The system should update the inventory by reducing the quantity of exported products.
* The system should generate a new bill for the exported goods, including relevant information such as the agent, order details, and total price.
* The system should record the export transaction with relevant details, such as the staff member's information and timestamp.
* The system should authenticate staff members' login credentials.
* The system should provide a user-friendly interface for staff members to access statistical analysis.
* The system should offer different types of statistical analysis, such as sales trends, top-selling products, or revenue by month.
* The system should generate statistical reports based on the selected type of analysis.
* The system should present the statistical information in a clear and organized manner, such as charts, graphs, or tables.
* The system should allow staff members to export or save the generated statistical reports for further analysis or presentation purposes.

1. **Non-functional requirements**

* Error Handling and Validation:

The system should handle errors gracefully and provide clear error messages to assist staff in troubleshooting and resolving issues during the import process. It should perform thorough validation of the imported product information, ensuring that the entered data is accurate, consistent, and meets specified criteria.

* Logging and Auditing:

The system should maintain a log of import activities, including the date, time, and user responsible for each import operation, for the purpose of tracking and auditing. This log should help in monitoring and ensuring accountability.

* Regulatory Compliance and Data Security:

The import process should adhere to any relevant regulatory requirements or industry standards governing the importation of goods. Additionally, the system should ensure the confidentiality and protection of sensitive product and pricing information during the import process.

* Accessibility:

The system should be accessible to staff members with disabilities, adhering to accessibility standards and guidelines. This ensures that all staff can effectively use the system regardless of their abilities.

* Efficiency and Timeliness:

The system should efficiently process and handle the export of goods to minimize delays and ensure timely delivery. It should be capable of handling export operations promptly and without significant performance degradation.

* Reporting and Documentation:

The system should allow staff to generate comprehensive reports and documentation related to the exported goods, such as invoices, shipping labels, and customs documentation. These reports and documents aid in the smooth execution and documentation of the export process.

* Scalability:

The system should be scalable to accommodate an increasing number of export operations and handle higher volumes of exported goods without performance degradation. This ensures that the system can handle growing demands efficiently.

* Statistical Analysis and Visualization:

The system should generate statistical analyses and reports within a reasonable timeframe, even when dealing with large datasets or complex calculations. It should provide visually appealing and informative charts, graphs, and visual representations of statistical data to enhance data interpretation and analysis.

* Accuracy and Data Retention:

The system should ensure the accuracy and reliability of the statistical results by employing appropriate statistical algorithms and techniques. It should also retain historical statistical data for future reference and comparison, allowing staff to track trends and patterns over time.

These non-functional requirements encompass aspects such as error handling, validation, logging, compliance, security, accessibility, efficiency, reporting, scalability, statistical analysis, visualization, accuracy, and data retention.

1. **Architecture**
2. **Architectural style(s) used**

* The architectural style used for this system is a three-tier architecture. The three-tier architecture separates the system into three different layers: presentation, application, and data. The presentation layer handles the user interaction, the application layer contains the business logic, and the data layer stores and manages the data.
* The three-tier architecture is a popular software architectural pattern used to separate a system into three layers or tiers: presentation, application, and data. In this system, the presentation layer is responsible for handling user requests, presenting the data to the user, and receiving input from the user. The application layer contains the business logic and processes user requests. The data layer stores and manages the data needed for the application. This separation of concern allows each layer to function independently of each other, making it easier to make changes or updates to one layer without affecting the other layers. Three-tier architecture also enables scalability by allowing each tier to be scaled up or down independently based on its changing needs.

1. **Architectural model**

* Architectural Model:
* The architectural model used for this system is Model-View-Controller (MVC). The MVC architectural pattern is used to separate the system into three different components: model, view, and controller. The model represents the data and business logic, the view represents the user interface, and the controller handles the communication between the model and view.
* The MVC (Model-View-Controller) architectural pattern is another popular software architecture used to separate a system into three components: model, view, and controller. The model represents the data and the business logic, the view is responsible for rendering the user interface, and the controller handles requests and manages the flow between the model and the view. In this system, the model component will contain the database schema, queries, and business logic associated with the data. The view component will define the user interface with the help of HTML, CSS, and JavaScript. The controller component will handle HTTP requests, facilitate communication between the view and the model, and map URLs to corresponding actions.
* The use of three-tier architecture and MVC architectural pattern offers several benefits, such as:
* Separation of concerns, which makes the system more modular and easier to maintain and update.
* Clear separation between data, business logic, and user interface, which improves scalability and extensibility.
* Simplification of debugging and testing, making it easier to identify and solve problems in the system.
* Greater flexibility for developers in modifying and updating the different architectural components.

Overall, these architectural styles and pattern help in delivering a high-quality software system that meets business requirements and users' needs.

1. **Technology, software, and hardware used**

|  |  |
| --- | --- |
| **Technology** | |
| user interface | ASP.NET MVC framework |
| business logic | C# programming language |
| database | SQL Server |
| **Software** | |
| development | Visual Studio 2022 |
| database | SQL Sever SQL Management 2019 |
| **diagram** | PowerDesign |
| Hardware | |
| run the application | 2 latops, 1 computer |

Table 5 Technology, software, and hardware used.

1. **Rationale for your architectural style and model**
2. **Design**
3. **Database design**

* The database design starts with creating a database called "PhoneDistribute". Then several tables are created:
* "Products" table containing columns as "id", "name", "price", and "quantity". "id" column is set as the primary key with auto-incremented values. This table will store all the products with their respective details.
* "Import" table containing columns as "id" and "created\_date". "id" column is set as the primary key with auto-incremented values. This table will store the details of product imports.
* "Import\_detail" table containing columns as "idImportDetail", "idImport", "idProduct", and "quantity". "idImportDetail" column is set as the primary key with auto-incremented values. This table will store the details of each imported products.
* "Agent" table containing columns as "id", "agent\_name", "phone", and "address". "id" column is set as the primary key with auto-incremented values. This table will store the details of all agents dealing with the product.
* "Order\_product" table containing columns as "id", "agentId", "orderDate", "statusOrder", "statusPay" and "methodPay". "id" column is set as the primary key with auto-incremented values. This table will store the orders received from agents.
* "Order\_detail" table containing columns as "id", "orderId", "productId", and "quantity". "id" column is set as the primary key with auto-incremented values. This table will store the details of products that are being ordered.
* "Sold" table containing columns as "id", "productId", and "quantity". "id" column is set as the primary key with auto-incremented values. This table will store the details of sold products.

1. **Static model – class diagrams**

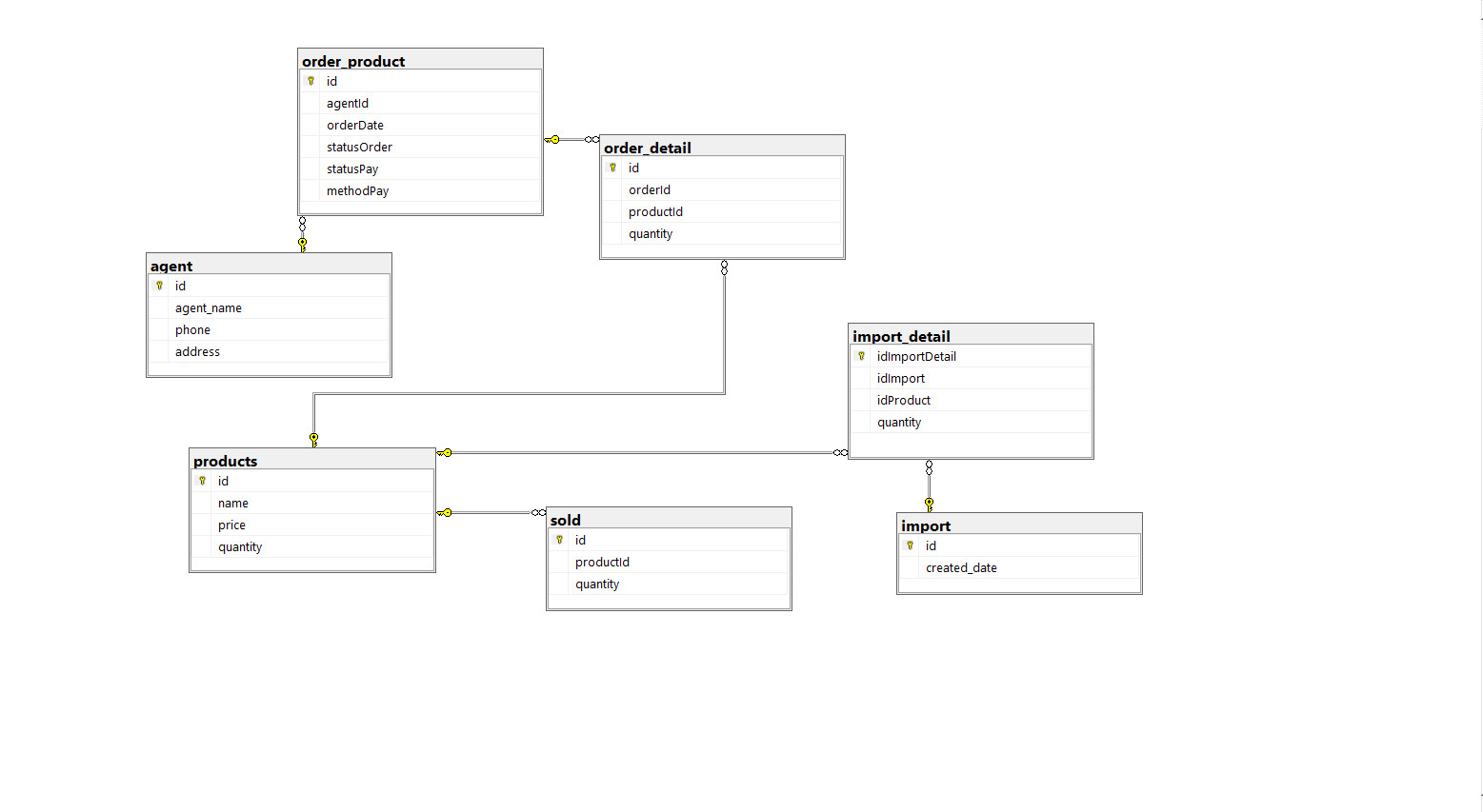


Fig 4 Class Diagram

1. **Dynamic model – sequence diagrams**

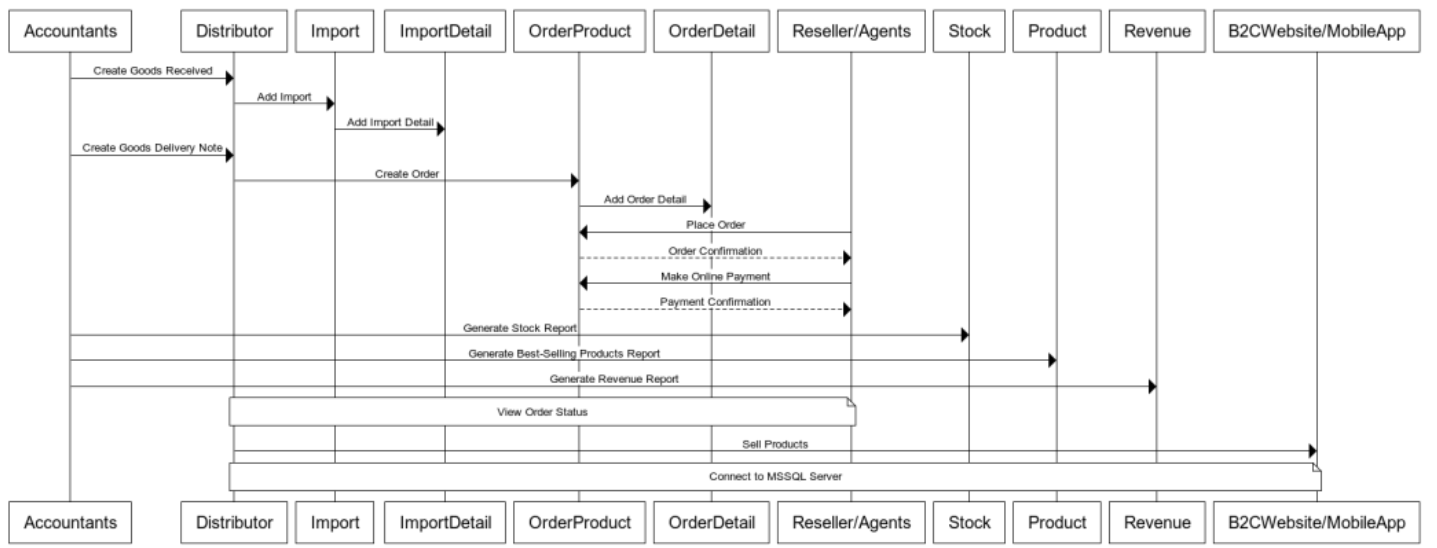


Fig 5 Sequence Diagram

1. **Rationale for your detailed design model**

* The three-tier architecture was chosen because it separates the concerns of the system into separate layers making the system more modular and easier to maintain.
* The MVC architectural pattern was chosen because it separates the user interface from the business logic making it easier to maintain and modify the system.
* These architectural styles and model provide a structured and organized approach to building software systems and are widely used and accepted in industry.

1. **Traceability from requirements to detailed design model**

* Requirements:
* Accountants shall be able to create Goods Received when the distributor imports goods (a warehouse receipt will include many items) (Win Form / Web Form)
* Reseller / Agents shall be able to place an order of items and choose a payment method (Cash, bank transfer, Momo...). Reseller / Agents would also make an online payment and see the status of their orders (Webform)
* Accountants shall be able to create Goods Delivery Note to deliver goods to agents (print delivery slips), update the status of orders as being transferred and update the payment status of agents. (Win Form / Web Form)
* Accountants shall be able to view incoming/outgoing stock report, best-selling products and revenue report monthly. (Form/Report)
* Design Model:
* Create entity-relationship diagram (ERD) that represents all the tables in the SQL Server database: products, import, import\_detail, agent, order\_product, order\_detail, and sold.
* Use Win Form or Web Form to develop the UI to implement the requirements.
* In Win Form/Web Form, use C# to write the logic for the functionalities.
* Develop a reporting module to generate monthly reports for incoming/outgoing stocks, best-selling products, and revenue using SSRS.
* Traceability:
* The Goods Received functionality can be implemented in the UI using Win Form or Web Form. The logic for creating a warehouse receipt can be written in C# and the data can be stored in the SQL Server database.
* The Reseller/Agent functionality can be implemented in the UI using a Web Form. The payment methods can be added as dropdown options and the payment status and order status can be updated in the database according to the Reseller/Agent's actions.
* The Goods Delivery Note functionality can be implemented in the UI using Win Form or Web Form. The Accountants can print delivery slips and update order and payment status in C# and the data can be stored in the SQL Server database.
* The reporting functionality can be implemented in the reporting module using SSRS. The SQL Server database can be queried to generate the monthly reports.

1. **Test Plan**
2. **Requirements/specifications-based system level test cases**
3. **Test Case: Goods Received Creation**

* Description: Verify that accountants can successfully create a goods received entry when the distributor imports goods.
* Test Steps:
* Login to the software as an accountant.
* Navigate to the Goods Received section.
* Enter the necessary details such as import date, warehouse receipt information, and item details.
* Save the goods received entry.
* Verify that the entry is successfully created in the system.

1. **Test Case: Order Placement and Payment**

* Description: Ensure that resellers/agents can place an order, choose a payment method, and make an online payment.
* Test Steps:
* Login to the software as a reseller/agent.
* Browse the available products and select the desired items.
* Add the selected items to the cart.
* Proceed to the checkout process.
* Choose a payment method (cash, bank transfer, Momo, etc.).
* Complete the payment transaction online.
* Verify that the order is successfully placed and the payment status is updated.

1. **Test Case: Goods Delivery and Order Status Update**

* Description: Validate that accountants can create a goods delivery note, update the order status as being transferred, and update the payment status of agents.
* Test Steps:
* Login to the software as an accountant.
* Access the Goods Delivery section.
* Create a new goods delivery note by selecting the appropriate order and entering the necessary details.
* Print the delivery slip.
* Update the status of the corresponding order as "transferred."
* Update the payment status of the agent associated with the order.
* Verify that the delivery note is created, the order status is updated, and the agent's payment status is modified.

1. **Test Case: Stock and Sales Reports**

* Description: Verify that accountants can generate incoming/outgoing stock reports, view best-selling products, and access revenue reports.
* Test Steps:
* Login to the software as an accountant.
* Navigate to the Reports section.
* Generate an incoming stock report for a specific period.
* Generate an outgoing stock report for a specific period.
* Analyze the best-selling products based on sales data.
* Access the revenue report for a specific month.
* Verify that the reports are generated accurately and provide the expected information.

1. **Traceability of test cases to use cases**

**Test Case: Goods Received Creation**

- Use Case(s) Covered: Create Goods Received

- Description: This test case verifies that accountants can successfully create a goods received entry when the distributor imports goods.

**Test Case: Order Placement and Payment**

- Use Case(s) Covered: Place Order, Choose Payment Method, Make Online Payment

- Description: This test case ensures that resellers/agents can place an order, choose a payment method, and make an online payment.

**Test Case: Goods Delivery and Order Status Update**

- Use Case(s) Covered: Create Goods Delivery Note, Update Order Status, Update Payment Status

- Description: This test case validates that accountants can create a goods delivery note, update the order status as being transferred, and update the payment status of agents.

**Test Case: Stock and Sales Reports**

- Use Case(s) Covered: Generate Stock Reports, View Best-Selling Products, Access Revenue Reports

- Description: This test case verifies that accountants can generate incoming/outgoing stock reports, view best-selling products, and access revenue reports.

1. **Techniques used for test generation**

* We used unit tests for the test cases mentioned:
* Identify the Units: Determine the specific units or components of the software that need to be tested. In this case, the units may include individual functions, modules, or classes responsible for the functionalities mentioned in the test cases.
* Set Up Test Environment: Prepare the necessary environment for running the unit tests. This may involve creating test data, initializing dependencies, and configuring the test environment to mimic the actual runtime environment.
* Write Test Cases: Create unit test cases based on the identified test cases. Each unit test case should focus on testing a specific aspect or behavior of the unit being tested. Write test assertions to verify the expected results or behavior.
* Implement Unit Tests: Implement the unit tests using a unit testing framework compatible with the programming language and framework. We used NUnit for .NET.
* Execute Unit Tests: Run the unit tests and observe the results. The unit testing framework will execute the test cases and report any failures or errors encountered during the test run.
* Analyze Results: Review the test results to identify any failed or erroneous tests. Investigate the cause of failures and identify potential defects or issues in the units being tested.
* Debug and Refine: If any test cases fail, debug the code and fix any identified issues. Update the unit tests as needed to reflect the changes made to the code.
* Repeat: Repeat the process for other units or components that need to be tested, creating additional unit tests as necessary.

1. **Assessment of the goodness of your testsuite (Which metrics were used for such assessment?)**
2. **Demo**
3. **Database**

* Run Database.sql and LogMange.sql (More details at Readme.txt)

1. **Source code**

* Extract source.zip and run ImportForm.sln (More details at Readme.txt)

1. **Testing**

# REFERENCE