**Vietnam General Confederation of Labor**

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**SOFTWARE ENGINEERING FINAL PROJECT**

**MANAGEMENT SOFTWARE SELLING MOBILE PHONE PRODUCT**

*Instructor*: **Mr. PHAM THAI KY TRUNG**

*Student*: **Le Nguyen Nhat Hao - 521H0047**

**Nguyen Duc Anh - 521H0005**

*Class* **: 21H50201**

*Year* **: K25**

**HO CHI MINH CITY, 2022**

Vietnam General Confederation of Labor

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**SOFTWARE ENGINEERING FINAL PROJECT**

**MANAGEMENT SOFTWARE SELLING MOBILE PHONE PRODUCT**

*Instructor*: **Mr. PHAM THAI KY TRUNG**

*Student*: **Le Nguyen Nhat Hao - 521H0047**

**Nguyen Duc Anh - 521H0005**

*Class* **: 21H50201**

*Year* **: K25**

**HO CHI MINH CITY, 2022**

**ACKNOWLEDGEMENT**

The first sincere thanks I want to give to Mr. Pham Thai Ky Trung, who enthusiastically taught and worked tirelessly to give me enough tools and skills to complete this report. He played an important role in improving my mathematical logic and knowledge. The second thanks I would like to give to the teachers of the Department of Information Technology of Ton Duc Thang University for giving me the opportunity to do this report.

I am very open to receiving feedback from teachers so that I can improve my report writing skills. Finally, I wish you good health and success in your noble career.

**THIS PROJECT WAS COMPLETED AT**

**TON DUC THANG UNIVERSITY**

We hereby undertake that this is our own project product and under the guidance of Mr. Pham Thai Ky Trung. Research contents and results in this topic are truthful and have not been published in any form. any before. The data in the tables serving the analysis, comments and evaluation collected by the author from different sources are clearly stated in the references.

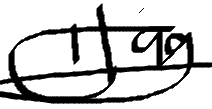
In addition, the project also uses a number of comments, assessments as well as data of other authors, other organizations and organizations with citations and origin notes.

**If I detect any fraud I take full responsibility for the content of our project.** Ton Duc Thang University is not related to any copyright or copyright infringement caused by me during the implementation process (if any).

*Ho Chi Minh city, 7th May, 2023*

*Author*

*(Sign and write full name)*

*Nguyen Duc Anh*

*Le Nguyen Nhat Hao*

**CONFIRMATION AND ASSESSMENT SECTION**

**Instructor confirmation section**

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

*Ho Chi Minh May 2nd 2022*

*(Sign and write full name)*

**Evaluation section for grading instructor**

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

*Ho Chi Minh May 2nd 2022*

*(Sign and write full name)*

**SUMMARY**

[LIST OF FIGURES 6](#_Toc2946)

LIST OF TABLES....................................................................................................................................................7

CHAPTER 1-INTRODUCTION .........................................................................................................................9

[1.1 Purpose and Scope 9](#_Toc14793)

[1.2 Product Overview 10](#_Toc7232)

[1.3 Structure of the Document 10](#_Toc1518)

[1. Introduction 10](#_Toc9247)

[1.1. Purpose and Scope 10](#_Toc30496)

[1.2. Product Overview (including capabilities, scenariosfor using the product, etc.) 10](#_Toc1742)

[1.3. Structure of the Document 10](#_Toc18789)

[1.4. Terms, Acronyms, and Abbreviations 10](#_Toc18858)

CHAPTER 2- PROJECT MANAGEMENT PLAN ......................................................................................11

[2. Project Management Plan 11](#_Toc23025)

[2.1. Project Organization 12](#_Toc31306)

[2.2. Lifecycle Model Used 12](#_Toc22632)

[2.3. Risk Analysis 12](#_Toc10369)

[2.4. Hardware and Software Resource Requirements 13](#_Toc16650)

[2.5. Deliverables and Schedule 13](#_Toc17508)

[2.6. Monitoring, Reporting, and Controlling Mechanisms 14](#_Toc23842)

[2.7. Professional Standards 14](#_Toc19828)

[2.8. Evidence all the artifacts have been placed under configuration management 15](#_Toc9228)

[2.9. Impact of the project on individuals and organizations 15](#_Toc25617)

[CHAPTER 3 - REQUIREMENT SPECIFICATIONS 1](#_Toc25068)5

[3.1 Stakeholders for the system 17](#_Toc20283)

[3.3 Functional requirements 24](#_Toc12357)

[3.4 Non – functional requirements 25](#_Toc32279)

[CHAPTER 4 - ARCHITECTURE 25](#_Toc2687)

[4.1 Architectural style used 25](#_Toc25631)

[4.2 Architectural model 27](#_Toc31080)

[4.3 Technology, software, and hardware used 29](#_Toc14083)

[4.4 Rationale architectural style and model 30](#_Toc6653)

[CHAPTER 5 – DESIGN 31](#_Toc8481)

5.1. Database design...............................................................................................................................31

5.2. Static model – class diagrams3..................................................................................................31

5.3. Dynamic model – sequence diagrams....................................................................................32

[5.4 Rationale for your detailed designn model 38](#_Toc12590)

5.5 Mookup Design..................................................................................................................................40

5.6 Logo Design.........................................................................................................................................49

[CHAPTER 6 – TEST PLAN 51](#_Toc12167)

[6.1 Requirements/specifications-based system level test cases 51](#_Toc3118)

[6.2 Traceability of test cases to use cases 53](#_Toc1424)

6.3. Techniques used for test generation......................................................................................53

6.4. Assessment of the goodness of your testsuite (Which metrics were used for such assessment?)........................................................................................................................................................54

CHAPTER 7 -DEMO…………………………………………………………………………………………………...55

**LIST OF FIGURES**

**1.Receipt**

**2.Delivery Bill**

**3.Statistion level**

**4.Order**

**5.Payment**

**LIST OF TABLES**

**1.Use case for Receipt**

**2.Use case for Delivery Bill**

**3.Use case for Statistion level**

**4.Use case for Order**

**5.Use case for Payment**

**CHAPTER 1 – INTRODUCTION**

**1.1 Purpose and Scope**

**Purpose:** Building import and export management software for many types of items, orders, payment methods and revenue statistics for the dietary supplement company for agents.

**Scope:**

**Mobile Phone Company**

**Distributes**

**Distributor**

**Sells to**

**Reseller/Agent**

**POS Web B2C Web System System /Mobile**

**1.2 Product Overview**

* The Mobile Phone Company distributes its products to the Distributor.
* The Distributor uses a POS system to manage inventory, create Goods Received, and Goods Delivery Notes.
* Reseller/Agents can place orders using the Web System and choose a payment method.
* The B2C website or Mobile App is used for selling products directly to customers.
* All systems are connected to the same MSSQL Server.
* Accountants can view incoming/outgoing stock reports, best-selling products, and revenue reports using a reporting system.

**1.3 Structure of the Document**

1. Introduction

1.1. Purpose and Scope

1.2. Product Overview (including capabilities, scenariosfor using the product, etc.)

1.3. Structure of the Document

1.4. Terms, Acronyms, and Abbreviations

2. Project Management Plan

2.1. Project Organization

2.2. Lifecycle Model Used

2.3. Risk Analysis

2.4. Hardware and Software Resource Requirements

2.5. Deliverables and Schedule

2.6. Monitoring, Reporting, and Controlling Mechanisms

2.7. Professional Standards

2.8. Evidence all the artifacts have been placed under configuration management

2.9. Impact of the project on individuals and organizations

3. Requirement Specifications

3.1. Stakeholders for the system

3.2. Use case model

3.2.1. Graphical use case model

3.2.2. Textual Description for each use case

3.3. Functional requirements

3.4. Non-functional requirements

4. Architecture

4.1. Architectural style(s) used

4.2. Architectural model

4.3. Technology, software, and hardware used

4.4. Rationale for your architectural style and model

5. Design

5.1. Database design

5.2. Static model – class diagram

5.3. Dynamic model – sequence diagrams

5.4. Rationale for your detailed design model

5.5. Traceability from requirements to detailed design model

6. Test Plan

6.1. Requirements/specifications-based system level test cases

6.2. Traceability of test cases to use cases

6.3. Techniques used for test generation

6.4. Assessment of the goodness of your testsuite

7. Demo

7.1. Database

7.2. Source code

7.3. Testing

**CHAPTER 2 - PROJECT MANAGEMENT PLAN**

**2.1 Project Organization Structure**

Project Manager

|

Project Team

/ | \

Design Team Development Team QA Team

/ \ / \

Designer Engineer Tester Test Lead

**2.2 Lifecycle Model Used**

Deployment

Requirements

Gathering

Testing

Analysis

Design

Coding

**2.3 Risk Analysis**

- Leave the input blank

- Duplicate primary key

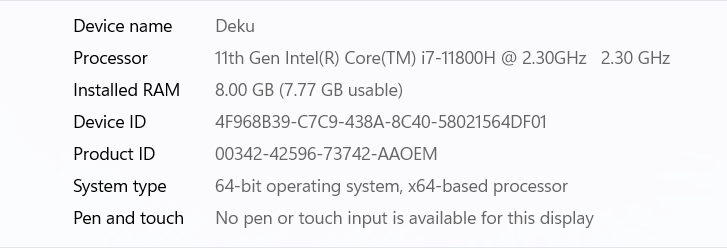
- Delete foreign key

- Edit foreign key

**2.4 Hardware and Software Resource Requirements**

-Hardware:

* Server machine with a minimum of 4 cores and 8GB RAM
* Desktop/Laptop machines with a minimum of 2 cores and 4GB RAM for each user
* Network infrastructure, including switches, routers, and cables
* Printers for printing delivery slips and reports



-Software:

Windows Server Operating System

Microsoft SQL Server Management Studio 18

Visual Studio for software development: Visual Studio Code, Visual studio 2019.

**2.5 Deliverbles and Schedule**

-Deliverables:

* Project plan and timeline
* Software requirements specification document
* Software design document
* Initial working prototype or mockup
* Test plan and test cases
* User documentation

-Schedule:

**Week 1:** Gather requirements, finalize project plan and timeline, start designing software

**Week 2:** Complete software design document, create initial working prototype or mockup

**Week 3:** Create test plan and test cases, start writing user documentation

**Week 4:** Conduct testing, refine and finalize software, complete user documentation, deliver final product

**2.6 Monitoring, Reporting, and Controlling Mechanisms**

**Monitoring and Controlling**

Monitor and control project work, perform integrated change control, validate scope, control scope, schedule, costs, quality, communications, risks, procurements, stakeholder engagement, and report when the project is completed. These activities involve gaining approval of deliverables, controlling scope, scheduling, costs, quality, communications, risks, procurements, stakeholder engagement, and reporting.

**2.7 Professional Standards**

-Professional standards are a set of guidelines, principles, and ethical codes that professionals are expected to uphold in their respective fields to ensure they provide high-quality services and maintain the trust and confidence of their clients or customers.

-Example:

* Experience in using Visual Studio.
* Experience in using Microsoft SQL Server Management Studio 19.

**2.8 Evidence all the artifacts have been placed under configuration management**

-To provide evidence that all the artifacts have been placed under configuration management, the following can be done:

* Version Control System, Change Management Plan, Configuration Management Plan, Regular Audits, and Documentation should be used to track changes made to project artifacts.
* Change Management Plan should outline steps involved in making changes, Configuration Management Plan should define process and procedures for managing artifacts, Regular Audits should be conducted to identify any issues or gaps, and Documentation should be stored alongside the artifact.

-By implementing these measures, it can be ensured that all project artifacts are placed under configuration management and are properly documented, tracked, and maintained.

**2.9 Impact of the project on individuals and organizations**

**- Accounting:** Increase productivity, personnel and reduce time to manage goods when entering and leaving warehouse. Easy revenue statistics.

**- Customers:** Convenient to buy goods from afar. Easy to choose payment method.

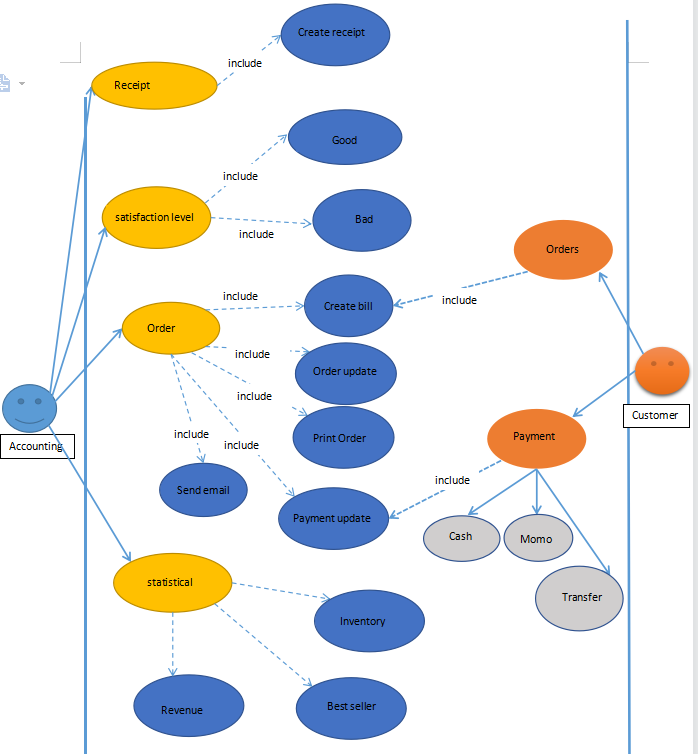
**- Organization:** Reduce human waste, modernize management processes.

**CHAPTER 3 - REQUIREMENT SPECIFICATIONS**

**3.1 Stakeholders for the system**

* The stakeholders for the system can be grouped into two main categories: internal and external.
* Internal stakeholders include distributor, accountants, resellers/agents, customers, suppliers, IT staff, managers, and regulatory bodies.
* External stakeholders include resellers/agents, customers, suppliers, IT staff, managers, and regulators.

***3.2.1 Graphical use case model***



***3.2.2 Use case specification***

**Use case for Receipt**

**-Use Case:** Goods Received for Accountant

**-Description:** This use case describes the process of creating a Goods Received record when the distributor receives goods.

**-Preconditions:**

* The distributor has imported goods and they are ready to be received into the warehouse.
* The accountant has access to the software and is logged in.

**-Basic Flow:**

* The accountant selects the "Goods Received" option from the menu.
* The system displays a form for creating a new Goods Received record.
* The accountant enters the details of the shipment, including the date, supplier, invoice number, and any additional notes.
* The accountant adds the items received, including the product name, quantity, and price.
* The system calculates the total value of the shipment and displays it on the form.
* The accountant saves the Goods Received record.

**-Alternative Flows:**

If the shipment includes damaged or defective items, the accountant can add a note to the Goods Received record indicating this. They can then follow the process for returning or replacing the items.

**Postconditions:**

* The Goods Received record is saved in the system and can be viewed by the accountant.
* The stock levels for the received items are updated in the system.

**Use case for Delivery bill**

**Use Case:** Delivery Bill

Actors: Accountant

**Description:** This use case describes the process of creating a delivery bill to deliver goods to agents.

**Preconditions:**

* The distributor has received an order from an agent.
* The accountant has access to the software and is logged in.

**Basic Flow:**

* The accountant selects the "Delivery Bill" option from the menu.
* The system displays a form for creating a new delivery bill.
* The accountant enters the details of the shipment, including the date, agent name, and any additional notes.
* The accountant adds the items to be delivered, including the product name, quantity, and price.
* The system calculates the total value of the shipment and displays it on the form.
* The accountant prints the delivery bill.
* The accountant updates the status of the order as "transferred" in the system.
* The accountant updates the payment status of the agent as "paid" in the system.

**Alternative Flows:**

If there are any discrepancies or issues with the shipment, the accountant can add a note to the delivery bill indicating this. They can then follow the process for resolving the issue before delivering the goods.

**Postconditions:**

The delivery bill is printed and can be attached to the shipment.

The status of the order and payment status of the agent are updated in the system.

**Use case for Payment**

**Use Case:** Payment

Actors: Reseller/Agent

**Description**:

This use case describes the process of making a payment for a mobile phone product order.

**Preconditions**:

The reseller/agent has placed an order for mobile phone products using the system.

The reseller/agent has access to the system and is logged in.

**Basic Flow:**

The reseller/agent selects the "Payment" option from the menu.

The reseller/agent selects their preferred payment method, such as cash, bank transfer, or Momo, ZaloPay

If the reseller/agent selects online payment, they enter their payment details and confirm the payment.

The system processes the payment and updates the status of the order as "paid".

The reseller/agent can view the status of their order in the system.

**Updating:**

If there are any issues with the payment, such as a declined transaction or technical difficulties, the reseller/agent can try again or select a different payment method.

**Postconditions:**

The payment is processed and the status of the order is updated in the system.

The reseller/agent can view the updated status of their order in the system.

**Use case for Order**

**Use Case:** Order

**Actors**: Reseller/Agent

**Description:**

This use case describes the process of placing an order for mobile phone products.

**Preconditions**:

- The reseller/agent has access to the system and is logged in.

**Basic Flow**:

1. The reseller/agent selects the "Order" option from the menu.

2. The system displays a list of available mobile phone products.

3. The reseller/agent selects the desired products and enters the quantity for each.

4. The reseller/agent selects their preferred payment method, such as cash, bank transfer, or Momo.

5. If the reseller/agent selects online payment, they enter their payment details and confirm the payment.

6. The reseller/agent submits the order.

7. The system generates an order confirmation and updates the inventory levels.

**Alternative Flows**:

- If the desired product is out of stock, the system displays a message and the reseller/agent can either choose a different product or wait for the product to become available.

- If the reseller/agent encounters any issues while placing the order, such as technical difficulties, they can contact customer support for assistance.

**Postconditions**:

- The order is placed and the system generates an order confirmation.

- The inventory levels are updated in the system.

- The reseller/agent can view the status of their order in the system.

**Use case for Receipt**

Title: Create Goods Received receipt

Actors: Accountant

Description: This use case describes the steps for an Accountant to create a Goods Received receipt when the distributor imports goods into their warehouse.

Preconditions:

- The Accountant is logged in to the system

- The distributor has imported goods into their warehouse

Postconditions:

- A Goods Received receipt is created and stored in the system

- The inventory is updated to reflect the goods that were received

Flow of Events:

1. The Accountant opens the Goods Received module in the system.

2. The system displays a form for the Accountant to enter the details of the receipt, including the date, the supplier's name, the invoice number, and any additional notes.

3. The Accountant enters the details of the receipt into the form and clicks "Submit."

4. The system validates the information and generates a unique receipt number.

5. The system updates the inventory to reflect the items that were received and the quantities.

6. The system saves the Goods Received receipt in the system with the unique receipt number.

7. The Accountant can print or email the receipt to the relevant parties if necessary.

Alternative Flows:

- If the system detects an error in the information entered by the Accountant, it displays an error message and prompts the Accountant to correct the information before proceeding.

**Use case for statistical**

**Use Case:** Statistical

**Actors:** Accountants

**Description:**

This use case describes the process of generating statistical reports for the distributor's mobile phone products.

**Preconditions:**

- The accountant has access to the system and is logged in.

**Basic Flow:**

1. The accountant selects the "Statistical" option from the menu.

2. The system displays a list of statistical reports that can be generated, such as incoming/outgoing stock report, best-selling products, and revenue report monthly.

3. The accountant selects the desired report.

4. The system generates the statistical report and displays it to the accountant.

5. The accountant can export the report in a desired format, such as PDF or Excel.

**Alternative Flows:**

- If the accountant encounters any issues while generating the statistical report, such as technical difficulties or missing data, they can contact the IT support team for assistance.

**Postconditions:**

- The statistical report is generated and displayed to the accountant.

- The accountant can export the report in a desired format for further analysis and distribution.

## 3.3 Functional requirements

**1. Goods Received:**

- The system should allow accountants to create Goods Received when the distributor imports goods, which includes multiple items.

- The system should support both Win Form and Web Form for creating Goods Received.

- The system should allow accountants to enter the details of each item received, such as the item name, quantity, and cost.

**2. Order:**

- The system should allow resellers/agents to place orders for items and choose a payment method, such as cash, bank transfer, or Momo.

- The system should allow resellers/agents to make online payments and view the status of their orders.

**3. Delivery bill:**

- The system should allow accountants to create Goods Delivery Note to deliver goods to agents, which includes a print delivery slip.

- The system should support both Win Form and Web Form for creating Delivery bills.

- The system should allow accountants to update the status of orders as being transferred and update the payment status of agents.

**4. Reviews:**

- The system should allow accountants to view incoming/outgoing stock reports, best-selling products, and revenue reports monthly.

- The system should provide forms and reports to help accountants generate statistical reports.

- The system should allow accountants to export such as PDF or Excel.

Overall, System should be easy to use, secure, reliable, and scalable.

## **3.4** Non – functional requirements

- The system should have fast response times for all user interactions, such as loading pages, submitting forms, and generating reports.

- The system should have a user-friendly interface that is easy to navigate and understand.

- The system should provide feedback and notifications to users about their actions and the status of their requests.

Overall, the system should be reliable, efficient, secure, and user-friendly to meet the needs of its stakeholders and ensure its long-term success.

# CHAPTER 4 - ARCHITECTURE

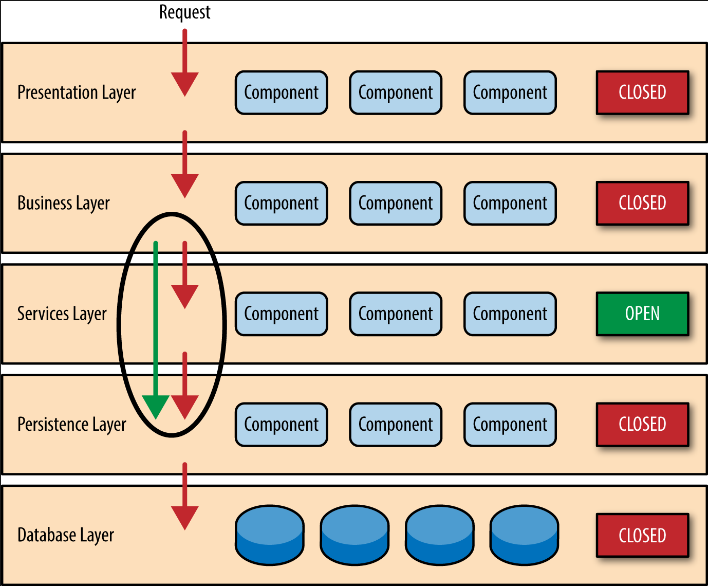
## 4.1 Architectural style used

1. A layered architecture is a software architecture pattern that separates the system into distinct layers, each with a specific responsibility and abstraction level. The layers are typically arranged in a hierarchical order, with each layer building upon the layer below it.

2. In a layered architecture, the business logic, data access, and presentation layers are separated into distinct modules. The business logic layer is responsible for implementing the application's core functionality and business rules, while the data access layer is responsible for managing the system's data storage and retrieval. The presentation layer is responsible for rendering the application's user interface and managing user interactions.

3. The benefits of a layered architecture include improved scalability, maintainability, and testability. Each layer can be independently developed, tested, and modified without affecting other layers. It also promotes separation of concerns and reduces the complexity of the overall system design.

4. In this case, the system's presentation layer could be implemented as a web application, and the business logic and data access layers could be implemented using a combination of ASP.NET and SQL Server technologies. By separating these concerns into distinct layers, the system would be easier to maintain and extend over time.



## 4.2 Architectural model

**1. Presentation Layer:**

- Provides a web interface for authorized resellers/agents to place orders, make payments, and view the status of their orders.

- Built using ASP.NET and web technologies.

**2. Business Logic Layer:**

- Implements the core business rules and processes for the system, including order processing, inventory management, and reporting.

- Manages the interactions between the presentation layer and data access layer.

- Built using C# and .NET technologies.

**3. Data Access Layer:**

- Provides access to the system's data storage and retrieval.

- Manages the database interactions with the MSSQL Server.

- Built using Entity Framework and ADO.NET technologies.

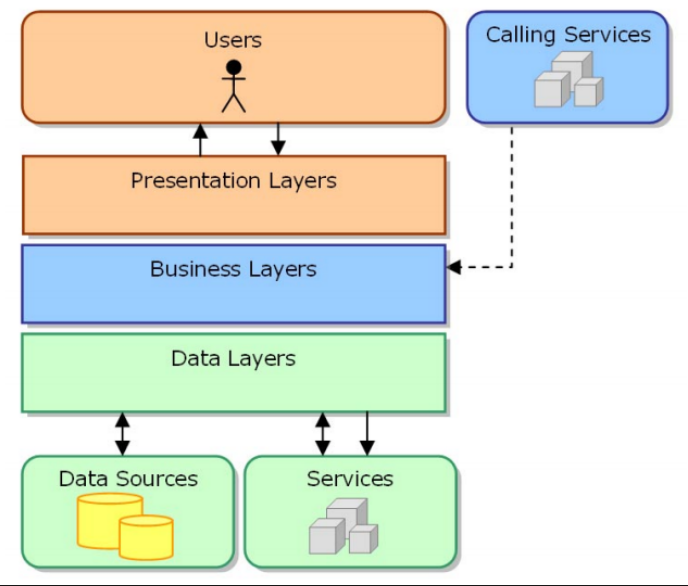
**4. Database:**

- Stores all system data including product information, inventory, orders, and payments.

- Built using Microsoft SQL Server.

-The architecture model follows a standard n-tier design, with each layer having its own distinct responsibilities and interfaces. The presentation layer interacts with the business logic layer through a web interface, while the business logic layer interacts with the data access layer to retrieve and update data in the database. The data access layer manages the communication with the database, and the database stores all system data.

-This architecture model promotes separation of concerns, which makes it easier to manage and modify each layer independently. It also improves the scalability, maintainability, and testability of the system.



## 4.3 Technology, software, and hardware used

**1. Technology:**

- ASP.NET for web development.

- C# for backend development.

- Microsoft SQL Server for database management.

- Entity Framework for data access.

**2. Software:**

- Visual Studio 2019.

- Microsoft SQL Server Management Studio.

- Github.

- Azure for hosting and deployment.

- PayPal, Momo, or other payment gateways for online payments.

**3. Hardware:**

- A server or cloud-based hosting service to run the system.

- Client devices (computers, tablets, smartphones) to access the system through the web interface.

-The software tools and technologies chosen are widely used, reliable, and have a large community of developers. Using Azure for hosting and deployment can provide scalability, availability, and security for the system. Additionally, using a payment gateway like PayPal or Momo can provide secure and convenient online payment options for authorized resellers/agents.

-The hardware requirements depend on the expected number of users and the size of the database. It is recommended to use a server or cloud-based hosting service that meets the required specifications and provides sufficient resources to ensure the system's smooth operation.

## 4.4 Rationale architectural style and model

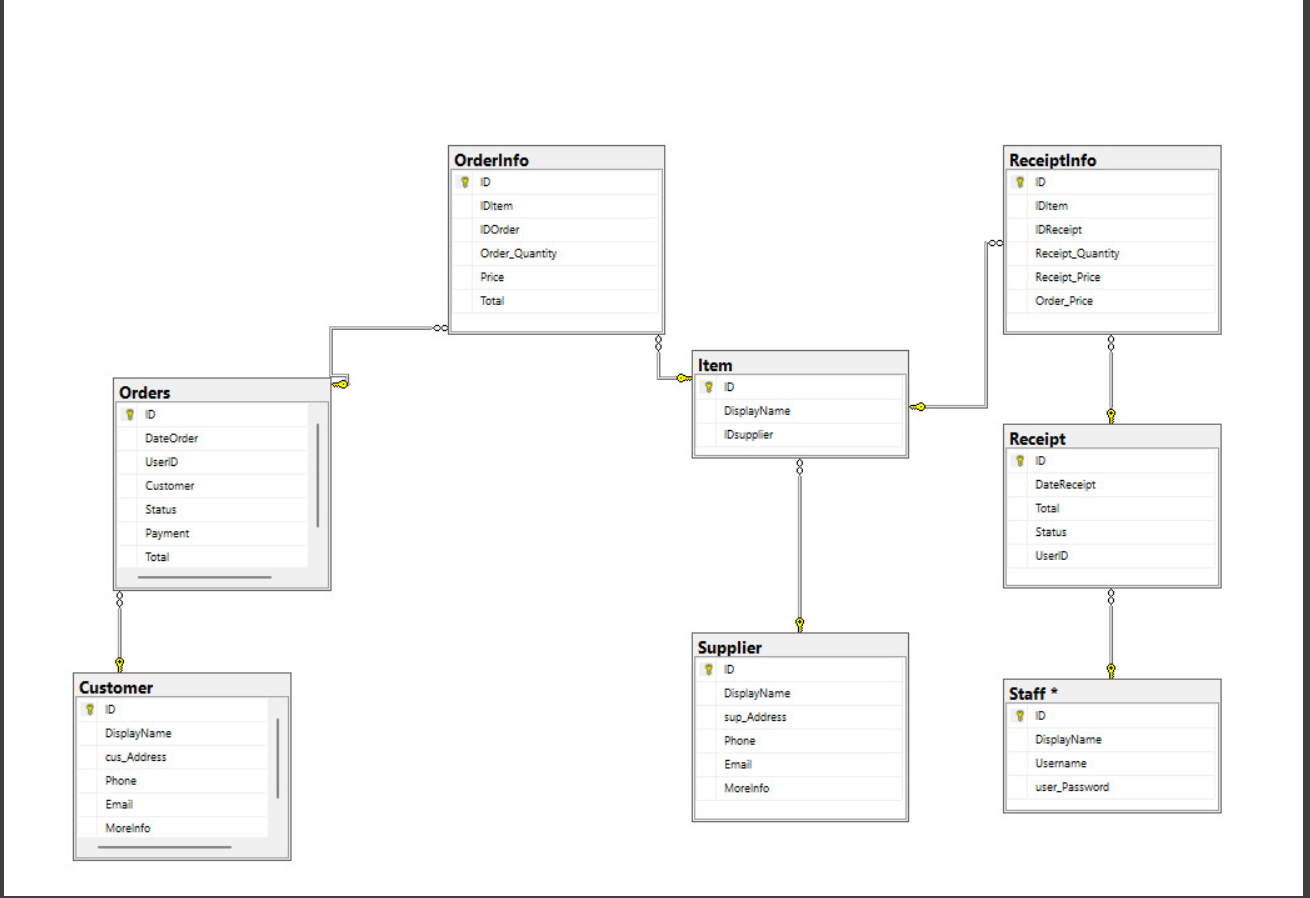
1. The architectural style and model chosen for the system are based on the requirements and non-functional requirements of the system.

1. The system follows a three-tier architecture model, which separates the presentation layer (frontend), business logic layer (backend), and data storage layer (database). This architecture provides modularity, scalability, and flexibility to the system. Changes to one layer do not affect the other layers, allowing developers to work on different parts of the system without interfering with each other.
2. The presentation layer uses ASP.NET and HTML/CSS/JavaScript to create a responsive and user-friendly web interface for authorized resellers/agents to place orders, make payments, and view order and payment status. Bootstrap is used to provide a consistent and modern design for the web interface.
3. The business logic layer is implemented using C# and Entity Framework, which provides a robust and efficient way to interact with the database. This layer handles the business rules and processes, such as order processing, payment handling, and inventory management.
4. The data storage layer uses Microsoft SQL Server to store and manage the data. The database schema is designed to accommodate the requirements and future expansion of the system, such as adding new product categories, inventory management, and reporting.

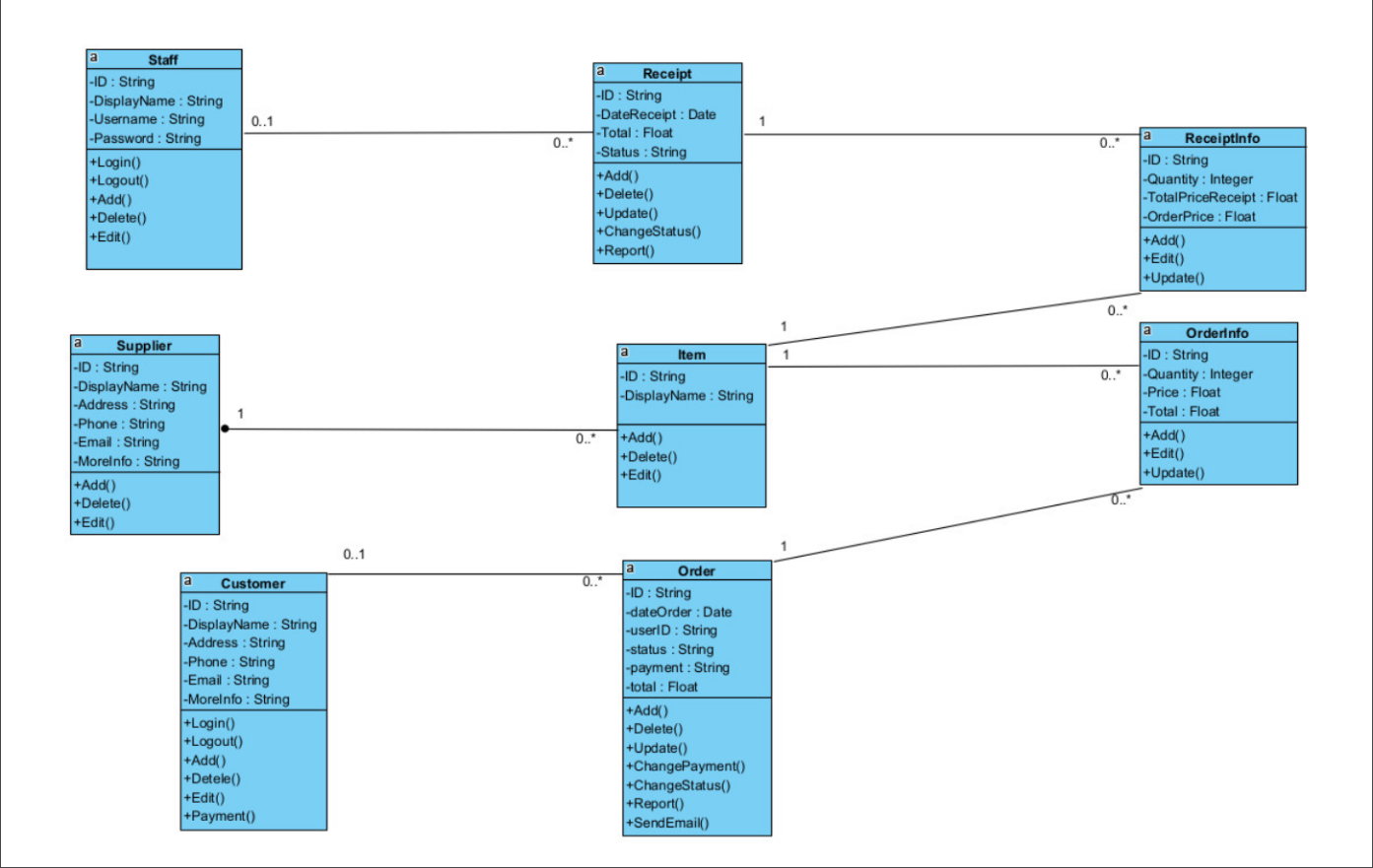
Overall, the chosen architectural style and model are appropriate for the system's requirements, providing modularity, scalability, and flexibility.

# CHAPTER 5 – DESIGN

* 1. **Database design**

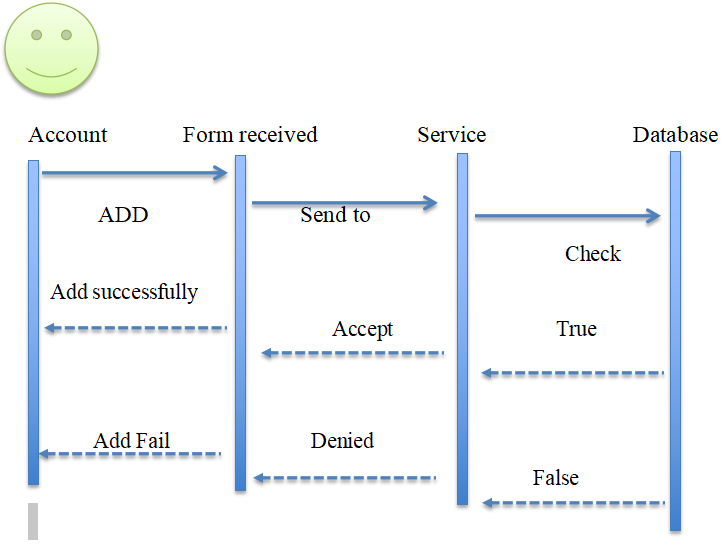


* 1. **Class Diagrams**

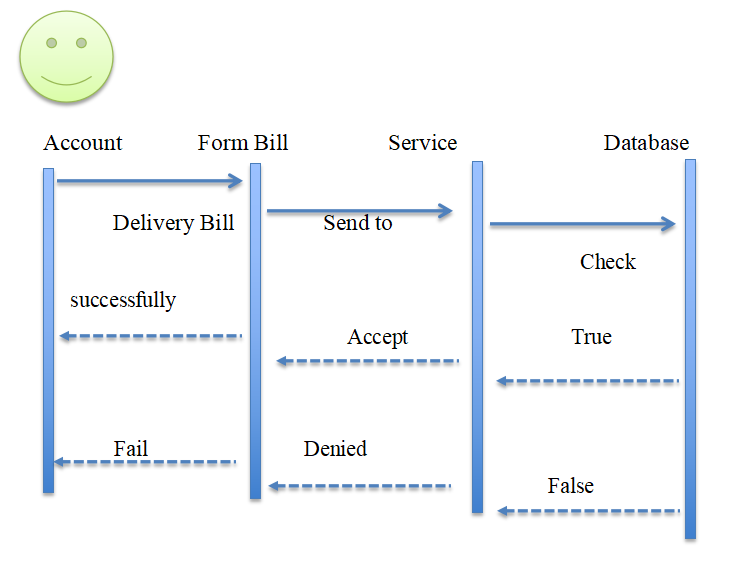


**5.3 Sequence diagrams**

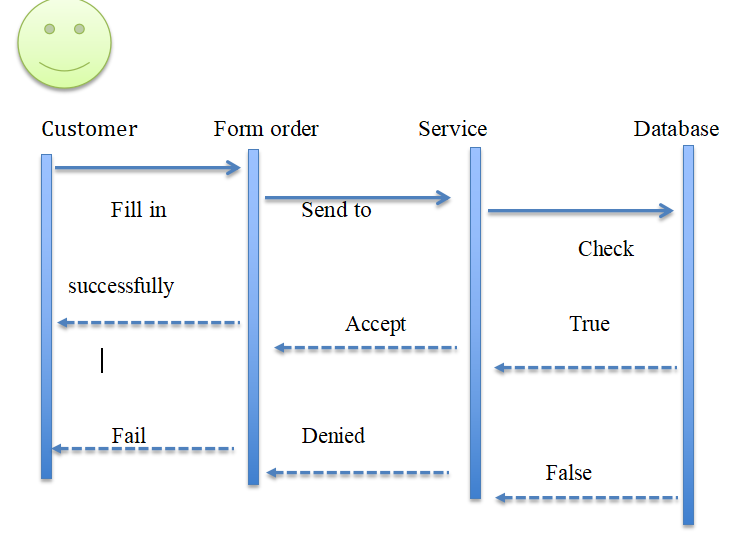
**Good recieved**



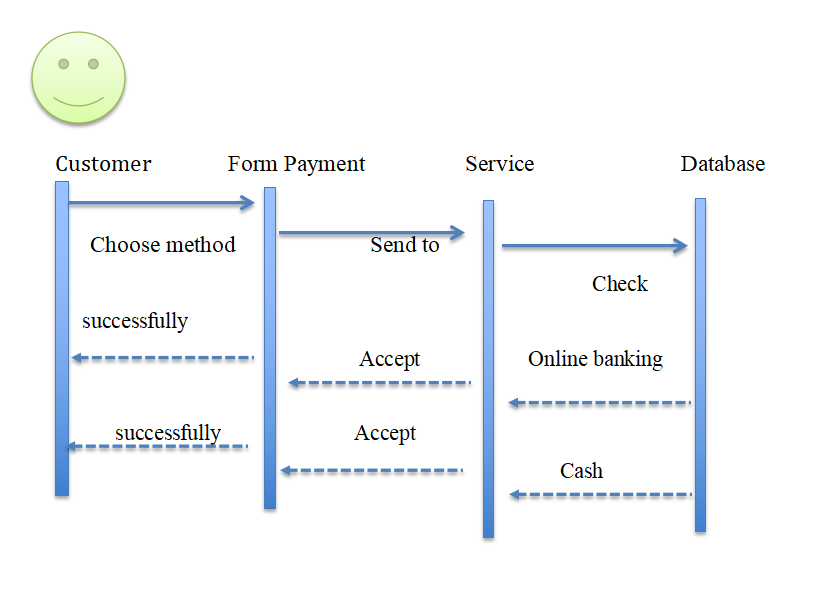
**Delivery Bill**



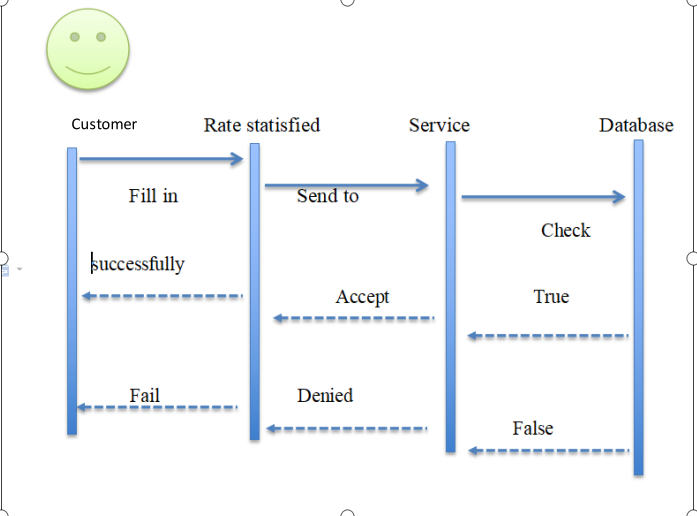
**Order**



**Payment**



**Statistics**



## 5.4 Rationale for your detailed designn model

CREATE TABLE [dbo].[Customer](

[ID] [nvarchar](128) NOT NULL,

[DisplayName] [nvarchar](max) NULL,

[cus\_Address] [nvarchar](max) NULL,

[Phone] [nvarchar](20) NULL,

[Email] [nvarchar](200) NULL,

[MoreInfo] [nvarchar](max) NULL,

PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Item] Script Date: 5/9/2023 11:48:01 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Item](

[ID] [nvarchar](128) NOT NULL,

[DisplayName] [nvarchar](max) NULL,

[IDsupplier] [nvarchar](128) NOT NULL,

PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[OrderInfo] Script Date: 5/9/2023 11:48:01 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[OrderInfo](

[ID] [nvarchar](128) NOT NULL,

[IDItem] [nvarchar](128) NOT NULL,

[IDOrder] [nvarchar](128) NOT NULL,

[Order\_Quantity] [int] NULL,

[Price] [float] NULL,

[Total] [float] NULL,

PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Orders] Script Date: 5/9/2023 11:48:01 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Orders](

[ID] [nvarchar](128) NOT NULL,

[DateOrder] [date] NULL,

[UserID] [nvarchar](128) NULL,

[Customer] [nvarchar](128) NULL,

[Status] [nvarchar](128) NULL,

[Payment] [nvarchar](128) NULL,

[Total] [float] NULL,

PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Receipt] Script Date: 5/9/2023 11:48:01 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Receipt](

[ID] [nvarchar](128) NOT NULL,

[DateReceipt] [date] NULL,

[Total] [float] NULL,

[Status] [nvarchar](128) NULL,

[UserID] [nvarchar](128) NULL,

CONSTRAINT [PK\_\_tmp\_ms\_x\_\_3214EC27AB970F6E] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[ReceiptInfo] Script Date: 5/9/2023 11:48:01 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ReceiptInfo](

[ID] [nvarchar](128) NOT NULL,

[IDItem] [nvarchar](128) NOT NULL,

[IDReceipt] [nvarchar](128) NOT NULL,

[Receipt\_Quantity] [int] NULL,

[Receipt\_Price] [float] NULL,

[Order\_Price] [float] NULL,

PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Staff] Script Date: 5/9/2023 11:48:01 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Staff](

[ID] [nvarchar](128) NOT NULL,

[DisplayName] [nvarchar](max) NULL,

[Username] [nvarchar](max) NULL,

[user\_Password] [nvarchar](20) NULL,

CONSTRAINT [PK\_Users] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Supplier] Script Date: 5/9/2023 11:48:01 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Supplier](

[ID] [nvarchar](128) NOT NULL,

[DisplayName] [nvarchar](max) NULL,

[sup\_Address] [nvarchar](max) NULL,

[Phone] [nvarchar](20) NULL,

[Email] [nvarchar](200) NULL,

[MoreInfo] [nvarchar](max) NULL,

PRIMARY KEY CLUSTERED

(

[ID] ASC

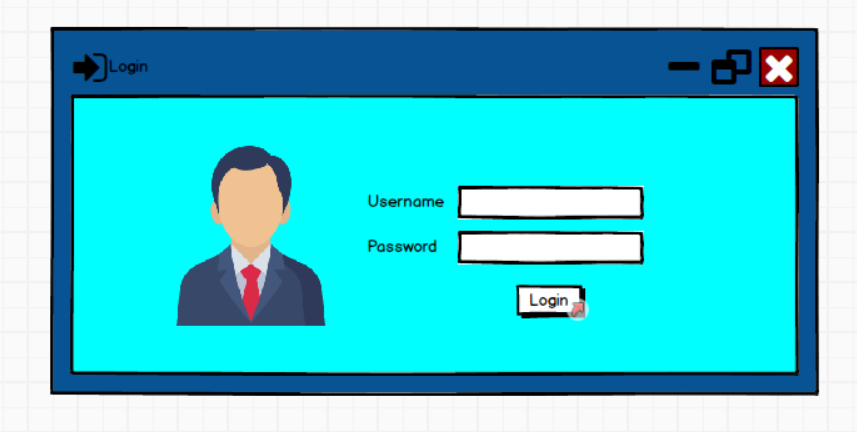
)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

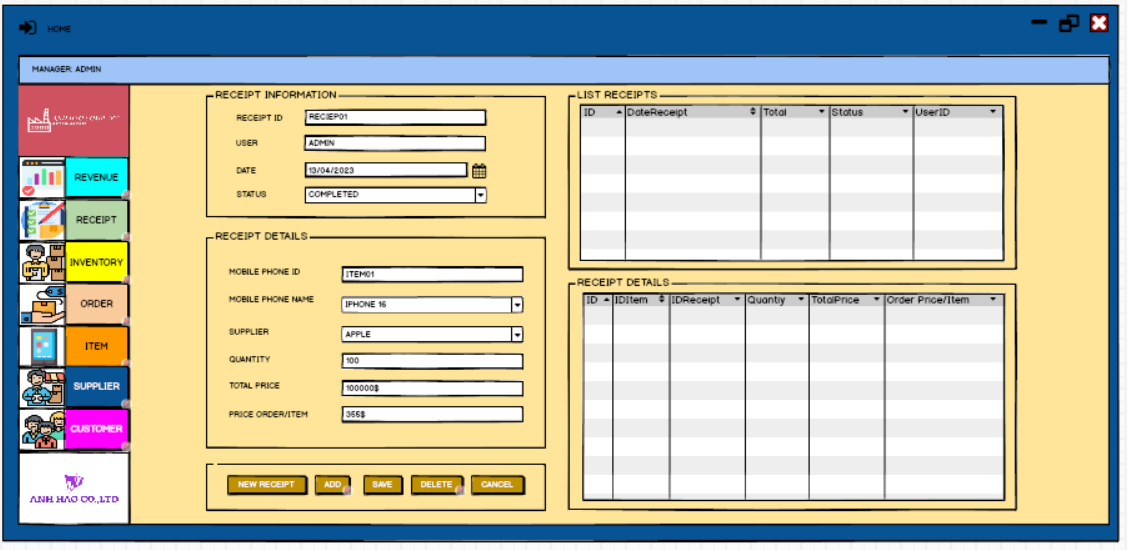
## **-**

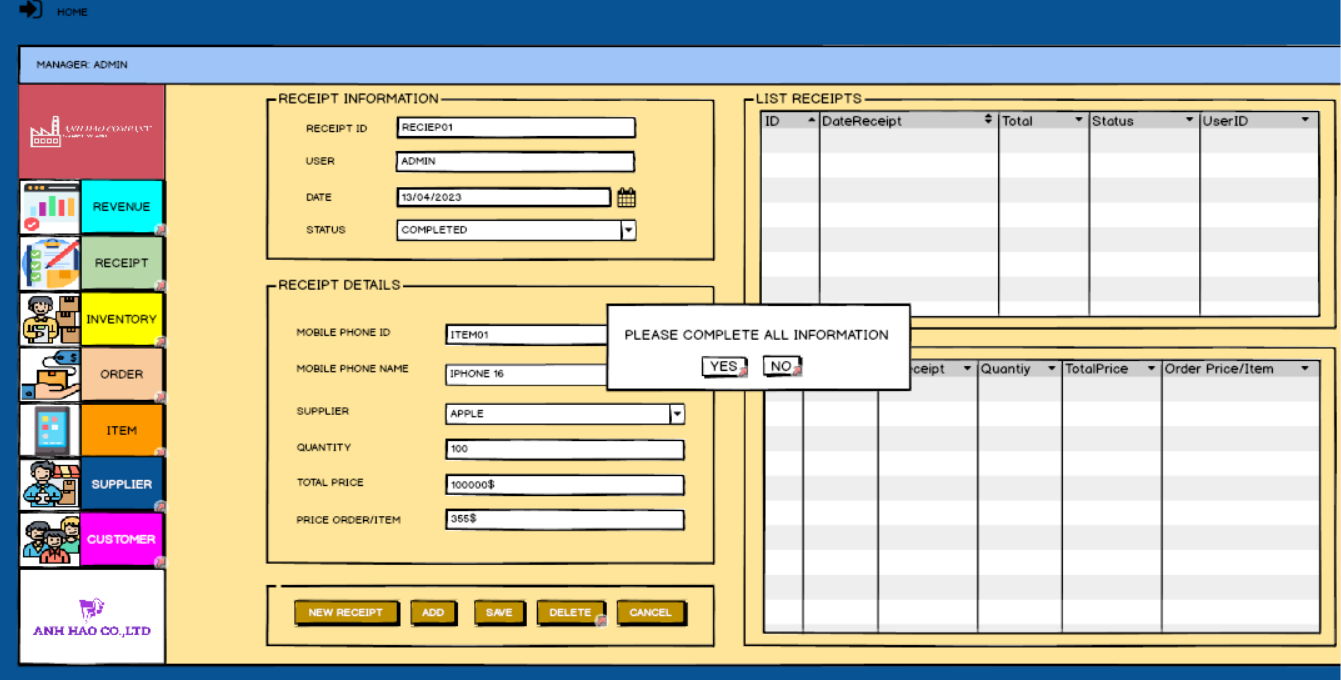
**5.5 Mookup Design**

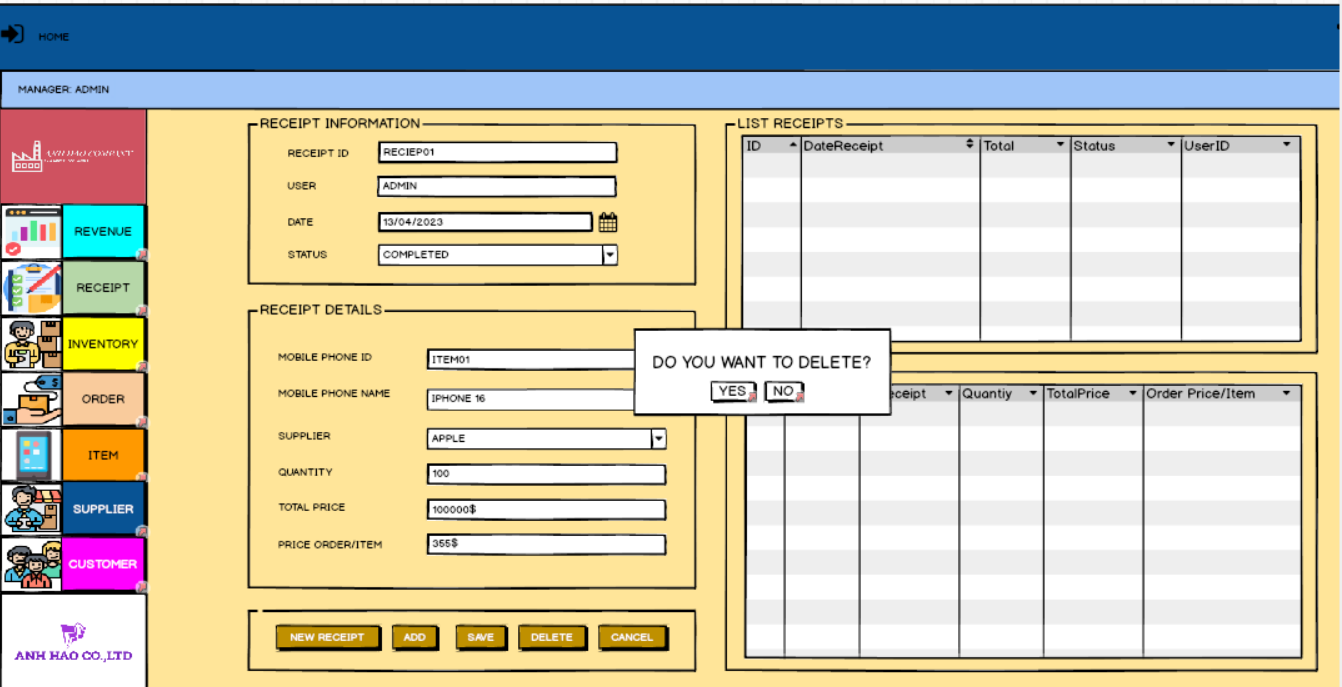
**Login:**



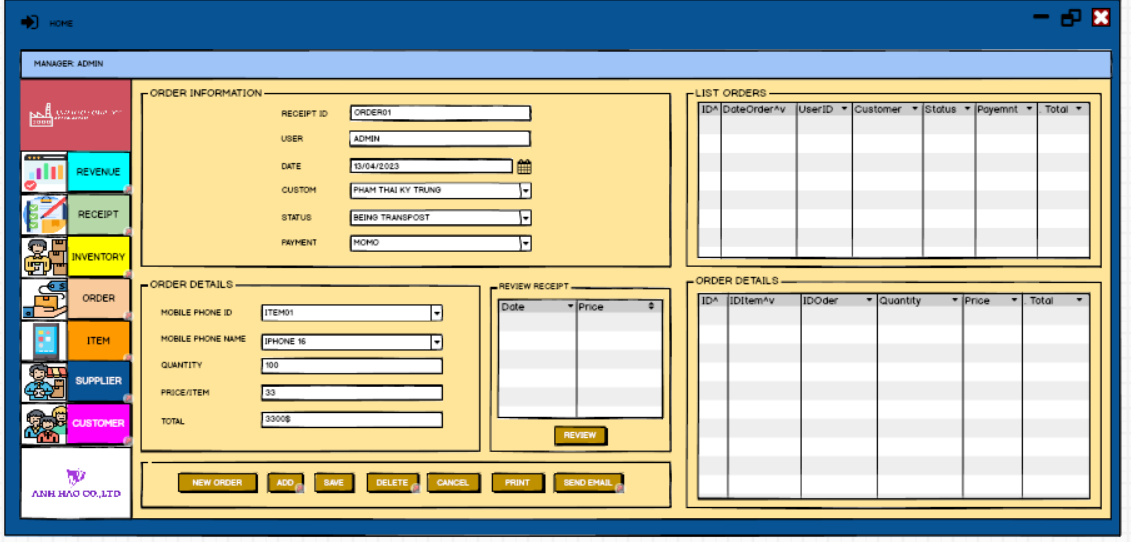
**Receipt:**

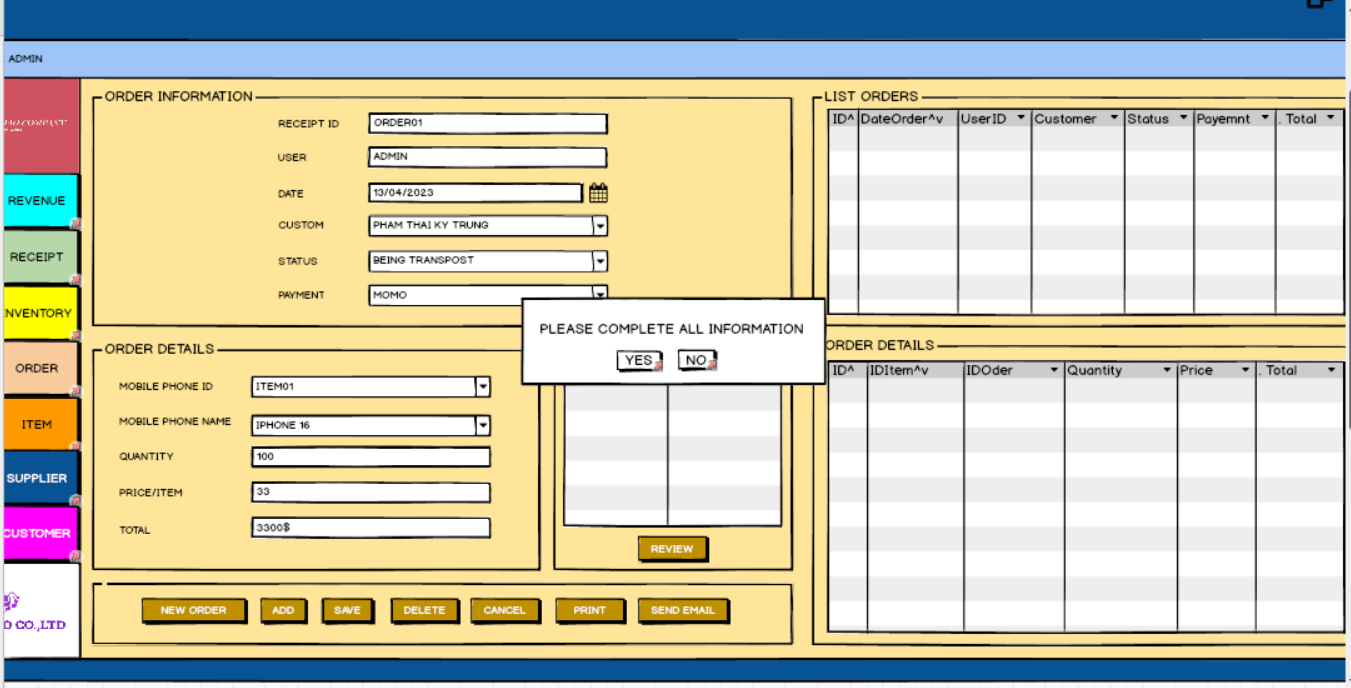


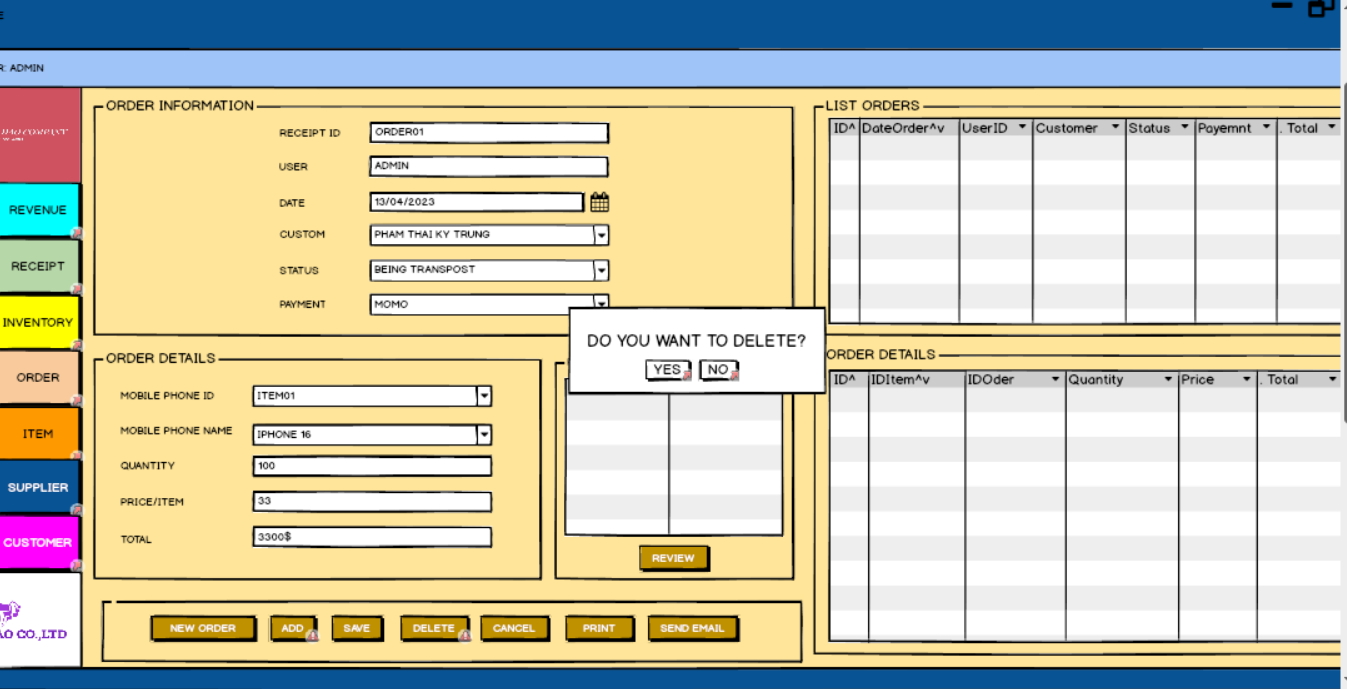


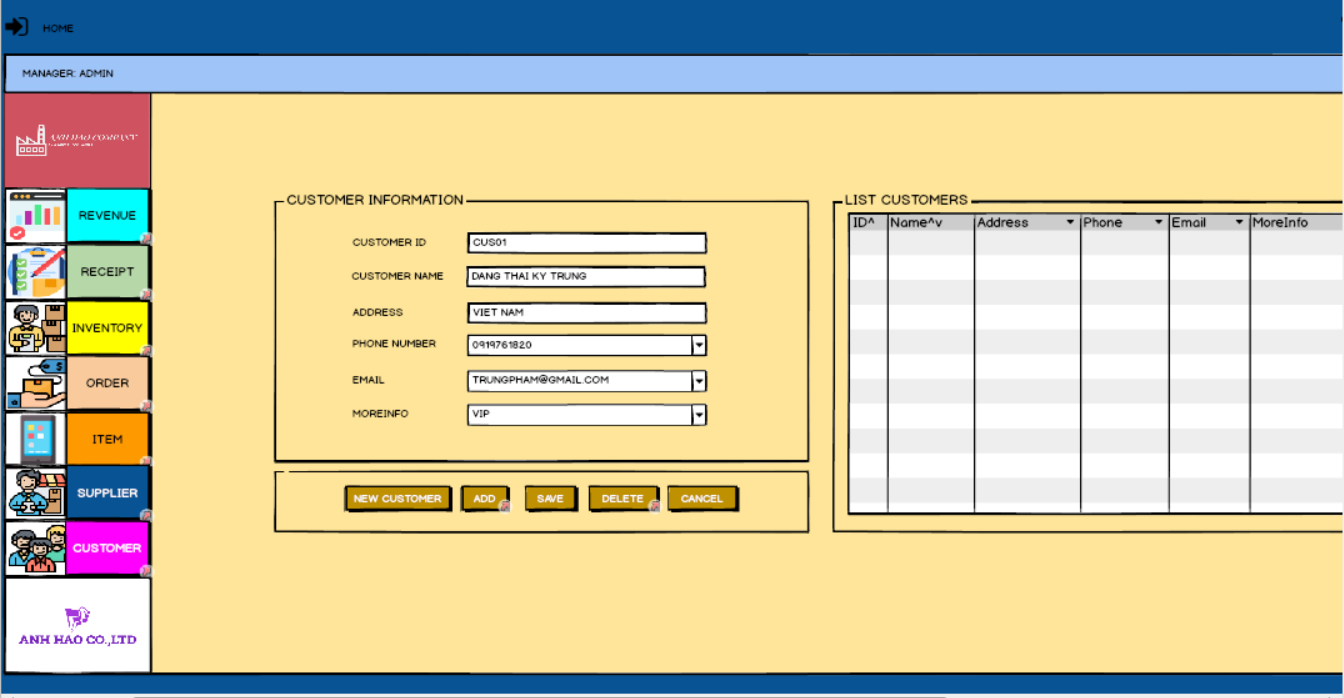


**Order:**

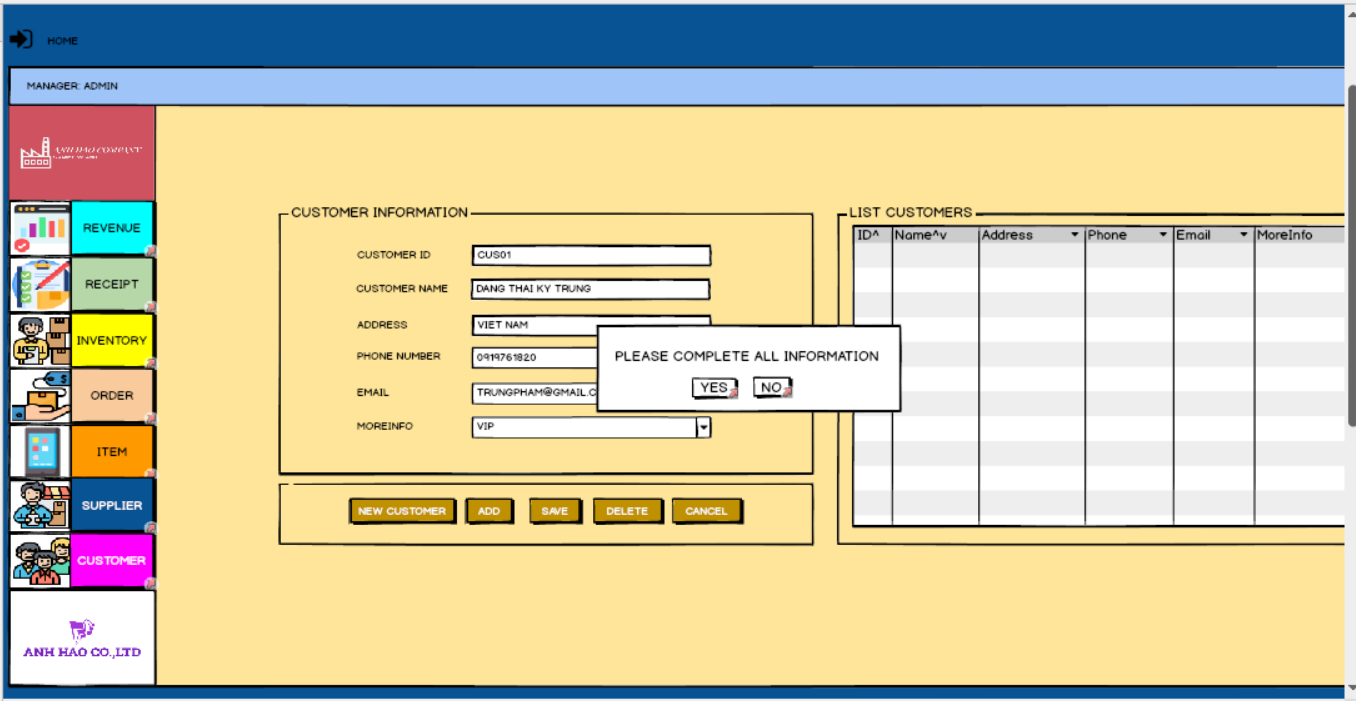


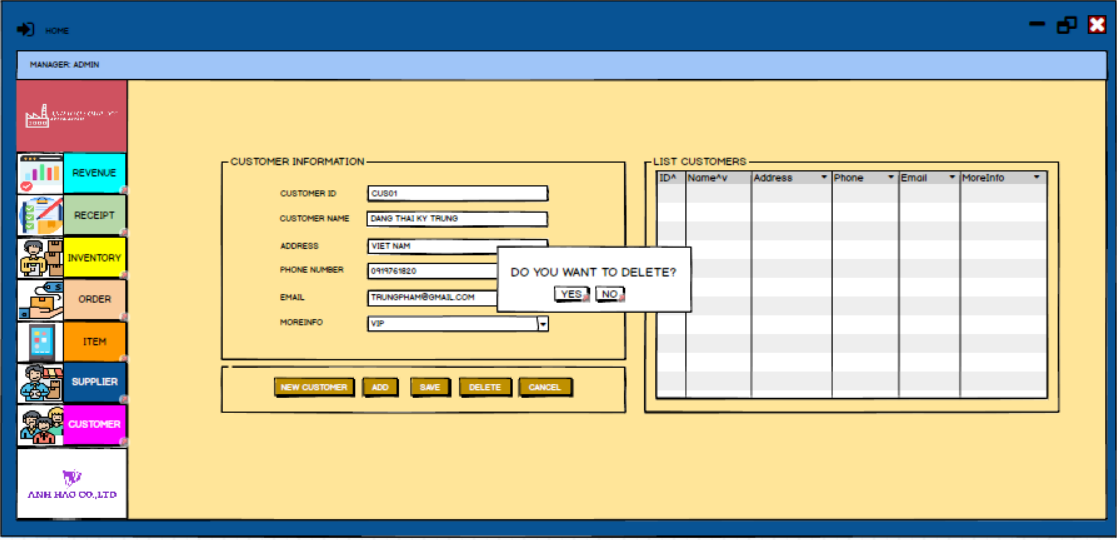




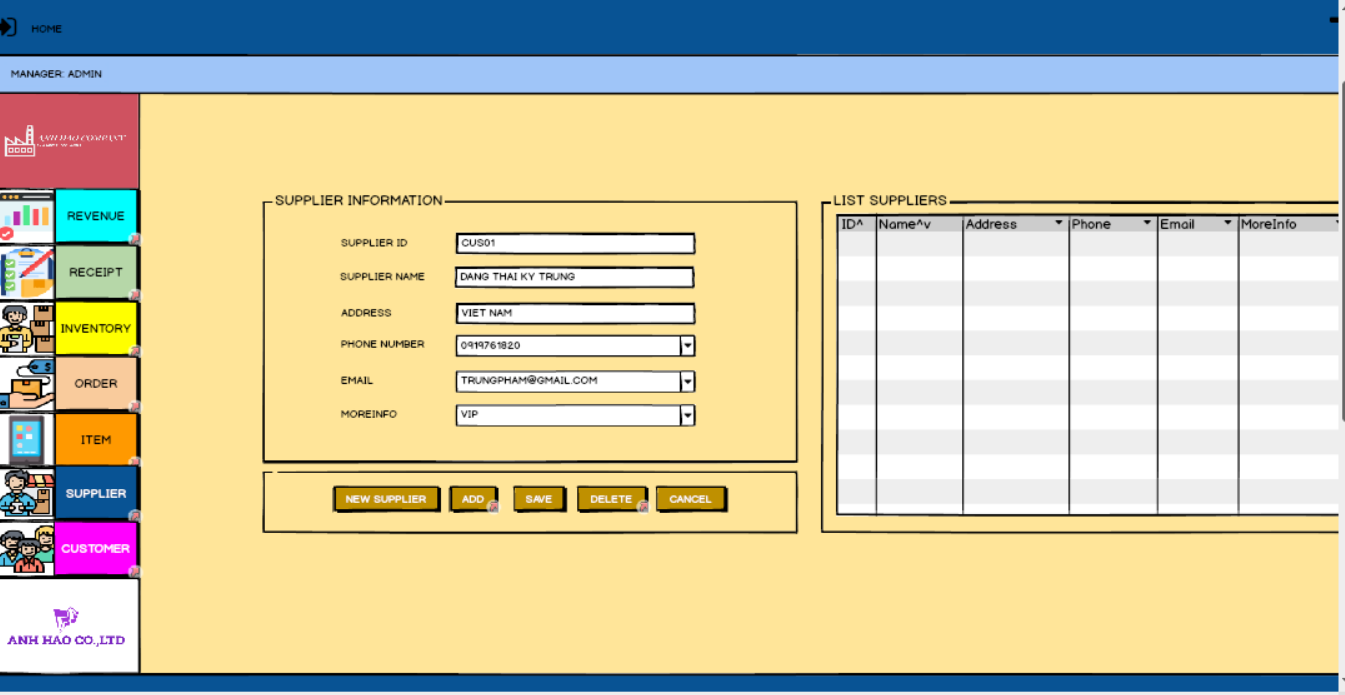


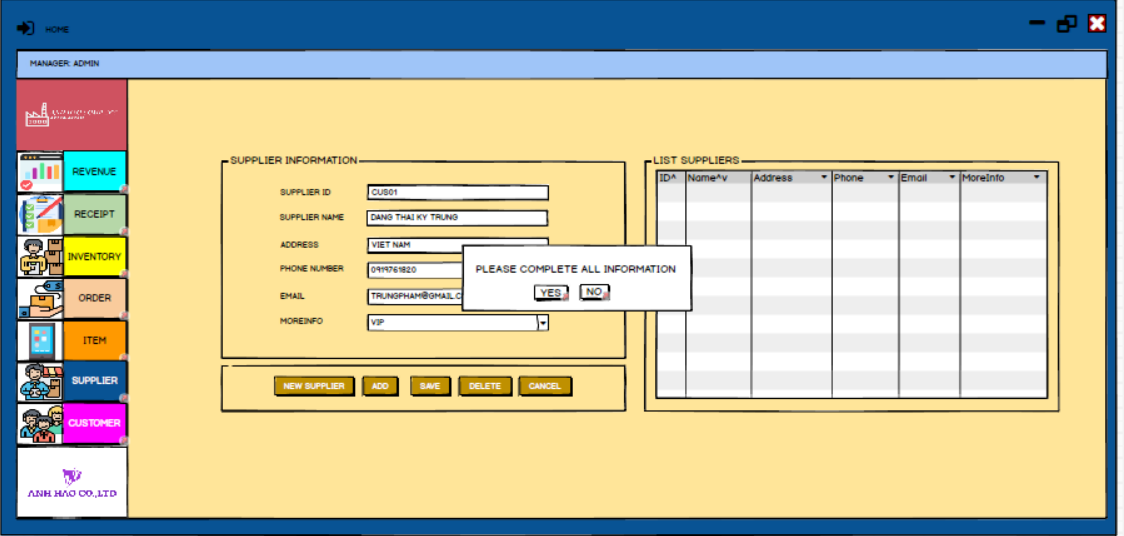
**Customer:**

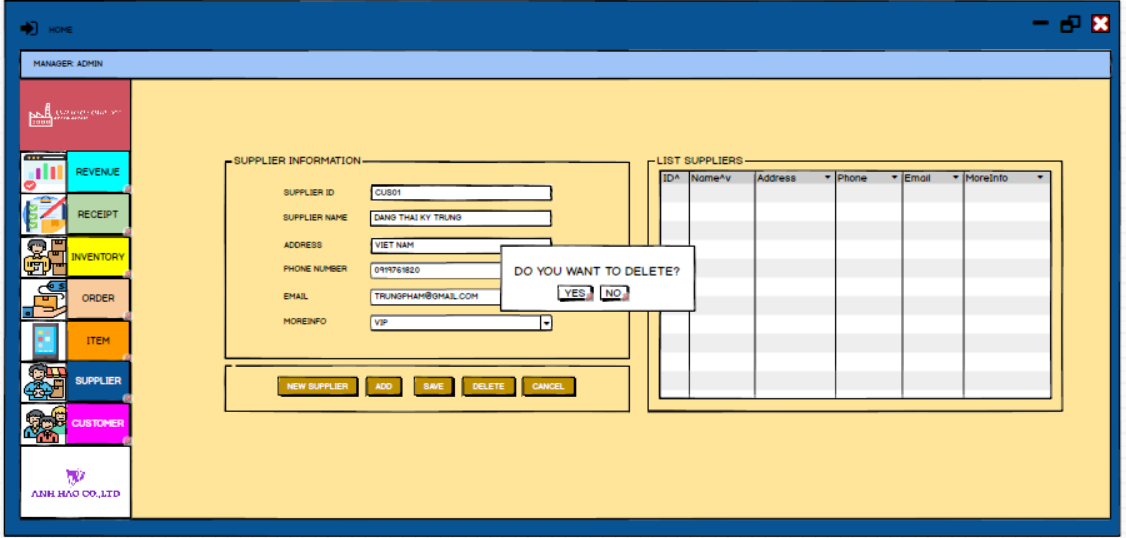




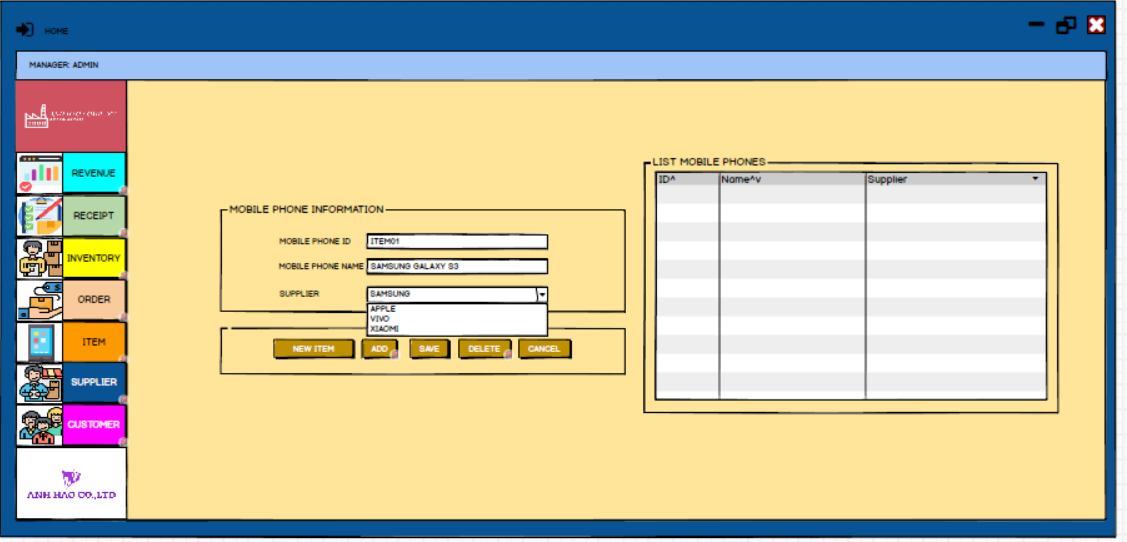
**Supplier:**



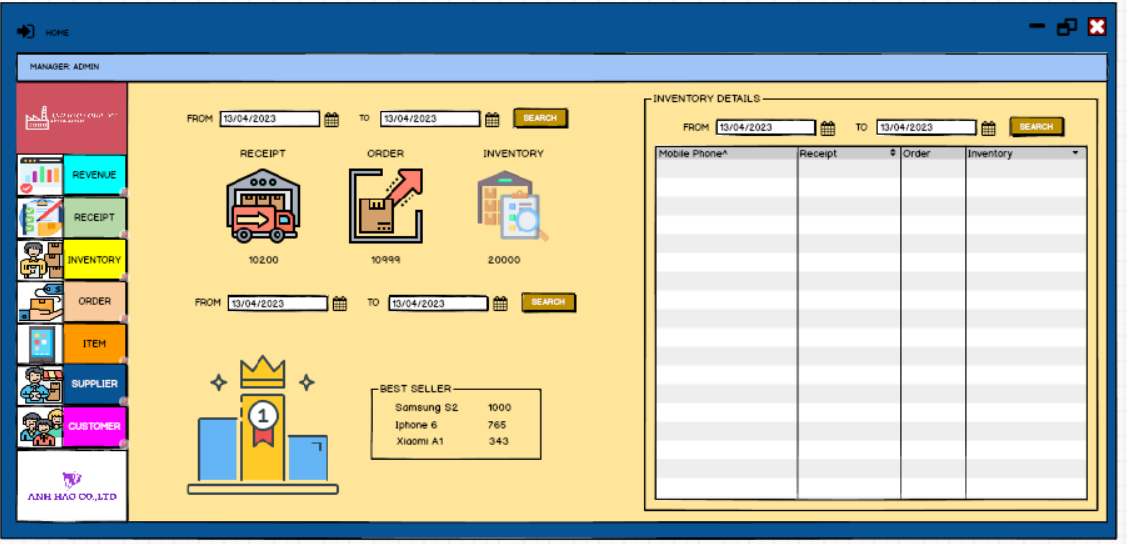


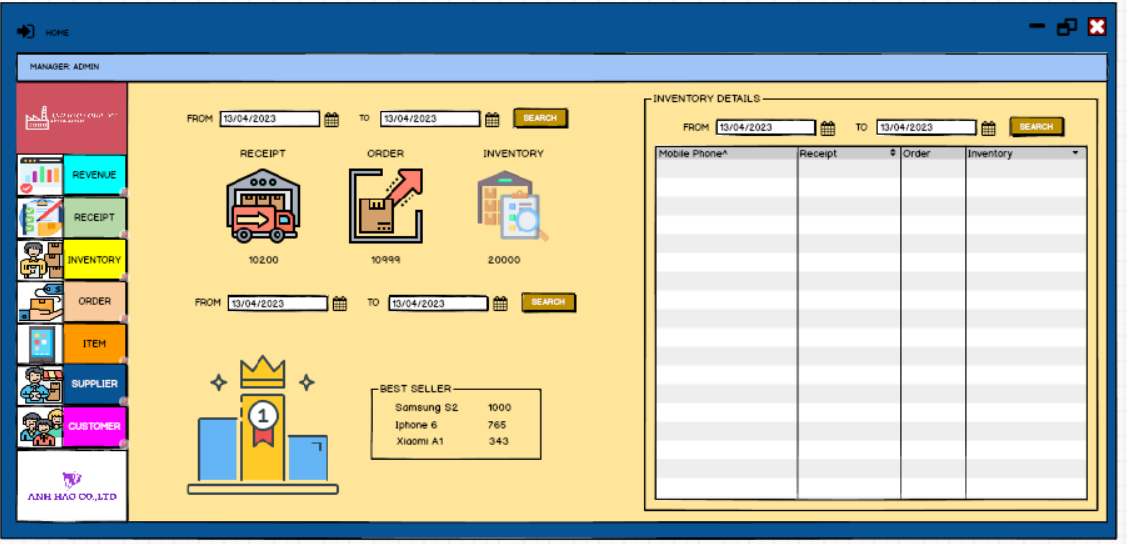


**MobilePhone:**

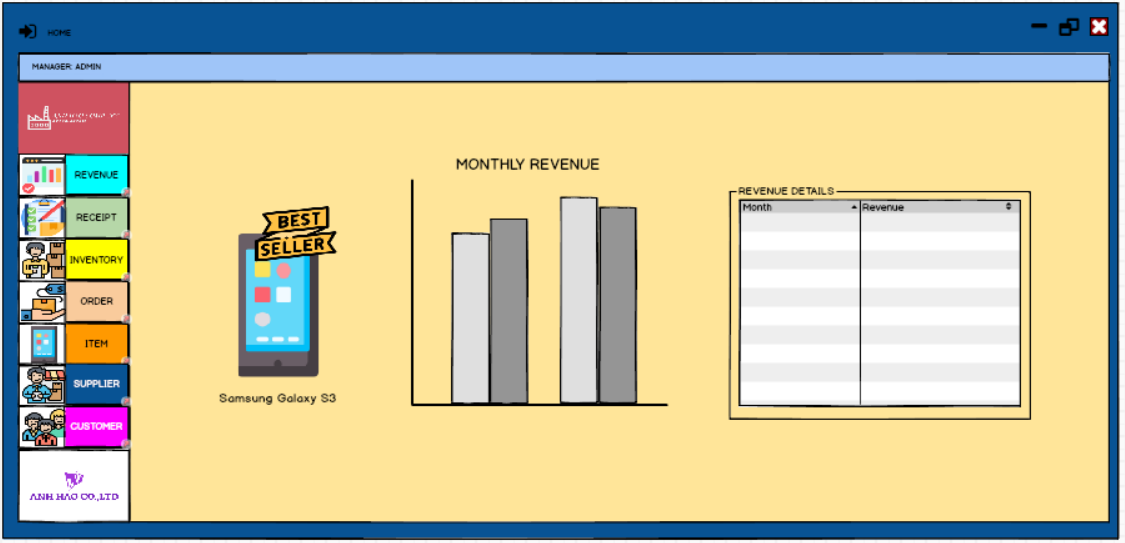


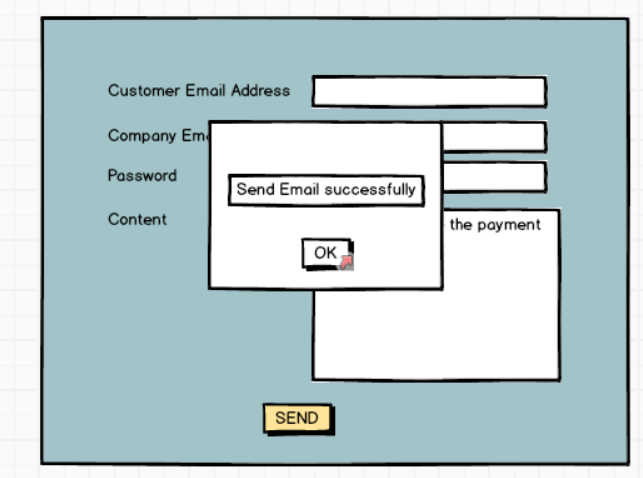
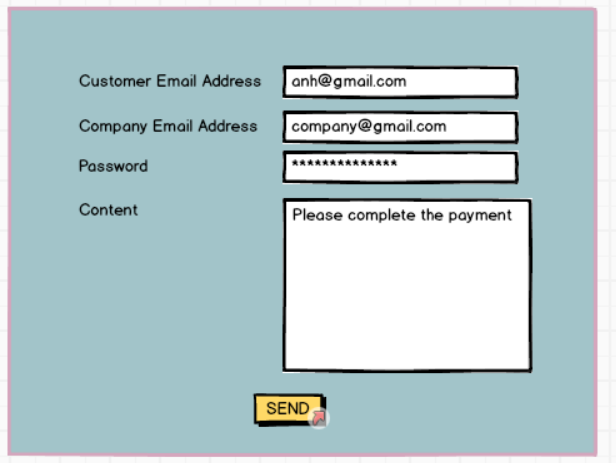
**Invertory:**





**Revenue and send Email:**





**5.6 Logo Design**



# CHAPTER 6 – TEST PLAN

## 6.1 Requirements/specifications-based system level test cases

**1. Test case for Goods Received :**

- Requirement: Accountants shall be able to create Goods Received when the distributor imports goods.

- Test case:

- Step 1: Accountant logs in to the system and navigates to the Goods Received section.

- Step 2: Accountant creates a new Goods Received document, entering all the required information, including the date, supplier, and items received.

- Step 3: Accountant saves the document and verifies that it is stored in the system's database.

- Expected result: The Goods Received document is created and stored in the system's database.

**2. Test case for Order :**

- Requirement: Reseller / Agents shall be able to place an order of items and choose a payment method.

- Test case:

- Step 1: Reseller/Agent logs in to the system and navigates to the Order section.

- Step 2: Reseller/Agent selects the items they wish to order and adds them to the cart, then selects their preferred payment method.

- Step 3: Reseller/Agent completes the order and verifies that the order is stored in the system's database and their payment is processed.

- Expected result: The order is created and stored in the system's database, and the payment is successfully processed.

**3. Test case for Delivery Bill :**

- Requirement: Accountants shall be able to create Goods Delivery Note to deliver goods to agents.

- Test case:

- Step 1: Accountant logs in to the system and navigates to the Delivery Note section.

- Step 2: Accountant creates a new Delivery Note document, entering all the required information, including the date, agent, and items being delivered.

- Step 3: Accountant saves the document and verifies that it is stored in the system's database and the agent's payment status is updated.

- Expected result: The Delivery Note document is created and stored in the system's database, and the agent's payment status is updated.

**4. Test case for Statistion level :**

- Requirement: Accountants shall be able to view incoming/outgoing stock report, best-selling products and revenue report monthly.

- Test case:

- Step 1: Accountant logs in to the system and navigates to the Statistical section.

- Step 2: Accountant selects the desired report type and date range.

- Step 3: Accountant views the report and verifies that the information is accurate and up-to-date.

- Expected result: The statistical report is generated and displayed accurately.

## 6.2 Traceability of test cases to use cases

-Log in, log out for Staff, Customer

- Add, edit, delete Staff

- Add, delete, update,status,report Receipt

- Add, edit, delete, payment Customer

- Add, edit, delete Supplier, Item

- Add, edit, update OrderInfo, ReceiptInfo

- Add, delete, update, status, payment, report, email Order

**6.3 Techniques used for test generation**

Equivalence partitioning, boundary value analysis, decision table testing, state transition testing, error guessing, exploratory testing, and model-based testing are all techniques that can be used for test generation. The choice of technique will depend on the nature of the system, the level of testing, and the available resources and expertise.

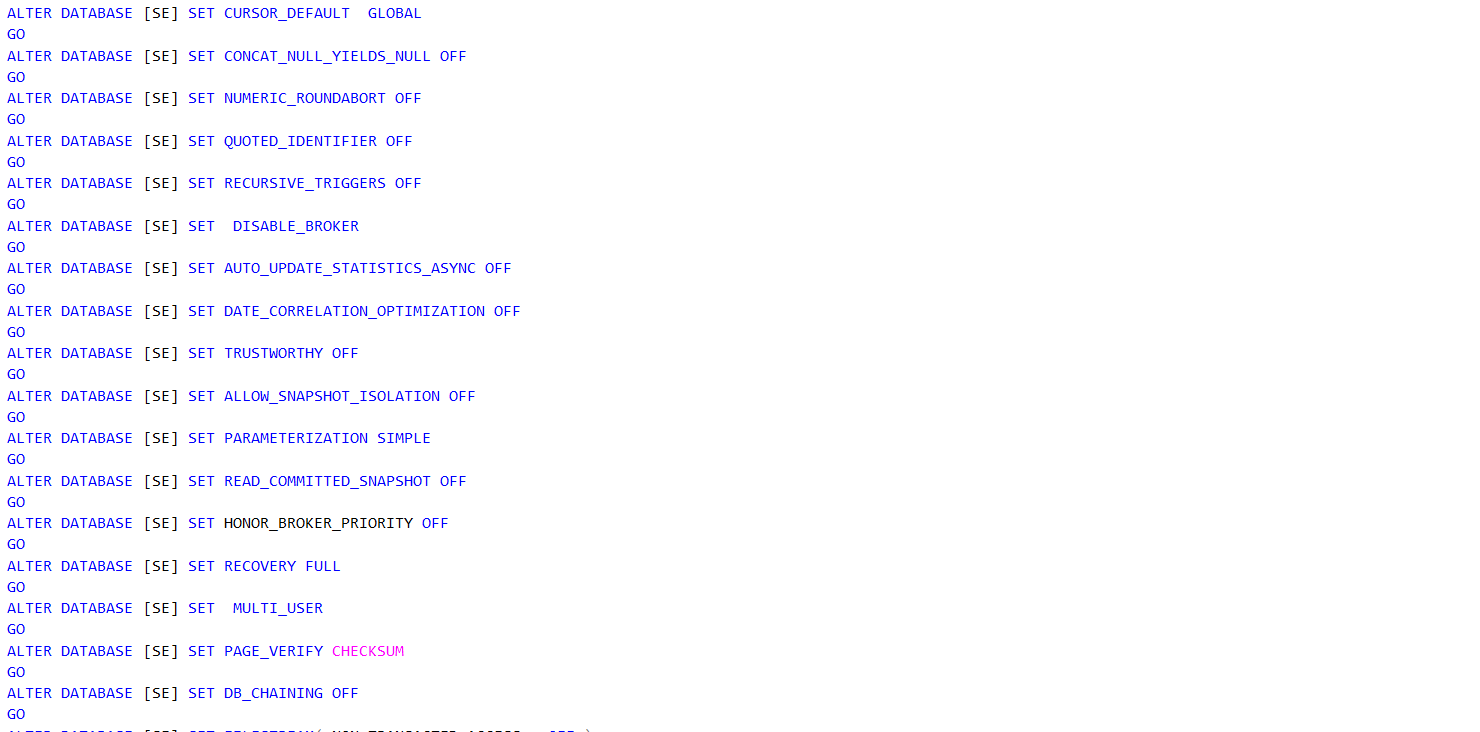
**6.4 Assessment of the goodness of our testsuite**

Testing at a high level is suitable for testing large segments of code and large functions. It does not require understanding the code written in the program, and does not require understanding the structure inside the function or how to get it. It is suitable for testing large segments of code and large functions.

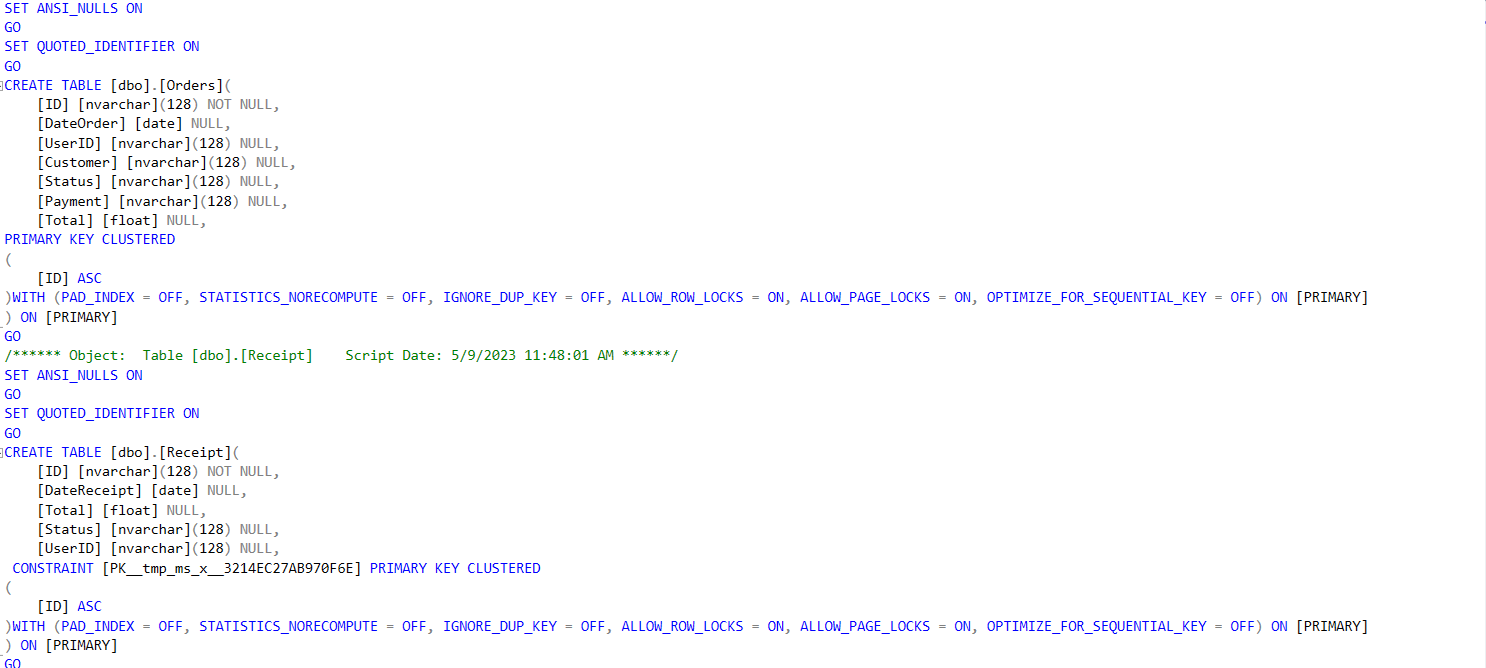
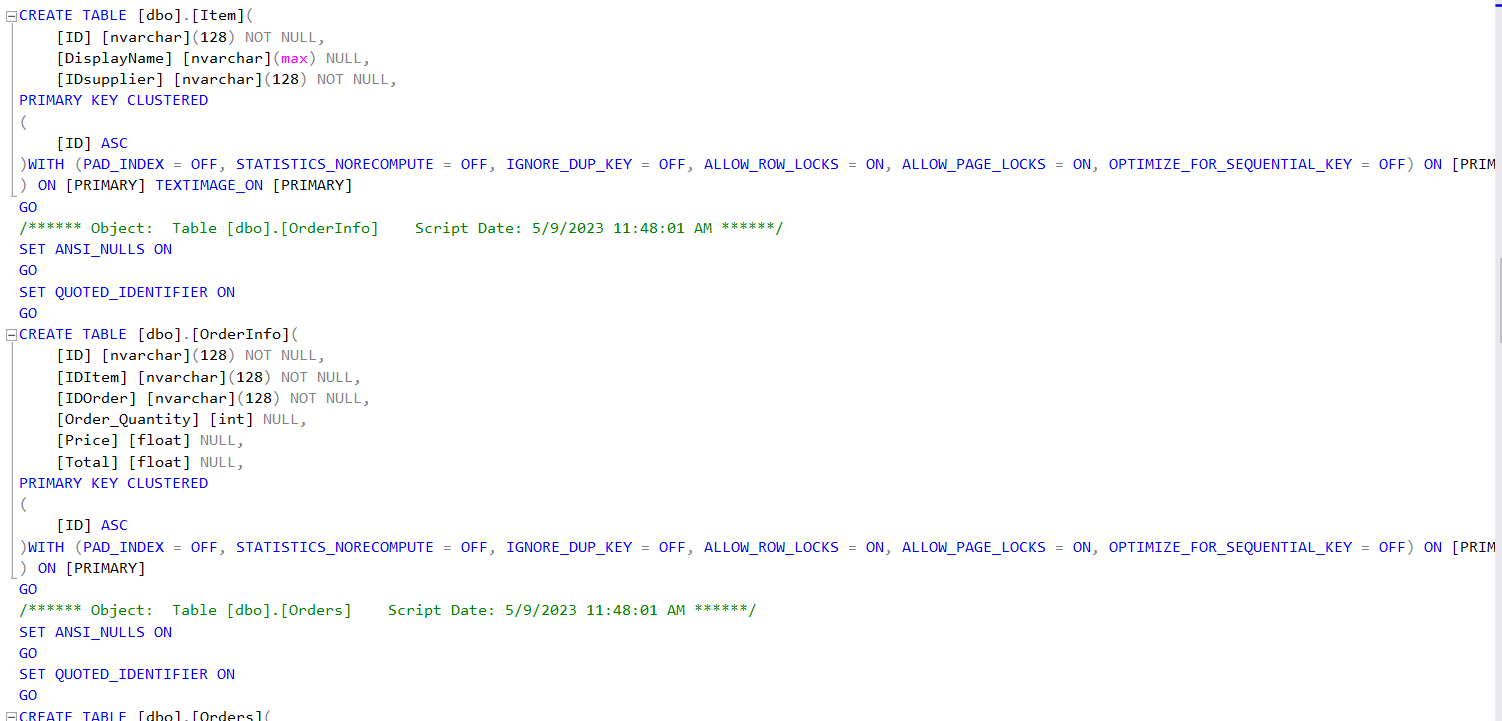
**CHAPTER 7 – DEMO**

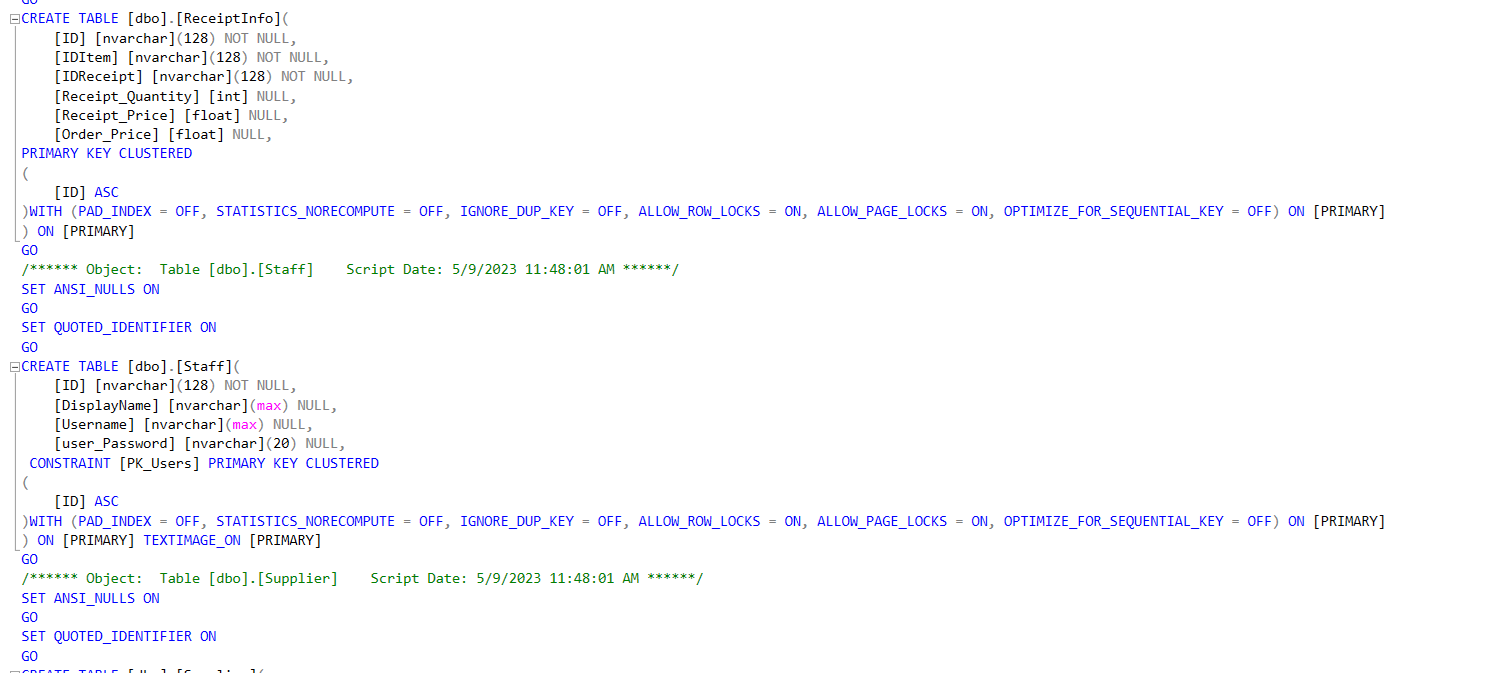
**7.1 Database**

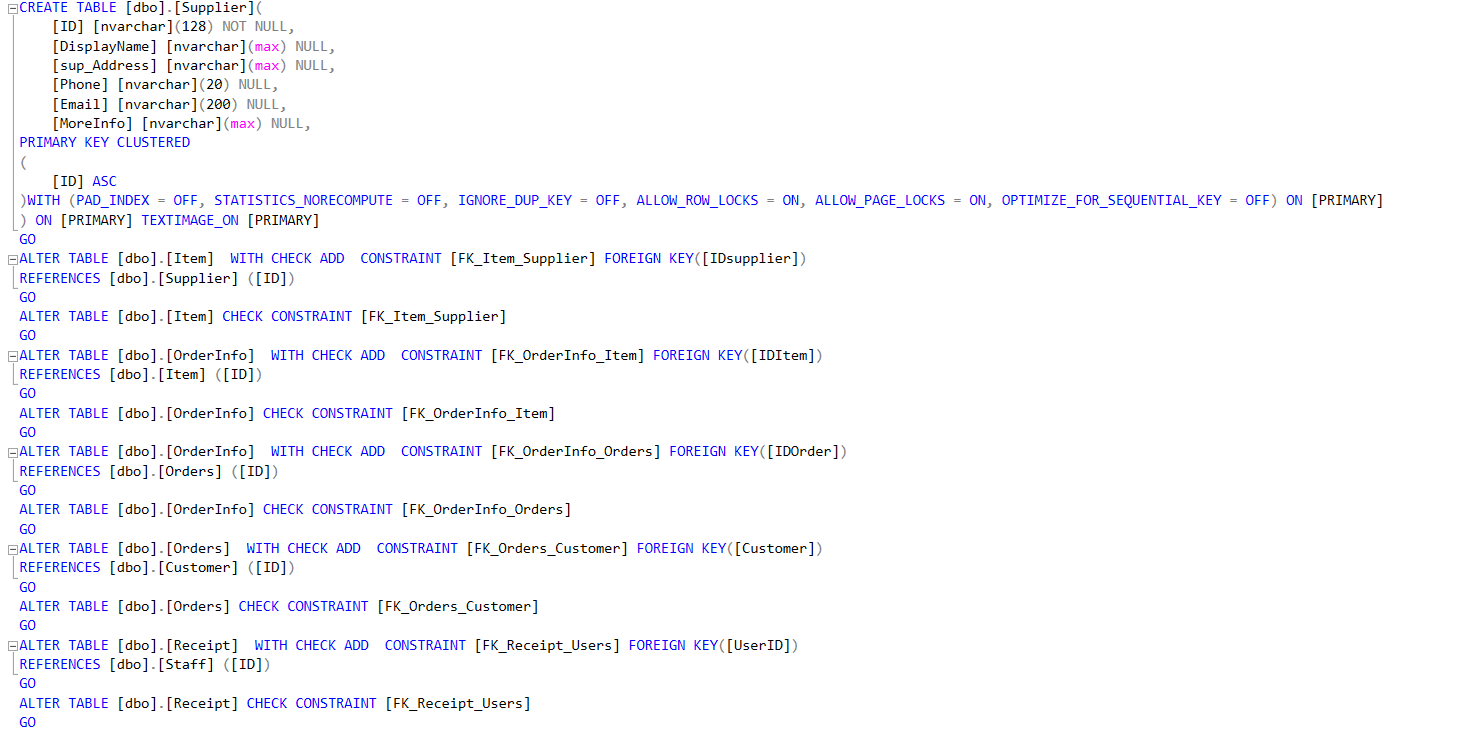


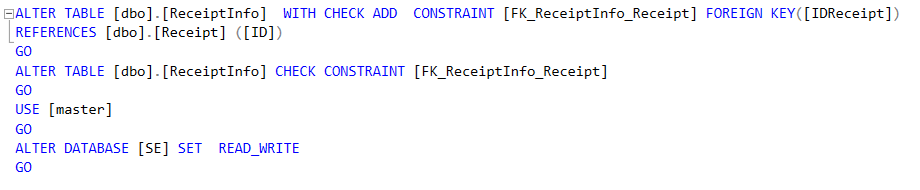




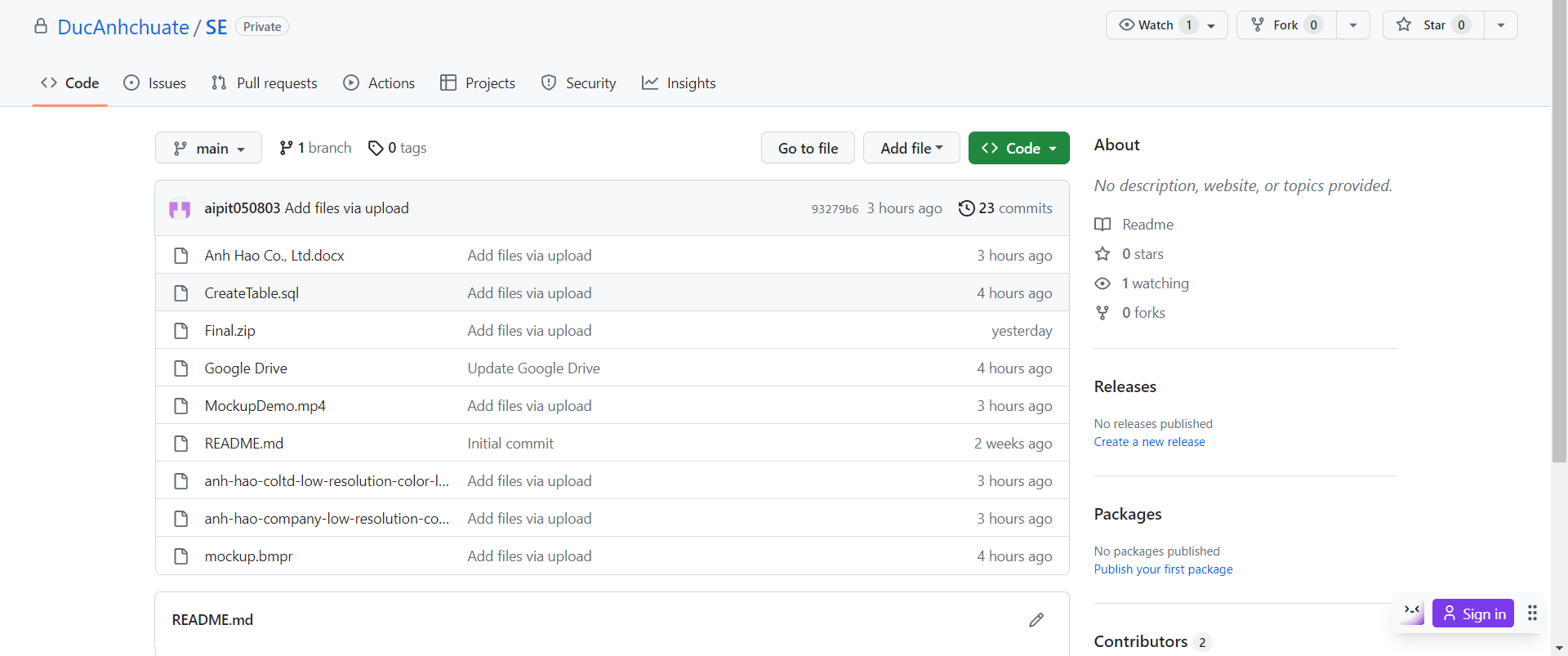


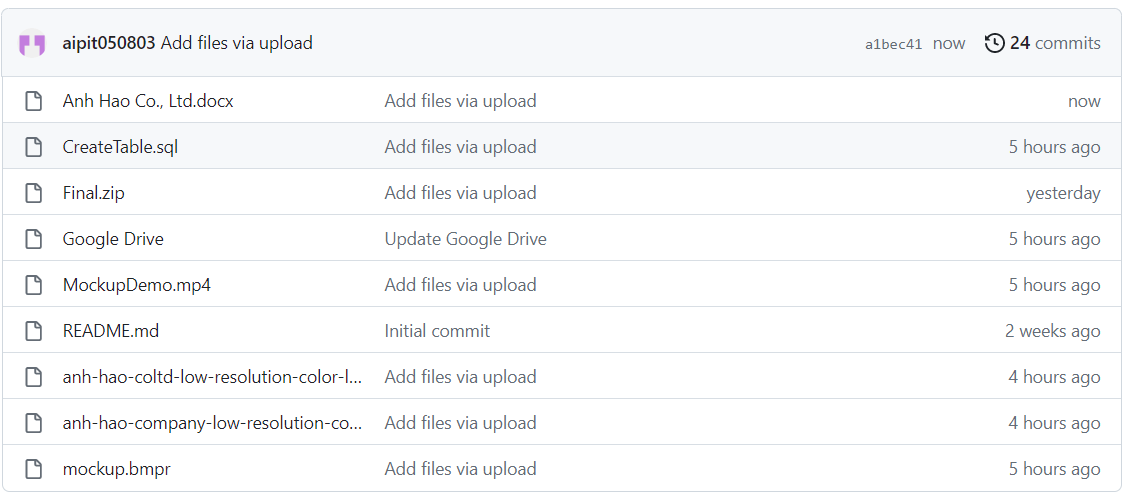






**LINK GITHUB:**





**https://github.com/DucAnhchuate/SE**