VIETNAM LABOR UNION GENERAL

**TON DUC THANG UNIVERSITY**

**FALCUTY OF INFORMATION TECHNOLOGY**



**FINAL PROJECT OF SOFTWARE ENGINEERING**

**MANAGEMENT SOFTWARE SELLING MOBILE PHONE PRODUCT**

*Instructor:* **MR PHAM THAI KY TRUNG**

*Authors*: **NGUYEN VIET MINH DUY - 521H0445**

**NGUYEN NGO DANG KHOA – 521H0084**

*Class***: 21H50202**

**21H50203**

*Course***: 25**

**HO CHI MINH CITY, 2023**

VIETNAM LABOR UNION GENERAL

**TON DUC THANG UNIVERSITY**

**FALCUTY OF INFORMATION TECHNOLOGY**



**FINAL PROJECT OF SOFTWARE ENGINEERING**

**MANAGEMENT SOFTWARE SELLING MOBILE PHONE PRODUCT**

*Instructor:* **MR PHAM THAI KY TRUNG**

*Authors*: **NGUYEN VIET MINH DUY - 521H0445**

**NGUYEN NGO DANG KHOA – 521H0084**

*Class***: 21H50202**

**21H50203**

*Course***: 25**

**HO CHI MINH CITY, 2023**

ACKNOWLEDGEMENT

Sincere thanks We want to give to Mr. Pham Thai Ky Trung, who enthusiastically taught and worked tirelessly to give us enough tools and skills to complete this report. He played an important role in improving our software skill and knowledge about learning using Web and Software . The second thanks we would like to give to the teachers of the Department of Information Technology of Ton Duc Thang University for giving us the opportunity to do this report.

Finally, I wish you good health and success in your noble career.

**THE TOPIC ARE 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, Monday, 7th May, 2023*

*Author*

*(Sign and provide full name)*

*Nguyen Viet Minh Duy*



*Nguyen Ngo Dang Khoa*

CONFIRMATION AND ASSESSMENT SECTION

**Instructor confirmation section**

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

*Ho Chi Minh January, 2022*

*(Sign and write full name)*

**Evaluation section for grading instructor**

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

*Ho Chi Minh January 2022*

*(Sign and write full name)*

TABLE OF CONTENTS

[CHAPTER 1 – INTRODUCTION 6](#_Toc134549185)

[ 6](#_Toc134549186)

[1.1 Purpose and Scope 6](#_Toc134549187)

[1.2 Product Overview 7](#_Toc134549188)

[1.3 Structure of the Document 7](#_Toc134549189)

[1.4 Terms, Acronyms, and Abbreviations 9](#_Toc134549190)

[CHAPTER 2 - PROJECT MANAGEMENT PLAN 9](#_Toc134549191)

[2.1 Project Organization Structure 9](#_Toc134549192)

[2.2 Lifecycle Model Used 10](#_Toc134549193)

[2.3 Risk Analysis 10](#_Toc134549194)

[2.4 Hardware and Software Resource Requirements 10](#_Toc134549195)

[2.6 Monitoring, Reporting, and Controlling Mechanisms 13](#_Toc134549196)

[2.7 Professional Standards 15](#_Toc134549197)

[2.8 Evidence all the artifacts have been placed under configuration management: 15](#_Toc134549198)

[2.9 Impact of the project on individuals and organizations 18](#_Toc134549199)

[CHAPTER 3 - REQUIREMENT SPECIFICATIONS 19](#_Toc134549200)

[3.1 Stakeholders for the system 19](#_Toc134549201)

[3.2 Use case model 19](#_Toc134549202)

[3.2.1 Graphical use case model 19](#_Toc134549203)

[3.2.2 Use case specification 20](#_Toc134549204)

[3.3 Functional requirements 25](#_Toc134549205)

[3.4 Non – functional requirements 27](#_Toc134549206)

[CHAPTER 4 - ARCHITECTURE 27](#_Toc134549207)

[4.1 Architectural style used 27](#_Toc134549208)

[4.2 Architectural model 29](#_Toc134549209)

[4.3 Technology, software, and hardware used 31](#_Toc134549210)

[4.4 Rationale architectural style and model 33](#_Toc134549211)

[CHAPTER 5 – DESIGN 35](#_Toc134549212)

[5.1 Database design 35](#_Toc134549213)

[5.2 Class Diagrams 36](#_Toc134549214)

[5.3 Sequence diagrams 36](#_Toc134549215)

[5.4 Rationale for your detailed design model 41](#_Toc134549216)

[CHAPTER 6 – TEST PLAN 42](#_Toc134549217)

[6.1 Requirements/specifications-based system level test cases 42](#_Toc134549218)

[6.2 Traceability of test cases to use cases 42](#_Toc134549219)

[6.3 Techniques used for test generation 43](#_Toc134549220)

[6.4 Assessment of the goodness of our testsuite 44](#_Toc134549221)

[CHAPTER 7 – DEMO 44](#_Toc134549222)

[7.1 Database 44](#_Toc134549223)

[7.2 Source Code 49](#_Toc134549224)

LIST OF TABLES

**1.Use case for Goods Received……………………………………………………..37**

**2.Use case for Delivery Bill…………………………………………………………38**

**3.Use case for Statistics Management…………………………………………….39**

**4.Use case for Order………………………………………………………………40**

**5.Use case for Payment……………………………………………………………41**

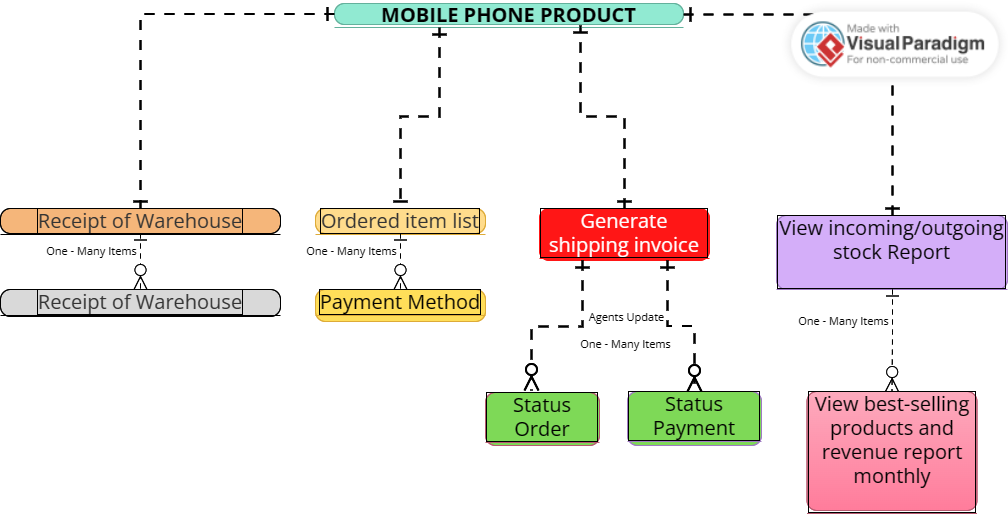
# CHAPTER 1 – INTRODUCTION

# 

*About Our Logo*

## 1.1 Purpose and Scope

Purpose: We are developing software that will help functional foods companies manage import/export, ordering, payment, and revenue statistics for their agents. Furthermore, our software development process allows us to closely collaborate with the client in creating the final product.

Scope:

*Work Breakdown Structure ̣(WBS)*

## 1.2 Product Overview

- Accountants shall be able to create Goods Received when the distributor imports goods (a warehouse receipt will include many items)

- Reseller / Agents shall be able to place an order of items and choose a payment method (Cash, bank transfer, Momo...)

- 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.

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

## 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

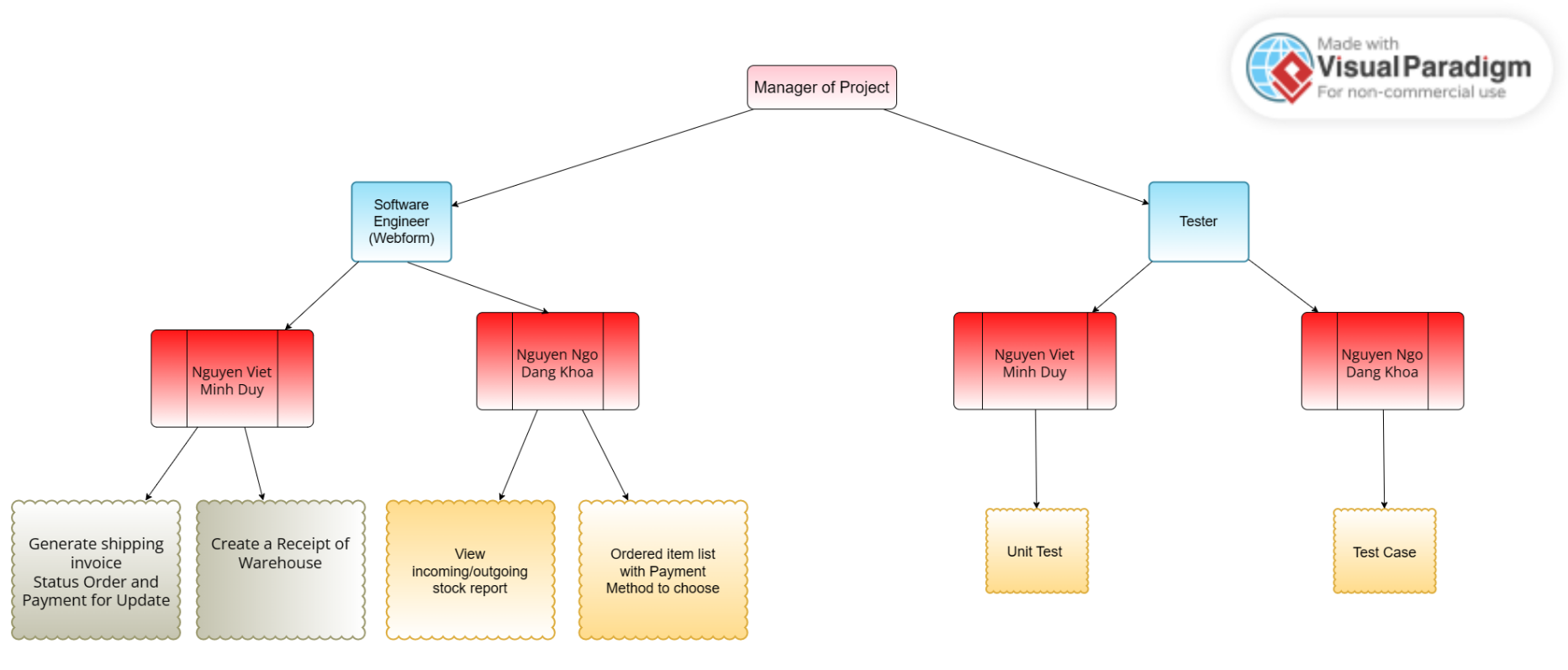
7.2. Source code

7.3. Testing

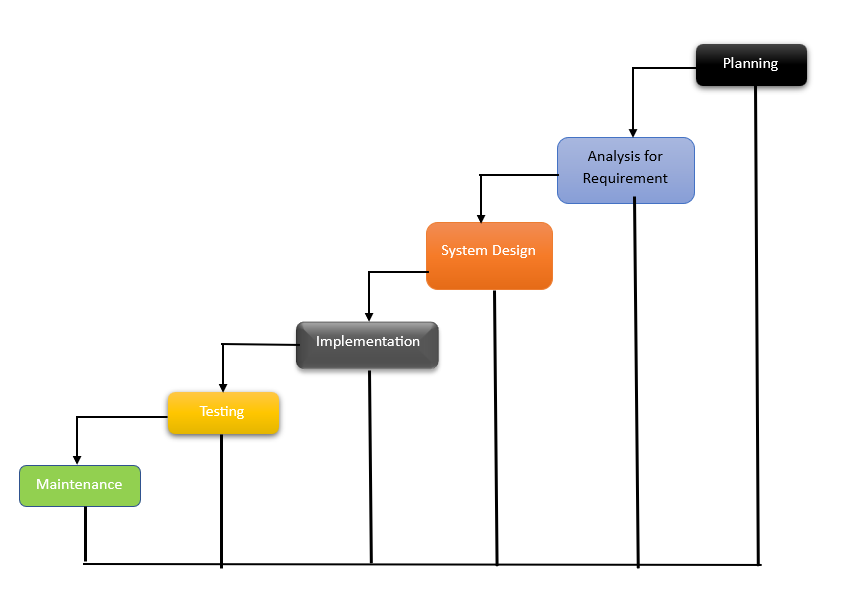
## 1.4 Terms, Acronyms, and Abbreviations

# CHAPTER 2 - PROJECT MANAGEMENT PLAN

## 2.1 Project Organization Structure

Project Organization

## 2.2 Lifecycle Model Used



## 2.3 Risk Analysis

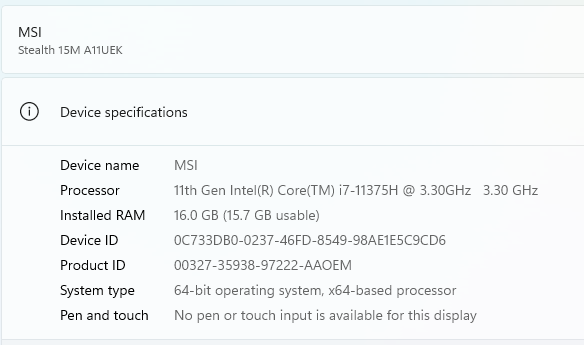
* Duplicate primary key
* Foreign key can be edited
* Remove foreign key
* Leave the Login section blank

## 2.4 Hardware and Software Resource Requirements

**Hardware** and **software** resource requirements vary depending on the specific application or task you want to accomplish. Here are some general guidelines for Software Engineering using Visual Studio *or not using it*:

*Hardware Requirements:*

* Processor: 1.8 GHz or faster processor with 2 or more cores. 1.8 GB of RAM; 8 GB of RAM recommended (3.5 GB minimum if running on a virtual machine). *If it’s not depend on using Visual, A modern processor, preferably quad-core or higher, with a clock speed minimum of 2.5 GHz or higher.*
* Memory (RAM): 8 GB of RAM (recommended) or more. *Without Visual, At least 8 GB of RAM. For more resource-intensive tasks such as video editing or gaming, you may need 16 GB or more.*
* *Storage: Solid-state drives (SSDs) are recommended over traditional hard disk drives (HDDs) for faster read and write speeds. At least 256 GB of storage is recommended.*
* Hard disk space: Minimum of 20 GB of free disk space. Minimum of 850MB up to 210 GB of available space, depending on features installed; typical installations require 20-50 GB of free space.
* Hard disk speed: to improve performance, install Windows and Visual Studio on a solid state drive (SSD).
* Display: 1280 x 800 resolution or higher. *The display is the primary way users interact with the phone. The display size, resolution, and type (e.g., LCD or OLED) determine the quality of the display.*
* Graphics card: Video card that supports a minimum display resolution of WXGA (1366 by 768); Visual Studio will work best at a resolution of 1920 by 1080 or higher. Win365: 2 vCPU and 8 GB RAM. 4 vCPU and 16 GB of RAM recommended. *DirectX 10-compatible graphics card with a minimum of 1024MB of video memory.*

Here is an example: 

*Software Requirements:*

- Visual Studio 2020

* Operating system: Windows 10 version 1803 or higher (64-bit)
* .NET Framework: Version 4.7.2 or higher
* Internet connection: An active internet connection is required for some features
* Optional: Android SDK, iOS SDK, or Universal Windows Platform tools for developing mobile apps.

- Microsoft SQL Server Management Studio 19

* Operating system: Windows 10 version 1607 or higher (64-bit), Windows Server 2016 or higher (64-bit)
* Processor: 1.8 GHz or faster processor
* Memory (RAM): 2 GB of RAM or more (4 GB recommended)
* Hard disk space: Minimum of 4 GB of free disk space
* Display: Minimum resolution of 1024x768 pixels
* .NET Framework: Version 4.6.1 or higher
* Internet Explorer: Version 11 or higher

- Visual Studio Code

2.5 Deliverbles and Schedule

* Initially, the exact date for product delivery will be determined during the project discussion. In the event that we surpass the appointed time, we will bear complete accountability for any absence of product or any deviations from the customer's specifications.
* Prior to the delivery, we will establish a date for a demonstration report that will showcase the project's completion timeline. We will gather feedback and modify the software to meet the users' requirements.
* The software will be fully functional and will meet the customer's requirements by the pre-arranged delivery date.
* The final day to complete the project is on May 7th, 2023.

## 2.6 Monitoring, Reporting, and Controlling Mechanisms

**Reporting about Monitoring and Controlling:**

- The process of ***monitoring and controlling*** project work is a broad category that includes all other related monitoring and controlling activities.

**- *Integrated change control*** refers to the tasks associated with modifying the project plan. If any adjustments need to be made to the schedule, budget, or any other aspect of the project management plan, the plan is revised, and approval is sought from the project sponsor.

- ***Validating scope*** refers to the actions taken to obtain approval for the project's deliverables. This includes tasks related to reviewing the completed work and verifying that it meets the requirements and expectations laid out in the project scope. The validation process is crucial for ensuring that the project is on track and meeting its objectives.

***- Controlling scope*** involves making sure that the project's scope remains unchanged and that any unauthorized activities, known as scope creep, are not carried out as part of the project plan. The goal is to maintain the project's original scope and prevent any deviations from it, which can cause schedule delays, budget overruns, and other complications.

***- Controlling the schedule*** refers to the activities aimed at ensuring that project tasks are completed within their designated timeframes, and project deadlines are met. The process involves monitoring progress, identifying potential delays, and taking corrective action when necessary to keep the project on track. The ultimate goal of schedule control is to ensure that the project is completed on time, within budget, and to the required quality standards.

***Controlling costs*** refers to the activities involved in managing and monitoring project expenses to ensure they remain within the approved budget. This includes tasks such as tracking expenses, identifying cost overruns, and taking corrective action to bring costs back in line with the budget. Effective cost control is essential for ensuring that the project's financial resources are managed efficiently and that the project is completed within the approved budget.

***Quality control*** involves making sure that the project's outcomes meet the quality expectations specified in the project management plan.

***Communication control*** involves addressing the communication requirements of every stakeholder involved in the project.

***Risk control*** aims to protect the project from unforeseen circumstances that could have adverse effects on the project's budget, schedule, stakeholder requirements, or any other factors critical to the project's success.

***Procurement control*** involves verifying that the project's vendors and subcontractors meet the objectives of the project.

***Stakeholder engagement control*** includes the activities required to ensure that all the project stakeholders are content with the project's deliverables.

*The report will be completed when the project is completed.*

## 2.7 Professional Standards

* Proficient in utilizing Windows Form Applications.
* Proficient in utilizing Visual Studio.
* Proficient in utilizing Microsoft SQL Server Management Studio 19.
* Possess knowledge of SQL databases.
* Possess knowledge of C#.

## 2.8 Evidence all the artifacts have been placed under configuration management:

Configuration Management (CM) is a crucial process in software development projects, including mobile phone projects. It involves identifying and tracking changes to software and hardware components throughout the development lifecycle to ensure that they are managed effectively, consistently, and securely. Here are the steps to implement **configuration management** in a mobile phone project:

1. **Version Control System**: A version control system (VCS) will be used to track changes to the source code and other project artifacts. The VCS will be hosted on a remote server to ensure that all team members can access it. The team will use Git as the VCS.
2. **Baselines** - Establish baselines for each stage of the project, including requirements, design, development, testing, and deployment. Each baseline should be approved by the project stakeholders and stored in a secure location. Evidence of baselines can be demonstrated by showing how each baseline is labeled and stored, and how it is used to track changes to the project.
3. **Repository Structure**: The repository will be organized into branches for each major release and for each feature. The master branch will be the mainline of development, and all changes will be merged into it after review and approval. Feature branches will be created for each feature or bug fix, and pull requests will be used to merge changes into the mainline.
4. **Release Management**: A release manager will be responsible for managing releases and ensuring that all artifacts are correctly tagged and versioned. Each release will be identified by a unique version number, and the release manager will be responsible for updating the version number in the source code and other artifacts.
5. **Build and Deployment Automation**: The project will use continuous integration and continuous deployment (CI/CD) to automate the build and deployment process. The team will use Jenkins as the CI/CD tool, and all builds and deployments will be triggered automatically when changes are pushed to the repository.
6. **Configuration Management Database**: A configuration management database (CMDB) will be used to track changes to project artifacts and their relationships. The CMDB will be hosted on a remote server and will be accessible to all team members. The CMDB will track changes to the source code, documentation, build scripts, deployment scripts, and other project artifacts.
7. **Change Control Process**: All changes to the source code and other project artifacts will go through a formal change control process. The process will include a request for change (RFC) form, a review and approval process, and a documentation process. The RFC form will be used to capture details of the change, including the reason for the change, the impact of the change, and the testing required. The review and approval process will involve peer review and sign-off by the release manager. The documentation process will ensure that all changes are documented in the CMDB.
8. **Testing and Validation**: All changes to the source code and other project artifacts will be thoroughly tested before they are merged into the mainline. The team will use automated testing tools and manual testing to ensure that changes are valid and do not introduce new bugs.
9. **Disaster Recovery**: The project will have a disaster recovery plan in place to ensure that all project artifacts are backed up and recoverable in case of a disaster. The disaster recovery plan will include regular backups of the repository, the CMDB, and other critical project artifacts.

By implementing this configuration management plan, the mobile phone project will be able to effectively manage changes to project artifacts, ensure that all changes are thoroughly tested and validated, and ensure that all artifacts are recoverable in case of a disaster.

## 2.9 Impact of the project on individuals and organizations

The impact of the Mobile Phone Product on individuals and organizations has been significant and far-reaching. Here are some of the key impacts:

**Impact on individuals:**

1. Increased connectivity: Mobile phones have allowed individuals to stay connected with each other regardless of distance or time zone. This has made communication easier and more efficient.

2. Improved productivity: Mobile phones have enabled individuals to work from anywhere and at any time, increasing productivity and flexibility.

3. Access to information: Mobile phones have made it easier for individuals to access information on the internet, enabling them to learn and educate themselves on a wide range of topics.

4. Enhanced entertainment: Mobile phones have become a primary source of entertainment for individuals, with features such as social media, streaming services, and gaming.

**Impact on organizations:**

1. Improved communication: Mobile phones have enabled organizations to stay connected with their employees and customers, improving communication and collaboration.

2. Increased efficiency: Mobile phones have allowed organizations to operate more efficiently by providing access to important information and resources on the go.

3. Better customer service: Mobile phones have enabled organizations to provide better customer service by responding to inquiries and issues quickly and efficiently.

4. Enhanced marketing: Mobile phones have become a primary channel for organizations to reach and engage with customers through targeted advertising and promotions.

# CHAPTER 3 - REQUIREMENT SPECIFICATIONS

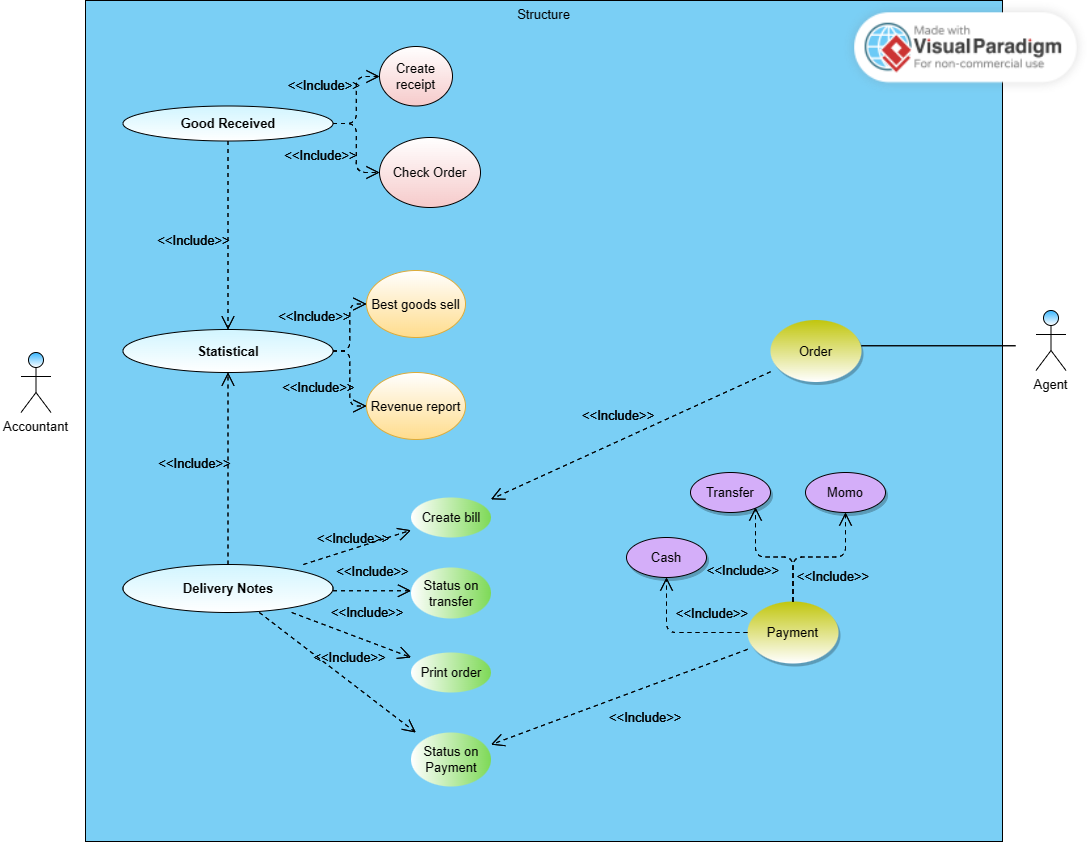
## 3.1 Stakeholders for the system

- The individuals or groups who have a vested interest in the system include developers, users, customers, management, project management, implementation management, operation, infrastructure management, accounting staff, security staff, and hotline support staff.

- Although it may not be feasible to satisfy every stakeholder completely, the system will strive to achieve a balance among all parties to optimize the overall satisfaction of everyone involved. stakeholders.

## 3.2 Use case model

### 3.2.1 Graphical use case model



### 3.2.2 Use case specification

Use case for **Goods Received**

|  |  |
| --- | --- |
| Use Case Name | Goods Received |
| Primary Actor | Accountant |
| Description | Use Case is utilized “create receipt” for generating record details for a fresh set of merchandise and storing it in the database when distributor receives goods.. |
| Preconditions | - The distributor has imported goods and ready to be received into the warehouse.  - Sign in using access privileges based on the accountant's role. |
| Postconditions: | - The system saves the Goods Received report, which is accessible to the accountant.  - Moreover, the stock quantities of the received items are updated in the database. |
| Stakeholders and any Relate | - Stakeholder receives the management part  - For other related, select the Login section |
| Basic Flow | **-** First, **accountant** login and select option “Good Received” in the menu  - Then, **system** display a form to access the receipt information.  - The **accountant** inputs the shipment particulars, such as the date, provider, invoice number, and supplementary remarks. Then they can add the items when system enter the receipt information.  - After that, the form shows the total cost of the shipment, which is automatically computed by the **system**.  - Finally, **accountant** save all the things they have done from Goods Received. |
| Alternative Flows | - In case the package contains items that are faulty or have been damaged during transportation, the accountant has the option to mention this in the Goods Received record by adding a note.   - After that, they can proceed with the procedure of either returning the items or getting them replaced. |
| Result | Goods receipt created successfullly |

Use case for **Delivery Bill**

|  |  |
| --- | --- |
| Use Case Name | Delivery bill |
| Primary Actor | Accountant |
| Description | Utilized to generate delivery slip details whenever a delivery bill is got by agents and store it in the database. |
| Preconditions | - An ordered has been received from an agent by the **distrubutor**  **-** Now, **accountant** has enter to the **system** and Log in. |
| Postconditions | - A printed delivery bill can be included with the shipment.  - The system is updated with the current order status and payment information of the agent, which is then saved in the database. |
| Stakeholders and any Relate | - Stakeholder receives the management part  - For other related, select the Login section |
| Basic Flow | - From the menu, the agent chooses the "Payment" alternative from menu.  - The agent picks their desired payment mode, which could be cash, bank transfer, or digital platforms like Momo and ZaloPay.  - If the agent opts for online payment, they need to input their payment information and verify the payment.  - The payment is handled by the system, and the order status is revised to "paid."  - The agent can examine the order's status within the system. |
| Alternative Flows | In case of any problems with the payment, like a transaction that has been declined or technical issues, the agent can opt to retry the payment or choose an alternative payment method. |
| Result | Create delivery note successfullly |

Use case for **Statistical management**

|  |  |
| --- | --- |
| Use Case Name | Statistical Management |
| Primary Actor | Accountant |
| Description | The Use Case allows for the visualization of the statistics sheet. |
| Preconditions | - Access the login feature with permissions based on the accountant's position  - The list should contain at least one statistical vote. |
| Postconditions | Observe data related to statistics. |
| Basic Flow | - The accountant chooses the "Statistical" choice from the menu.  - Then system check the Login permission. If success, a roster of statistical reports that can be produced appears on the system, including reports on incoming/outgoing stock, top-selling items, and monthly revenue.  - The accountant picks the report they want.  - The system creates the statistical report and display it to the accountant.  - System will display Statics interface and information of statistics sheet have been chosen.  - The accountant has the option to export the report in a preferred format, such as PDF or Excel. |
| Alternative Flows | If the accountant faces any problems during the creation of the statistical report, such as technical glitches or incomplete information, they may reach out to the IT support team for assistance. |

Use case for **Order**

|  |  |
| --- | --- |
| Usecase name | Order Information |
| Primary Actor | Agency |
| Context | An order placed by an agency on the website of the company |
| Description | This Use Case is used for ordering from agency |
| Preconditions | - Agency can access to the system via Login  - Generate the Order |
| Postconditions | - Once the order is submitted, the system generates a confirmation.  - The system also updates the inventory levels.  - The agent can track the progress of their order within the system.  - Updating all to database. |
| Stakeholders | Company |
| Basic Flow | - From the menu, the agent chooses the "Order" option.  - The system presents a list of mobile phone products that are available.  - The agent chooses the desired products and specifies the quantity for each.  - The agent picks their preferred payment method, which may include cash, bank transfer, or Momo.  - If the agent selects online payment, they provide their payment details and confirm the transaction.  - The agent then submits the order.  - The system produces an order confirmation and updates the inventory levels accordingly. |
| Alternative Flows | - In the event that the product is not available, the system will show a notification, and the agent may opt to select another product or wait for the product to be restocked.  - If the agent experiences any challenges while placing the order, such as technical problems, they can reach out to customer support for help. |
| Result | Order successfully |

Use case for **Payment**

|  |  |
| --- | --- |
| Use Case name | Payment |
| Context | An payment method by an agency on the website of the company |
| Selection | Select “Payment” -> Choose payment method |
| Primary Actor | Agency |
| Description | This use case is used for payment |
| Preconditions | - Agency can access to the system via Login  - The agency's order has been successfully submitted. |
| Postconditions | - The payment is handled, and the status of the order is revised in the system.  - The agent can monitor the updated status of their order within the system. |
| Related | Login |
| Stakeholders | Accounting |
| Basic Flow | - The agent chooses the "Payment" option from the menu.  - They select their preferred payment method, which could be cash, bank transfer, or digital payment platforms such as Momo or ZaloPay.  - If the agent decides to pay online, they enter their payment information and verify the transaction.  - Once the payment is processed, the system changes the status of the order to "paid".  - The agent can check the status of their order within the system.  - System display form paid and update databse. |
| Alternative Flows | In the event of payment problems like a failed transaction or technical difficulties, the agent may retry the payment or opt for an alternative payment method. |
| Result | Payment successfully |

## 3.3 Functional requirements

* The system must allow the user to input sales information.
* Sales reports must be produced on a daily basis, every 24 hours.
* The service system must function well and align with the user's requirements.
* Every screen must display details of the operations carried out.
* The system should incorporate a logical approach to handling data.
* It must provide descriptions of system reports or other types of outputs.
* The functional document should include comprehensive information about the workflows executed by the system.
* The functional document should clearly state which users are authorized to create, modify, or delete data within the system.
* The functional document should outline how the system will comply with relevant regulatory and compliance requirements.

**Goods Received:**

- Accountants must be able to create Goods Received in the system when the distributor imports goods, which may include multiple items.

- The system should have provisions for creating Goods Received via both Win Form and Web Form.

- The system should enable accountants to input the specifics of each item received, such as its name, quantity, and cost.

+ Save information of goods: name, date of import, ...

+ Edit, update, delete types of items if input went wrong.

+ Imported item using Function Searching.

**Delivery bill** and **Order:**

- Accountants should be able to use the system to create Goods Delivery Notes with a printed delivery slip to deliver goods to agents.

- The system should allow for the creation of Delivery bills using both Win Form and Web Form.

- The system should enable accountants to update the order status as "transferred" and also update the payment status of agents.

+ Information of export is saved.

+ Choosing method add, delete, edit the orders.

+ Print out the export slip.

+ Updating status Orders like: delivery on the way, payment method (*cash, banking or Momo*), location

+ Can search which goods has been exported, or only search to find them: show the all or one by one.

**Payment management:**

- The system should automatically calculate and verify the balance after receiving payment, save it automatically, and provide notifications.

- The system should facilitate the management and analysis of goods and revenue for each month, which includes the following features:

+ Every item sold will be recorded and calculated by the system. The system should be capable of tallying the quantity sold for each type of item or all items sold in a month.

+ The system should calculate the total revenue for each month and should be able to calculate the revenue for each type of item.

+ It should be possible to make changes to the quantity of items sold, such as adding, deleting, or modifying them if necessary.

## 3.4 Non – functional requirements

- Good service system, capable of working well 24/24 hours and 7 days a week.

- Make sure the system is ready to be used: Time can be used by the system and the dependent elements to operate the system.

- There is periodic maintenance with each system upgrade that will not last more than 30 minutes.

- The system must ask for confirmation when there are data deletion operations.

- The product must comply with the provisions of law, state, company, financial law, ...

- The scalability of the system:

+ Server has the ability to upgrade configuration.

+ It is possible to separate the database on a separate server and backend on a separate server.

- The features of import, export, payment, statistics of goods, revenue of each month must be fast and accurate, ensuring security. Avoid the case of leaked information affecting the company.

# CHAPTER 4 - ARCHITECTURE

## 4.1 Architectural style used

**Architectural model use: Layered Architecture**

- The layered architecture pattern bears a strong resemblance to the conventional IT framework.

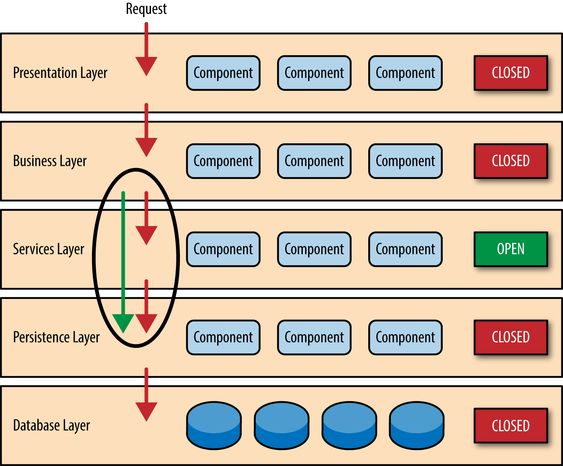
- The architecture defines multiple layers, including an outer and inner layer.

The components of the outer layer handle user interface operations (Presentation Layer).

- The components of the inner layer execute operating system interactions (such as the Accounting Screen, Agency Screen, and Statistics Screen).

- The inner layers comprise the application layer, utility layer, and core layer (including Payment, Delivery bill, Goods Received, and Database).

- In some situations, it may be possible to use multiple patterns, and alternative architectural styles can be devised and assessed.



*[Software Architecture Patterns](https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html" \t "_blank)*

## 4.2 Architectural model

- The **Presentation Layer** is responsible for creating a user-friendly web interface that allows authorized resellers and agents to perform tasks such as placing orders, making payments, and checking the status of their orders. This layer is developed using ASP.NET and various web technologies.

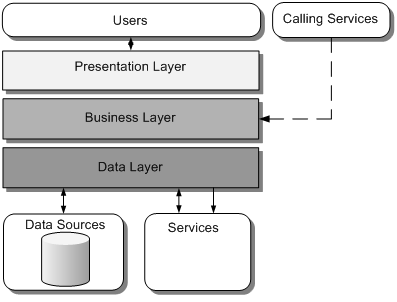
- The **Business Logic Layer** is responsible for implementing the system's core business rules and processes, including order processing, inventory management, and reporting. It facilitates the communication between the Presentation Layer and the Data Access Layer, and is developed using C# and .NET technologies.

- The **Data Access Layer** is responsible for managing data storage and retrieval in the system. It manages the interactions between the Business Logic Layer and the Microsoft SQL Server database using Entity Framework and ADO.NET technologies.

- The **Database** is responsible for storing all the system data, including product information, inventory, orders, and payments. It is built using Microsoft SQL Server.

The system architecture model adopts a standard n-tier design approach, where each layer has its unique 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, on the other hand, manages communication with the database.

By promoting separation of concerns, this architecture model enhances the system's scalability, maintainability, and testability, while also making it easier to manage and modify each layer independently.

****

*Basic Layers of App Architecture*

## 4.3 Technology, software, and hardware used

**Technology**

- Microsoft SQL Server for database management.

- C# for backend development.

- ASP.NET for Web development.

- Entity Framework for data access.

**Software**

-Microsoft SQL Server Management Studio 19.

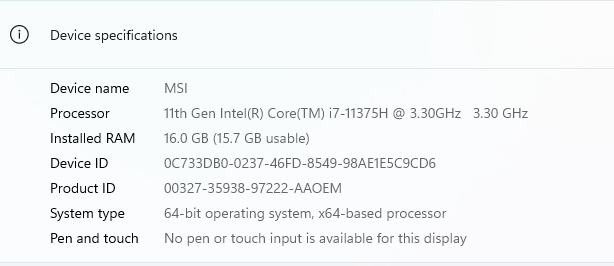
-Visual Studio 2020.

-Visual Studio Code.

- Github

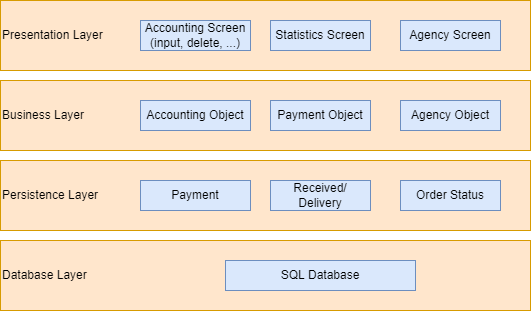
- PayPal, Momo, or other payment gateways for online payments.

**Hardware**





## 4.4 Rationale architectural style and model



- The Layer Architecture simplifies things by requiring us to only comprehend the functions of the classes that are lower in the hierarchy than the current class.

- It's possible to substitute each class with a similar one without causing any disruptions to other classes. A class may be utilized by multiple superior classes.

**Following to the model above:**

+ The Presentation Layer enables users to interact with the interface and perform fundamental functions, such as creating, deleting, modifying, and ordering items.

+ The Business Layer identifies the authorized users who can attempt to access and operate the system (e.g., Accounting Object, Agents Object, Payment Object).

+ The Persistence Layer serves as a mediator between the system functions and the stored database, handling processes such as Payment, Order Status, and Received/Delivery.

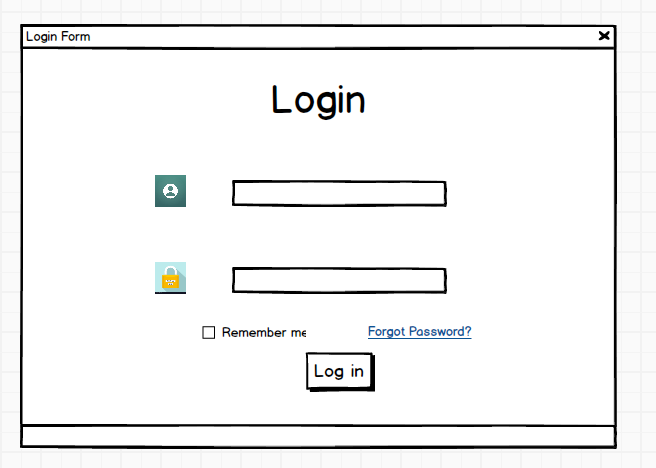
+ The Database Layer is responsible for storing data (using SQL Database).

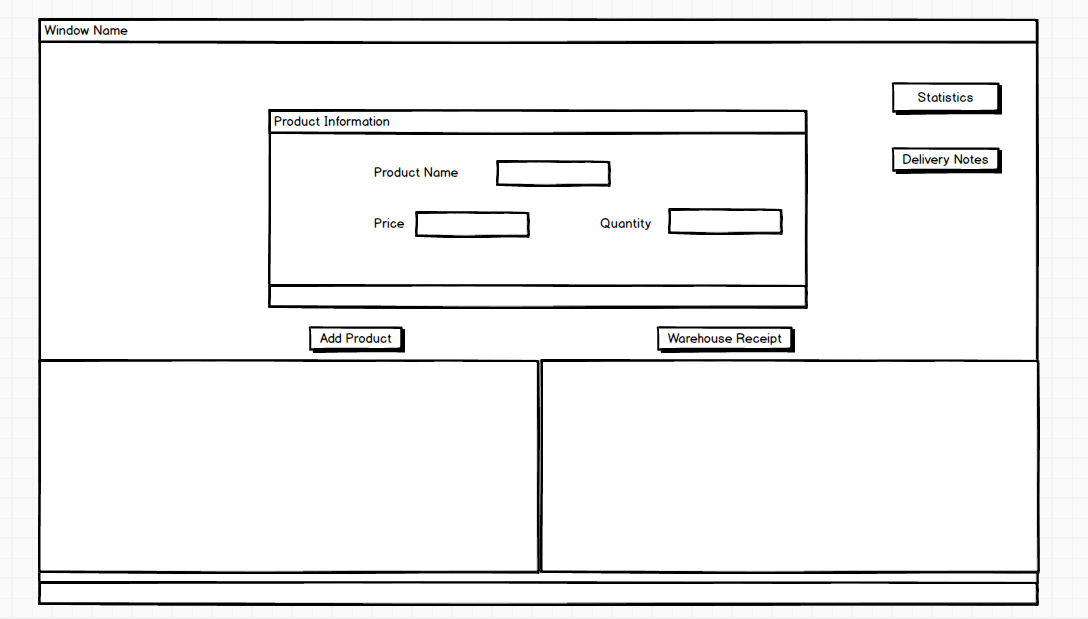
**Testability:** Due the components are assigned to particular layers in the architecture, it's possible to create mock versions of other layers. This means that testing this pattern is relatively straightforward. For instance, a developer can create a mock version of a presentation component or screen to confine testing to a business component. Similarly, it's possible to create a mock version of the business layer to assess specific screen functionality.

**Ease of development:** The pattern is highly valued for its ease of development, largely because it's widely recognized and not overly intricate to put into practice. Given that most companies separate skill sets by layer (such as presentation, business, and database), this pattern is a logical choice for most business-application development.

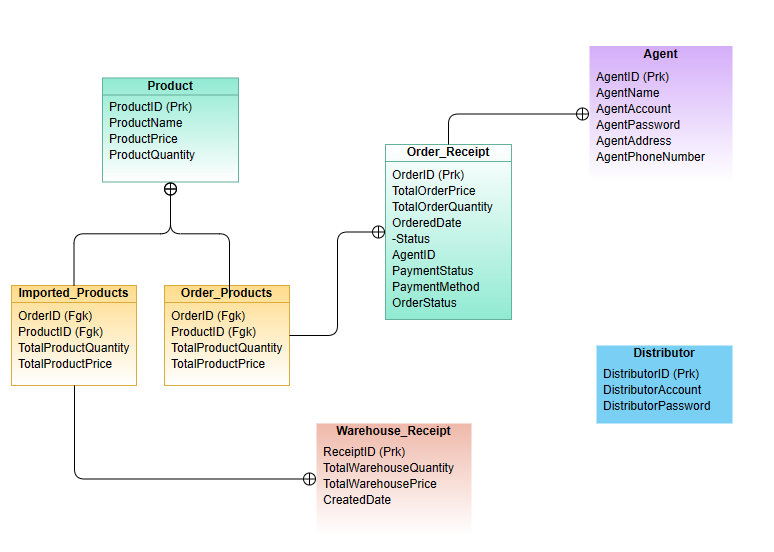
# CHAPTER 5 – DESIGN

**Design with Balsamiq:**

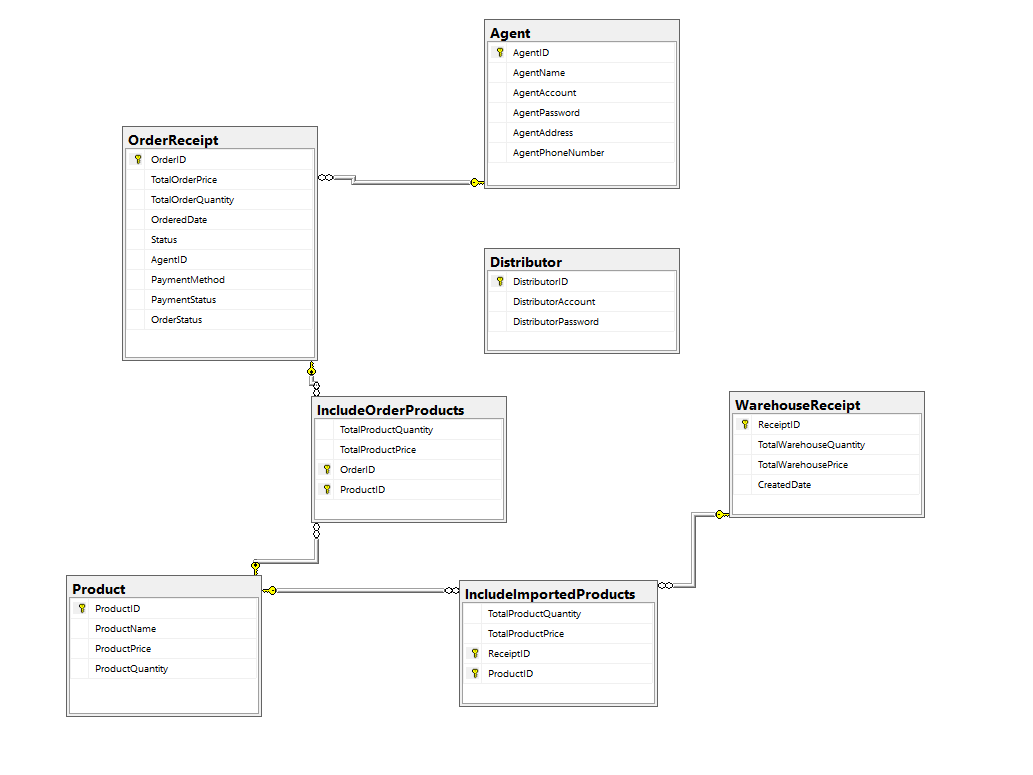




## 5.1 Database design

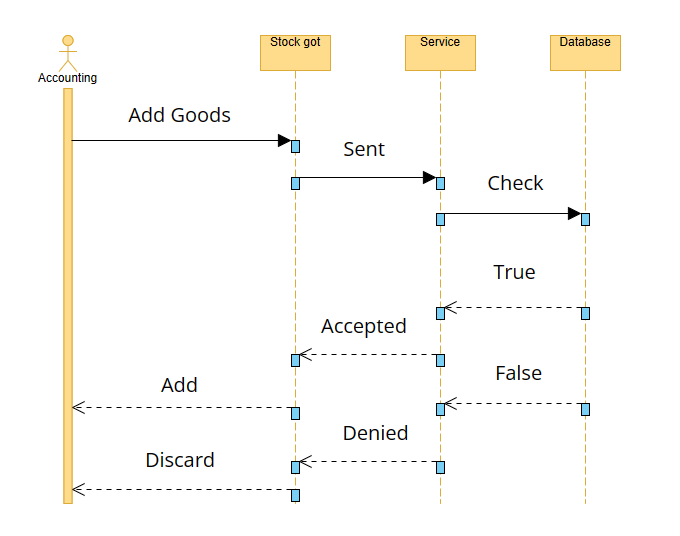


## 5.2 Class Diagrams

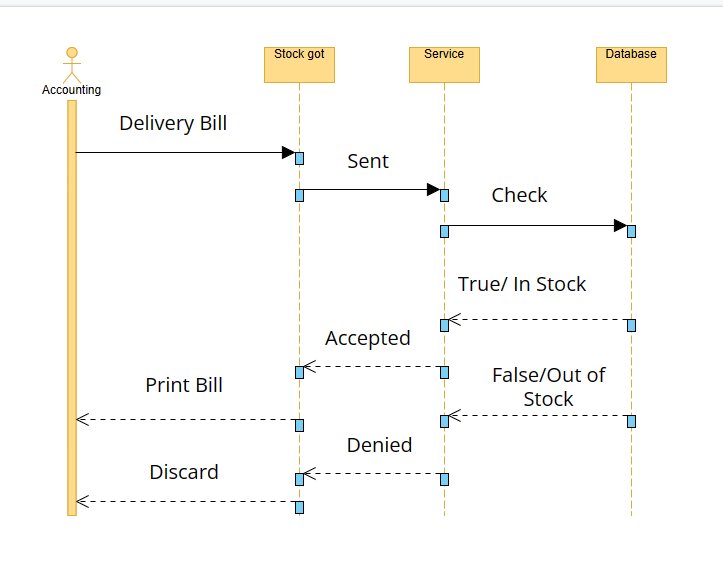


## 5.3 Sequence diagrams

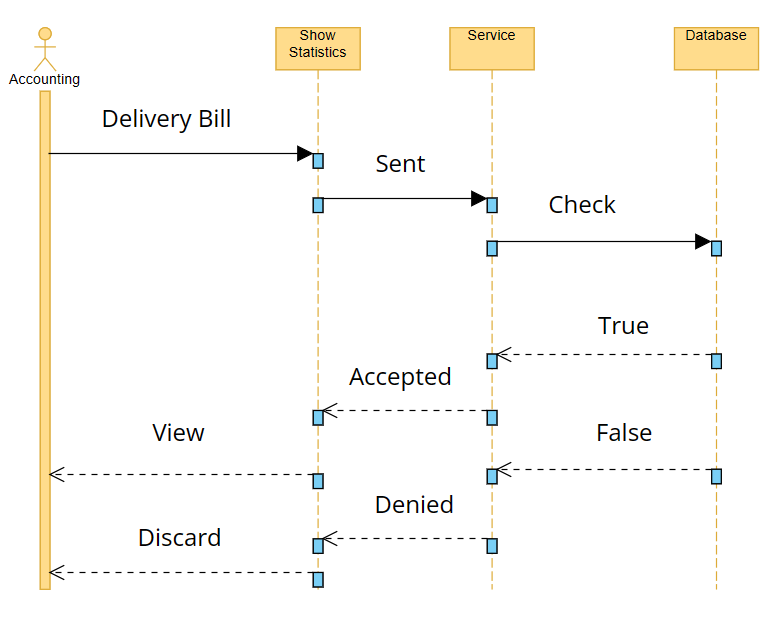
**Goods Received**

****

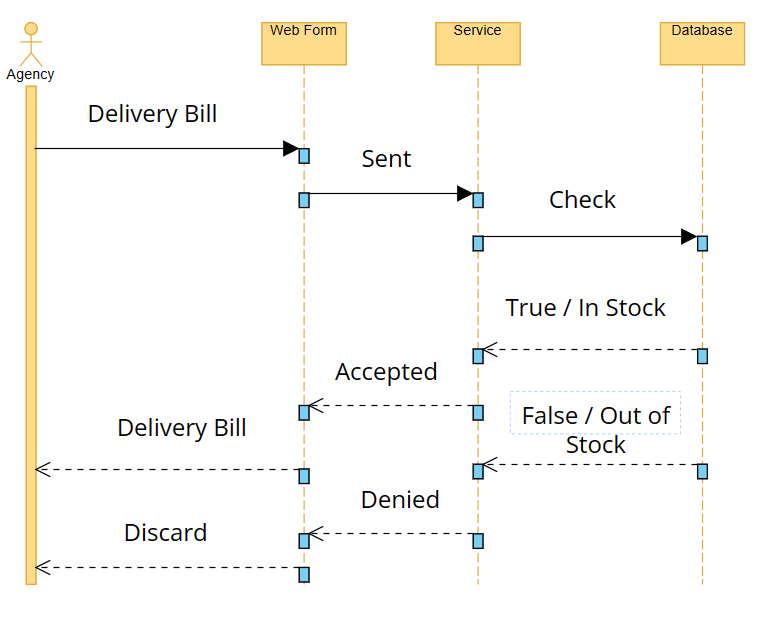
**Delivery Bill**

****

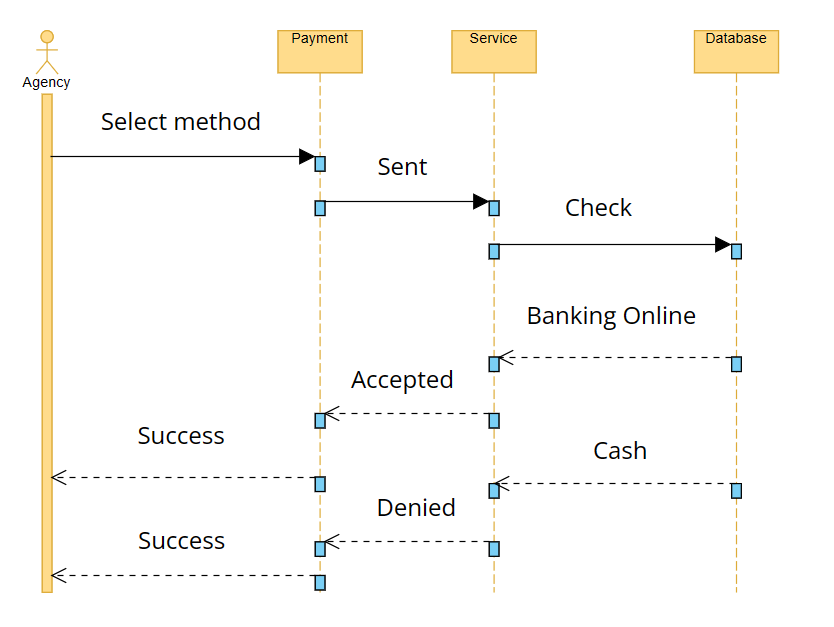
**Statistics management**

****

**Order**

****

**Payment**

****

## 5.4 Rationale for your detailed design model

*Database MobilePhone*

- WarehouseReceipt (ReceiptID, TotalWarehouseQuantity, TotalWarehousePrice, CreatedDate)

- Product (ProductID, ProductName, ProductPrice, ProductQuantity)

- Agent (AgentID, AgentName, AgentAccount, AgentPassword, AgentAddress, AgentPhoneNumber)

- OrderReceipt (OrderID, TotalOrderPrice, TotalOrderQuantity, OrderedDate, Status, AgentID, PaymentMethod, PaymentStatus, OrderStatus). AgentID have references to the Agent table with Foreign Key.

- IncludeOrderProducts (TotalProductQuantity, TotalProductPrice, OrderID, ProductID). Set OrderID\_ProductID to Primary Key and include the Foreign Key OrderID table OrderReceipt and ProductID table Product.

- IncludeImportedProducts (TotalProductQuantity, TotalProductPrice, ReceiptID, ProductID). Table from WarehouseReceipt and Product have a Foreign Key ReceiptID and ProductID

- Distributor (DistributorID, DistributorAccount, DistributorPassword)

- Insert information of Distributor, IncludeOrderProducts, Agent, Product, WarehouseReceipt and IncludeImportedProducts.

- Create Trigger trIncludeOrderProductsUpdateForProduct

- Calculate the value of OrderReceipt then insert it.

# CHAPTER 6 – TEST PLAN

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

**Test Case for Goods Received:**

- Important: Accountants must be able to generate Goods Received records whenever the distributor imports goods.

- Test Case:

- The accountant logs into the system and goes to the Goods Received section.

-The accountant generates a new Goods Received document, filling in all the necessary details such as the date, supplier, and items received.

- The accountant saves the document and confirms that it is saved in the system's database.

- Expected Outcome: The Goods Received document should be successfully created and stored in the system's database.

**Test Scenario for Order:**

- Important: Agents must be able to place an order for items and select a payment method.

Test Case:

- The agent logs into the system and goes to the Order section.

- The agent chooses the items they want to order, adds them to the cart, and selects their preferred payment method.

- The agent completes the order and confirms that the order is saved in the system's database and their payment is processed.

- Expected Outcome: The order should be successfully created and stored in the system's database, and the payment should be successfully processed.

**Test Case for Delivery Bill:**

- Important: Accountants must be able to generate a Goods Delivery Note to deliver goods to agents.

Test Case:

- The accountant logs into the system and goes to the Delivery Note section.

- The accountant creates a new Delivery Note document and enters all the necessary details such as the date, agent, and items being delivered.

- The accountant saves the document and confirms that it is saved in the system's database and the agent's payment status is updated.

- Expected Outcome: The Delivery Note document should be successfully created and stored in the system's database, and the agent's payment status should be updated.

**Test Case for Statistical:**

- Important: Accountants must be able to access monthly reports on incoming/outgoing stock, best-selling products, and revenue.

- Test Case:

- The accountant logs into the system and goes to the Statistical section.

- The accountant chooses the desired report type and specifies the date range.

- The accountant views the report and confirms that the information is precise and current.

- Expected Outcome: The statistical report should be successfully generated and displayed accurately.

## 6.2 Traceability of test cases to use cases

Add, edit, delete: items, accountants, receipt details, customers, delivery bill, orders and order details

## 6.3 Techniques used for test generation

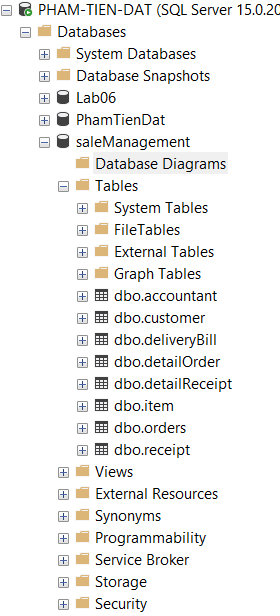
Various methods can be employed to generate tests, such as equivalence partitioning, boundary value analysis, decision table testing, state transition testing, error guessing, exploratory testing, and model-based testing. The selection of a particular technique depends on factors such as the system's characteristics, the level of testing required, and the resources and proficiency available.

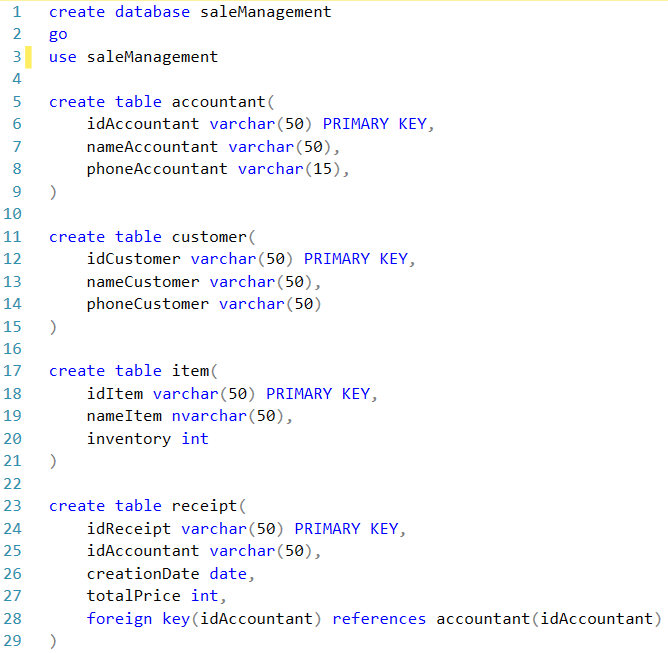
## 6.4 Assessment of the goodness of our testsuite

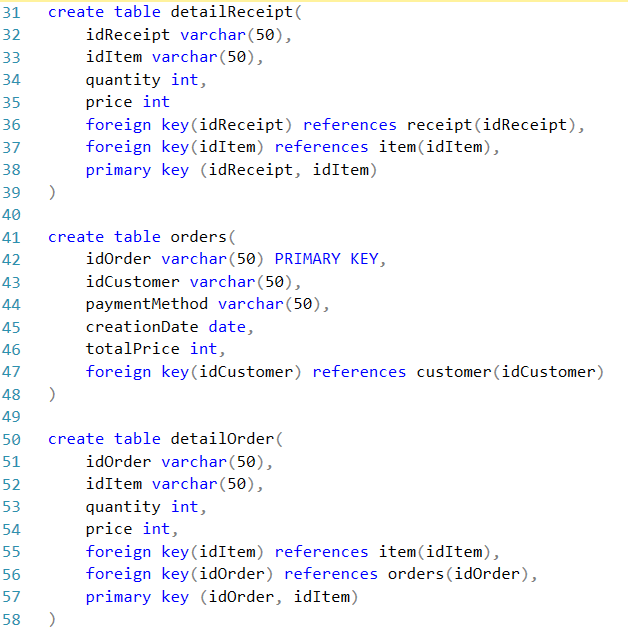
High-level testing is appropriate for evaluating significant sections of code and sizable functions. This type of testing does not necessitate comprehension of the code used in the program or knowledge of the structure within the function and how to access it. High-level testing is useful for scrutinizing large code segments and substantial functions.

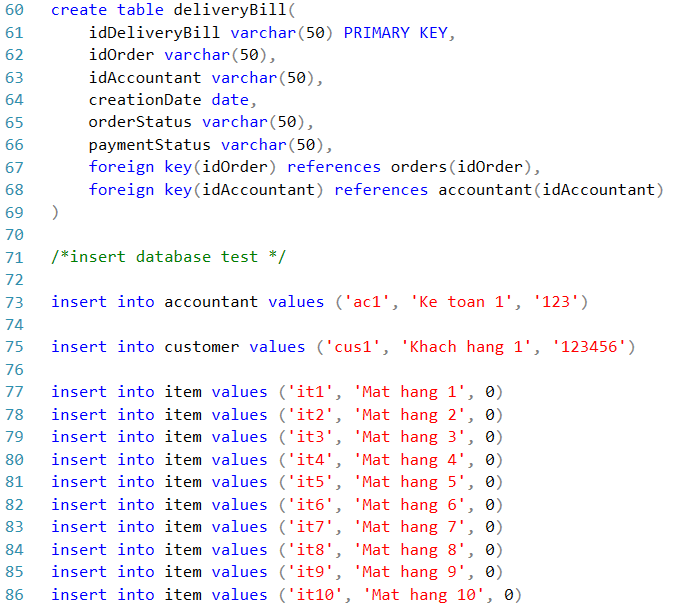
# CHAPTER 7 – DEMO

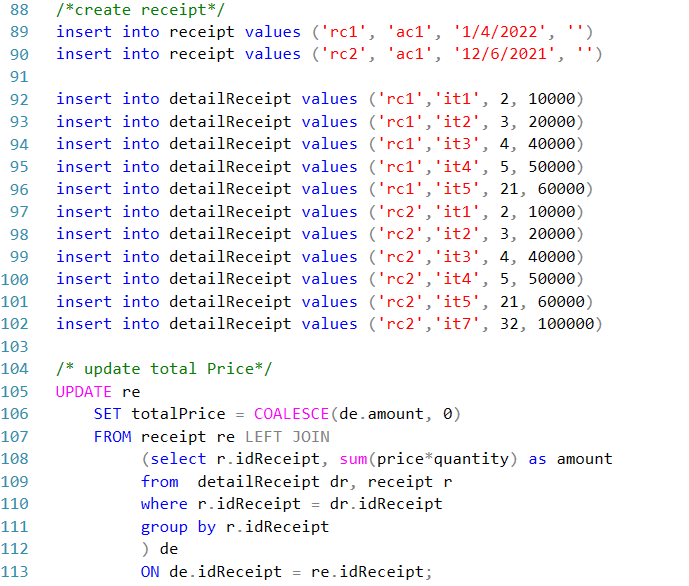
## 7.1 Database

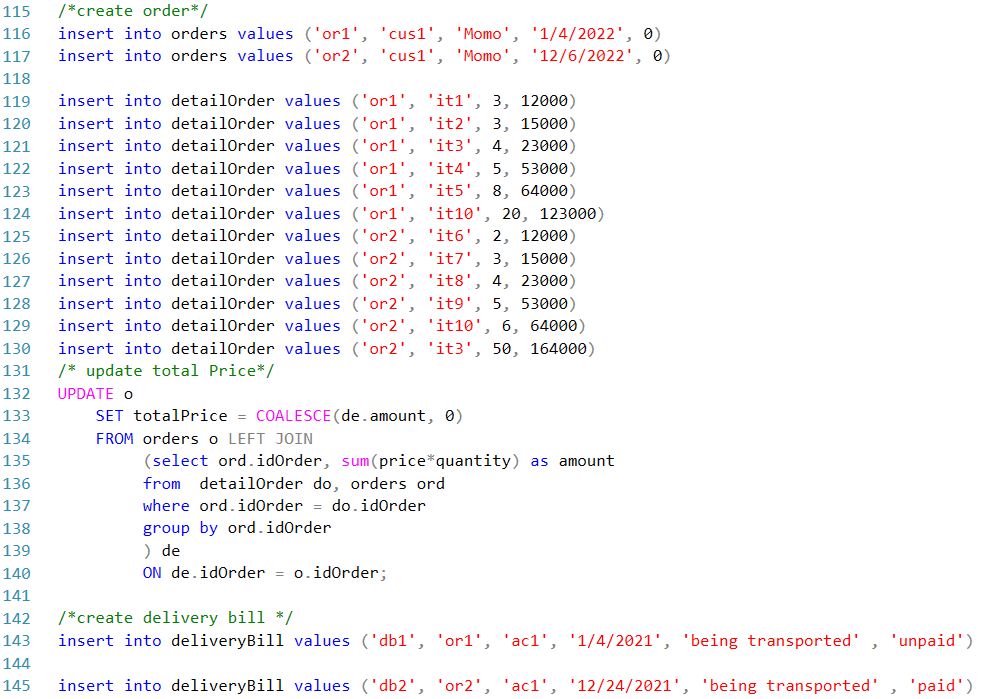
****

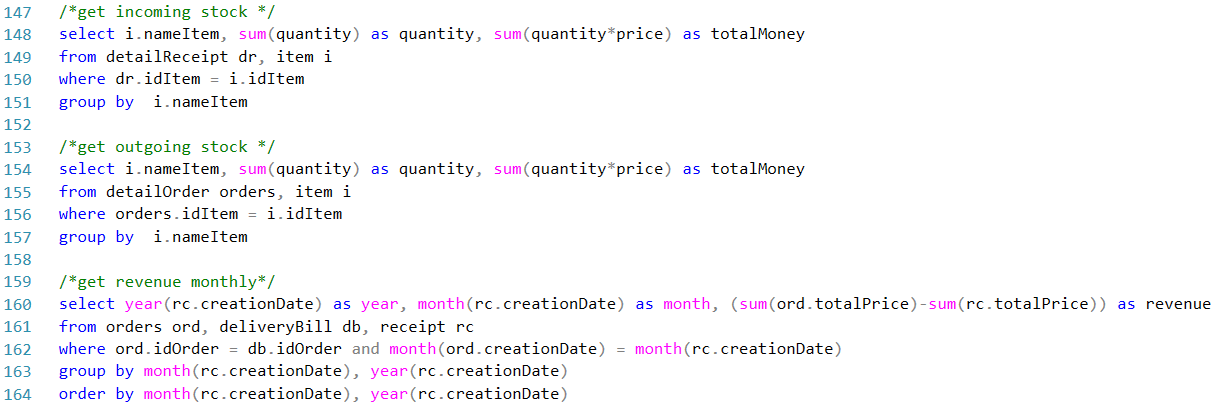
****

****

****

****

****

****

## 7.2 Source Code