VIETNAM LABOR UNION GENERAL

**TON DUC THANG UNIVERSITY**

**FALCUTY OF INFORMATION TECHNOLOGY**



**SOFTWARE ENGINEERING FINAL PROJECT**

**MANAGEMENT SOFTWARE SELLING MOBILE PHONE PRODUCT**

*Supervisor:* **MR. PHAM THAI KY TRUNG**

*Authors*: **NGUYEN LE MINH TAN\_521H0150**

**HUYNH AN NGUYEN\_521H0467**

*Class***: 21H50201**

*Course***: 25**

**HO CHI MINH CITY, 2023**

VIETNAM LABOR UNION GENERAL

**TON DUC THANG UNIVERSITY**

**FALCUTY OF INFORMATION TECHNOLOGY**



**SOFTWARE ENGINEERING FINAL PROJECT**

**MANAGEMENT SOFTWARE SELLING MOBILE PHONE PRODUCT**

*Supervisor:* **MR. PHAM THAI KY TRUNG**

*Authors*: **NGUYEN LE MINH TAN\_521H0150**

**HUYNH AN NGUYEN\_521H0467**

*Class***: 21H50201**

*Course***: 25**

**HO CHI MINH CITY, 2023**

LETTER OF APPRECIATION

Dear Ton Duc Thang University and Mr. Pham Thai Ky Trung,

I wanted to express my heartfelt gratitude for the incredible opportunity to study and learn from your institution and your team. The knowledge and skills I have gained during my time at Ton Duc Thang University will undoubtedly shape my future in meaningful ways.

Your dedication to providing exceptional education and resources to your students is truly inspiring, and I feel honored to have been a part of your community. Thank you for your unwavering support and encouragement throughout my studies, and for helping me to achieve my academic goals.

I would also like to extend a special thank you to Mr. Pham Thai Ky Trung for his guidance and mentorship. His expertise and passion for teaching have been invaluable to my growth and development, and I am grateful for his contributions to my education.

Once again, thank you for everything. I will cherish the experiences and knowledge gained at Ton Duc Thang University for years to come.

Sincerely,

Minh Tan

An Nguyen

**THE PROJECT ARE ACCOMPLISHED**

**AT TON DUC THANG UNIVERSITY.**

To whom it may concern,

We hereby confirm that the project we have submitted is our own work, developed under the guidance of Mr. Pham Thai Ky Trung. The research content and results presented in this project are accurate and have not been published previously in any form. The data used in the analysis, comments, and evaluations were collected from various sources, and all sources have been clearly cited in the references.

Furthermore, this project also includes comments, assessments, and data from other authors, organizations, and institutions, which have been properly cited and attributed.

We understand that any fraudulent activity detected in our project will result in full responsibility on our part. Ton Duc Thang University shall not be held liable for any copyright or intellectual property infringement arising from the project's development and implementation.

Thank you for your time and consideration.

Sincerely,

Minh Tan  
 An Nguyen

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

*Author*

*(Sign and provide full name)*

*Nguyen Le Minh Tan*

*Huynh An Nguyen*

**VERIFICATION AND EVALUATION FROM LECTURER**

**Instructor endorsement**

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

Ho Chi Minh city,

(Sign and provide full name)

TABLE OF CONTENTS

[LETTER OF APPRECIATION i](#_Toc387692905)

[**LIST OF FIGURES 6**](#_Toc92334172)

[**LIST OF TABLES 7**](#_Toc92334173)

[**CHAPTER 1 – INTRODUCTION 8**](#_Toc92334174)

[**1.1 Purpose and Scope 8**](#_Toc92334175)

[**1.2 Product Overview 8**](#_Toc92334176)

[**1.3 Structure of the Document 9**](#_Toc92334177)

[**1.4 Terms, Acronyms, and Abbreviations 9**](#_Toc92334178)

[**CHAPTER 2 - PROJECT MANAGEMENT PLAN 9**](#_Toc92334179)

[**2.1 Project Organization Structure 9**](#_Toc92334180)

[**2.2 Lifecycle Model Used 9**](#_Toc92334181)

[**2.3 Risk Analysis 10**](#_Toc92334182)

[**2.4 Hardware and Software Resource Requirements 10**](#_Toc92334183)

[**2.5 Deliverbles and Schedule 11**](#_Toc92334184)

[**2.6 Monitoring, Reporting, and Controlling Mechanisms 12**](#_Toc92334185)

[**Monitoring and Controlling 12**](#_Toc92334186)

[**2.7 Professional Standards 13**](#_Toc92334187)

[**2.8 Evidence all the artifacts have been placed under configuration management 13**](#_Toc92334188)

[**2.9 Impact of the project on individuals and organizations 13**](#_Toc92334189)

[**CHAPTER 3 - REQUIREMENT SPECIFICATIONS 13**](#_Toc92334190)

[**3.1 Stakeholders for the system 13**](#_Toc92334191)

[**3.2 Use case model 14**](#_Toc92334192)

[***3.2.1 Graphical use case model* 14**](#_Toc92334193)

[***3.2.2 Use case specification* 15**](#_Toc92334194)

[**3.3 Functional requirements 20**](#_Toc92334195)

[**3.4 Non – functional requirements 21**](#_Toc92334196)

[**CHAPTER 4 - ARCHITECTURE 22**](#_Toc92334197)

[**4.1 Architectural style used 22**](#_Toc92334198)

[**4.2 Architectural model 23**](#_Toc92334199)

[**4.3 Technology, software, and hardware used 24**](#_Toc92334200)

[**4.4 Rationale architectural style and model 25**](#_Toc92334201)

[**CHAPTER 5 – DESIGN 27**](#_Toc92334202)

[**5.1 Database design 27**](#_Toc92334203)

[**5.2 Class Diagrams 28**](#_Toc92334204)

[**5.3 Sequence diagrams 28**](#_Toc92334205)

[**5.4 Rationale for your detailed designn model 32**](#_Toc92334206)

[**5.5 Traceability from requirements to detailed design model 33**](#_Toc92334207)

[**CHAPTER 6 – TEST PLAN 33**](#_Toc92334208)

[**6.1 Requirements/specifications-based system level test cases 33**](#_Toc92334209)

[**6.2 Traceability of test cases to use cases 34**](#_Toc92334210)

[**6.3 Techniques used for test generation 34**](#_Toc92334211)

[**6.4 Assessment of the goodness of our testsuite 35**](#_Toc92334212)

[**CHAPTER 7 – DEMO 36**](#_Toc92334213)

[**7.1 Database 36**](#_Toc92334214)

[**7.2 Source Code 41**](#_Toc92334215)

[**7.3 Testing 43**](#_Toc92334216)

**LIST OF FIGURES**

**1.Goods Received…………………………………………………………………..29**

**2.Delivery Bill………………………………………………………………………29**

**3.Statistics Management…………………………………………………………..30**

**4.Order……………………………………………………………………………...31**

**5.Payment…………………………………………………………………………..31**

**LIST OF TABLES**

**1.Use case for Goods Received……………………………………………………..29**

**2.Use case for Delivery Bill…………………………………………………………29**

**3.Use case for Statistics Management…………………………………………….30**

**4.Use case for Order………………………………………………………………31**

**5.Use case for Payment……………………………………………………………31**

**CHAPTER 1 – INTRODUCTION**

**1.1 Purpose and Scope**

\* Purposes:

- Streamline the sales process: By using a software system specifically designed for phone sales,we can streamline the sales process and make it more efficient. This can save time and improve productivity, which can ultimately lead to increased sales.

- Improve inventory management: A phone selling software can help us keep track of inventory levels and prevent overstocking or understocking. This can save us money and help us avoid stockouts that can lead to lost sales.

- Provide better analytics and reporting: By tracking sales data, a phone selling software can provide valuable insights into our business, such as which products are selling well and which ones are not. This can help us make data-driven decisions and adjust our sales strategy accordingly.

\*\* Scope:

- Sales management: The software can provide tools to manage and track phone sales, including lead management, call tracking, appointment scheduling, and order processing.

- Inventory management: The software can help manage inventory levels, including stock management, product catalog, and order fulfillment.

- Customization and scalability: The software can be customized to meet specific business needs and can scale as the business grows.

**1.2 Product Overview**

- Staffs are able to create products, categories of the products and manage the sellers. They can also view the sell lists history of the sellers. They can edit or delete the products.

- Agents can print the bill, place orders and view products

**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**

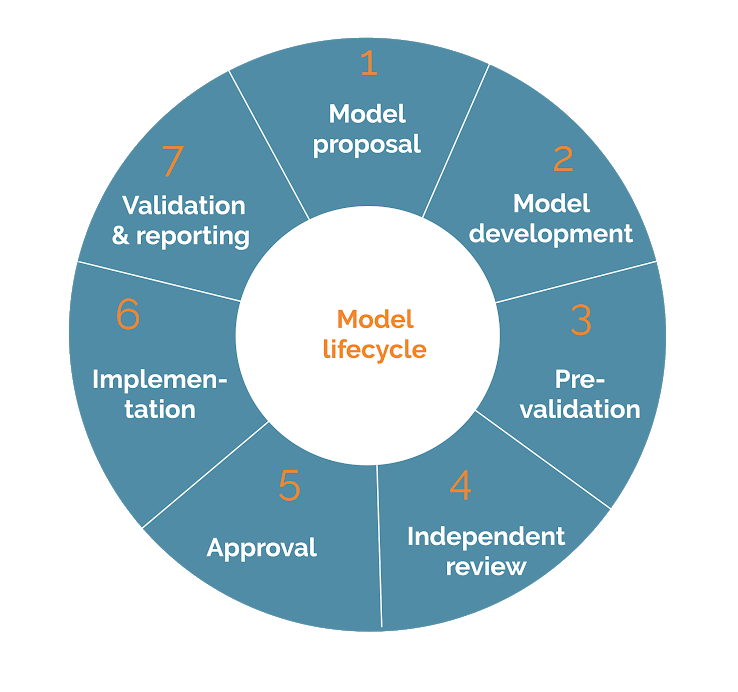
Project Organization**:**

- Software Engineer and create database: Minh Tan and An Nguyen.

- Minh Tan: create SellingForm, SellerForm, Splash and Form1, Database sql

- An Nguyen: create CategoryForm, ProductForm, edit Databasesql

**2.2 Lifecycle Model Used**

****

**2.3 Risk Analysis:**

- Security Breach.

- Technical Issues.

- Competitor Releases Similar Software.

- Data Loss or Corruption.

**2.4 Hardware and Software Resource Requirements**

\*\* Hardware Requirements:

* Processor: Intel Core i5 or higher
* RAM: 8GB or higher
* Storage: 256GB SSD or higher
* Display: 1920x1080 resolution or higher
* Graphics: Dedicated graphics card with 2GB or more VRAM

\*\* Software Requirements:

* Operating System: Windows 10 or Mac OS X 10.14 or later
* Development Environment: Microsoft Visual Studio or JetBrains Rider
* Programming Language: C# or Java
* Web Server: Apache or Microsoft IIS
* Database: MySQL or Microsoft SQL Server
* Source Control: Git or SVN

**2.5** Deliverables and Schedule:

\*\* Deliverables:

1. Project Plan and Requirements Specification Document
2. User Interface Design and Wireframes
3. Database Schema Design and Implementation
4. Back-End and Front-End Development
5. Testing and Debugging
6. User Acceptance Testing and Deployment
7. User Manual and Technical Documentation

\*\* Schedule:

1. Week 1-2: Project Planning and Requirements Gathering
2. Week 3-4: User Interface Design and Wireframes
3. Week 5-6: Database Schema Design and Implementation
4. Week 7-10: Back-End and Front-End Development
5. Week 11-13: User Manual and Technical Documentation

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

1. Monitoring:

* Regularly checking server uptime and response time
* Tracking website traffic to monitor user behavior, popular products, and sales trends.
* Monitoring database usage and performance to ensure smooth and efficient operation.

1. Reporting:

* Generating weekly or monthly sales reports to track revenue, best-selling products, and customer demographics.
* Generating error reports to identify and address any technical issues or bugs.
* Generating inventory reports to track product availability and manage restocking.

1. Controlling:

* Implementing access control mechanisms to restrict sensitive data access to authorized personnel only.
* Implementing authentication and authorization mechanisms to ensure secure login and prevent unauthorized access.
* Implementing backup and disaster recovery mechanisms to ensure data safety and continuity in case of any data loss or system failures.

**2.7 Professional Standards:**

- Security: The software should comply with industry standards for security.

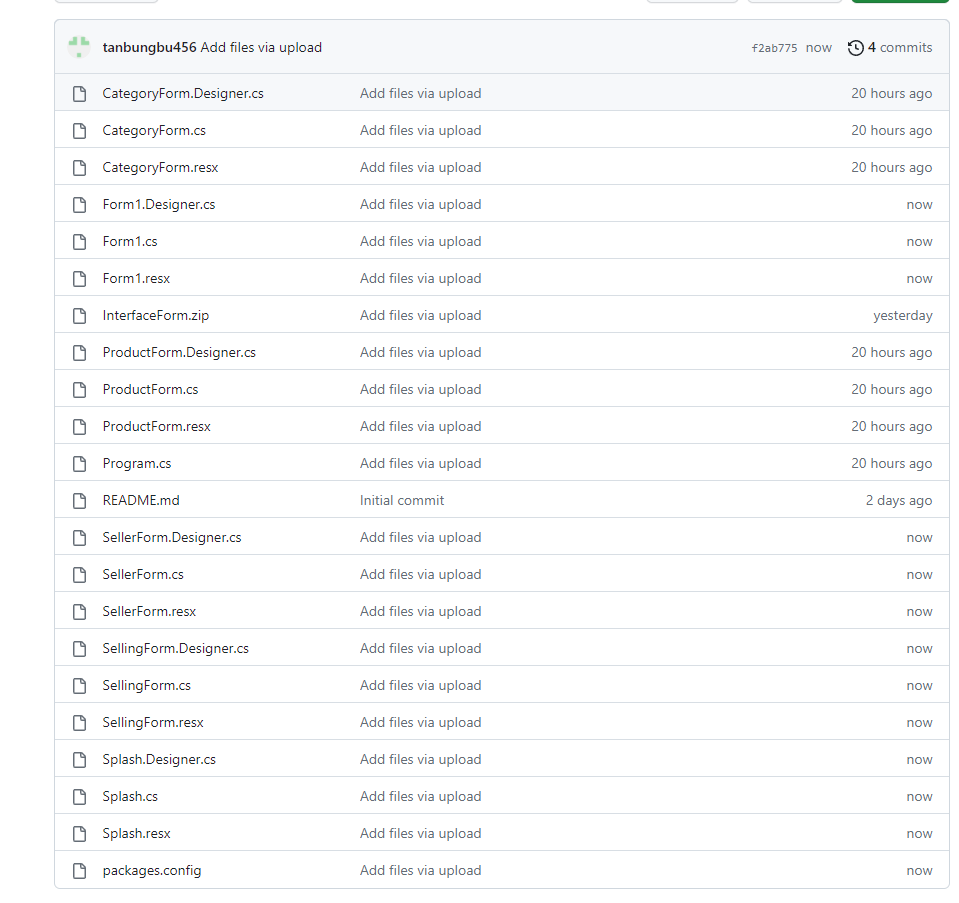
- Accessibility: The software should be designed to be accessible to users with disabilities, with features such as keyboard navigation and screen reader compatibility.

- Maintenance: The software should be designed with ease of maintenance and updating in mind, with clear documentation and version control mechanisms in place.

- Performance: The software should be optimized for efficient and speedy operation, with minimal delays or downtime.

- Usability: The software should be designed to be user-friendly and intuitive, with a clear and consistent interface that is easy to navigate and understand.

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

****

* 1. **Impact of the project on individuals and organizations**
* Increased sales: The software could help increase sales for the organization by providing a more efficient way to manage orders and inventory, resulting in more satisfied customers.
* Improved customer satisfaction: With the phone selling software, customers can easily browse and purchase products online, reducing the need for in-person visits to the store. This convenience could lead to increased customer satisfaction and loyalty.
* Improved inventory management: The software's inventory management capabilities could help organizations better manage their stock levels, reducing the risk of overstocking or stockouts.
* Enhanced data analysis: The software's reporting and analytics capabilities could provide valuable insights into customer behavior and sales trends, helping organizations make informed business decisions.
* Improved productivity: By automating certain tasks and streamlining the sales process, the software could help employees become more productive, freeing up time for other important tasks.
* Increased competition: With the growing trend of online sales, organizations that do not adopt similar software may be left behind in terms of competition.
* Cost savings: The phone selling software may result in cost savings for organizations by reducing the need for manual labor, improving inventory management, and reducing the risk of errors.

**CHAPTER 3 - REQUIREMENT SPECIFICATIONS**

**3.1 Stakeholders for the system**

Sales team: The team responsible for selling the phones and managing customer orders and shipments.

Marketing team: The team responsible for promoting the software and driving traffic to the online store.

Technical team: The team responsible for maintaining and updating the software to ensure it runs smoothly.

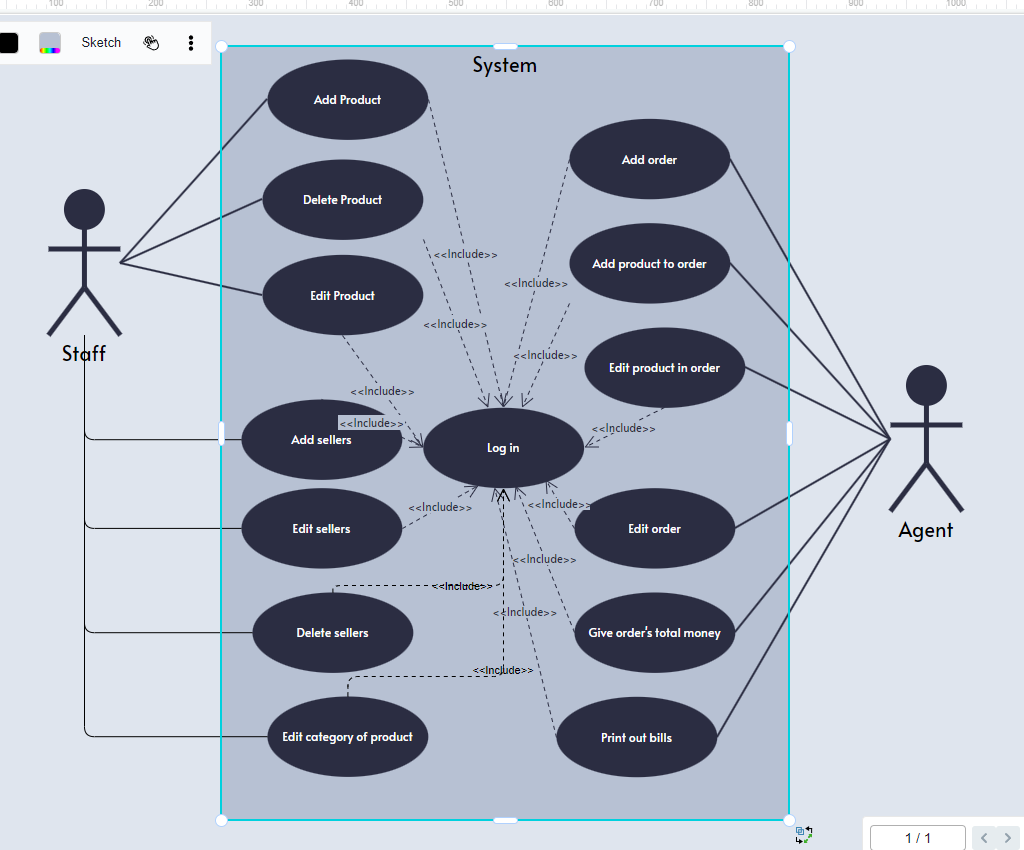
Management team: The team responsible for overseeing the project and ensuring that it aligns with the organization's goals and objectives.

Partners: Suppliers, manufacturers, or other organizations involved in the supply chain of the phones being sold.

Investors: Individuals or organizations who have invested in the project and have a stake in its success.

Regulators: Government agencies or other entities responsible for enforcing rules and regulations that impact the software and its users. 3.2 Use case model

***3.2.1 Graphical use case model***

******

### *3.2.2 Use case specification:*

1. Use Case Name: Sell Phone

Actors: Salesperson, Customer

Description: This use case describes the process of selling a phone to a customer.

Preconditions:

The salesperson must have access to the phone selling system.

The customer must be present and interested in purchasing a phone.

Postconditions:

The customer has purchased a phone.

The phone inventory has been updated to reflect the sale.

Basic Flow:

The salesperson selects the "Sell Phone" option in the phone selling system.

The salesperson selects the phone the customer wants to purchase.

The salesperson checks the phone inventory to ensure the phone is available.

If the phone is available, the salesperson confirms the sale with the customer.

The salesperson enters the customer's information into the phone selling system.

The salesperson completes the transaction and gives the phone to the customer.

Alternate Flows:

If the phone is not available in the inventory, the salesperson informs the customer and recommends a similar phone.

If the customer decides not to purchase the phone, the salesperson cancels the transaction and updates the phone inventory accordingly.

Exceptions:

If the phone selling system crashes during the transaction, the salesperson restarts the system and resumes the transaction.

If the customer's payment method is declined, the salesperson informs the customer and cancels the transaction.

2. Use Case: Update Product Information

Primary Actor: Staff

Goal in Context: The staff member wants to update the information of a product in the system.

Preconditions:

The staff member is logged into the system.

The product to be updated already exists in the system.

Trigger: The staff member selects the "Update Product" option from the system menu.

Main Success Scenario:

The system displays a list of all available products.

The staff member selects the product they want to update.

The system displays the details of the selected product.

The staff member updates the information of the product (e.g., name, price, description, etc.).

The staff member confirms the changes and submits the updated information to the system.

The system validates the updated information and updates the product information in the system database.

The system displays a confirmation message to the staff member that the product information has been successfully updated.

Extensions:

2a. If no products exist in the system, the system displays an empty list message to the staff member.

3a. If the selected product cannot be found in the system, the system displays an error message to the staff member.

5a. If the staff member decides to cancel the update, the system returns to step 3 without making any changes to the product information.

6a. If the updated information is invalid (e.g., the price is negative), the system displays an error message to the staff member and returns to step 4.

Postconditions:

The product information in the system database is updated with the new information provided by the staff member.

3. Use Case Name: Place Order

Primary Actor: Staff

Stakeholders and Interests:

Staff: Wants to efficiently place an order for a customer

Customer: Wants their order to be processed quickly and accurately

Inventory Manager: Wants to ensure that there is enough inventory to fulfill orders

Financial Department: Wants to track all orders and their associated costs

Preconditions:

The staff is logged into the system.

The customer's desired products are in stock.

Postconditions:

An order has been placed and is ready for fulfillment.

Main Flow:

The staff selects "Place Order" from the system menu.

The system prompts the staff to enter the customer's information, including name and contact information.

The staff enters the customer's information and selects "Next".

The system displays a list of available products and prompts the staff to select the desired products.

The staff selects the desired products and enters the quantity for each product.

The system verifies that the selected products are in stock.

If any selected product is out of stock, the system notifies the staff and returns to step 4.

The system calculates the total cost of the order.

The staff confirms the order and selects "Place Order".

The system updates the inventory and financial records, and generates an order confirmation for the customer.

4. Use Case Name: Generate Sales Report

Primary Actor: Agent

Goal in Context: The agent wants to generate a sales report for a specified time period.

Preconditions:

The agent is logged into the phone selling software.

The agent has the necessary permissions to access the sales report feature.

Trigger: The agent clicks on the "Generate Sales Report" button.

Main Flow:

The system displays a form where the agent can specify the time period for which the sales report is to be generated.

The agent fills in the form with the required details and clicks on the "Generate Report" button.

The system retrieves the sales data for the specified time period and generates a sales report.

The system displays the sales report on the screen.

The agent can choose to download the report or print it.

Postconditions:

The sales report is generated and displayed to the agent.

The agent can download or print the report as required.

Alternative Flows:

2a. The agent enters invalid or incomplete information.

The system displays an error message and prompts the agent to enter valid information.

3a. No sales data is available for the specified time period.

The system displays a message indicating that no sales data is available for the specified time period.

4a. The agent chooses to download the report.

The system generates a PDF file of the report and prompts the agent to save it to their device.

4b. The agent chooses to print the report.

The system sends the report to the default printer configured on the agent's device.

## Functional requirements

* The system shall allow customers to browse products by category.
* The system shall allow customers to add products to their shopping cart.
* The system shall allow customers to checkout and make payments.
* The system shall allow staff to add new products to the system.
* The system shall allow staff to view and manage customer orders.
* The system shall provide real-time inventory tracking.
* The system shall allow customers to view their order history.
* The system shall allow customers to search for products by name, description, or keyword.
* The system shall generate confirmation emails to customers after they place an order.
* The system shall generate reports for staff to track sales and inventory.

## Non – functional requirements:

* Performance: The software should be able to handle a large number of users and transactions without significant slowdown or crashes.
* Security: The software should be designed with robust security measures to protect user data and prevent unauthorized access or hacking.
* Usability: The software should be intuitive and user-friendly, with clear navigation and easy-to-understand instructions.
* Reliability: The software should be highly reliable and available, with minimal downtime or system failures.
* Scalability: The software should be designed to accommodate growth and future expansion, with the ability to handle increasing numbers of users and transactions over time.
* Compatibility: The software should be compatible with a wide range of devices and platforms, including different operating systems and web browsers.
* Maintainability: The software should be designed with easy maintenance and updates in mind, with clear documentation and accessible code.

# CHAPTER 4 - ARCHITECTURE

## 4.1 Architectural style used

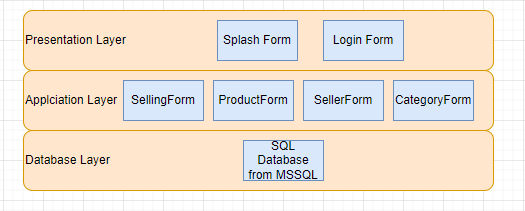
\*\* Layered architecture:

Presentation layer: Splash Form, Login Form.

Application layer: CategoryForm, SellingForm, ProductForm, SellerForm.

Database Layer: SQL Database from MSSQL.

## 4.2 Architectural model

****

## 4.3 Technology, software, and hardware used

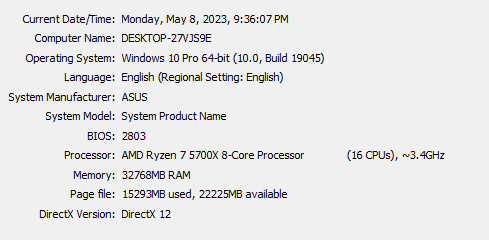
\*\* Technology used:

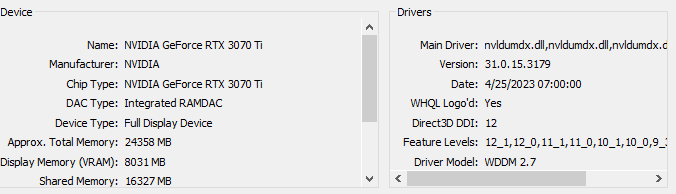
* Sql Server from MSSQL
* C# from Microsoft Visual Studio
* Github

\*\* Software used:

* MSSQL
* Micorsoft Visual Studio 2022

\*\* Hardware:





## 4.4 Rationale architectural style and model

- The layered architecture model is commonly used for developing software systems because it separates concerns and allows for modularity and flexibility. This model consists of several layers, each with its own specific responsibility, and each layer communicates only with the layer directly above or below it.

* In the case of the phone selling software, the layered architecture model can help to separate the concerns of data management, business logic, and presentation, making the system easier to maintain, test, and modify.
* The presentation layer is responsible for handling user input and displaying output to the user. This layer communicates with the business logic layer to retrieve and update data. The business logic layer contains the application's core functionality and rules, and communicates with the data access layer to retrieve and update data from the database.
* The data access layer handles the low-level database operations, such as inserting, updating, deleting, and retrieving data from the database. This layer provides a level of abstraction that isolates the rest of the application from the underlying data storage technology and allows for easier maintenance and evolution.
* The rationale behind using the layered architecture model for the phone selling software is to improve the system's maintainability, scalability, and flexibility. By separating the concerns into layers, the software can be developed and tested more easily, and changes to one layer can be made without affecting the others. Additionally, the model can facilitate the reuse of components and make the system more adaptable to changing requirements.

\*\* Advantages:

* Separation of concerns: Each layer has its own responsibility and focus, making it easier to manage and change specific parts of the system without affecting the others.
* Modularity: The layered architecture promotes modularity and flexibility, as each layer can be developed, tested, and deployed independently of the others.

Scalability: The architecture is easily scalable, as new layers can be added or existing ones can be modified without affecting the rest of the system.

* Ease of maintenance: Because of the separation of concerns and modularity, maintaining and upgrading the system becomes easier and less risky.

Encapsulation: Each layer has its own APIs, which allows for better encapsulation and abstraction.

\*\* Disadvantages:

* Overhead: The layered architecture can introduce additional overhead due to the need for communication and coordination between layers.
* Performance: Because of the communication overhead and potential duplication of functionality across layers, performance can be negatively impacted.

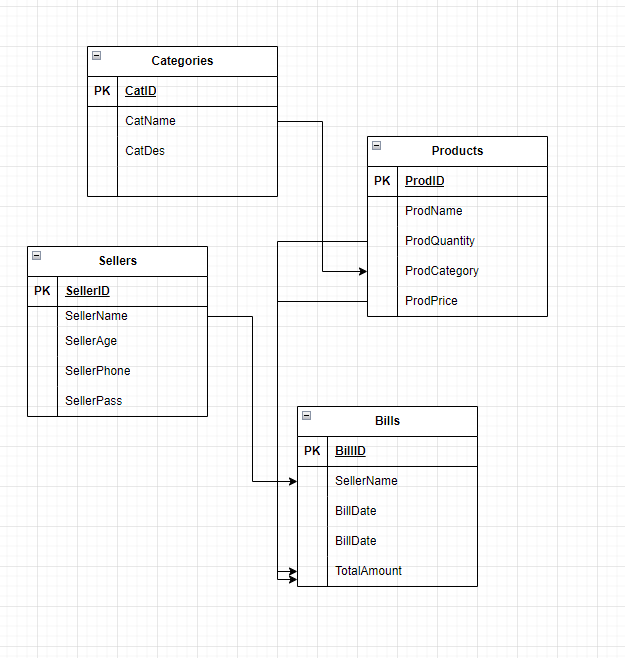
Rigidity: The layered architecture can be inflexible and may not be suitable for systems with complex and changing requirements.

* Increased complexity: The layered architecture can increase the complexity of the system, particularly when the number of layers increases.

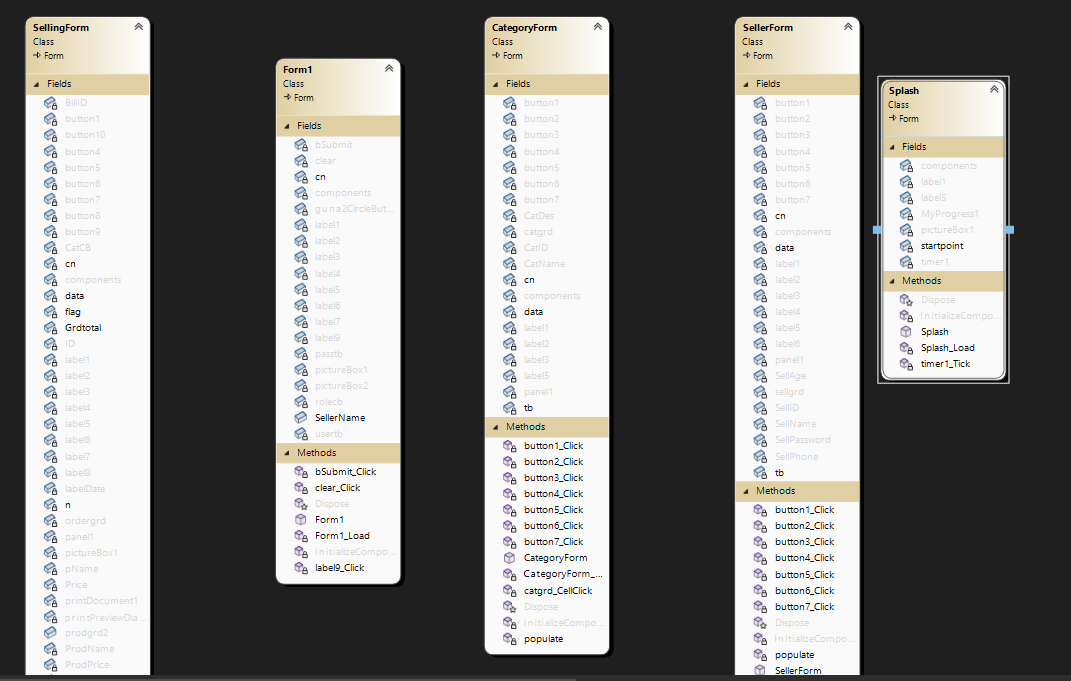
Duplication of functionality: It is possible for functionality to be duplicated across layers, leading to unnecessary complexity and maintenance overhead.

# CHAPTER 5 – DESIGN

## 5.1 Database design



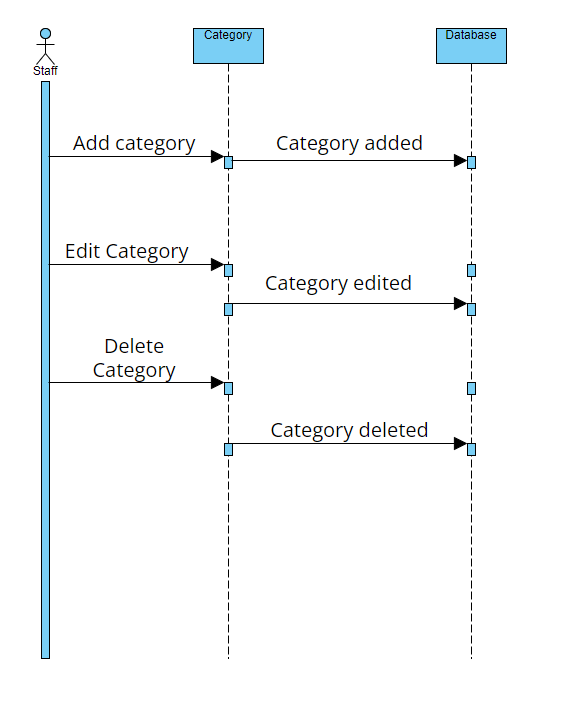
## 5.2 Class Diagrams

****

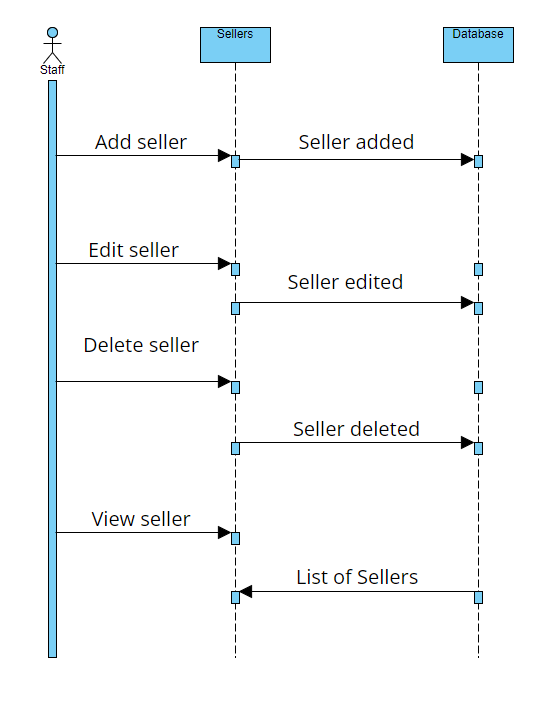
## 

## 5.3 Sequence diagrams

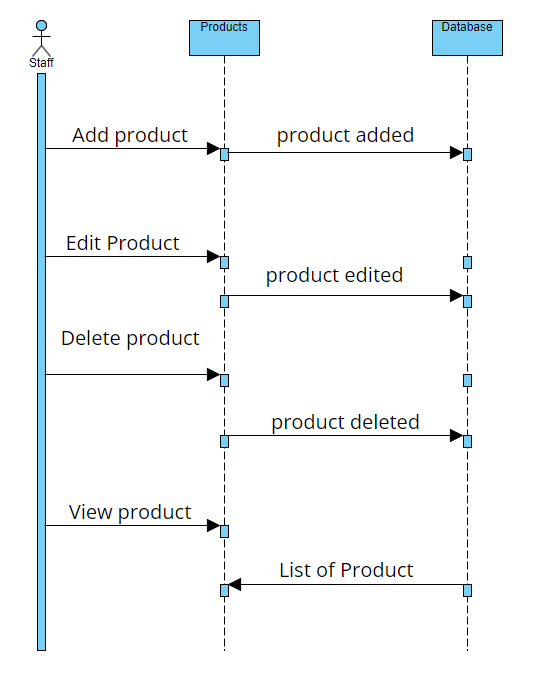
**\*\* Category Management:**

****

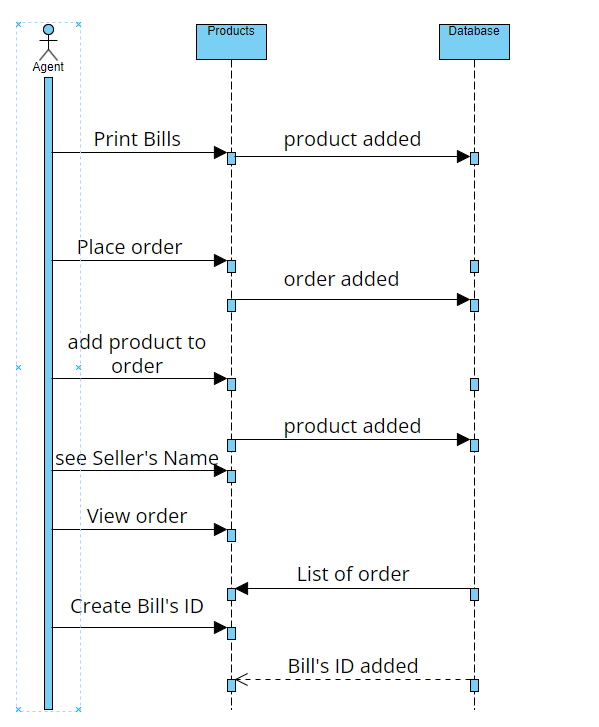
**\*\* Seller Management:**

****

**\*\* Product Management:**

****

**\*\* Selling Mangement:**

****

## 5.4 Rationale for your detailed designn model

- Each Agent(Seller) has an username, a password, phone and their age.

- There is only three admin with given usernames and passwords to monitor all the form and process.

- A product has a name(ProdName), an ID(ProdID), belongs to a category(ProdCategory) and quantity of it(ProdQuantity).

- A category has a name(CatName), an ID(CatID) and a Description about it(CatDes).

- An order has a Bill ID (BillID), Seller’s Name(SellerName), Bill’s date (BillDate) and the total cost of that order(TotalAmount) which equals to Quantity\*Price of every products combined.

## 5.5 Traceability from requirements to detailed design model

- The Staff(admin) will be able to add, edit and delete the category of a product.

🡪 Table Categories(CatID, CatName, CatDes).

* The staff(admin) will be able to add, edit and delete the seller’s information.
  + Table Sellers(SellerName, SellerID, SellerPhone, SellerAge, SellerPass)..
* The staff(admin) will be able to ad, edit and delete the products.
  + Table Products(ProdName, ProdID, ProdCategory, ProdQuantity, ProdPrice).
* The Agent(Seller) will be able to place order and print out bills.
  + Table Bills(BillID, SellerName, TotalAmount, BillDate).

# CHAPTER 6 – TEST PLAN

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

- Add, edit, delete categories.

- Add, edit, delete sellers.

- Add, edit, delete products.

- Add, edit bills, orders.

## 6.2 Traceability of test cases to use cases

\*\* Use case for Product:

- Add product

- Edit product

- Delete product.

\*\* Use case for Category:

- Add Category.

- Edit category.

- Delete category.

\*\* Use case for Sellers:

- Add seller.

- Edit seller.

- Delete seller.

\*\* Use case for Bills:

- Add Bill’s ID.

- Add products to order

- Edit order.

- Delete order.

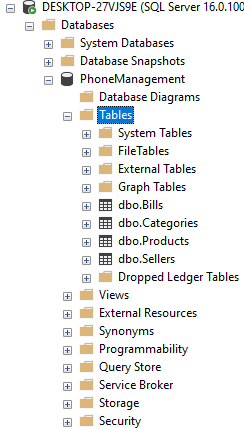
- Print out bills.

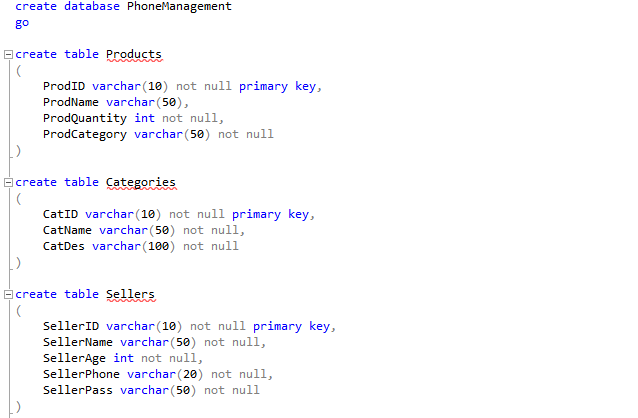
## 6.3 Techniques used for test generation

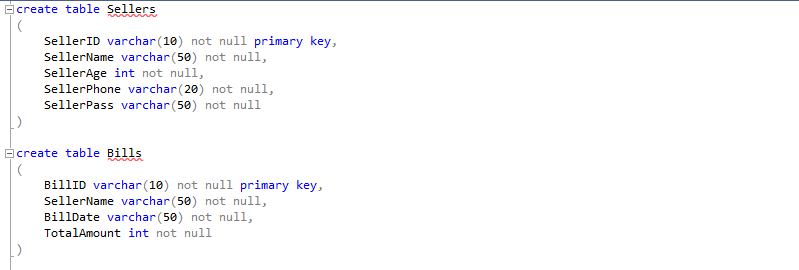
## 6.4 Assessment of the goodness of our testsuite

# CHAPTER 7 – DEMO

## 7.1 Database

****

****

****

****

## 7.2 Source Code: