VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL REPORT**

**SUPPLEMENT FACTS PRODUCTS**

*Instructor*: **Master PHAM THAI KY TRUNG**

*Implementer*: **NGUYEN QUOC AN - 520H0601**

**NGUYEN TU BAO - 520H0515**

Class**: 20H50304-20H50201**

Course**: 24**

**HỒ CHÍ MINH CITY, YEAR 2022**

VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL REPORT**

**SUPPLEMENT FACTS PRODUCTS**

*Instructor*: **Master PHAM THAI KY TRUNG**

*Implementer*: **NGUYEN QUOC AN - 520H0601**

**NGUYỄN TU BAO - 520H0515**

Class**: 20H50304-20H50201**

Course**: 24**

**HỒ CHÍ MINH CITY, YEAR 2022**

THANK YOU

First of all, I would like to thank Ton Duc Thang University for Software Engineering into the curriculum. In particular, I would like to express my deep and sincere gratitude to Mr. Pham Thai Ky Trung who wholeheartedly guided and guided me during the study and writing of this essay. Thank you for your kind words and enthusiasm, which helped me to complete my thesis.

Thank you to my friends who have accompanied and encouraged me throughout the research process.

Thank you very much.

**SUBJECT TO COMPLETED AT TON DUC THANG UNIVERSITY**

I hereby declare that this is my/our own project and is under the guidance of Dr. PHAM THAI KY TRUNG; The research contents and results in this topic are honest and have not been published in any form before. The data in the tables for analysis, comments and evaluation are collected by the author himself from different sources, clearly stated in the reference section.

In addition, the project also uses a number of comments, assessments as well as data from other authors, other agencies and organizations, with citations and source annotations.

**If I find any fraud, I will take full responsibility for the content of my project.** Ton Duc Thang University is not related to copyright and copyright violations caused by me during the implementation process (if any).

*Hồ Chí Minh City, day 5 month 12 year 2022*

*Author*

*(Sign and write full name)*

*Nguyen Quoc An*

*Nguyen Tu Bao*

TEACHER’S CONFIRMATION AND ASSESSMENT SECTION

**The confirmation part of instructor**

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

Hồ Chí Minh city, day month year

(Sign and write fullname)

**The evoluation part of the teacher marks the test**

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

Hồ Chí Minh city, day month year

(Sign and write fullname)

TABLE OF CONTENTS

[THANK YOU i](#_Toc121753131)

[TEACHER’S CONFIRMATION AND ASSESSMENT SECTION iii](#_Toc121753132)

[TABLE OF CONTENTS 1](#_Toc121753133)

[SUMMARY 4](#_Toc121753134)

[LIST OF SYMBOLS AND ABBREVIATIONS 5](#_Toc121753135)

[LIST OF TABLES, PICTURES, GRAPHS 6](#_Toc121753136)

[CHAPTER 1: INTRODUCTION 7](#_Toc121753137)

[1.1 Purpose 7](#_Toc121753138)

[1.2 Scope 7](#_Toc121753139)

[1.3 Product Overview 7](#_Toc121753140)

[1.4 Structure of the Document 8](#_Toc121753141)

[1.5 Terms, Acronyms, and Abbreviations 8](#_Toc121753142)

[CHAPTER 2: PROJECT MANAGEMENT PLAN 9](#_Toc121753143)

[2.1. Project Organization 9](#_Toc121753144)

[2.2 Lifecycle Model Used 10](#_Toc121753145)

[2.3 Risk Analysis 12](#_Toc121753146)

[2.4 Hardware and Software Resource Requirements 12](#_Toc121753147)

[2.5 Deliverables and Schedule 13](#_Toc121753148)

[2.6 Monitoring, Reporting, and Controlling Mechanisms 13](#_Toc121753149)

[2.7 Professional Standards 13](#_Toc121753150)

[2.8 Evidence all the artifacts have been placed under configuration management 13](#_Toc121753151)

[2.9 Impact of the project on individuals and organizations 14](#_Toc121753152)

[CHAPTER 3: REQUIREMENT SPECIFICATIONS 15](#_Toc121753153)

[3.1 Stakeholders for the system 15](#_Toc121753154)

[3.2 Use case model 15](#_Toc121753155)

[3.2.1 Goods Received Note 15](#_Toc121753156)

[3.2.1.1 Textual Description for Goods Reveived Note 16](#_Toc121753157)

[3.2.2 Web Agency Order 17](#_Toc121753158)

[3.2.2.1 Textual Description for Web Agency Order 17](#_Toc121753159)

[3.2.3 Goods Delivery Note 19](#_Toc121753160)

[3.2.3.1 Textual Description for Goods Delivery Note 19](#_Toc121753161)

[3.2.4 Statistics 19](#_Toc121753162)

[3.2.4.1 Textual Description for Statistics 20](#_Toc121753163)

[3.3 Functional requirements 20](#_Toc121753164)

[3.3.1 Goods Received Note 20](#_Toc121753165)

[3.3.2 Web Agency Order 21](#_Toc121753166)

[3.3.3 Goods Delivery Note 21](#_Toc121753167)

[3.3.4 Statistic 21](#_Toc121753168)

[3.4 Non-functional requirements 22](#_Toc121753169)

[3.4.1 Goods Received Note 22](#_Toc121753170)

[3.4.2 Web Agency Order 23](#_Toc121753171)

[3.4.3 Goods Delivery Note 24](#_Toc121753172)

[3.4.4 Statistics 24](#_Toc121753173)

[CHAPTER 4: ARCHITECTURE 25](#_Toc121753174)

[4.1 Architectural style(s) used 25](#_Toc121753175)

[4.2 Architectural model 27](#_Toc121753176)

[4.3. Technology, software used 28](#_Toc121753177)

[4.4. Rationale for your architectural style and model 29](#_Toc121753178)

[CHAPTER 5: DESIGN 31](#_Toc121753179)

[5.1 Database design 31](#_Toc121753180)

[5.2 Static model – class diagrams 31](#_Toc121753181)

[5.3 Dynamic model – sequence diagrams 31](#_Toc121753182)

[5.3.1 Goods Received Note 31](#_Toc121753183)

[5.3.2 Web Agency Order 32](#_Toc121753184)

[5.3.3 Goods Delivery Note 33](#_Toc121753185)

[5.3.3.1 View Order and Order Detail 33](#_Toc121753186)

[5.3.3.2 Update Status 34](#_Toc121753187)

[5.3.3.3 Payment Success 35](#_Toc121753188)

[5.3.4 Statistics 36](#_Toc121753189)

[5.3.4.1 View Incoming Goods 36](#_Toc121753190)

[5.3.4.2 View OutComing Goods 37](#_Toc121753191)

[5.3.4.3 View Best Selling Goods 37](#_Toc121753192)

[5.3.4.4 View Monthly Revenue 37](#_Toc121753193)

[5.4 Rationale for your detailed design model 38](#_Toc121753194)

[5.5 Traceability from requirements to detailed design model 38](#_Toc121753195)

SUMMARY

As mentioned above, the functional food program was written by us for the purpose of practical application and specifically the importation to the company and the export to the agents, in order to help the management. Manage items more conveniently and optimize the performance of management work.

Applying knowledge of databases and analysis - designing and building programs on functional foods, automatically performing tasks by computers that can partially replace human work.

The program's primary features are Winform software for accountants to create warehouse receipts, order webforms for agents to place orders and Winform delivery notes for accountants to create delivery notes and update status. Order status, software for examining import and export records, top-selling goods, and monthly income reports. Finally, a website where people may purchase goods.

LIST OF SYMBOLS AND ABBREVIATIONS

LIST OF TABLES, PICTURES, GRAPHS

**List of tables**

**List of pictures**

CHAPTER 1: INTRODUCTION

* 1. Purpose

This document offers a thorough architectural overview of the new Final\_Exam modules, utilizing a variety of architectural views to illustrate various module features. The document, which is designed to capture and communicate the key architectural decisions made about the modules, acts as a link between the software requirements and the specifics of the Final\_Exam design. It will also assist software architects in ensuring that the modules they are building will satisfy user requirements in terms of functionality (selected functional view), platform, and technology (logical views).

* 1. Scope

From a high level perspective, the paper describes the software architecture that satisfies the Final\_Exam criteria for functionality, availability, reliability, scalability, maintainability, and management.

The introduction of n-tier architecture, which is shown in the diagrams of the proposed structure, is the best way to conceptually express the new Final\_Exam modules.

* 1. Product Overview

The sections that follow will give you a more in-depth look at the architecture of the new Final\_Exam modules, explaining the functionality that it can offer and highlighting its scalability to accommodate future changes resulting from user requirements or alterations to how the system interfaces with other third-party applications/components. We can make sure the design will fulfill all stated needs and yet provide room for future development by looking at it from the key perspectives of Functional and Logical.

This will also include any relevant technologies that will be used for a particular function of the software architecture.

* 1. Structure of the Document
  2. Terms, Acronyms, and Abbreviations

CHAPTER 2: PROJECT MANAGEMENT PLAN

2.1. Project Organization

My group consists of 2 members:

Member 1: Nguyen Quoc An (Winform application)

+ Missions:

* Analyze winform requirements and draw corresponding diagrams
* Software development: Nguyen Quoc An will write code and evolve the software products about Winform application because developers are the actual members who write code to make the software function.
* Software testing: Nguyen Quoc An is responsible for designing test scenarios for software usability, running these tests, and preparing reports on the effectiveness and defects to the production team.
* Software maintenance: Nguyen Quoc An be charge of maintaining software functionality, make upgrades to the coding, and ensure any repairs needed to the software are completed

Member 2: Nguyen Tu Bao (Webform)

+ Missions:

* Analyze webform requirements and draw corresponding diagrams
* Software development: Nguyen Tu Bao will write code and evolve the software products about Webform because developers are the actual members who write code to make the software function.
* Software testing: He is responsible for designing test scenarios for software usability, running these tests, and preparing reports on the effectiveness and defects to the production team.
* Software maintenance: He be charge of maintaining software functionality, make upgrades to the coding, and ensure any repairs needed to the software are completed.

2.2 Lifecycle Model Used

The waterfall model is a linear, sequential approach to the software development lifecycle (SDLC) that is popular in software engineering and product development.

Waterfall model include five stages: planning, design, implementation, verification, and maintenance.

* Requirements and Planning

The requirements and planning phase of waterfall project management identifies what the project should do. The project manager tries to understand the project’s requirements based on what the project sponsors need. This phase involves identifying and describing the project’s risks, assumptions, dependencies, quality metrics, costs, and timeline.

* Design

The design phase solidifies and documents all your decisions. In this case, you develop solutions that can solve the project’s requirements. The best way to do so is to note all the actions you’ll take to deliver the project scope to execute them. Design covers the project’s schedule, budget, and objectives, and you can think of design as a blueprint or road map to the complete project.

* Implementation

The implementation phase executes your project plan and design to produce the desired product. If your company develops software, you will spend this phase coding the software functionalities. Or, if you’re managing a project at a construction company, you will construct a house in this phase. Implementation takes up a significant portion of waterfall project management. Everything that happens during this phase should be carefully documented.

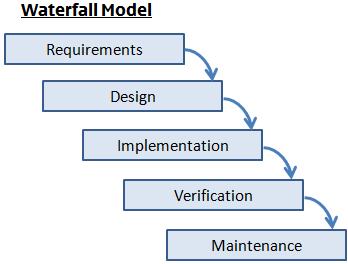
* Verification/Testing

Testing verifies that the product developed in the implementation phase fulfills the entire project’s requirements. If this is not the case, the project team must review the project from phase one to identify what went wrong. The testing phase uses various quality metrics and customer satisfaction to measure the project’s success.

* Maintenance

The maintenance phase extends beyond the five stages of project management into the project’s lifetime. This phase involves making minor modifications to improve the product developed during implementation and performing other routine maintenance tasks. It’s also a phase to identify any errors you might have missed during the testing phase

Illustrating images:



As shown in content 2.1, you can see that our group has followed the waterfall model process. So our team decided to use waterfall model because Waterfall relies on teams following a sequence of steps and never moving forward until the previous phase has been completed.

2.3 Risk Analysis

Since the project was developed rapidly, mistakes like sql injection, interface conflicts in interaction, etc., might be discovered if the code is closely examined. Both the database and the user experience will suffer greatly as a result of these problems.

The greatest strategy to lessen these harmful mistakes is to eliminate challenging functions that are still unfinished. Ensure that the database operates properly at the same time by carefully completing the fundamental tasks. Even if there aren't any challenging tasks, it nonetheless makes sure the project is safe.

2.4 Hardware and Software Resource Requirements

|  |  |
| --- | --- |
| Hardware | Software |
| Processor: x86 or x64  RAM: 512 MB (minimum), 1 GB (recommended)  Hard disk: up to 4 GB of space may be required. However, even if you install in a different drive, 400 MB of free space is required in the boot drive. | Winform, Web Application: Microsoft Visual Studio 2017/2019/2022  System Framework: .NET Framework 2.0/3.5/4.0/4.5/4.5.1/4.6  Database: Microsoft SQL server |

2.5 Deliverables and Schedule

Do all requirements (Code, draw usecase, diagram, report, search new information ,..) 3 day for built a plan, 2 week for search new information and complete the requirements, after that do report and complete all other requirement

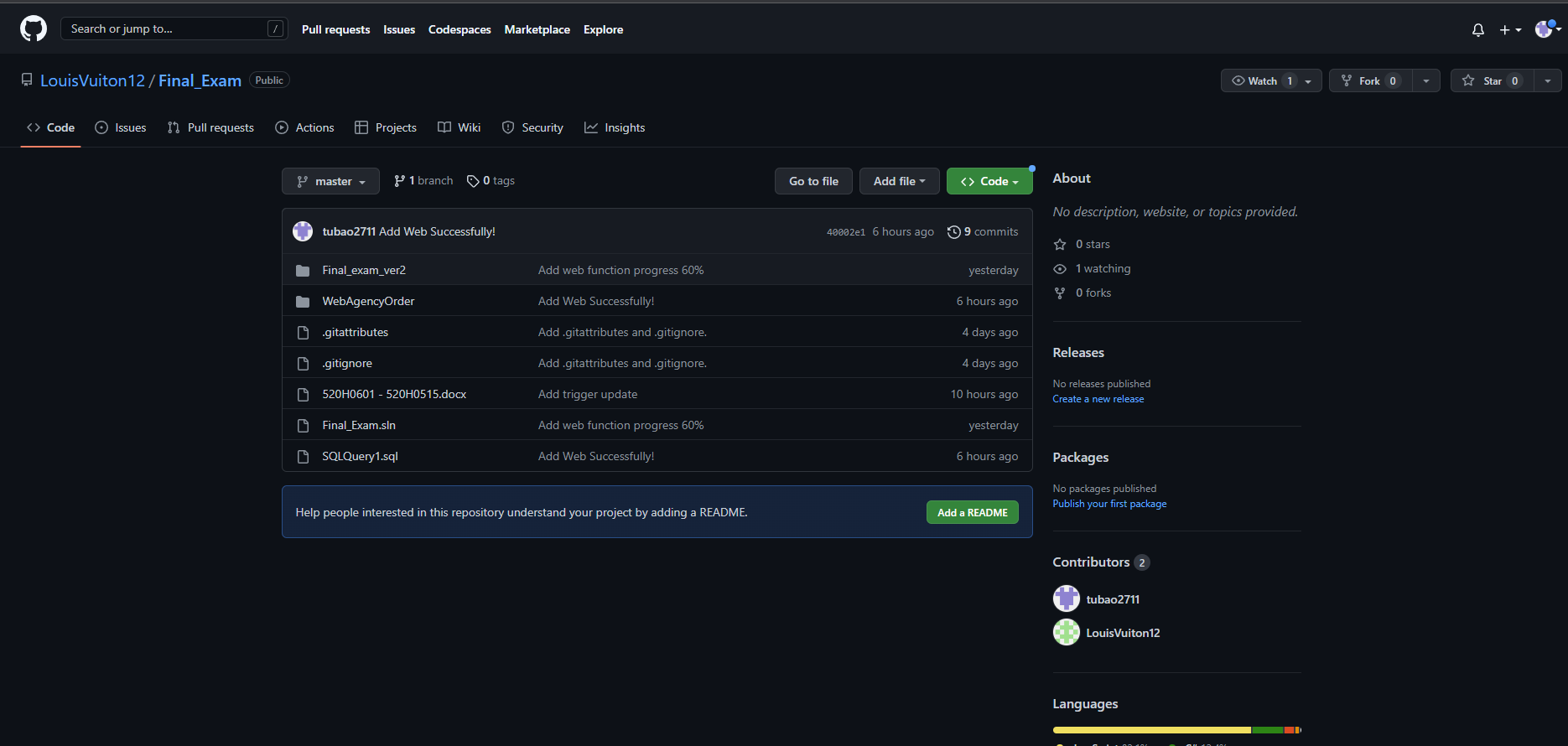
2.6 Monitoring, Reporting, and Controlling Mechanisms

In group 2 people, we monitoring, reporting and controlling by github. We push everything we done on github then we continue the unfinished problems. Github very useful we can saw code each other and update, delete or insert more information.

2.7 Professional Standards

We plan and complete the work for each person, there will be penalties for those who do not complete the schedule but our team did not make any mistakes and completed each individual's part as expected. But besides that, there are still some mistakes in the process of completing each person's work, sometimes checking again is not according to the requirements of the problem, errors in the code, but those problems have also been resolved and continued. keep up the good work.

2.8 Evidence all the artifacts have been placed under configuration management



2.9 Impact of the project on individuals and organizations

The project tells us what solidarity and collective thinking is, lets us understand the feeling of disagreement among members, the feeling of joy when completing the work, the project brings many emotions for the individual as well as the group. We are happy to have the opportunity to work together and hope to have the opportunity to work and grow together with such groups in the future.

CHAPTER 3: REQUIREMENT SPECIFICATIONS

3.1 Stakeholders for the system

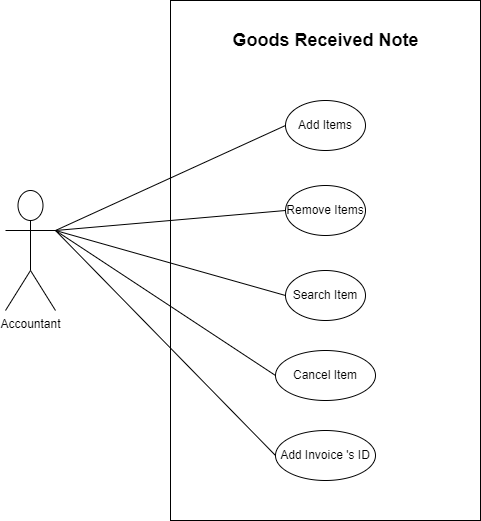
Accountant: Winform softwares will be used by people to produce Goods Received Note as well as see incoming and outgoing stock reports, the best-selling items, and monthly income reports.

Agency: Agents can order items, choose a payment method and view order status

Customers: Customers can select the ideal product to purchase.

3.2 Use case model

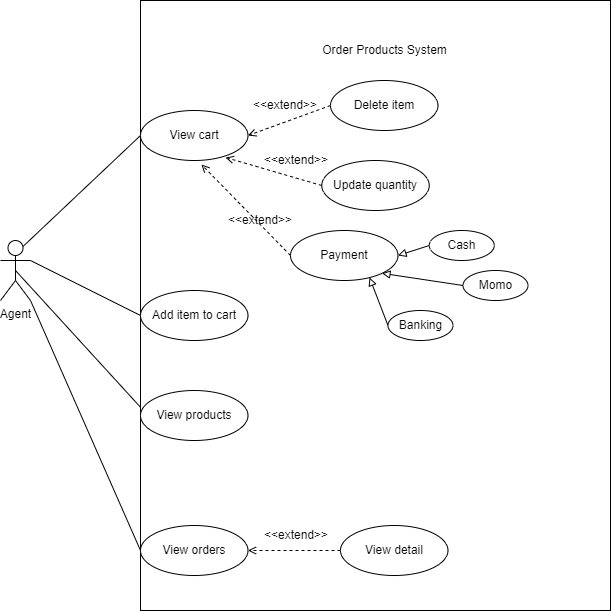
3.2.1 Goods Received Note



3.2.1.1 Textual Description for Goods Reveived Note

|  |  |
| --- | --- |
| *Add items* | The items will be added to the inventory by the accountant through the sytem |
| *Remove items* | If there is a mistake after adding an item, ask the system to remove the incorrect products. |
| *Search item* | Search the product through its id |
| *Cancel item* | In the process of entering information, if there are too many errors, the accountant has the right to request the system to delete all information and re-enter it |
| *Add Invoice’s ID* | Add invoice's id for items and export the Goods Received Note |

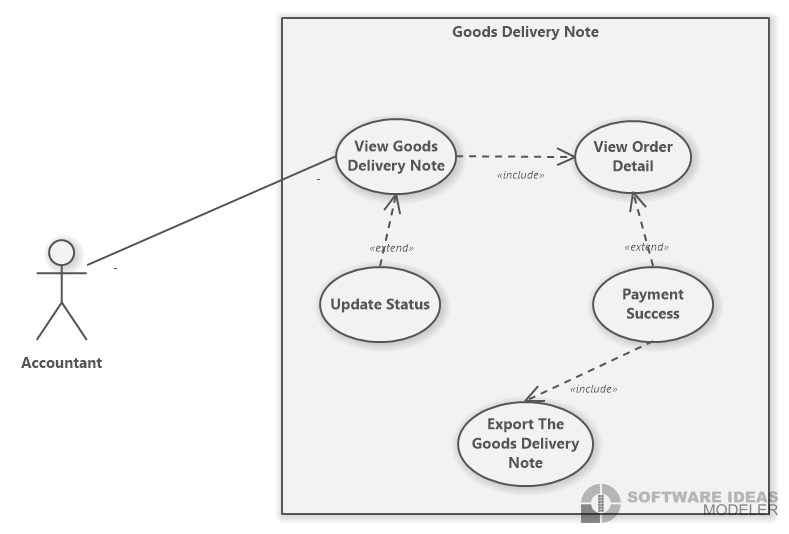
3.2.2 Web Agency Order



3.2.2.1 Textual Description for Web Agency Order

|  |  |
| --- | --- |
| *View product* | A list of products is displayed along with information such as quantity, price, product name and benefits of each product. |
| *Add item to cart* | Add the product which user want to buy and the product, the product information will be saved in the cart |
| *View cart* | Information about the products that have been added to the cart will be displayed |
| *Delete item* | Delete the products user want to delete |
| *Update quantity* | Update quantity for items which user want, agent can be increase or decrease the amount. |
| *Payment* | User can choose to pay by cash, momo or bank transfer |
| *View orders* | Review all invoices that user have paid or are waiting to pay, each invoice of each user will have a different code, date and status |
| *View details* | View details of each invoice, including the invoice code, product name, quantity and price |

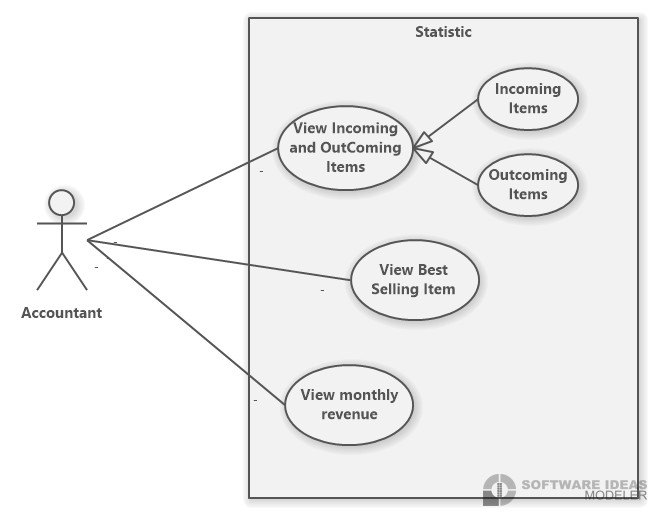
3.2.3 Goods Delivery Note



3.2.3.1 Textual Description for Goods Delivery Note

|  |  |
| --- | --- |
| *View Goods Delivery Note* | View information of Goods Delivery Note |
| *Update Status* | Update status when agents have completed payment |
| *View Order Detail* | View order details |
| *Payment Success* | Notify accountant of completed orders and export to Excel file |

3.2.4 Statistics



3.2.4.1 Textual Description for Statistics

|  |  |
| --- | --- |
| *View Incoming and OutComing Items* | Statistics of goods coming in and out of the company |
| *View Best Selling Item* | Show best selling items |
| *View monthly revenue* | Show monthly revenue |

3.3 Functional requirements

3.3.1 Goods Received Note

* Allow input of product information
* Click the Add item button to be able to add products to the system
* During the input process, if there are too many mistakes, the user can click the cancel button to delete the input again.
* We can remove the item if the item list has an error through the remove item button
* Find product information through ID
* When adding an invoice successfully, a notification will be sent to the user, otherwise, an alert dialog will be displayed for the user
* The other buttons will be locked, apart from the add item button, if there is no product on the receipt view

3.3.2 Web Agency Order

* User can view list products in Homepage by click on Shop now.
* The user has access to the products' information.
* User can view the orders.
* User can choose their agent.
* The goods' quantities can be updated by the user.
* User can delete product.
* The user has the option of paying for their receipt.
* User can view detail the order.

3.3.3 Goods Delivery Note

* Users can view products
* User can view product details
* User can alter the status of the product.
* User can confirm the order is completed and export to Excel file.

3.3.4 Statistic

* By selecting the View Incoming Good button, users can see the incoming items.
* By selecting the View Outcoming Good button, users can see the outcoming items.
* User can see the Best-Selling Goods.
* User can view Monthly Revenue.

3.4 Non-functional requirements

3.4.1 Goods Received Note

|  |  |
| --- | --- |
| Performance | Users shouldn't have to wait more than a second for the Winform to load. |
| Reliability | Can access products without errors |
| Availability | Accountant has the right to add products on Monday to Saturday. In the case of unplanned system downtime, all features will be available again after one working day. |
| Capacity | Up to 1,000,000 items can be stored. |
| Usability | The Winform’s interface has to be user-friendly and easy to use. |
| Manageability | Only people with relevant positions can view product information |
| Recoverability | If a major incident happens on the Winform and company must take measures to go back to being fully operational within three days. |
| Environmental | Goods Received Note are exported Monday through Friday from 9 AM to 6 PM. |
| Portability | Works very well on windows operating system |

3.4.2 Web Agency Order

|  |  |
| --- | --- |
| Usability | The system must be intuitive and simple in the way it displays all relevant data and relationships. |
| Reliability | The System must successfully add product, update product by the user and provide estimations and inventory status in relevance with the newly updated entities |
| Performance | All the functions of the system must be available to the user every time the system is turned on. |
| Supportability | The software is designed such that it works even on systems having the minimum configuration. |
| Packaging | The system must be able to run on the Windows operating systems beginning with Windows XP, and must be able to run on future releases. |

3.4.3 Goods Delivery Note

|  |  |
| --- | --- |
| Usability | The list of product must be easily navigable by the users with buttons that are easy to understand |
| Reliability | The system should not update the data in any database for any failed processes. |
| Performance | The system must not lag, because users using it don’t have down-time to wait for it to complete an action. |
| Supportability | The software is designed such that it works even on systems having the minimum configuration. |
| Packaging | The system must be able to run on the Windows operating systems beginning with Windows XP, and must be able to run on future releases. |

3.4.4 Statistics

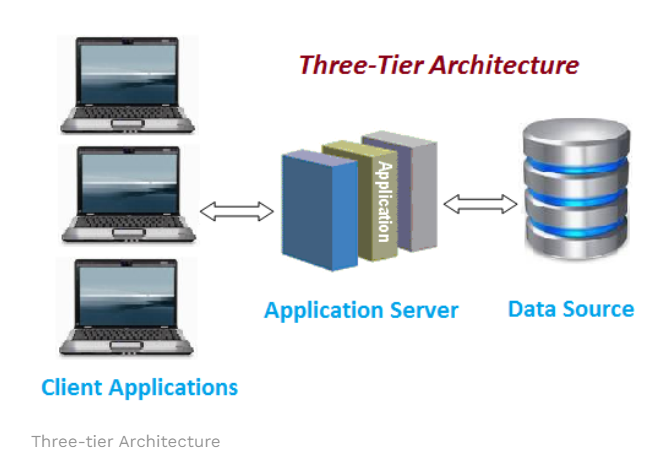
|  |  |
| --- | --- |
| Usability | The Winform’s interface has to be user-friendly and easy to use. |
| Reliability | The System must give accurate inventory status to the user continuously. Any inaccuracies are taken care by the regular confirming of the actual levels with the levels displayed in the system |
| Performance | Users shouldn't have to wait more than a second for the Winform to load. |
| Supportability | The software is designed such that it works even on systems having the minimum configuration. |
| Packaging | The software must incorporate a license key authentication process.  The system must be able to run on the Windows operating systems beginning with Windows XP, and must be able to run on future releases. |

CHAPTER 4: ARCHITECTURE

4.1 Architectural style(s) used

This desktop based application is based on 3-tier architecture of .Net Framework. The 3-tier architecture provides three hierarchies for the programming logic flow from user interface to database and back from database to user interface with the clients' required information. In between there involves the logic layer for effectively and correctly manipulating the request. The 3-tier includes the following:

|  |  |
| --- | --- |
| Client Tier | The visual part, which does not perform database calls, is implemented using a variety of swing components. This tier's primary purpose is to respond to user requests for information that are initiated by user inputs such button presses. If a user wishes to see a list of the organization's remaining stock, for instance, he or she might click the "display" button. |
| Business Tier | For database queries, the client uses the *business logic* intermediate layer. It offers the system's essential features and connectivity to the data tier, which makes it easier for the Client Tier to do tasks. |
| Data Tier | Data layer is also the class which gets the data from the business tier and sends it to the database or gets the data from the database and sends it to business tier. This is the actual DBMS (database management system) access layer or object layer also called the business object. Information that can be obtained via MySql database Connectivity is stored in the database backend. By sending intricate database queries, mysql database connection is utilized to manage the communication between the middle tier and the backend database. |



4.2 Architectural model

- Designing a website with order-taking functionality for agents using the MVC (Model – View – Controller) model

- Create three sections for the interactive application, including:

+ Model – includes functions and error information

+ View – show data for the user (Can define multiple views)

+ Controller – handle input from the user



- This model makes it easier to distinguish between the layers that offer internal information and the information that users are presented with and accept. Code may be written more effectively since it separates component dependencies.

4.3. Technology, software used

|  |  |
| --- | --- |
| Application Server | |
| Winform Application | Visual Studio 2022 |
| Operating Systems | Windows Server |
| HTTP Server | |
| Web Server | IIS EXPRESS |
| Operating Systems | Windows Server |
| Database Server | |
| Operating Systems | Windows Server |
| Windows Server | Microsoft SQL Server Management Studio 2018 |

4.4. Rationale for your architectural style and model

- An architectural design called microservices organizes an application as a group of tiny, loosely linked services that cooperate to complete a single task. They may be added, deleted, or upgraded without affecting other programs because they operate separately.

- There are several advantages to using a microservices architecture, including simpler deployment and testing, increased productivity, flexibility, and scalability. They do have certain drawbacks, though, as independent microservices need a flawless mechanism of communication to operate as a bigger program.

- Real-time communication between microservices is made possible by event-driven microservices, allowing data to be consumed in the form of events before it is even necessary.

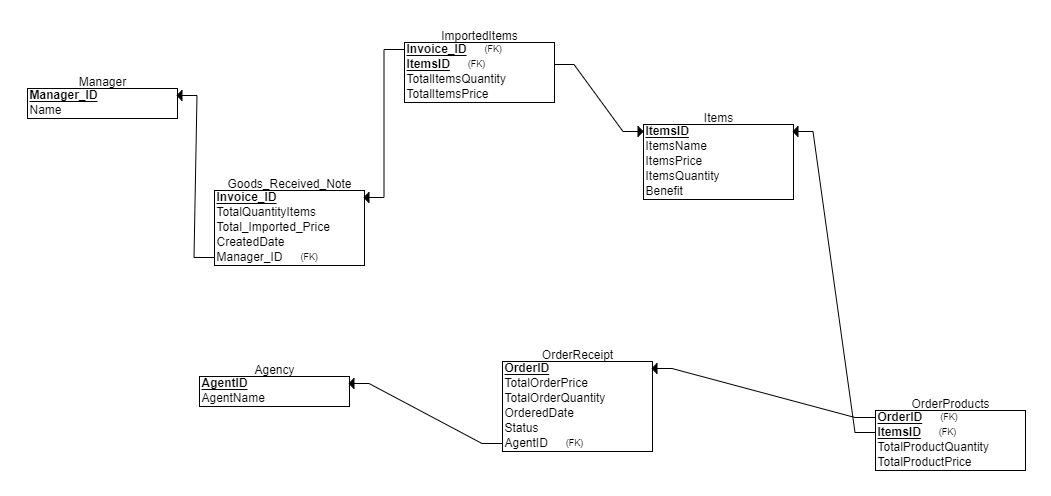
- When at the data center, we primarily concentrate on the present state of the data.

- When employing an event-driven architecture, event producers create and deliver event notifications, and one or more event information receivers may be present. Event information receivers may then initiate handling logic.

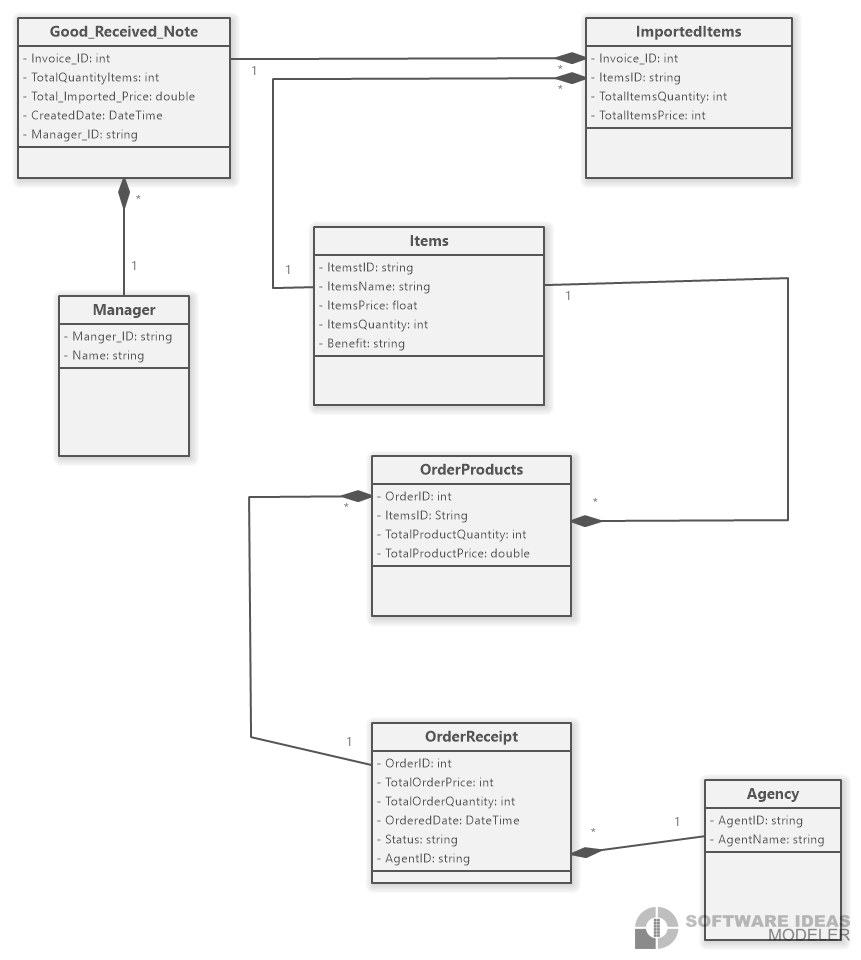
- Example: The Accounting application may be waiting and listening for the notice of the new order that Web Example just got before activating their own internal system to communicate information about the event to their users.

CHAPTER 5: DESIGN

5.1 Database design

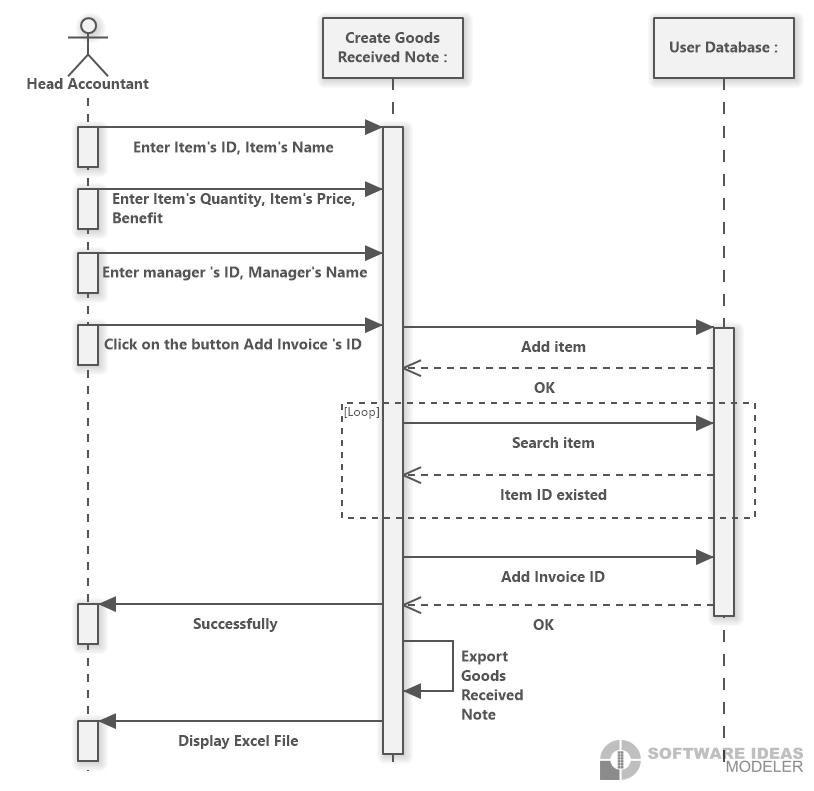


5.2 Static model – class diagrams

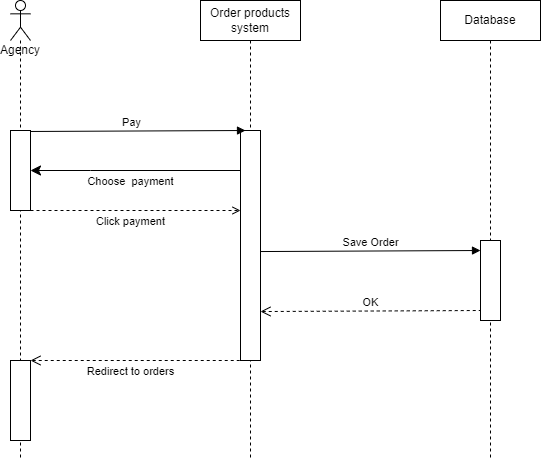


5.3 Dynamic model – sequence diagrams

5.3.1 Goods Received Note

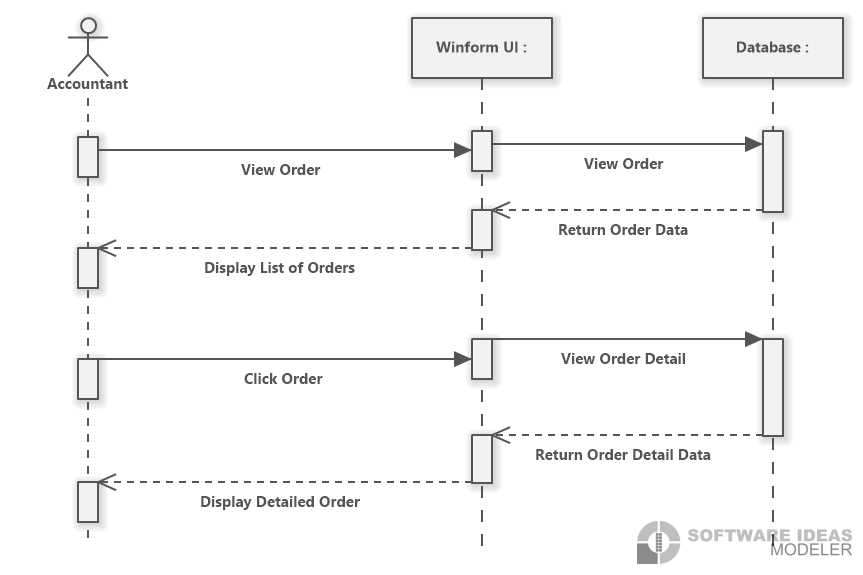


5.3.2 Web Agency Order

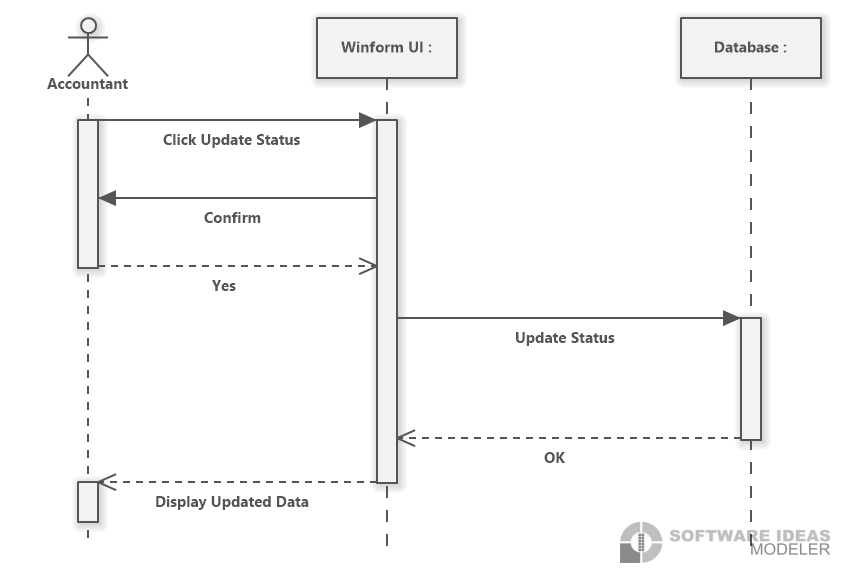


5.3.3 Goods Delivery Note

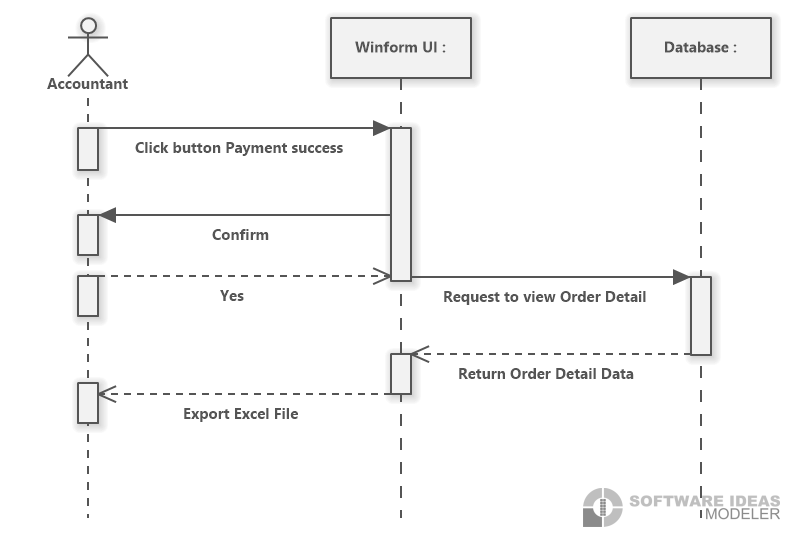
5.3.3.1 View Order and Order Detail



5.3.3.2 Update Status

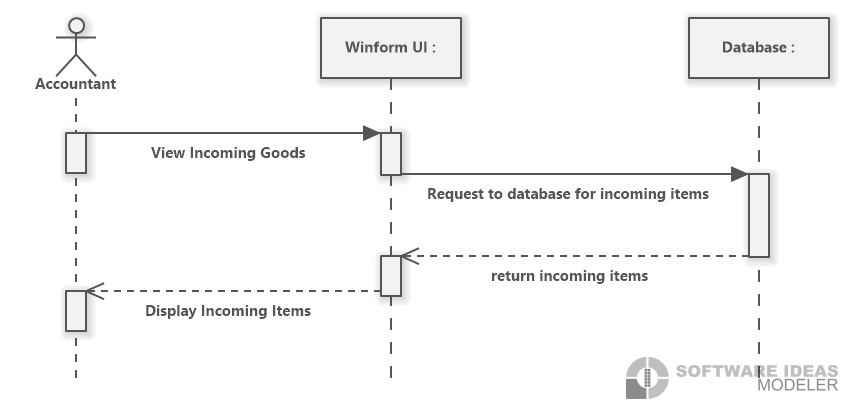


5.3.3.3 Payment Success

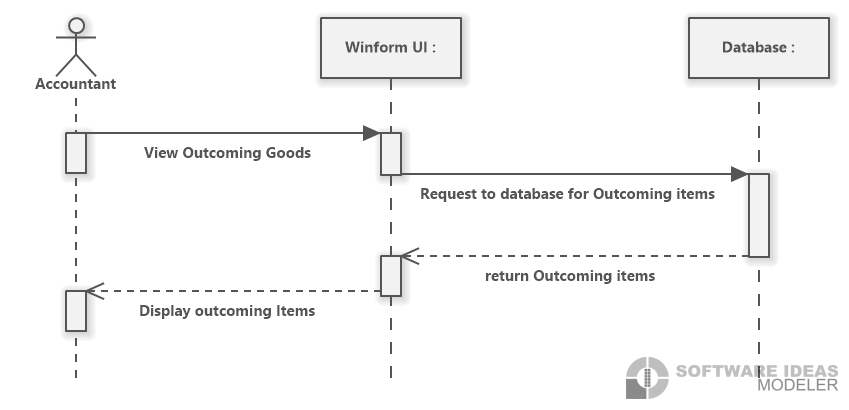


5.3.4 Statistics

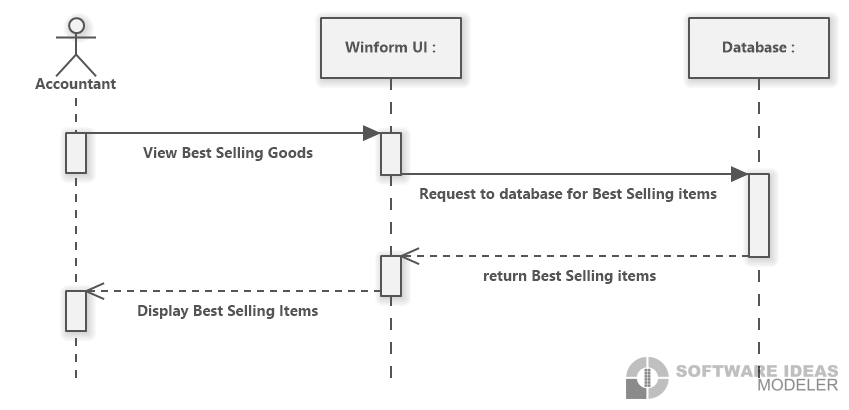
5.3.4.1 View Incoming Goods



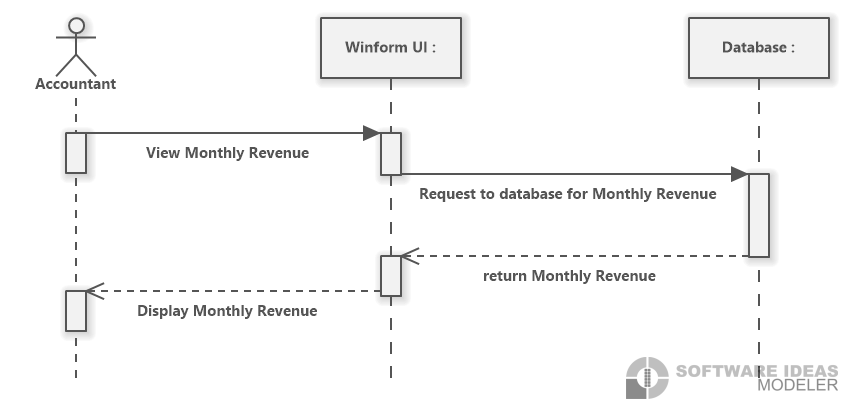
5.3.4.2 View OutComing Goods



5.3.4.3 View Best Selling Goods



5.3.4.4 View Monthly Revenue



5.4 Rationale for your detailed design model

Our team finds that the challenge has five software needs. For each request, a unique piece of software is created for the customer. On the database side, however, all the software uses the same database, SUPPLEMENT FACTS PRODUCTS, thus our team created the database based on the details from the product's specifications. Of course, each software's features are taken into account while designing the use cases and sequence diagrams. Only the class diagram has a different overall design.

5.5 Traceability from requirements to detailed design model

*Requirement 1*: Create a window form application for accountants to export warehouse receipts, according to the requirement. Many items are on each receipt and for each receipt there's a manager. Therefore, we made the decision to create a product table for the warehouse's items. We require a warehouse receipt table and a thorough warehouse receipt since we must keep warehouse receipt. Products that are imported will be stored according to a detailed warehouse invoice.

*Requirement 2:* Create a web application so that agents may place orders, according to the assignment. So, we made the decision to establish three tables: an agent table that contains information about agents, an order table that contains orders for agents, and a detailed order table that contains information about detailed orders for agents.

*Requirement 3:* We will reuse the tables in request 2 to make delivery notes to agents

*Requirement 4:* To compute or evaluate, this program only uses already constructed tables. The warehouse receipt table and order receipt table may be used to list the monthly revenue report, best-selling goods, and incoming/outgoing stock

CHAPTER 6: TEST PLAN

6.1 Requirements/specifications-based system level test cases

Below table lists the features that will be tested during the current test or the subsequent planned tests.

6.1.1 Goods Received Note

|  |  |
| --- | --- |
| ***Features to be tested*** | ***Description*** |
| Adding Invoice ID to database | This test is conducted to verify if a invoice is successfully added to the database. This will also check if an invoice can contain multiple products |

6.1.2 Web Agency Order

6.1.3 Goods Delivery Note

|  |  |
| --- | --- |
| ***Features to be tested*** | ***Description*** |
| Updating a status | This test checks for the correct updating of the selected order. |

6.2. Traceability of test cases to use cases

6.3. Techniques used for test generation

*Manual Test data generation*: In this technique, all the datasets are generated manually by the tester with respect to all the required test case through experience and anticipations.

* Pros: Easy to implement, no additional tools are needed to be deployed.

Increase the confidence of the tester.

* Cons: Accuracy of data sets generated by this scheme mostly doubtful.

Time-consuming process.

6.4. Assessment of the goodness of your testsuite

6.4.1 Testcase 1: Adding Invoice ID to database

* Case 1.1: Testing the Quantity input field.
* Case 1.2: Testing the Item Name field.
* Case 1.3: Testing the available Manager list in database.
* Case 1.4: Testing the all the above cases together and checking if the entries are updated to the tables in database.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case Identifier** | **Test Items** | **Input Specifications** | **Output Specifications** | **Special Procedural Requirements** |
| Case 1.1 | Quantity text field | 1) Input negative numbers.  2) Input String  3) Input zero.  4) Leave the field blank  5) Enter special character in the field.  6) Input integer numbers | 1) Input specifications 1, 2, 3, 4, 5 must generate ‘Message Box’ asking the user to re-enter the text in the field. 2) Input specification 6 should not generate any error. | Enter a quantity in the quantity field and press add item button. |
| Case 1.2 | Item Name | 1) Input numerical value for the name  2) Leave the field blank  3) Enter an existing Item name.  4) Enter special characters in the field  5) Input String | 1) Input specifications 3 and 4 must generate “Message Box ” asking the user to re-enter the text in the field.  2) Input specification 1, 5 should not generate any error. | Enter a name for the item along with appropriate quantity and press the add Item button. |
| Case 1.3 | Available Manager list in database. | 1) One ID is added twice to the list. | 1) The input specification 1 result in an exception being thrown. | Selected one Manager, enter an appropriate quantity and press Add Invoice ID |
| Case 1.4 | Testing the components mentioned above together and adding a Invoice ID to the database | 1) All the required quantities are inserted into their respective fields. | 1) If all the above tests are passed without an exception, the Invoice ID is successfully added to the database | Selected one Manager ,enter an appropriate quantity and press Add Invoice ID |

**Preliminary test results for test case 1:**

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Completed / Not Completed** | **Result summary** |
| Case 1.1 | Completed | The results for all the input specification for this test is passed and no difference was detected between the actual and the expected results. |
| Case 1.2 | Completed | The results for the mentioned input specifications have been passed |
| Case 1.3 | Completed | The results for all the input  specification for this test is  passed and no difference was detected between the actual and the expected results. |
| Case 1.3 | Completed | The results for all the input specification for this test is passed and no difference was detected between the actual and the expected results. |

6.4.2 Testcase 2: Updating a status

* Case 2.1: Check the Order.
* Case 2.2: Check if the status are updated in the database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case Identifier** | **Test Items** | **Input Specifications** | **Output Specifications** | **Special Procedural Requirements** |
| Case 2.1 | Order | Load data to the winform UI | The Datagridview must show all the current Orders on the database | N.A |
| Case 2.2 | Testing if the selected data is processed properly and updated to the database | Click on any Order on DataGridView to update | A Dialog pops up to confirm again through Yes and No.If click Yes, the database must be checked for appropriate updates, otherwise vice versa. |  |