VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**

****

**SOFTWARE ENGINEERING FINAL PROJECT**

**FUNCTIONAL FACTS SELLING SYSTEM**

*Instructor*: **Master Phạm Thái Kỳ Trung**

*Người thực hiện*: **Trương Trung Hiếu – 519H0164**

**Thích Cảnh Long -**

Lớp **: 19H50204**

Khoá  **: 23**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**

****

**SOFTWARE ENGINEERING FINAL PROJECT**

**FUNCTIONAL FACTS SELLING SYSTEM**

*Instructor*: **Master Phạm Thái Kỳ Trung**

*Student’s name*: **Trương Trung Hiếu – 519H0164**

**LikeD Cảnh Long -**

Class **: 19H50204**

Course  **: 23**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

THANKS

Đây là phần tác giả **tự viết** ngắn gọn, thể hiện sự biết ơn của mình đối với những người đã giúp mình hoàn thành Luận văn/Luận án. Tuyệt đối không sao chép theo mẫu những “lời cảm ơn” đã có.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Purpose and Scope

- Create a system including a website and a winform for selling functional facts.

- Website will be used by agents and winform will be used by the company selling functional facts.

- Help manage the purchase and sale between the two parties more convenient and easier and avoid the risk of loss of goods.

- Reduce paperwork.

- Help the company increase the quality of their services through the website.

1.2 Product Overview

- This system consists of 2 components: a website and a winform.

- Agents will place orders on the company's website and the company manages the items in its inventory via winform.

- Agents when accessing the list will be able to see the list of items, then select the items to buy, add to the cart and proceed to pay.

- The accountant in the company will be the one using the winform, managing the incoming and outgoing items, the order status and viewing the statistics.

CHAPTER 2: PROJECT MANAGEMENT PLAN

2.1 Project Organization

- Some text here…

2.2 Lifecycle Model Used

- For this project we used waterfall model. In this model, the software development process is divided into different phases and executed sequentially, the output of one phase is the input of the next phase and there is no overlap.

- There are 6 phases in this model:

* Requirement Gathering and analysis
* System Design
* Implementation
* Integration and Testing
* Deployment of system
* Maintenance

- The reason why our group chose this model is because:

* This is a simple model, easy to apply, clear step-by-step process.
* Suitable for simple small projects
* The input and output criteria are clearly defined from the beginning
* Easily to arrange schedule work
* It doesn't take much to manage when the project has clear requirements like this

2.3 Risk analysis

- All projects always have potential risks. If not identified early, the risks may affect the work progress and project quality, …

- Possible risks when doing this project:

|  |  |  |  |
| --- | --- | --- | --- |
| Possible risks | The likelihood of risks | Reason of risks | Risk reduction strategies |
| Improper scheduling | High | - Omit important parts  - Delaying 1 thing causes the following problems to be delayed  - Don't know how long it took to complete that task | - Discuss with each other to make sure we have the right schedule  - Make sure the project is on track  - Know exactly what we are doing and how long does it take to do it |
| Changing business requirements too many times | Low | - Do not understand the requirements clearly  - Misinterpreted requirements | - Understand the requirements very clear from the start  - If our team do not understand a problem, we must ask the teacher (in class) or discuss it within the group |
| Manpower attrition | Medium | - 1 member suddenly got sick  - Have a covid infection, or have a fever after getting the vaccine  - There is an unexpected matter at home that must be resolved | - Make sure team members know each other's content well  - Good health protection  - Limit going out near the project deadline |
| Not understanding the technology being used | Medium | - Never used technology before | - Refer to online documents, read documents of that technology  - Exchange of knowledge among members to improve knowledge about that technology |
| Link failure or slow performance | High | - Inexperience when doing group projects  - Can't or difficult to link each member's code together  - Error while linking leads to slow system performance | - After completing a certain function, link them together  - Do not link at once with many functions  - Thoroughly test the test cases before and after linking |

2.4 Hardware and Software Resource Requirements

- For hardware requirements, we require each person to have at least 1 laptop/PC operating stably without errors and powerful enough to do the set requirements.

- For software requirements, each team member must use an operating system that supports the software used in the project.

- The software used in the project includes: Visual Studio Code for developing, Google Chrome/FireFox for debugging and xampp for running localhost and database. Visual Studio Code can be replaced with similar software

- The Laptop/PC also needs to install frameworks like: Node JS, Express JS, Boostrap 4.6.

Throughout the project we’ve learned new………

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

- We use github throughout the whole process of making this project

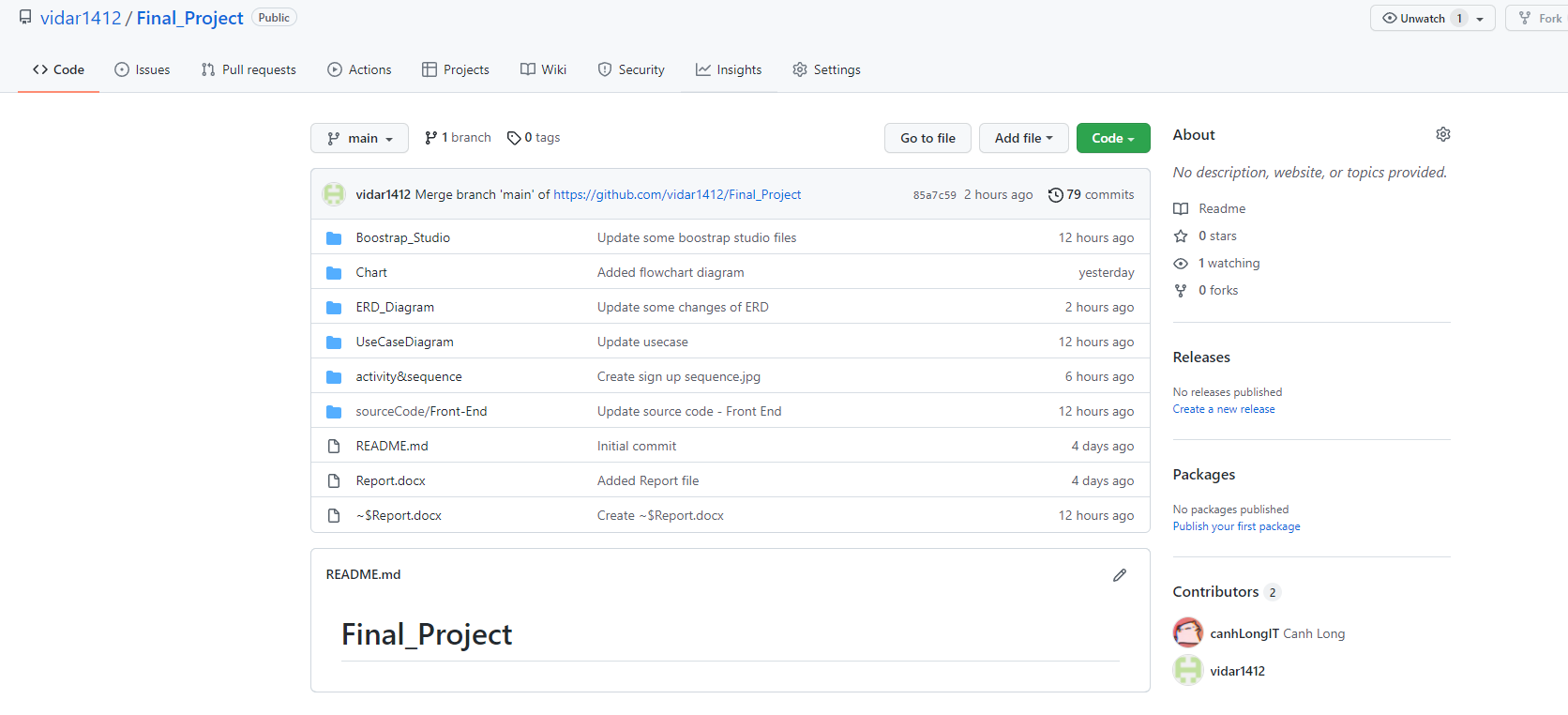


Figure 1: Github Evidence

Here is the link: https://github.com/vidar1412/Final\_Project.git

2.9 Impact of the project on individuals and organizations

- Help individuals in the company work more easily and efficiently with the created winform, it also helps reduce paperwork for them

- Makes importing goods more convenient, because the agency can directly see all the items displayed on the company's website.

- Increase the link between the company and the agent, the working process is fast and partially automated.

- Reduce the cost of items because now there is a working system, so there is no need to hire many people.

- Reduce the loss of goods, increase the reputation of the company.

CHAPTER 3: REQUIREMENT SPECIFICATIONS

3.1 Stakeholders for the system

There are 6 stakeholders for this system:

- Development team

- Company shareholders

- Sales staff

- Customer service department

- Marketing team

- Social media manager

3.2 Use case model

3.2.1 Graphical Use Case Model

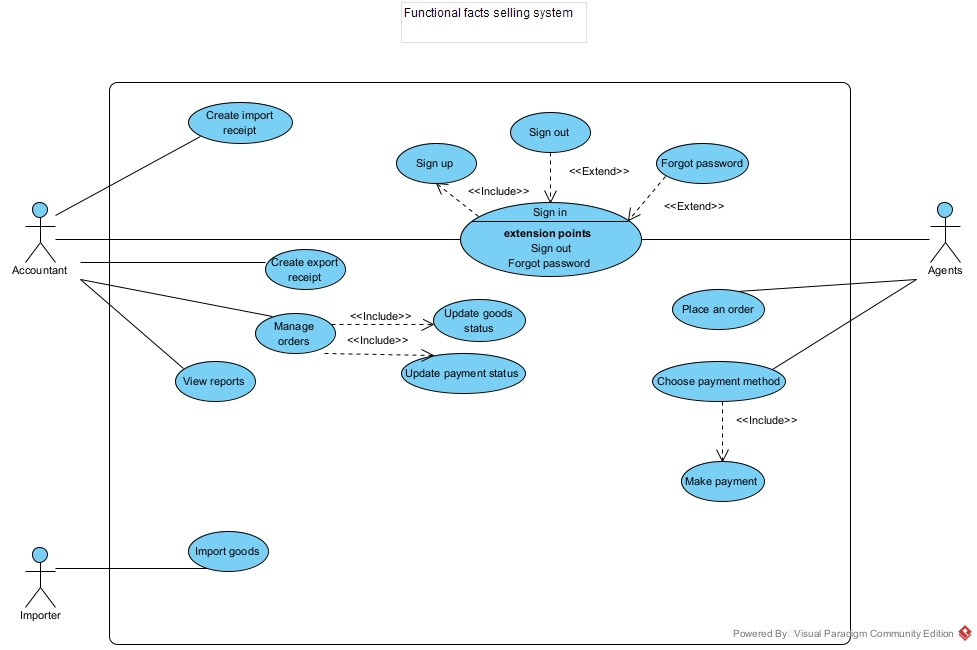


Figure 2: Use Case Diagram

3.2.2 Textual Description for each use case

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | RC-1 | |
| **Use Case Name** | Create warehouse receipt | |
| **Scenario** | Create receipt when company import goods | |
| **Actor(s)** | Accountant, Company | |
| **Priority** | Must have | |
| **Trigger** | The accountant receive a message from the system that the company has import goods | |
| **Pre-Condition(s)** | * Goods must be delivered * Storage status must not be full | |
| **Post-Condition(s)** | * After delivered, system must send an email of the receipt to the company. * The accountant must be able to get information about imports good and confirm or cancel it if the storage is full stocks | |
| **Flow of activities** | Actor | System |
| 1. Company imports goods through the website 2. The accountant checks the goods 3. The accountant clicks the create receipt button | 1. Display a form for company to enter import goods information  2. The system will display the import goods information  4. The website will display a message to notify company to check their email and they will receive email 5 minutes at the latest. |
| **Non-Functional Requirement** | * The form must be loaded smoothly * The email must be sent after the accountant click the create receipt button | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | DN-1 | |
| **Use Case Name** | Create Goods delivery note | |
| **Scenario** | Create Goods delivery note when agents finished payment | |
| **Actor(s)** | Accountant | |
| **Priority** | Must have | |
| **Trigger** | The accountant receive a message form the system that Agents place an order and finished payment | |
| **Pre-Condition(s)** | * The agent has submitted the form successfully * The agent has email validated * The system has received their information | |
| **Post-Condition(s)** | * The note must contain agent submitted information and the delivery slips | |
| **Flow of activities** | Actor | System |
| 1. Accountant check the agent information  2. Accountant checking if the items are in stock  3.Accountant click into the create delivery notes | 1.The system will display the agent information  2.The system will navigate storage and show their current status  3.The system notify agent to check their email and they will receive email of the delivery note in 5 minutes at the latest. |
| **Non-Functional Requirement** | * The email must be sent after the accountant click the create delivery notes | |

3.3 Functional Requirements

Some text here….

3.4 Non-functional requirements

Some text here…

CHAPTER 4: ARCHITECTURE

4.1 Architectural style(s) used

Some text….

4.2 Architectural model

Some text..

4.3 Technology, software, and hardware used

- In this project, for the front end of the website, we use:

* Boostrap Studio
* HTML, CSS and Javascript
* Visual Studio 2019

- For the back end, we use:

* Node JS
* Microsoft SQL
* C#

- For hardware, we use:

* 2 laptops: Acer Nitro 5 and Dell Precision M4800

4.4 Rationale for your architectural style and model

Some text…

CHAPTER 5: DESIGN

5.1 Database design

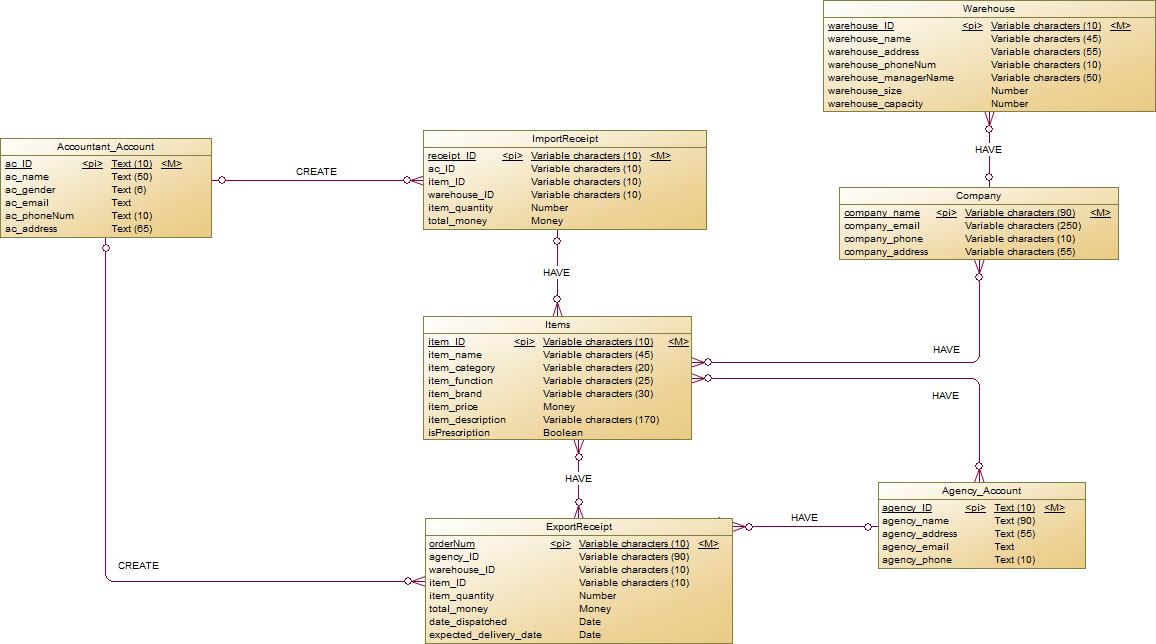


Figure 3: ERD

5.2 Static model – class diagrams

- Some images…..

5.2 Activity Diagrams

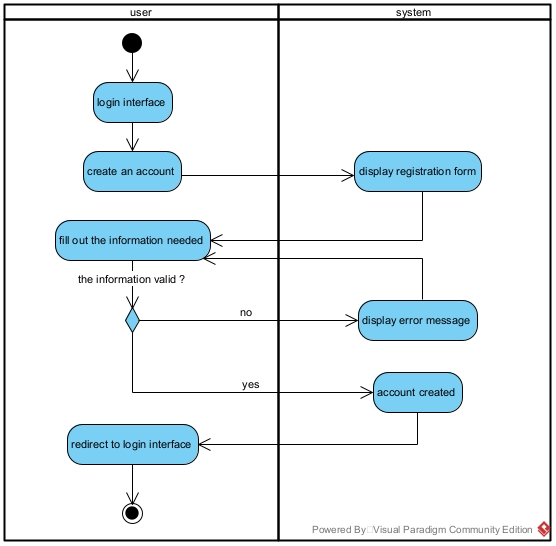


Figure 4: Sign Up – Activity Diagram

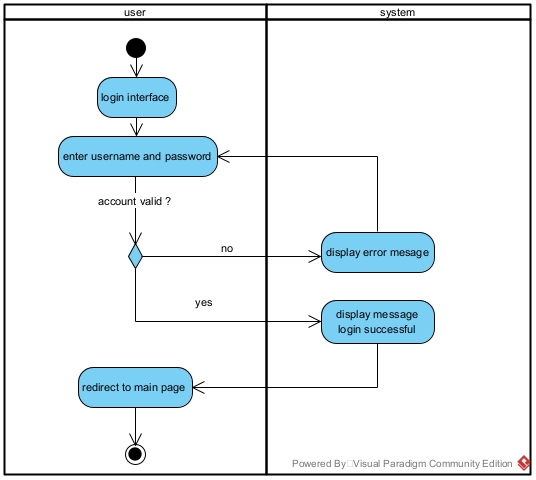


Figure 5: Sign In – Activity Diagram

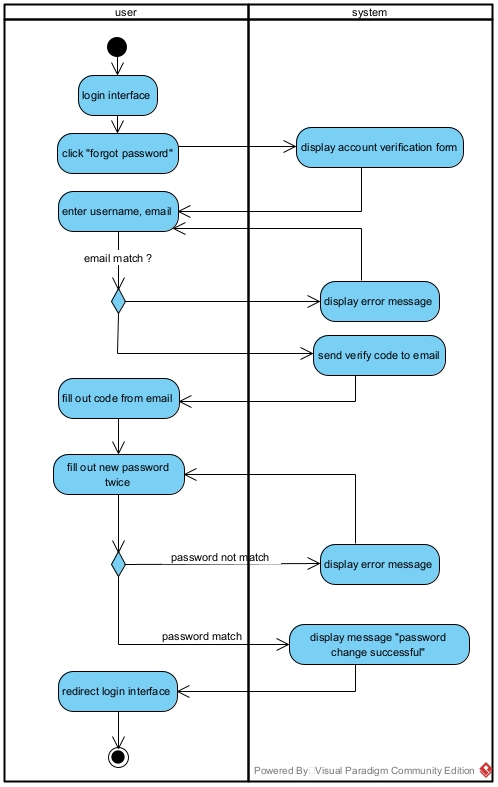


Figure 6: Forgot Password – Activity Diagram

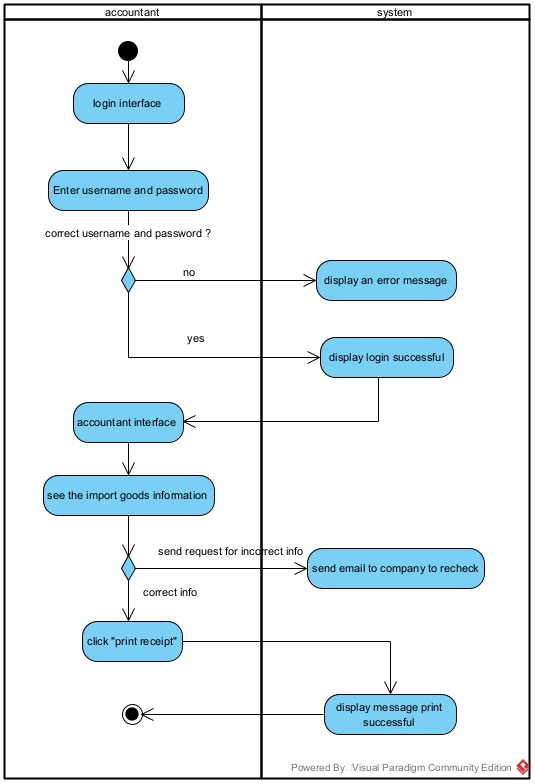


Figure 7: Create Import Receipt – Activity Diagram

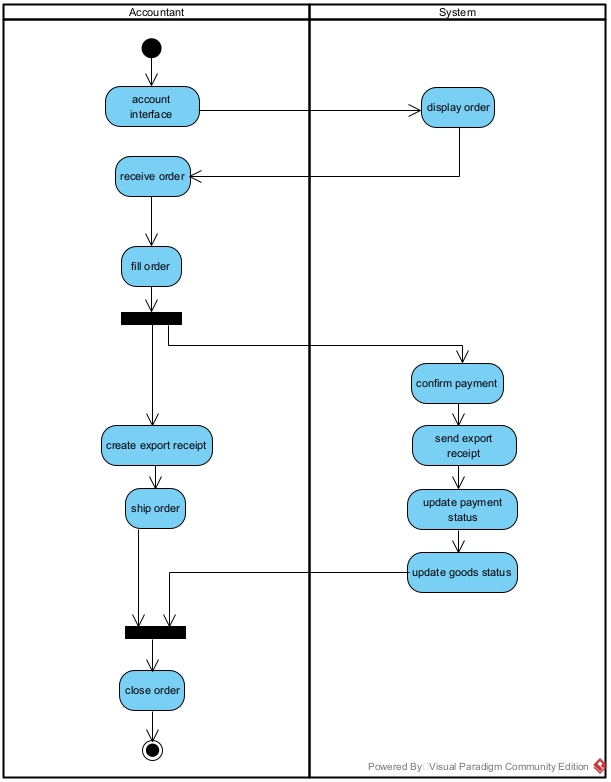


Figure 8: Manage Status – Activity Diagram

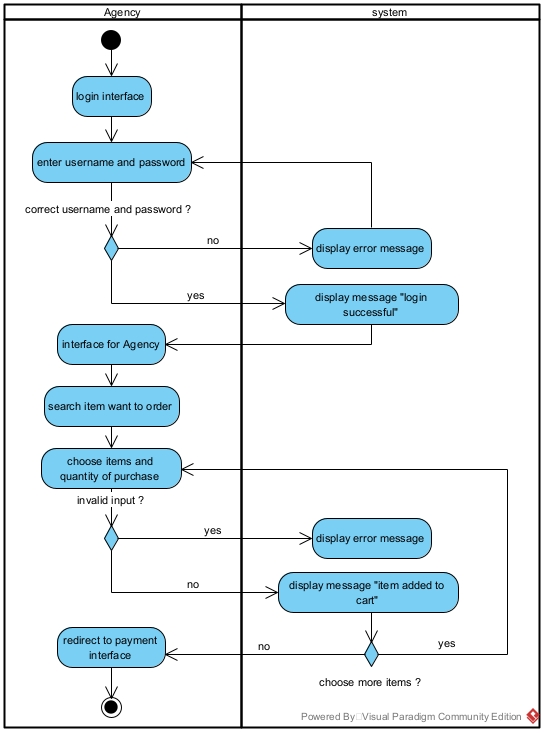


Figure 9: Place an order – Activity Diagram

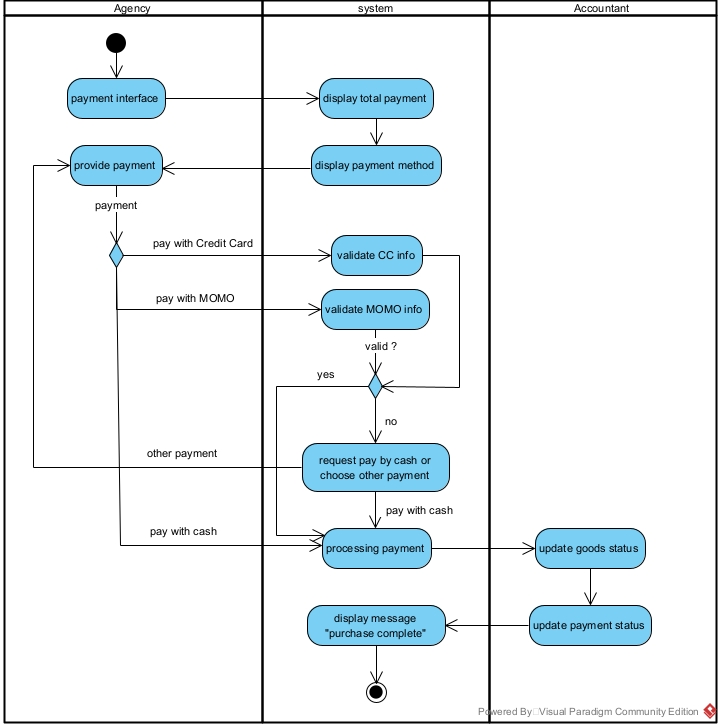


Figure 10: Make payment – Activity Diagram

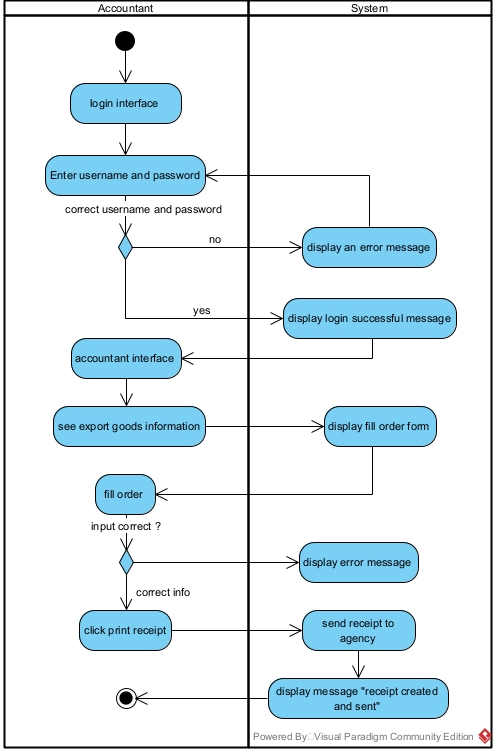


Figure 11: Create Export Receipt – Activity Diagram

5.3 Dynamic model – sequence diagrams

- Here are our sequence diagrams:

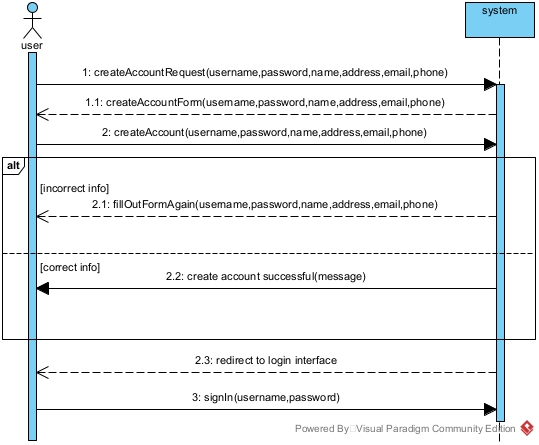


Figure 12: Sign up – System Sequence Diagram

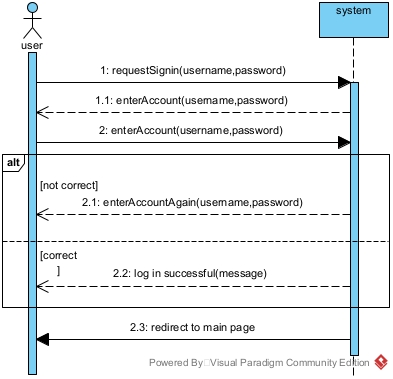


Figure 13: Sign in – System Sequence Diagram

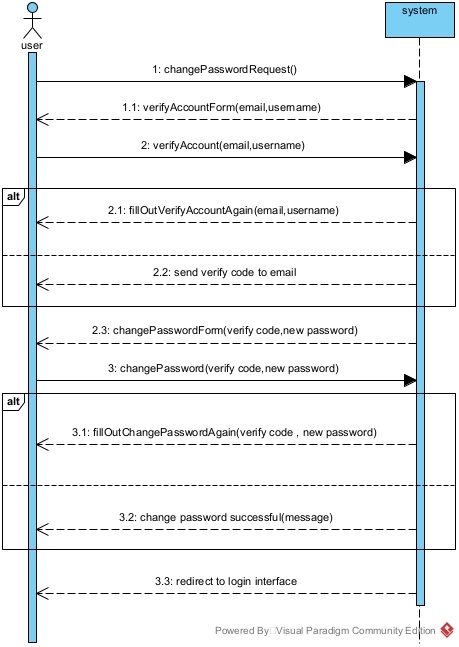


Figure 14: Forgot Password – System Sequence Diagram

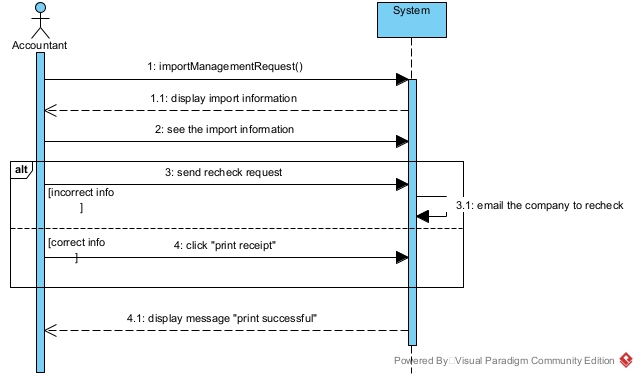


Figure 15: Create import receipt – System Sequence Diagram

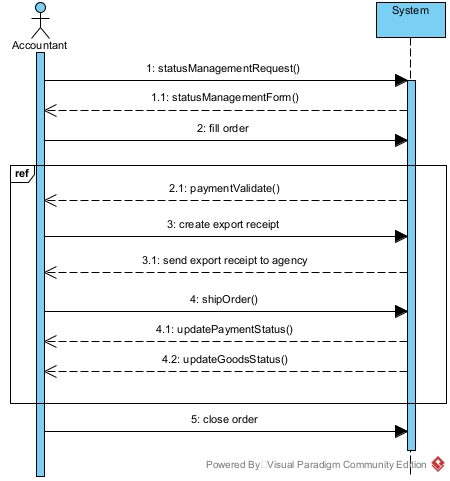


Figure 16: Manage Status – System Sequence Diagram

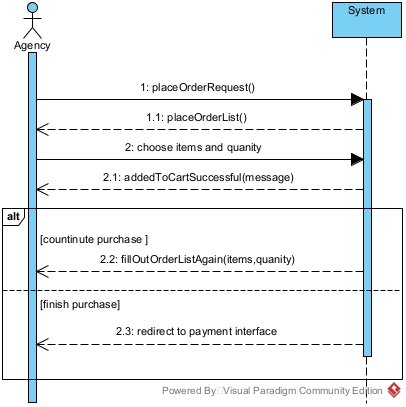


Figure 17: Place an order – System Sequence Diagram

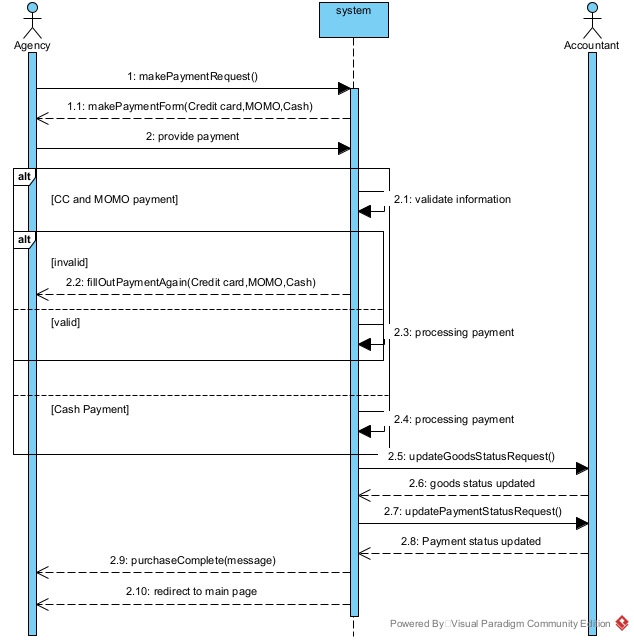


Figure 18: Make payment – System Sequence Diagram

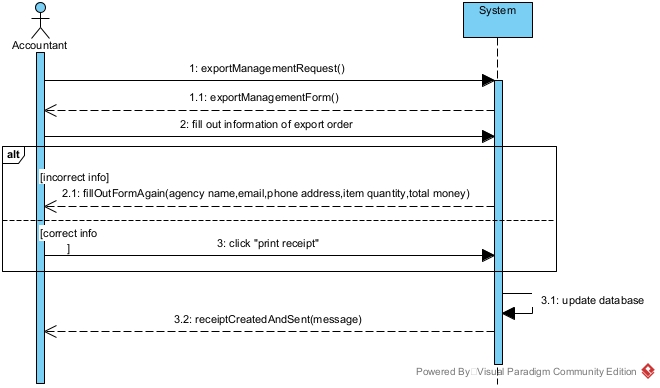


Figure 19: Create Export Receipt – System Sequence Diagram

5.4 Rationale for your detailed design model

Some text here…

5.5 Traceability from requirements to detailed design model

Some text here…

CHAPTER 6: DEMO

6.1 Database

- Some text here…

6.2 Source code

- some text here…