VIETNAM GENERAL CONFEDERATION OF LABOUR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL REPORT**

**SOFTWARE ENGINEERING**

**DANG TAN Co,.Ltd**

**RESOURCE TRACKING SOFTWARE**

**FOR FUNCTIONAL FOOD COMPANY**

*Instructor:* **Mr. PHẠM THÁI KỲ TRUNG**

*Implementer:Lê Minh Đăng* **–** *SID:520K0108*

*Cao Khánh Tân***–** *SID:520K0342*

*Group:2*

**HO CHI MINH CITY, 2022**

# ACKNOWLEDGEMENTS

A completed study would not be done without any assistance. Therefore, the author who conducted this research gratefully gives acknowledgement to their support and motivation during the time of doing this report as a requirement of completing my program of Software Engineering.

I would like to express my endless thanks and gratefulness to my supervisor Phạm Thái Kỳ Trung. His kindly support and continuous advices went through the process of completion of my thesis. His encouragement and comments had significantly enriched and improved my work. Without his motivation and instructions, the thesis would have been impossible to be done effectively.

**THE REPORT IS COMPLETED**

**AT TON DUC THANG UNIVERSITY**

I assure you that this is my own essay product and under the guidance of Mr. Phạm Thái Kỳ Trung. The research contents and results in this topic are honest and have not been published in any previous forms. The figures in the tables serving the analysis, comments and assessments were collected by the author himself from different sources specified in the references.

**If any fraud is found, I am fully responsible for the content of my report.** Ton Duc Thang University is not related to copyright and copyright infringement caused by me during the implementation process (if any).

*Ho Chi Minh city, date: / /2022*

*Author*

*(Sign and state full name)*

*Lê Minh Đăng*

*Cao Khánh Tân*

# PARTY CERTIFICATION AND ASSESSMENT OF THE TEACHERS

**The certification part of the instructor**

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

Ho Chi Minh city, date: / /2022

(sign)

**The evaluation part of the teacher rated**

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

Ho Chi Minh city, date: / /2022

(sign)

# EXECUTIVE SUMMARY

A cosmetic company needs to make a resource tracking software with main functions such as creating receipts for warehousing, ex-warehousing, statistics and the function of processing orders from agents. From this requirement, we provide a tracking software called Dang Tan Co,.Ltd.

This report provides a comprehensive architectural overview of Dang Tan Co,.Ltd, using a number of different architectural views to depict different aspects of the application. the report serves as a bridge between the software requirements and the detail design of Dang Tan Co,.Ltd.

# TABLE OF CONTENTS

[ACKNOWLEDGEMENTS](#_heading=h.30j0zll) **1**

[PARTY CERTIFICATION AND ASSESSMENT OF THE TEACHERS](#_heading=h.1fob9te) **3**

[EXECUTIVE SUMMARY](#_heading=h.3znysh7) **1**

[TABLE OF CONTENTS](#_heading=h.2et92p0) **2**

[**LIST OF FIGURES**](#_heading=h.vraadg4feum5) **3**

[LIST OF TABLES](#_heading=h.tyjcwt) **5**

[1. INTRODUCTION](#_heading=h.3dy6vkm) 6

[1.1. Purpose and Scope](#_heading=h.1t3h5sf) 6

[1.1.1. Purpose](#_heading=h.4d34og8) 6

[1.1.2. Scope](#_heading=h.2s8eyo1) 6

[1.2. Product Overview](#_heading=h.17dp8vu) 6

[1.3. Structure of the Document](#_heading=h.3rdcrjn) 7

[1.4. Terms, Acronyms, and Abbreviations](#_heading=h.26in1rg) 7

[2. PROJECT MANAGEMENT PLAN](#_heading=h.35nkun2) 8

[2.1. Project Organization](#_heading=h.1ksv4uv) 8

[2.2. Lifecycle Model Used:](#_heading=h.44sinio) 8

[Waterfall model](#_heading=h.2jxsxqh) 8

[2.3. Risk Analysis](#_heading=h.3j2qqm3) 9

[2.4. Hardware and Software Resource Requirements](#_heading=h.4i7ojhp) 9

[2.5. Deliverables and Schedule](#_heading=h.2xcytpi) 10

[2.6. Professional Standards](#_heading=h.1ci93xb) 11

[3. REQUIREMENT SPECIFICATIONS](#_heading=h.3whwml4) 12

[2.1. List of requirements](#_heading=h.2bn6wsx) 12

[2.2. Use case diagram](#_heading=h.3as4poj) 13

[2.3. Use case model](#_heading=h.49x2ik5) 14

[2.3.1. Use Case: Create Good Received Note](#_heading=h.2p2csry) 14

[2.3.2. Use Case: Create Good Delivery Note](#_heading=h.ihv636) 17

[2.3.3. Use Case: Order Goods](#_heading=h.3fwokq0) 20

[2.3.4. Use Case: Make Statistics](#_heading=h.28h4qwu) 23

[2.3.5. Use Case: Login](#_heading=h.111kx3o) 26

[2.3.6. Use Case: Change Password](#_heading=h.3ygebqi) 29

[4. ARCHITECTURE](#_heading=h.3q5sasy) 34

[**Architectural model: MVC**](#_heading=h.25b2l0r) **34**

[5. DESIGN](#_heading=h.34g0dwd) 35

[5.1. Database design](#_heading=h.1jlao46) 35

[5.2. Static model – class diagrams](#_heading=h.2iq8gzs) 36

[5.3. Dynamic model – sequence diagrams](#_heading=h.3hv69ve) 37

[6. TEST PLAN](#_heading=h.4h042r0) 38

[6.1. Requirements/specifications-based system level test cases](#_heading=h.2w5ecyt) 38

[6.2. Traceability of test cases to use cases](#_heading=h.1baon6m) 38

[6.3. Techniques used for test generation](#_heading=h.3vac5uf) 38

[6.4. Assessment of the goodness of your test suite](#_heading=h.2afmg28) 38

[7. DEMO](#_heading=h.pkwqa1) 39

[REFERENCES](#_heading=h.2fk6b3p) **49**

[**APPENDIX**](#_heading=h.upglbi) **50**

# **LIST OF FIGURES**

[Figure 2. 1. Waterfall model 9](#_heading=h.z337ya)

[Figure 3. 1. Use case diagram 13](#_heading=h.1pxezwc)

[Figure 3. 2. Activity diagram for Create Good Received Note 14](#_heading=h.147n2zr)

[Figure 3. 3. System sequence diagram for Create Good Received Note 15](#_heading=h.3o7alnk)

[Figure 3. 4. Activity diagram for Create Good Delivery Note 17](#_heading=h.1hmsyys)

[Figure 3. 5. System sequence for Create Good Delivery Note 18](#_heading=h.2grqrue)

[Figure 3. 6. Activity diagram for Order Goods 20](#_heading=h.4f1mdlm)

[Figure 3. 7. System sequence diagram for Order Goods 21](#_heading=h.19c6y18)

[Figure 3. 8. Activity diagram for Make Statistics 23](#_heading=h.37m2jsg)

[Figure 3. 9. System sequence diagram for Make Statistics 24](#_heading=h.46r0co2)

[Figure 3. 10. Activity diagram for Login 26](#_heading=h.206ipza)

[Figure 3. 11. System Sequence diagram for Login 27](#_heading=h.2zbgiuw)

[Figure 3. 12. Activity diagram for Change Password 29](#_heading=h.sqyw64)

[Figure 3. 13. System sequence diagram for Change Password 30](#_heading=h.1rvwp1q)

[Figure 4. 1. MVC model 34](#_heading=h.kgcv8k)

[Figure 5. 1. Database design 35](#_heading=h.43ky6rz)

[Figure 5. 2. Class diagram 36](#_heading=h.xvir7l)

[Figure 5. 3. Sequence diagram 37](#_heading=h.1x0gk37)

[Figure 7. 1. Login interface 39](#_heading=h.1opuj5n)

[Figure 7. 2. Admin (manage users) interface 40](#_heading=h.48pi1tg)

[Figure 7. 3. Main interface 41](#_heading=h.2nusc19)

[Figure 7. 4. Main interface (2) 42](#_heading=h.1302m92)

[Figure 7. 5. Good Delivery Note interface 43](#_heading=h.3mzq4wv)

[Figure 7. 6. Print Delivery Note 44](#_heading=h.2250f4o)

[Figure 7. 7.Order Management interface 45](#_heading=h.haapch)

[Figure 7. 8. Statistics interface 46](#_heading=h.319y80a)

[Figure 7. 9. Edit profile interface 47](#_heading=h.1gf8i83)

[Figure 7. 10. Change password interface 48](#_heading=h.40ew0vw)

# LIST OF TABLES

[Table 1. 1. Terms, Acronyms and Abbreviations 7](#_heading=h.lnxbz9)

[Table 2. 1. Risk analysis 10](#_heading=h.1y810tw)

[Table 3. 1. List of Requirements 12](#_heading=h.qsh70q)

[Table 3. 2. Use case description for Create Good Received Note 16](#_heading=h.23ckvvd)

[Table 3. 3. Use Case Description for Create Goods Delivery Note 19](#_heading=h.vx1227)

[Table 3. 4. Use Case Description for Order Goods 22](#_heading=h.3tbugp1)

[Table 3. 5. Use Case Description for Make Statistics 25](#_heading=h.2lwamvv)

[Table 3. 6. Use case description for Login 28](#_heading=h.1egqt2p)

[Table 3. 7. Use case description for Change Password 31](#_heading=h.4bvk7pj)

[Table 3. 8. Use case description for Manage Users 33](#_heading=h.1664s55)

## 1. INTRODUCTION

### Purpose and Scope

#### Purpose

This report goes into great detail about the software development process. It provides a project management plan to determine the project outcome, how a successful outcome will be achieved, who will be involved in the project, and how the project will be measured and communicated.

The document presents a complete architectural overview of the new Dang Tan Co,.Ltd software, depicting different elements of the software utilizing a variety of architectural viewpoints. The document serves as a bridge between the software requirements and the detail design of Dang Tan Co,.Ltd, it will also assist software architects in ensuring that the modules to be built will meet the needs of the users in terms of functionality (selected functional view), platform, and technology (logical views).

#### Scope

The document outlines the software architecture that fulfills the Dang Tan Co,.Ltd requirements in the areas of functionality, availability, reliability, scalability, maintainability, and management at a high level.

The introduction of n-tier architecture, which is reflected in the diagrams of the proposed structure, better describes the Dang Tan Co,.Ltd from a logical standpoint.

### Product Overview

Dang Tan Co,.Ltd Software with the following features is required by a corporation selling cosmetics to agents:

- When a corporation imports products, accountants must be able to establish Goods Received (a warehouse receipt will include many items).

- Agents will be able to make a purchase order and select a payment method (Cash, bank transfer, Momo...) (Web).

- Accountants will be able to produce Products Delivery Notes to send goods to agents (print delivery slips), change order status as being transferred, and update agent payment status.

- Accountants will be able to check the monthly incoming/outgoing stock report as well as the revenue report.

### Structure of the Document

The document is divided into six major sections as follows:

- Project management plan.

- Requirement specifications.

- Architecture.

- Design.

- Test plan.

- Demo.

### Terms, Acronyms, and Abbreviations

| # | Item | Description |
| --- | --- | --- |
| 1 | GRN | Good Received Note |
| 2 | GDN | Good Delivery Note |
| 3 | MVC | Model – View – Control |

*Table 1. 1. Terms, Acronyms and Abbreviations*

## PROJECT MANAGEMENT PLAN

### Project Organization

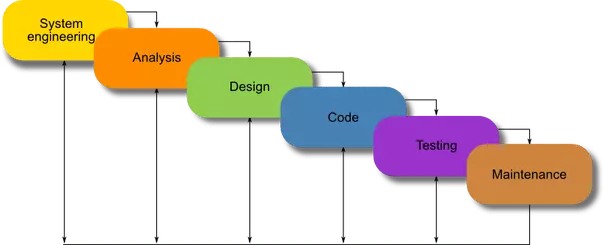
Project table time :

* Day 1: Wednesday 26th, 2021
* Day 2: Friday 28th, 2021
* Day 3: Monday 31st, 2021
* Day 4: Wednesday 2nd, 2022
* Day 5: Friday 4th, 2022

Each meeting day (online) is up to 1 hour.

### Lifecycle Model Used:

#### **Waterfall model**

****

*Figure 2. 1. Waterfall model*

**Requirements:** During this initial phase, the potential requirements of the application are methodically analyzed and written down in a specification document that serves as the basis for all future development. The result is typically a requirements document that defines what the application should do, but not how it should do it.

**Analysis:** During this second stage, the system is analyzed in order to properly generate the models and business logic that will be used in the application.

**Design:** This stage largely covers technical design requirements, such as programming language, data layers, services, etc. A design specification will typically be created that outlines how exactly the business logic covered in analysis will be technically implemented.

**Coding:** The actual source code is finally written in this fourth stage, implementing all models, business logic, and service integrations that were specified in the prior stages.

**Testing:** During this stage, QA, beta testers, and all other testers systematically discover and report issues within the application that need to be resolved. It is not uncommon for this phase to cause a “necessary repeat” of the previous coding phase, in order for revealed bugs to be properly squashed.

**Operations:** Finally, the application is ready for deployment to a live environment. The operations stage entails not just the deployment of the application, but also subsequent support and maintenance that may be required to keep it functional and up-to-date.

### Risk Analysis

| Risk item | Risk management techniques |
| --- | --- |
| Personal shortfall | Staffing with top talent, job matching, team building, key personnel agreements, training, prescheduling key people. |
| Developing wrong software functions | Organization analysis, machine analysis, user surveys, prototyping, early user’s manual. |
| Developing wrong software interface | Prototyping, scenarios, task analysis, user characterization. |

*Table 2. 1. Risk analysis*

### Hardware and Software Resource Requirements

Hardware system requirements:

* Minimum CPU or processes speed.
* Minimum GPU or video memory.
* Minimum system memory (RAM).
* Storage space need.

Software requirements:

* Operating System.
* Internet connection.

### Deliverables and Schedule

Schedule of completion of each item:

* Winform completes on.
* Main function completes on.
* Webform completes on.
* Report completes on.
* Delivery time in on January 6th, 2022.

### Professional Standards

**PUBLIC** – act consistently with the public interest.

**CLIENT AND EMPLOYER** – act in a manner that is in the best interests of client and employer consistent with the public interest.

**PRODUCT** – ensure that their products and related modifications meet the highest professional standards possible.

**JUDGMENT** – maintain integrity and independence in their professional judgment.

**MANAGEMENT** – managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.

**PROFESSION** – advance the integrity and reputation of the profession consistent with the public interest.

**COLLEAGUES** – be fair to and supportive of their colleagues.

**SELF** – participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

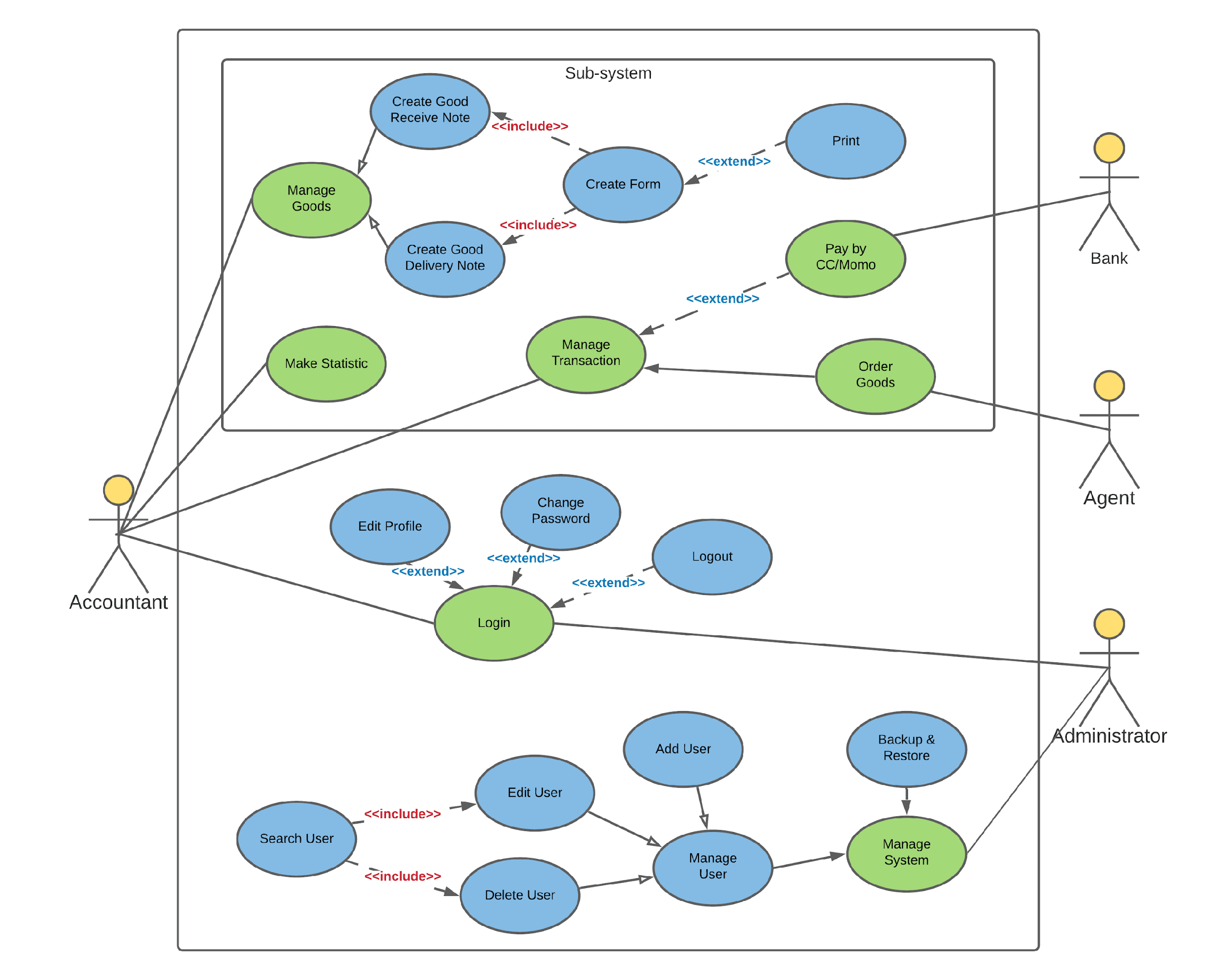
## 3. REQUIREMENT SPECIFICATIONS

### List of requirements

| Requirements | Functional | Non-functional |
| --- | --- | --- |
| The system must allow administrator to manage users (add, edit, delete). | X |  |
| The system must allow administrator to backup and restore system data. | X |  |
| Accountant must be allowed to manage goods receives and deliveries. | X |  |
| Accountant must be allowed to print delivery slips. | X |  |
| Accountant is allowed to make statistics and print them. | X |  |
| The system must allow agent to order online. | X |  |
| Agents are able to choose suitable payment method. | X |  |
| Friendly system with good theme, good colors. |  | X |
| Users can access the system at any time, except the maintenance time (to be announced in advance) |  | X |
| System must NOT store customers’ banking card information. |  | X |

*Table 3. 1. List of Requirements*

### Use case diagram

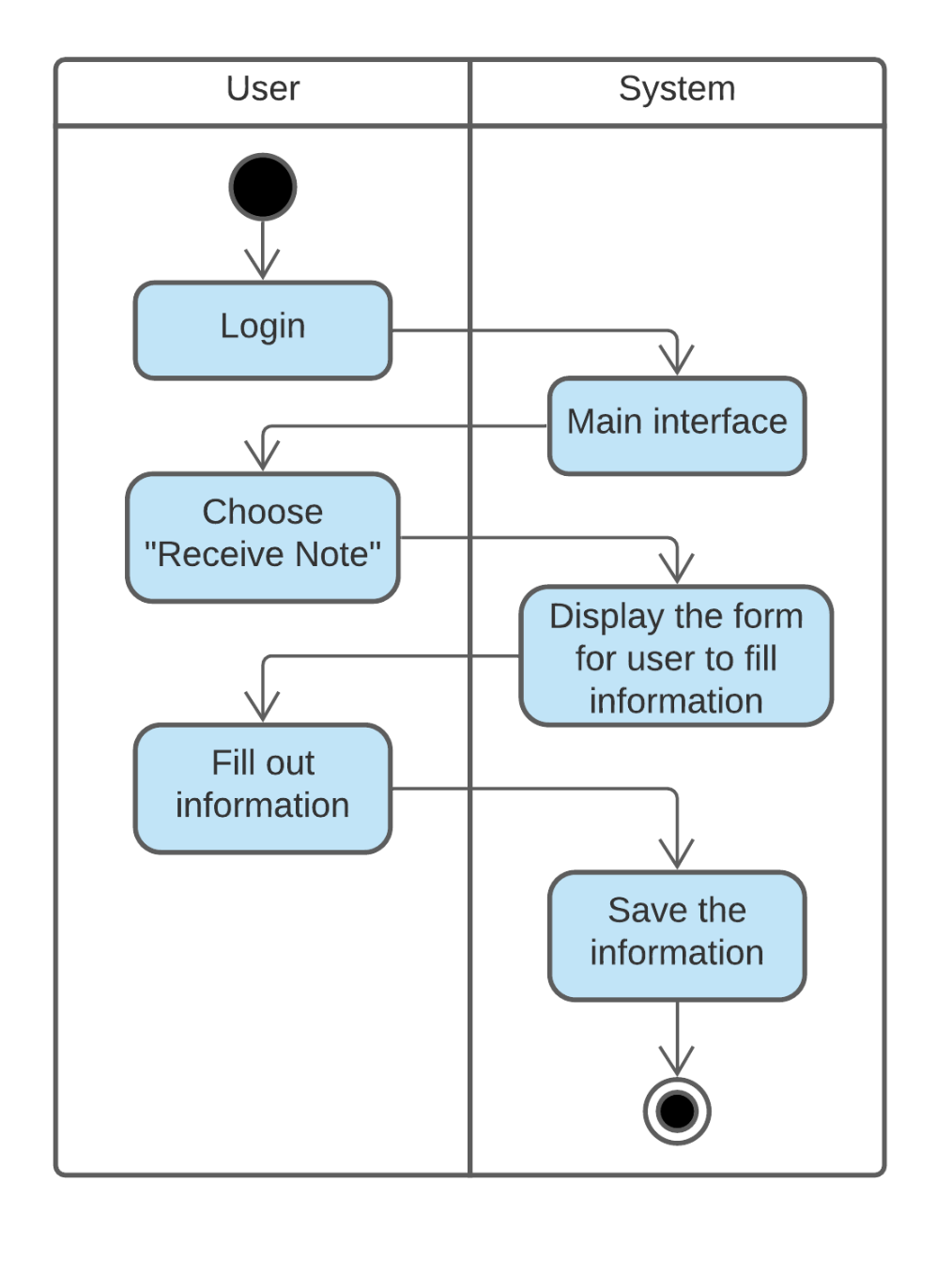
****

*Figure 3. 1. Use case diagram*

### Use case model

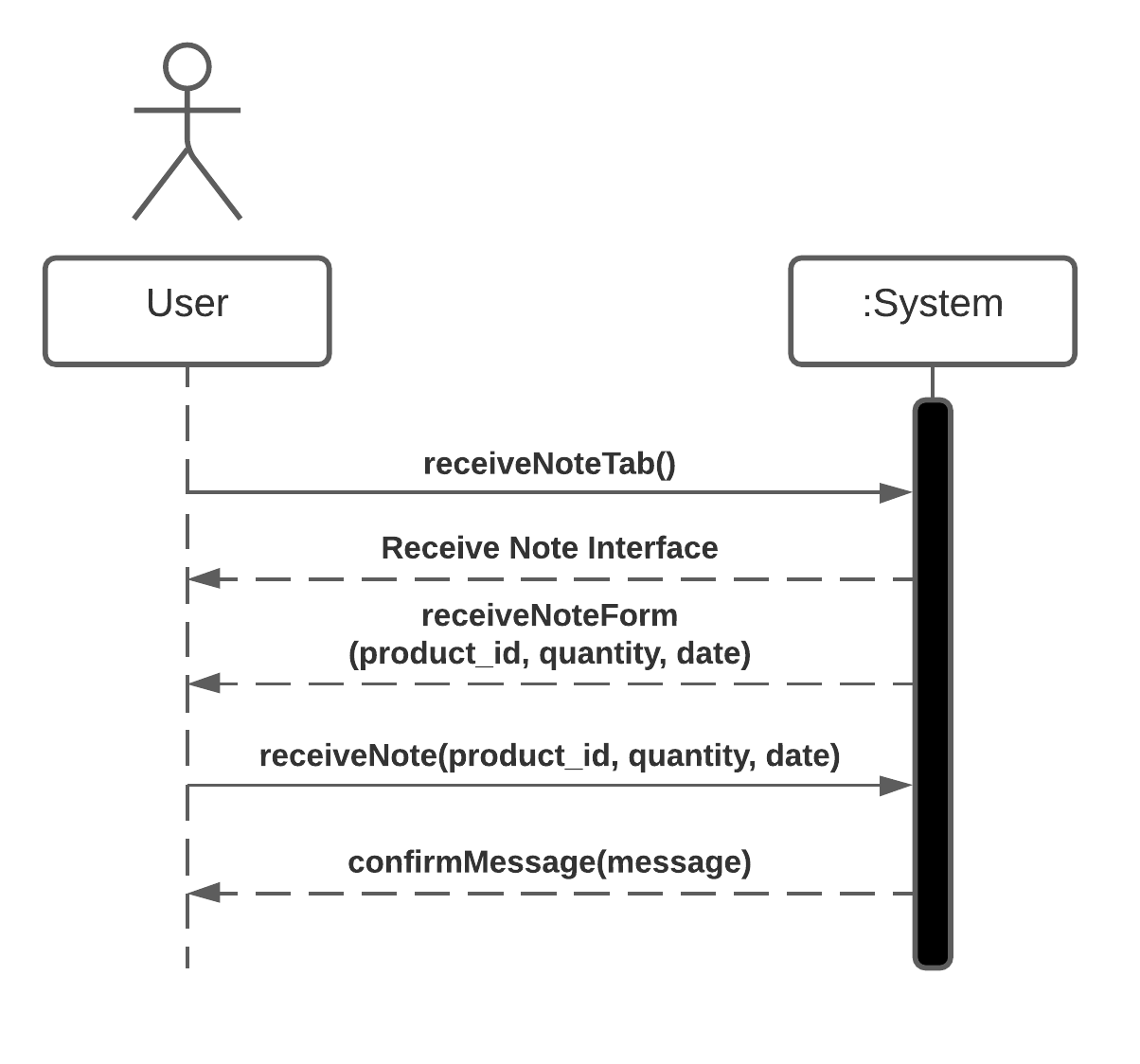
#### Use Case: Create Good Received Note

* **Graphic use case model**
  + **Activity Diagram**

****

*Figure 3. 2. Activity diagram for Create Good Received Note*

* + **System Sequence Diagram**

****

*Figure 3. 3. System sequence diagram for Create Good Received Note*

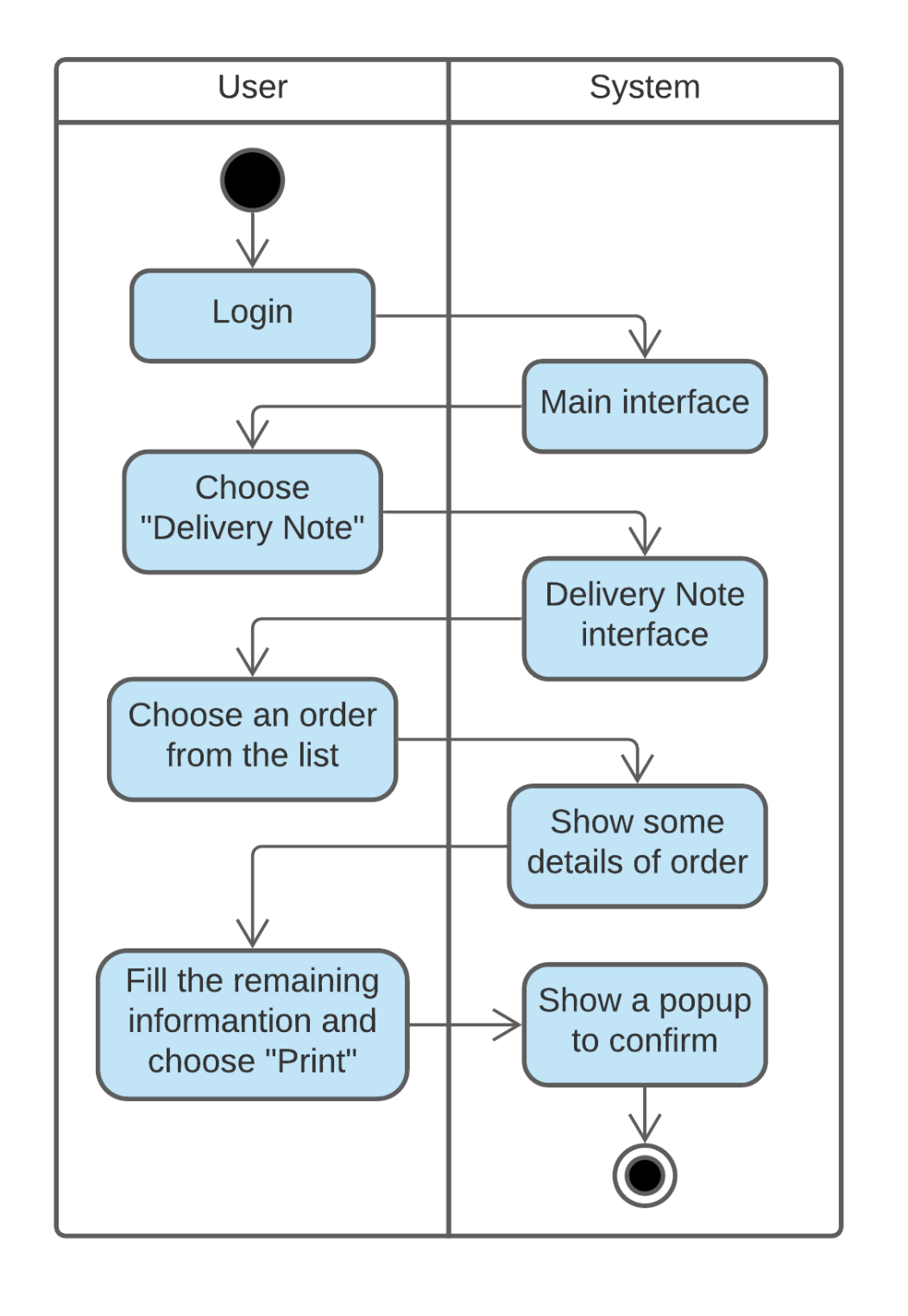
* **Use Case Description**

| Use case name: | Create Goods Received Note | |
| --- | --- | --- |
| Scenario: | Company imports goods to warehouse | |
| Triggering event: | Accountant wants to create note to store information about items | |
| Brief description: | Company imports goods, then accountant create a note to store information about items in order to manage goods easily. | |
| Actor: | Accountant | |
| Related use cases: | Login | |
| Stakeholders: | - | |
| Preconditions: | Accountant Account sub-system must be available. | |
| Post-conditions: | Accountant account, information must be created if it has not been created, and must be saved.  Information, account, login session must be associated with accountant. | |
| Flow of activities: | Actor | System |
| 1. Accountant logins to the system.  2. Accountant clicks “Received Note” tab.  3. Accountant fills the note and confirms the note. | 1.1. System shows the main screen interface.  2.1. System shows the interface for accountant to fill the information about goods and import date.  3.1. System stores the note with unique note id. |
| Exception conditions: | 1.1. Accountant fails to login because of wrong username or password.  3.1. System returns to the interface if accountant clicks cancel from the confirm popup. | |

*Table 3. 2. Use case description for Create Good Received Note*

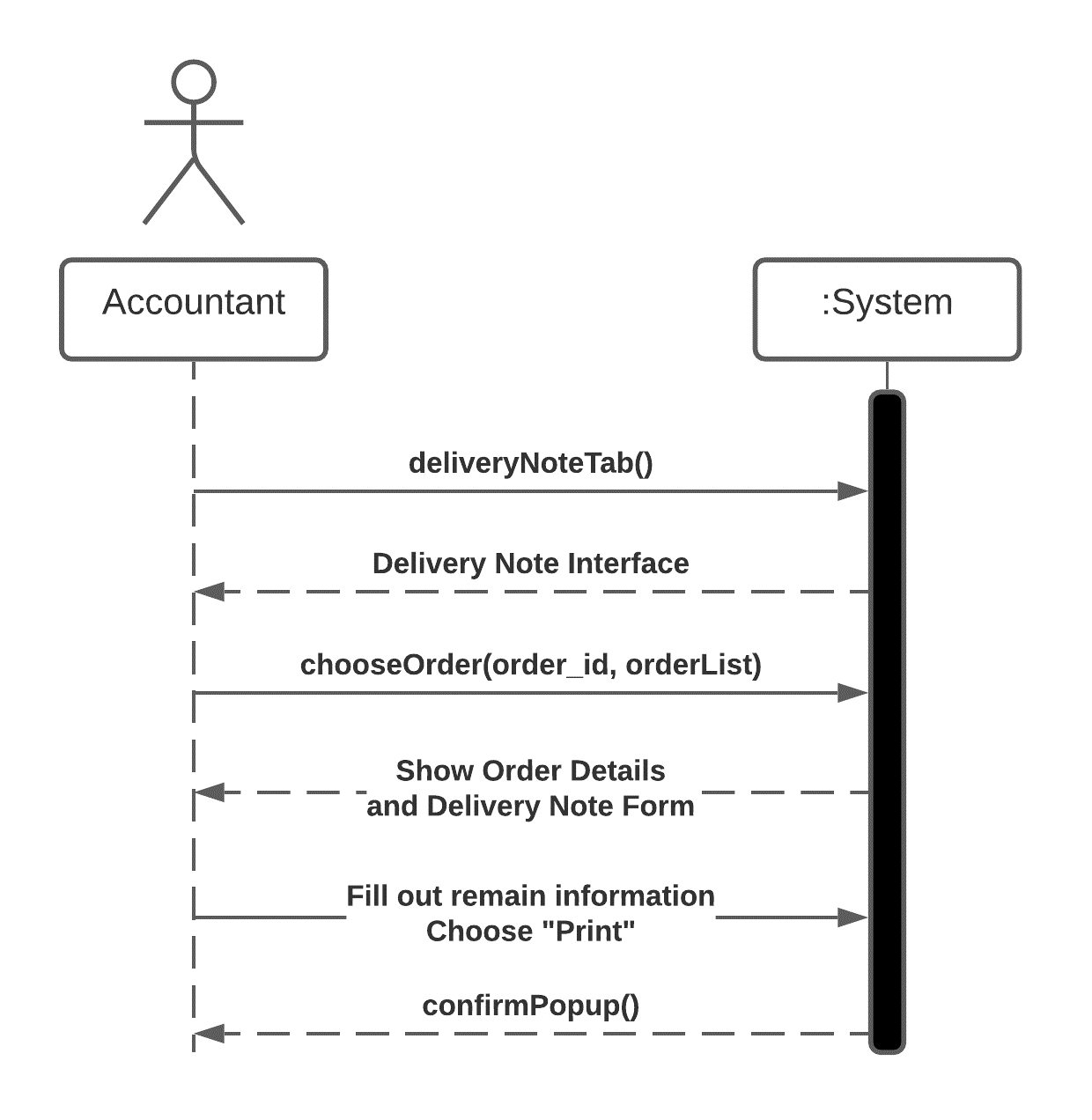
#### Use Case: Create Good Delivery Note

* **Graphic use case model**
  + **Activity Diagram**

****

*Figure 3. 4. Activity diagram for Create Good Delivery Note*

* + **System Sequence Diagram**

****

*Figure 3. 5. System sequence for Create Good Delivery Note*

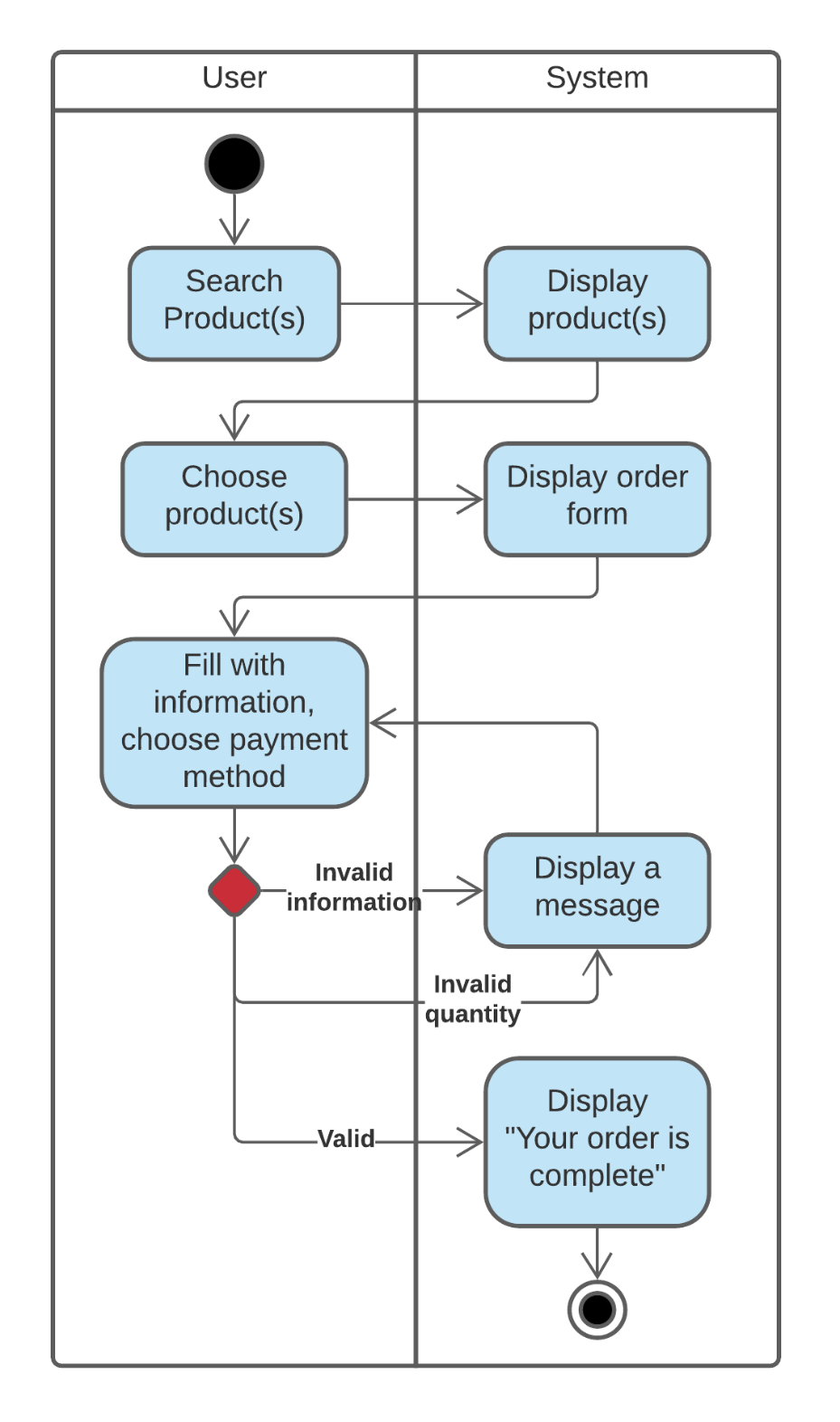
* **Use Case Description**

| Use case name: | Create Goods Delivery Note | |
| --- | --- | --- |
| Scenario: | Company deliveries goods to agent | |
| Triggering event: | Accountant wants to create delivery note of goods that will be transferred. | |
| Brief description: | After agent ordering goods, accountant creates note to store information of goods that being delivered, updates the delivery and payment status. | |
| Actor: | Accountant | |
| Related use cases: | Login, Order Goods | |
| Stakeholders: | Agents | |
| Preconditions: | Accountant Account sub-system must be available. | |
| Post-conditions: | Accountant account, information must be created if it has not been created, and must be saved.  Information, account, login session must be associated with accountant. | |
| Flow of activities: | Actor | System |
| 1. Accountant logins to the system.  2. Accountant chooses “Delivery Note” tab.  3. Accountant chooses an order form from the list in order to create a GDN.  4. Accountant fills the note, then choose “print”. | 1.1. System shows the main screen interface.  2.1. System shows the “Delivery Note” interface.  3.1. System shows details of the order form and some spaces for accountant to fill.  4.1. System displays a popup for accountant to confirm. |
| Exception conditions: | 1.1. Accountant fails to login because of wrong username or password.  4.1. System returns the previous interface if accountant does not confirm. | |

*Table 3. 3. Use Case Description for Create Goods Delivery Note*

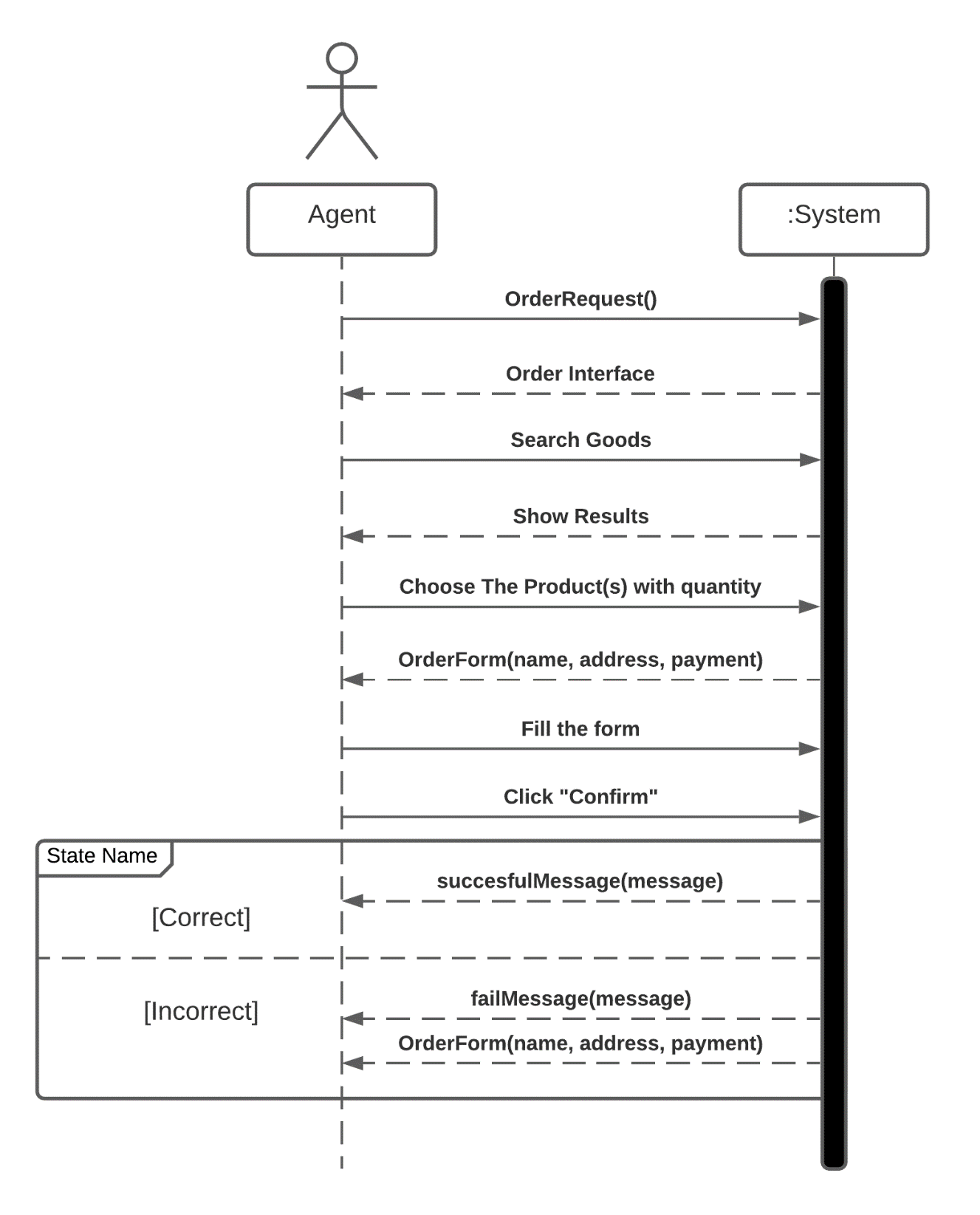
#### Use Case: Order Goods

* **Graphic use case model**
  + **Activity Diagram**

****

*Figure 3. 6. Activity diagram for Order Goods*

* + **System Sequence Diagram**

****

*Figure 3. 7. System sequence diagram for Order Goods*

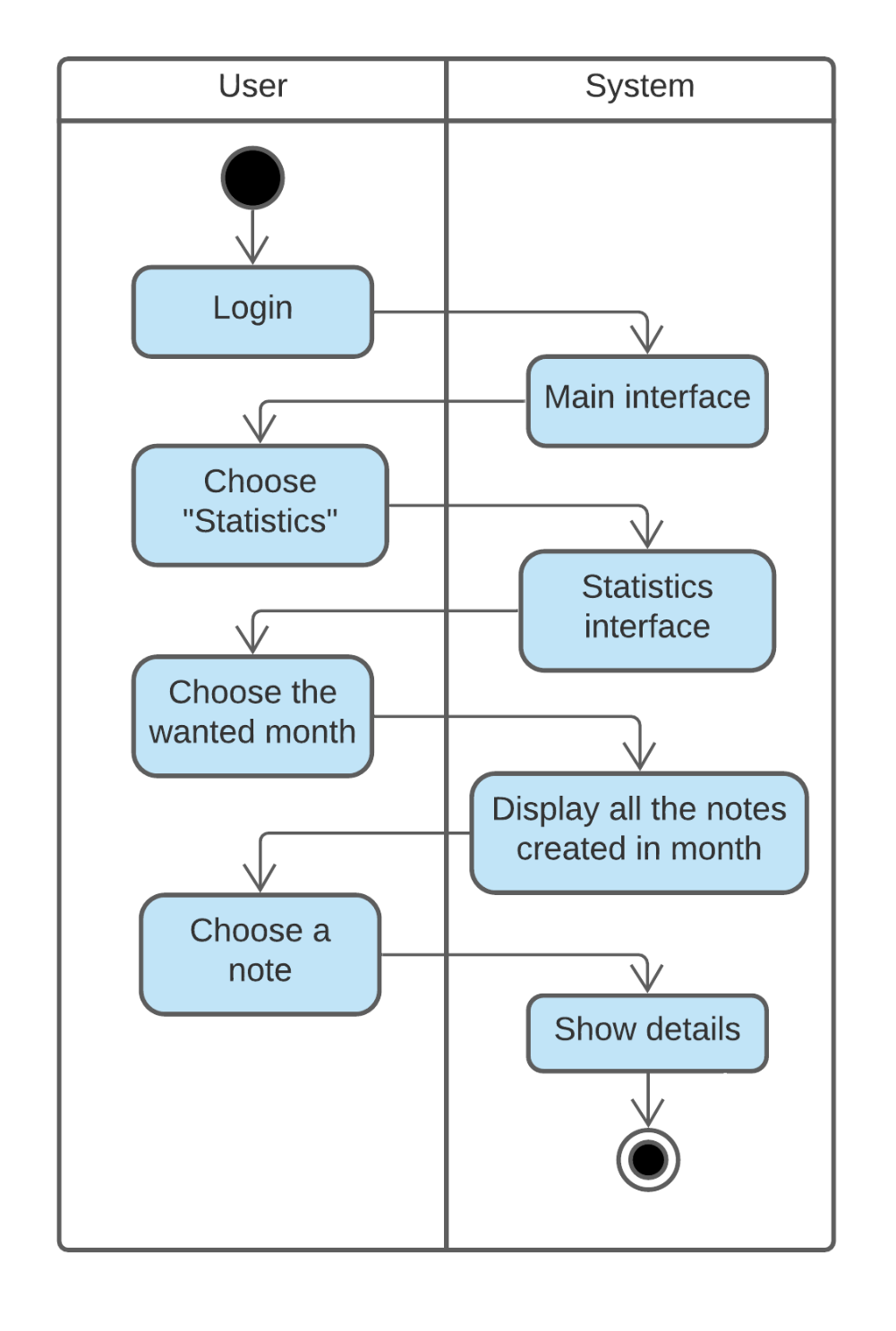
* **Use Case Description**

| Use case name: | Order Goods | |
| --- | --- | --- |
| Scenario: | Agent needs goods, supplies. | |
| Triggering event: | Agent wants to order goods from company | |
| Brief description: | Agent searches goods and choose them from website, fill the information, address(es) and choose the suitable payment method. | |
| Actor: | Agent | |
| Related use cases: | - | |
| Stakeholders: | Accountant | |
| Preconditions: | Must have goods in storage and their information is on the system. | |
| Post-conditions: | System can store their information, address(es) but not their bank account. | |
| Flow of activities: | Actor | System |
| 1. Agent searches the products.  2. Agent chooses the products.  3. Agent fills the form with address, information, quantity of products and payment method. | 1.1. System displays the products.  2.1. System displays the order form.  3.1. System checks if the information is correct.  3.2. System checks if the quantity is valid at the time the agent ordering.  3.3. System makes associative connection between agent and the order.  3.4. System shows the notification if the order is complete. |
| Exception conditions: | 3. System returns to the previous interface when agent does not confirm the form.  3.1. System shows a popup to notify agent to change the information due to incorrection.  3.2. System shows a popup to notify agent to change the quantity due to the number of products in storage. | |

*Table 3. 4. Use Case Description for Order Goods*

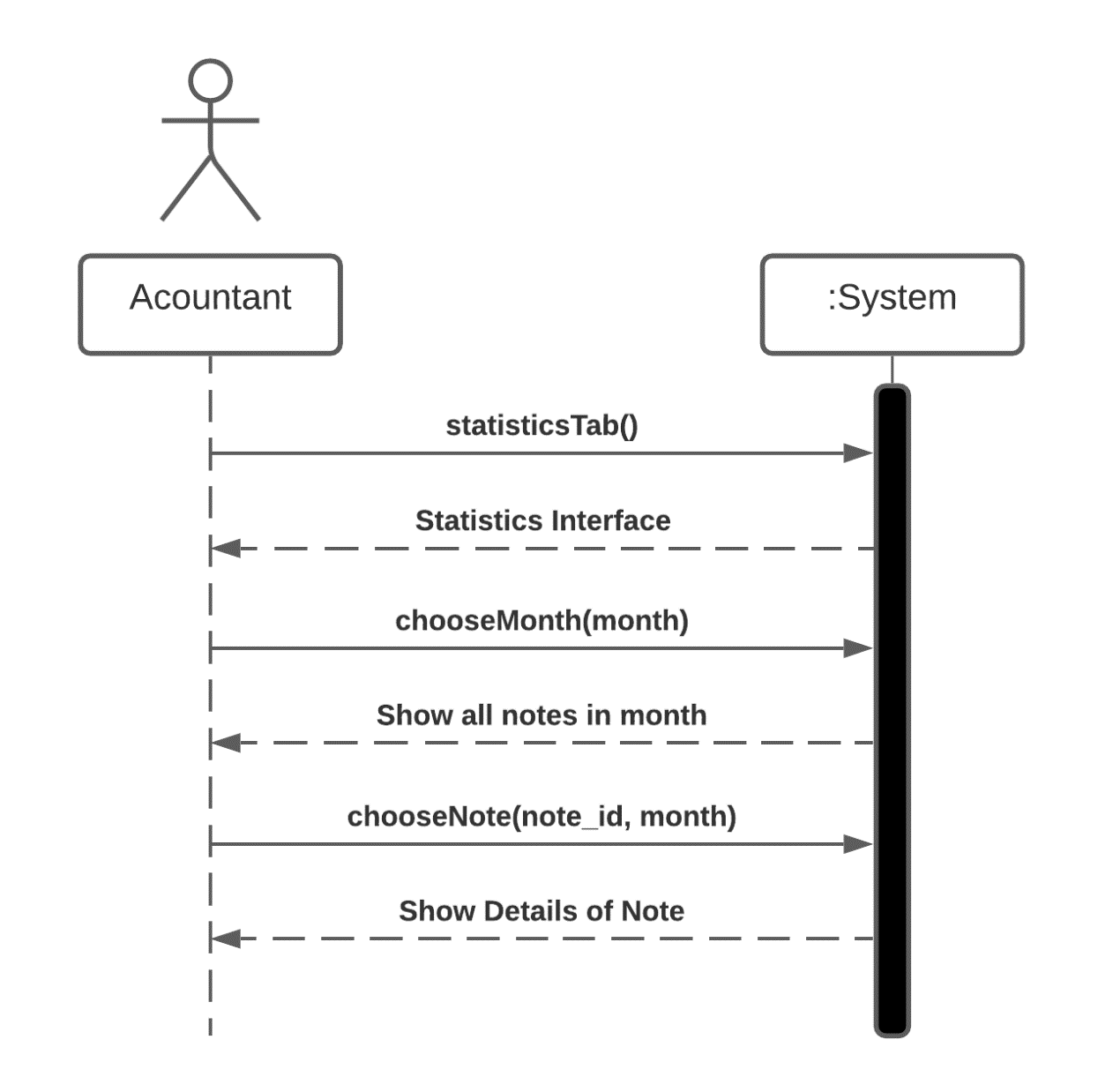
#### Use Case: Make Statistics

* **Graphic use case model**
  + **Activity Diagram**

****

*Figure 3. 8. Activity diagram for Make Statistics*

* + **System Sequence Diagram**

****

*Figure 3. 9. System sequence diagram for Make Statistics*

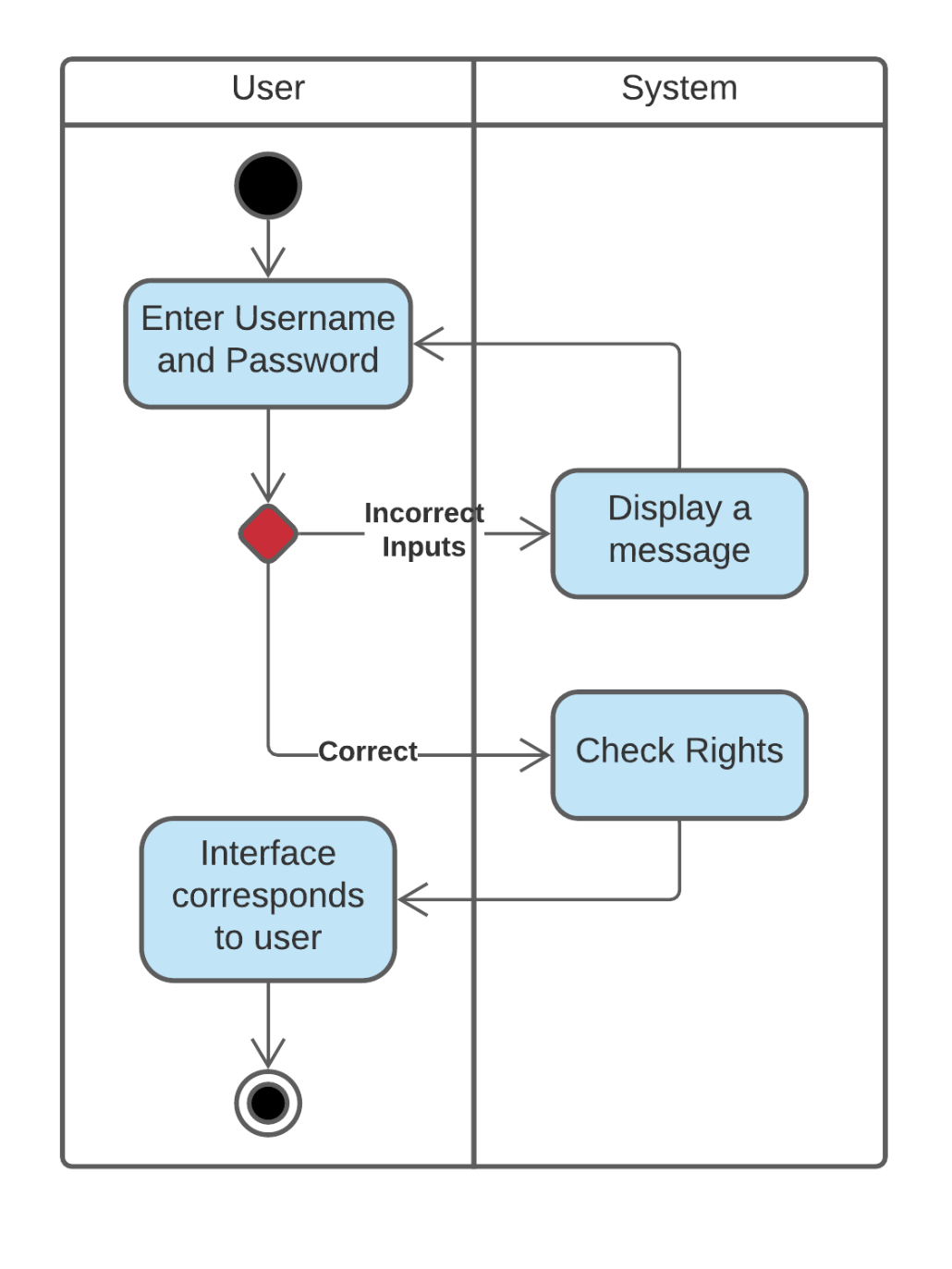
* **Use Case Description**

| Use case name: | Make statistic | |
| --- | --- | --- |
| Scenario: | Make a monthly summary | |
| Triggering event: | Accountant wants to make statistics. | |
| Brief description: | Accountant wants to make a monthly summary in order to manage received/delivered goods in the storage. | |
| Actor: | Accountant | |
| Related use cases: | Login | |
| Stakeholders: | - | |
| Preconditions: | Accountant Account sub-system must be available. | |
| Post-conditions: | Accountant account, information must be created if it has not been created, and must be saved.  Information, account, login session must be associated with accountant. | |
| Flow of activities: | Actor | System |
| 1. Accountant logins to the system.  2. Accountant clicks “Statistics” tab.  3. Accountant chooses a month.  4. Accountant choose a note. | 1.1. System shows the main screen interface.  2.1. System displays the “Statistic” interface.  3.1. System displays all the notes that being created and the quantity of goods that being transferred in month.  4.1. System displays the details of the chosen note. |
| Exception conditions: | 1.1. Accountant fails to login due to wrong username or password. | |

*Table 3. 5. Use Case Description for Make Statistics*

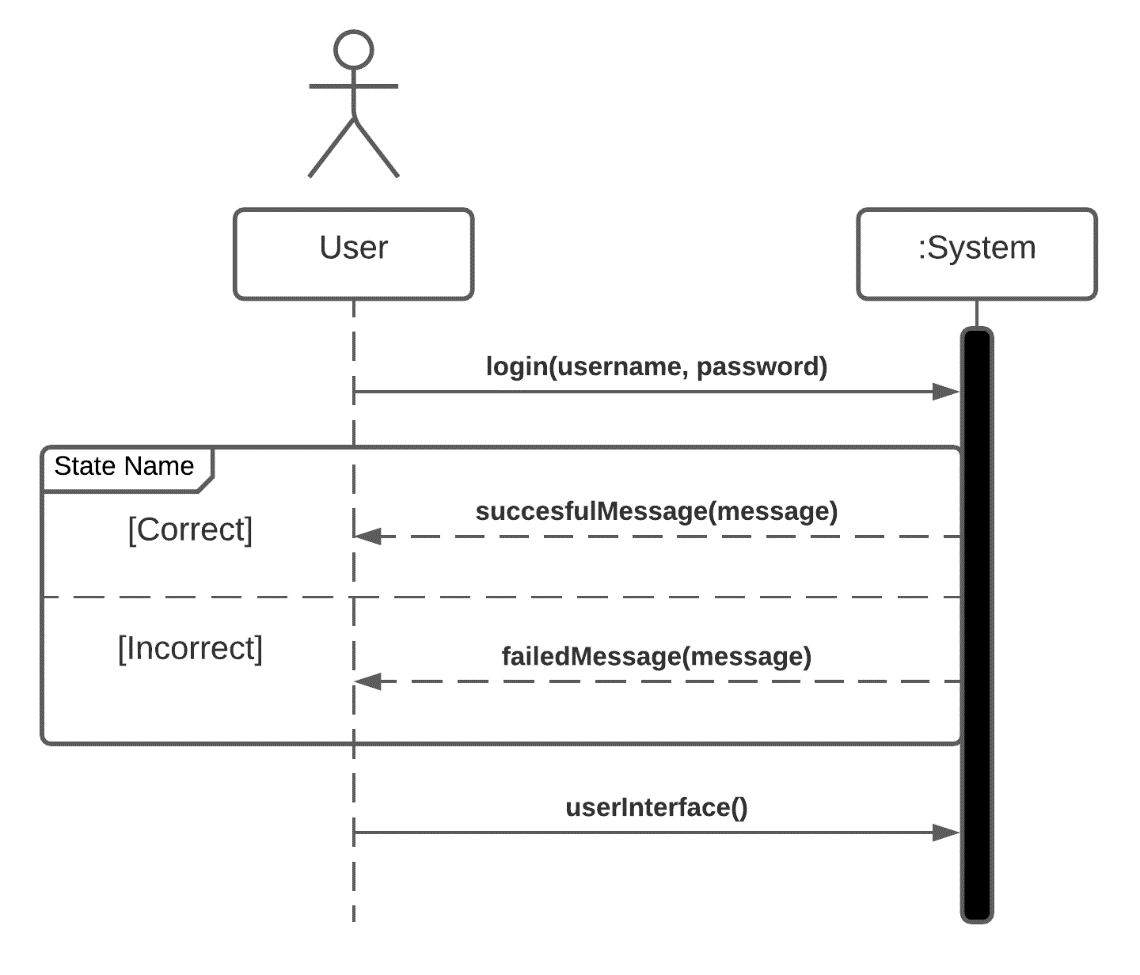
#### Use Case: Login

* **Graphic use case model**
  + **Activity Diagram**

****

*Figure 3. 10. Activity diagram for Login*

* + **System Sequence Diagram**

****

*Figure 3. 11. System Sequence diagram for Login*

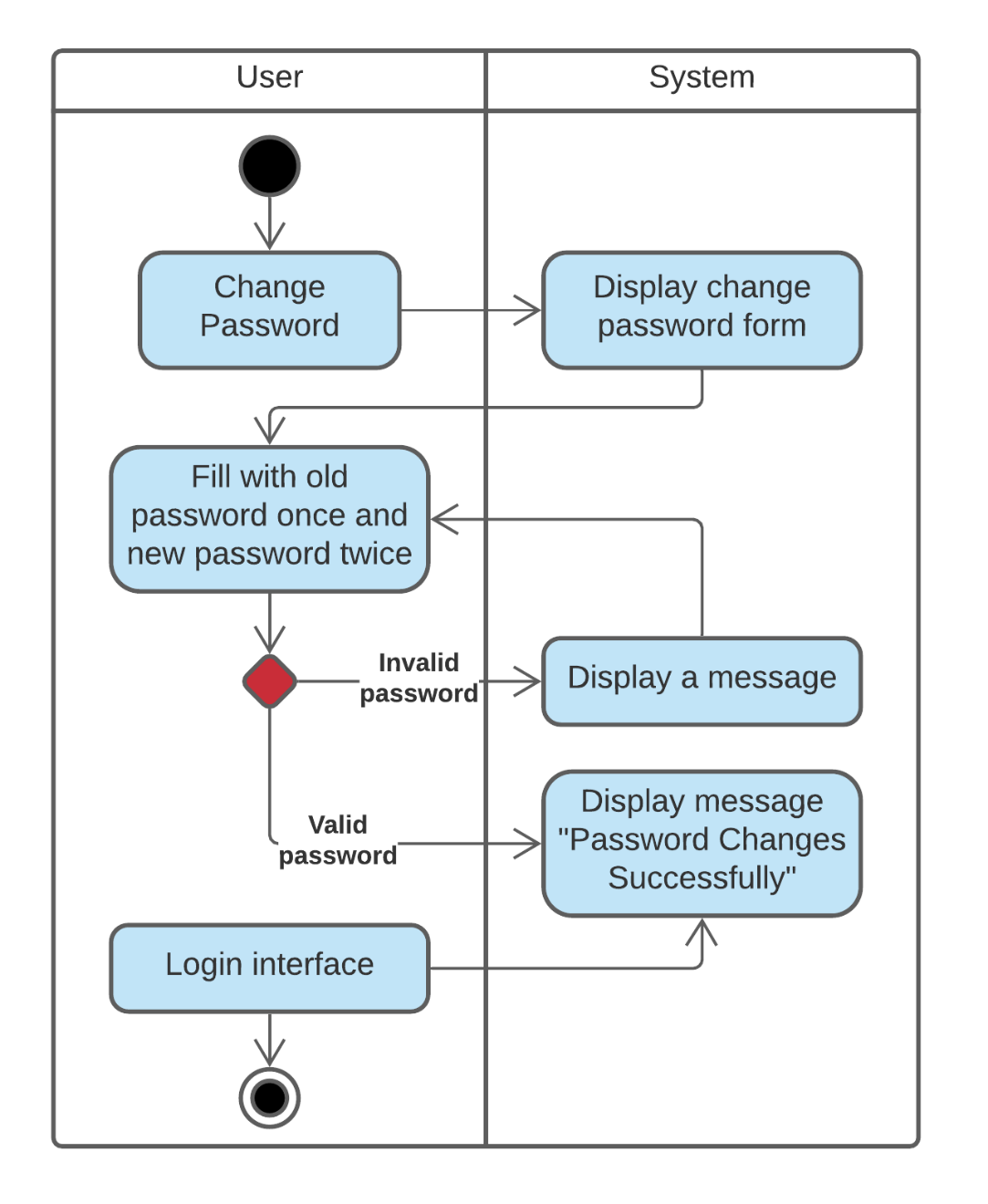
* **Use Case Description**

| Use case name: | Login | |
| --- | --- | --- |
| Scenario: | To log into the system. | |
| Triggering event: | User wants to log into the system. | |
| Brief description: | The user sign into the system by his/her Username and  Password. | |
| Actor: | Accountant, Admin | |
| Related use cases: | Use case corresponds to their function. | |
| Stakeholders: | - | |
| Preconditions: | Login system must be available. | |
| Post-conditions: | The user must have a registered account. | |
| Flow of activities: | Actor | System |
| 1. The user enters his/her  username and password.  2. The user enters wrong  username. | 1.1. The system checks for  correct login information.  2.1. The system will check if  the information is wrong.  2.2. Display the message:  “Wrong username”. |
| Exception conditions: | 1.1. Login information not available.  2.1. Login system failed. | |

*Table 3. 6. Use case description for Login*

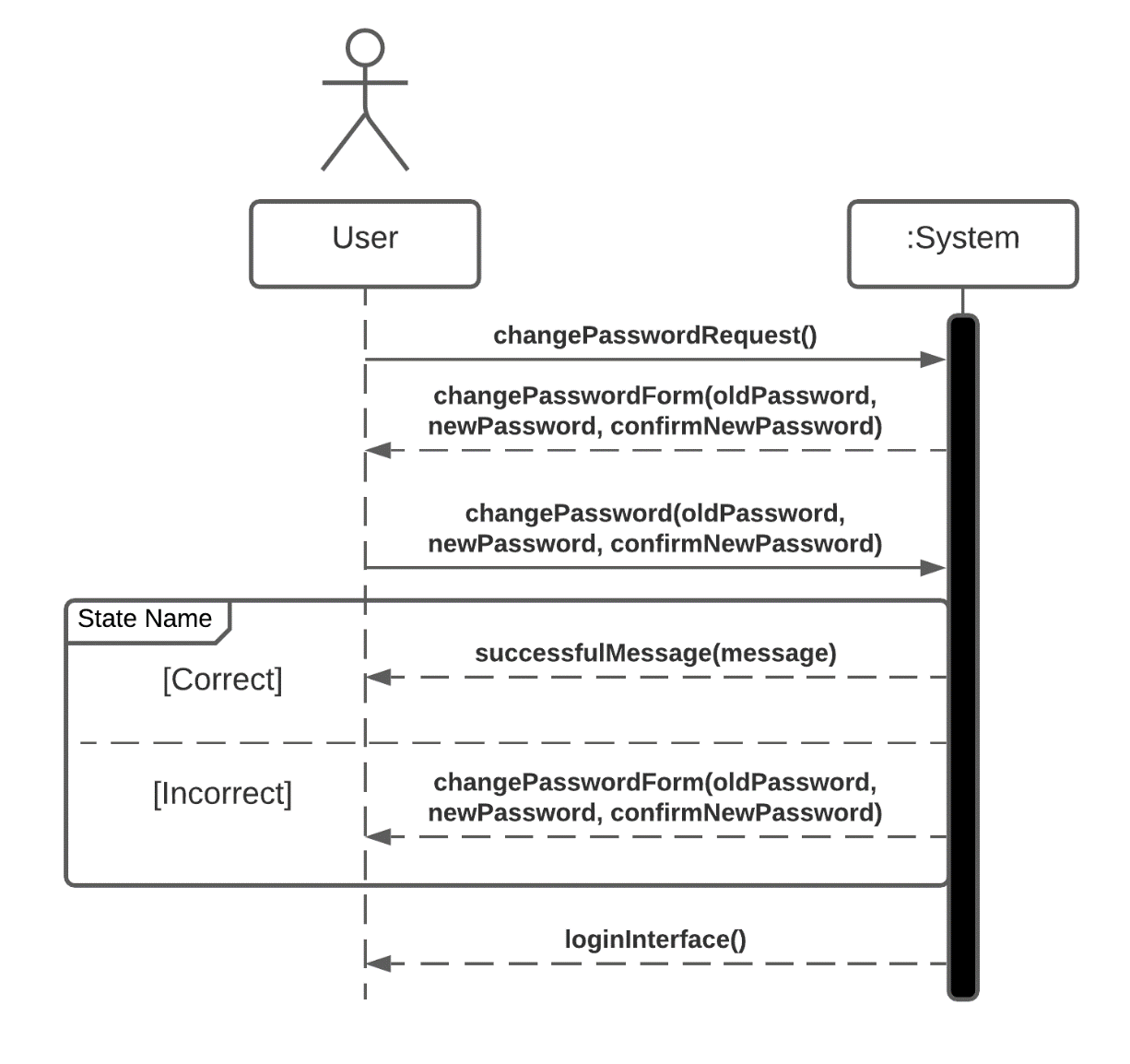
#### Use Case: Change Password

* **Graphic use case model**
  + **Activity Diagram**

****

*Figure 3. 12. Activity diagram for Change Password*

* + **System Sequence Diagram**

****

*Figure 3. 13. System sequence diagram for Change Password*

* **Use Case Description**

| Use case name: | Change Password | |
| --- | --- | --- |
| Scenario: | Change password. | |
| Triggering event: | User wants to use a new password. | |
| Brief description: | The user will enter the new password. | |
| Actor: | Accountant, Admin | |
| Related use cases: | Login | |
| Stakeholders: | - | |
| Preconditions: | Change password subsystem must be available. | |
| Post-conditions: | The user must be entered the new password twice.  Then user saved this new password. | |
| Flow of activities: | Actor | System |
| 1. The user enters the new  password.  2. The user only enters the  new password once.  3. The user don’t fill out the  new password. | 1.1. The system will check for  a valid password  1.2. The system will display  the message: “Password  changed successfully”.  1.3. Then the system will back  to the login interface.  2.1. The system will display  the message: “Enter the new  password two time.  3.3. The system will prompt  for fill out. |
| Exception conditions: | 1.1. Change password subsystem failed. | |

*Table 3. 7. Use case description for Change Password*

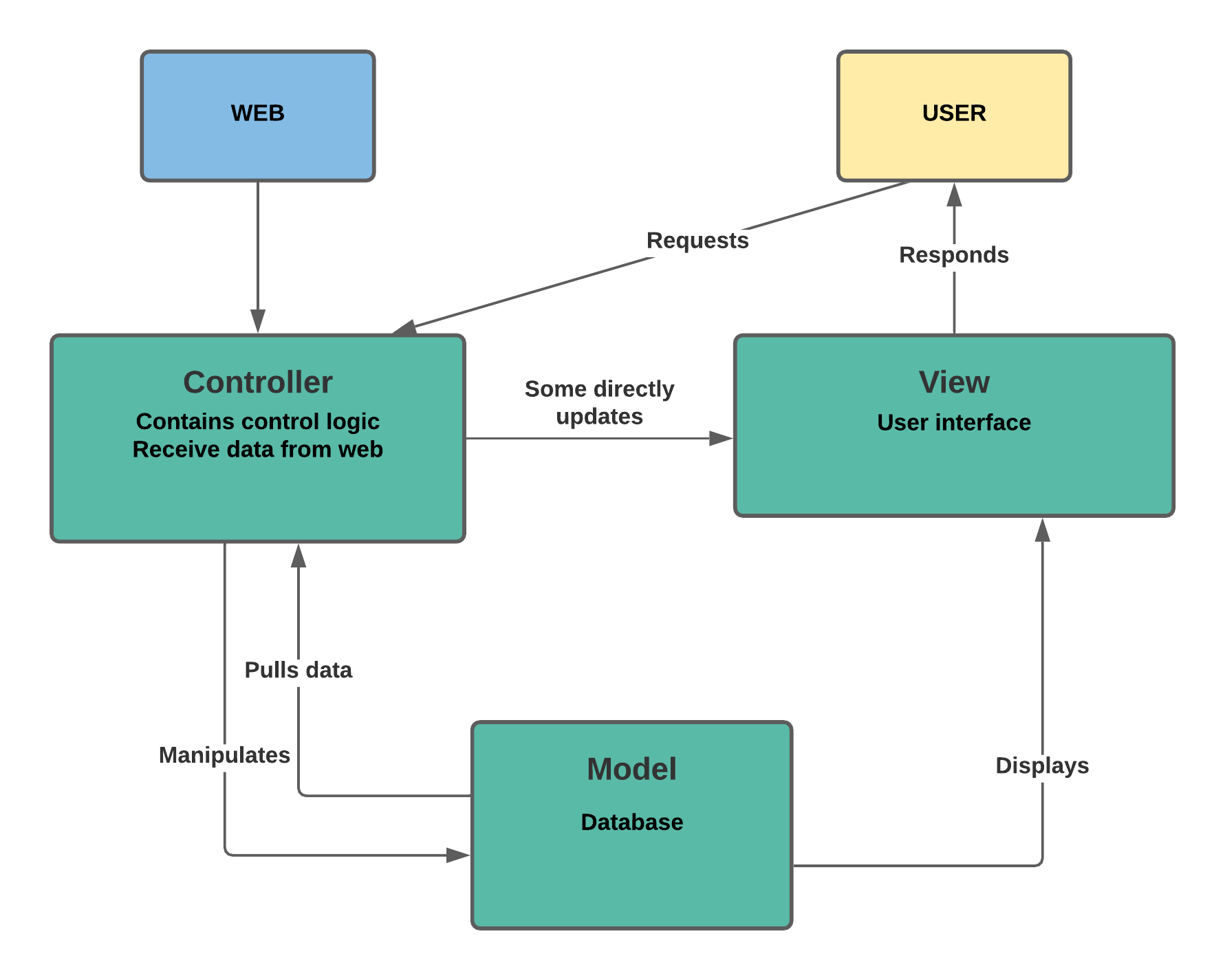
* + 1. ***Use Case: Manage Users***
* **Description**

| Use case name: | Manage users | |
| --- | --- | --- |
| Scenario: | View information of accountant and can add, edit, delete  information. | |
| Triggering event: | The admin wants to view information of accountant.  The admin wants to add new accountant into the system.  The admin wants to edit information of accountant.  The admin wants to delete an accountant.  The admin wants to search an accountant quickly. | |
| Brief description: | When company has a new accountant, the admin will fill out their  information and add into the system.  When there is a wrong information of the accountant, the  admin will edit to the right information and save this.  When there is an unused account for a long time due to retirement or fired, the admin will delete that account.  When the admin wants to see an accountant, the  admin will enter information in the search bar. | |
| Actor: | Admin | |
| Related use cases: | Login, Add user, Edit user, Delete user, Search user. | |
| Stakeholders: | - | |
| Preconditions: | The system has granted rights to admin. | |
| Post-conditions: | If “add”, the admin must have the information the system  needs.  If “edit”, the admin must have the correct information about  accountant.  If “delete”, the admin must be searched the accountant want to  delete. | |
| Flow of activities: | Actor | System |
| 1. The admin wants to add  new accountant into the system. The admin fills out all information of new account. Then, click “Add”.  2. The admin wants to edit  wrong information of accountant.  The admin fills out correct  information of accountant. Then,  click “Edit”.  3. The admin wants to delete  an unused account. The  admin searches information of  account, chooses the account  want to delete and clicks  “Delete”.  4. When the admin adds or  edits information, he/she fills  out wrong information.  5. When the admin  searches the accountant, he/she wants to view the information  quickly. But he/she fills out  wrong information in the  search bar. | 1.1. The system will check for  a valid information.  1.2. The system will display  the message: “Add  successfully”.  2.1. The system will check for  a valid information  2.2. The system will display  the message: “Edit  successfully”.  3.1. The system will display  the information of that account.  3.2. The system will display  the message: “Delete  successfully”.  4.1. The system will check the  information.  4.2. The system will display  the message: “Fill out wrong  information. Enter again,  please”.  5.1. The system will check for  information that doesn’t exist.  5.2 The system will not  display anything because the  search information does not  exist. |
| Exception conditions: |  | |

*Table 3. 8. Use case description for Manage Users*

## 4. ARCHITECTURE

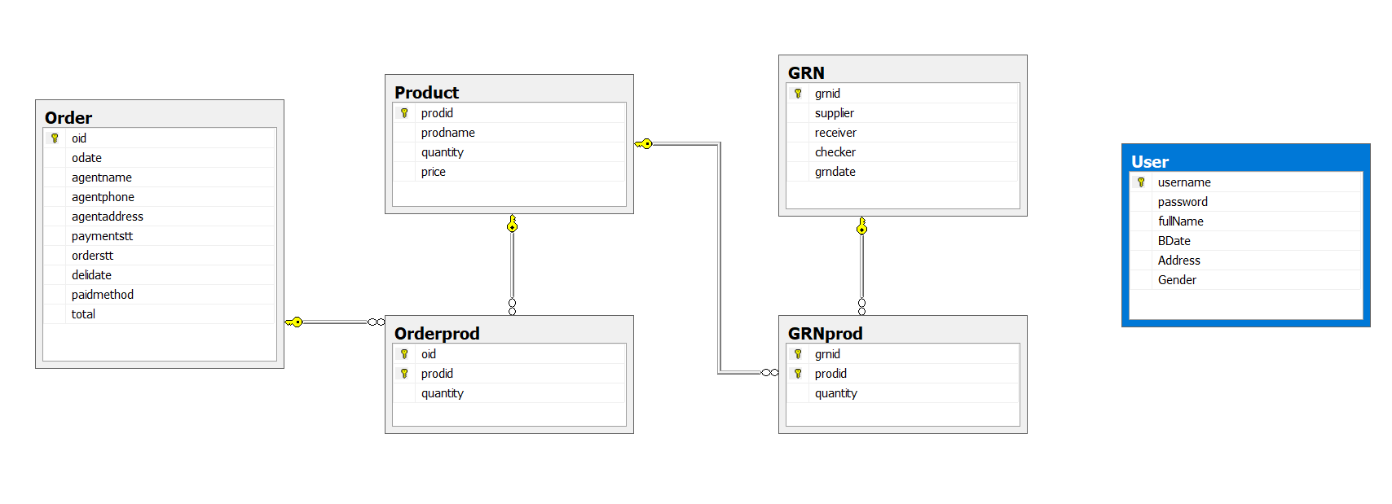
### Architectural model: MVC

****

*Figure 4. 1. MVC model*

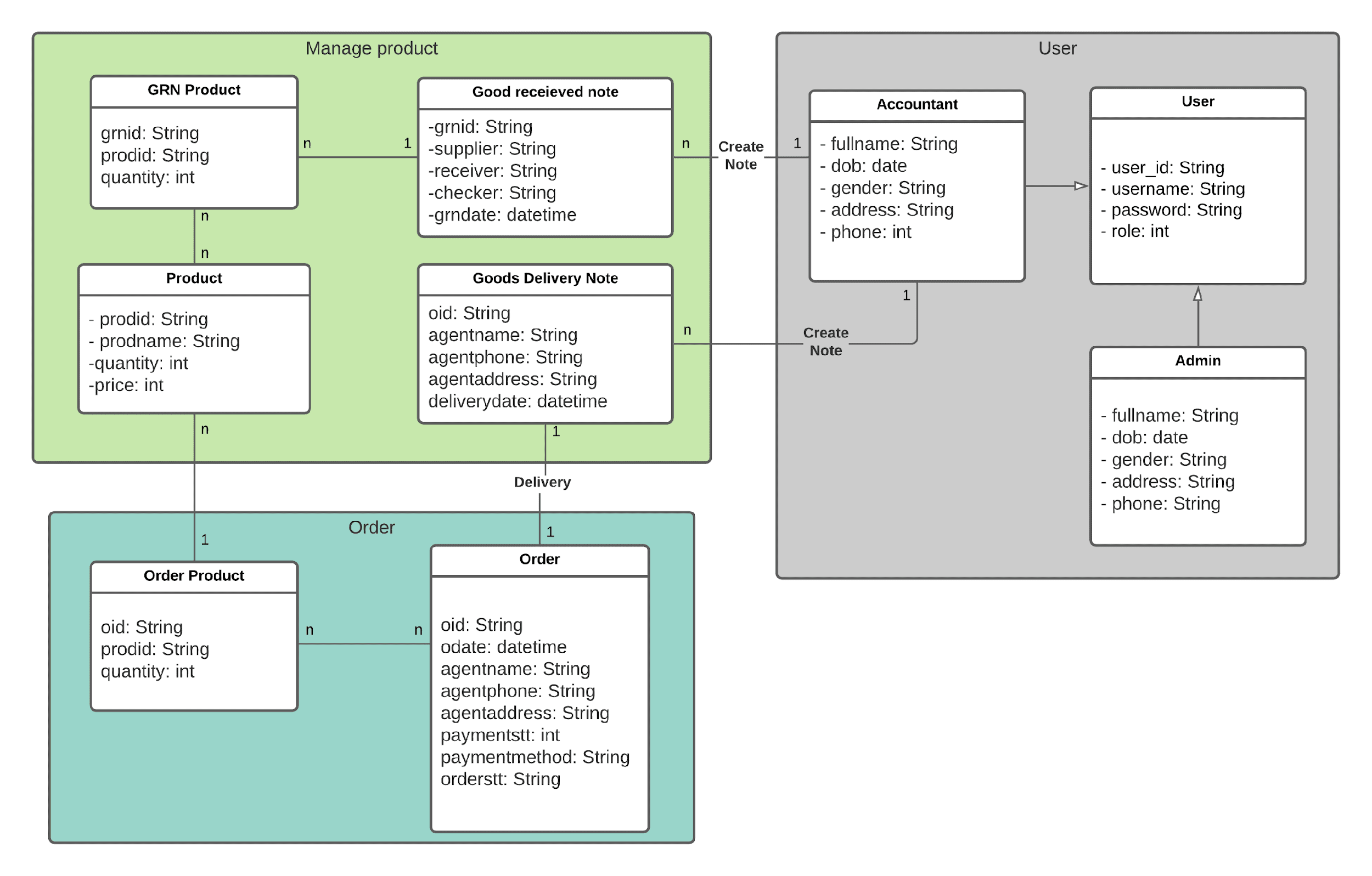
## 5. DESIGN

### 5.1. Database design

****

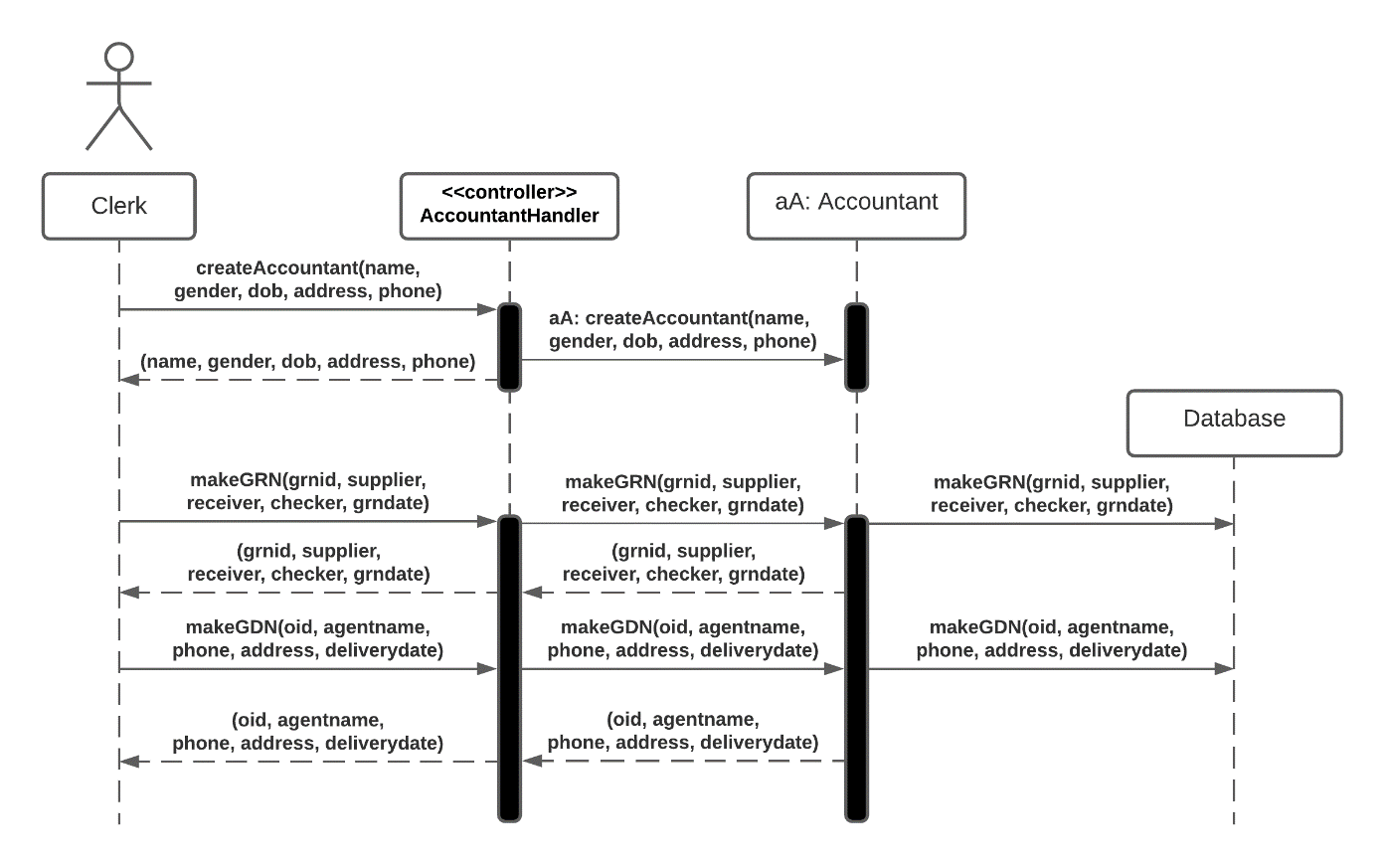
*Figure 5. 1. Database design*

### 5.2. Static model – class diagrams

****

*Figure 5. 2. Class diagram*

### 5.3. Dynamic model – sequence diagrams

****

*Figure 5. 3. Sequence diagram*

## 6. TEST PLAN

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

Test case 1:

Username: admin

Password: admin

Test case 2:

Username: user

Password: user

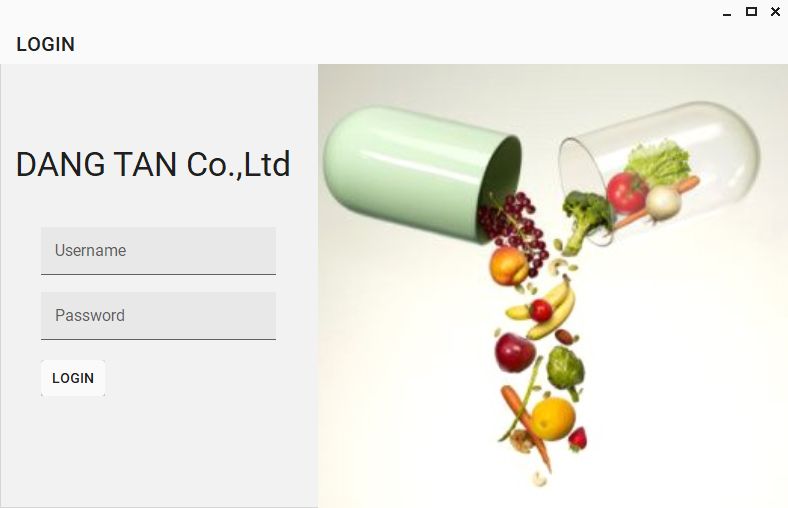
### 6.2. Traceability of test cases to use cases

### 6.3. Techniques used for test generation

### 6.4. Assessment of the goodness of your test suite

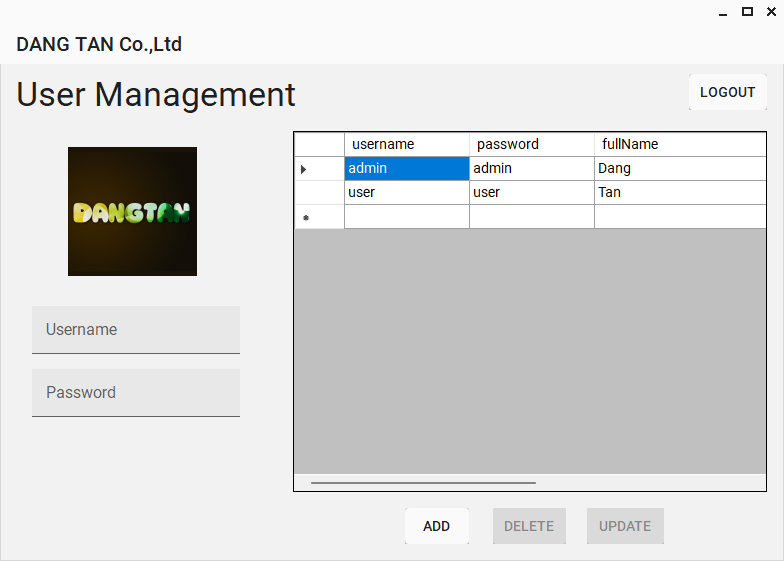
## 7. DEMO

*Login interface*



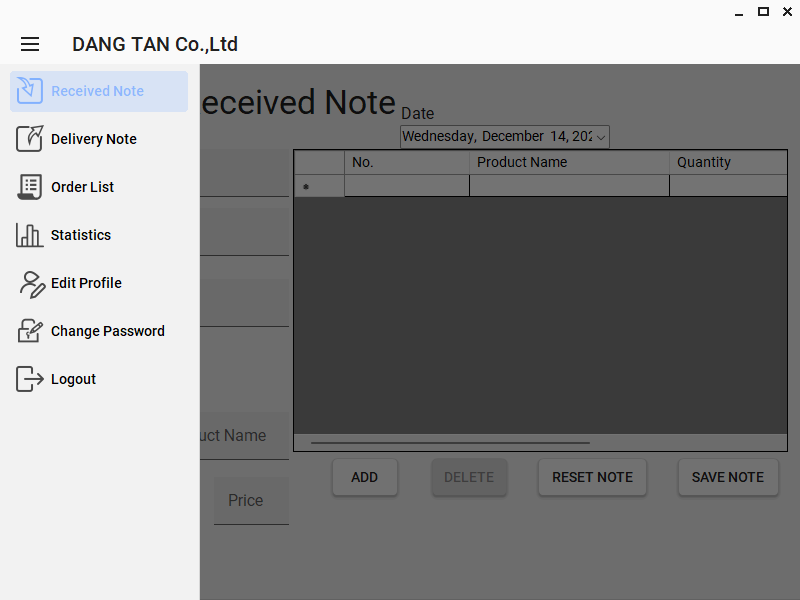
*Figure 7. 1. Login interface*

*Admin (manage users) interface*

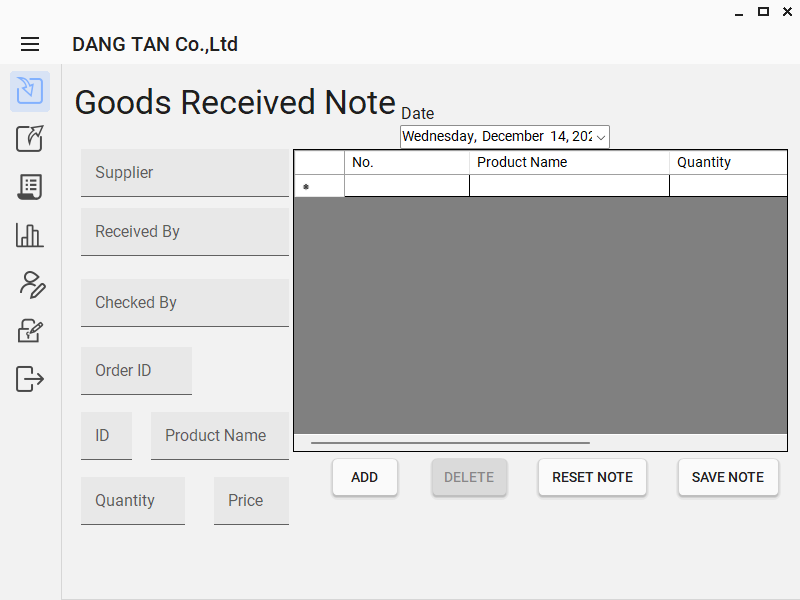


*Figure 7. 2. Admin (manage users) interface*

*Main interface*

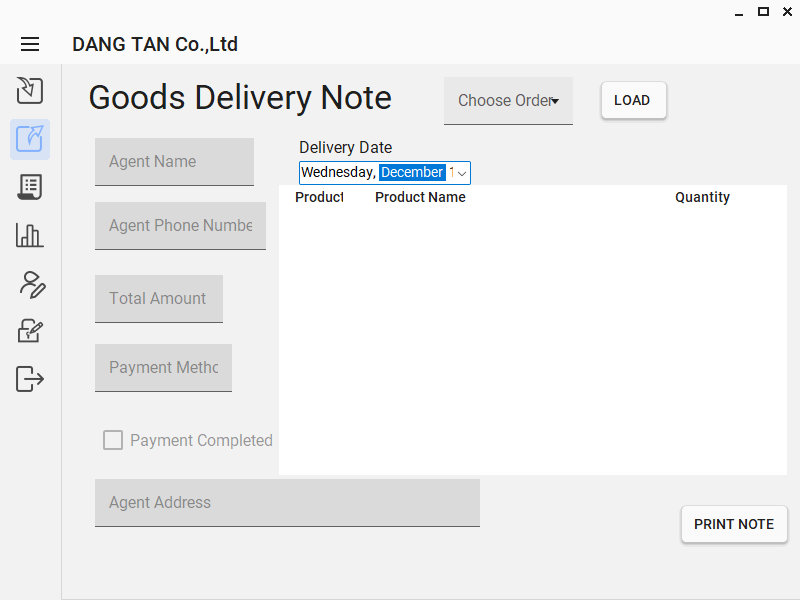


*Figure 7. 3. Main interface*



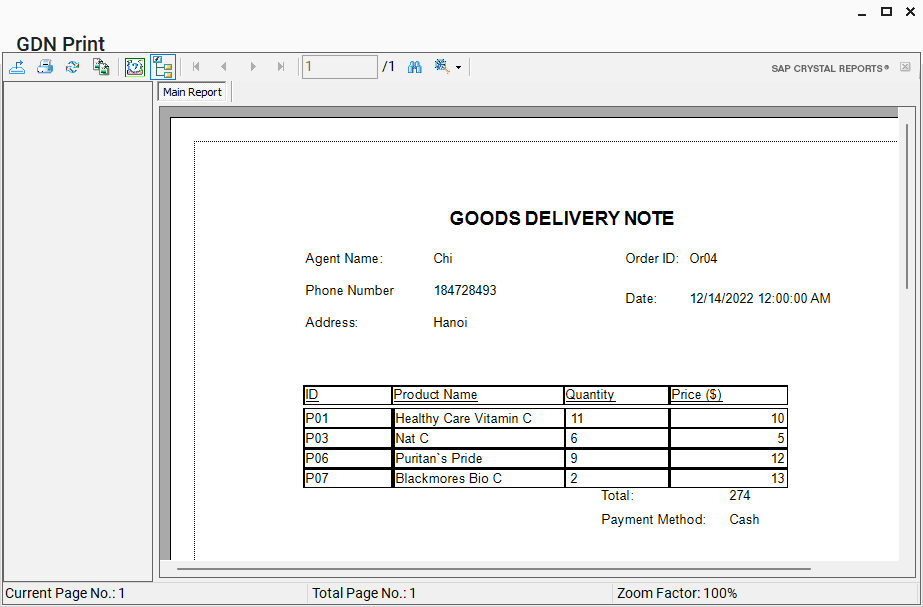
*Figure 7. 4. Main interface (2)*

*Good Delivery Note interface*



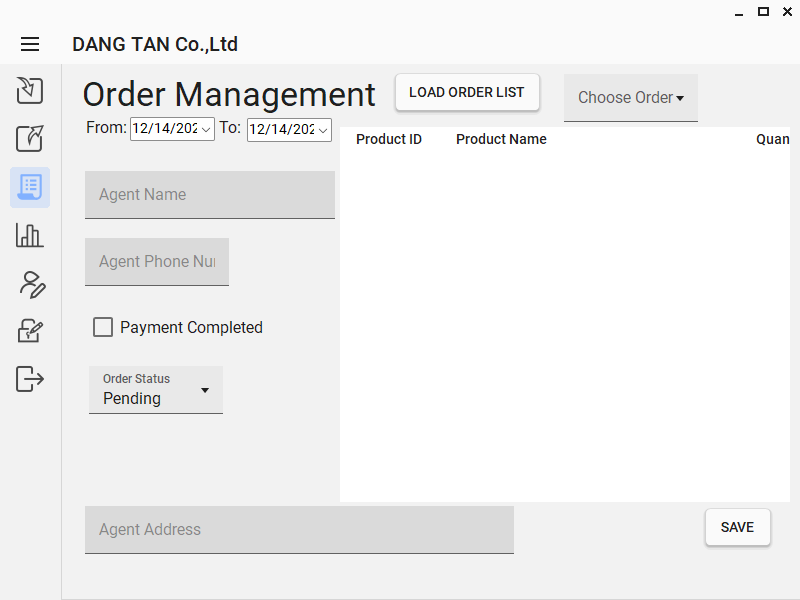
*Figure 7. 5. Good Delivery Note interface*

*Print Delivery Note*



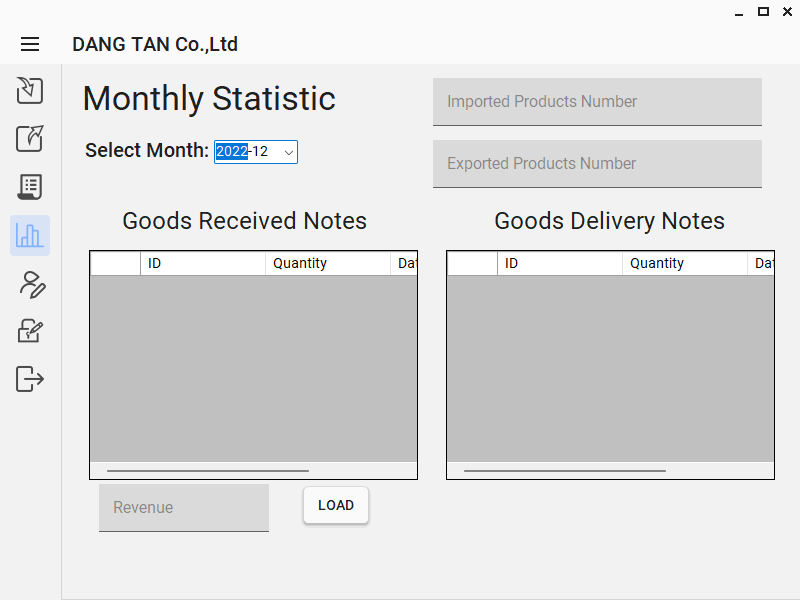
*Figure 7. 6. Print Delivery Note*

*Order Management interface*



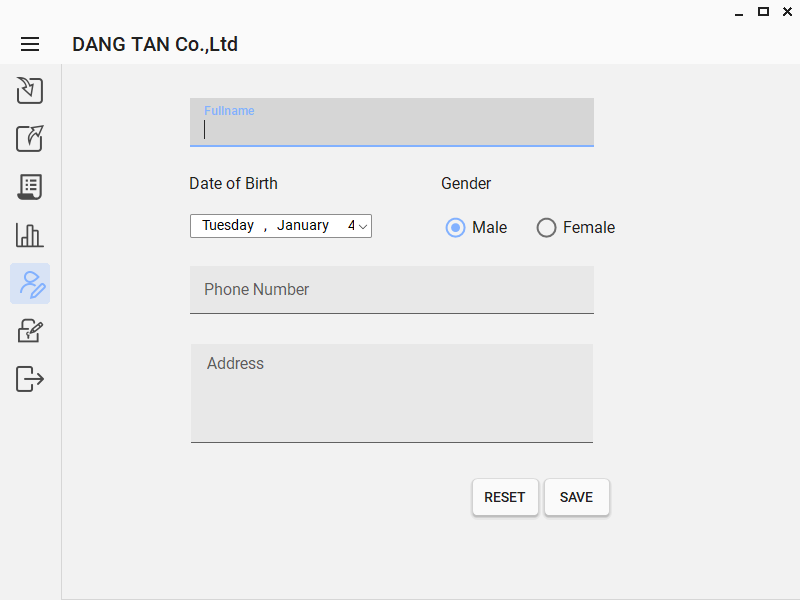
*Figure 7. 7.Order Management interface*

*Statistics interface*



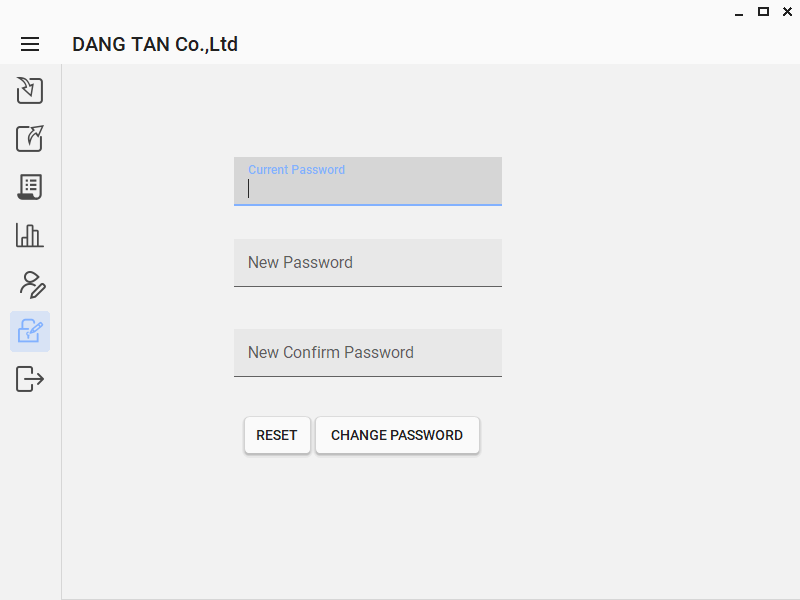
*Figure 7. 8. Statistics interface*

*Edit profile interface*



*Figure 7. 9. Edit profile interface*

*Change password interface*



*Figure 7. 10. Change password interface*

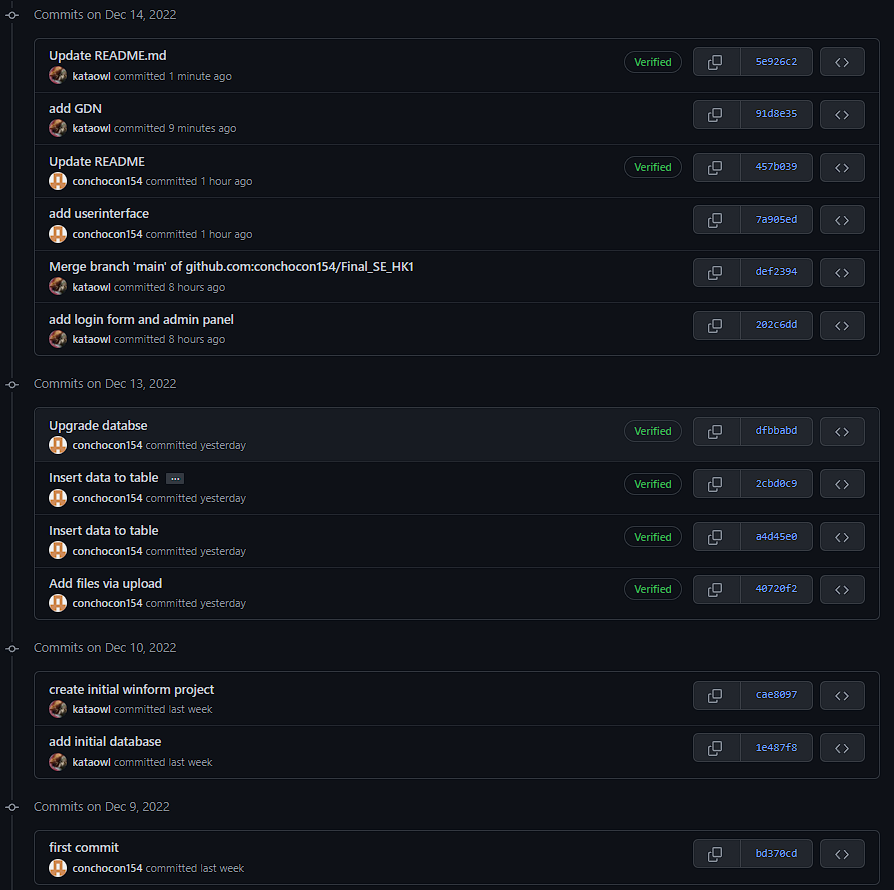
# REFERENCES

[1] SAP Crystal Reports 2020 User Guide

[2] Pro PHP 8 MVC Model View Controller Architecture-Driven Application Development by Christopher Pitt

[3] Sommerville Software Engineering 10th edition

# **APPENDIX**



*Figure 8. 1 Git-commits interface*

| MSSV | Name | Process |
| --- | --- | --- |
| 520K0108 | Lê Minh Đăng | 100% |
| 520K0342 | Cao Khánh Tân | 100% |