HAAGA-HELIA
University of Applied Sciences

Tin Phong Sales Management System

Simeon Mushimiyimana

Thesis report
Business Information Technology
2009

HAAGA-HELIA
University of Applied Sciences

ABSTRACT

04.06.2009

Business Information Technology

Author Simeon Mushimiyimana	**Group**
Title Tin Phong Sales Management System	**Number of pages and appendices** 16 + 117
Supervisor Kirsti Jalasoja	

The purpose of this project was to design and implement the Tin Phong Sales Management System**.** The Tin Phong Sales Management System is a sales management system offers services of handling customer, product, stock and sales data.

The implementation of Tin Phong Sales Management System was based on system and software requirements specifications that were also analyzed at Haaga-Helia in Autumn 2008 and was developed for Tin Phong Trading Co., Ltd.

Tin Phong Trading Co., Ltd. is spare spare-parts trading company located in Ho Chi Minh city, Vietnam. There are three different categories of sales are in Tin Phong Trading Co., Ltd.; retail, Wholesale and contract sales where information about these sales as well as customer and product related information is stored on paper based information which often causes problems in data storage, data transferring and sometimes data lost where one can understand how time consuming it is to search and find information when needed or how much it can cost the company if a given information is lost.

This project was carried out as student Thesis with which there is a time limitation, due to this limitation only part of the system was developed, hence the use of **Sub-system**.

This sub-system covered only customer, product and contract sales information. Contract sales contains information about contract, deliveries and payments while a contract is more about terms and conditions, information about product sold as well as who was the customer who signed the contract. In contract sales a customer makes contract sings it and makes payment, the company delivers products to customer and with the sub-system that is ready now, information about these activities can easily be stored and retrieved any time needed.

Being a desktop application, this sub-system has security control that allows users to perform a number of activities according to their rights and the rights are based on position occupied in the organization.

Java programming language was used in implementation, eclipse as text editor, Microsoft Access as DBMS, Rational Unified Process as software implementation process and quality standards were applied as well.

The implemented sub-system is ready and can be taken into use and it is likely that from the value it will bring to the company, company will decide to finalize its part that is not implemented yet.

Key words
Technical design, sales management application, software development, implementation.

Contents

Appendices

- Design document

- Implementation document

-User guide document

VOCABULARY

Term	Description
Measurable Organizational Value	In Information Technology project, MOV is the value that the organization receives when the system is implemented. The role of MOV is to help determining the IT project's success relaying on value it brings to the Organization rather than on whether the project was completed on time and within budget as it might be possible to complete a project on time and within budget without fulfilling system requirements thus not bringing any value to the organization.
Contract Sales	When the customer want to buy big amount quantities of products from the company, the customer negotiates the product price, payment, delivery, and other terms and conditions with the company. After this negotiation, if both parties agree, data about this agreement, data of contract payments that will be made by customer as well as deliveries made by company to the customer constitute contract sales.

ABBREVIATION

Acronym	Meaning
GUI	Graphical User Interface
UML	Unified Modelling Language
RUP	Rational Unified Process
IT	Information Technology
MOV	Measurable Organizational Value

1. INTRODUCTION

1.1 Background

Tin Phong Sales Management System, is used to promote the company in managing its sales information. When thinking about sales information, the customer related information, product information, payment information, delivery information as well as contract information product information were given priorities in this work.

By this, it is clear that information about customer, product, contract, delivery, payment, different sales(retail, wholesale and contract), customer feedback as well as information about returned product (in any) is recorded in the system, and once in the system the same information can be retrieved, updated or deleted according to the business rules and when needed.

While managing information, a certain number of activities need to be performed for example while viewing information, this information has to be synchronized in certain way for better view, updating information needs some specific rules as well as printing or deleting. So, this system takes care of all these activities and furthermore checks the user authentications by determining what the current user is allowed to do and what is not allowed as well.

1.2 Objectives

The objectives of this work can be classified in at least three kinds of objectives

- **Learning objectives:** After completing this work successfully, the student has learned how to plan, organize and monitor an IT work as individual, and has gained experience that can be basis for future tasks that will be assigned to him. The student has also learned how to work in communication with a steering group which is considered to be the decision maker and this is very important when thinking to work in real life.

- **Deliverables:** Being key elements of carrying out this work, deliverables are the ones that will bring the Measurable Organizational Value (MOV) to the company and it is from this that the work done can easily be evaluated.

- **Quality:** Quality and MOV are parallel in IT Projects. So, if a system is poorly implemented or has not reached quality achievements, then it fails to fulfil its requirements and can't offer services as planned or needed and the result is that the company will not gain from it and therefore will not get any MOV from it.

1.3 Scope

This project has focused on Manager Customer, Product, Staff and Contract Sales Information and system security in which the system has to check the current user and thus allow the services to be performed according to the security rules, for these use cases and security it has been decided to focus on design, Implementation, testing and documentation. Apart from the technical documentation there has also been done the user guide which guides step-by-step the normal users how to user the system.

Due to the time limit assigned to the Thesis some use cases that were left and will be implemented later after the thesis is complete these use cases are;

- Retail
- Wholesale
- Return Product
- Feedback

1.4 Company

Tin Phong Trading Co., Ltd is motorcycle spare-part trading company in Ho Chi Minh City, located in Vietnam. Tin Phong has as a business, supplying different kind of motorcycle spare parts imported from abroad or from manufactures in Vietnam to the middle man, motorcycle workshop or motorcycle spare part stores in retail or wholesale scale.

Tin Phong has an office and two warehouses, from where the Sales Assistants can promote products to the customers or customer can visit themselves the area to explore the products. Currently Tin Phong operates under three departments; Accounting Department, Sales Department and Warehouse department.

Accounting Department takes care of keeping, inspecting and auditing financial records of the company.

Sales Department is a place where all sales activities of the company take place. Among these activities include customer orders, sign contract, make quotation list, payment, receive and register feedback, handle returned product information and so on.

Warehouse Department The warehouse department, it deals with the product storage, the warehouse has stock in which product are stored. Even though it has been mentioned in feasibility study that this department is an external agent it plays a key role in managing our product information, thus being partly needed in our application.

1.5 Development methodology

Methods that have been learned at Haaga-Helia such: Rational Unified Process –modeling method and Unified Modeling Language –notation was applied in this work.

Software incremental development was used as a development type. For quality assurance, ISO 9000-3 Quality management and quality assurance standards Part 3: Guidelines for the application of ISO 9001 to the development, supply, installation and maintenance of computer software was also applied.
The choice of the RUP process in this application development was based on many reasons

but the many two ones are;

- **System size:** The size of the application, this application was big so that due to the thesis time limit it was not possible to implement it as whole. Therefore, designing, implementing, testing and documenting one particular part of it is very important because we make sure that that part is completely done so that the one who will continue the rest will exactly know where to start.

- **Developer's experience:** The experience of developer was given a priority, this means while doing this work I was learning from experience, what is learned in the first phase is also applied it in the second phase and so on. This reflects "**Learning objectives**" which are parts of the Objectives of this work..

1.6 Development Environment

The system was developed as an open source under Java environment using eclipse as test editor tool.

For design and documentation

- MS Office word 2003 and 2007
- MS Office Excel 2003 and 2007
- MS Office Visio 2007
- Rational Rose Enterprise Edition
- F-Secure Anti-virus (2008)
- Internet Explorer

For Implementation and testing

- Eclipse 3.2
- Microsoft Office Access 2003
- Java 2 Runtime Environment

1.7 Structure‘s factors

This project was developed from the requirements that were available at Haaga-Helia and it is understandable that these requirements are major factors to it structure. Among other factors are; the fact the commission part was not in Finland were the project was carried out and there was no direct communication with the developer due to language obstacles has an influence on the project structure as well.

2. PRESENTATION

The presentation part shows how the work was completed comparing to the plan and step-by-step how the phases were implemented. So, there were phases that have been planned to be implemented as follows:

2.1 Plan

- Phase 1: Get acquainted with requirements
- Phase 2: Software top level and Database Architecture Design
- Phase 3: Product and Customer development
- Phase 4: Contract sale development
- Phase 5: Wholesale development
- Phase 6: Retail sale development
- Phase 7: Software testing
- Phase 8: User Guide documentation
- Phase 9: Thesis reporting

It had been planned for the phases 3-6 that these phases would be developed as increments, where at the end of each phase the results were:

- Communication diagram
- Graphical User Interface
- Tested code
- Finished document

2.2 Design

In implementation of the plan the implementation methods planed were maintained, it had planned to use software incremental method. This was because the developer was at the same time learning and it assumed that experiences learned from the previous increments or pieces of the application. However there as a slightly change in phase's outcomes because all planned results for each phase were not achieved at the end of each phase. The Finished document was not done at the end of each phase and this was because the developer found it easy for him to concentrate on system implementation for at least phases 3-6 and documentation after.

- Get acquainted with requirements:

Problems: Given the fact that this project was based on requirements that were not ready yet, they have been all the time some changes in these requirements and this caused the delay and some confusing things**.**

Solutions: Rapid adaption on changes and work on week-ends like Sundays and Saturdays to recover time lost.

-**Software architecture design**: It is very useful because before starting an implementation, it is the one that holds the components in communication while the system is running and remembering that classes reside in component it is a good idea to start from the component itself make sure components are communicating as they should and after that it will be possible to open each component to add classes.

Problems: Due to lack of experience, it was not easy to get to know exactly what was needed.

Solutions: The regular contacts with supervisor to get to know what is exactly needed and how model it.

-**Database architecture design**: Database tables as well as fields in tables are very important because database can read the rest of the system an way that when information comes to the database to be stored or as a request to retrieve and information before the answer is given the database constraints apply, in other words the database has to have the same type of information needed or have a place to store information that is coming. It is critical to add that even if the class model that comes from the software requirements needs to be transformed in a logical database model where the developer has to analyse and synchronize tables while adding other tables to handle some complex matters and one can not forget the use of some auto-generated fields that you will sometimes not find in class model from software requirements.

Problems: In this sub-phase there was not major problem, the minor problems were just to analyze database table communication to get the right information as needed but even though some changes were being made it did not attend the level of a real problem.

2.3 Implementation and Testing

Business and Control Implementation

Now that database and architecture components are ready, I open system components to add classes, I have preferred to start from business classes as they the ones that provide methods to set and get (return) valuables that are needed in process communication, though one might think that this is not as important as other things but it is a good idea to remember that business classes are somehow in direct communication with database through get and set methods when they are needed by database utility classes such ResultSet that handle database transactions.

Once you have succeeded to match valuable types and length in both business classes and in database table's fields, you have reduced the work to be done, remember that one business class can be called as many times as needed in adding information, updating or deleting, they are made once and used several times.

Controller

The unique class in controller is the core of the system because it has the main method that runs the whole application. As the database has been initial made and business classes are ready even though the might be changed through out the implementation process the next step is to have this class that will run the application so that when I make the user interface or Data access class I will make sure database connectivity can be checked.

GUI Implementation

Now that database tables are, business classes are ready and my application can run as the main method has been implemented, I add GUI which I will make sure it has all fields needed to match the fields in business class as well as in corresponding table in my database. Now I can connect by two components that is **guiComponent and controlComponent** by this I can see my first window, see how it looks like and make it as I want it to be, but I still have some thing to do, my application is half connected as there is no connection to my database and the next is to establish connection to the database.

Data access Implementation

With the Data access class the connection to the database is established, so, having GUI and controller class it is possible to get inputs and send them to the said database but before that, in professional way the objects that are going to be in communication have to be organized in a certain way to ensure the request as well as a response will reach the destination in a very proper way, and to solve this problem we use so called Object collaboration or sequence diagram.

Sequence diagram

Sequence diagram helps me as a developer to check how if processes will operate one another and in order that it is wanted. Once the sequence diagram is ready it allows the implementation without delaying on how to connect this Object to that one, and it seems that all has already done and the rest is to transform the sequence diagram logic into real code.

Unit Testing

After the codes is ready and the three layers are in communication, that is, the inputs can be entered in GUI sent to the Controller which in turn will send them to then database through Transaction then the test can be carried out. We made our unit test is a test in order to check if one piece of software(for example, a unique business class) is offering the services that it has to offer according the business rules or technically speaking to the software requirements.

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct and problems are found early in development cycle and can be corrected with less costs comparing to these found for example in application testing where correction cause other errors which need to be corrected as well which in turn becomes time consuming and proportionally increasing in costs.

Integration testing

It was done by integrating all pieces of application to see if they were communicating as needed, for example after I have entered customer information and want to make a contract I had need to combine customer and product data. So, the act of checking if customer data and product data can be combined to make contract and is answering business needs is what we call integration testing, because we try to retrieve data from different sources (objects). For example customer information I have in my contract gives me access to the rest of other customer information and thus allowing me to access that information that was not in contract but needed for example in making report.
-Ex: Use customer number from contract to retrieve customer names, phones address and other needed data.

System testing

System testing was done by the customer while the developer was taking notes of errors found so that they can be corrected and the purpose of this test was to verify that our sub-system responds to requirements and thus to the business needs and it was done three times and each time there was test, a test results was made.

After each type of test the errors found were immediately or later fixed to ensure quality of deliverables. System test is very important because it involves the outsides parts that are not familiar with the system which will help to find errors than if this tests ware carried out by the developer himself.

3. EVALUATION

They ways and methods used in this project development are really helpful, the results achieved respond to the system and software requirements for the corresponding use cases because the results can not be referenced to the system as whole while only part of it was developed during this time. It is hoped that results achieved and especially this sub-system are ready generate the MOV to the company.

However, despite of the efforts furnished by developer, there were some problems in making reports as the tool itself did not provide enough features to make these reports. This was especially noticed when data on screens is wanted on normal print out format because it could not be do ne as requirements were stipulating, however it has been agreed with the customer to change the printed format, thus coming up with new printed out format that was can not be seen in requirements.

Apart from the deliverables, which can physically be evaluated, the student has also gained some experiences in doing things, among other include;

- Working independently
- Decision making
- Project communication to stakeholders
- Personally learning
- Project management

4. RECOMMENDATIONS

This application is not complete yet; only the following phases were covered in this work

- Phase 3: Product and Customer development
- Phase 4: Contract sale development
- Phase 7: Software testing
- Phase 8: User Guide documentation
- Phase 9: Thesis reporting

It is to be understood that the rest will be implemented after the thesis has been approved and the company will take care of that.

The company and staff business entities' requirements specifications were not defined and due to this it can not be easier to determine at which level their functionality can be evaluated as there are no requirements to match with.

5. BIBLIOGRAPHY

The system work does not require many sources but here are some used sources

- **Internet sources**

- http://myy.haaga-helia.fi/~jalki/sys8tf060/ by Kirsti Jalasoja
- http://myy.haaga-helia.fi/~silka/atk84f/ Kari Silpiö
- http://myy.haaga-helia.fi/~the7tf007/ by Kirsti Jalasoja and Markku Somerkivi

- http://java.sun.com/docs/books/tutorial/uiswing/components/jcomponent.html on 02.02.2009

- **Book sources**

-Eclipse in Action (David Gallardo, Ed Burnette & Robert McGovern)
-Information Technology Project Management (by Jack T Marchewka, 2003: First Edition)
-Java Swing (by Robert Eckstein, Mark Loy, and Dave Wood; September 1998: First Edition).

Appendices

- Design document
- Implementation document
- User guide document

HAAGA-HELIA
University of Applied Sciences

Tin Phong Sales Management System

Simeon Mushimiyimana

Design Document

Business Information Technology

2009

Contents

Appendices

I. Definitions, Acronyms or Abbreviation

Acronym/Abbreviation	Definition
GUI	Graphical User Interface
AWT	Abstract Window Toolkit
J2SE	Java 2 Platform Standard Edition
DBMS	Database Management System
SQL	Structured Query Language
JDBC	Java Database Connectivity
MS	Microsoft
FK	Foreigner Key
PK	Primary Key
TPSMS	Tin Phong Sales Management System

Table 1: Definition, Acronyms or Abbreviations

1 Introduction

Tin Phong Sales Management System (**TPSMS**) is a system that offers a set of services needed to handle sales in Tin Phong Trading Co., Ltd. These services are;

- Manage Customer information
- Manage Staff information
- Manage Product information
- Manage Sales information
- Manage Returned product
- Manage Feedback
- Manage Reports

Before the customer makes his or her first buy from the company, his/her personal information (also known as customer information) is recorded into the system.

The product information includes the stock status and product prices. The stock status is recorded into the system each time there is a change in product quantity whether by delivering or adding product quantity. The product prices are recorded into the system and can be updated each time and when needed.

Sales information, which is divided in three types of sales: retail, wholesale, and contract sale, should be recorded into system when the customer buys the products at the company. For every type of sales, there are delivery and payment records. The payment made by the customer is also recorded into the system and a receipt for payment is given. This also applies for delivery whose information has to be recorded into the system through the delivery receipt.

For contract sale, when the customer is interested in buying a certain product in large quantity or wants to monopolize the product, s/he has to sign a contract with the company by providing the negotiated terms and conditions of the contract (contract information) and the contract information is recorded into the system and after this the information about contract is printed out and signed by both parties (customer and company's staff).

The feedback and returned product information given by the Customer is also recorded into the system. The sum of the returned product will be deducted from the current debt of the Customer.

1.1. Overview

1.1.1 The Company

Tin Phong Trading Co., Ltd. is a motorcycle spare-part trading company located in Ho Chi Minh city, Vietnam.

1.1.2 Office and organization

Tin Phong Trading Co., Ltd. has an office and two warehouses. The office includes Accounting Department and Sales Department. Two warehouses are considered as Warehouse Department.

All the main sales activities happen in the Sales Department. The Customers can come over to see the product samples or the salesmen promote the products the Customers through the Sales Department.

1.1.3 Business domain

The business idea of the company is to supply different kinds of motorcycle spare parts imported from abroad or from the manufacturers in Vietnam to the middle man, motorcycle workshop or the motorcycle spare-part stores in retail or wholesale scale.

The requirements of the market have been changed quite often because of many reasons, such as the unstable resources of products, the out-of-date model of the motorcycle, and so on. That makes the selection of products change quite often as well. However, there are about 2.000 products in selection usually. The most popular product groups are brake shoes, bolt, spring, head light cover, rear light cover, signal light cover, bearing, piston, gas filter, kick starter arm, corn, relay, IC, cushion.

1.1.4 Present state of business process and data process

-**The main actor**
Sales Department is place where all the sales activity of the company takes place, such as Customer orders product, signs contract, picks up quotation list, makes payment, gives feedback, returns product, and so on.

There are Sales Assistant and salesmen in this department. The Sales Assistant is the main person who deals with the Sales Management System for all sales activities in this department. Because of the scale of the company, the director and the Accountant also takes part of this sales management process.

-The main external agents

Accounting Department is a place where keeps, inspects, and audits financial record of the company.
Accounting Department is the external agent of the sales management process and internal agent of the company.

Customer may be a person, a workshop or a store that buys products at the company. Customer is an external agent of Tin Phong Trading Ltd., Co.

A warehouse is a place where storages the company's products. Each ware house has its own stock. There is one Warehouse Keeper in each warehouse.

The warehouse is external agent of the sales management process and internal agent of the company.

1.2 Purpose

This software design document provides a comprehensive architectural overview of the sub-system what was developed during this work and which is first part of the Tin Phong Sales Management System (**TPSMS**); it will use different architectural views to describe different aspects of this sub-system. Furthermore it is also intended to bring out and communicate all architectural decisions, which have been made while developing the said sub-system

1.3 Scope

As it has been seen that the **TPSMS** will offer a numbers of services not all are covered, and the fact is that this work has been accomplished as Thesis which requires a limited amount of time which has led to a decision of leaving out some of services and concentrate on what were judged most important services. Hence, this document will only describe the following services that have been developed during this work.

-Manage customer information
-Manage Staff information
-Manage Product information
-Manage Contract Sales information

In this sub-system, the Contract Sales information is combination of contract, delivery and payment information. Meanwhile, Manage Staff information includes system account and passwords management as well as normal staff information management. As mentioned it above that time has been the key factor in deciding which services to implement and which ones to leave for later implementation, the following services will be implemented later.

- Manage Returned product
- Manage Feedback
- Manage Reports
- Retail sales
- Wholesale sales.

For each of the implemented parts there is a GUI, Communication diagram to show how Objects respond to their services when called or how they call in case there a need on a given service, Unit testing and integration testing and at the end all tested together in so called application (system) testing to ensure quality goals were reached. However, regardless on expected results at the end of each part, after noticing that some parts follow the same procedures, it has been decided to not repeat the same things and thus fore example if adding product in the system follows the same logic as adding customer, the communication diagram for adding customer was left out but instead the reference to adding customer was given just for a guide, this is so called obsolete repetition.

2 Architecture Design

The software architecture, being the structure of the software components, communication between them, it allows future software redesign because it gives clear understanding of the details of the system being developed. In this application we will be developing so called, Two-Tiers software architecture, which is considered to be the environment in which there is a client that holds user interface and the server that holds the database; Depending on software configuration both database and user interface can reside in one machine.

Layers	Tiers		Tools	Products
Presentation Layer	1	Client	Swing/AWT	Sun J2SE
			Java Classes	
Business Layer			JDBC	
Data storage Layer	2	DBMS	SQL	MS Access 2003

Table 2: Two-Tier software architecture

2.1 Layers:

2.1.1 Presentation Layer

The presentation Layer in this case refers to the graphical, textual and auditory information the system present to the user and the control sequences the user uses to control the program. While communicating with the system, user can enter some input get outputs, these outputs can either be information retrieved or some error messages indicating what has happened in the specific part of communication and at one lap of time depending on the input or request.

All classes' codes that handle information display are contained in this Layer.

2.1.2 Business Layer

The function algorithms that handle information exchange between database and user interface. Called also business logic, it has some roles to play such as:

- Modelling real life business objects
- Prescribing how business objects interact one to another

- Enforcing the routes and the methods by which business objects are accessed and updated

2.1.3 Data storage Layer

It stands for Database Management System plus database itself.
Accessing data varies depending on source of the data, Access to persistent storage, such as to a database, varies greatly depending on the type of storage (relational databases, object-oriented databases, flat files, and so forth) and the vendor implementation.

By using the Java API that is able to communicate with SQL classes, the connection to the database is established by localizing the database drivers.

2.2 Tiers

In many cases the concepts of tier and layer are used interchangeably. However, indeed there is a difference between them; that is a layer is a logical structuring mechanism for the elements that make up the software solution, while a tier is a physical structuring mechanism for the system infrastructure. Thus, in our case layers holds our codes while tiers are structure of computer environment in which our application is running.

Client: The client here describes the application that access the database
DBMS: (Database Management System), the location of the database in use.

According to our picture there are two tiers meaning that the database is located in one client operating as a server and the application in another client to access this database thus tow tiers.

2.3 Tools

Tools here describes so called built-in library that are referenced while making implementation. Here, tools are described in general but in products there specific ones used in this specific application.

Swing/AWT: For user interface design, windows features procurement.

Java classes: To run the application.

JDBC: (Java database connectivity), to establish database connection

SQL: (Structured Query Language), to handle database transactions.

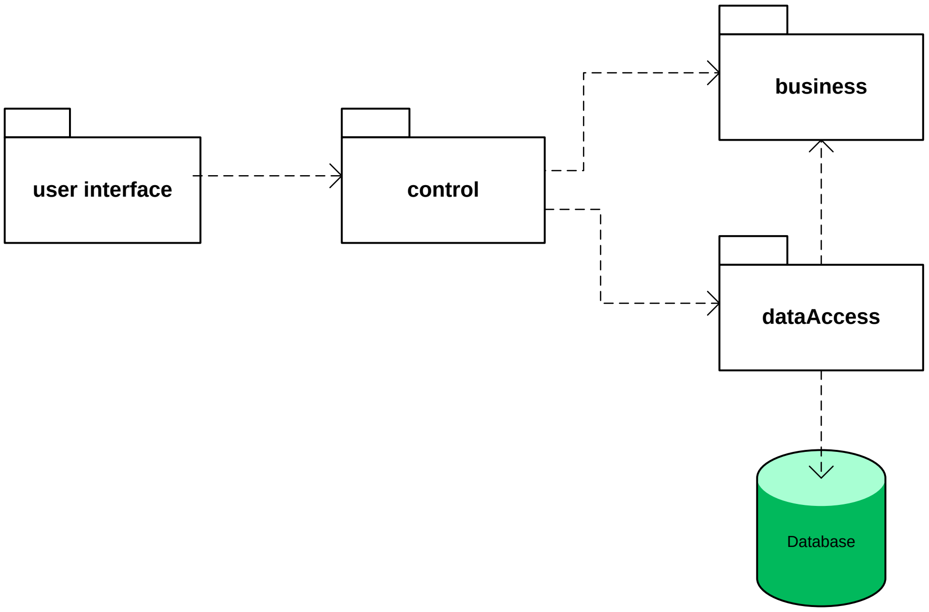
2.4 Products

Specific tool that is currently in use while this application is running

Sun J2SE: Java version.
MS Access 2003: Database product

3 Application functionality

Functionality describes which components compose the application and how they communicate with each other to handle their duties. It is also in this part that we present all the classes as well as their descriptions.



Picture1. Component Model

3.1 Components description

3.1.1 User Interface

The role played by a user interface is undoubtedly important in software development as it holds all classes that let the user communicate with the system. In fact, with the user interface the user is able to command the system what to do which in turn shows the user what was the results of the action performed or failure message in case of errors.

3.1.2 Control

The component that holds the unique class that is considered to be the main class or the class controller because it has the main method that runs the application when the user wants to start the application, without this main method the whole application is useless as it can not be run. This component also serves as bridge between other components and user Interface component

3.1.3 Business

Classes in business component play the role of letting data circulating through other components as when called they set the data in one component and thus this data can be retrieved in other component also by calling them to get is back.

3.1.4 DataAccess

Data access component handles all necessary requests by initializing the classes such as database connection classes, ResultSet and SQL classes that can connect directly to the database and retrieve data; these classes handle also some SQL and other exceptions which help avoiding the system clash when these exceptions occur.

3.2. Java classes description

Only business classes, control class and data access class are described here, GUI classes are described in User Interface Style guide description in which also all windows are described and shows as screenshots.

3.2.1 TinController

As the name specifies, TinController is the program that controls the application's behavior as it triggers the system start up, which in turn displays the Login window that is the starting point of this application.
It initializes other classes when needed and serves as Communication Bridge between classes in all components involved in this application.

3.2.2 DataAccessObect

The DataAccessObect together with other environment built in classes handle all database transactions. These transactions include adding data to, retrieving data from, updating data, searching and deleting data from the system according to the requests presented to it.

3.2.3 Customer

Business class that serves for carrying customer data when control class is communicating with data access class; this is done either by setting variables or getting them using its getters and setters methods.

3.2.4 Product

The same, as in customer but in this case for product data

3.2.5 Staff

The same, as in customer but in this case for Staff data

3.2.6 Contract

The same, as in customer but in this case for Contract data

3.2.7 Delivery

The same, as in customer but in this case for Delivery data

3.2.8 Payment

The same, as in customer but in this case for Payment data

3.2.9 ContractProduct

The same, as in customer but in this case for ContractProduct data

3.3 User Interface Style guide

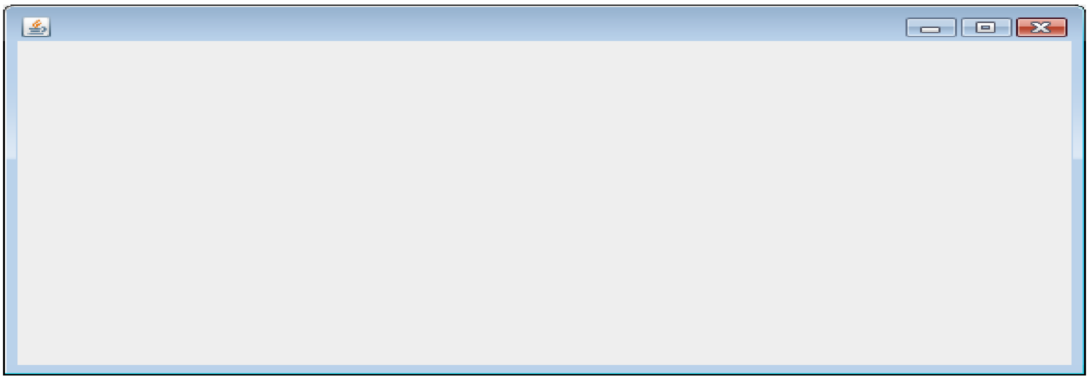
The classes in this component are the ones that are in direct communication to the user, in order words; they perform user's commands and display answers. This component holds classes that will be explained in details when describing the user interface style guide.
In the user interface style guide, we define what fields are going to be seen in the application as well as their meanings to the user.

The user interface style guide sets the standards for all the applications in Tin Phong Sales Management System. It will describe the basic outlook and features of the all windows implemented in this system (back colour, font size, style, colour, etc.), some messages that are displayed will also be shown here and. However, not all controls pictures as shown here.

3.3.1 List of Controls

The names of these controls are the normal name of Java control but because we use Java Swing in implementation of this application it is a good idea to write them with their technical names where the letter "J" that starts every control stands for the name "Java". However with them you can do what you could do using Java AWT controls.

- **JFrame:** Normally the frame represents what the uses sees, but in technical point of view frames are java components that are supposed to be top-level other components and with them we can easily navigate from one to another.



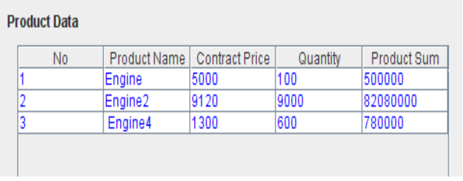
Picture 2: JFrame window

- **JLabel:** The Label is used to identify the name of the field that it stands for.

- **JTextField**: This is a control that allows the user to enter some values and thus save them in the system, it also displays to the user the data retrieved.

- **JButton:** Buttons are very useful GUI components as they tell the system what the user wants and thus letting the system behaving according to the user needs.

- **JTable:** For data synchronization, once the data is loaded into table it organized without any other special transformation to it.

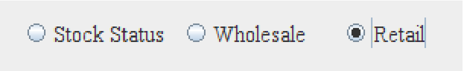


Product Data

No	Product Name	Contract Price	Quantity	Product Sum
1	Engine	5000	100	500000
2	Engine2	9120	9000	82080000
3	Engine4	1300	600	780000

- **JPanel:** For components grouping, it is useful when positioning many components on window, as they are added to panel and the panel itself added to the window where it can be used as many times as needed.

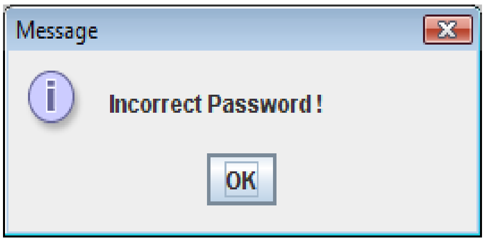
- **JCheckBoxGorup**: For elements selection, boxes are selected according the information needed.



○ Stock Status ○ Wholesale ● Retail

- **Message Box**

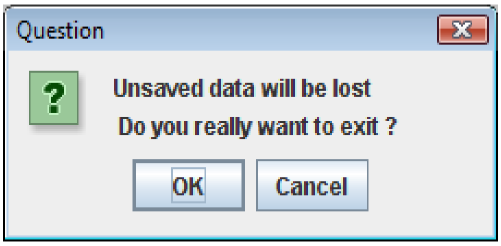
Message box displays an error message, a warning or just a simple message. Messages require different reaction about them, for example a message such a warning can wait until the actor decides on what to do in order to continue while other simple messages are just to show that the action was performed or an error occurred.



Message

Incorrect Password !

Message Box showing only information.



Question

Unsaved data will be lost
Do you really want to exit ?

A message showing some risks while putting the system on alert to wait the decision form the use, any button clicked in the second message will tell the system what to do while in the first message it was only to inform the user and the system does nothing after that.

3.3.2 Terms of commands used

Command's name	Functionality
Add	This button is used when entering new data into the system.
Change password	Change password button is used to change old password to a new one.
Delivery	To manage delivery information; add new delivery listing all deliveries and thus printing out the delivery list.
Exit	Exit button is used to close the currently working window. But can also while closing the whole application in case of exiting the Login window.
Empty All	When the current window has been filled with data, and there is no way to change the data it can be possible to empty the current window with this command to be able to search new data or perform other activities.
Find	Used to search data from the system regardless on which searching criteria.
List All	To display data for the current use case, this is performed without any access key all related data is displayed regardless on size or amount available in the system.
Login	To authenticate to the system, for getting rights to perform any activity the actor needs to log in to the system.
Manage Customer	Found on Main Window, this button is used to display Customer window from which customer information can be easily managed.
Manage Feedback	This command is used to display Feedback window for viewing or adding new feedback into the system.
Manage Product	Manage product information such adding, updating, searching or listing all.

Command's name	Functionality
Manage Staff	Command that once performed will open the ManagingStaffWindow.
Manage Report	Display report window. For Adding, viewing or updating report.
Manage Sales	
New User	This command button is for creating new user account
Payment	Once clicked the Payment sub-window opens from which a new payment can be made or a list of all payments belonging to the contract can be displayed.
Return Product	Display Return window to manage returned product data.
Select	Once the current window displays a list that allows selection the actor will use this command button to let the system now on which item from the list the actor is interested.
Update	When updating data on current window and the fields on this window are disabled, the update command will enable only fields that can be changed for the actor to start changing the information displayed in these.

Table 3: Terms of commands used

3.3.3 Colour Schema

Even though the colour might change depending on some specific business rules for a given part of information but in general the following colours and fonts will be used.

Fonts: *TRUETYPEFONT*
Sizes: 15pt, (however for the titles, it might be possible to have 25pt).
Colour: Self.

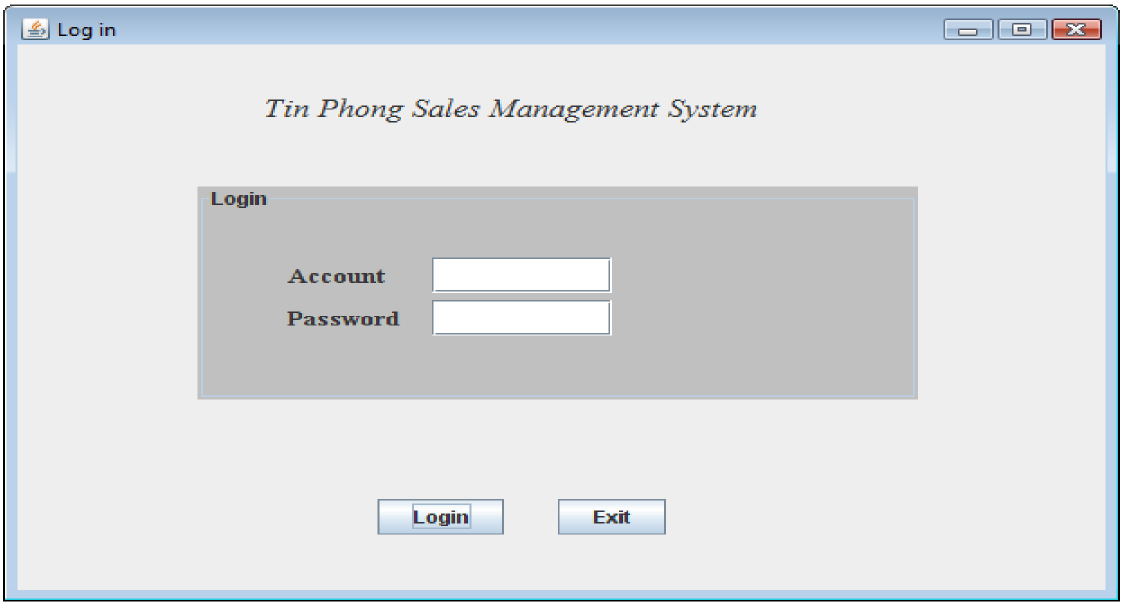
For the windows the background colour has been set to: Light gray

For the sake of highlighting the importance of a given control, some controls, are shadowed or marked with other colours. This is because it is thought that users normal pay attention to a control that looks differently from the group.

3.3.4 Windows description

The purpose is to describe windows that are used to display information to the actor, what can be done with each of them (the services that are offered by the window). In parentheses there is a java name that makes reference to the java class name in source codes while going to the application.

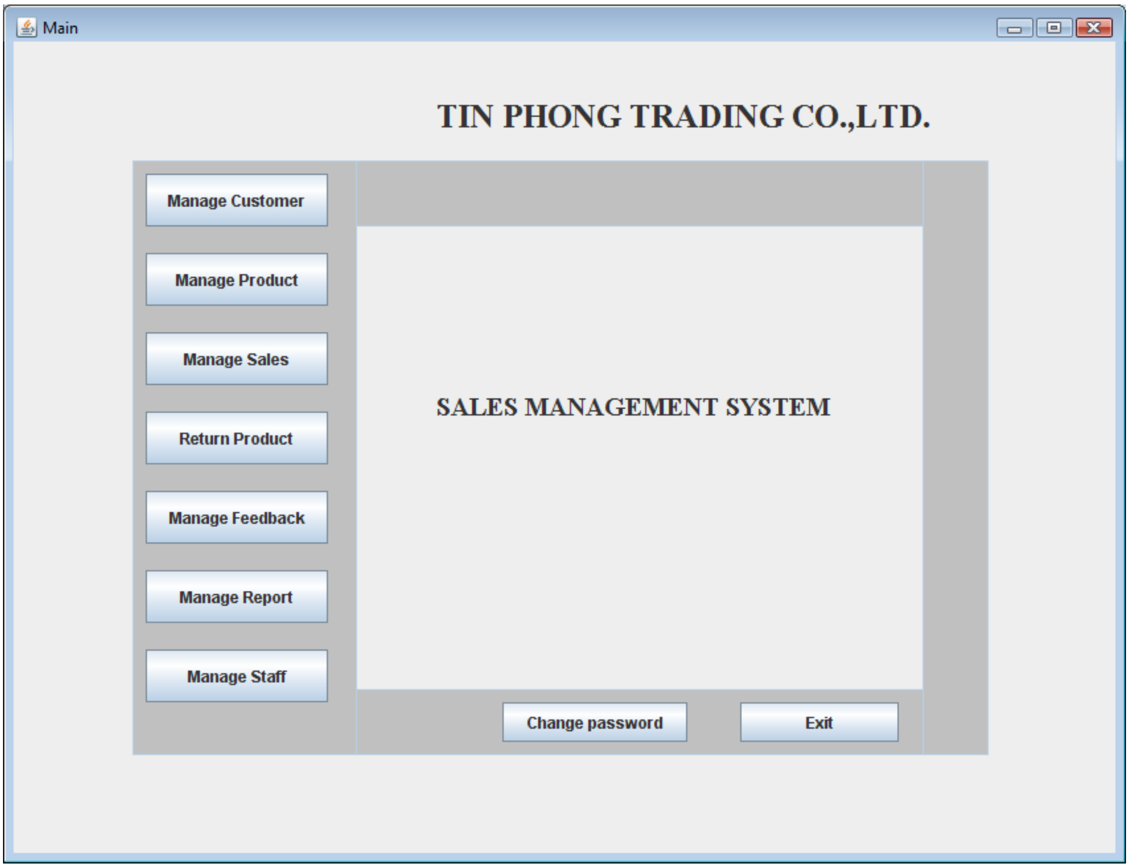
Log in window (LogIn.java)



Picture 3: Log in window

When the application is run, the first window the user sees is the Log in window. This window allows the getting into system or prohibits it if the authentication has failed.

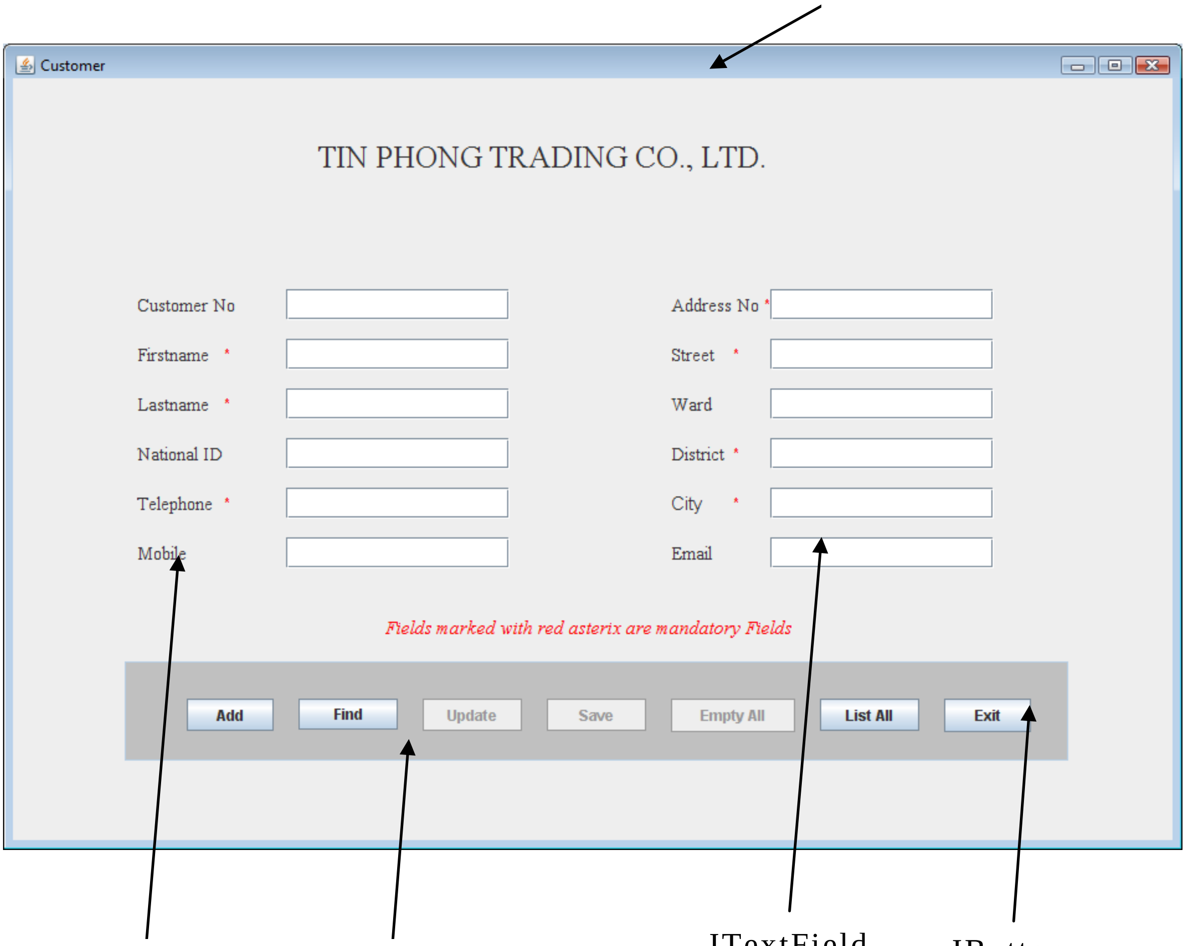
-Main window (MainWindow.java)



Picture 4: Main window

The Main window as the name sounds holds all primary navigations. From this window you can display other windows according to the actor's needs, the commands buttons on the window above show in details through which windows you can navigate from this window (services that can be done).

-Customer window (CustomerWindow.java)



Menu with Title

Fields marked with red asterix are mandatory Fields

JLabel

JPanel

JTextField

JButton

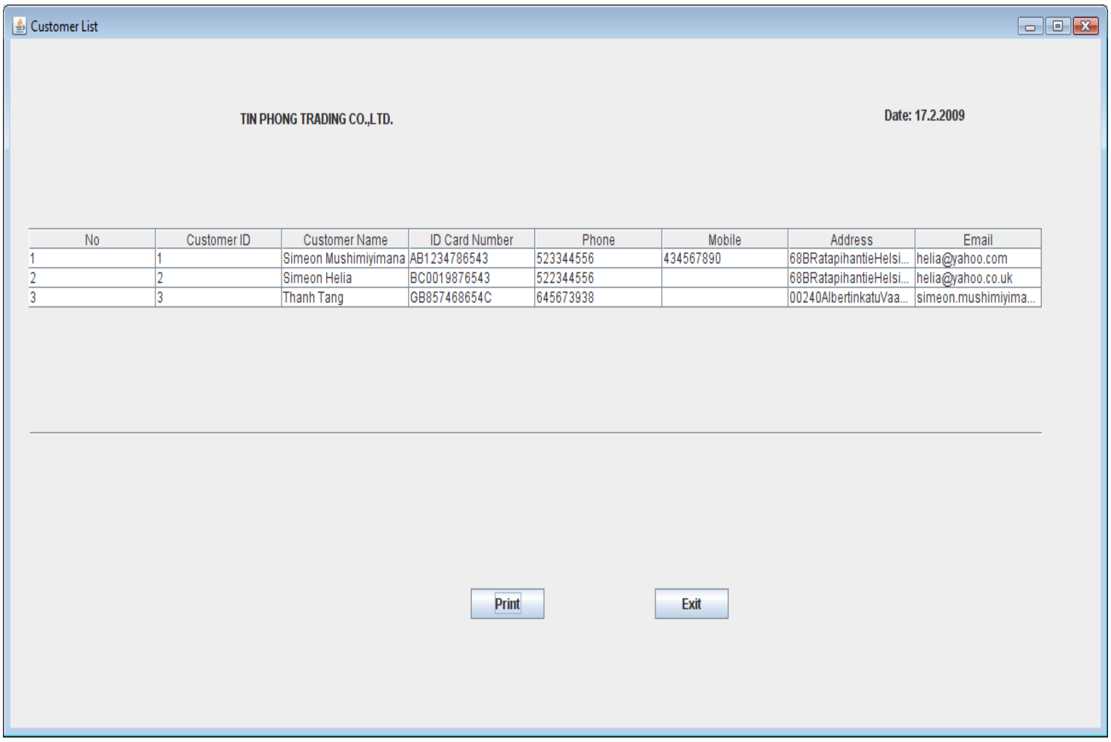
Picture 5: Customer window

The Customer window as the name specifies, is used for managing customer data.

The managing customer data includes, adding customer into the system, search existing customer from the system which may bring a list of customers or one customer information depending on searching criteria, it serves also for updating customer data.

Behind the Customer window there are other sub-windows that together with this window offer services of displaying customer data these sub-windows include;

-CustomerList sub-window (CustomerListWindow.java).



TIN PHONG TRADING CO.,LTD.

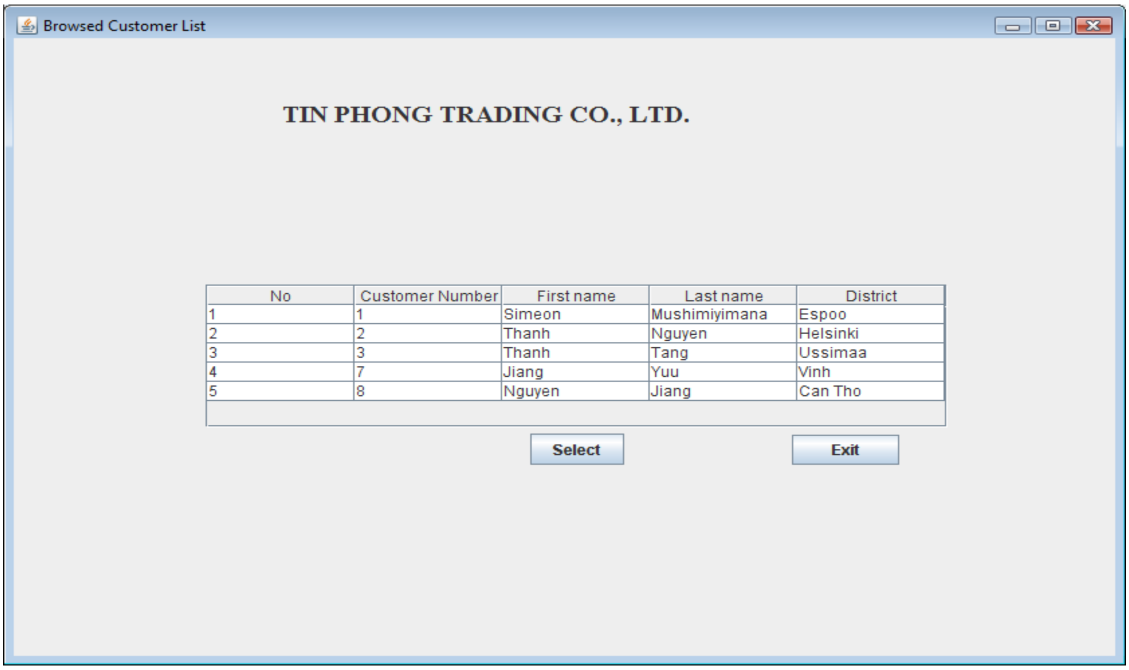
Date: 17.2.2009

No	Customer ID	Customer Name	ID Card Number	Phone	Mobile	Address	Email
1	1	Simeon Mushimiyimana	AB1234786543	523344556	434567890	68BRatapihantieHelsi...	helia@yahoo.com
2	2	Simeon Helia	BC0019876543	522344556		68BRatapihantieHelsi...	helia@yahoo.co.uk
3	3	Thanh Tang	GB857468654C	645673938		00240AlbertinkatuVaa...	simeon.mushimiyima...

Picture 6: Customer list window

CustomerList is a sub-window that displays a list all customers' needed data from the system and gives the option to print the displayed list.

-Browsed Customer list sub-window (BrowsedCustomerList.java)

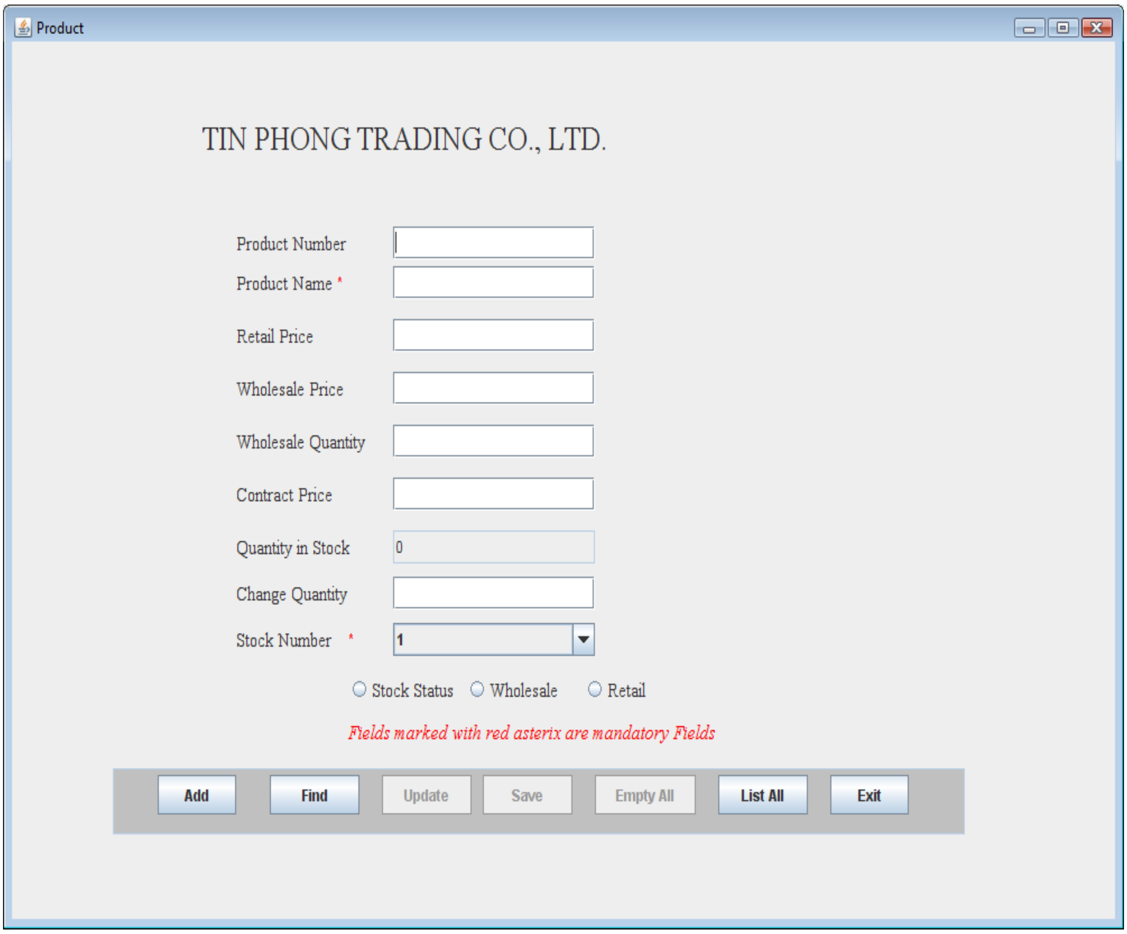


No	Customer Number	First name	Last name	District
1	1	Simeon	Mushimiyimana	Espoo
2	2	Thanh	Nguyen	Helsinki
3	3	Thanh	Tang	Ussimaa
4	7	Jiang	Yuu	Vinh
5	8	Nguyen	Jiang	Can Tho

Picture 7: Browsed Customer list window

BrowsedCustomerList sub-window lists list of customers found when a search using some keys to filter information is performed. The list

-Product window (ProductWindow.java)



Product

TIN PHONG TRADING CO., LTD.

Product Number
Product Name *
Retail Price
Wholesale Price
Wholesale Quantity
Contract Price
Quantity in Stock: 0
Change Quantity
Stock Number *: 1

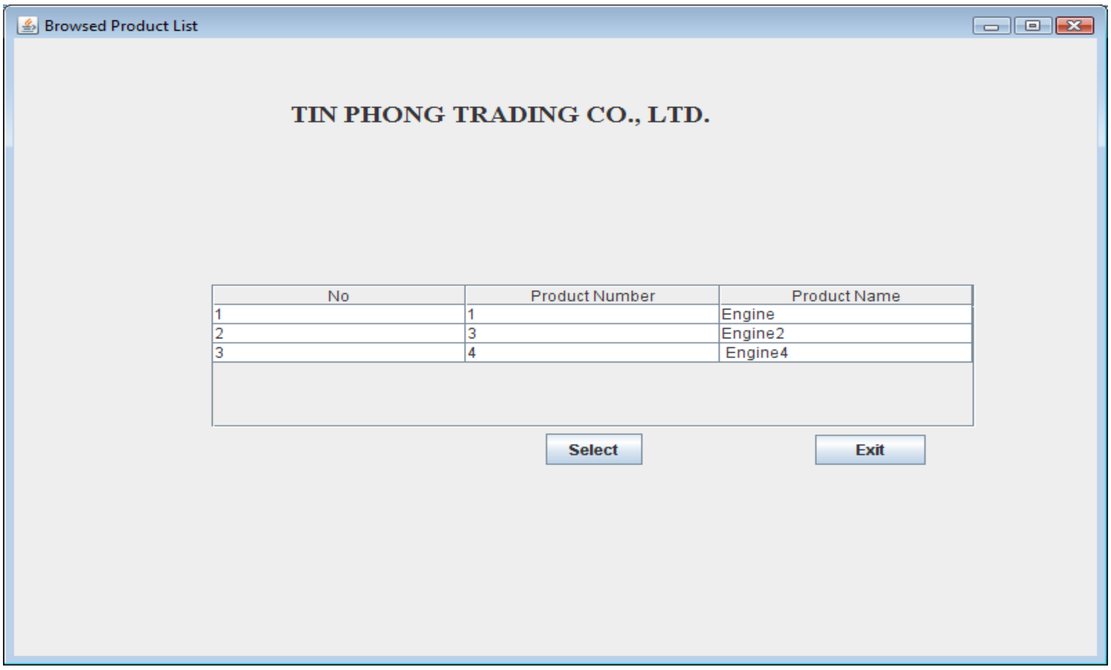
○ Stock Status ○ Wholesale ○ Retail

Fields marked with red asterix are mandatory Fields

Picture 8: Product window

With the Product window the Actor can add new product, update, search product information or display a list of all existing products in the system.

-Browsed Product list sub-window (BrowsedProductList.java)



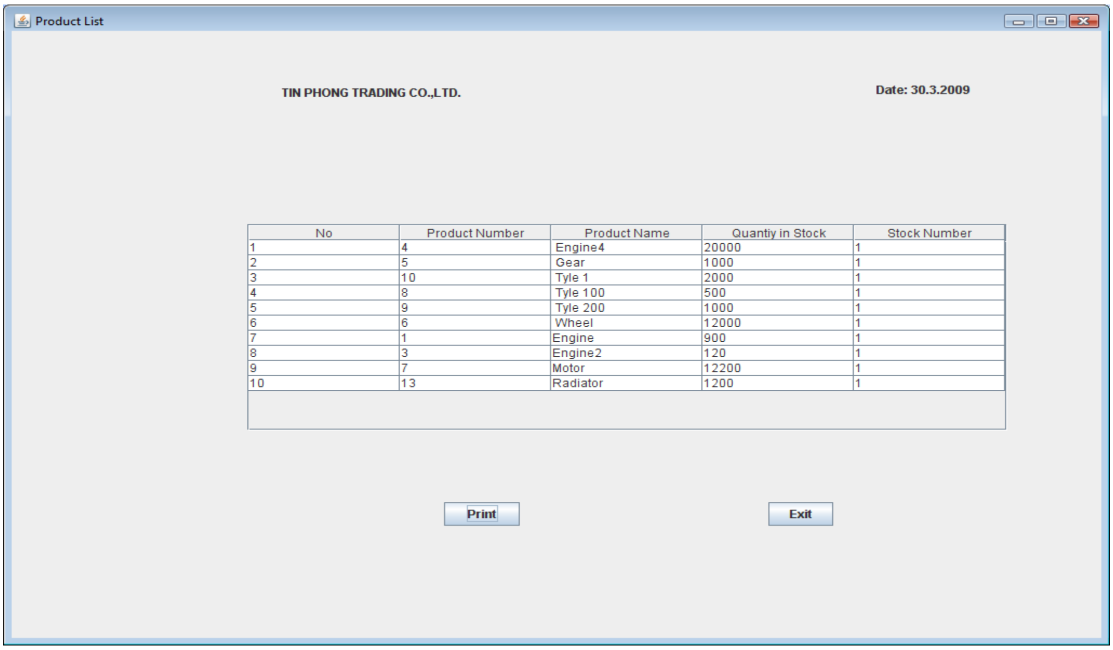
TIN PHONG TRADING CO., LTD.

No	Product Number	Product Name
1	1	Engine
2	3	Engine2
3	4	Engine4

Picture 9: Browsed Product list window

This window serves for selecting one specific product from the list of product found when a search is made.

-Product list sub-window (PrintableProductList.java)



TIN PHONG TRADING CO.,LTD.

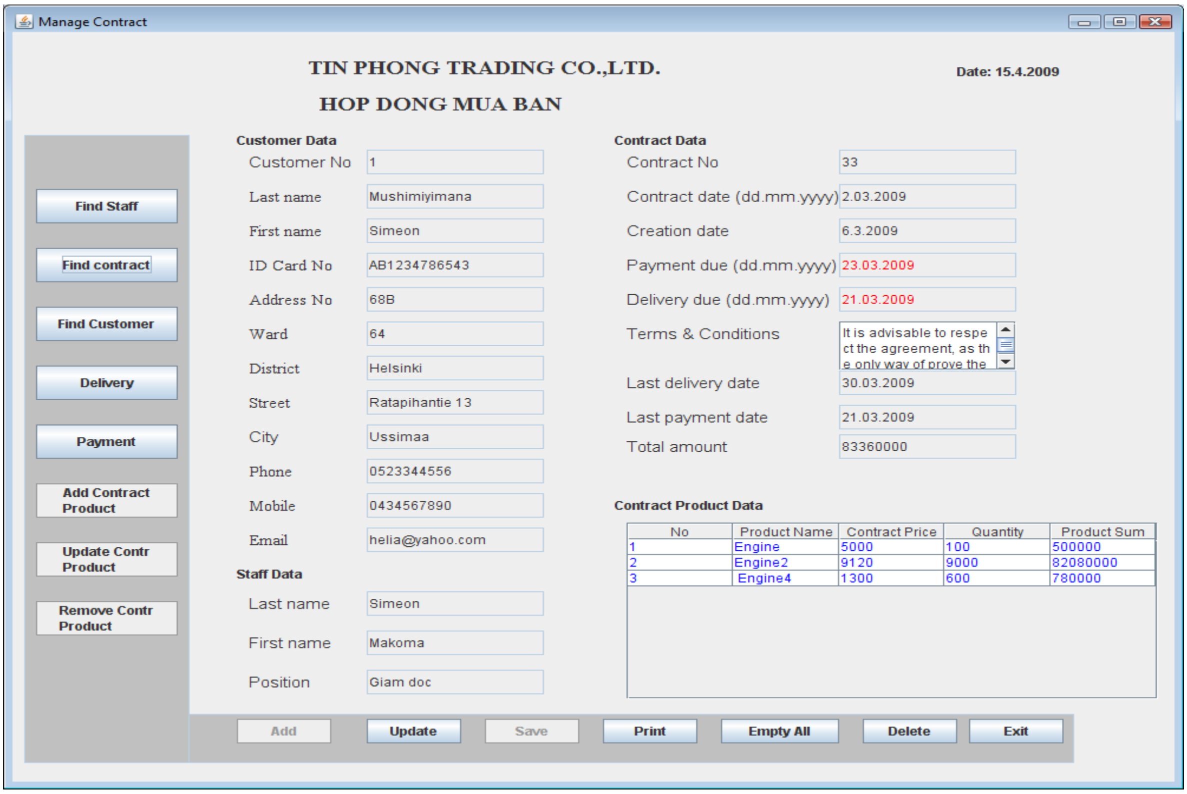
Date: 30.3.2009

No	Product Number	Product Name	Quantiy in Stock	Stock Number
1	4	Engine4	20000	1
2	5	Gear	1000	1
3	10	Tyle 1	2000	1
4	8	Tyle 100	500	1
5	9	Tyle 200	1000	1
6	6	Wheel	12000	1
7	1	Engine	900	1
8	3	Engine2	120	1
9	7	Motor	12200	1
10	13	Radiator	1200	1

Picture 10: Product list window

The product list window is a window that prints all products in the system, and it lets you print the shown list.

-Sales window (SalesWindow.java)



TIN PHONG TRADING CO.,LTD.

HOP DONG MUA BAN

Date: 15.4.2009

Customer Data

Customer No	1
Last name	Mushimiyimana
First name	Simeon
ID Card No	AB1234786543
Address No	68B
Ward	64
District	Helsinki
Street	Ratapihantie 13
City	Ussimaa
Phone	0523344556
Mobile	0434567890
Email	helia@yahoo.com

Staff Data

Last name	Simeon
First name	Makoma
Position	Giam doc

Contract Data

Contract No	33
Contract date (dd.mm.yyyy)	2.03.2009
Creation date	6.3.2009
Payment due (dd.mm.yyyy)	23.03.2009
Delivery due (dd.mm.yyyy)	21.03.2009
Terms & Conditions	It is advisable to respect the agreement, as the only way of prove the
Last delivery date	30.03.2009
Last payment date	21.03.2009
Total amount	83360000

Contract Product Data

No	Product Name	Contract Price	Quantity	Product Sum
1	Engine	5000	100	500000
2	Engine2	9120	9000	82080000
3	Engine4	1300	600	780000

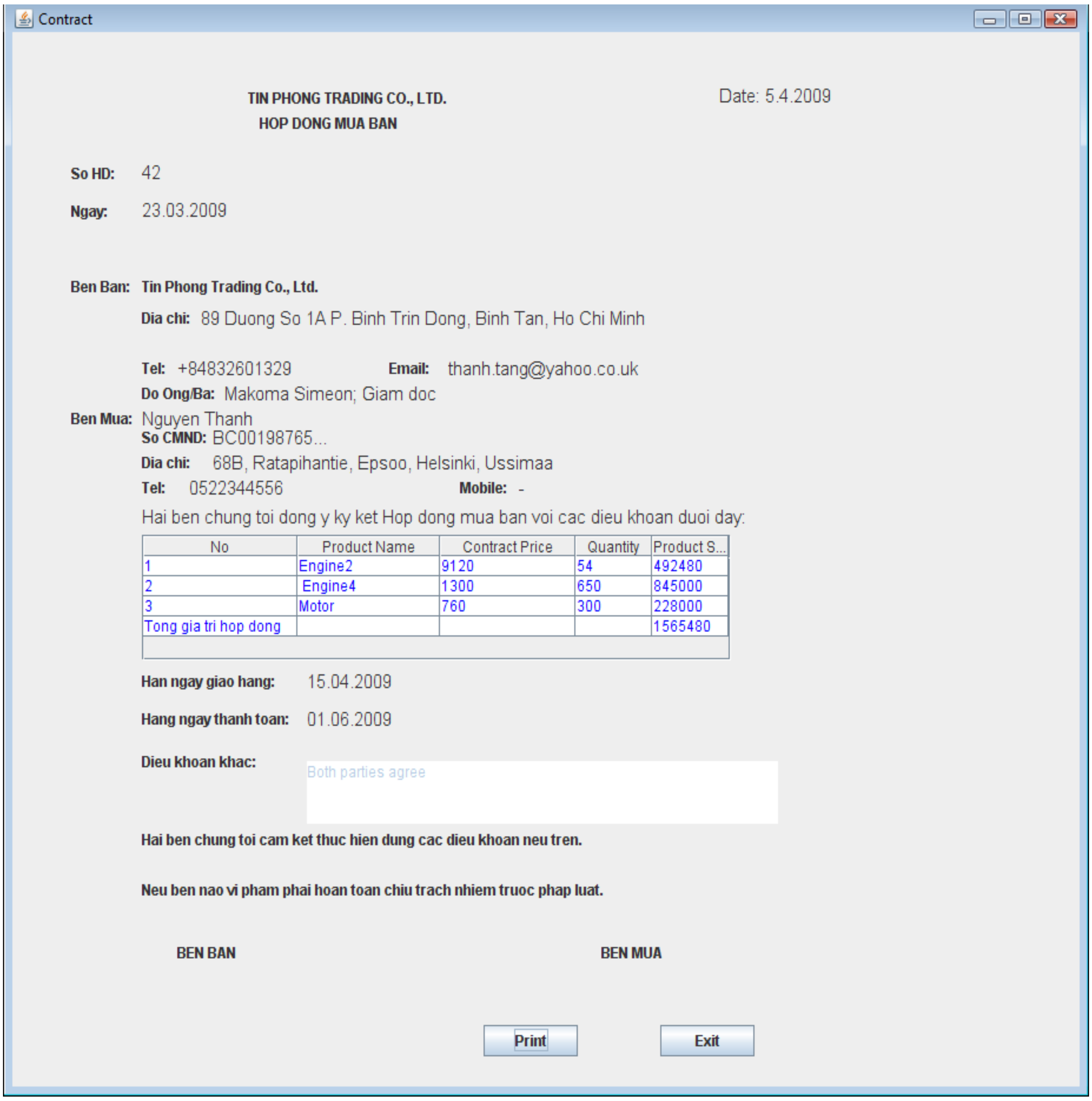
Picture 11: Sales window

This is the complex window that allows you search customer, staff and product for adding new contract; it also allows searching for contract data if the contract exists in the system.

With this window you can update the existing contract or delete it according to business rules. It might be possible to handle payment or delivery, the SalesWindow gives an option to do that by opening Delivery window or Payment window according to the user needs. Furthermore, you can display the printable format of the existing contract from this Sales window.

With this window it is possible to update contract product, which include reducing quantity, removing any product or adding new product to the list of contract products.

-Printable format contract sub-window (PrintableContractWindow.java)



TIN PHONG TRADING CO., LTD.
HOP DONG MUA BAN

Date: 5.4.2009

So HD: 42

Ngay: 23.03.2009

Ben Ban: Tin Phong Trading Co., Ltd.
Dia chi: 89 Duong So 1A P. Binh Trin Dong, Binh Tan, Ho Chi Minh
Tel: +84832601329 Email: thanh.tang@yahoo.co.uk
Do Ong/Ba: Makoma Simeon; Giam doc

Ben Mua: Nguyen Thanh
So CMND: BC00198765...
Dia chi: 68B, Ratapihantie, Epsoo, Helsinki, Ussimaa
Tel: 0522344556 Mobile: -

Hai ben chung toi dong y ky ket Hop dong mua ban voi cac dieu khoan duoi day:

No	Product Name	Contract Price	Quantity	Product S...
1	Engine2	9120	54	492480
2	Engine4	1300	650	845000
3	Motor	760	300	228000
Tong gia tri hop dong				1565480

Han ngay giao hang: 15.04.2009

Hang ngay thanh toan: 01.06.2009

Dieu khoan khac: Both parties agree

Hai ben chung toi cam ket thuc hien dung cac dieu khoan neu tren.

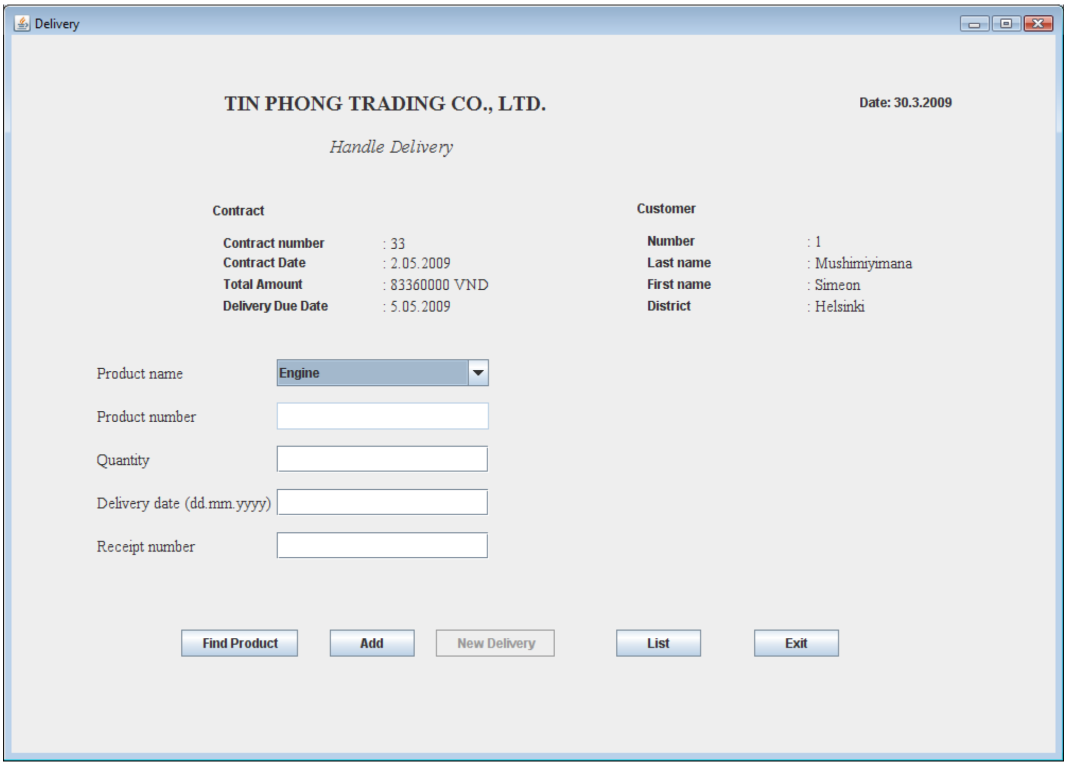
Neu ben nao vi pham phai hoan toan chiu trach nhiem truoc phap luat.

BEN BAN BEN MUA

Picture 12: Contract window

The contract sub-window displayed when the Print button on Sales window is clicked, is to show the contract print preview. This is because all contract data is not printed out, the only needed data is transferred from Sales window to the contract sub-window to be printed out.

-Delivery sub-window (DeliveryWindow.java)



Contract

Contract number	: 33
Contract Date	: 2.05.2009
Total Amount	: 83360000 VND
Delivery Due Date	: 5.05.2009

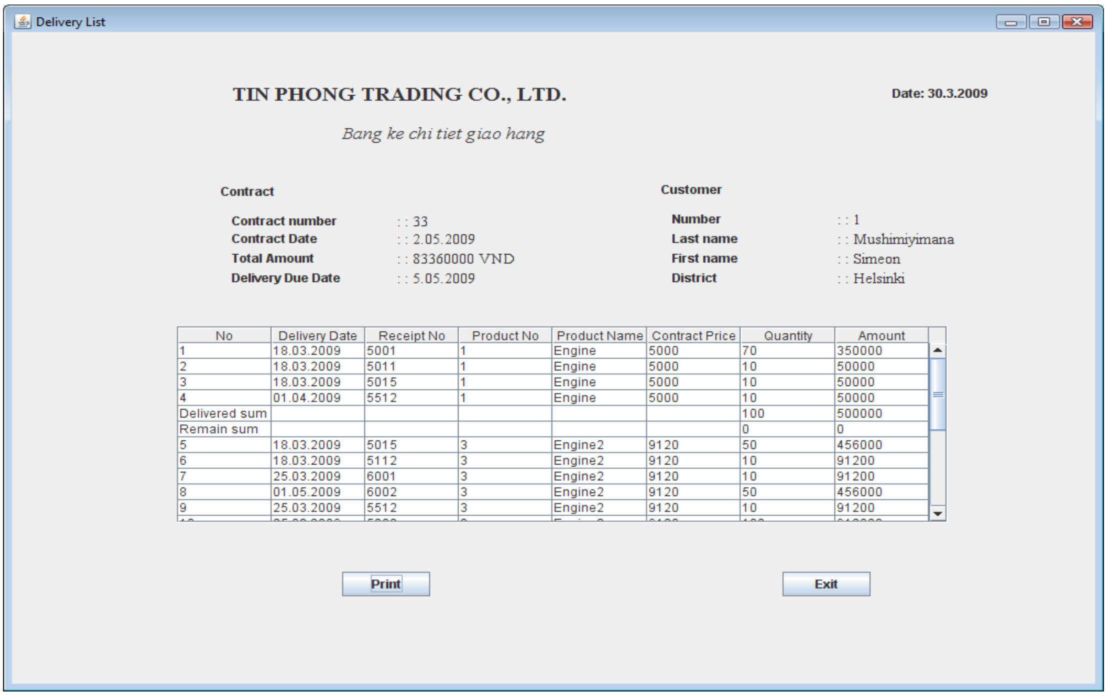
Customer

Number	: 1
Last name	: Mushimiyimana
First name	: Simeon
District	: Helsinki

Picture 13: Delivery window

With delivery window you can make new delivery or display a list of deliveries for a specific contract, it is also used to find information about product that is needed for delivering.

-Delivery list sub-window (DeliveryListWindow.java)



TIN PHONG TRADING CO., LTD.

Date: 30.3.2009

Bang ke chi tiet giao hang

Contract

Contract number	: : 33
Contract Date	: : 2.05.2009
Total Amount	: : 83360000 VND
Delivery Due Date	: : 5.05.2009

Customer

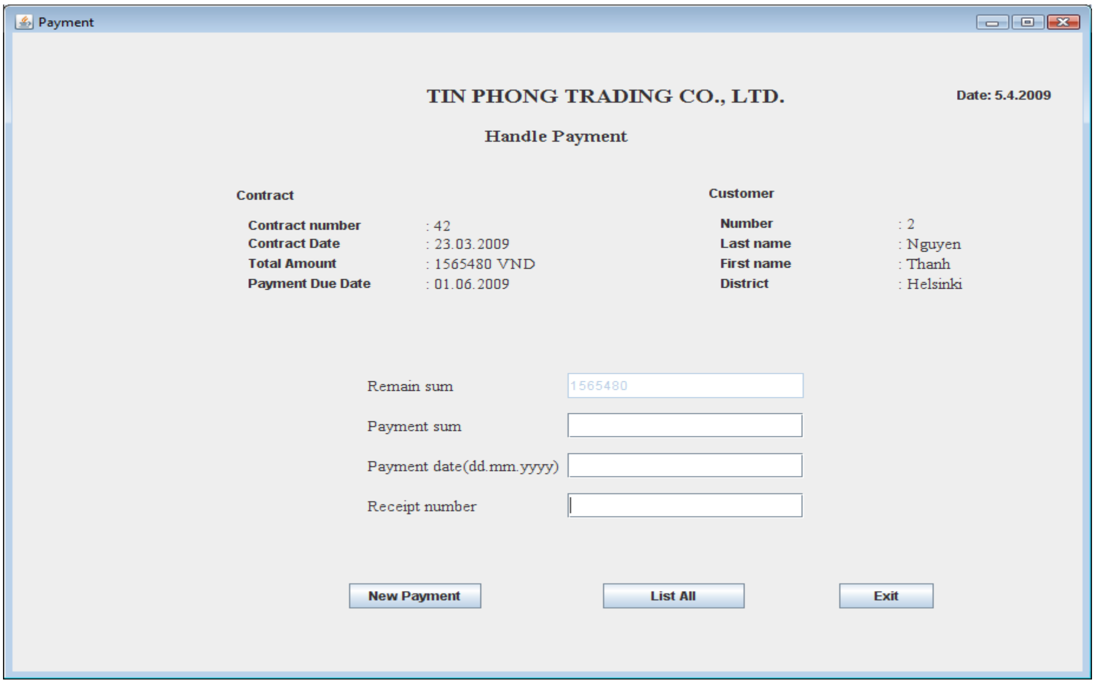
Number	: : 1
Last name	: : Mushimiyimana
First name	: : Simeon
District	: : Helsinki

No	Delivery Date	Receipt No	Product No	Product Name	Contract Price	Quantity	Amount
1	18.03.2009	5001	1	Engine	5000	70	350000
2	18.03.2009	5011	1	Engine	5000	10	50000
3	18.03.2009	5015	1	Engine	5000	10	50000
4	01.04.2009	5512	1	Engine	5000	10	50000
Delivered sum						100	500000
Remain sum						0	0
5	18.03.2009	5015	3	Engine2	9120	50	456000
6	18.03.2009	5112	3	Engine2	9120	10	91200
7	25.03.2009	6001	3	Engine2	9120	10	91200
8	01.05.2009	6002	3	Engine2	9120	50	456000
9	25.03.2009	5512	3	Engine2	9120	10	91200

Picture 14: Delivery list window

This Delivery list window displays all deliveries relating to a specific product in one contract, it also allows printing the list of that information found.

-Payment sub-window (PaymentWindow.java)



Payment

TIN PHONG TRADING CO., LTD.

Date: 5.4.2009

Handle Payment

Contract	
Contract number	: 42
Contract Date	: 23.03.2009
Total Amount	: 1565480 VND
Payment Due Date	: 01.06.2009

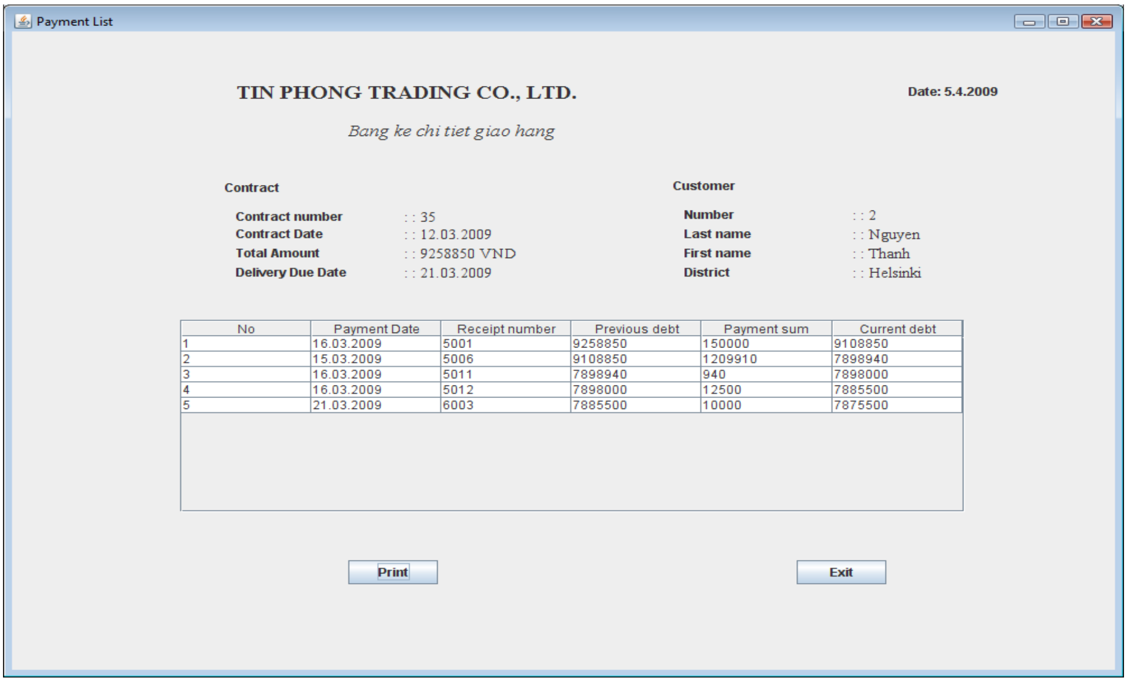
Customer	
Number	: 2
Last name	: Nguyen
First name	: Thanh
District	: Helsinki

Remain sum	1565480
Payment sum	
Payment date(dd.mm.yyyy)	
Receipt number	

Picture 15: Payment window

This Payment sub-window gives an option of adding new payment; list all payments relating to the contract to which the being made payment belongs.

-Payment list sub-window (PaymentListWindow.java)



TIN PHONG TRADING CO., LTD.

Date: 5.4.2009

Bang ke chi tiet giao hang

Contract

Contract number	: : 35
Contract Date	: : 12.03.2009
Total Amount	: : 9258850 VND
Delivery Due Date	: : 21.03.2009

Customer

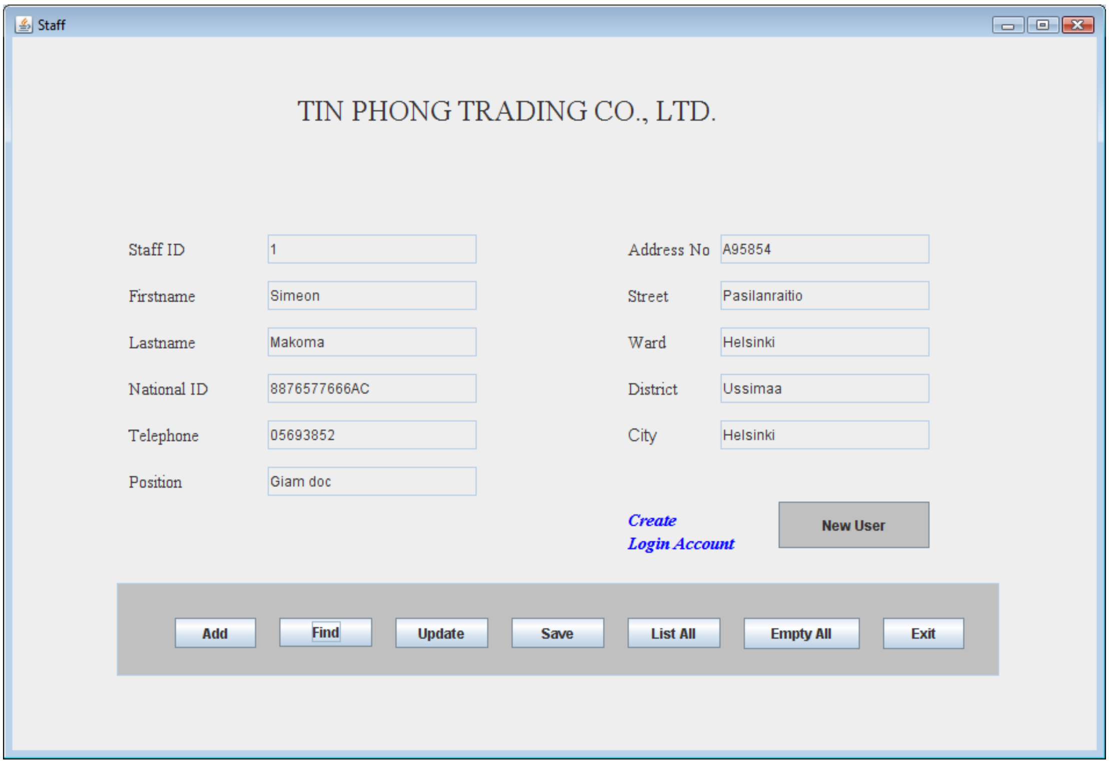
Number	: : 2
Last name	: : Nguyen
First name	: : Thanh
District	: : Helsinki

No	Payment Date	Receipt number	Previous debt	Payment sum	Current debt
1	16.03.2009	5001	9258850	150000	9108850
2	15.03.2009	5006	9108850	1209910	7898940
3	16.03.2009	5011	7898940	940	7898000
4	16.03.2009	5012	7898000	12500	7885500
5	21.03.2009	6003	7885500	10000	7875500

Picture 16: Payment list window

Payment list sub-window displays a list of all payments made for a given contract, you can easily see the current debt as well as dates of all previous payments and respectively the amount related to these.

- Staff window (StaffWindow.java)



Staff

TIN PHONG TRADING CO., LTD.

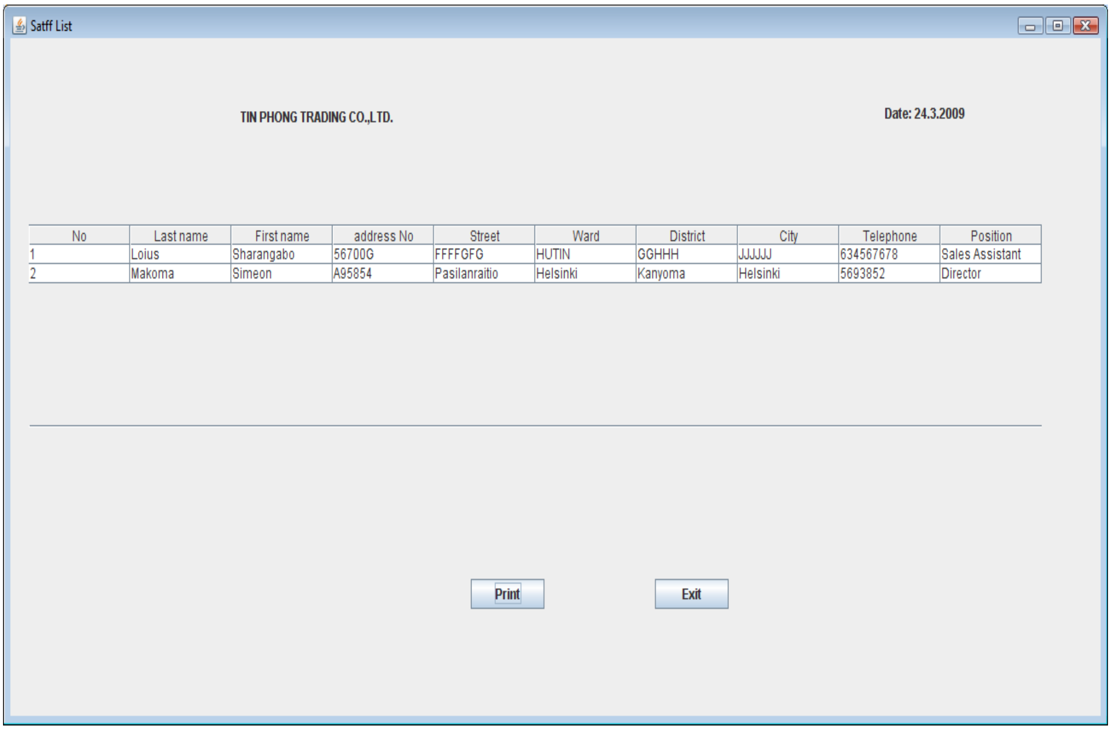
Staff ID	1	Address No	A95854
Firstname	Simeon	Street	Pasilanraitio
Lastname	Makoma	Ward	Helsinki
National ID	8876577666AC	District	Ussimaa
Telephone	05693852	City	Helsinki
Position	Giam doc		

Picture 17: Staff window

With Staff window the new staff information is entered in the system, the existing staff data can be updated, the Actor can search staff information using either staff first name or staff number.

Once a new staff has been created it is possible to create the log in account for that staff. So, from the staff window the user sub window is opened from which new account is created.

-Staff list sub-window (StaffListWindow.java)



TIN PHONG TRADING CO.,LTD.

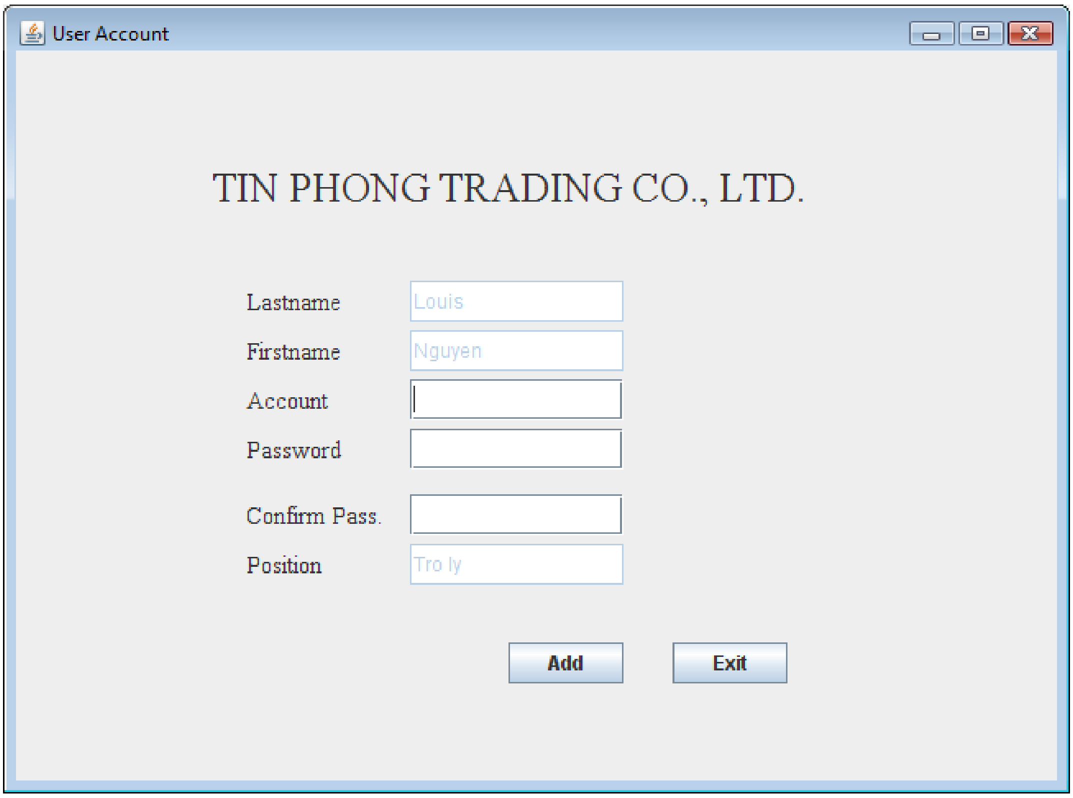
Date: 24.3.2009

No	Last name	First name	address No	Street	Ward	District	City	Telephone	Position
1	Loius	Sharangabo	56700G	FFFFGFG	HUTIN	GGHHH	JJJJJJ	634567678	Sales Assistant
2	Makoma	Simeon	A95854	Pasilanraitio	Helsinki	Kanyoma	Helsinki	5693852	Director

Picture 18: Staff list window

This sub-window displays all existing staff's information either for viewing or printing.

-User Account sub-window (ManageUserAccount.java)



User Account

TIN PHONG TRADING CO., LTD.

Lastname: Louis

Firstname: Nguyen

Account:

Password:

Confirm Pass.:

Position: Tro ly

The purpose of this sub-window is to create a user's account, for security sake each user is given an account and password to use while accessing the system

3.4 Sequence diagrams

3.4.1 Sequence diagram description

The sequence diagram, in other words interaction diagram describes the interaction between the actor and the system, meaning the actor's requests and the system's responses to these requests, it also describes how processes interact with one another and in which order. Actual when talking about process interaction we are at the same time talking about objects communication.

The request is shown on diagram, which message it carries (variables) and the answer is also shown, the answer might be any information, a message that something went wrong.

To make our diagrams readable and shorten the space they could take, it has been decided to describe only the communication for GUI to the Transaction Class, the rest such as how ResultSet manages data or how the Transaction is managed in database is left out as it is common almost to all desktop applications using RDBM (Relational Database Management System). Meanwhile, the sequence diagrams for Product and Staff are left out as they are performed in the same way as Customer Diagrams, by this; it means that when one wants to add, update, retrieve Product or Staff information, the Customer respectively related diagram will serve as an integral example.

In general the sequence diagrams described concerned with;

-**Adding new information in the system**

-Retrieving the existing information according to search criteria

-Updating information

-Deleting information

Because of time limit sequence diagrams for Payment and Delivery are not described here.

3.4.2 Add Customer sequence diagram

See appendix 1. Add Customer Sequence diagram.

3.4.3 Find Customer sequence diagram

See appendix 2. Find Customer Sequence diagram.

3.4.4 Update Customer sequence diagram

See appendix 3. Update Customer Sequence diagram.

3.4.5 Add Contract sequence diagram

See appendix 4. Add Contract Sequence diagram.

3.4.6 Find Contract sequence diagram

See appendix 5. Find Contract Sequence diagram.

3.4.7 Update Contract sequence diagram

See appendix 6. Update Contract Sequence diagram.

3.4.8 Delete Contract sequence diagram

See appendix 7. Delete Contract Sequence diagram.

4 Database design

4.1 Database design model

Database design model or logical model shows tables and entities of the database in a state before there are really transformed in database schema which is normal generated by the tool itself, it has been decided to show the database design model here because the tool used here does not give opportunities to see all fields types at the same time.

See appendix 9: Database design model

4.2. Database tables and fields descriptions

The idea of this description is to show database tables as in database schema, fields they contain and type and the length of these fields as well as some rules applied. Full names of fields are not given here but full description of each field is given to tell what the field stands for in the table. It also has the role of showing if the constraints were respected, meaning if a given field is parent and there is a child field in other table the child should preserve the type of the parent as well as the length in order words to enforce the field's integrity.

For some fields these is so called initial value, this value is the value from which the system will be adding or increasing when new data is added. In this case the system takes care of that otherwise if no initial value the system will store input data, the system also check the null constraints, if a given field is not to store null values and the system receives a null value the last will be rejected and thus an exception caught.

4.2.1 Table ContractSale

Attribute	Description	Type	Length	Initial value	Nulls
contractId	Unique identification number of a contract	Integer	4 digits	0001	Not null
customerId	identification of the customer owner of this contract	Integer	4 digits		Not null
staffId	identification number of the staff who makes the contract	Integer	3 digits		Not null
contractDate	date on which the contract is signed	Date			Not null
terms_Conditions	contract agreement terms and conditions	String	Maximum characters		Not null
paymentDueDate	last date on which the contract should have been paid	Date			Not null
deliveryDueDate	last date on which the contract should have been delivered	Date			Not null
totalAmount	district where the staff resides	Long integer	15 digits		Not null
contractCreation-Date	sytem date on which a contract is entered in the system.	Date			Not null
lastDeliveryDate	Date, when was the last product was delivered	Date			
lastPaymentDate	Date, when the last payment was made	Date			

4.2.2 Table ContractDelivery

Attribute	Description	Type	Length	Initial value	Nulls
contractDeliveryId	unique payment identification number	Integer	4 digits	0001	Not null
contractId	contract identification number	Integer	4 digits		Not null
deliveryDate	date on which the delivery is made	Date			Not null
deliveryReceiptNo	number identifying the receipt for the delivery	Integer	4 digits		Not null

4.2.3 Table ContractDeliveryContractProduct

Attribute	Description	Type	Length	Initial value	Nulls
contractDelivery-ProductId	Contract-DeliveryProduct identification number	Integer	4 digits	0001	Not null
contractId	contract identification number	Integer	4 digits		Not null
productId	product identification number	Integer	4 digits		Not null

4.2.4 Table ContractDeliveryProduct

Attribute	Description	Type	Length	Initial value	Nulls
contractDelivery-ProductId	unique identification number of the Contract-DeliveryProduct	Integer	4 digits	0001	Not null
productId	product identification number	Integer	4 digits		Not null
contractDeliveryId	contractDelivery identification number	Integer	4 digits		Not null
deliveryPrice	amount(money) of delivered quantity	Long integer	15 digits		Not null
contractDelivery-Quantity	delivered quantity	Integer	6 digits		Not null

4.2.5 Table ContractProduct

Attribute	Description	Type	Length	Initial value	Nulls
contractId	identification number of a contract	Integer	4 digits	0001	Not null
productId	identification of the product	Integer	4 digits		Not null
contractPrice	price of the product	Long integer	8 digits		Not null
contractQuantity	quantity of the product	Integer	6 digits		Not null
contractDelv-Quantity	remnant quantity to be delivered of the product	Integer	6 digits		Not null
contractDelv-TotalQuantity	quantity delivered of the product	Integer	13 digits		
contractDelivDate	last delivery date of the product	Date			
productSum	the amount(in money) of the product	Integer	15 digits		Not null

4.2.6 Table Customer

Attribute	Description	Type	Length	Initial value	Nulls
customerId	Unique identification number of a customer	Integer	4 digits	0001	Not null
firstname	first name of a customer	String	30 characters		Not null
lastname	last name of a customer	String	20 characters		Not null
idNumber	National identification number	String	12 characters		
addressNo	address number where the customer resides	String	15 characters		Not null
street	street where the customer resides	String	20 characters		Not null
ward	ward where the customer resides	String	15 characters		
district	district where the customer resides	String	15 characters		Not null
city	city where the customer resides	String	15 characters		Not null
phone	customer's line phone number	Long integer	10 digits		Not null
cell	customer's Mobile phone number	Long integer	10 digits		
email	customer's electronic address	String	60 characters		

4.2.7 Table LogIn

Attribute	Description	Type	Length	Initial value	Nulls
userId	unique identification number of stock	Integer	2digits	0001	Not null
password	the name of stock	String	10 characters		Not null
staffId	identification number of the staff to whom the account belongs	Integer	3 digits		Not null
position	the position of the staff in the company	String	20 characters		Not null

4.2.8 Table Payment

Attribute	Description	Type	Length	Initial value	Nulls
paymentId	unique payment identification number	Integer	4 digits	0001	Not null
customerId	customer identification number	Integer	4 digits		Not null
contractId	contract identification number	Long integer	4 digits		Not null
receiptNo	number of the receipt given to the customer	Integer	4 digits		Not null
paymentDate	date on which the payment is made	Date			Not null
previousDebt	debt before this payment is made	Long integer	15 digits		Not null
paymentSum	sum of money paid by the customer	Long integer	15 digits		Not null
currentDebt	amount to be paid	Long integer	15 digits		Not null

4.2.9 Table Product

Attribute	Description	Type	Length	Initial value	Nulls
productId	Unique identification number of a product	Long integer	4 digits	0001	Not null
stockId	Stokc identification number	Long integer	2 digits	01	Not null
productName	Name of the product	String	30 characters		Not null
retailPrice	Price of the product per unit for retailing.	Long integer	8 digits		
wholesalePrice	Price of the product per unit for wholesale	Long integer	8 digits		
wholesaleQuantity	Minimum quantity defined for wholesale	Integer	6 digits		
contractPrice	Price of the product per unit for contract sales	Long integer	8 digits		
quantiyInStock	Available quantity in Stock.	Integer	7 digits		

4.2.10 Table Staff

Attribute	Description	Type	Length	Initial value	Nulls
staffId	Unique identification number of a staff	Integer	4 digits	0001	Not null
firstname	first name of a staff	String	30 characters		Not null
lastname	last name of a staff	String	20 characters		Not null
idNumber	National identification number of a staff	String	12 characters		
addressNo	address number where the staff resides	String	10 characters		Not null
street	street where the staff resides	String	20 characters		Not null
ward	ward where the staff resides	String	15 characters		
district	district where the staff resides	String	15 characters		Not null
city	city where the staff resides	String	15 characters		Not null
phone	staff's line phone number	Long integer	10 digits		Not null
position	the post occupied by the staff in the company	String	20 characters		Not null

4.2.11 Table Stock

Attribute	Description	Type	Length	Initial value	Nulls
stock	unique identification number of stock	Integer	2digits	01	Not null
stockName	the name of stock	String	20 characters		Not null

4.2.12 Table StockEvent

Attribute	Description	Type	Length	Initial value	Nulls
stockEventId	Contract-DeliveryProduct identification number	Integer	4 digits	0001	Not null
contractDelivery-ProductId	contractDelivery-Product identification number	Integer	4 digits		
stockId	stock identification number	Integer	2digits		Not null
productId	product identification number	Integer	4 digits		Not null
eventDate	the date in which the event takes place	Date			Not null
quantity	product quantity in the event	Integer	6 digits		Not null

5. Test Planning

To ensure the system meets requirements specification, the system has to be tested thus the tests plan as well as test cases have to be made, for this application only three type of tests are made and these are:
-Unit or Module test
-Integration test
-System test

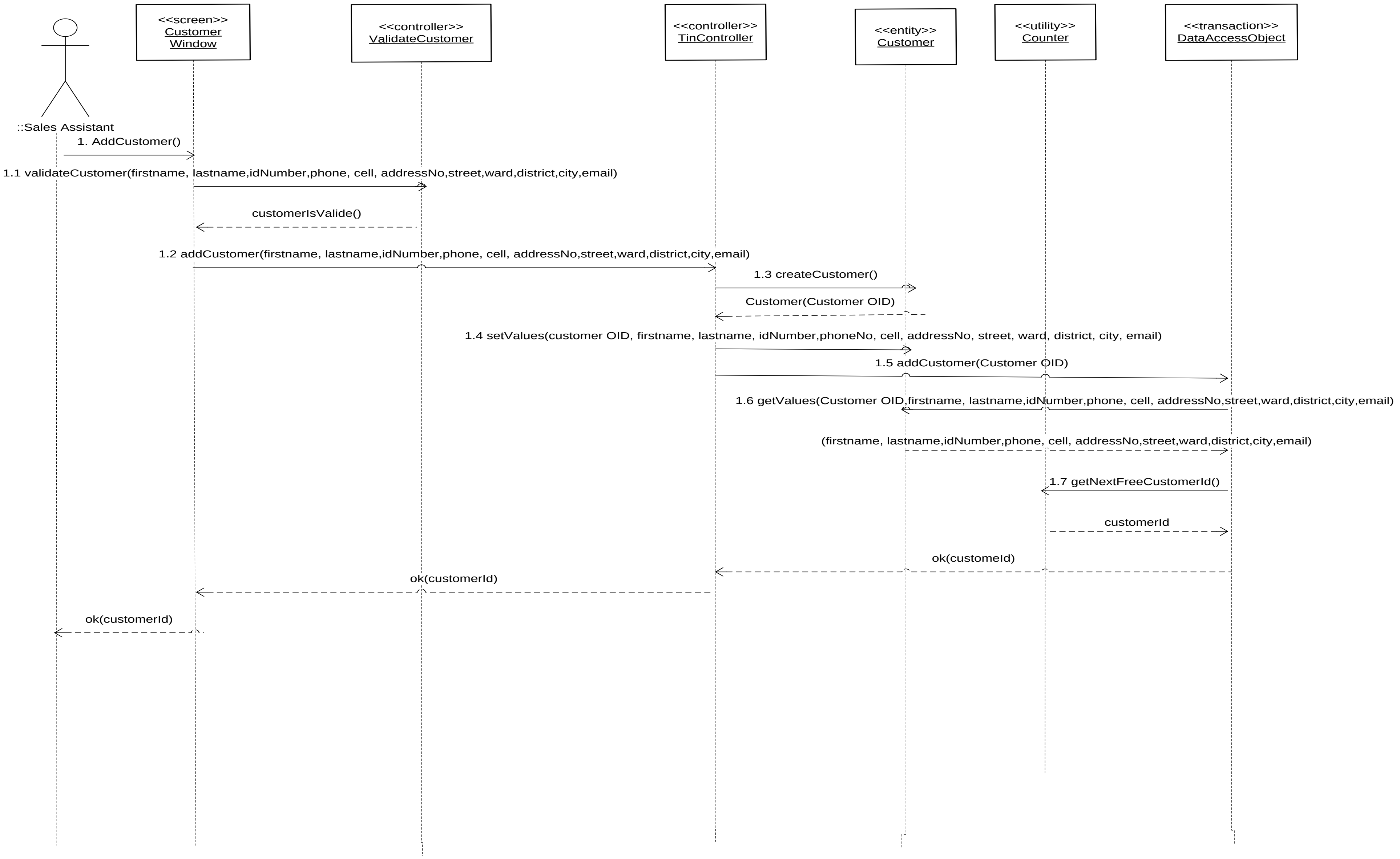
For this, it has to been understood that respectively three kinds testing plan have been made as we have these three kinds of tests.

-Unit test planning
-Integration test planning
- System test planning
More details see Appendix 10: Test plan

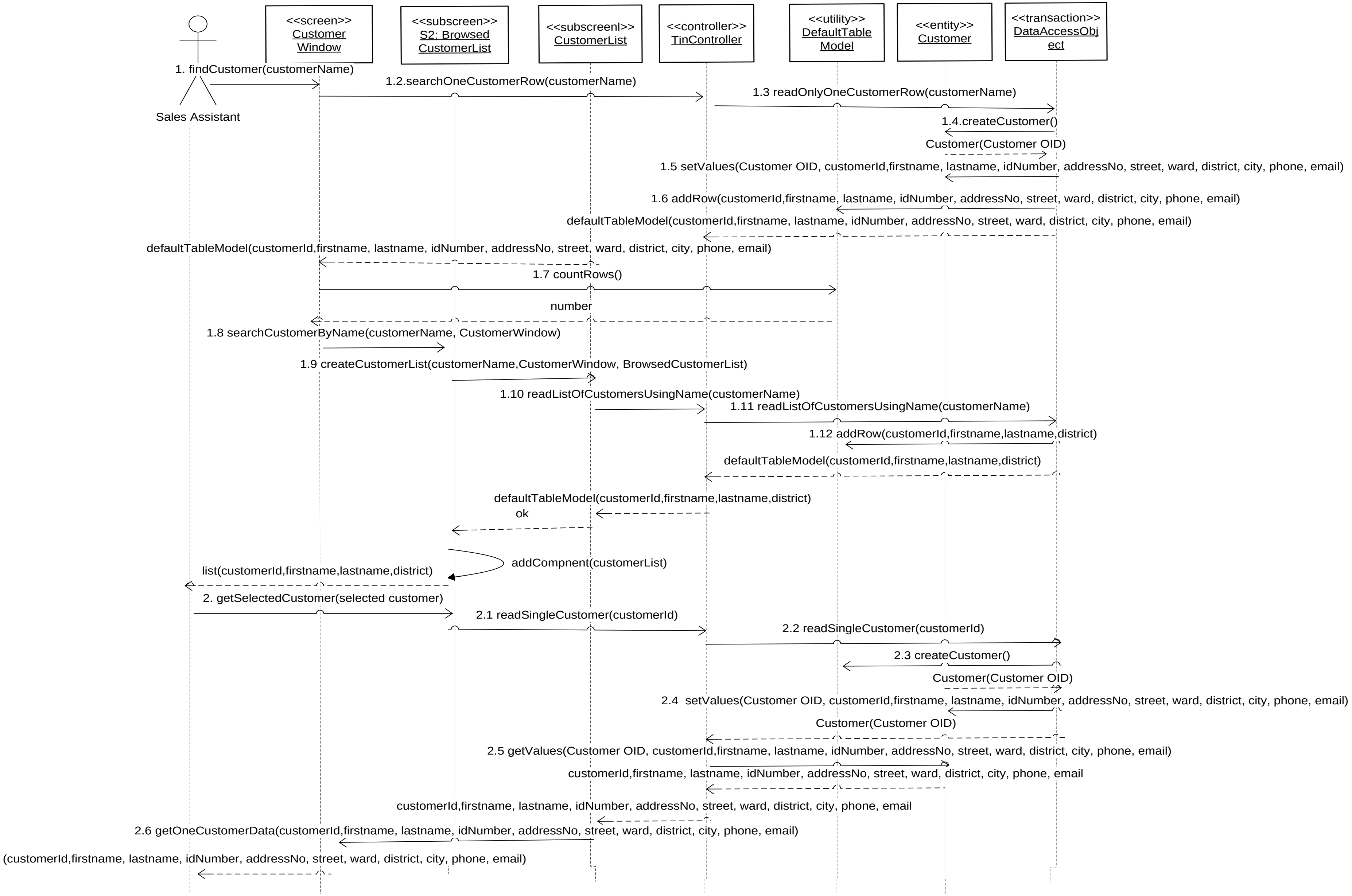
Appendices

Appendix 1: Add Customer sequence diagram
Appendix 2: Find Customer sequence diagram
Appendix 3: Update Customer sequence diagram
Appendix 4: Add Contract sequence diagram
Appendix 5: Find Contract sequence diagram
Appendix 6: Update Contract sequence diagram
Appendix 7: Delete Contract sequence diagram
Appendix 8: Database design model
Appendix 9: Test plan
Appendix 10: List of changes

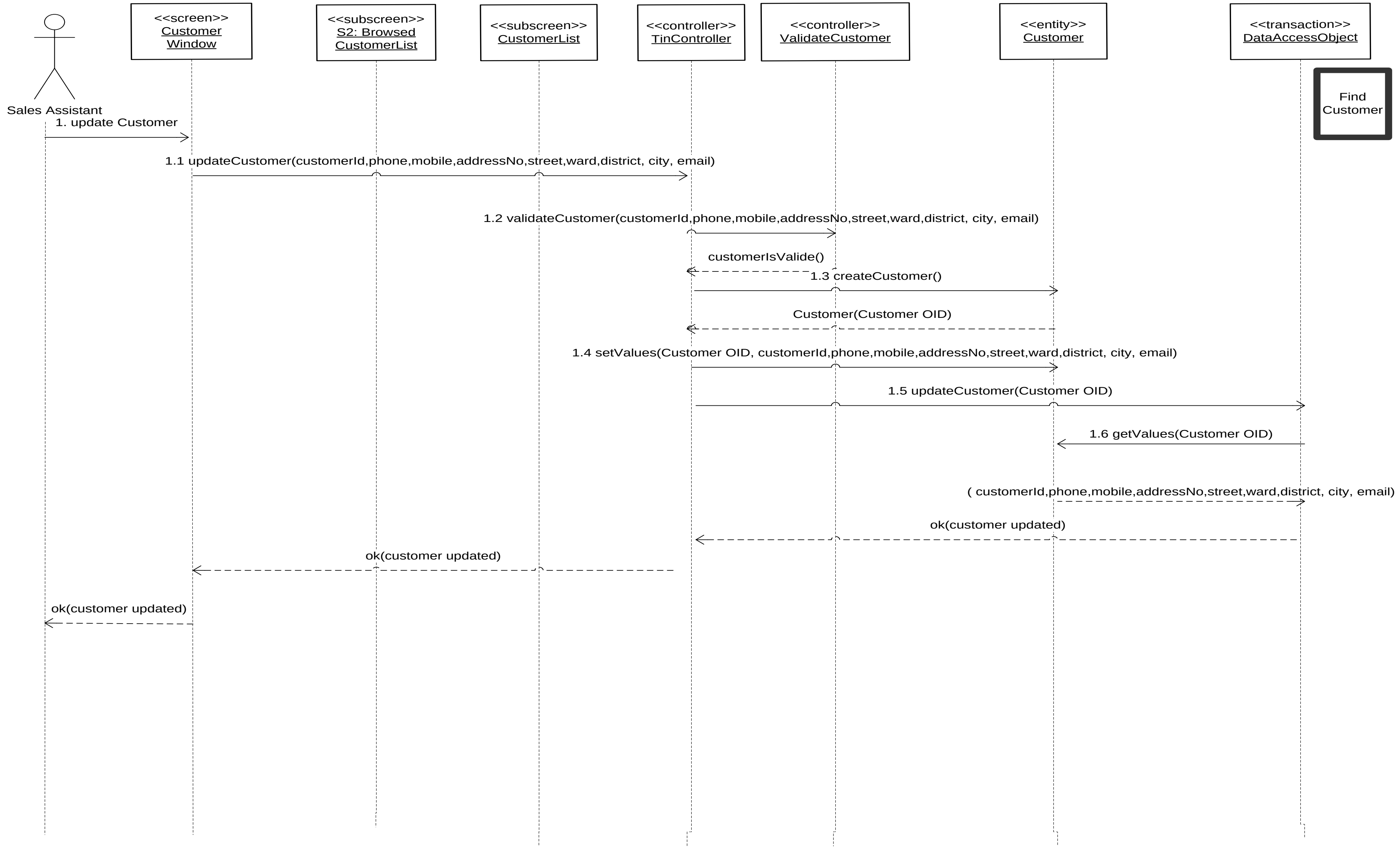
Appendix 1: Add Customer


::Sales Assistant
<<screen>>
Customer
Window
<<controller>>
ValidateCustomer
<<controller>>
TinController
<<entity>>
Customer
<<utility>>
Counter
<<transaction>>
DataAccessObject
1. AddCustomer()
1.1 validateCustomer(firstname, lastname,idNumber,phone, cell, addressNo,street,ward,district,city,email)
customerIsValide()
1.2 addCustomer(firstname, lastname,idNumber,phone, cell, addressNo,street,ward,district,city,email)
1.3 createCustomer()
Customer(Customer OID)
1.4 setValues(customer OID, firstname, lastname, idNumber,phoneNo, cell, addressNo, street, ward, district, city, email)
1.5 addCustomer(Customer OID)
1.6 getValues(Customer OID,firstname, lastname,idNumber,phone, cell, addressNo,street,ward,district,city,email)
(firstname, lastname,idNumber,phone, cell, addressNo,street,ward,district,city,email)
1.7 getNextFreeCustomerId()
customerId
ok(customeId)
ok(customerId)
ok(customerId)

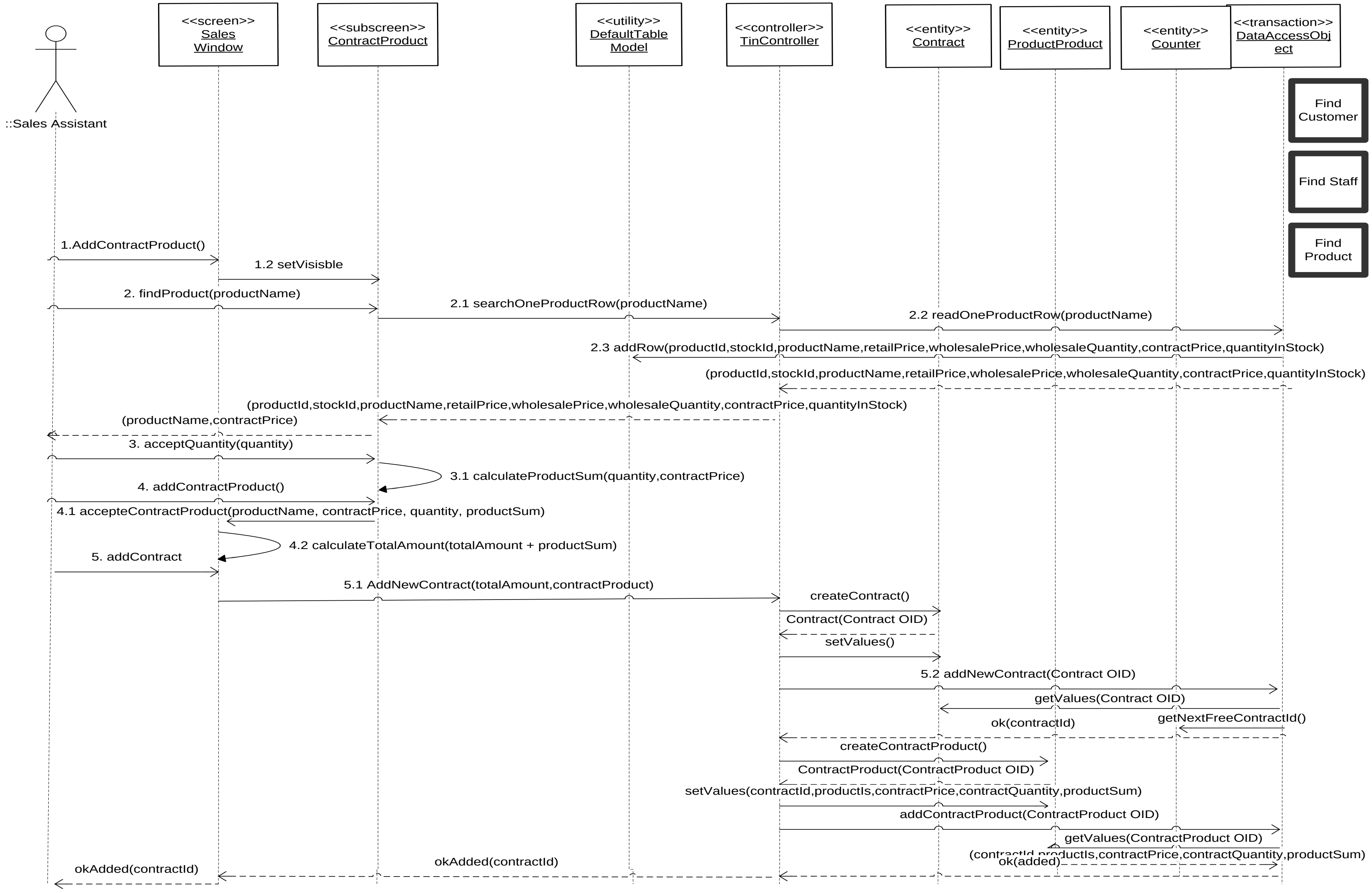
Appendix 2: Find Customer


Sales Assistant
<<screen>> Customer Window
<<subscreen>> S2: Browsed CustomerList
<<subscreenl>> CustomerList
<<controller>> TinController
<<utility>> DefaultTable Model
<<entity>> Customer
<<transaction>> DataAccessObj ect
1. findCustomer(customerName)
1.2.searchOneCustomerRow(customerName)
1.3 readOnlyOneCustomerRow(customerName)
1.4.createCustomer()
Customer(Customer OID)
1.5 setValues(Customer OID, customerId,firstname, lastname, idNumber, addressNo, street, ward, district, city, phone, email)
1.6 addRow(customerId,firstname, lastname, idNumber, addressNo, street, ward, district, city, phone, email)
defaultTableModel(customerId,firstname, lastname, idNumber, addressNo, street, ward, district, city, phone, email)
defaultTableModel(customerId,firstname, lastname, idNumber, addressNo, street, ward, district, city, phone, email)
1.7 countRows()
number
1.8 searchCustomerByName(customerName, CustomerWindow)
1.9 createCustomerList(customerName,CustomerWindow, BrowsedCustomerList)
1.10 readListOfCustomersUsingName(customerName)
1.11 readListOfCustomersUsingName(customerName)
1.12 addRow(customerId,firstname,lastname,district)
defaultTableModel(customerId,firstname,lastname,district)
defaultTableModel(customerId,firstname,lastname,district)
ok
addCompnent(customerList)
list(customerId,firstname,lastname,district)
2. getSelectedCustomer(selected customer)
2.1 readSingleCustomer(customerId)
2.2 readSingleCustomer(customerId)
2.3 createCustomer()
Customer(Customer OID)
2.4 setValues(Customer OID, customerId,firstname, lastname, idNumber, addressNo, street, ward, district, city, phone, email)
Customer(Customer OID)
2.5 getValues(Customer OID, customerId,firstname, lastname, idNumber, addressNo, street, ward, district, city, phone, email)
customerId,firstname, lastname, idNumber, addressNo, street, ward, district, city, phone, email
customerId,firstname, lastname, idNumber, addressNo, street, ward, district, city, phone, email
2.6 getOneCustomerData(customerId,firstname, lastname, idNumber, addressNo, street, ward, district, city, phone, email)
(customerId,firstname, lastname, idNumber, addressNo, street, ward, district, city, phone, email)

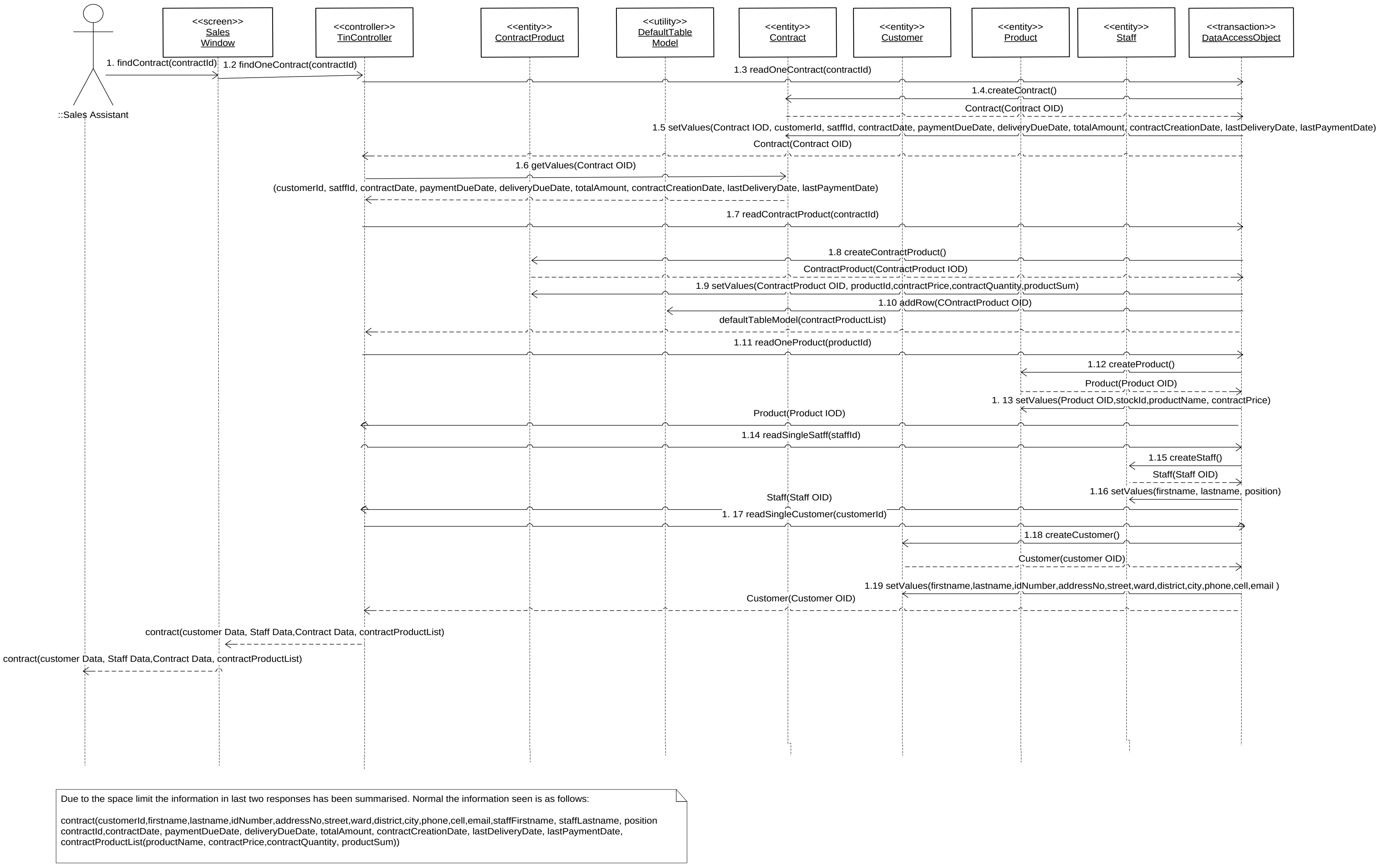
Appendix 3: Update Customer


Sales Assistant
<<screen>>
Customer Window
<<subscreen>>
S2: Browsed CustomerList
<<subscreen>>
CustomerList
<<controller>>
TinController
<<controller>>
ValidateCustomer
<<entity>>
Customer
<<transaction>>
DataAccessObject
Find Customer
1. update Customer
1.1 updateCustomer(customerId,phone,mobile,addressNo,street,ward,district, city, email)
1.2 validateCustomer(customerId,phone,mobile,addressNo,street,ward,district, city, email)
customerIsValide()
1.3 createCustomer()
Customer(Customer OID)
1.4 setValues(Customer OID, customerId,phone,mobile,addressNo,street,ward,district, city, email)
1.5 updateCustomer(Customer OID)
1.6 getValues(Customer OID)
(customerId,phone,mobile,addressNo,street,ward,district, city, email)
ok(customer updated)
ok(customer updated)
ok(customer updated)

Appendix 4: Add Contract


::Sales Assistant
<<screen>> Sales Window
<<subscreen>> ContractProduct
<<utility>> DefaultTable Model
<<controller>> TinController
<<entity>> Contract
<<entity>> ProductProduct
<<entity>> Counter
<<transaction>> DataAccessObject
Find Customer
Find Staff
Find Product
1.AddContractProduct()
1.2 setVisisble
2. findProduct(productName)
2.1 searchOneProductRow(productName)
2.2 readOneProductRow(productName)
2.3 addRow(productId,stockId,productName,retailPrice,wholesalePrice,wholesaleQuantity,contractPrice,quantityInStock)
(productId,stockId,productName,retailPrice,wholesalePrice,wholesaleQuantity,contractPrice,quantityInStock)
(productId,stockId,productName,retailPrice,wholesalePrice,wholesaleQuantity,contractPrice,quantityInStock)
(productName,contractPrice)
3. acceptQuantity(quantity)
3.1 calculateProductSum(quantity,contractPrice)
4. addContractProduct()
4.1 accepteContractProduct(productName, contractPrice, quantity, productSum)
4.2 calculateTotalAmount(totalAmount + productSum)
5. addContract
5.1 AddNewContract(totalAmount,contractProduct)
createContract()
Contract(Contract OID)
setValues()
5.2 addNewContract(Contract OID)
getValues(Contract OID)
getNextFreeContractId()
ok(contractId)
createContractProduct()
ContractProduct(ContractProduct OID)
setValues(contractId,productIs,contractPrice,contractQuantity,productSum)
addContractProduct(ContractProduct OID)
getValues(ContractProduct OID)
(contractId,productIs,contractPrice,contractQuantity,productSum)
ok(added)
okAdded(contractId)
okAdded(contractId)

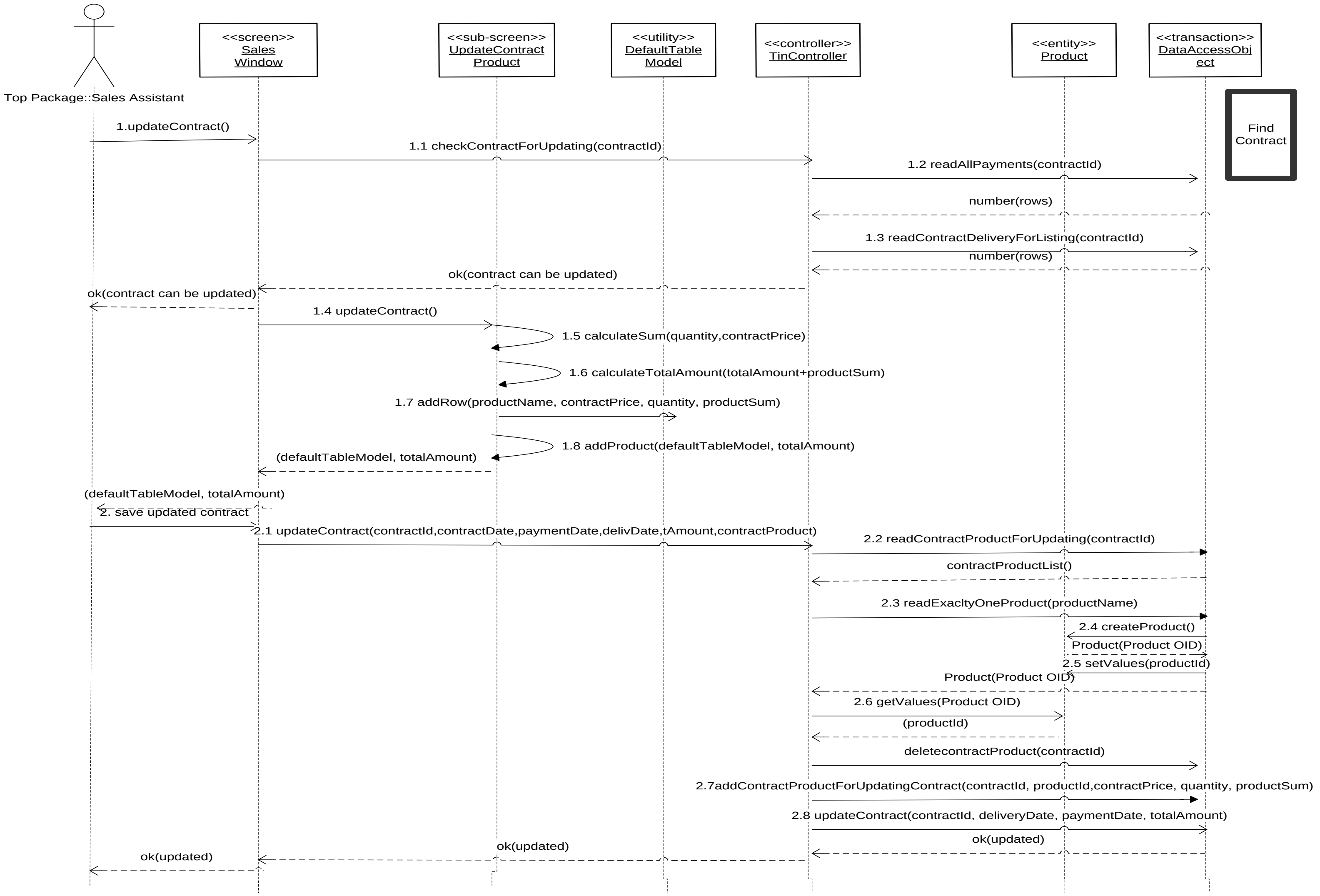
Appendix 5: Find Contract


::Sales Assistant
<<screen>>
Sales
Window
<<controller>>
TinController
<<entity>>
ContractProduct
<<utility>>
DefaultTable
Model
<<entity>>
Contract
<<entity>>
Customer
<<entity>>
Product
<<entity>>
Staff
<<transaction>>
DataAccessObject
1. findContract(contractId)
1.2 findOneContract(contractId)
1.3 readOneContract(contractId)
1.4.createContract()
Contract(Contract OID)
1.5 setValues(Contract IOD, customerId, satffId, contractDate, paymentDueDate, deliveryDueDate, totalAmount, contractCreationDate, lastDeliveryDate, lastPaymentDate)
Contract(Contract OID)
1.6 getValues(Contract OID)
(customerId, satffId, contractDate, paymentDueDate, deliveryDueDate, totalAmount, contractCreationDate, lastDeliveryDate, lastPaymentDate)
1.7 readContractProduct(contractId)
1.8 createContractProduct()
ContractProduct(ContractProduct IOD)
1.9 setValues(ContractProduct OID, productId,contractPrice,contractQuantity,productSum)
1.10 addRow(COntractProduct OID)
defaultTableModel(contractProductList)
1.11 readOneProduct(productId)
1.12 createProduct()
Product(Product OID)
1. 13 setValues(Product OID,stockId,productName, contractPrice)
Product(Product IOD)
1.14 readSingleSatff(staffId)
1.15 createStaff()
Staff(Staff OID)
1.16 setValues(firstname, lastname, position)
Staff(Staff OID)
1. 17 readSingleCustomer(customerId)
1.18 createCustomer()
Customer(customer OID)
1.19 setValues(firstname,lastname,idNumber,addressNo,street,ward,district,city,phone,cell,email)
Customer(Customer OID)
contract(customer Data, Staff Data,Contract Data, contractProductList)
contract(customer Data, Staff Data,Contract Data, contractProductList)

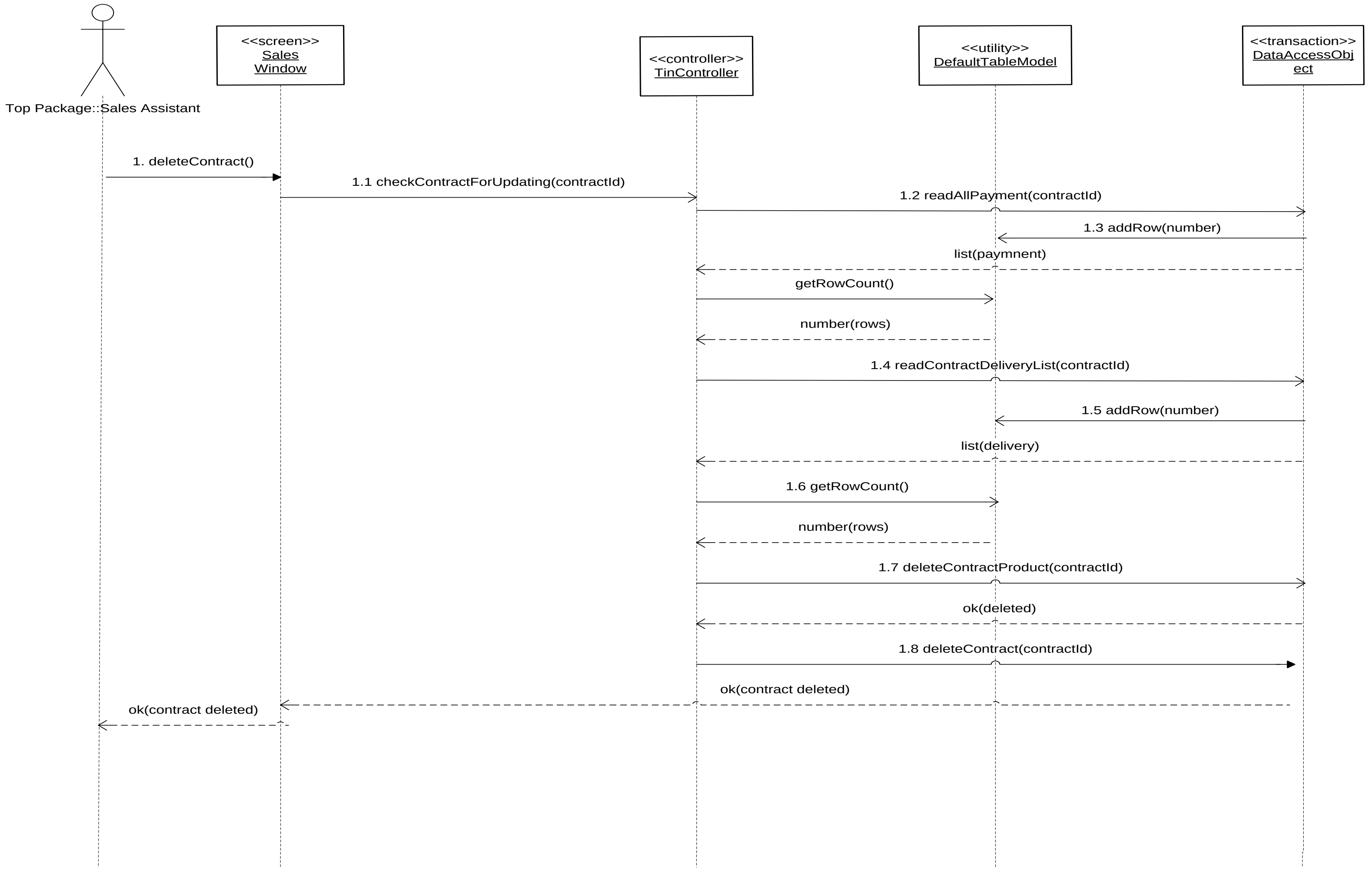
Due to the space limit the information in last two responses has been summarised. Normal the information seen is as follows:

contract(customerId,firstname,lastname,idNumber,addressNo,street,ward,district,city,phone,cell,email,staffFirstname, staffLastname, position
contractId,contractDate, paymentDueDate, deliveryDueDate, totalAmount, contractCreationDate, lastDeliveryDate, lastPaymentDate,
contractProductList(productName, contractPrice,contractQuantity, productSum))

Appendix 6: Update Contract

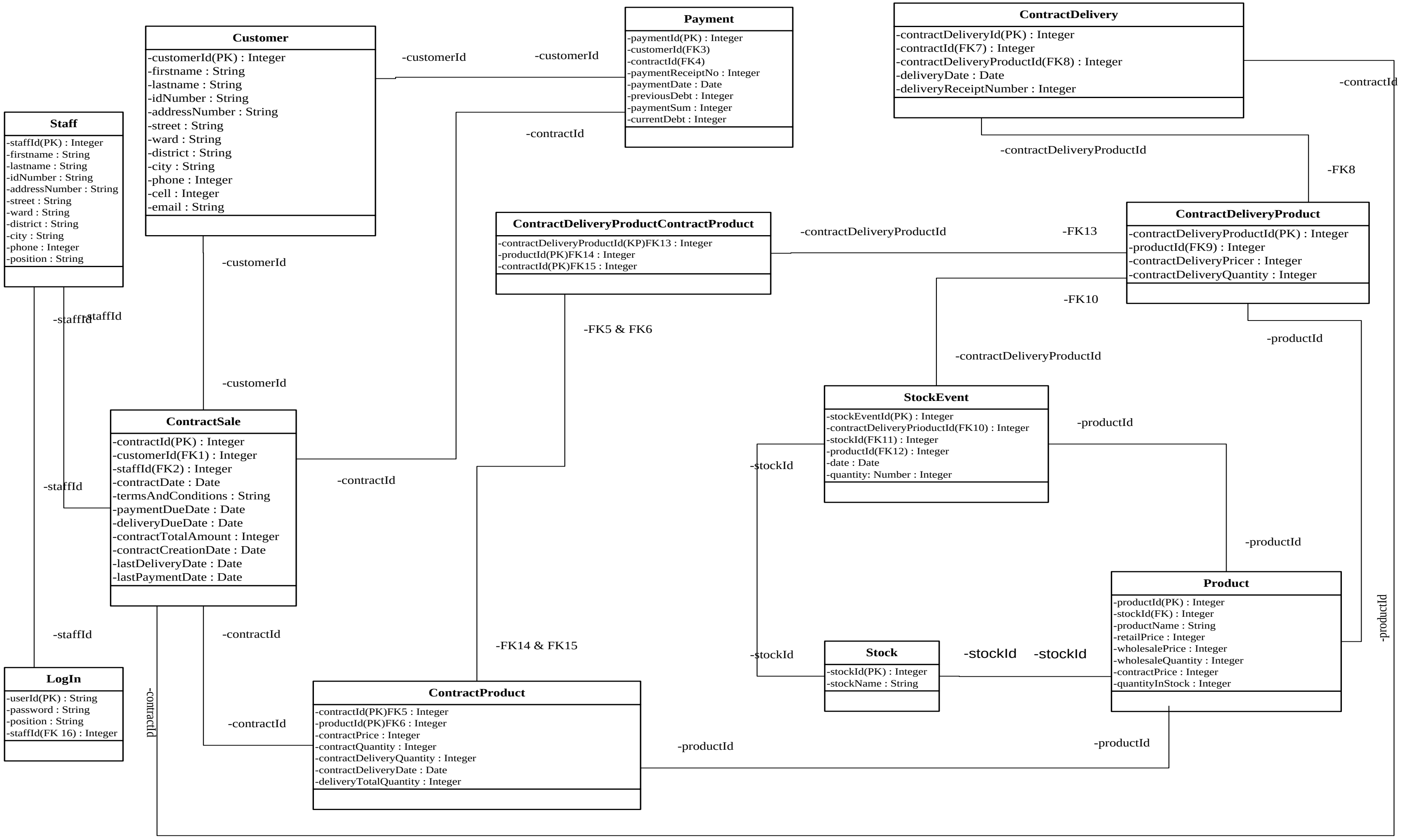

Top Package::Sales Assistant
<<screen>>
Sales Window
<<sub-screen>>
UpdateContract Product
<<utility>>
DefaultTable Model
<<controller>>
TinController
<<entity>>
Product
<<transaction>>
DataAccessObj ect
Find Contract
1.updateContract()
1.1 checkContractForUpdating(contractId)
1.2 readAllPayments(contractId)
number(rows)
1.3 readContractDeliveryForListing(contractId)
number(rows)
ok(contract can be updated)
ok(contract can be updated)
1.4 updateContract()
1.5 calculateSum(quantity,contractPrice)
1.6 calculateTotalAmount(totalAmount+productSum)
1.7 addRow(productName, contractPrice, quantity, productSum)
1.8 addProduct(defaultTableModel, totalAmount)
(defaultTableModel, totalAmount)
(defaultTableModel, totalAmount)
2. save updated contract
2.1 updateContract(contractId,contractDate,paymentDate,delivDate,tAmount,contractProduct)
2.2 readContractProductForUpdating(contractId)
contractProductList()
2.3 readExacltyOneProduct(productName)
2.4 createProduct()
Product(Product OID)
2.5 setValues(productId)
Product(Product OID)
2.6 getValues(Product OID)
(productId)
deletecontractProduct(contractId)
2.7addContractProductForUpdatingContract(contractId, productId,contractPrice, quantity, productSum)
2.8 updateContract(contractId, deliveryDate, paymentDate, totalAmount)
ok(updated)
ok(updated)
ok(updated)

Appendix 7: Delete Contract



Appendix 8: Database design model

PK = Primary Key
FK = Foreign Key



Primary Keys are indexed.

1. Test plan

1.1 Overview

Test plan is designed to check the completeness, security and quality of system and specification documentation. The plan will give the management an overview of the test activities and time needed in order to perform the test efficiently. The plan is carried out by the end user and it is intend to find error and a way to improve the application in order to achieve quality goals. The plan will show and analyse how the test specifications will be implemented.

1.2 Unit test

Unit or Model test will be done and tested individually in order to make sure that each class performs its required functions and displays the required output of data. This is also to ensure that the input follows the requirement specification. Module testing is white box testing, this means the test uses the internal perspective of the system to design test cases based on internal structure, thus it requires programming skills to identify all paths though the software

1.3 Integration test

In this test, several modules are treated as a black box; several modules are integrated and the integrated interface between the modules is tested and must provide parameter values when the design documentation does not specify calling requirements for client functions.

1. What is tested: Manage Sales use case.
2. Who is testing: The developer can carry out the test.
3. Where and when the test takes place: In the test environment (Java, Eclipse) at the same time with the module testing
4. Test case type used: Black box

5. How each use case is tested: we use bottom up method.
6. How the found errors are corrected: Through fixing the code of the interface as well as other parts in programming logic.
7. When are the corrections retested: Immediately after errors found have been fixed
8. What are the acceptance criteria for this test? When the test required outputs are produced.
9. What are the follow up procedures: Testing the interface from time to time.

1.4 System test

The system testing is carried out by people who are not involved in the development process. System test phase is checked so that its compliance with the customer requirements specifications.

1. What is tested: The software components that have passed through integrated testing are tested.
2. Who is testing: The System end users (client, customers)
3. Where and when the test takes place: In the operation environment.
4. What test case type is used (white or black box): Black box.
5. How each use case is tested: Each use case is tested using top down method
6. How the found errors are corrected: by using the regression test
7. When are the corrections retested: as soon as the corrections are made.
8. What are the acceptance criteria for this test?; When the end users are satisfied with the system.
9. What are the follow up procedures: Error found and fixed can cause another error in other modules, so testing the system time to time is necessary.

When working on this project time has come, that it was necessary to make some changes but due to the time shortage, these changes could not be made in each and every document that is why they are reported in this appendix which is annexed to both Design document and User guide document.

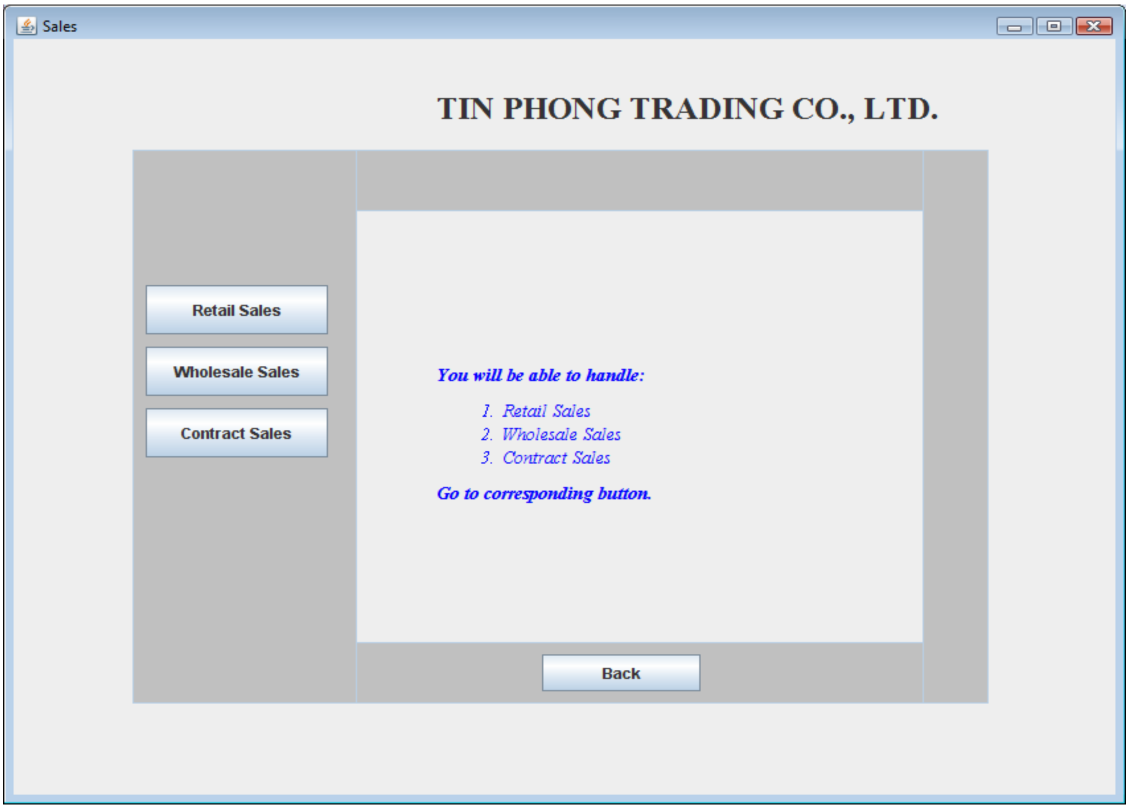
Changes that have been made affected only, terms of commands used in this sub-system and some screens that have changed their outlook.

Terms of command used

-After analysing the meaning of the word "**Exit**" in software development and how it should be used; Except on "**Main window**" where the command name was and remained "**Exit**" and on "**LogIn window**" where the command name was "**Exit**" and has been changed to "**Cancel**" It has been decided to change the name "**Exit**" to "**Back**" and this means that apart from the exceptions mentioned above wherever we have described "**Exit**" keep in your mind that the functionality it had has been assigned to "**Back**" command.

Screens (Windows)

For perfection purpose some screens have changed the outlook but activities that were described to be performed are still the same. This can be seen for example on Main window where according to the instructions in both "Design document" and "User guide", it was said that when you click "**Manage sales**" you go directly to contract window, but currently is not the case because now you go to the following window



You will be able to handle:

1. *Retail Sales*
2. *Wholesale Sales*
3. *Contract Sales*

Go to corresponding button.

And from here once you click the "**Contract sales**" the system opens contract window as it was described before. This change was made because we thought about when the Retail and Wholesale Sales will be implemented will make life easier thus letting the system be friendly user.

Closing the system

In the same way, it was said that the application can be closed or quitted from "**LogIn window"**, but currently has been changed to "**Main Window**" and this is because after you have logged in to the system "**LogIn Window"** automatically disappears.

Tin Phong Sales Management System

Simeon Mushimiyimana

Implementation document

Business Information Technology

2009

Contents

Appendices

1. Introduction

1.1 Purpose

The purpose of this document is to give the future developer or system maintainer the clear understanding of the structure of this –subsystem, the structure is more about folders and their contents. Therefore folders location, content as well as meaning are given in more details in document.

While providing the location of each document, I have ignored the main Student Driver (/M :), this means that as it is known in Haaga-Helia Environment that each student has his/her documents under this driver it would have been no sense to start describing it again.

2. Database schema

The purpose of database schema is to show the database tables, fields in each table, and the relationship between tables and fields. It also shows access key from one table to another, the access keys are useful when collecting data from different tables by joining them. Even though the fields types cannot be seen here due to the tool used it is better to remind that one of the roles of the database schema is also to show the fields types

Company information is not described in this document it will be described later.

See appendix 1: Database schema

3. Classes and Sample codes

3. 1. Sample codes

```
package guiLayer;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import controlLayer.TinController;

/**
 * With the contract product window, you can collect to your contract
as many
 * product as you want, each time the send button is clicked the
 product data is
 * sent to the contract window and is ready to be added into the system.
The
 * product name and contract price comes from the database and once
they are
 * ready the quantity is given and finally the system is able to
calculate so
 * called product sum, which is the amount according to the quantity
that was
 * given
 */
public class ContractProductWindow extends JFrame {

           private TinController theController;

           private SalesWindow theSales;

           private BrowsedProductList browsedProductList;

           ContractProductWindow thisWindow = this;

           DefaultTableModel mod;

           private int number = 0;

           private static final long serialVersionUID = 1L;

           /**
            * Constructor with parameters
            *
            */
           public ContractProductWindow(SalesWindow other) {
                      setSize(600, 350);
                      setLocation(150, 80);
                      setTitle("Contract Product");
                      this.theSales = other;
                      theController = new TinController();
                      mod = new DefaultTableModel(new String[][] {}, new
String[] { "No",
                                            "Product Name", "Contract
Price", "Quantity", "Product Sum" });

                      setComponents();
           }

           /**
            * Components initialization
            */
           public void setComponents() {
                      setLayout(null);
                      add(tfProductName);
                      add(tfcontrPrice);
                      add(tfquantity);
                      add(btPrint);
                      add(btExit);

                      add(btFindProdut);
                      add(lbProductName);
                      add(lbContrPrice);
                      add(lbProductName);
                      add(lbquant);

                      lbName.setBounds(150, 20, 300, 25);

                      lbProductName.setBounds(130, 50, 100, 25);
                      tfProductName.setBounds(240, 50, 100, 25);

                      lbquant.setBounds(130, 130, 100, 25);
                      tfquantity.setBounds(240, 130, 100, 25);

                      lbContrPrice.setBounds(130, 90, 100, 25);
                      tfcontrPrice.setBounds(240, 90, 100, 25);

                      btFindProdut.setBounds(140, 250, 120, 25);
                      btPrint.setBounds(300, 250, 80, 25);
                      btExit.setBounds(400, 250, 80, 25);

                      btFindProdut.addActionListener(new
AlsFindProduct());
                      btPrint.addActionListener(new AlsSend());

                      btExit.addActionListener(new AlsExit());
                      tfcontrPrice.setEditable(false);
           }

           /**
            * Search product name and contract price accordingly
            */
           private class AlsFindProduct implements ActionListener {
                      public void actionPerformed(ActionEvent ae) {
                                 try {
                                            if
(tfProductName.getText().length() > 0) {
                                                       DefaultTableModel
model;
```

```
                                                      model =
theController.searchOneProductRow(tfProductName
           .getText().trim());
                                                      if
(model.getRowCount() == 1) {
                                                                 try {

           tfProductName.setText((String) model.getValueAt(0,
                                3));

           tfcontrPrice.setText((String) model
                                .getValueAt(0, 7));

           tfcontrPrice.setEditable(false);
                                                                 }
catch (Exception a) {

                                                                 }
                                                      } else if
(model.getRowCount() == 0) {
           JOptionPane
                      .showMessageDialog(thisWindow,
                                           "No data found, please use
different search criterion");
                                                      } else if
(model.getRowCount() > 1) {
           thisWindow.browsedProductList = new BrowsedProductList();

           browsedProductList.getProductNameForContractProduct(
                      tfProductName.getText().trim(), thisWindow);

           browsedProductList.setVisible(true);
                                                      }
                                           } else {
           JOptionPane.showMessageDialog(thisWindow,
           "Please Select searching criteria");
                                           }
                                } catch (Exception e) {
```

```
           JOptionPane.showMessageDialog(thisWindow,

           "Unhandled exception prohibited "

                      + "\n This action to be perforemd ");

                                 }

                      }
           }

           /**
            * get data of product from database
            */
           public void getProductData(ArrayList<String> vect) {

                      thisWindow.browsedProductList.dispose();
                      try {

                                 tfProductName.setText(vect.get(2));
                                 tfcontrPrice.setText(vect.get(6));
                                 tfProductName.setEditable(false);
                      } catch (Exception e) {
                                 JOptionPane.showMessageDialog(this,
                                                       " Some errors
occured while calculating ");

                      }
           }

           /**
            * Add a contract product information to the contract window
            */
           private class AlsSend implements ActionListener {
                      public void actionPerformed(ActionEvent ae) {
                                 String prodName = (String)
tfProductName.getText();

                                 boolean enableAddButton = true;

                                 if (prodName.length() > 0 &&
tfcontrPrice.getText().length() > 0
                                                       &&
tfquantity.getText().length() > 0) {

                                            try {
                                                       int contractPrice
= Integer

           .parseInt(tfcontrPrice.getText());
                                                       int quantity =
Integer.parseInt(tfquantity.getText());
                                                       int productSum =
0;
                                                       if (quantity > 1)
{

           productSum = contractPrice * quantity;
```

```
          boolean exists = true;
                                                                if
(mod.getRowCount() > 0) {
          for (int a = 0; a < mod.getRowCount(); a++) {
                     String name = (String) mod.getValueAt(a, 1);

                     if (prodName.trim().equals(name.trim())) {

                                exists = false;

                     }

          }
                                                                }

          number++;
                                                                if
(exists == true) {

          mod.addRow(new String[] { "" + number, prodName,
                                "" + contractPrice, "" + quantity,
                                "" + productSum });
          tfProductName.setEditable(true);
          tfcontrPrice.setText("");
          tfquantity.setText("");
          tfProductName.setText("");

          thisWindow.theSales.constructTable(mod,
                                enableAddButton, true);
                                                                } else
{
          if (number == 1) {
                     number = 0;
          } else if (number > 1) {
                     number = number - 1;
```

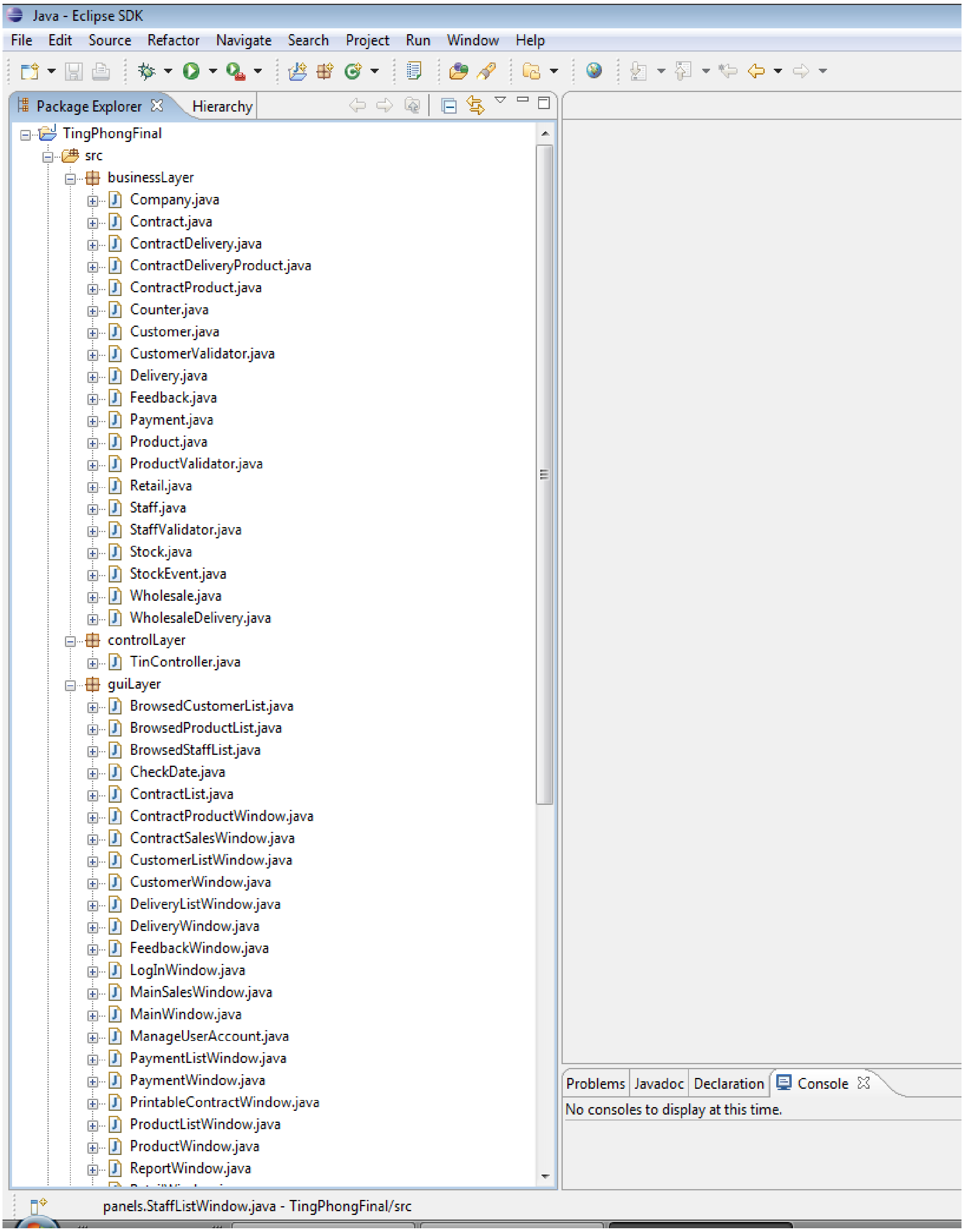
```
                    }

                    if (mod.getRowCount() > 0) {
                              for (int a = 0; a < mod.getRowCount(); a++) {
                                       String name = (String) mod.getValueAt(a,
          1);

                                       if (prodName.trim().equals(name.trim()))
          {
                                                  int oldQuantity = Integer

                    .parseInt((String) mod

                              .getValueAt(a, 3));

                                                  if (oldQuantity != quantity)
          {

                                                           mod.removeRow(a);

                                                           int posi = a + 1;
                                                           mod.insertRow(a,
          new String[] {

                    "" + posi, prodName,

                    "" + contractPrice,

                    "" + quantity,

                    "" + productSum });

                    tfcontrPrice.setText("");

                    tfquantity.setText("");

                    tfProductName.setText("");

                    tfProductName.setEditable(true);
```

```
           thisWindow.theSales.constructTable(
           mod, enableAddButton, true);
                                            } else {
                                                       JOptionPane
           .showMessageDialog(
                                 thisWindow,
                                 "Same Product exists"
                                                       + "\n You can
only change quantity.");
                                            }
                                 }
                      }
           }
                                                                  }
                                                       } else {
           JOptionPane.showMessageDialog(thisWindow,
                      " Quantity should be greater than 1 ");
                                                       }
                                            } catch (Exception z) {
           JOptionPane.showMessageDialog(thisWindow,
           "Quantity has to be a number greater than 1");
                                            }
                                 } else {
           JOptionPane.showMessageDialog(thisWindow, "Enter
quantity !");
                                 }
                      }
           }
```

```
          /**
           * Colose the currently window.
           */
          private class AlsExit implements ActionListener {
                     public void actionPerformed(ActionEvent ae) {

                                thisWindow.dispose();
                     }
          }

          /**
           * Fields declaration
           */
          private JButton btPrint = new JButton("Add");

          private JButton btFindProdut = new JButton("Find Product");

          private JButton btExit = new JButton("Back");

          private JTextField tfProductName = new JTextField();

          private JTextField tfcontrPrice = new JTextField();

          private JTextField tfquantity = new JTextField();

          private JLabel lbProductName = new JLabel("Product name");

          private JLabel lbContrPrice = new JLabel("Contract Price");

          private JLabel lbquant = new JLabel("Quantity");

          JLabel lbName = new JLabel("Tin Phong Trading CO,.LTD");

}
```

3.2 Java class names

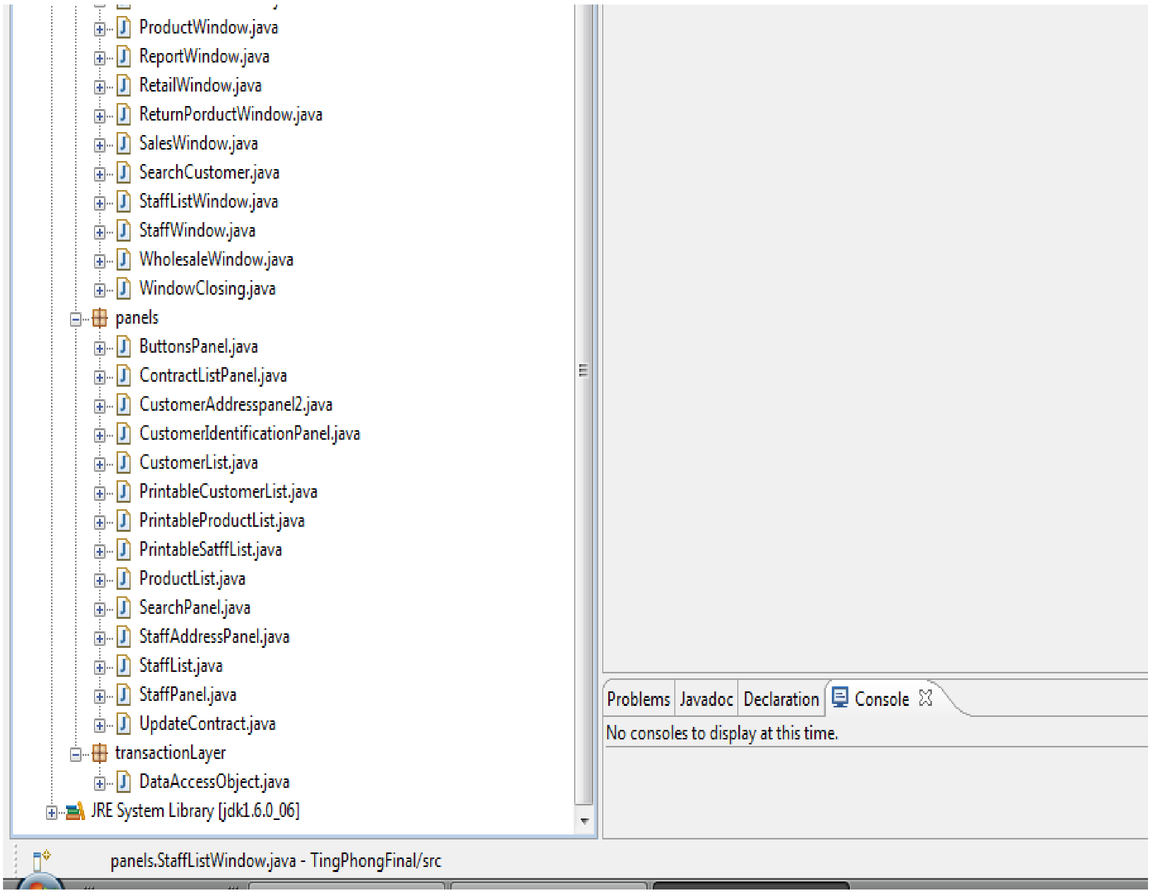
The picture below shows java classes as they are in the application.



```
TingPhongFinal
  src
    businessLayer
      Company.java
      Contract.java
      ContractDelivery.java
      ContractDeliveryProduct.java
      ContractProduct.java
      Counter.java
      Customer.java
      CustomerValidator.java
      Delivery.java
      Feedback.java
      Payment.java
      Product.java
      ProductValidator.java
      Retail.java
      Staff.java
      StaffValidator.java
      Stock.java
      StockEvent.java
      Wholesale.java
      WholesaleDelivery.java
    controlLayer
      TinController.java
    guiLayer
      BrowsedCustomerList.java
      BrowsedProductList.java
      BrowsedStaffList.java
      CheckDate.java
      ContractList.java
      ContractProductWindow.java
      ContractSalesWindow.java
      CustomerListWindow.java
      CustomerWindow.java
      DeliveryListWindow.java
      DeliveryWindow.java
      FeedbackWindow.java
      LogInWindow.java
      MainSalesWindow.java
      MainWindow.java
      ManageUserAccount.java
      PaymentListWindow.java
      PaymentWindow.java
      PrintableContractWindow.java
      ProductListWindow.java
      ProductWindow.java
      ReportWindow.java
```

No consoles to display at this time.

panels.StaffListWindow.java - TingPhongFinal/src



```
ProductWindow.java
ReportWindow.java
RetailWindow.java
ReturnPorductWindow.java
SalesWindow.java
SearchCustomer.java
StaffListWindow.java
StaffWindow.java
WholesaleWindow.java
WindowClosing.java
panels
    ButtonsPanel.java
    ContractListPanel.java
    CustomerAddresspanel2.java
    CustomerIdentificationPanel.java
    CustomerList.java
    PrintableCustomerList.java
    PrintableProductList.java
    PrintableSatffList.java
    ProductList.java
    SearchPanel.java
    StaffAddressPanel.java
    StaffList.java
    StaffPanel.java
    UpdateContract.java
transactionLayer
    DataAccessObject.java
JRE System Library [jdk1.6.0_06]
```

No consoles to display at this time.

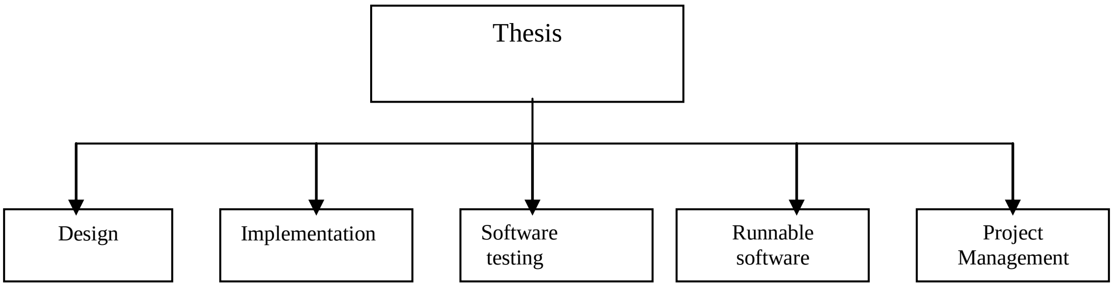
panels.StaffListWindow.java - TingPhongFinal/src

4. Folders and content

The idea is to show the location of each item, document or folder in details so that even one who does not know how any thing about this work can easily manage to find out what being looked for, it helps also for the maintenance as well as future continuation of this work.

4.1. Document folders

The picture below show the top level folders, we will open each of them to detail the document's location, each of them contains other sub-folders and the information about documents in each is given in details.



4.1.1 Design

The design sub-folder contains five sub-folders as follows;

- **Database architecture**

- **Software architecture;** Contains Design document and "**Appendices"** sub-folder in which all sequences diagrams, List of changes, and Database design model are found. It also contains the "**Test planning"** sub-folder in which we find TestPlan and TestForm documents as well as "**User guide**" sub-folder that holds

- **Test planning**
  This sub-folder contains Test forms, Test plan document

- **User guide**
  This sub-folder contains these documents; User guide document, List of Changes and List of Controls as an appendices to this document

4.1.2 Implementation:

The implementation sub-folder contains one sub-folders and one document.

- **28042009_FinalProject**: This is the sub-folder in which the application itself is located as follows and will be described in details later on.
- **Implementation.doc**: This is this document we are using right now to describe the content of the project.

4.1.3 Software testing

To ensure quality goals; testing is the most important part in IT Projects, so, this sub-system has been tested from unit testing, integration testing and system (sub-system in this case) testing as whole. In this project there has been a greater number of tests but only some system tests results are shown here and for more details about test planning one can see "**Design document, Test planning**"
For test results see Addendix 2: System Test results.

4.1.4 Runnable software

When the application code is burned it becomes runnable as it can be deployed and be understood by the operation system to perform its services. However, regardless of the current sub-system status which is not the whole system yet, and of not deploying the subsystem that is ready at the moment; this sub-system will be burned on Compact Disk (CD) thus the term "**Runnable software**" used here.
When you run the application, you need as user name and password you can use ("sam" as user name and "helia123" as password).

4.1.5Project Management

Project Management folder holds all document that tell how this project was management as well as all reports made including Thesis report and Thesis Final report; these reports are saved in many sub-folders and the structure looks like follows .

- **Project Plan:** Holds the Project plan and the schedule made to accomplish the plan.
- **Agendas:** All agendas of steering meetings are located in this sub-folder.
- **Minutes of meetings:** All minutes of the meetings that tell when, what was the meeting purpose, where, participants as well as decisions are held in this sub-folder.
- **Progress reports:** This sub-folder holds the progress report made to communicate the project status to the steering group
- **Thesis report:** This sub-folder contains information about Thesis report, Thesis final report as well as workloads that shows use of resources.

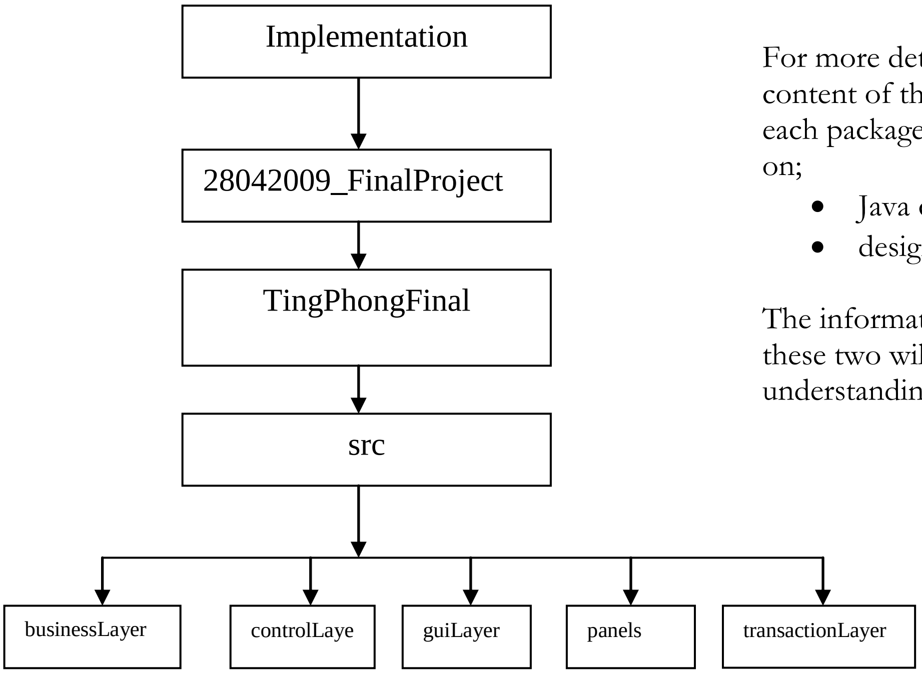
It is wise to remind that all appendices belonging to a given document are located in the same sub-folder and in which the document they belong to is located.

4.1.6 Database

Database used in this sub-system, all tables and fields they contain are located in this sub-folder. Remember the database name is “**Tin Phong**””

4.2. Application folders

The application contains so called, project folders which can be classified in two different types; one type groups system auto-generated folders and the second one the so called application folders. The first one contains class libraries that compose the platform of the tool to allow our own implementation and the second categories is the one that folds our own made classes that hold our programming logic or technically speaking the application itself.



For more details of the content of the classes in each package take a look on;

- Java class name
- design document

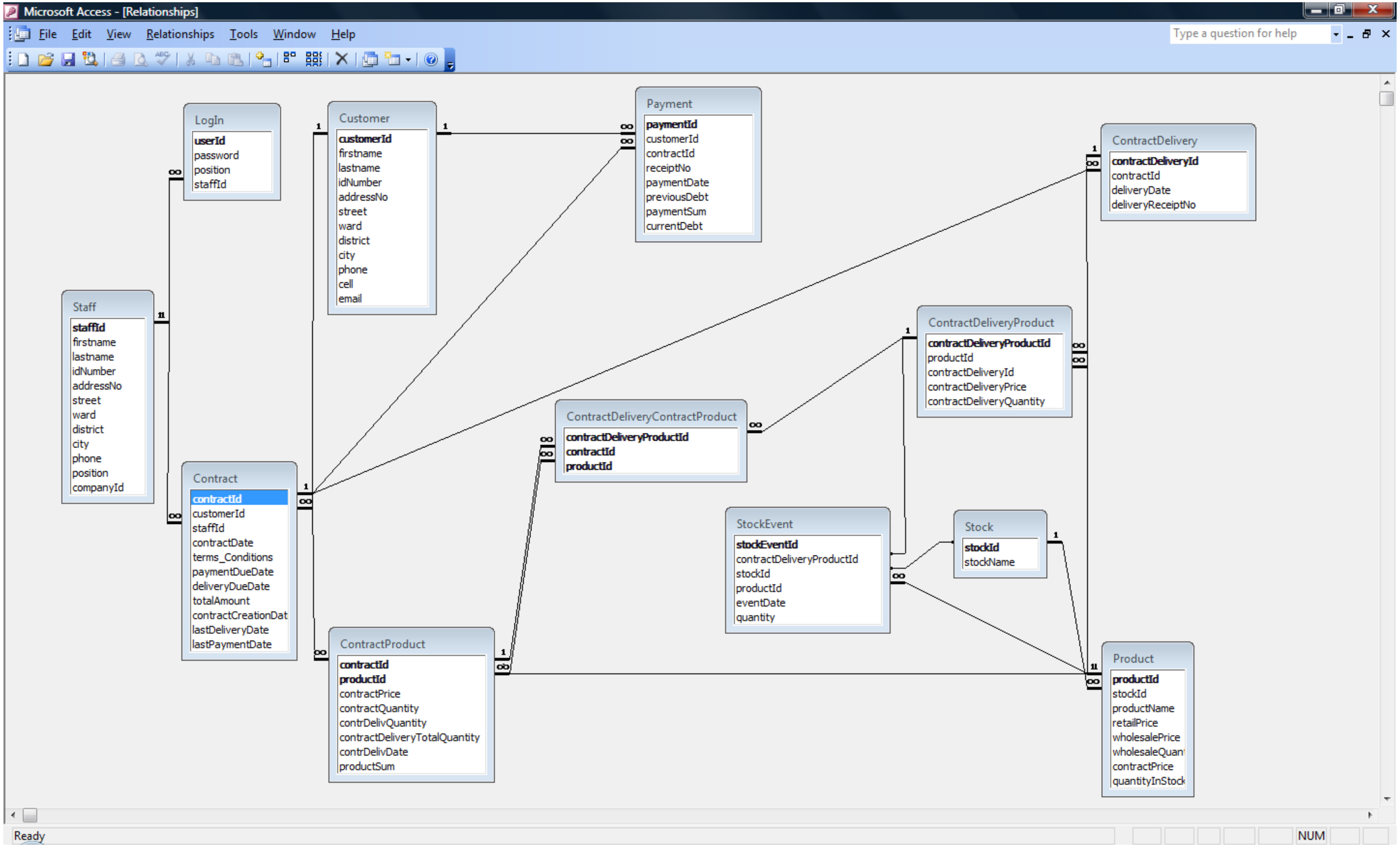
The information from these two will help you understanding.

Only own made folders are shown in this picture, built-in folders are not part of it.

Appendices

Appendix 1: Database schema
Appendix 2: First System Test results
Appendix 3: Second System Test result

Appendix 1: Database schema



LogIn
userId
password
position
staffId

Customer
customerId
firstname
lastname
idNumber
addressNo
street
ward
district
city
phone
cell
email

Payment
paymentId
customerId
contractId
receiptNo
paymentDate
previousDebt
paymentSum
currentDebt

ContractDelivery
contractDeliveryId
contractId
deliveryDate
deliveryReceiptNo

Staff
staffId
firstname
lastname
idNumber
addressNo
street
ward
district
city
phone
position
companyId

ContractDeliveryProduct
contractDeliveryProductId
productId
contractDeliveryId
contractDeliveryPrice
contractDeliveryQuantity

ContractDeliveryContractProduct
contractDeliveryProductId
contractId
productId

Contract
contractId
customerId
staffId
contractDate
terms_Conditions
paymentDueDate
deliveryDueDate
totalAmount
contractCreationDat
lastDeliveryDate
lastPaymentDate

StockEvent
stockEventId
contractDeliveryProductId
stockId
productId
eventDate
quantity

Stock
stockId
stockName

ContractProduct
contractId
productId
contractPrice
contractQuantity
contrDelivQuantity
contractDeliveryTotalQuantity
contrDelivDate
productSum

Product
productId
stockId
productName
retailPrice
wholesalePrice
wholesaleQuan
contractPrice
quantityInStock

1.1. Test Results of Add Customer use case selection.

Use Case: Record Potential customer (Add Customer) Date: 02.04.2009

Tester: Steering group meeting

No	Testing procedure step by step	Expected result	Errors/ Exceptions	Has to be fixed? Yes/No
1	The Actor run browses	Log in window opens		
2	Actor enters user name and password and clicks Login button	Main window opens		
3	Actor clicks Manage Customer button	Customer window opens		
4	Actor fills the fields (first name, last name, National ID, Telephone, Mobile, Address number, street, ward, district, city and email) and clicks Add button.	The system displays a message that the customer has been added and shows the customer number.	After a customer information is saved the system should display the customer number in the specified field - Test telling the customer what to do (update, save) -Database field not matching the input type	Yes

* Even though Log in is not part of this use case, it is assumed that the Actor is not logged in yet. If the Actor is already logged in, then start from step 3.

1.2. Test Results of Browse customer use case selection

Use Case: Record Potential customer (Browse Customer) **Date:** 02.04.2009

Tester: **Steering group meeting**

No	**Testing procedure step by step**	**Expected result**	**Errors/ Exceptions**	**Has to be fixed? Yes/No**
1	From the Main window, the Actor clicks Manage Customer button	Customer window opens		
2	Actor input search criteria (first name) and clicks Find button.	- If the are many customer with the same name *Lists of customers matching searching criterion is displayed on Browsed Customer List sub-window. - If only one customer the customer data is displayed on Customer window.	Customer data on window is not visible the fields out look has to be changed.	Yes
4	Actor selects required customer and clicks Select button.	The selected customer data is displayed on the Customer window.	Buttons need to be enabled disabled depending on which action is next to be performed	Yes
5	If Actor wants to make another search, clicks the Empty all button.	All fields on Customer window are empted. -Repeat step 2 to 5 as long as needed.		

1.3. Test Form for Handle contract sale use case

Use Case: Handle contract sale (Add Contract) **Date:** 02.04.2009

Tester: Steering group meeting

No	Testing procedure step by step	Expected result	Errors/ Exceptions	Has to be fixed? Yes/No
1	From the Main window, the Actor clicks Manage Sales button	Contract window opens		
2	The Actor enters customer number.	The data of the customer is displayed on Contract window.	The system disable all fields which prohibit the user to continue other activities	Yes
3	The Actor enters customer search criteria (customer first name) and clicks Find Customer button.	See steps in Browse Customer sub-use case. (at the end the customer data is displayed on Contract window)	The system disable all fields which prohibit the user to continue other activities	Yes
4	The Actor enters staff search criteria (staff fist name) and clicks Find Staff button.	Same in step 3 but this time for staff data and staff windows		Yes
5	Actor fills all other fields			
6	Actor clicks Add Contract Product button.	Contract product sub-window opens.	.	Yes
7	Actor enters product full name and clicks Find product button.	Product name and contract price is displayed on Contract Product sub-window.		

No	Testing procedure step by step	Expected result	Errors/ Exceptions	Has to be fixed? Yes/N
8	Actor enters quantity and clicks Add button.	Product sum and total amount is calculated and product name, contract price, quantity and product sum are displayed in a table on Contract window, while the total amount is displayed on total amount text field also on Contract window. Repeat step 7and 8 as many times as needed.	It should be possible to remove contract product if needed. Improve the sales window out look, when removing contract product, some information is not correctly removed.	Yes
9	Actor clicks Exit button to close the Contract product sub-window.	Contract product sub-window closes.		
10	Actor clicks Add button on Contract window.	A message that the contract was added is displayed as well as new contract number.	The system does not display the new contract Id in the field	Yes

1.4. Test Form for Handle contract sale use case

Use Case: Handle contract sale (Browse Contract) **Date:** 02.04.2009

Tester: Steering group meeting

No	Testing procedure step by step	Expected result	Errors/ Exceptions	Has to be fixed? Yes/No
1	From the Main window, the Actor clicks Manage Sales button	Contract window opens		
2	The Actor enters customer number.	The data of the customer is displayed on Contract window.		
3	The Actor clicks Find Contract button	-If there is many contract for the target customer ✚ A list of all contracts for the target customer is displayed on Contract List sub-window. -If only one contract. ✚ Contract data is displayed on Contract window.	Data on fields not clearly visible.	Yes
4	The Actor selects one contract and clicks Select button.	Contract data is displayed on contract window.		
5	If the Actor wants to make another search, click the Empty All button.	All fields are empted. Repeat step 2 to 4 5 as many times as needed.		

1.4. Test Form for Handle contract sale use case

Use Case: Handle contract sale (Print Contract) **Date:** 02.04.2009

Tester: Steering group meeting

No	Testing procedure step by step	Expected result	Errors/ Exceptions	Has to be fixed? Yes/No
1	Actor browses contract	See steps in Browse product.		
2	The Actor clicks print button	The data of the current contract is displayed on Printable contract window. (just to show how data will look like when printed)		Yes
3	The Actor clicks Print button Printable contract window	Contract data is printed on paper.	The out put contract has to be improved	Yes
4	The Actor clicks on Exit button	Printable contract window closes.		

1.1. Test Results for Browse Staff

Use Case: Manage Staff (Browse Staff) **Date:** 22.04.2009

Tester: Customer (Acquirer)

No	Testing procedure step by step	Expected result	Errors/ Exceptions	Has to be fixed? Yes/No
1	The Actor enterers first name or part of it like "**tha**" when searching "**Thanh**"	If only one staff is found, data of this staff is displayed on StaffWindow. If more than one staff are found list of staffs (staff number, first name, last name and position) is displayed on BrowsedStaffWindow.	System allows to search with first and last name but uses only first name in searching, this is not correct way.	Yes
2	Select one staff from the list and click "**Selec**t" button	Browsed staff window disappears and data of selected staff is displayed on StaffWindow		
3	Click "**Update**" button to allow start updating staff information.	Only allowed fields change background color to allow Actor updating information they contain.		
4	Actor changes the fields as liked and clicks"**Save**" button.	The system displays a message that the Staff information has been updated, or a message if some entered information related to existing one according to the business rules.	In all cases system displays error that street not correct.	Yes

1.2. Test Results of Manage Staff

Use Case: Manage Staff (Create user account) **Date:** 22.04.2009

Tester: Customer (Acquirer)

No	Testing procedure step by step	Expected result	Errors/ Exceptions	Has to be fixed? Yes/No
1	Actor activates Find Staff function	Data of staff is displayed on StaffWindow		
2	Actor clicks “**New User**” button.	System opens “**New Account**” window with staff first name, last name and position filled with information of the current staff and empty fields of (account, password and confirm password.)		
4	Actor fills account, password and confirm password fields and clicks “**Add**” button		-If two accounts are found system does not display right message thus user is confused -If the user has already an account the system does not display a clear message	Yes

1.3. Test Results Manage Product

Use Case: Manage Product (Browse Product) **Date:** 22.04.2009

Tester: Customer (Acquirer)

No	Testing procedure step by step	Expected result	Errors/ Exceptions	Has to be fixed? Yes/No
1	Actor activates Manage Product use case	System displays ProductWindow with all empty fields except stock number that is filled with all stock IDs that exist.		
2	The Actor enters product number	If the entered number is recognized by the system the data of the Product is displayed on ProductWindow	If the Actor makes a second search, the system does not remove previous stock IDs thus duplication of numbers	Yes

HAAGA-HELIA
University of Applied Sciences

Tin Phong Sales Management System

Simeon Mushimiyimana

User Guide document

Business Information Technology

2009

Contents

Appendices

1. Introduction

Welcome to the Tin Phong Sales Management System, Tin Phong Sales Management System will offer a set of sales services which will allow you keeping customer, product, staff, and report and customer feedback related information as well as return product. It also allows handling Sales information, which includes contract, retail, wholesale, delivery, and payment information. Currently, not all stated services are ready. Thus, in this document we have so called "**Sub-system**", this sub-system is part of the whole system that regroups services which are ready.

-**Manage customer information:** Manage customer information in this case a general term used to globalise a number of activities related to the customer information which are; adding customer information into the system, changing customer information , searching and listing customer information from the system and printing out the result of this information.

-Manage Staff information: adding staff information into the system, changing staff information, searching and listing staff information from the system and printing out the result of this information. Furthermore, as a staff with full authorities your have been given right to add other staffs assign them a user account for logging in to the system, after when the staff can log in and change his/her own password.

-Manage Product Information: The product is a key element in business and it has not been forgotten here as well; so, the product information such, prices, quantity and the location like stock where the product is found are managed in this activity, in this system you are able to add new product information, change existing product information.

-Manage Sales: Information about sales is very critical in business point of view that why in this system is considered to be the centre of other information in the system, the fact if that sales contains information about customer, staff as well as about product. Contract, payment and delivery information is handled.

Information about deployment and application start-up is not described here and will be added later after the whole system is completely done

1.1 System

The purpose of the Tin Phong Sales Management System is to store customer, product and sales information. Sales information includes retail sales, wholesale and contract sales. Contract sales includes; information about contract, payment and delivery and contract details shows customer who owns the contract, staff who represents the company and product information(product name, quantity, contract price and product sum) as well as terms and conditions regarding the contract.

1.2 Sub-System

The so called "**Sub-system**" regroups only the services that are by now implemented and ready to be used. In these services includes;

- Manage Customer
- Manage product
- Manage Sales (Only contract sales)
- Manage Staff

Therefore, the information that is provided in this document concerns only this sub-system as the rest will be provided once the whole system will be ready.

2. Purpose

The purpose of this document is to provide with clear details on how to use what we called "Sub-system" in introduction and which is a part of Tin Phong Management System (we call it sub-system because the whole system is not ready yet). It provides information step by step on how to use this sub-system, together with screenshots and explanations about them as well as a very user friendly interface that will enable you to use the sub-system without any other technical training.

The screens shown here are exactly the ones that you will see while using the sub-system; however in some cases some minor changes have been made.
The changes that are said here are not of system functionality but only of the user interface and fewer of terms of commands used in this document and thus that appear on the screenshots, you will see at the end this document an appendices of these changes.

2.1 Screen map

The purpose of screen map is to list all screens that are used in this document. However one screen might be met more many times in the document depending on which activity to perform.

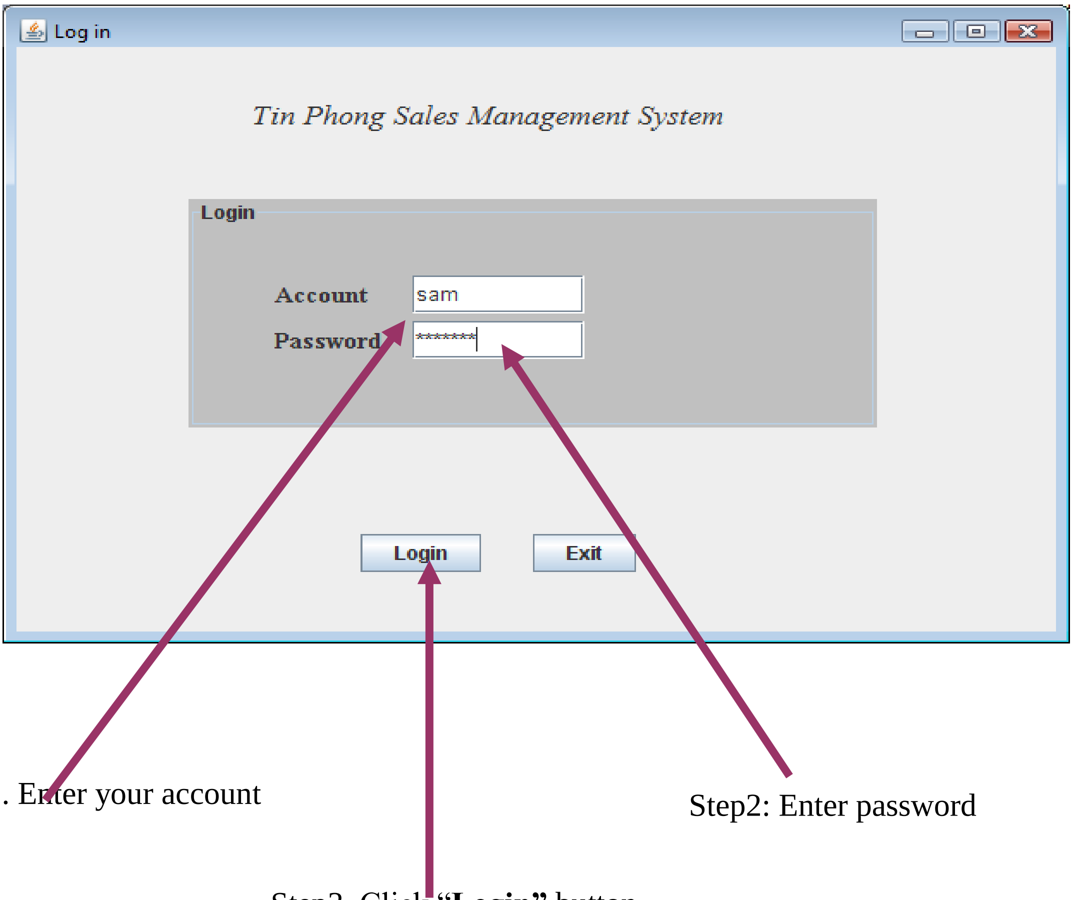
- Login screen
- Main screen
- Customer screen
- List of Customers screen
- Product screen
- List of Products screen
- Contract screen
- Contract(printable format) screen
- Payment screen
- List of Payments screen
- Delivery screen
- List of Deliveries screen
- User account screen
- Change password screen

3. Step by Step guide

Step by Step gives the user a full understanding of how the system is used.

3.1 Login into system.

To start working with the application the user has to provide user account and password, the window bellow is displayed for the fist time when the application runs.



Account sam

Password ********

Step1. Enter your account

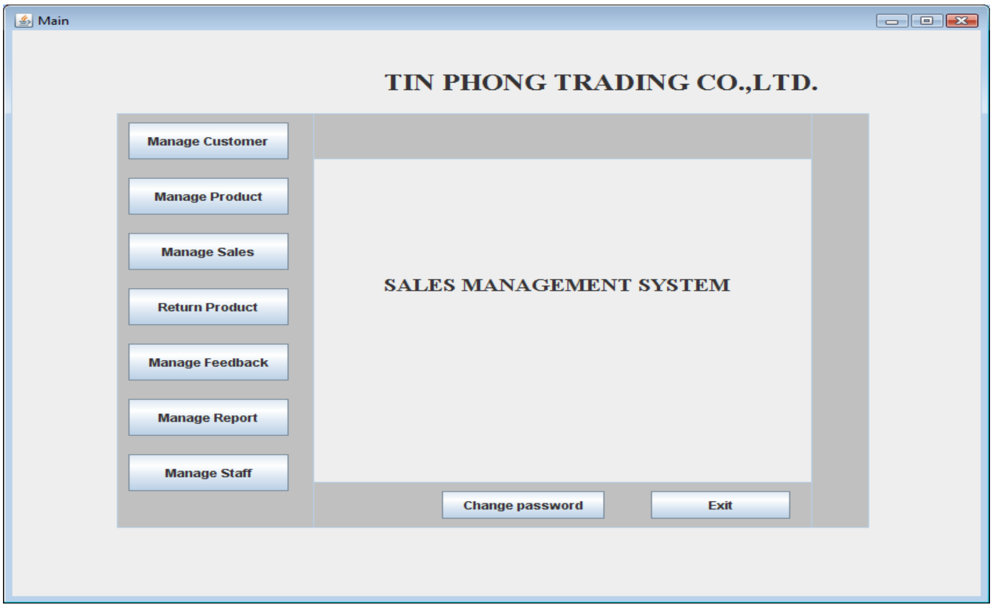
Step2: Enter password

Step3. Click "**Login**" button

Picture1. Log in screen.

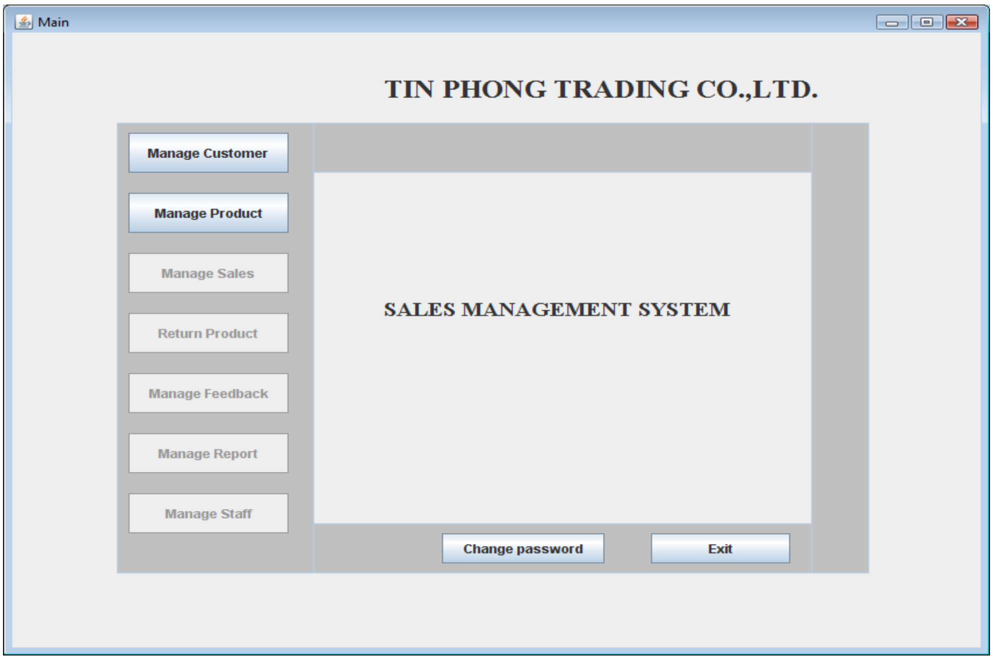
After you have clicked "**Login**" button as it is mentioned above you can see one of the following pictures depending on your user rights in using the system.

-If you have administrator's rights then the picture will look like the following,



Picture 2: Main screen (with full rights)

-If you have normal user rights the window that will be opened to you will look like this.

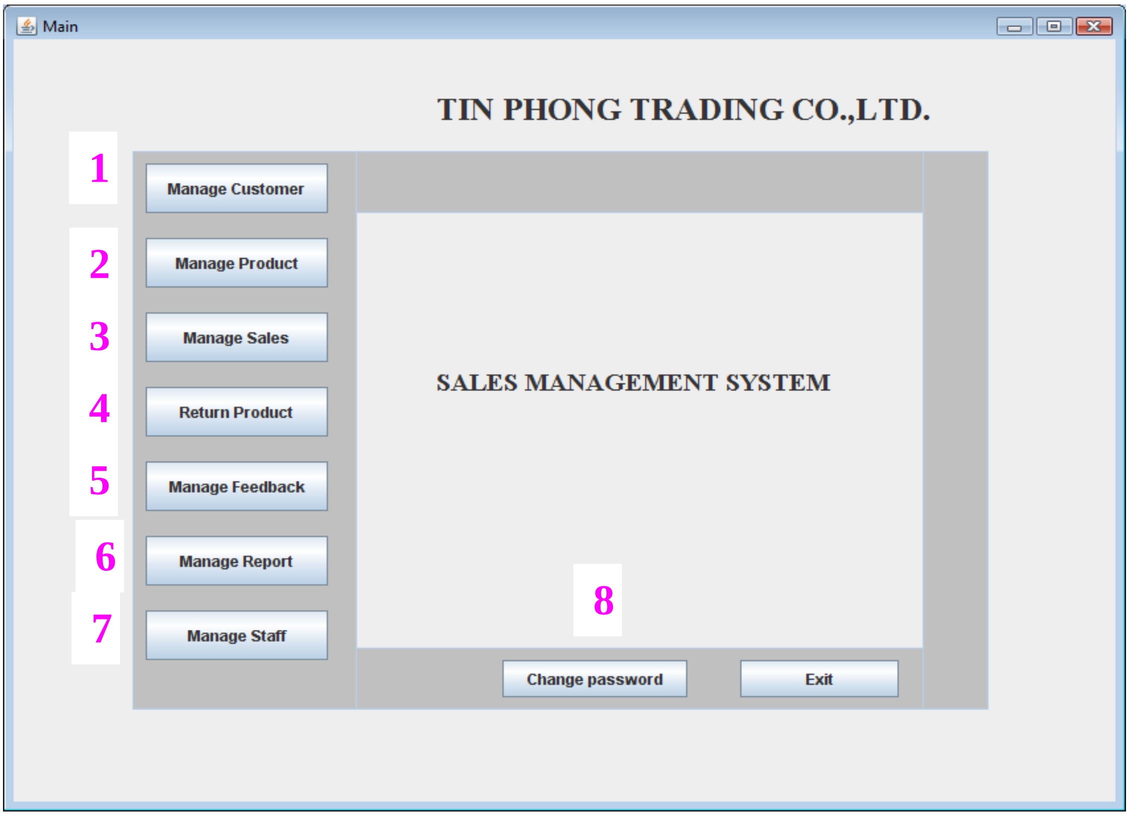


Picture 2: Main screen (user with limited rights)

As you can see this window looks little different from the previous, basically they are the same but in the second some functionalities are protected because the current user's right do not allow him/her to use them. Now, as we have seen the Main window let us see briefly what services can be performed using it.

3.2. Main window

-I tried to add these numbers from 1 to 9 just to serve as references later on in window description.



Picture 2: Main screen

The main window as the name sounds is a window from with which the user will perform all services as mentioned on it.

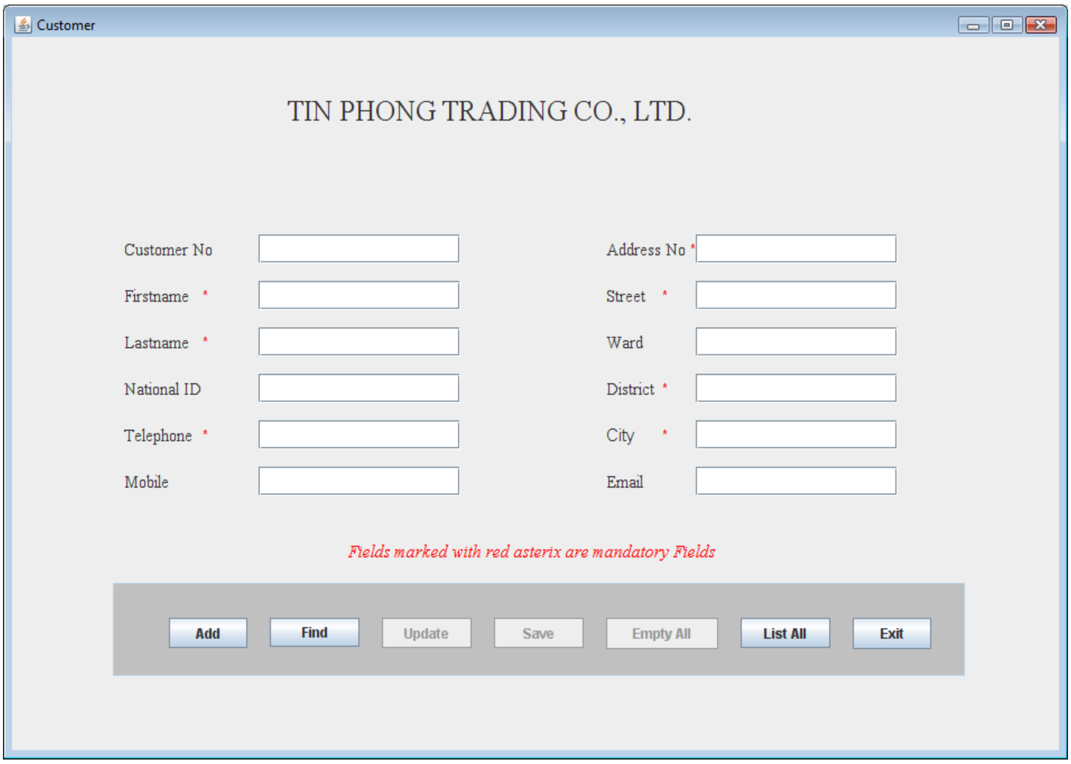
By clicking one button you will see that they are other activities that can be done behind the functionality selected, the reason why this window was given the name “Main”.

3.3 Manage Customer

You go to this function if you want to manage customer information, manage customer information includes a number of activities such as;

- Add new customer information into the system
- Search customer using customer name or customer number
- Change existing customer's information
- Display a list of all customers in the system

The button "**Manage customer"** is clicked, the following window opens.



TIN PHONG TRADING CO., LTD.

Fields marked with red asterix are mandatory Fields

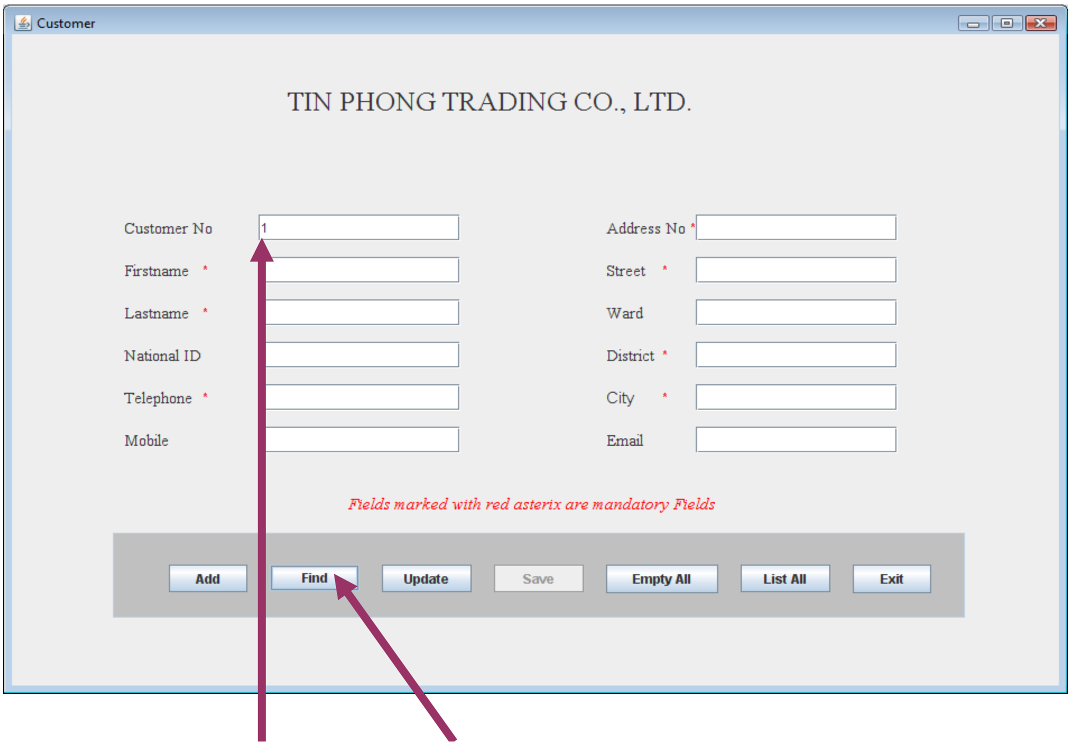
Picture 3: Customer screen

- Adding a new customer into the system

As you can see on the screen they are a number of fields to be filled. First of all read the read message on screen which tells you that “Fields marked with red asterix are mandatory Fields” these fields have to be filled before clicking “**Add**” button to add your new customer into the system; Other fields can be filled or not depending on user’s wish. After you have entered customer information and click “**Add**” button you will get a message that the customer was successfully added and a new customer number will be shown together with the message.

- Search customer using customer number

When you know that you have customer in the system and want to display information about your target customer using customer number, follow the explanations bellow.



Customer

TIN PHONG TRADING CO., LTD.

Customer No	1	Address No *	
Firstname *		Street *	
Lastname *		Ward	
National ID		District *	
Telephone *		City *	
Mobile		Email	

Fields marked with red asterix are mandatory Fields

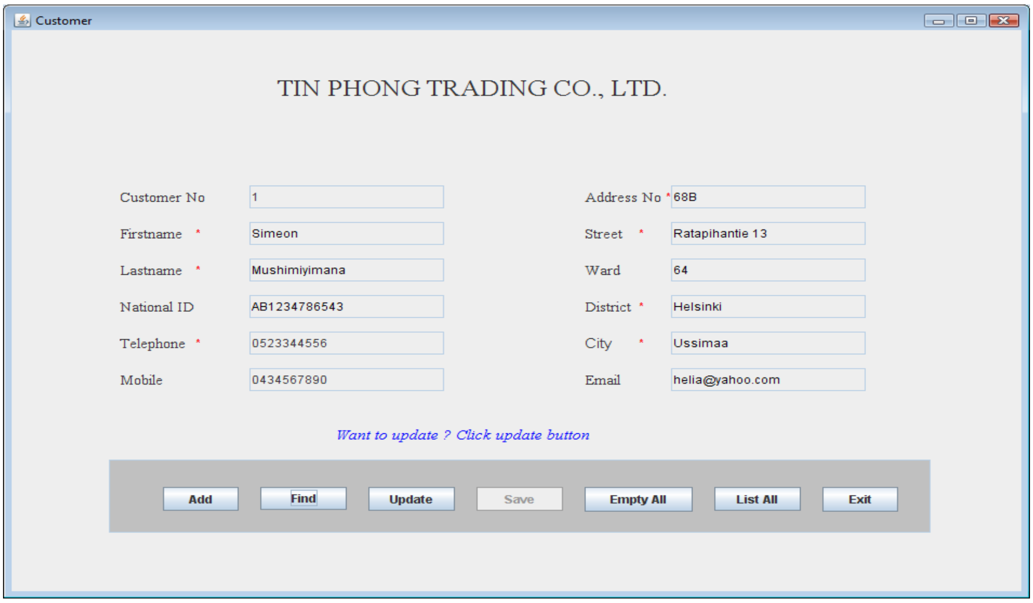
Enter Customer number, for example 1.

Click “**Find**” button

Picture 3: Customer screen

To make another search click “**Empty All**” button, all fields will be empted and you can start searching again or you can click “**List All**” to see a list of all customers or “**Exit**” to close this window.

- If the customer is found, the system will display information about that customer as it’s in the picture bellow.



TIN PHONG TRADING CO., LTD.

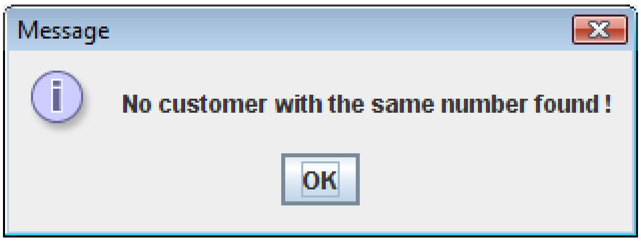
Customer No	1	Address No *	68B
Firstname *	Simeon	Street *	Ratapihantie 13
Lastname *	Mushimiyimana	Ward	64
National ID	AB1234786543	District *	Helsinki
Telephone *	0523344556	City *	Ussimaa
Mobile	0434567890	Email	helia@yahoo.com

Want to update ? Click update button

Picture 9: Customer screen (with customer information)

For making new search click on “**Empty All**”, to see the list of all customers click on ” **List All**” to close this window click “**Exit**”

It might be possible that the customer number you have given does not correspond to any customer in the system; in that case the following message will be displayed to you.



No customer with the same number found !

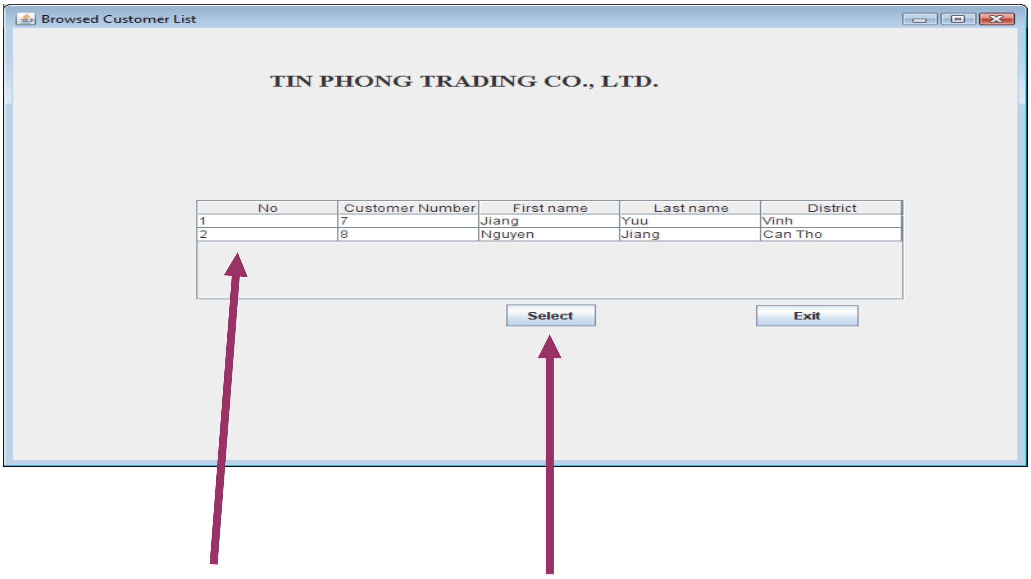
You can ignore this message by simply clicking on “**OK**” to continue.

- Search using customer's first name.

If you don't remember the customer number, but you remember customer's first name or you doubt and don't exactly remember the name just enter the name or part if it in the corresponding field to find customer information. In this case, there are two possibilities; one is that if there is only one customer with the name given the information about that customer will be displayed to as in "**Picture 9**" or you will see the following window opened to you.

Example of searching using part of name: Let assume you want to display a customer who's first name is "**Nguyen**" use; "**nguy**" or "**gu**" or "**yen**" this name will always appear in the list of names that will be displayed. As you increase the number of characters as you get short list, thus making easier your search.

In this case, the customer you want from the list and click "**Select**" button. Again the information about the customer you have selected will be displayed to you as it was displayed in "**Picture 9**".



No	Customer Number	First name	Last name	District
1	7	Jiang	Yuu	Vinh
2	8	Nguyen	Jiang	Can Tho

1. Select one customer from the list

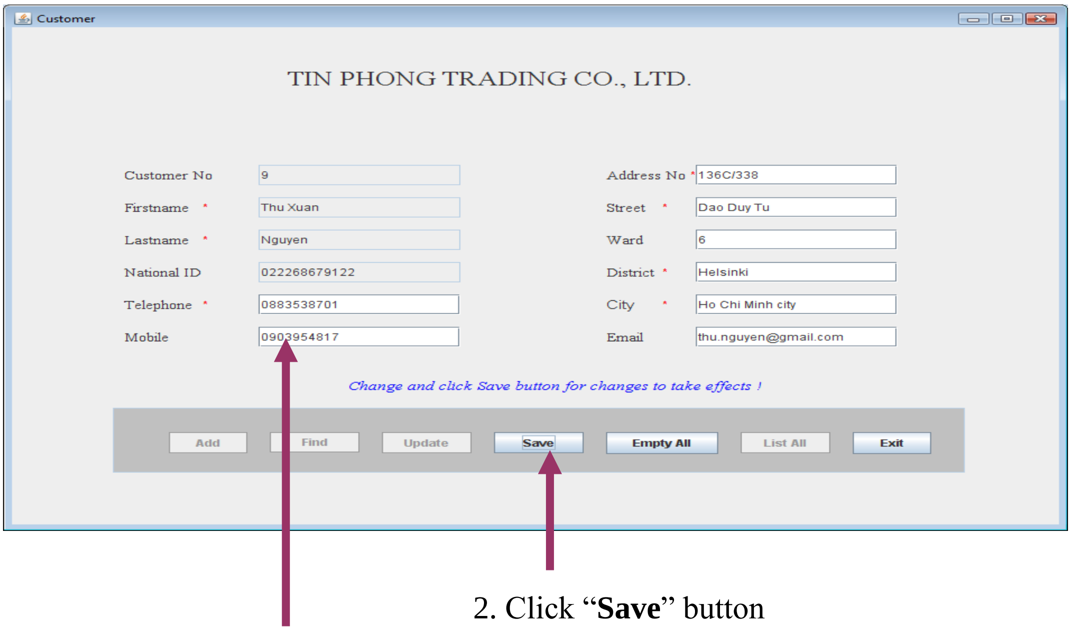
2. Click "**Select**" button

Picture 10: List of Customers screen

Now that the customer information is in front of you (on screen) you can start manipulating it. Let see how one can change the customer information.

- **Change existing customer's information**

Look at careful the Customer window you see that the information displayed is protected and thus can not be changed. It has been thought to display it in this way to avoid unnecessary or mistakenly changes but in the same way has been thought to reserve the rights to the use to start changing them according to his /her desires.
How to do it then?
By simply clicking "**Update**" button the fields that are allowed to be changed will become white and thus letting the user manipulating the information they hold.
Once you have finished the fields you want to change click "**Save**" button to let the system store the new customer's information. After the "**Save**" button has been clicked the message will be shown to you that "**The customer was successfully updated**" you do not have to do anything else you have reached your goal.



TIN PHONG TRADING CO., LTD.

Customer No	9	Address No *	136C/338
Firstname *	Thu Xuan	Street *	Dao Duy Tu
Lastname *	Nguyen	Ward	6
National ID	022268679122	District *	Helsinki
Telephone *	0883538701	City *	Ho Chi Minh city
Mobile	0903954817	Email	thu.nguyen@gmail.com

Change and click Save button for changes to take effects !

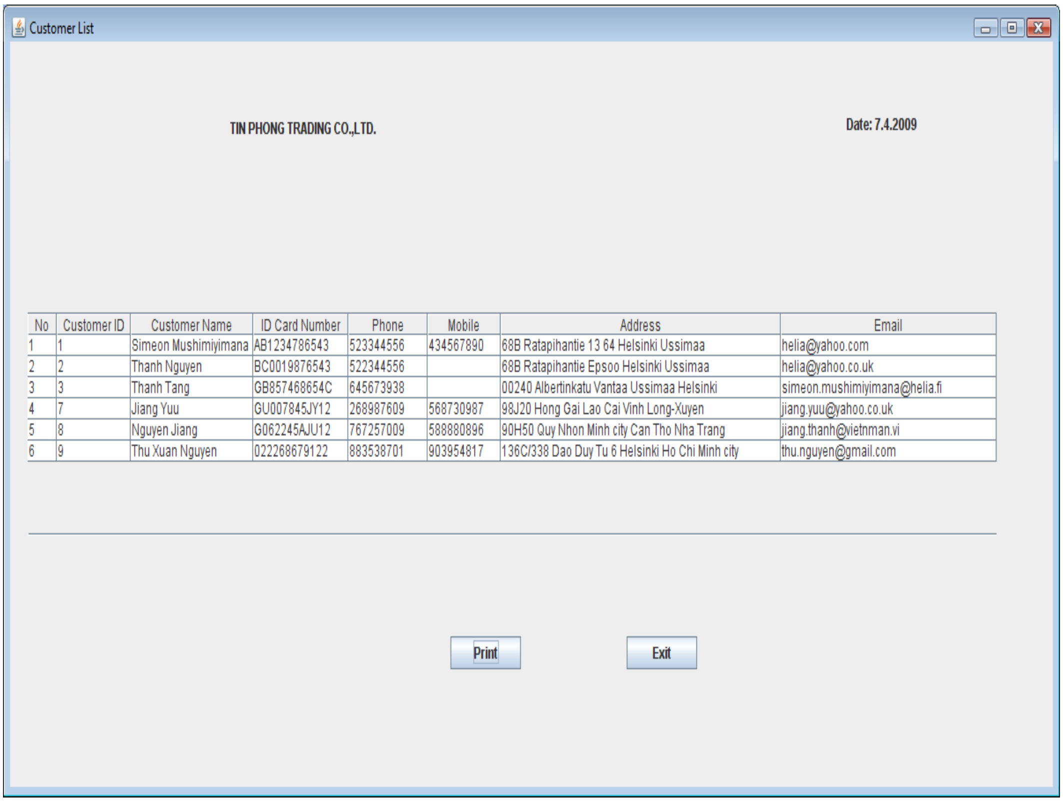
2. Click "**Save**" button

1. Field in changeable state

Picture 10: Customers screen (ready to be changed)

- Display a list of all customers in the system

If you want to see a list of all customers in the system just click on "**List All**" button and the following window will be opened to you.



TIN PHONG TRADING CO.,LTD.

Date: 7.4.2009

No	Customer ID	Customer Name	ID Card Number	Phone	Mobile	Address	Email
1	1	Simeon Mushimiyimana	AB1234786543	523344556	434567890	68B Ratapihantie 13 64 Helsinki Ussimaa	helia@yahoo.com
2	2	Thanh Nguyen	BC0019876543	522344556		68B Ratapihantie Epsoo Helsinki Ussimaa	helia@yahoo.co.uk
3	3	Thanh Tang	GB857468654C	645673938		00240 Albertinkatu Vantaa Ussimaa Helsinki	simeon.mushimiyimana@helia.fi
4	7	Jiang Yuu	GU007845JY12	268987609	568730987	98J20 Hong Gai Lao Cai Vinh Long-Xuyen	jiang.yuu@yahoo.co.uk
5	8	Nguyen Jiang	G062245AJU12	767257009	588880896	90H50 Quy Nhon Minh city Can Tho Nha Trang	jiang.thanh@vietnman.vi
6	9	Thu Xuan Nguyen	022268679122	883538701	903954817	136C/338 Dao Duy Tu 6 Helsinki Ho Chi Minh city	thu.nguyen@gmail.com

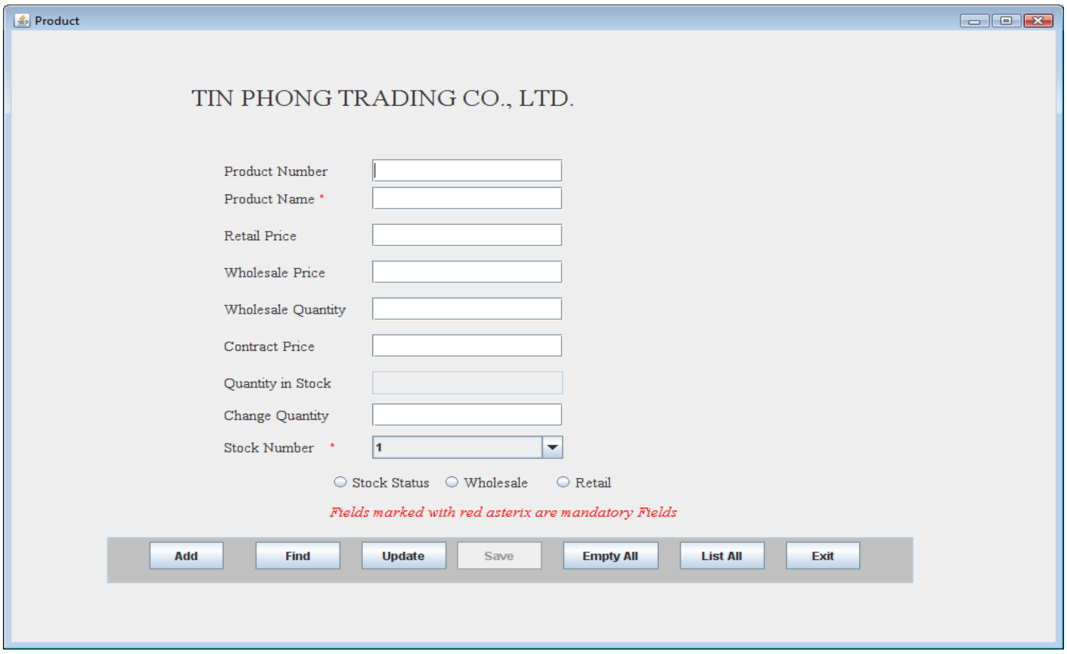
Picture 11: Customer list screen

Remember also that from this window you can print out the list that is displayed to you, to do so simply click on "**Print**" button. There is no any other information while printing just the general printing rules applied also while printing from this window.

3.4 Manage Product

Manage Product information follows the same procedures like the ones that are explained in “**Manage Customer**”, read them careful, step by step they will guide you on how to manage Product information.

Here is product window to Manage Product information.



TIN PHONG TRADING CO., LTD.

Stock Number * 1

Fields marked with red asterix are mandatory Fields

Picture 12: Product screen

Note: - Information about Stock status, Wholesale and Retail functionalities is not used at the moment and thus not described, it will be described when other part of this application will be ready.

-Also for stock status instead of filling the field, select on of the information provided to, the stock status information exists already in the system.

3.5 Manage Sales

Managing contract includes a set of activities which are;

- Retail Sales
- Wholesale Sales
- Contract Sales

The two pervious ones are not implemented yet thus not described in this document. Let use now go into details about how contract sales are managed. First of all keep it in your mind that contract sales in not an event, it is instead a group of services that by the time being can be done (with this sub-system), thus contract sales refers to;

- Contract
- Delivery
- Payment

For contract we perform the following activities.

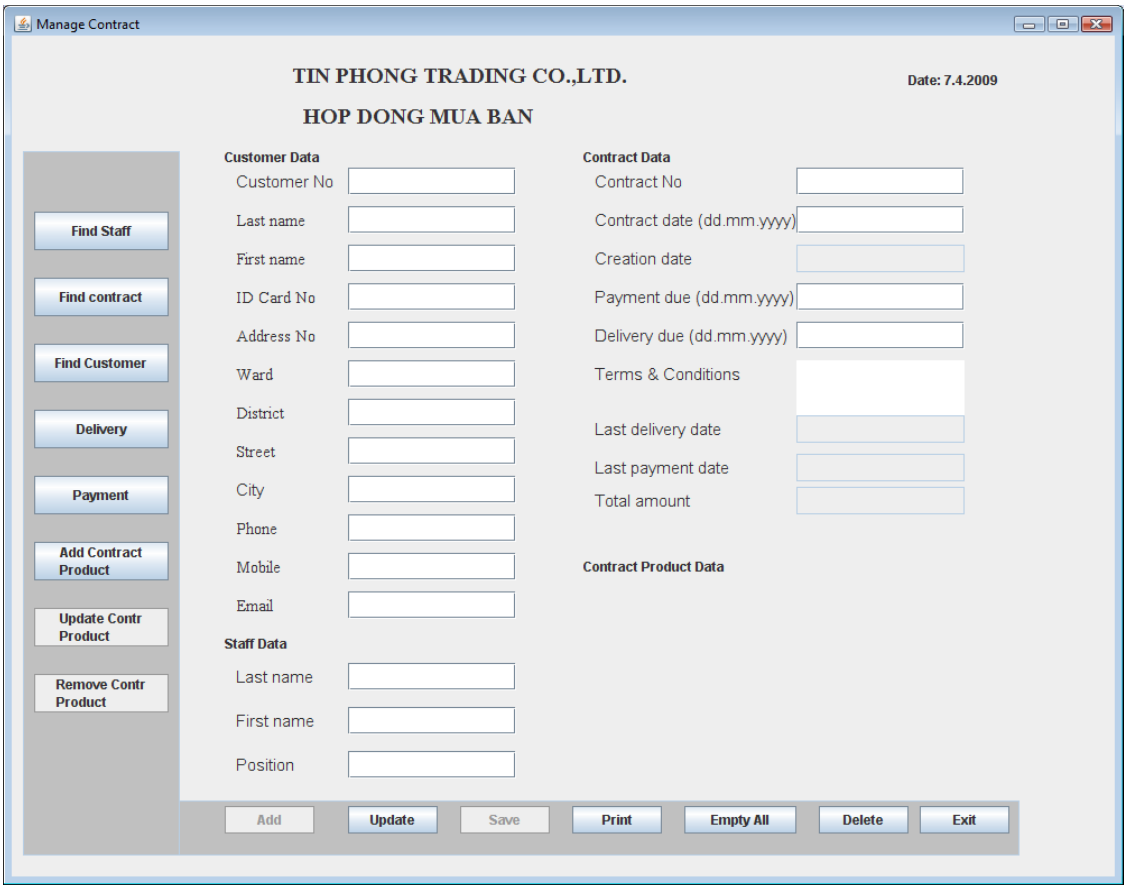
- Add new contract
- View Contract
- Print contract
- Update contract
- Remove contract

With payment and delivery the following activities are also done;

- Add new (payment or delivery)
- List all (payments or deliveries)
- Print list (payments or deliveries)

3.5.1 Contract

If you want to manage a contract just click on "**Manage contract**" button, then you will see the following window.



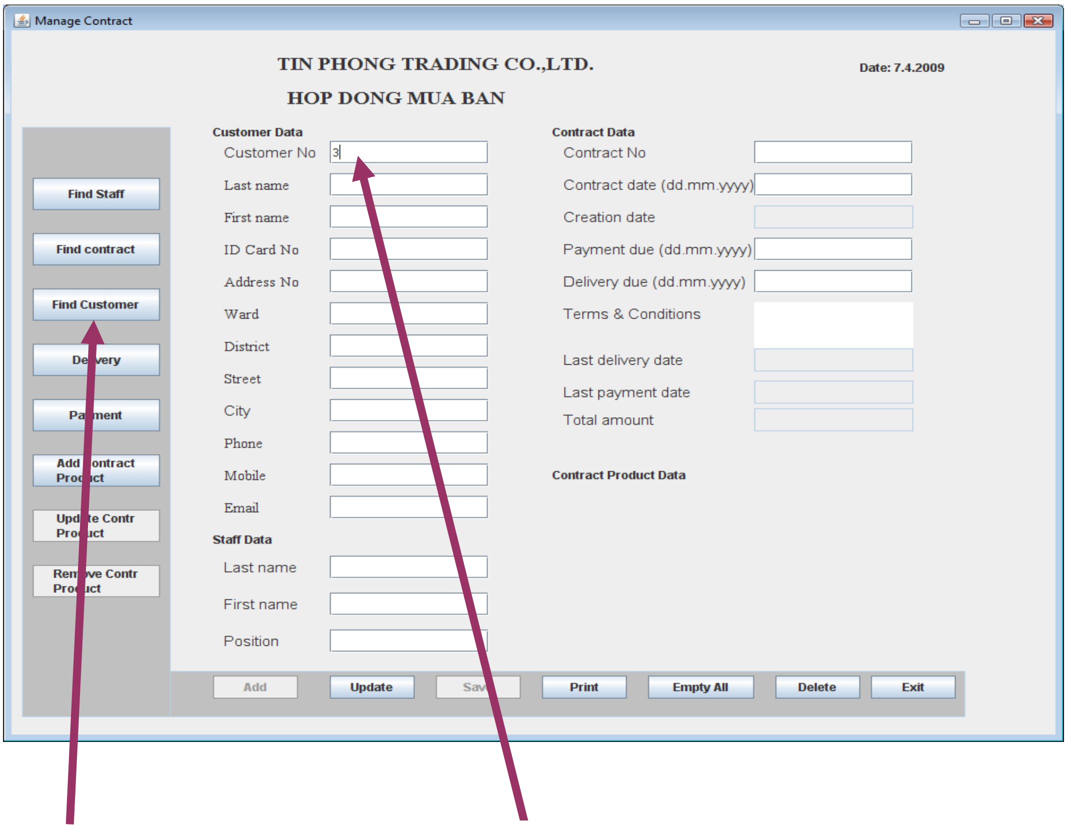
Picture 13: Contract screen

- Adding new contract

While adding new contract, the customer information, staff and product data is needed. However, this data is not entered by the user; it has to be existed in the system and to find this information the steps that were used to search customer information are the same here also for finding customer information to be put in contract.

See: Manage Customer

-Enter either customer Number or customer first name and click "**Find Customer**" button.



Manage Contract

TIN PHONG TRADING CO.,LTD.

HOP DONG MUA BAN

Date: 7.4.2009

Customer Data

Customer No: 3

2. Click "**Find Customer**" button.

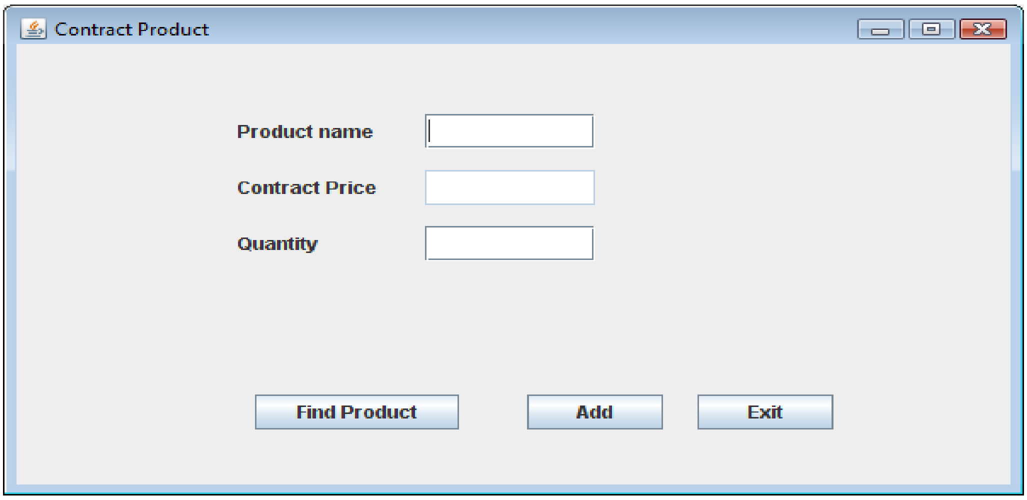
1. Enter Customer number, for example 3.

Picture 14: Contract screen (filled for searching)

After the button is clicked, customer information will be shown to you on "**Manage contract widow",** do the same for the staff by entering staff first name and click **"Find Staff"** button, the staff fields will be filled and now what you need to do is to fill all other fields, remember that all fields are mandatory, thus have all to be filled.

Now that you have filled these fields, you need to add product information because you will not be able to add a new contract without product information. How to do it ?

If you can identify the button "**Add Contract Product**" go ahead and click on it; the following window will be opened to you to fill product information.



Contract Product

Product name

Contract Price

Quantity

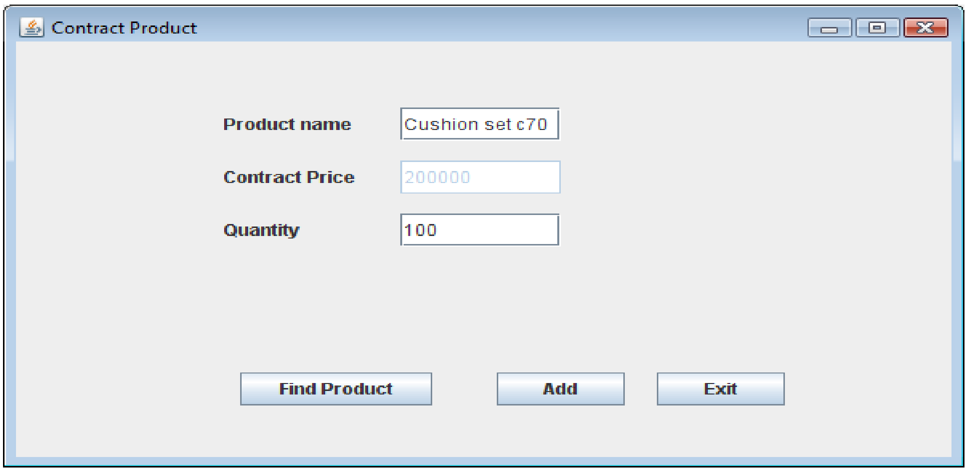
Picture 15: Contract Product screen

-Enter product name you want to add in product name field.

- Click on "**Find Product**" button, it might be possible that you have entered a part of product name that might result in a list of products;

Refer to the **Manage product**, to perform your search, when you are ready you will see the product name and contract price filled, you can not change neither of them; add quantity and click on "**Add"** button.

ContractProduct screen allows adding new contract product into the system.



Product name: Cushion set c70

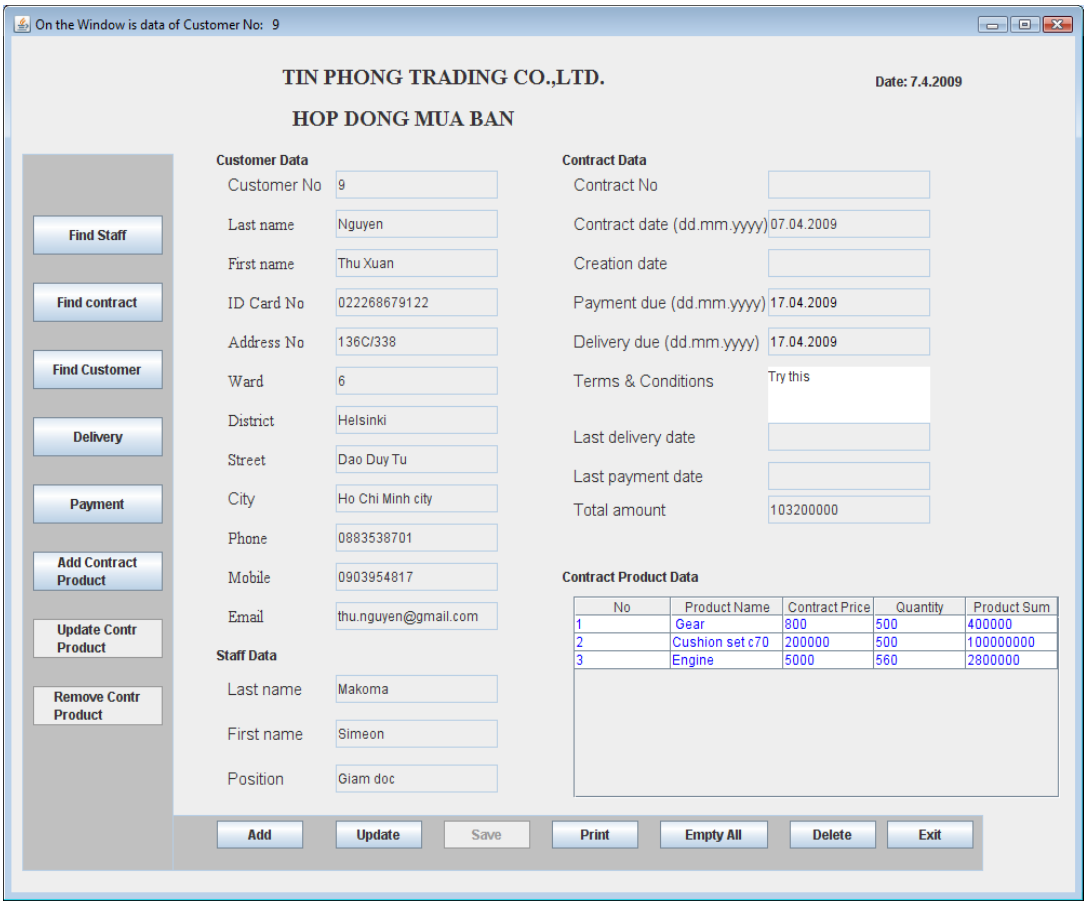
Contract Price: 200000

Quantity: 100

,

Picture 16: Contract product screen

Continue as you have added as many products as you like. At the end you have contract window filled and it will look like the one on next page.



On the Window is data of Customer No: 9

TIN PHONG TRADING CO.,LTD.

HOP DONG MUA BAN

Date: 7.4.2009

Customer Data

Customer No	9
Last name	Nguyen
First name	Thu Xuan
ID Card No	022268679122
Address No	136C/338
Ward	6
District	Helsinki
Street	Dao Duy Tu
City	Ho Chi Minh city
Phone	0883538701
Mobile	0903954817
Email	thu.nguyen@gmail.com

Staff Data

Last name	Makoma
First name	Simeon
Position	Giam doc

Contract Data

Contract No	
Contract date (dd.mm.yyyy)	07.04.2009
Creation date	
Payment due (dd.mm.yyyy)	17.04.2009
Delivery due (dd.mm.yyyy)	17.04.2009
Terms & Conditions	Try this
Last delivery date	
Last payment date	
Total amount	103200000

Contract Product Data

No	Product Name	Contract Price	Quantity	Product Sum
1	Gear	800	500	400000
2	Cushion set c70	200000	500	100000000
3	Engine	5000	560	2800000

Picture 17: Contract screen (with a list of contract product)

-The contract date is the date when the customer is going to sign the contract, thus it can not be earlier that today's date.

-The Payment due date and Delivery due date is at least one day from the Contract date.

Note: Dates are checked by the system enter them as in really life while respecting the format that has been given to you.

-If you notice that you made a mistake and you want to change some thing especially in product information;

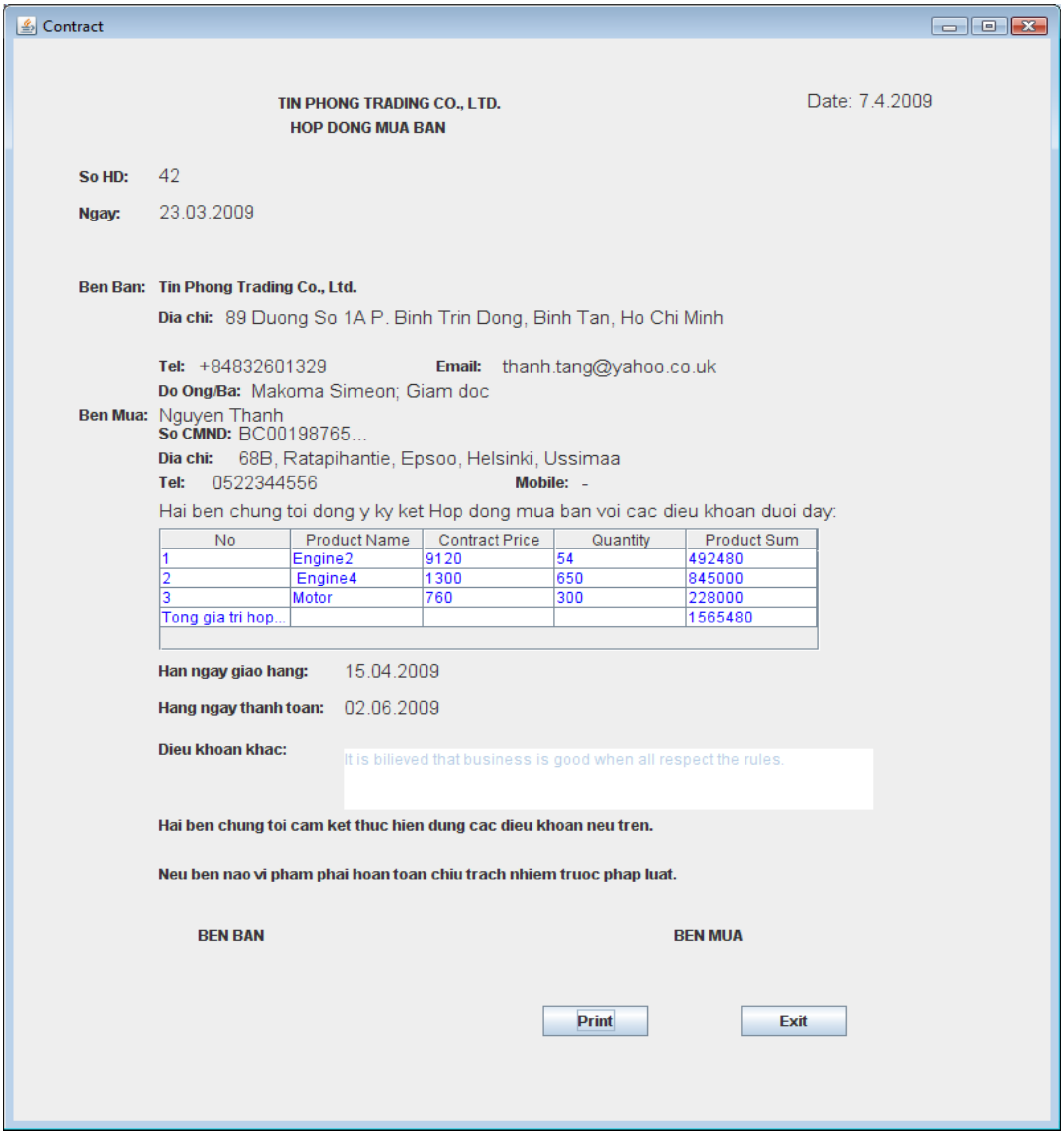
-To change quantity; Find the product name for which you want to change quantity enter a new quantity, and click "**Add**" button on Contract Product window, the system will do the rest for you.

-To remove whole product information; select the row you want to remove form the table and click on "**Remove Cont Product**" the system is ready to remove it for you and you should not worry about money all calculations will be done at the same time.

You don't have to enter the contract number or worry about it, it will be generated by the system.

-Is your information ready? Click "**Add**" button to add your contract.
- Now if the customer is ready to sign the contract what you need to do is just click on "**Print**" button and to open your contract in printable format and thus print it out.

Contract screen in Printable format



TIN PHONG TRADING CO., LTD.
HOP DONG MUA BAN

Date: 7.4.2009

So HD: 42

Ngay: 23.03.2009

Ben Ban: **Tin Phong Trading Co., Ltd.**

Dia chi: 89 Duong So 1A P. Binh Trin Dong, Binh Tan, Ho Chi Minh

Tel: +84832601329 **Email:** thanh.tang@yahoo.co.uk

Do Ong/Ba: Makoma Simeon; Giam doc

Ben Mua: Nguyen Thanh

So CMND: BC00198765...

Dia chi: 68B, Ratapihantie, Epsoo, Helsinki, Ussimaa

Tel: 0522344556 **Mobile:** -

Hai ben chung toi dong y ky ket Hop dong mua ban voi cac dieu khoan duoi day:

No	Product Name	Contract Price	Quantity	Product Sum
1	Engine2	9120	54	492480
2	Engine4	1300	650	845000
3	Motor	760	300	228000
Tong gia tri hop...				1565480

Han ngay giao hang: 15.04.2009

Hang ngay thanh toan: 02.06.2009

Dieu khoan khac: It is bilieved that business is good when all respect the rules.

Hai ben chung toi cam ket thuc hien dung cac dieu khoan neu tren.

Neu ben nao vi pham phai hoan toan chiu trach nhiem truoc phap luat.

BEN BAN **BEN MUA**

Picture 18: Contract screen (printable format)

When you print the contract you get it on a normal paper and it is ready to be signed by both customer and Company representative.

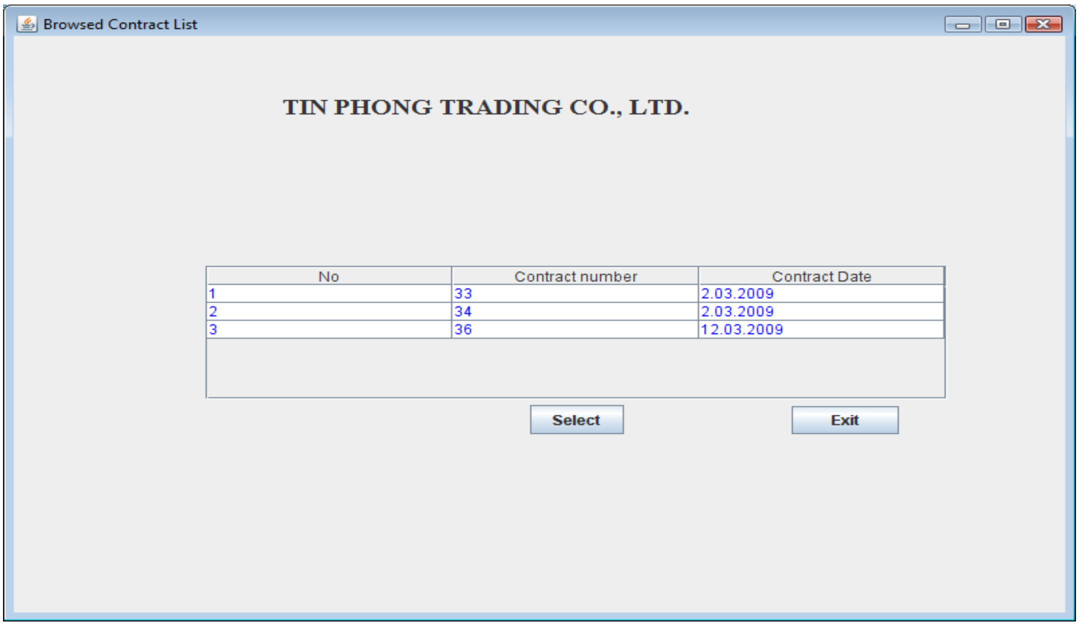
-**Changing Contract Information.**

Change contract information includes search contract, so, to search contract information, you can simply enter contract number and click find contract or start by searching customer and once you have customer information one screen you can also easily click "**Find contract**" button.

-If the customer in question has only one contract with the company, the contract information will be shown on screen.

-On other hand if the customer has more than one contract a list of contract will be shown on contract list and there also you will need to select one contract to be shown.

Contract list looks like this.



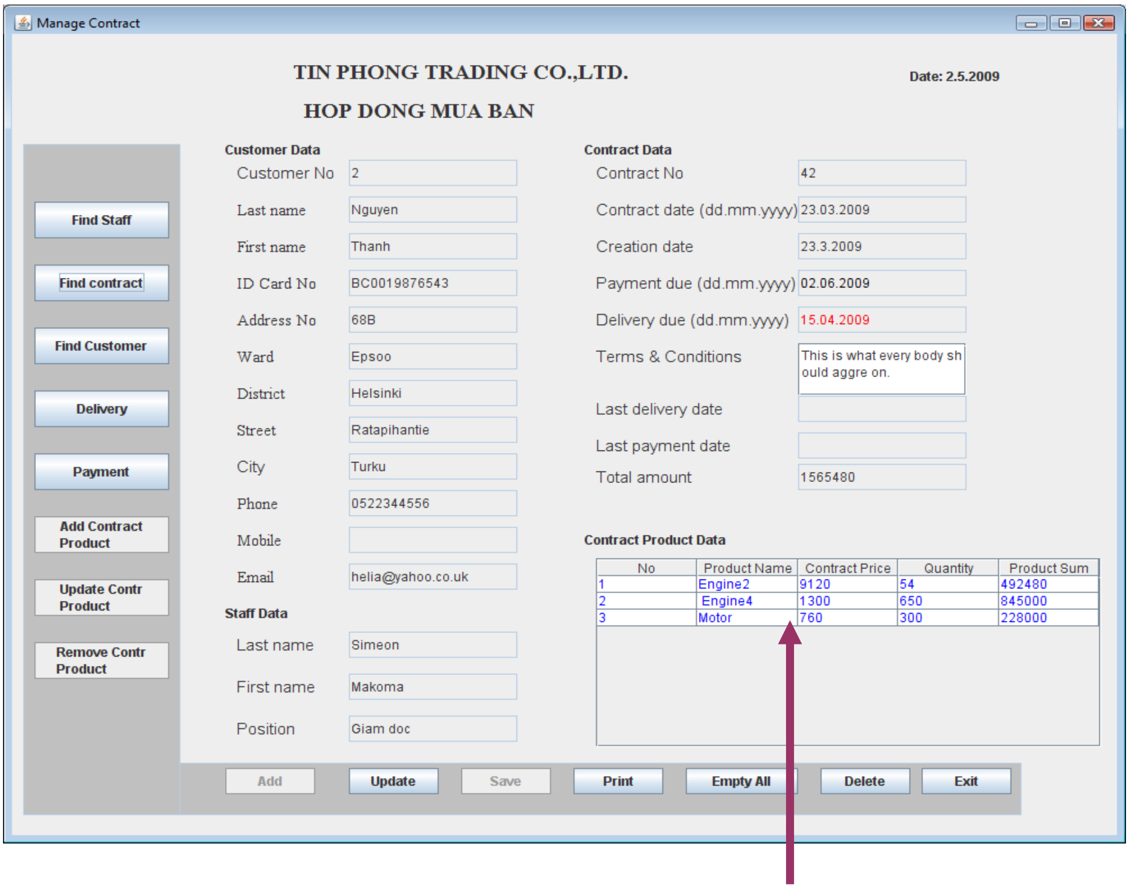
TIN PHONG TRADING CO., LTD.

No	Contract number	Contract Date
1	33	2.03.2009
2	34	2.03.2009
3	36	12.03.2009

Picture 19: List of contracts screen

Now that your contract information is on screen, you are going to start changing it,

-Click on “**Update**” button, if the customer has not started paying the contract or the company has not started the delivery, then, you will be able to change it; in all other cases you won’t be able to change the contract. Not all contract information can be changed; when the “**Update**” button is clicked the fields that can be updated will change their background colour to white thus letting you change their content. The contract product data can also be changed either by adding new product or removing it or just by changing the contract quantity of a given contract in contract product list. Below is a screen showing contract data that is ready to be changed.



TIN PHONG TRADING CO.,LTD.

HOP DONG MUA BAN

Date: 2.5.2009

Customer Data

Customer No	2
Last name	Nguyen
First name	Thanh
ID Card No	BC0019876543
Address No	68B
Ward	Epsoo
District	Helsinki
Street	Ratapihantie
City	Turku
Phone	0522344556
Mobile	
Email	helia@yahoo.co.uk

Staff Data

Last name	Simeon
First name	Makoma
Position	Giam doc

Contract Data

Contract No	42
Contract date (dd.mm.yyyy)	23.03.2009
Creation date	23.3.2009
Payment due (dd.mm.yyyy)	02.06.2009
Delivery due (dd.mm.yyyy)	15.04.2009
Terms & Conditions	This is what every body sh ould aggre on.
Last delivery date	
Last payment date	
Total amount	1565480

Contract Product Data

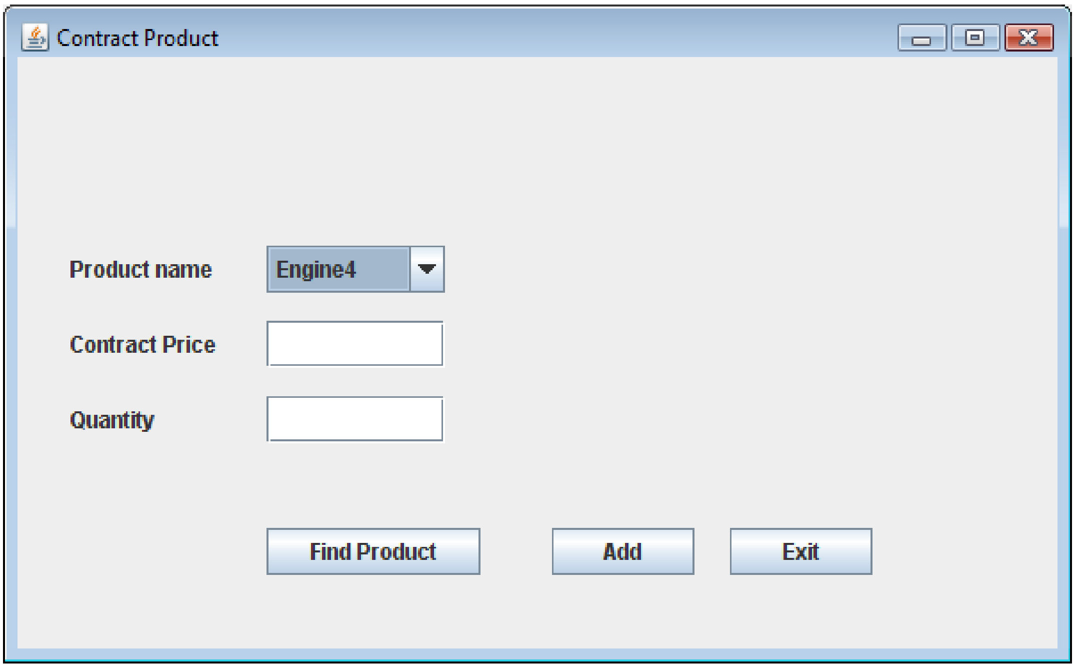
No	Product Name	Contract Price	Quantity	Product Sum
1	Engine2	9120	54	492480
2	Engine4	1300	650	845000
3	Motor	760	300	228000

One table row represents
one contract product

Picture 20: Contract screen (ready to be updated)

-In case you want to remove one contract product you have to select one row in the table and then click "**Remove Contr Product**" button.

-Or if you are changing quantity for one product, click on "**Update Contr Product**" you will be directed to the following screen;



Contract Product

Product name: Engine4

Contract Price:

Quantity:

Picture 21: Contract product screen (for updating contract)

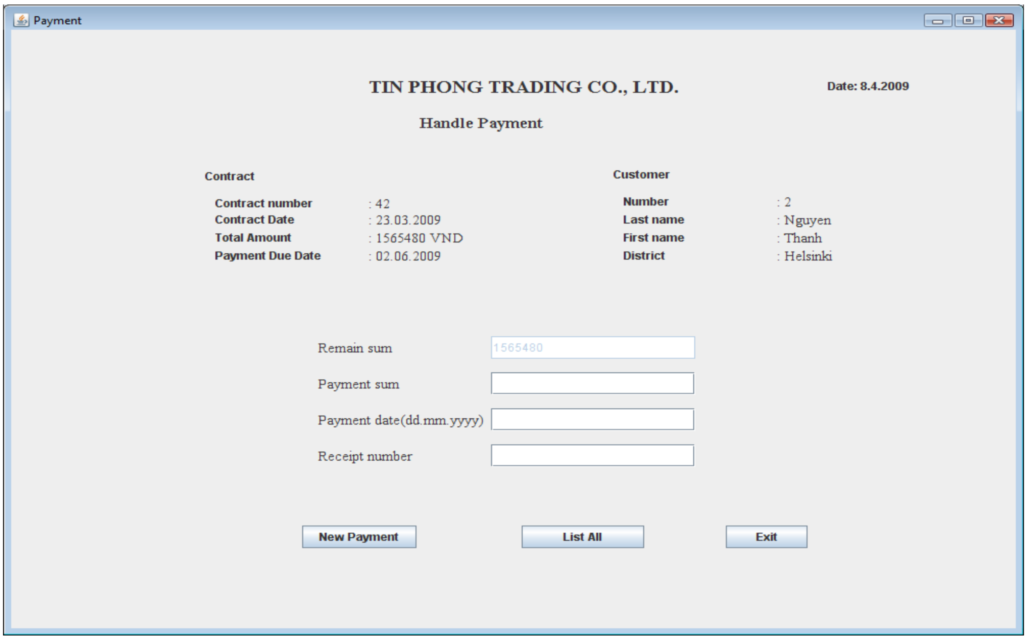
Select the same product name as the one in table that you want to change and then click "**Find Product**" button, enter the quantity you need to have and finally click on "**Add**" button. Your contract product has been already changed and you can notice changes in quantity, in product sum as well as Total Amount.

With this screen you can also add a new contract product by selecting a product that does not appear in the table and proceed as in the previous step. The result is the same as in previous step.

- It is also possible to cancel the contract if the customer has not started paying it yet or if the company did not start delivering its products to the customer. In this case you just simply click "**Delete**" button and the contract will be cancelled from the system as well as all contract products information relating to this particular contract.

3.5.2 Payment

To make a payment click on “**Payment**” button and you will se the following window opened to you.



TIN PHONG TRADING CO., LTD.

Date: 8.4.2009

Handle Payment

Contract

Contract number	: 42
Contract Date	: 23.03.2009
Total Amount	: 1565480 VND
Payment Due Date	: 02.06.2009

Customer

Number	: 2
Last name	: Nguyen
First name	: Thanh
District	: Helsinki

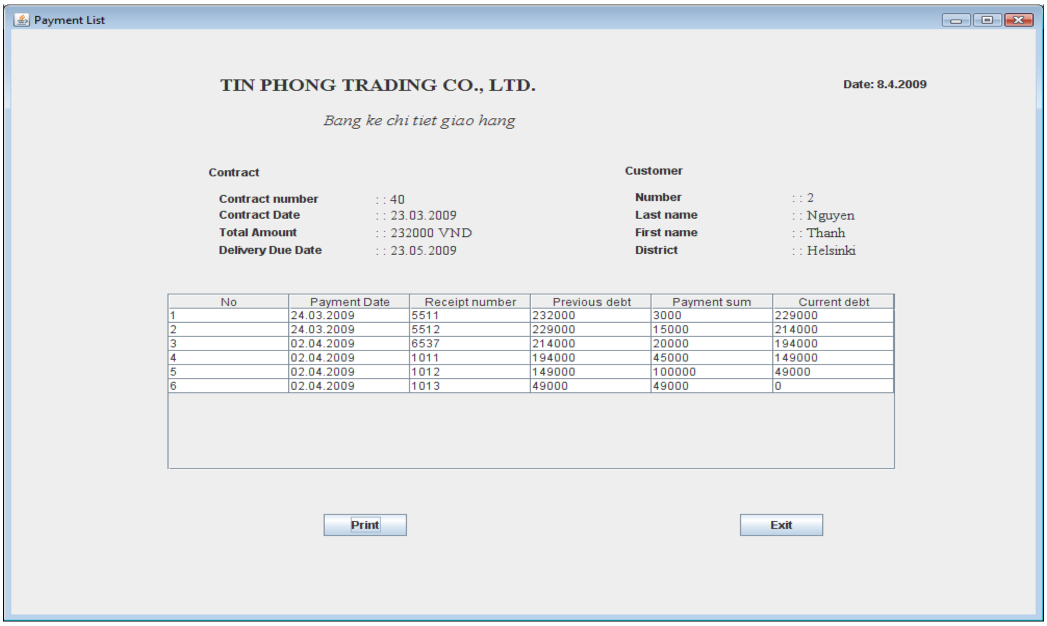
Remain sum	1565480
Payment sum	
Payment date(dd.mm.yyyy)	
Receipt number	

Picture 22: Payment screen

-Enter the amount of money you want to pay, the date has to be from the current date backwards and provide the receipt number. If you are done click “**New Payment**” the information about the payment will be saved.

-If you want to see the list of all payment for the current contract, click "**List All**" button you will see the following screen from which you can either print by clicking "**Print**" button or close by clicking "**Exit**" button.

List of Payment that belong to one contract



TIN PHONG TRADING CO., LTD.

Date: 8.4.2009

Bang ke chi tiet giao hang

Contract

Contract number : : 40
Contract Date : : 23.03.2009
Total Amount : : 232000 VND
Delivery Due Date : : 23.05.2009

Customer

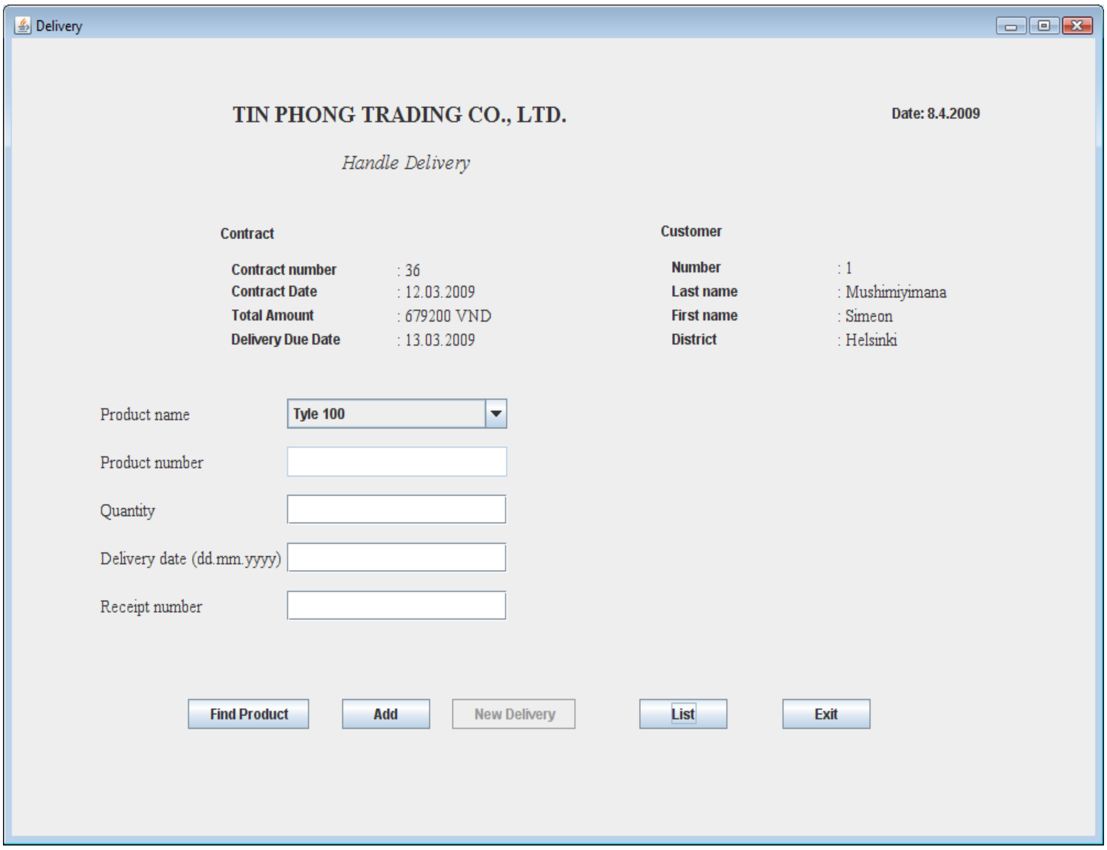
Number : : 2
Last name : : Nguyen
First name : : Thanh
District : : Helsinki

No	Payment Date	Receipt number	Previous debt	Payment sum	Current debt
1	24.03.2009	5511	232000	3000	229000
2	24.03.2009	5512	229000	15000	214000
3	02.04.2009	6537	214000	20000	194000
4	02.04.2009	1011	194000	45000	149000
5	02.04.2009	1012	149000	100000	49000
6	02.04.2009	1013	49000	49000	0

Picture 23: List of payments

3.5.3 Delivery

To make a delivery click on "**Delivery**" button which will open the Delivery window as follows;



Delivery

TIN PHONG TRADING CO., LTD.

Date: 8.4.2009

Handle Delivery

Contract

Contract number	: 36
Contract Date	: 12.03.2009
Total Amount	: 679200 VND
Delivery Due Date	: 13.03.2009

Customer

Number	: 1
Last name	: Mushimiyimana
First name	: Simeon
District	: Helsinki

Product name: Tyle 100

Product number

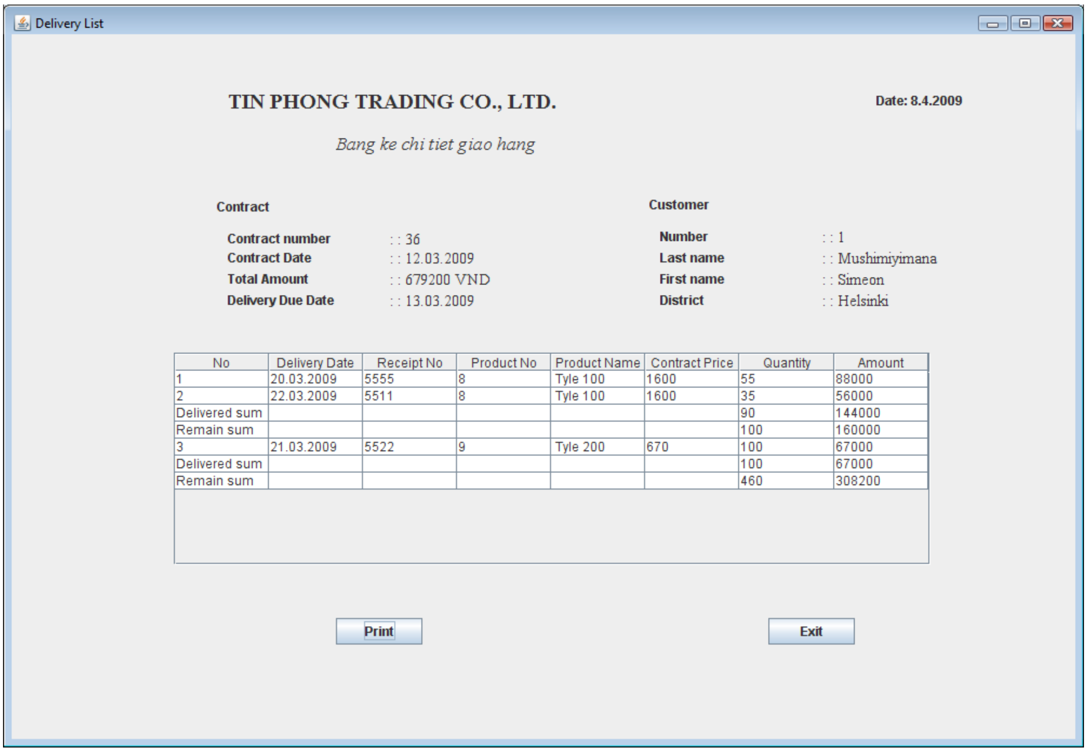
Quantity

Delivery date (dd.mm.yyyy)

Receipt number

Picture 24: Delivery screen

-The list for products you can see, is a list of only those product in the contract, select one product to deliver, and then "**Find Product**" button, enter quantity to be delivered and delivery date and click "**Add**" button; do this as for many products as you want to delivery for this date, if you think you have enough product in the list then click "**New Delivery**" button, the information about this delivery will be saved and you can see it by simply clicking on "**List All**" button to see the list of all deliveries including the one you have made few minutes ago. See an example of the picture bellow.



TIN PHONG TRADING CO., LTD.

Date: 8.4.2009

Bang ke chi tiet giao hang

Contract

Contract number	: : 36
Contract Date	: : 12.03.2009
Total Amount	: : 679200 VND
Delivery Due Date	: : 13.03.2009

Customer

Number	: : 1
Last name	: : Mushimiyimana
First name	: : Simeon
District	: : Helsinki

No	Delivery Date	Receipt No	Product No	Product Name	Contract Price	Quantity	Amount
1	20.03.2009	5555	8	Tyle 100	1600	55	88000
2	22.03.2009	5511	8	Tyle 100	1600	35	56000
Delivered sum						90	144000
Remain sum						100	160000
3	21.03.2009	5522	9	Tyle 200	670	100	67000
Delivered sum						100	67000
Remain sum						460	308200

Picture 25: Delivery list screen

The deliveries list shows the total quantity delivered so far for one specific product as well as the remnant quantity for the same product; it shows also that was the date one which the delivery was made. This list can be printed out by clicking "**Prin**t" button, otherwise click "**Exit**" button to go back to the delivery screen.

3.6 Return Product

Return Product was not implemented yet, thus not described, the information about how to use it will be provided when it will be ready for use.

3.7 Manage Feedback

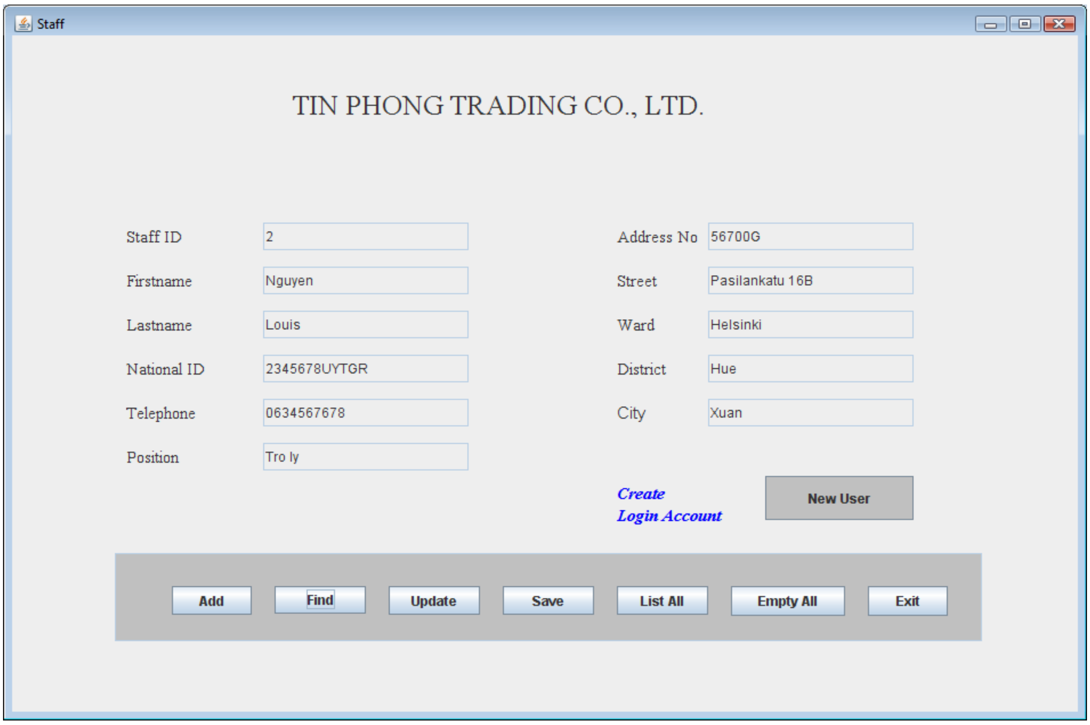
Manage Feedback was not implemented yet, thus not described, the information about how to use it will be provided when it will be ready for use.

3.8 Manage Report

Manage Report was not implemented yet, thus not described, the information about how to use it will be provided when it will be ready for use.

3.9 Manage Staff

Manage Staff is also done in the same way Manage customer information, however there is one functionality that needs to be described here that is how to create system user after a staff information can be found in the system.



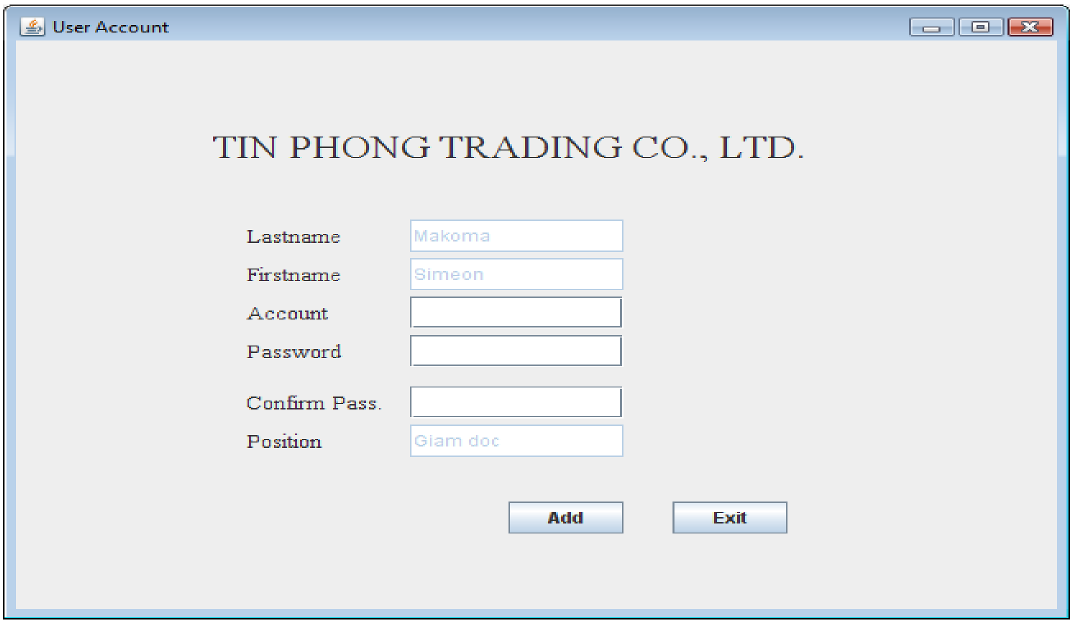
TIN PHONG TRADING CO., LTD.

Staff ID	2	Address No	56700G
Firstname	Nguyen	Street	Pasilankatu 16B
Lastname	Louis	Ward	Helsinki
National ID	2345678UYTGR	District	Hue
Telephone	0634567678	City	Xuan
Position	Tro ly		

Picture 26: Product screen

3.9. 1 Create user account

When you have staff information on screen, you can create the logging account for him/her by clicking "**New User**" button which will open the user account window



TIN PHONG TRADING CO., LTD.

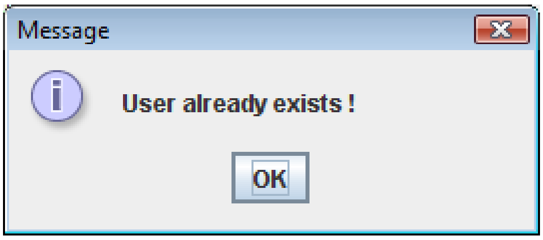
Lastname	Makoma
Firstname	Simeon
Account	
Password	
Confirm Pass.	
Position	Giam doc

Picture 27: User account screen

The information about staff such as last name, first name and position is filled automatically; enter the login account.

- Choose a suitable password (has to contain at least one digit from 0-9 and has to have all together at least 5 characters).
- Confirm password, for you to remember, both passwords have to be the same.
- Click on "**Add**" to save your information.

The system will check if the staff you are entering does not have a user account, if not then the information will be saved otherwise you will get the following message.

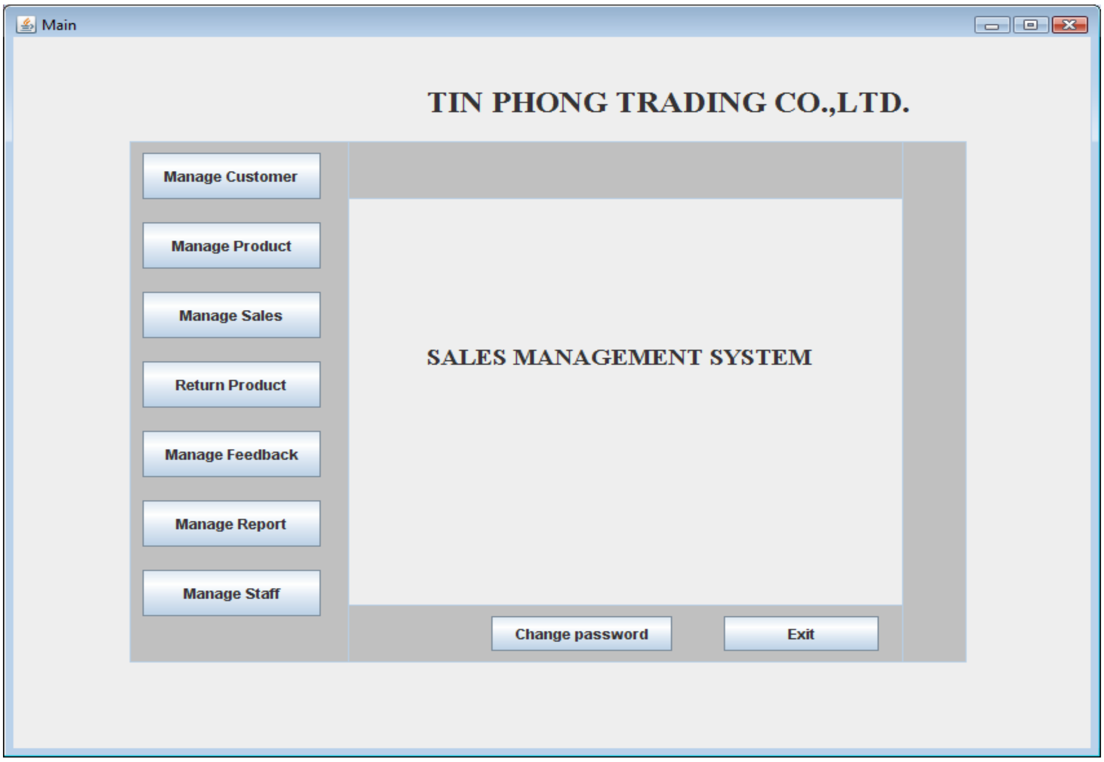


Message

User already exists !

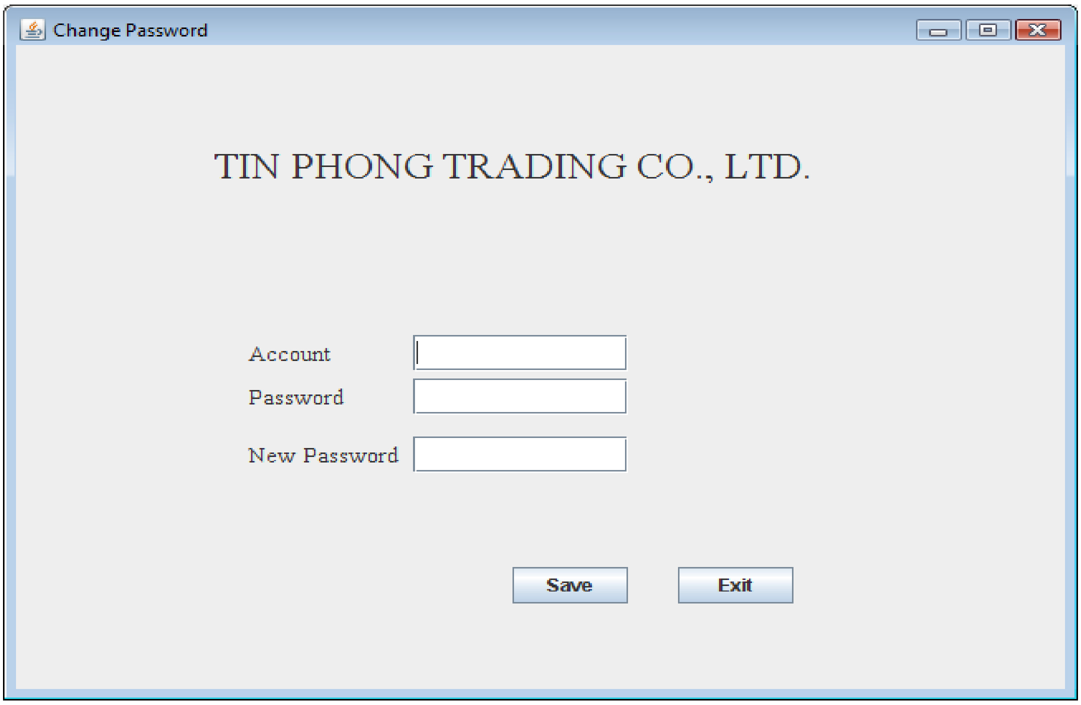
3.9. 2 Change password

On the main window, after logging in you can change your password by clicking on "**Change password**" button.



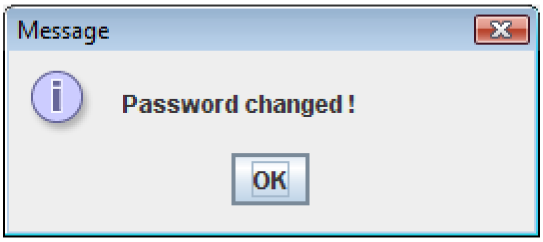
Picture 28: Main screen (change password)

When “**Change password**” is clicked the Change password window opens and by providing your user account and old password you can change your old password to a new one



Picture 29: Change Password screen

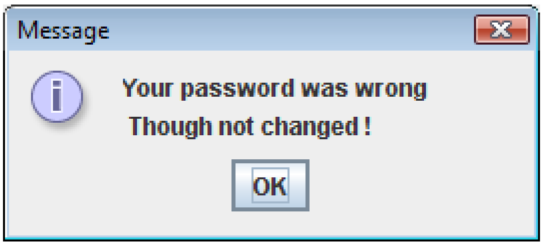
Enter your information as mentioned above and click”**Save**” button, if you get this message



Password changed !

Your password has been changed you should remember the new one as the old is not in use anymore.

Otherwise if you get this message



Message

Your password was wrong
Though not changed !

It means that the password was not changed you should try again or keep the old one.

Appendices

Appendix 1: List of changes

Appendix 2: Terms of commands used

When working on this project time has come, that it was necessary to make some changes but due to the time shortage, these changes could not be made in each and every document that is why they are reported in this appendix which is annexed to both Design document and User guide document.

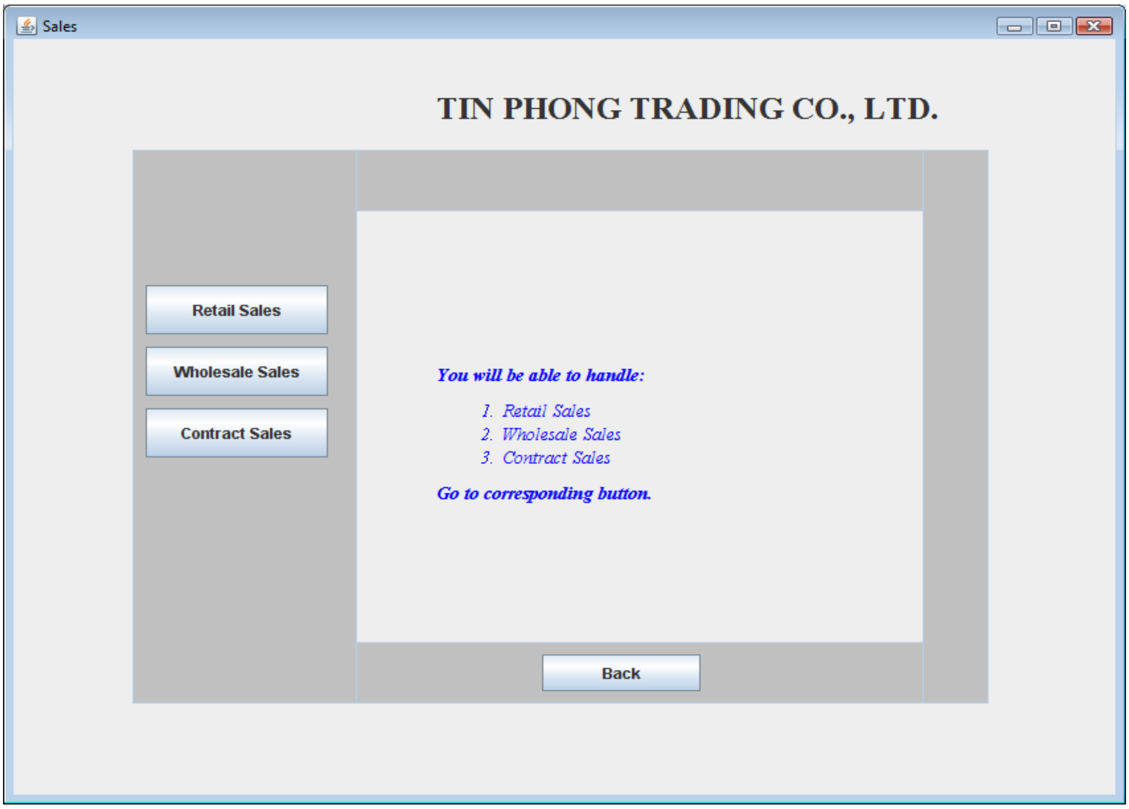
Changes that have been made affected only, terms of commands used in this sub-system and some screens that have changed their outlook.

Terms of command used

-After analysing the meaning of the word "**Exit**" in software development and how it should be used; Except on "**Main window**" where the command name was and remained "**Exit**" and on "**LogIn window**" where the command name was "**Exit**" and has been changed to "**Cancel**" It has been decided to change the name "**Exit**" to "**Back**" and this means that apart from the exceptions mentioned above wherever we have described "**Exit**" keep in your mind that the functionality it had has been assigned to "**Back**" command.

Screens (Windows)

For perfection purpose some screens have changed the outlook but activities that were described to be performed are still the same. This can be seen for example on Main window where according to the instructions in both "Design document" and "User guide", it was said that when you click "**Manage sales**" you go directly to contract window, but currently is not the case because now you go to the following window



Sales

TIN PHONG TRADING CO., LTD.

You will be able to handle:

1. *Retail Sales*
2. *Wholesale Sales*
3. *Contract Sales*

Go to corresponding button.

And from here once you click the "**Contract sales**" the system opens contract window as it was described before. This change was made because we thought about when the Retail and Wholesale Sales will be implemented will make life easier thus letting the system be friendly user.

Closing the system

In the same way, it was said that the application can be closed or quitted from "**LogIn window**", but currently has been changed to "**Main Window**" and this is because after you have logged in to the system "**LogIn Window**" automatically disappears.

3.3.2 Terms of commands used

Command's name	Functionality
Add	This button is used when entering new data into the system.
Change password	Change password button is used to change old password to a new one.
Delivery	To manage delivery information; add new delivery listing all deliveries and thus printing out the delivery list.
Exit	Exit button is used to close the currently working window. But can also while closing the whole application in case of exiting the Login window.
Empty All	When the current window has been filled with data, and there is no way to change the data it can be possible to empty the current window with this command to be able to search new data or perform other activities.
Find	Used to search data from the system regardless on which searching criteria.
List All	To display data for the current use case, this is performed without any access key all related data is displayed regardless on size or amount available in the system.
Login	To authenticate to the system, for getting rights to perform any activity the actor needs to log in to the system.
Manage Customer	Found on Main Window, this button is used to display Customer window from which customer information can be easily managed.
Manage Feedback	This command is used to display Feedback window for viewing or adding new feedback into the system.
Manage Product	Manage product information such adding, updating, searching or listing all.

Command's name	Functionality
Manage Staff	Command that once performed will open the ManagingStaffWindow.
Manage Report	Display report window. For Adding, viewing or updating report.
Manage Sales	
New User	This command button is for creating new user account
Payment	Once clicked the Payment sub-window opens from which a new payment can be made or a list of all payments belonging to the contract can be displayed.
Return Product	Display Return window to manage returned product data.
Select	Once the current window displays a list that allows selection the actor will use this command button to let the system now on which item from the list the actor is interested.
Update	When updating data on current window and the fields on this window are disabled, the update command will enable only fields that can be changed for the actor to start changing the information displayed in these.

Table 3: Terms of commands used