Introduction:

This report aims to clarify the installation, explain the implementation of Ecommerce system by using Hibernate, Spring MVC to design the system back-end. By storing the database in PostGreSQL and json files, this system can be tested using Junit test. Moreover, this system contains all the requirements that stated in the system specification.

Business Requirement:

System Description

This is the ERP (Enterprise Resource Planning) system. More specifically, this system will manage all the buying as well as the warehouse problems and decentralize them as a whole system. The company will place orders to the providers and when all of the products arrived with the same amount that the company had booked, staffs will take note and this note will be stored in the Receiving Note. Moreover, any details about the products, providers and orders will also be stored in the database. In the products section, there would another section for categories to differentiate the type of the products. Then, a certain amount of a products that the customers choose will be delivered. This information about the delivery will be noted as well as the information of the customers. When the products are ready to be delivered, a sale invoice will be printed with the quantity and the total value that the customers must pay. And this sale invoice will also be stored in the database with the respective customers and the products.

Functionality:

Category:

In this section, users can add, update, delete and search categories by page.

Customer:

In this section, users can add, update, delete and search customers by page. Moreover, Users can search customers information by name, address and phone.

Delivery:

In this section, users can add, update, delete and search deliveries by page. Moreover, Users can search by id, add more details, search delivery by date and as well as a period of date.

Order:

In this section, users can add, update, delete and search orders by page. Moreover, Users can search by id, add more details, search delivery by date.

Product:

In this section, users can add, update, delete and search products by page.

Provider:

In this section, users can add, update, delete and search providers by page.

Receiving Note:

In this section, users can add, update, delete and search receiving note by page. Moreover, Users can search by id, add more details, search receiving note by date.

Sale Invoice:

In this section, users can add, update, delete and search sale invoice by page. Moreover, Users can search by id, add more details, search sale invoices by date, search sale invoices by period of date with staffID or customerID or without any staffID or customerID and search the total revenue made by a certain staff or by a certain customer or both of them in a period of time.

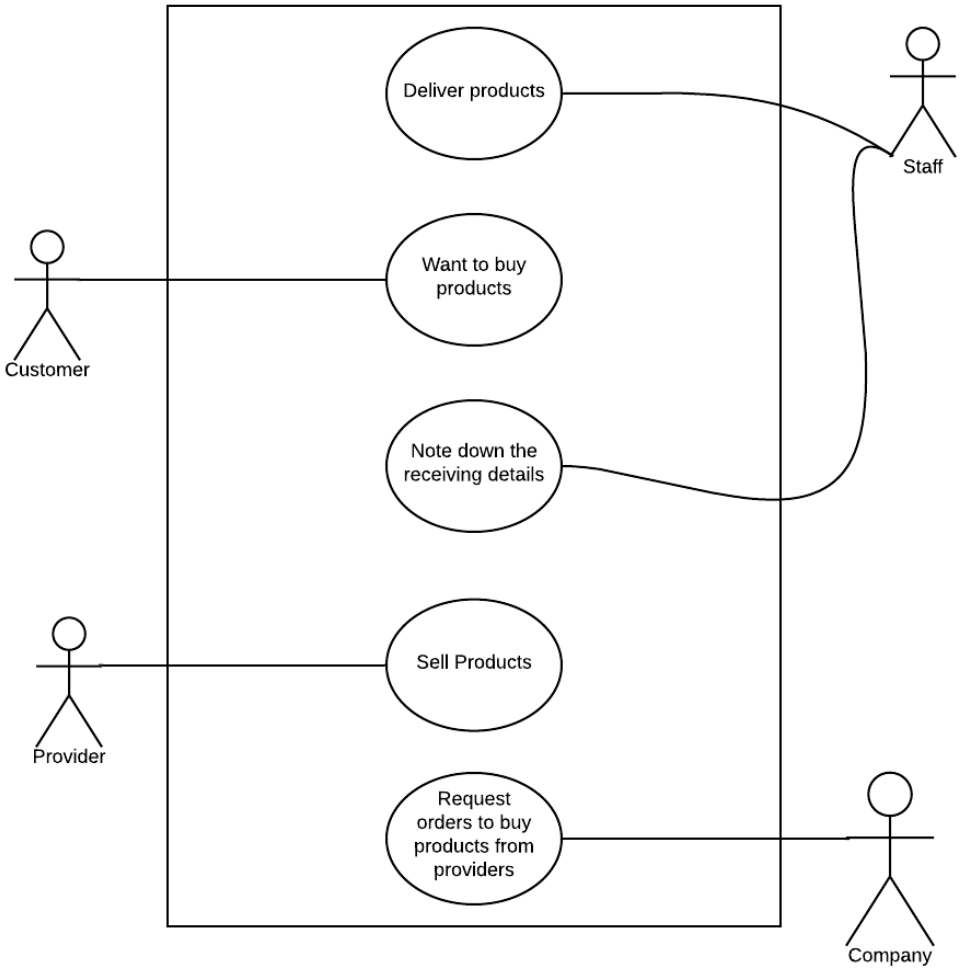
Staff:

In this section, users can add, update, delete and search staff by page. Moreover, Users can search customers information by name, address.

Inventory:

In this section, users can check whether statistic of the amount of delivery or received status of a product.

Use case Diagram:



Technology use:

Libraries:

Hibernate, PostGreSQL, Gson, Junit, Jackson and SpringMVC library.

Framework:

Spring Framework

Database:

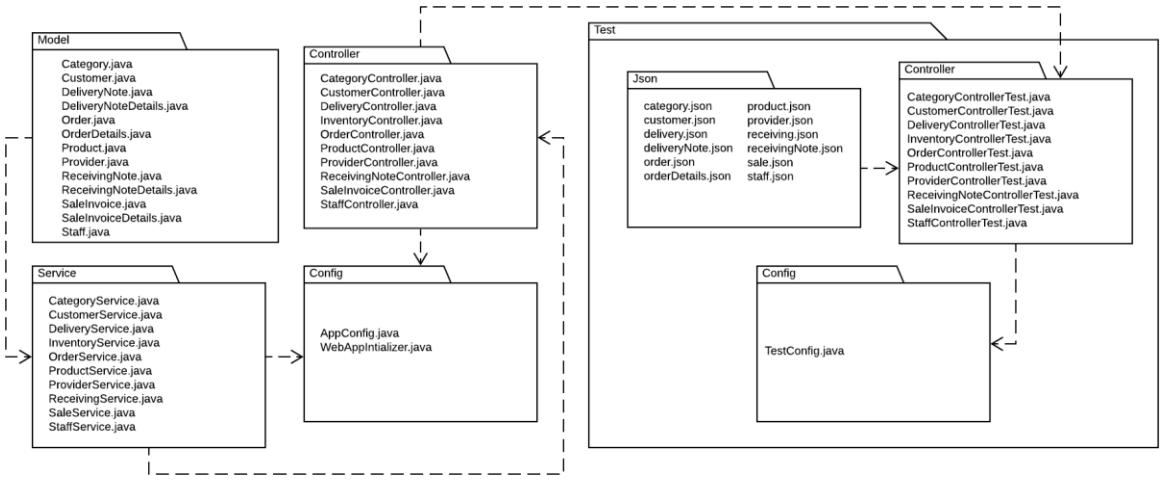
PostgreSQL

Programming Languages:

Java and json

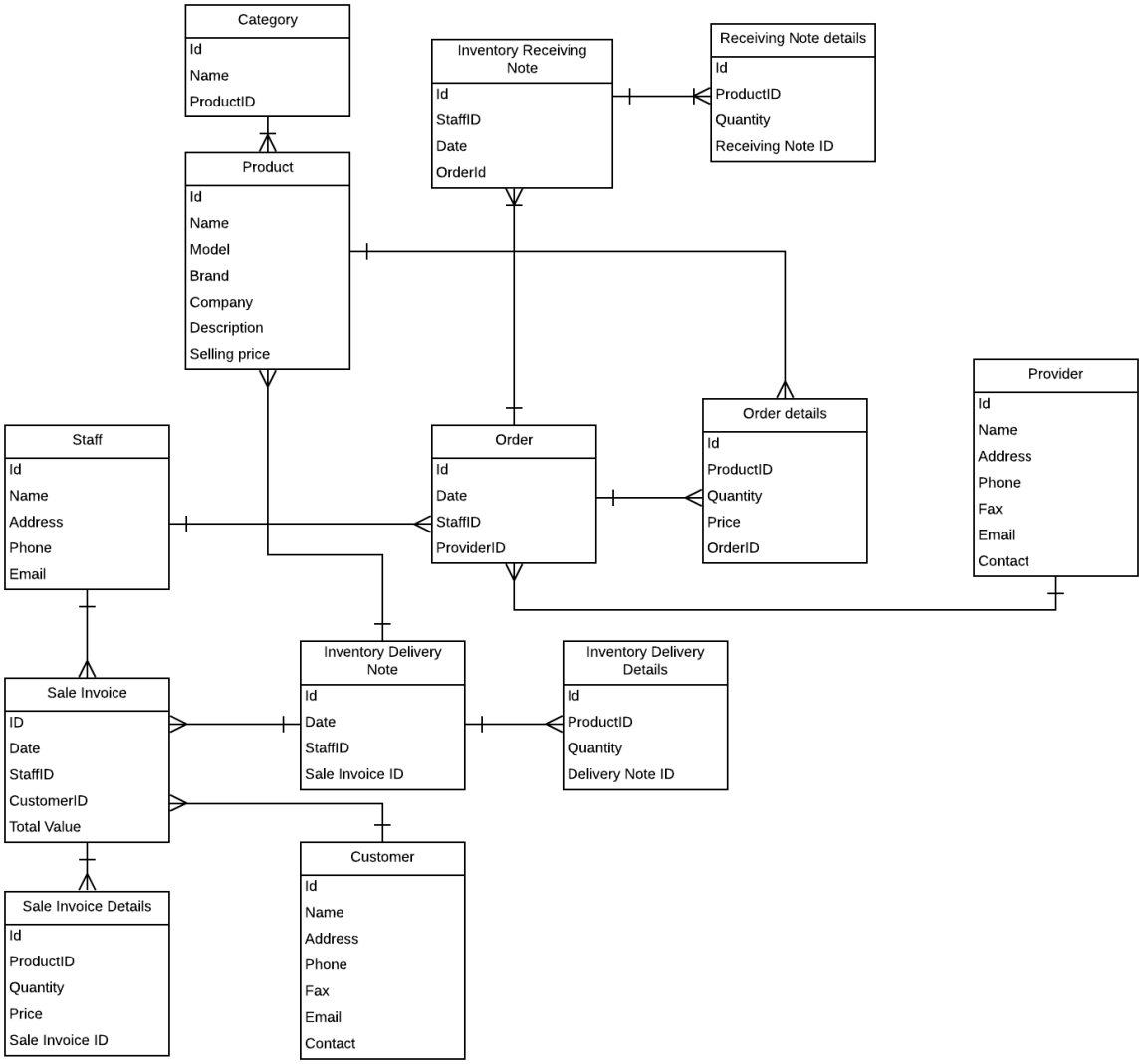
Architecture:

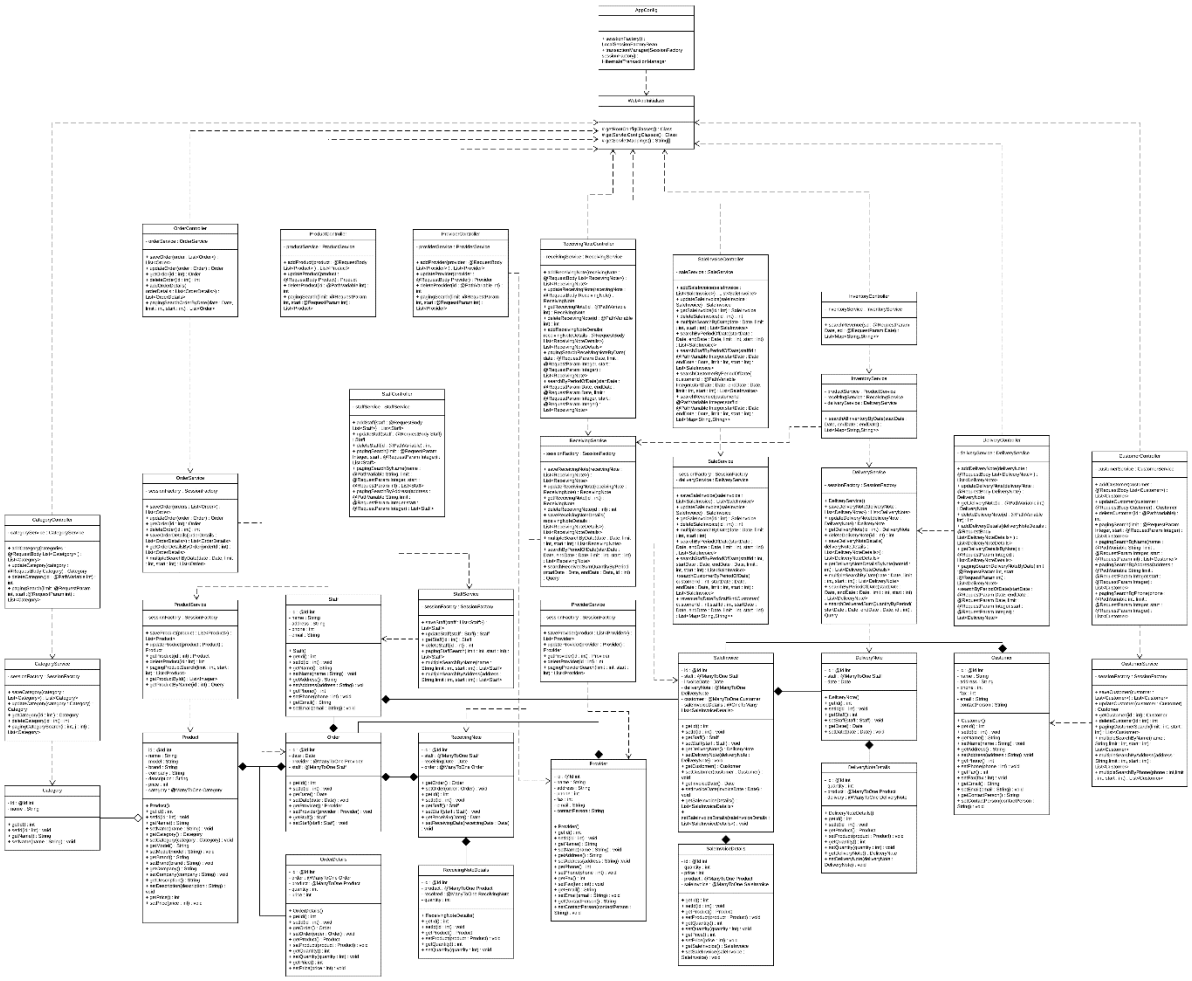
By looking at the package diagram, we can see clearly the architecture of the system.



Class Diagram:

There are 35 classes in the system. This ER diagram will illustration the relationship between the model class, while the class diagram will demonstrate the relationship of the whole system.


Category
Id
Name
ProductID
Inventory Receiving Note
Id
StaffID
Date
OrderId
Receiving Note details
Id
ProductID
Quantity
Receiving Note ID
Product
Id
Name
Model
Brand
Company
Description
Selling price
Provider
Id
Name
Address
Phone
Fax
Email
Contact
Order details
Id
ProductID
Quantity
Price
OrderID
Staff
Id
Name
Address
Phone
Email
Order
Id
Date
StaffID
ProviderID
Inventory Delivery Note
Id
Date
StaffID
Sale Invoice ID
Inventory Delivery Details
Id
ProductID
Quantity
Delivery Note ID
Sale Invoice
ID
Date
StaffID
CustomerID
Total Value
Customer
Id
Name
Address
Phone
Fax
Email
Contact
Sale Invoice Details
Id
ProductID
Quantity
Price
Sale Invoice ID



Implementation result:

The implementation produces the expected results compared with the system specifications.

Testing:

Except for the Statistic section, all tests are handled with “assertEquals”. In the statistic section, the output is printed to the console. Any methods that are implemented in the controllers are tested with 100%. However, users should not run it in one run. Instead, users should run each method from the top to the bottom. (Choose one method from the top to test in one time first and then run the next one after).

The order to test is:

CategoryControllerTest.java

ProductControllerTest.java

ProviderControllerTest.java

StaffControllerTest.java

OrderControllerTest.java

DeliveryControllerTest.java

ReceivingControllerTest.java

SaleInvoiceControllerTest.java

InventoryControllerTest.java

Limitation, known bugs:

All the bugs are handled successfully.