



PARA Trading stock and sales management system (PharmaPluz)

Student Name : G K D K Gajanayeka

BIT Reg. Number : R092721

Index Number : 0927211


Supervisor : Suneth Jayamane

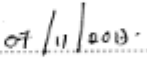
12/ 2017

**This dissertation is submitted in partial fulfilment of the requirement of the
Degree of Bachelor of Information Technology of the University of Colombo
School of Computing**


DECLARATION

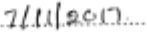
I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also here by give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.


.....
G.K.D.K. Gunayeka
(Name of Candidate)


.....
Date

Countersigned,


.....
Suneth Jayamanne
(Name of Supervisor)


.....
Date

ABSTRACT

PARA Trading's is a newly established and fast-growing company in the field of pharmaceutical. They mainly focused on distributing pharmaceutical, surgical, and diagnostic products in the Sri Lankan market and currently they are based on northwestern province. Company is in the growing stages and they are planning to expand their business and distribute their products all over the country.

Presently, PARA Trading's do not have a proper automated system to manage their process and it has been identified as a major issue for the sustainability of the company. It has been noticed that the company facing lot of difficulties such as delays in on time distribution, unmanaged inventories, unavailability of accurate information, stock maintenance, etc.

Therefore, owner has decided to address the matter with a fully functional IT system which will have a better automated centralized business system process complies with the new strategic planning of the company. The proposed system will be a great advantage to accomplish their requirements of the business and move forward quickly and reducing time wastage, over cost and wasting human resources. The developed system has been properly tested and assists to users and the management to work very efficiently.

The frontend of the system developed by using AngularJs, HTML5 and CSS3 order to provide an attractive, responsive and reusable system. Use Angular material UI component framework and reference implementation of Google material design to provide a set of well-tested and accessible UI component to system user.

Java spring boot framework used for backend development and it is the latest popular Java-based framework for building web and enterprise applications. Spring boot have powerful database transaction management capabilities and simplifies integration with other java frameworks like JPA/ Hibernate. System used PostgreSQL as the database management system, with an emphasis on extensibility and standards compliance.

Acknowledgement

First of all, I would like to offer my heartiest gratitude to Mr. Ranjith Bandara who is the owner of PARA Trading for giving me such a great opportunity to develop this stock and sales management system for his company and also my heartiest thank goes to rest of the employees at PARA Trading's for their support to success this project at the given time line.

My special thanks go to project supervisor Mr. Suneth Jayamane his support to drive the project towards the path of success by providing feedback constantly for the work done and providing the necessary guidance for the successful completion of the project.

Special thank goes to Mr. Ruwan Ponweera for his valuable ideas and guidance for the successful completion of the project.

Next I would like to give my thanks to all my friends who supported with their ideas and supporting whenever possible which is countless to write here.

Last but not least my heartfelt of gratitude goes to my wife, parents for their kind support and guidance given to me through the duration of the completion of the project and throughout my life time.

Table of Contents

ABSTRACT.....	iii
Acknowledgement	iv
Table of Contents	v
List of Figures	vii
List of Tables	viii
List of Acronyms	ix
CHAPTER 01: INTRODUCTION	1
1.1 Brief introduction of Para Trading.....	1
1.2 The Project.....	1
1.2.1 Objectives	1
1.2.2 Scope	2
1.2.3 Motivation for the project.....	2
CHAPTER 02: ANALYSIS.....	3
2.1 Introduction.....	3
2.2 Requirements Gathering	3
2.3 Existing similar system.....	4
2.4 Proposed system and process.....	5
2.5 Functional Requirements	8
2.6 Non-Functional Requirements	8
2.7 System Development Methodology for proposed system	9
CHAPTER 03: DESIGN OF SOLUTION.....	11
3.1 Alternative Solution.....	11
3.2 Solutions	11
3.3 Database Design	12
3.3.1 Database Normalization	12
3.4 User Interface design	13
CHAPTER 04: IMPLEMENTATION.....	17
4.1 Introduction.....	17
4.2 Implementation Environment	17
4.3 Development Tools.....	18
4.4 Codes of Main Module	21
CHAPTER 05: EVALUATION	27
5.1 Introduction.....	27
5.2 Test Plan	28

5.3 Implemented System Test Cases	28
CHAPTER 06 : CONCLUTION.....	32
6.1 Introduction.....	32
6.2 Future Improvements	32
6.2.1 PO process automation	32
6.2.2 Android App.....	32
6.2.3 Printing of reports via Android App.....	33
6.3 Lessons Learnt	33
REFERENCES	34
Appendix A - System Documentation	36
Appendix B - Design Documentation.....	44
Appendix C - User Documentation.....	50
Appendix D - Management Reports	55
Appendix E - Test Result	57
Appendix F – Code Listening	61
Appendix G - Client Certification.....	70
Index	71

List of Figures

Figure 2.1 Main Menu of the Insta Hospital Management System	4
Figure 2.2 PO of the Insta Hospital Management System	5
Figure 2.3 User Diagram.....	6
Figure 2.4 Agile Method	10
Figure 3.1 Class diagram for Proposed System	12
Figure 3.2 Database Diagram For Proposed System.....	13
Figure 3.3 User Login.....	14
Figure 3.4 User Creation.....	14
Figure 3.5 Register new Supplier	15
Figure 3.6 Add New Product	15
Figure 3.7 New Order.....	16
Figure 3.8 Dashboard	16
Figure 4.1 Netwaor Diagram.....	20
Figure 4.2 Login Module Frontend	21
Figure 4.3 Login Module Backend	23
Figure 4.4 Login Module User Details Service	24
Figure 4.5 Login Module Security Configuration.....	25
Figure 4.6 Database connection.....	26
Figure A.1 Creating an executable Jar file	36
Figure A.2 New Server register	37
Figure A.3 SQL Commands	37
Figure B.1 Activity Diagram Inventory	45
Figure B.2 Activity Diagram Order	46
Figure B.3 Activity Diagram Arrears Payment	47
Figure B.4 Sequence Diagram Order	48
Figure B.5 Sequence Diagram Order Report	49
Figure B.6 Sequence Diagram Customer Create	49
Figure B.7 Sequence Diagram Product Stock.....	50
Figure B.8 Sequence Diagram Payment.....	50
Figure C.1 Login Page.....	51
Figure C.2 Update User.....	52
Figure C.3 Add New Product.....	52
Figure C.4 Add New Product.....	53
Figure C.5 AllProduct.....	53
Figure C.6 All Invoice	54
Figure C.7 New Stock	54
Figure C.8 Order Invoice Details	54
Figure C.9 Order Invoice Details	55
Figure C.10 Order Invoice Details	55
Figure D.1 Customer Information Report	56
Figure D.2 Supplier Information Report	56
Figure D.3 Product Price Report	57
Figure D.4 Invoice Report	57
Figure F.1 Inventory Module	63
Figure F.2 Order Module	67
Figure F.3 All Product Module	70

List of Tables

<i>Table 2.1 Agile Method</i>	9
<i>Table 4.1 Hardware Requirment</i>	18
<i>Table 4.2 Software Requirment</i>	18
<i>Table 5.1 Verification and Validation</i>	27
<i>Table 5.2 User Authantication</i>	28
<i>Table 5.3 Register New User</i>	29
<i>Table 5.4 Update the System user</i>	29
<i>Table 5.5 Adding Product to the Stock</i>	30
<i>Table 5.6 Add Product Module</i>	30
<i>Table 5.7 New Order Place</i>	31
<i>Table 5.8 Adding Product to the Stock</i>	31
<i>Table E.1 User Authantication</i>	58
<i>Table E.2 Register New User</i>	58
<i>Table E.2 Update the System user</i>	59
<i>Table E.2 Adding Product to the Stock</i>	60
<i>Table E.2 Add Product Module</i>	60
<i>Table E.2 New Order Place</i>	60
<i>Table E.2 Adding Product to the Stock</i>	61
<i>Table E.2 Goods Returning Module</i>	61

List of Acronyms

BIT	-	Bachelor of Information Technology
CSS	-	Cascading Style Sheet
DB	-	Database
GHz	-	Gigahertz
HTML	-	Hypertext Mark- up Language
IDE	-	Integrated Development Language
IT	-	Information Technology
JPA	-	Java Persistence API
OOAD	-	Object Orient Analysis and Design
OS	-	Operation System
PO	-	Purchase Order
RAM	-	Random Access Memory
SAD	-	System Analysis Definition
SQL	-	Structured Query Language
UAT	-	User acceptance testing
UCSC	-	University of Colombo School of Computing
UI	-	User Interface
WWW	-	World Wide Web

CHAPTER 01: INTRODUCTION

1.1 Brief introduction of Para Trading

Para Trading is one of largest pharmaceutical items distributor in north western province. Mr. Ranjith Bandara, owner and the founder of PARA trading started his business couple of years ago and gradually established with the help of workers. His hard work and dedication helped him to become successful in the pharmaceutical industry.

Earlier the company had very few clients, but now they developed their client base widely. They are planning to expand the business while computerizing the existing business process. He manages his workers productively while providing satisfactory services to their customers according to their demands. Competence, efficiency and the way of the managing people helped him to run his business to fruitfully achieve the cooperate goal of the company. Therefore, PARA Trading is able to earn a prominent place in the industry.

1.2 The Project

1.2.1 Objectives

The Main objective of this proposed system is reducing over cost, reduce duplicating work, and reduce time wastage of the staff while creating an accurate and efficient centralized system to run the business.

Objectives of this project consist of

- Computerized, hassle free and centralized system for workers with undue effort for business processes in order to make more efficient working process.
- Reducing over cost of the wasting resources of the workers.
- Reduce the duplication work of the manual work process
- Making quick and accurate decisions easily to the management through the reports.

1.2.2 Scope

Following tasks to be implemented in the propose system, in order to accomplish the main objectives of the project.

- Sales Handling
- Order Entry and Order Processing
- Inventory Handling
- Stock Maintenance & Control
- Inquiry Handling
- Providing required operational & Management reports
- PO automation process

1.2.3 Motivation for the project

PARA Trading is a newly established pharmaceutical company which involved in distributing pharmaceutical, surgical, and diagnostic products in the Sri Lankan market and currently they are based on northwestern province. Recently, owner of the company is planning to expand their business and distribute their products all over the country. The owner has lots of concerns about the manual process and facing considerable difficulties to manage the manual work. Expanding the business affected on managing human resources, time management and duplication of work. As a solution he decided to implement suitable and centralized system for his business.

The proposed system will be a great subsidy which leads to full fill his necessities of the business and move forward speedily by reducing time wastage, over cost and wasting human resources. Also, system enables him to easily manage Sales, inquiry handling, inventory handling, PO processing, stock maintain and generate required reports with accurate information.

I believe that the knowledge I gained throughout my BIT degree program enables me to practically apply in a realistic environment and give alternate solutions to the customer successfully.

CHAPTER 02: ANALYSIS

2.1 Introduction

Systems Analysis Definition - (SAD) The analysis of the role of a proposed system and the identification of the requirements that it should meet. SAD is the starting point for system design. The term is most commonly used in the context of commercial programming, where software developers are often classed as either systems analysts or programmers. The systems analysts are responsible for identifying requirements (i.e. systems analysis) and producing a design. The programmers are then responsible for implementing it. [1]

2.2 Requirements Gathering

Before implementing the system, a software engineer must gather all the data and information related to the system. It is necessary to well understand core functions and critical functions of the system in order to accomplish the overall objective. Gathering client requirements clearly and accurately helps to implement a good system unless otherwise system may be failed. In the system it used common fact-finding methods and gathered the system requirements for proposed system as follows,

- **Interview**

Conducting face-to-face interviews is a basic requirement gathering method which is more effective. Owner and some of the staff members have been interviewed by the author individually and gathered relevant information about the existing process.

- **Observation**

Having observed the existing process allows to clearly understand the requirements of the client. Author physically visited and observed the staff members to get an overview of the process, issues and the difficulties they faced when using current methods.

- **Investigating the manual process**

Currently they are using a manual process and gathered all the data related to the system while getting an idea of what they need and what are the expectations of the new system.

- **Scrutinizing similar systems**

Browsing internet and view stock and sales management systems to understand how practically applies the stock controlling mechanism to a realistic system.

2.3 Existing similar system

I have studied the Stock and Inventory Control Module of Insta Hospital Management System to obtain a good knowledge on Stock Inventory Management and its behaviors.

Stock and Inventory Control Module consist of

- Purchase Order
- Stock Entry (GRN)
- Process of Stock Ordering for Sub Store
- Stock Ledger Transactions
- Stock Reports



Figure 2.1 Main Menu of the Insta Hospital Management System

Raise Purchase Order

PO Details

* Store: MAIN STORE Department: --Select-- Quantity UOM: Package UOM

* Supplier: BC Pharma Service (Pvt) Ltd Supplier Address: Ph: 0115337648 PO Allotted To: --Select--

* PO Date: 15-02-2018 Enquiry No: Enquiry Date:

Quotation No: Credit Period: Expected Delivery Date: C-Form: Tax Type:

Item Details

Item: ZINCOPLUS SYRUP

MRP: MUPIRAX 5G

Qty: RESPIRATORY CLINIC - FORM

Discount (%): SURGICEL 2*3 W1913

Manufacturer: STAMLO 5MG

Package UOM: LIBRIUM 10MG PER TAB

Stock: ATENOLOL 25MG

Tax Basis: PYBACTUM 4.5G INJECTION

Item Status: METHIBION TABLET

Remarks: OSELTAMIVIR 75MG

MEBERVER 200

Buttons: Add, Cancel, << Previous, Next >>, New Item

Totals

Item Discounts: PO Discount Type: Round Off:

Supplier Terms And Conditions:

Disc Tax Amt Total

Total Tax: 0.00

PO Discount Amt: 0.00

Figure 2.2 PO of the Insta Hospital Management System

2.4 Proposed system and process

There are different types of users in the proposed system and they are playing various roles.

Types of the users are as follows,

- **Admin Clerk**

Responsible for keeping general operations running efficiently.

- **Stores Clerk**

Facilities Management performs manual and moderately complex clerical work of a general warehouse or stockroom nature including receiving, inventorying, storing, accounting for, and issuing supplies

- **Manager**

Information systems planning, installation, and maintenance, including hardware and software upgrades.

- **Sales person**

Bridge the gap between customer needs and the product/service that fulfils that need, which builds credibility and therefore trust.

- **Admin**

Analyzing system logs and identifying potential issues on systems. Applying system updates, patches, and configuration changes. Including installing and configuring new hardware and software.

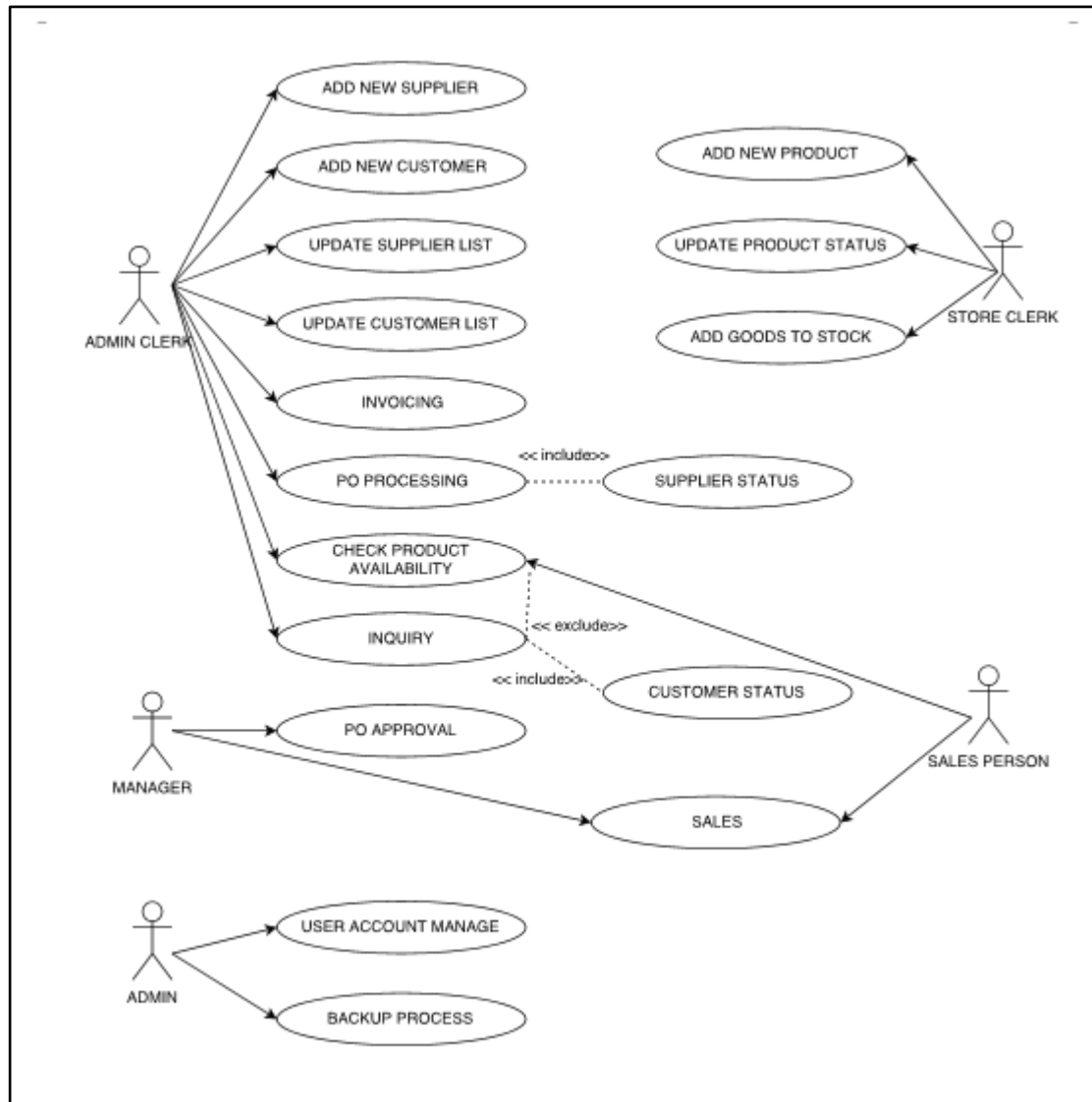


Figure 2.3 User Diagram

1. Add New Supplier

First of all, it is required to create a supplier profile before starting the buying process. After that only system allows to proceed with the rest of the process.

2. Add New Customer

Selling goods need to create a customer profile.

3. Invoicing

Invoice is a financial document, or a bill of sale issued to the buyer by the seller. Invoice will be issued after the transaction and it contains the details of trading parties, description, quantities of the items sold, date of the transaction, prices and discounts and payment terms

4. Purchasing Order Processing

Purchase order will be placed by the seller to a particular supplier or a customer to the seller before the transaction. Processing of a purchasing order is a very important activity in the system.

5. Check Product Status

Checking status of the items or the products is an essential activity in a stock and sales management system. All the processes related to the system such as, inquiries, purchasing depends on this option.

6. Inquiry

Inquiries received from the customer it will be stored in the system with details. Inquiry in the sense, it enables customer to view the details of the product to be purchased or else check the availability of the item whether they are in stock or out of stock.

7. Customer Status

When a customer made an inquiry about a product it is required to check the status of the customer to process further. As an example, if a customer status shows as default user might not further process a delivery order for the customer or seek for approval.

8. Supplier Status

Whenever seller wants to check the availability of goods in the supplier, system enables to check that through supplier status.

9. PO Approval

When a purchasing order placed, every single order must be verified and approved by the authorized person or the admin. Upon verification authority will be granted to the supplier.

10. User Account Manage

System administrator only will be maintained User Logins and User profiles for system security purposes and permission will be granted for the relevant parties to perform assign tasks.

11. Generate Backups

In order to make sure the system recovery at any time the system administrator will be able to generate periodical backups of the database system.

12. Add New Product

This process will be used to add new products to the product profile. When supplier introduced a new product, it will be added to the profile.

13. Add Goods to Stock

After the buying products from supplier, need to be added the stock

2.5 Functional Requirements

Following are the functional requirements of the proposed system

- System should facilitate for customer inquiry handling through searching
- System should facilitate for checking customer status
- System should provide available stocks in hand at any time.
- System should facilitate for invoicing with parallel process of delivery order to the client
- System should facilitate handling buying goods
- System should facilitate handling PO and update them through Good received note.
- System should provide Maintaining Bills of the supplying goods separately.
- System should provide Stock, sales and other reports in given time periods.

2.6 Non-Functional Requirements

Below are the Non-Functional requirements of the proposed system.

- System should be user friendly.
- User interfaces must be easy to work with.

- Should be more efficient compared to the current manual process.
- Should not waste computer resources as there is an account application package and normal office packages use together in same machines.
- In case of disaster, hardware failure or software crash to recover quickly, there should be a well-defined backup process must be implemented.
- Security mechanism should implement with new system

2.7 System Development Methodology for proposed system

AGILE MODEL

Agile model focuses on process adaptableness and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iteration. Iteration is the routine of developing small sections of a project at a time. Each iteration is reviewed and assessed by the development team and client. The comprehensions gained from the assessment are used to determine the next step in development. Agile model delivers increased value, visibility, and compliance much earlier in the life cycle, considerably reducing project risk. This model reflects an attempt to increase the flexibility of the development process by allowing the client to interact and experiment with a working representation of the product.

Principle	Description
Customer involvement	Customers should be closely involved throughout the development process. Their role is to provide and prioritize new system requirements, and to evaluate the Iterations of the system.
Incremental delivery	The software is developed in increments with the customer specifying the requirements to be included in each increment.
People not process	The skills of the development team should be recognized and exploited. Team members should be left to develop their own ways of working without prescriptive processes.
Embrace change	Expect the system requirements to change and so design the system to accommodate these changes
Maintain simplicity	Focus on simplicity in both the software being developed and in the development process. Wherever possible, actively work to eliminate complexity from the system.

Table 2.1 Agile Method [2]

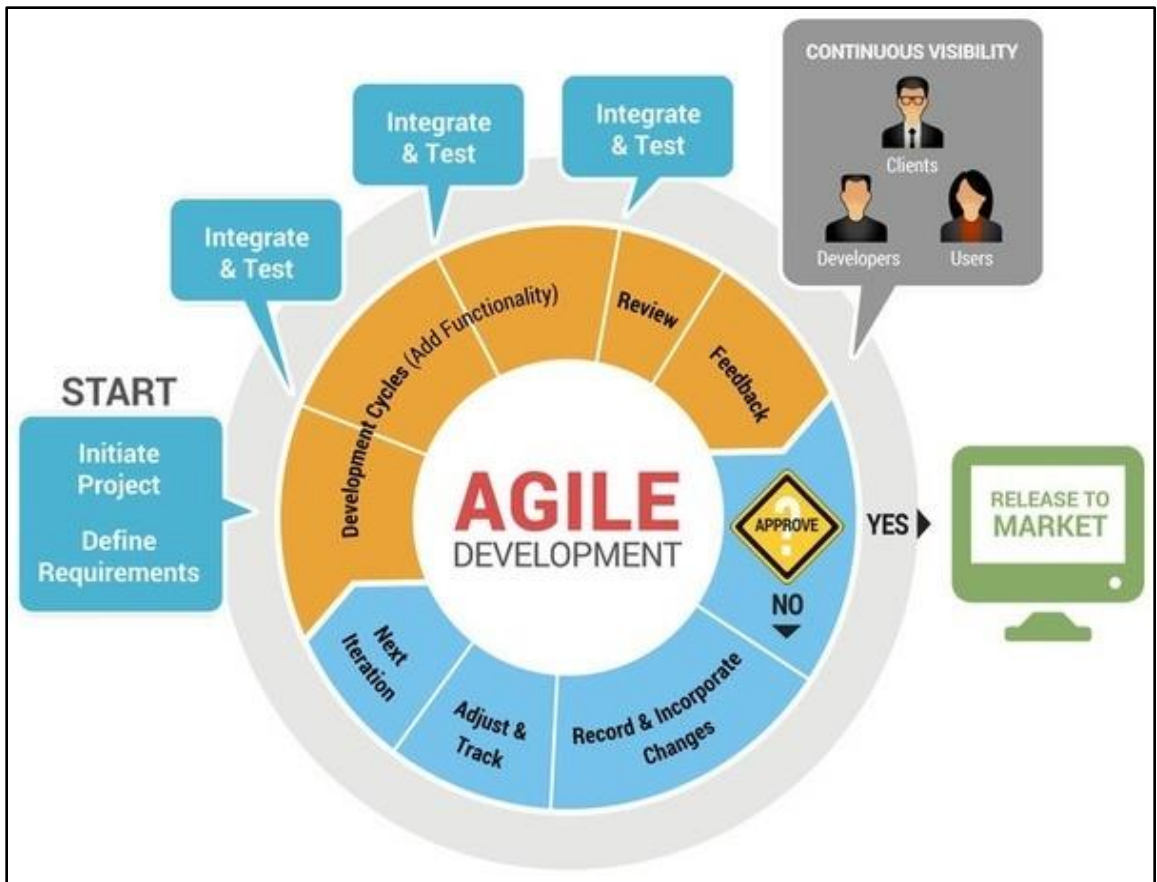


Figure 2.4 Agile model [3]

CHAPTER 03: DESIGN OF SOLUTION

System design is the process of defining the components, modules, interfaces, and data for a system to satisfy specified requirements. System development is the process of creating or altering systems, along with the processes, practices, models, and methodologies used to develop them. [4]

3.1 Alternative Solution

Standalone system as an alternative solution would not be the matched client's requirement. Unable to access from outside, expensive, maintain not easy and run on single computer are main disadvantages of the standalone system.

3.2 Solutions

Web based System proposed to the client as a complementary solution. Proposed system will be implemented based on the client requirements and client will be benefited with required solutions. System enables following benefits to the client, cost effective development, accessible anytime, anywhere, easier to customize accessible for a range of devices, easily installation and maintenance, increased security are the main advantages of the system.

OBJECT-ORIENTED ANALYSIS & DESIGN

Object-oriented analysis and design (OOAD) is a technical approach used in the analysis and design of an application or system through the application of the object-oriented paradigm and concepts including visual modeling. This is applied throughout the development life cycle of the application or system, fostering better product quality and even encouraging stakeholder participation and communication. [5]

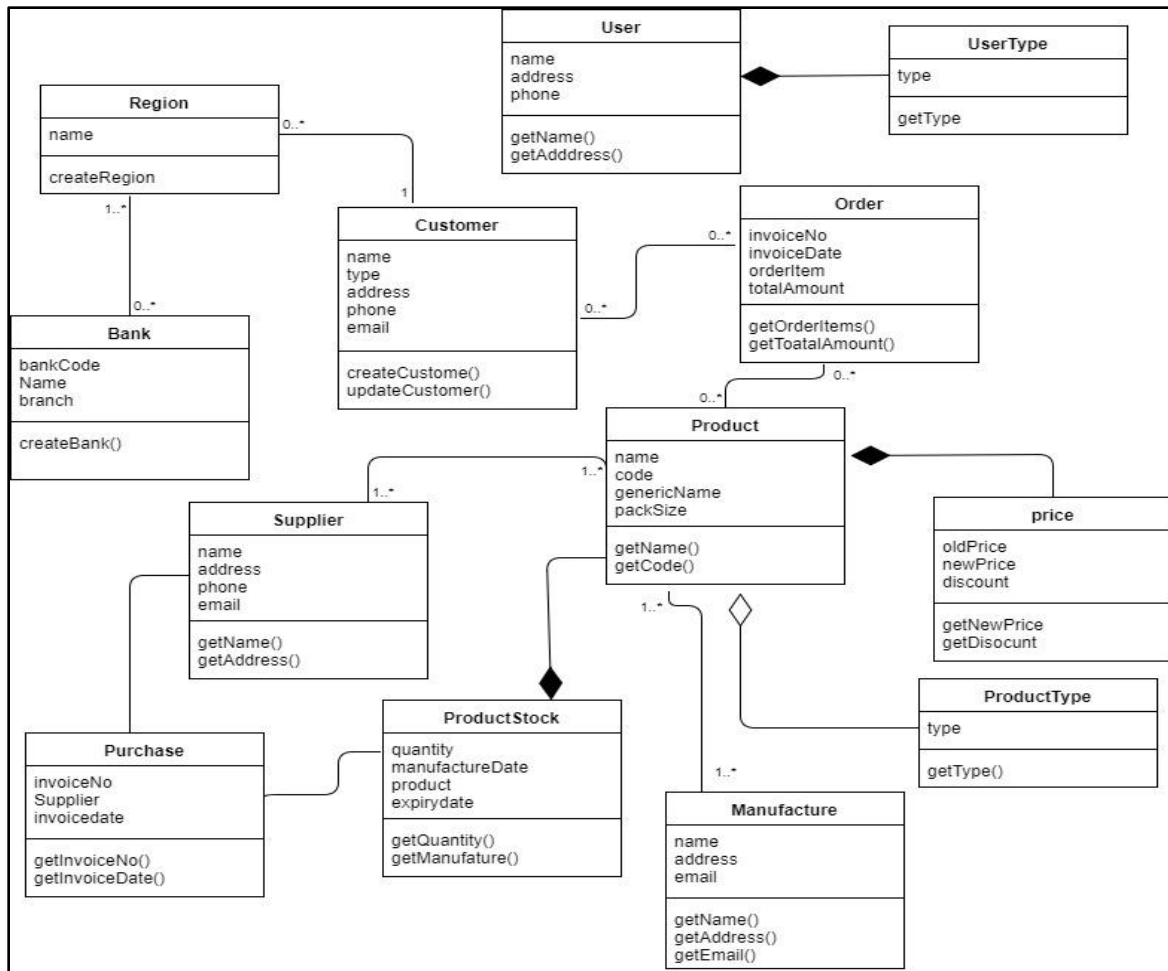


Figure 3.1 Class diagram for Proposed System

3.3 Database Design

A good database design starts with a list of the data which is needed to include in the database and what you want to be able to do with the database later on. In order to enhance the performance while increasing the efficiency of the database it is necessary to follow the database normalization standard processes as explained below.

3.3.1 Database Normalization

Database normalization is the process of organizing data into tables in such a way that the results of using the database are always unambiguous and as intended. Such normalization is intrinsic to relational database theory. It may have the effect of duplicating data within the database and often results in the creation of additional tables.

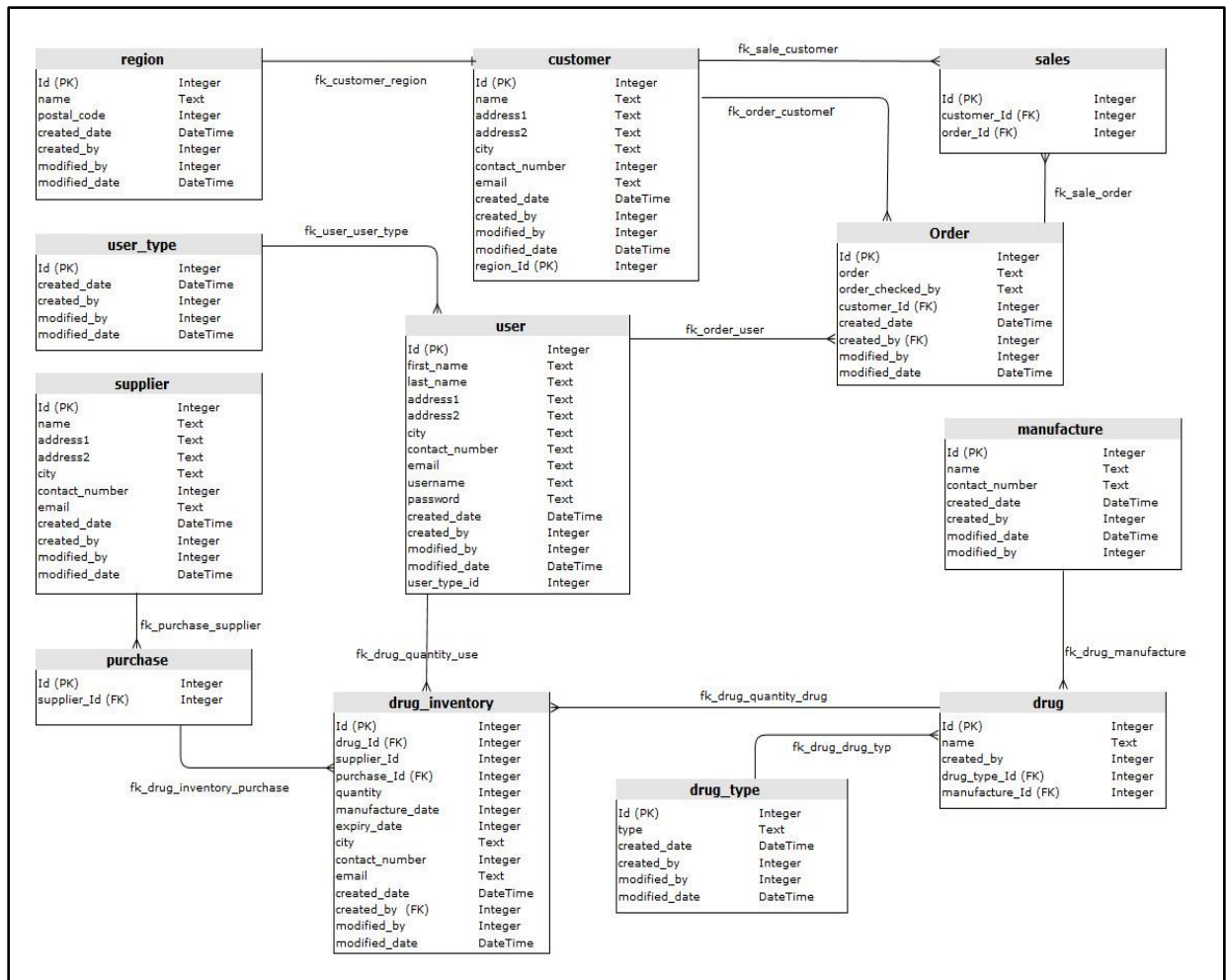


Figure 3.2 Database diagram for Proposed System

3.4 User Interface design

User Interface (UI) Design focuses on expecting what users might need to do and ensuring that the interface has elements that are easy to approach, understand, and use to enable those actions. Familiar user interface will help with task completion, efficiency, and satisfaction. Therefore, user involvement will be more important to implement the interfaces and also final version of the system will be developed with more attractive way to users. UI need to be useful, as user's desires satisfied by the interface functionality.

A good UI reflects how easy is it for the user to fulfill basic tasks when using the system for the first time also after the user knows the interface, how fast is he able to accomplish the given tasks. Reliability and user friendliness are also important features of a productive UI.

Below show some screen shots of the system.

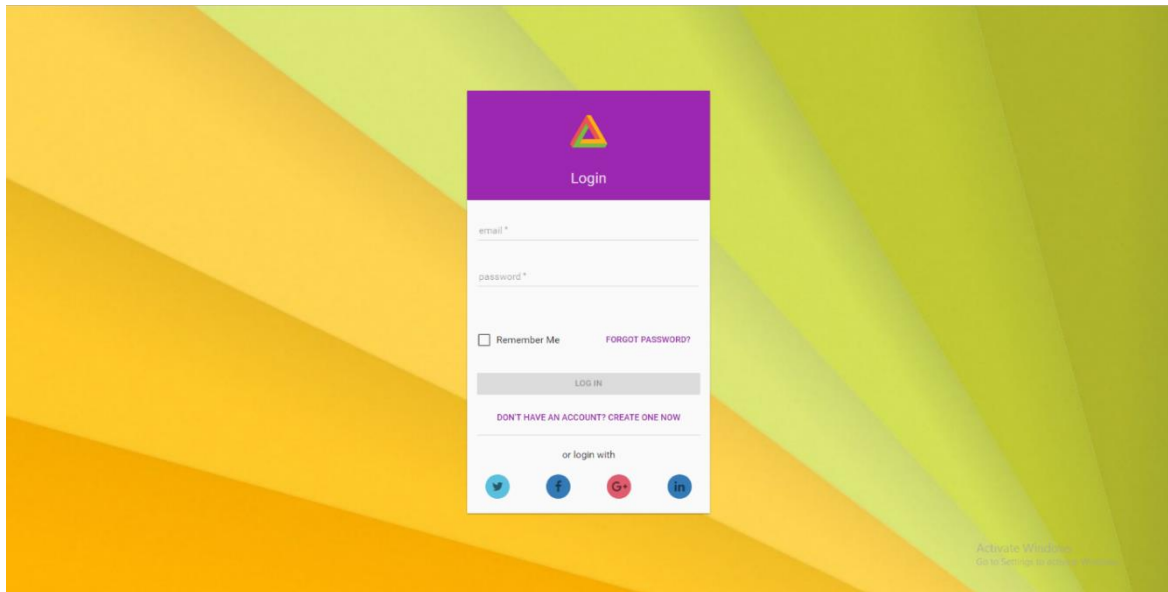


Figure 3.3 User Login

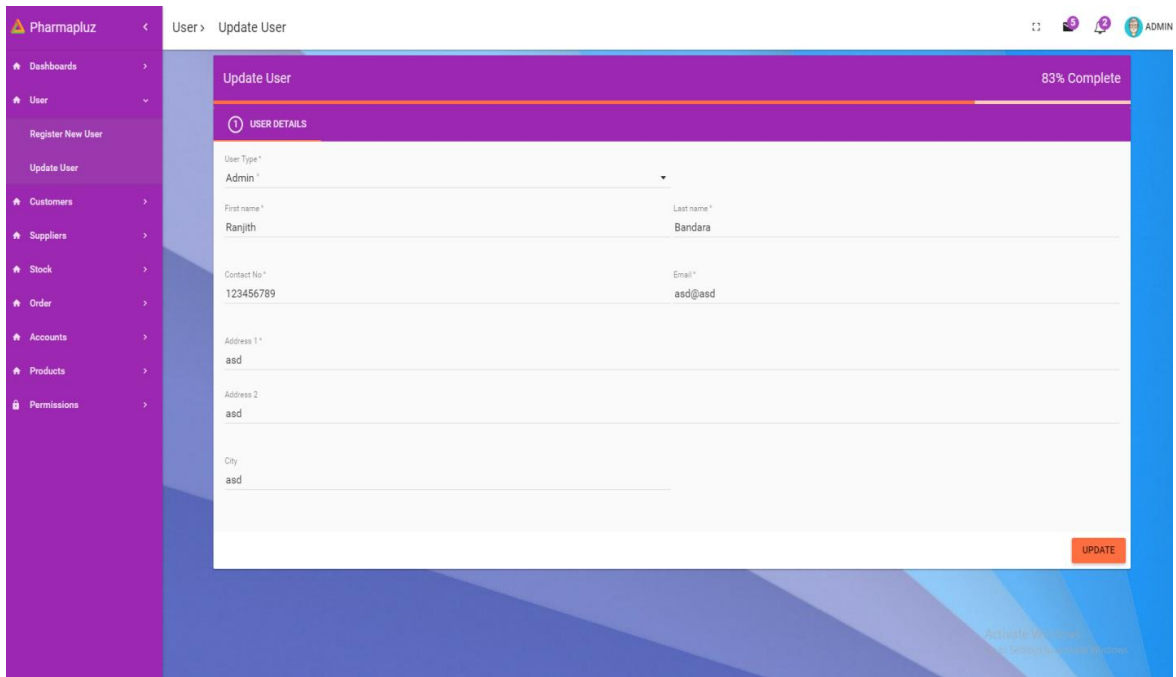


Figure 3.4 User Creation

Pharmapluz < Suppliers > Register New Suppliers

Register New Supplier 0% Complete

1 SUPPLIER DETAILS

Company Name *

Contact No * Email *

Address 1 *

Address 2

City

SEND

Figure 3.5 Register new Supplier

Pharmapluz < Delivery Order > New Order

Order Invoice Details

Order Items

Total Amount: 0

NEW INVOICE

COMPLETE ORDER CANCEL

Select Product

Stock No	Date	Qty	Free Qty	Dis%	Unit Price
00001	2018-04-01	0	0	100	1500
P0051	2018-01-31	468	468		20 ✓
0956	2018-03-01	25	25		5 ✓
123	2018-02-01	0	0		125

Figure 3.6 Add new Order

Product	Product Type	Qty	Manufacture	Supplier	Exp D
asd pandol					2017-
test1					2017-

Add Invoice Details

Invoice No *

Supplier *

Invoice Date

Total Amount *

ADD

CLEAR

Figure 3.7 Add an Invoice

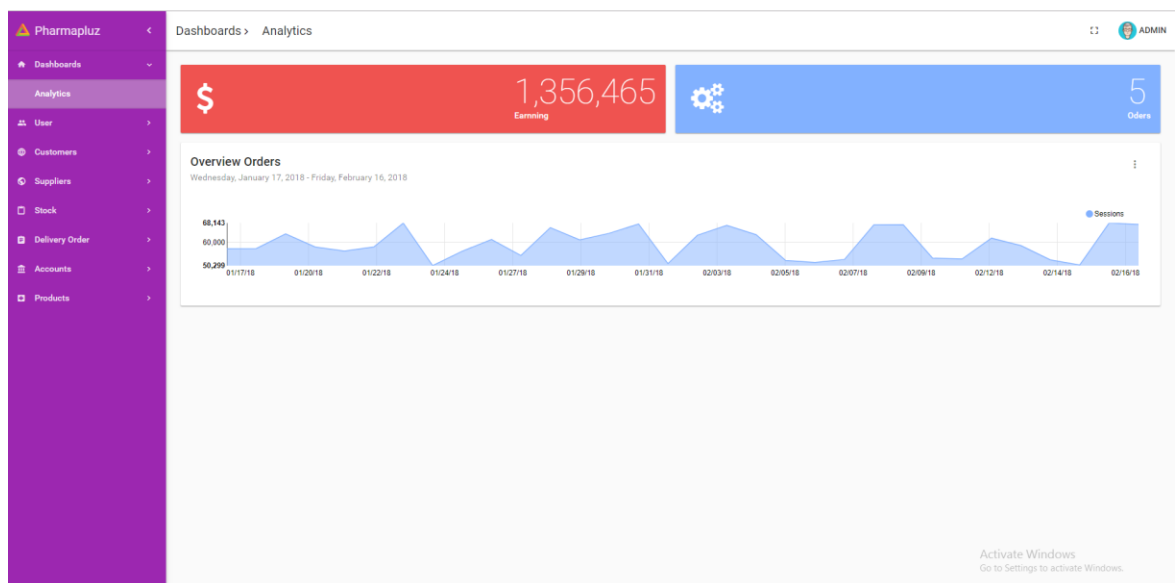


Figure 3.8 Dashboard

CHAPTER 04: IMPLEMENTATION

4.1 Introduction

Overall activities carried out while developing PARA Trading stock and sales management system and the outputs of the system will be illuminated in the Implementation chapter. As developed system turns in to a working system, implementation becomes the most important stage of the system. In order to facilitate the requirements as per the design phase, system was implemented with the support of modern system implementation tools. This chapter mainly focuses on what are the modern techniques used to develop, tools used to implement the system and matters related to the implementation stage.

4.2 Implementation Environment

The frontend of the system developed by using AngularJs, HTML5 and CSS3 I order to provide an attractive, responsive and reusable system. Use Angular material UI component framework and reference implementation of Google material Design to provide a set of well-tested and accessible UI component to system user.

Java spring boot framework used for backend development and it is the latest popular Java-based framework for building web and enterprise applications. Spring boot have powerful database transaction management capabilities and simplifies integration with other java frameworks like JPA/ Hibernate. System used PostgreSQL as the database management system, with an emphasis on extensibility and standards compliance.

.

Hardware requirements,

Server	Client
CPU: Intel ® Core™ I5 4460M @ 3.20GHz RAM: 8GB HDD: 500GB	CPU: Intel ® Core™ I3 3210M @ 2.50GHz RAM: 4GB HDD: 120GB

Table 4.1 Hardware Requirement

Software requirements,

Server	Client
Microsoft Server 2008 or Later version OS AngularJs PostgreSQL Apache Maven Spring Boot	Microsoft Windows 7 or later OS Google Chrome web Browser Android mobile phone

Table 4.2 Software Requirement

4.3 Development Tools

Angular

AngularJS is a JavaScript-based open-source front-end web application framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. [6]

PostgreSQL

PostgreSQL is a powerful, open source object-relational database system. It has more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness. It runs on all major operating systems, including Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, macOS, Solaris, Tru64),

and Windows. It is fully ACID compliant, has full support for foreign keys, joins, views, triggers, and stored procedures (in multiple languages). It includes most SQL:2008 data types, including INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, and TIMESTAMP. It also supports storage of binary large objects, including pictures, sounds, or video. It has native programming interfaces for C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, among others, and exceptional documentation. [7]

Apache Maven

Apache Maven is a build automation tool used primarily for Java projects. In Yiddish, the word maven means "accumulator of knowledge". Maven addresses two aspects of building software: first, it describes how software is built, and second, it describes its dependencies. [8]

Spring Boot

Spring Boot offers a fast way to build applications. It looks at your class path and at beans you have configured, makes reasonable assumptions about what you're missing, and adds it. With Spring Boot, you can focus more on business features and less on infrastructure. [9]

JetBrains IntelliJ IDEA

Every aspect of IntelliJ IDEA is specifically designed to maximize developer productivity. Together, powerful static code analysis and ergonomic design make development not only productive but also an enjoyable experience. [10]

Implementation Method

In the proposed system implemented as a client -server architecture. Client and server computers communicate with each other over a network. In the client-server computing, a server takes requests from client and shares its resources, applications and data with one or more client computers on the network, and a client is a computing device that initiates contact with a server in order to make use of a shareable resource.

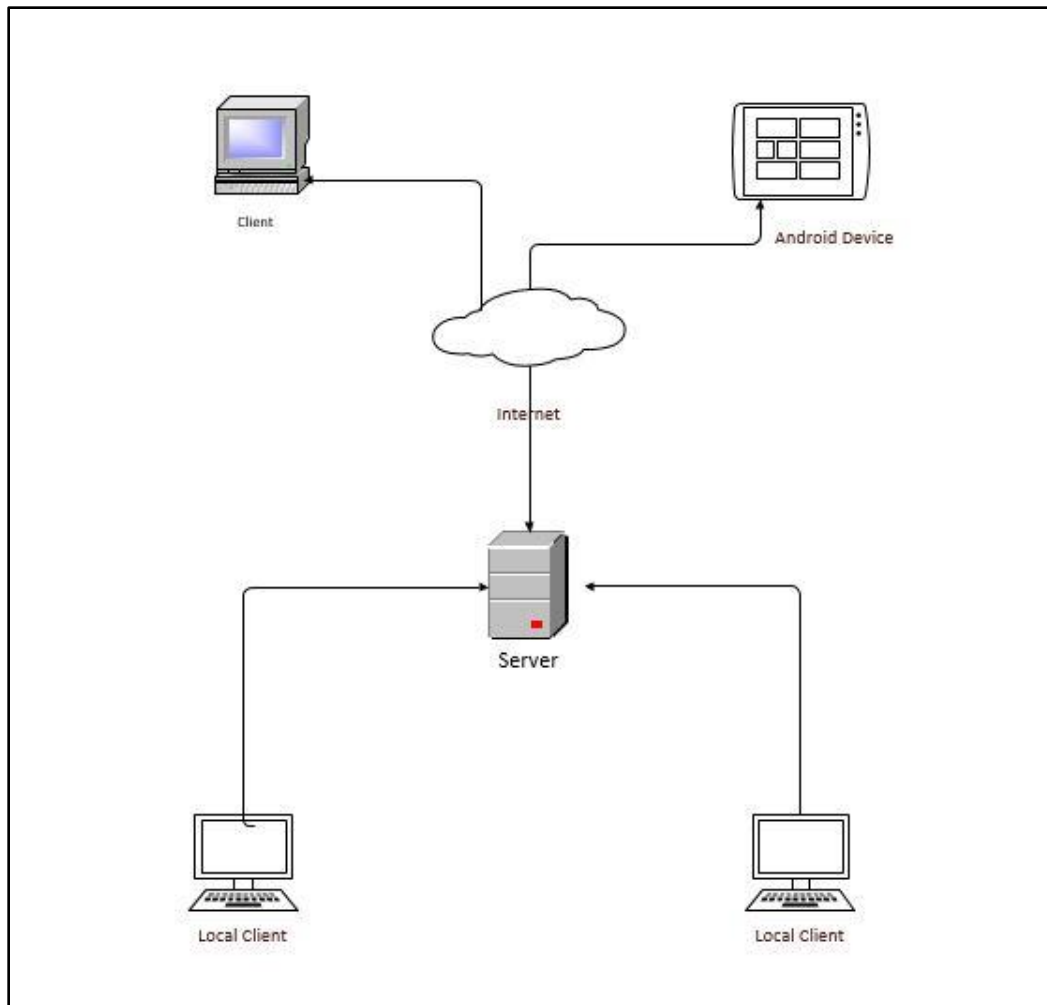


Figure 4.1 Network Diagram

4.4 Codes of Main Module

Login Module – Front end

```
@(function() {
  'use strict';

  angular
    .module('app.pharmapluz.authentication')
    .controller('LoginController', LoginController);

  LoginController.$inject = ['$state', 'triSettings', 'TestService', '$http'];

  function LoginController($state, triSettings, TestService, $http) {
    var vm = this;
    var test = new TestService.Test();
    vm.login = login;
    function login() {
      var data = btoa(vm.user.email + ':' + vm.user.password);
      $http.defaults.headers.common['Authorization'] = 'Basic ' + data;
      $http.get('http://localhost:8080/authenticate').success(function (response) {
        console.log(response);
        $state.go('triangular.dashboard-analytics');
      });
    }
  }
})();
```

Figure 4.2 Login Module - Frontend

Login Module – Backend

```
public class CustomUserDetails implements UserDetails {
    private static final long serialVersionUID = 1L;
    private final Integer id;
    private final String password;
    private final String username;
    private final String userFirstName;
    private final List<GrantedAuthority> authorities;
    private final boolean accountNonExpired;
    private final boolean accountNonLocked;
    private final boolean credentialsNonExpired;
    private final boolean enabled;

    public CustomUserDetails(Integer id, String username, String password, String
userFirstName,
        List<GrantedAuthority> authorities, boolean accountNonExpired, boolean
accountNonLocked,
        boolean credentialsNonExpired, boolean enabled) {
        this.id = id;
        this.username = username;
        this.password = password;
        this.userFirstName = userFirstName;
        this.authorities = authorities;
        this.accountNonExpired = accountNonExpired;
        this.accountNonLocked = accountNonLocked;
        this.credentialsNonExpired = credentialsNonExpired;
        this.enabled = enabled;
    }
}
```

```

public Integer getId() {
    return id;
}
@Override
public Collection<? extends GrantedAuthority> getAuthorities() {
    return authorities;
}
@Override
public String getPassword() {
    return password;
}
@Override
public String getUsername() {
    return username;
}
public String getUserFirstName() {
    return userFirstName;
}
@Override
public boolean isAccountNonExpired() {
    return accountNonExpired;
}
@Override
public boolean isAccountNonLocked() {
    return accountNonLocked;
}
@Override
public boolean isCredentialsNonExpired() {
    return credentialsNonExpired;
}

@Override
public boolean isEnabled() {
    return enabled;
}
public boolean isUserInRole(String authority) {
    return authorities.contains(new SimpleGrantedAuthority(authority));
}
@Override
public String toString() {
    return "CustomUserDetails{" + id + ',' + username + ',' + authorities + '}';
}
}

```

Figure 4.3 Login Module - Backend

User Login – User Details Service

```
Component("userDetailsService")
public class UserDetailsService implements
org.springframework.security.core.userdetails.UserDetailsService {
    @Autowired
    UserRepository userRepository;
    @Autowired
    UserTypeRepository userTypeRepository;
    @Override
    @Transactional
    public UserDetails loadUserByUsername(final String username) throws
UsernameNotFoundException{
        Optional<User> usersFromDB = userRepository.findOneByUsername(username,
userTypeRepository.findAllUserTypes());
        return usersFromDB.map(user -> {
            List<UserType> userTypes = new ArrayList<UserType>();
            userTypes.add(user.getUserType());
            List<GrantedAuthority> grantedAuthorities = userTypes.stream()
                .map(authority->new SimpleGrantedAuthority(authority.getType()))
                .collect(Collectors.toList());
            return new
CustomUserDetails(user.getId(),username,user.getPassword(),user.getFirstName(),grantedAuthoritie
s,true,true,true,true);
        }).orElseThrow(() -> new UsernameNotFoundException("User "+ username + " was not found
in the Database"));
    }
}
```

Figure 4.4 Login Module - User Detail Service

User Login – Security Configuration

```
public class SecurityConfiguration extends WebSecurityConfigurerAdapter{
    @Autowired
    UserDetailsService userDetailsService;
    @Bean
    public PasswordEncoder passwordEncoder(){
        return new BCryptPasswordEncoder();
    }
    @Autowired
    protected void configure(AuthenticationManagerBuilder auth) throws Exception{
        auth
            .userDetailsService(userDetailsService)
            .passwordEncoder(passwordEncoder());
    }
    @Override
    protected void configure(HttpSecurity httpSecurity) throws Exception{
        httpSecurity.httpBasic()
            .and()
            .authorizeRequests()
            .antMatchers("/*")
            .permitAll().anyRequest().authenticated()
            .and()
            .csrf().disable();
        httpSecurity.cors().configurationSource(new CorsConfigurationSource() {
            @Override
            public CorsConfiguration getCorsConfiguration(HttpServletRequest request) {
                return new CorsConfiguration().applyPermitDefaultValues();
            }
        });
    }
}
```

Figure 4.5 Login Module - Security Configuration

Database Connection

```
spring.datasource.url=jdbc:postgresql://localhost:5432/postgres?autoReconnect=true
spring.datasource.username=postgres
spring.datasource.password=1234
spring.datasource.driver-class-name=org.postgresql.Driver
error.path=/error
spring.datasource.maxActive=15
spring.datasource.maxIdle=8
spring.datasource.initial-size=10
server.session-timeout=6004
spring.jpa.generate-ddl=true
```

Figure 4.6 Database connection

CHAPTER 05: EVALUATION

5.1 Introduction

Main objective of the developed software solution should be efficient, productive and good quality system delivered to the client. Testing plays the most important role as it helps to check whether the final system meets the end user requirements and compile necessary changes if required. In order to accomplish this phase fruitfully entire system will be tested using Verification and Validation methods.

Criteria	Verification	Validation
Definition	The process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase.	The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements.
Objectives	To ensure that the product is being built according to the requirements and design specifications. In other words, to ensure that work products meet their specified requirements.	To ensure that the product actually meets the user's needs, and that the specifications were correct in the first place. In other words, to demonstrate that the product fulfills its intended use when placed in its intended environment.
Question	Are we building the product right?	Are we building the right product?
Evaluation Items	Plans, Requirement Specs, Design Specs, Code, Test Cases	The actual product/software.
Activities	<ul style="list-style-type: none">• Reviews• Walkthroughs• Inspections	<ul style="list-style-type: none">• Testing

Table 5.1 Verification and Validation [11]

5.2 Test Plan

A document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process. [12]

5.3 Implemented System Test Cases

Proposed system tested with incorrect data and check the expected output are occurred.

Below mentioned some test cases of the main modules tested.

User Authentication

No.	Scenario	Output Expected	Status
1	User Login with correct user name and correct password	Login to main menu	Pass
2	User Login with correct user name and wrong password	Not allow login to main menu	Pass
3	User Login with wrong user name and correct password	Not allow login to main menu	Pass
4	User Login with wrong user name and wrong password	Not allow login to main menu	Pass

Table 5.2 User Authentication

Register new system user

No.	Scenario	Output Expected	Status
1	Without enter the data to mandatory (*) fields	Not allowed to Save the record. Button not enable to save	Pass
2	Enter the data to mandatory (*) fields correctly	Allowed to Save the record. Button enable to save	Pass
3	Enter a contact number more than 10 digit	Showing under the input field “Invalid Number”	Pass
4	Enter a contact number less than 10 digit	Showing under the input field “Invalid Number”	Pass
5	Enter an email address without proper email address	Showing under the input field “Invalid email format”	Pass
6	Enter an email address correctly	Allowed to Save the record. Enable the save button.	Pass

Table 5.3 Register New User

Update the System User

No.	Scenario	Output Expected	Status
1	Erase the data on mandatory (*) fields	Not allowed to update the record. Button not enable to update.	Pass
2	Update the data to mandatory (*) fields correctly	Allowed to update the record. Button enable to update.	Pass
3	Update a contact number more than 10 digit	Showing under the input field “Invalid Number”	Pass
4	Update a contact number less than 10 digit	Showing under the input field “Invalid Number”	Pass
5	Update an email address without proper email address	Showing under the input field “Invalid email format”	Pass
6	Update an email address correctly	Allowed to update the record. Button enable to update.	Pass
7	User Account Delete	Pop up Alerting and ask, “Do you want to remove this user permanently”	Pass

Table 5.4 Update the System User

Adding Product to the stock

No.	Scenario	Output Expected	Status
1	Without enter the data to mandatory (*) fields	Not allowed to add the record. Button not enable to update.	Pass
2	Enter the data to mandatory (*) fields correctly	Allowed to Save the record. Button enable to save	Pass
3	Enter the letter to total amount	Showing under the input field “Invalid Amount”	Pass
4	Enter the numbers to total amount	Allowed to add the record. Button enable to add.	Pass
5	Expire date older than or equal manufacture date	Showing under the input field “Invalid Expired date”	Pass
6	Expire date equal to manufacture date	Showing under the input field “Invalid Expired date”	Pass
7	Enter the letter to Purchase price and sales price	Showing under the input field “Invalid Amount”	Pass
8	Delete a product from stock	Pop up Alerting and ask, “Do you want to remove this product from stock permanently”	Pass

Table 5.5 Adding Product to the stock

Add Product module

No.	Scenario	Output Expected	Status
1	Visible all the adding products on the drop-down menu	Display all the product details	Pass
2	When placing new order, display the available item quantity	Display the available item quantity of the stock	Pass
3	Enter the items more than available in the stock	Pop up Alerting “not enough stock “	Pass
4	Without entering Quantity of the form	Showing under the input field “This filed required”	Pass
5	Cancelling order	Pop up Alerting and ask, “Do you want to cancelling order”	Pass
6	Adding an Order without invoice details	Not allowed to add the record. Button not enable to update.	Pass

Table 5.6 Add Product module

New Order Place

No.	Scenario	Output Expected	Status
1	Without enter the data to mandatory (*) fields	Not allowed to add the record. Button not enable to update.	Pass
2	Add an existing product code to the Product Inventory	Showing under the input field “Already Exist the product code”	Pass
3	Add an existing product Name to the Product Inventory	Showing under the input field “Already Exist the product Name”	Pass
4	Delete a product	Pop up Alerting “Do you want to remove this product permanently”	Pass

Table 5.7 New Order Place

Adding Product to the stock

No.	Scenario	Output Expected	Status
1	Enter the data to mandatory (*) fields correctly	Allowed to Save the record. Button enable to save	Pass
2	Enter the letter to total amount	Showing under the input field “Invalid Amount”	Pass
3	Enter the numbers to total amount	Allowed to add the record. Button enable to add.	Pass
4	Expire date older than or equal manufacture date	Showing under the input field “Invalid Expired date”	Pass
5	Expire date equal to manufacture date	Showing under the input field “Invalid Expired date”	Pass
6	Enter the letter to Purchase price and sales price	Showing under the input field “Invalid Amount”	Pass
7	Delete a product from stock	Pop up Alerting “Do you want to remove this product from stock permanently”	Pass

Table 5.8 Adding Product to the stock

User Acceptance Testing

Before applying this system to production environment need to get the user acceptance of the system. Set of selected users to assign tasks with giving the privileges and UAT carried out the actual data for check the business requirements are met and the system and functions working properly. Familiar system will help for customer satisfaction.

CHAPTER 06 - CONCLUSION

6.1 Introduction

PARA Trading is a newly established and fast-growing company in the field of pharmaceutical. They mainly focused on distributing pharmaceutical, surgical, and diagnostic products in the local market and currently they are based on northwestern province. Currently they do not own a computerized system and all functionalities carried out manually which is not helpful to accommodate user requirements. The software solution delivered to the client, productively covered all operations which critical to the business. System includes easily manage sales, inquiry handling, inventory handling, order processing, stock maintain, and can get and manage required reports.

This system support to the effective working environment for PARA Trading's employees and it helps to their management for making quick decisions on the business activities. Productivity of the company will be enhanced with the improvement of the users on new system as they are very keen interest to use the system and motivated.

6.2 Future Improvements

Developer delivered a system which complements the requirements defined by the client after a good effort. But still there are some possible elements which can be developed in this system in order to enhance the productivity and efficiency. Developer identified following attributes as future improvements to the system.

6.2.1 PO process automation

Presently PARA Trading's does not enable the facility to place online purchase orders for their Customers and orders will be taken over the phone or through a sales staff member. To avoid this activity, purchasing order process will be fully automated which will be time consuming and easy for both parties to complete the process effectively.

6.2.2 Android App

Enabling this feature avoids unnecessary issues to the process. Sales staff member will be able to take sales orders through an Android App and system will be updated accordingly.

6.2.3 Printing of reports via Android App

Android App will allow printing the reports generated by the system such as invoices as soon as products deliver to the customer. This function will be easy for both customer and the sales staff to have a better business.

It is not practicable to implement the above elements within the limited time frame given by the UCSC. But this system persuades to develop above attributes in future to have a fully automated system increasing productivity.

6.3 Lessons Learnt

- Managing the time effectively is the main concern through entire project. Developer supposed to stick to the project plan while managing the time as well. While working on the system development project developer practically experienced how difficult to handle every task and how tactically use the time. This experience will help author in any future project management in terms of time factor.
- Gathering user requirements by using requirement gathering techniques is the most important key factor of the project as the success of the project depends on up to which extent will be able to deliver the product by satisfying client. This stage basically, helps to define the main scope of the project and set objectives accordingly. Understanding of user requirements clearly and observe the real process could help to develop system more efficiently.
- Keeping a proper communication with the client and relevant parties is very important and It helps to gained lot of experience throughout the project. Learnt how to connect with different types of users and get their ideas in order to enable the facilities require by them.
- Finally, by completing this software development project developer able to deliver a software solution which satisfies client and learnt the importance of report writing and how to conduct a business presentation to present work done in a professional manner towards the professional growth.

REFERENCES

[1] encyclopedia2.thefreedictionary Web Site - Systems Analysis Definition [Online]

<https://encyclopedia2.thefreedictionary.com/Systems+Analysis+Definition>

[2] Slideshare Website – Agile Development [Online]

<https://www.snyxius.com/implement-agile-development-process-easy-steps/>

[3] Snyxius Website – Agile Development [Online]

<https://www.snyxius.com/implement-agile-development-process-easy-steps/>

[4] MITRE Website – System Design and Development

<https://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/system-design-and-development>

[5] Techopedia Website – Object Oriented Analysis and Design

<https://www.techopedia.com/definition/21104/object-oriented-analysis-and-design-ooad>

[6] Wikipedia Website - AngularJs

<https://en.wikipedia.org/wiki/AngularJS>

[7] PostgreSQL Website

<https://www.postgresql.org/about/>

[8] Wikipedia Website – Apache Maven

https://en.wikipedia.org/wiki/Apache_Maven

[9] Spring Website – Spring boot

<https://spring.io/guides/gs/spring-boot/>

[10] JetBrains Website – IntelliJ IDEA

<https://www.jetbrains.com/idea/>

[11] Software Testing Fundamentals Website – Verification Vs Validation

<http://softwaretestingfundamentals.com/verification-vs-validation/>

[12] Software Testing Fundamentals Website – Test Plan

<http://softwaretestingfundamentals.com/test-plan/>

Appendix A - System Documentation

System documentation is the facts that describe the product to its users. End user will be able to take in proper guidance to install the stock and sales management system successfully. System documentation consists of the product technical manuals, online versions of the technical manuals and help facility descriptions. System administrator enables users to practice the system by installing the developed system to users' devices such as server, client PCs and Laptops. Documentation becomes the most important part hereafter as it either explains how the system operates or how to use the developed system or may indicate different things to people in different roles.

1. Creating an executable Jar

In the Java spring boot web application system need to create a complete self-contained jar file to run in a production environment. Executable Jars are archives containing compiled along with all of the jar dependencies that code needs to run.

To create an executable jar file need to add the “spring-boot-maven-plugin” in to our “pom.xml” file dependencies section.

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
```

Figure A.1 Creating an executable Jar file

Save “pom.xml” and run “mvn package” from the command line and in the “target” folder executable jar file build as “pharmapluz-0.0.1-SNAPSHOT.jar”. This is the original jar file that Maven created before it was repackaged by Spring Boot.

Run that application use “Java -jar” command in the server.

1. Install PostgreSQL in server

In here need to install PostgreSQL in the server and create a new connection by using New server registration interface.

New Server Registration

Properties | SSL | SSH Tunnel | Advanced

Name:

Host:

Port:

Service:

Maintenance DB:

Username:

Password:

Store password: ☒

Colour:

Group:

Help OK Cancel

Figure A.2 New Server register

After that want to create a new database (Schema) name as “Pharmapluz” and create all the table by using bellow SQL commands.

```
CREATE SCHEMA pharmapluz
AUTHORIZATION postgres;
```

Execute below SQL to create all the table, column fields, primary keys and foreign keys

```
/****** Add Table: pharmapluz.customer *****/

/* Build Table Structure */
CREATE TABLE pharmapluz.customer
(
    id serial NOT NULL,
    name TEXT NOT NULL,
    contact_number TEXT NOT NULL,
    address1 TEXT NOT NULL,
    address2 TEXT NULL,
    email TEXT NOT NULL,
```

Figure A.3 SQL Commands

```

created_by INTEGER NOT NULL,
    created_date TIMESTAMP NOT NULL,
    modified_by INTEGER NOT NULL,
    modified_date TIMESTAMP NOT NULL,
    region_id INTEGER NOT NULL
);

/* Add Primary Key */
ALTER TABLE pharmapluz.customer ADD CONSTRAINT pkcustomer
    PRIMARY KEY (id);

/***** Add Table: pharmapluz.drug *****/

/* Build Table Structure */
CREATE TABLE pharmapluz.drug
(
    id serial NOT NULL,
    name TEXT NOT NULL,
    created_by INTEGER NOT NULL,
    drug_type_id INTEGER NOT NULL,
    manufacture_id INTEGER NOT NULL
);

/* Add Primary Key */
ALTER TABLE pharmapluz.drug ADD CONSTRAINT pkdrug
    PRIMARY KEY (id);

/***** Add Table: pharmapluz.drug_inventory *****/

/* Build Table Structure */
CREATE TABLE pharmapluz.drug_inventory
(
    id serial NOT NULL,
    created_by INTEGER NOT NULL,
    quantity INTEGER NOT NULL,
    manufacture_date TIMESTAMP NOT NULL,
    expiry_date TIMESTAMP NOT NULL,
    created_date TIMESTAMP NOT NULL,
    modified_date TIMESTAMP NOT NULL,
    modified_by INTEGER NOT NULL,
    purchase_id INTEGER NOT NULL,
    drug_id INTEGER NOT NULL
);

/* Add Primary Key */
ALTER TABLE pharmapluz.drug_inventory ADD CONSTRAINT pkdrug_inventory
    PRIMARY KEY (id);

```

```
/****** Add Table: pharmapluz.drug_type *****/
```

```
/* Build Table Structure */
```

```
CREATE TABLE pharmapluz.drug_type  
(  
    id serial NOT NULL,  
    type TEXT NOT NULL,  
    created_by INTEGER NOT NULL,  
    modified_by INTEGER NOT NULL,  
    created_date TIMESTAMP NOT NULL,  
    modified_date TIMESTAMP NOT NULL  
);
```

```
/* Add Primary Key */
```

```
ALTER TABLE pharmapluz.drug_type ADD CONSTRAINT pkdrug_type  
    PRIMARY KEY (id);
```

```
/****** Add Table: pharmapluz.manufacture *****/
```

```
/* Build Table Structure */
```

```
CREATE TABLE pharmapluz.manufacture  
(  
    id serial NOT NULL,  
    name TEXT NOT NULL,  
    contact_number TEXT NOT NULL,  
    created_by INTEGER NOT NULL,  
    created_date TIMESTAMP NOT NULL,  
    modified_by INTEGER NOT NULL,  
    modified_date TIMESTAMP NOT NULL  
);
```

```
/* Add Primary Key */
```

```
ALTER TABLE pharmapluz.manufacture ADD CONSTRAINT pkmanufacture  
    PRIMARY KEY (id);
```

```
/****** Add Table: pharmapluz.order *****/
```

```
/* Build Table Structure */
```

```
CREATE TABLE pharmapluz.order  
(  
    id serial NOT NULL,  
    orders TEXT NOT NULL,  
    order_checked_by INTEGER NOT NULL,  
    created_date TIMESTAMP NOT NULL,  
    created_by INTEGER NOT NULL,  
    modified_date TIMESTAMP NOT NULL,  
    modified_by INTEGER NOT NULL,  
    customer_id INTEGER NOT NULL  
);
```



```

);

/* Add Primary Key */
ALTER TABLE pharmaapluz.order ADD CONSTRAINT pkorder
    PRIMARY KEY (id);

/***** Add Table: pharmaapluz.purchase *****/

/* Build Table Structure */
CREATE TABLE pharmaapluz.purchase
(
    id serial NOT NULL,
    supplier_id INTEGER NOT NULL
);

/* Add Primary Key */
ALTER TABLE pharmaapluz.purchase ADD CONSTRAINT pkpurchase
    PRIMARY KEY (id);

/***** Add Table: pharmaapluz.region *****/

/* Build Table Structure */
CREATE TABLE pharmaapluz.region
(
    id serial NOT NULL,
    name TEXT NOT NULL,
    postal_code INTEGER NOT NULL,
    created_date TIMESTAMP NOT NULL,
    created_by INTEGER NOT NULL,
    modified_by INTEGER NOT NULL,
    modified_date TIMESTAMP NOT NULL
);

/* Add Primary Key */
ALTER TABLE pharmaapluz.region ADD CONSTRAINT pkregion
    PRIMARY KEY (id);

/***** Add Table: pharmaapluz.sale *****/

/* Build Table Structure */
CREATE TABLE pharmaapluz.sale
(
    id serial NOT NULL,
    order_id INTEGER NOT NULL,
    customer_id INTEGER NOT NULL
);

/* Add Primary Key */
ALTER TABLE pharmaapluz.sale ADD CONSTRAINT pksale

```

```

PRIMARY KEY (id);

/***** Add Table: pharmapluz.supplier *****/

/* Build Table Structure */
CREATE TABLE pharmapluz.supplier
(
    id serial NOT NULL,
    name TEXT NOT NULL,
    address1 TEXT NOT NULL,
    address2 TEXT NULL,
    city TEXT NOT NULL,
    contact_number TEXT NOT NULL,
    email TEXT NOT NULL,
    created_by INTEGER NOT NULL,
    created_date TIMESTAMP NOT NULL,
    modified_by INTEGER NOT NULL,
    modified_date TIMESTAMP NOT NULL
);

/* Add Primary Key */
ALTER TABLE pharmapluz.supplier ADD CONSTRAINT pksupplier
    PRIMARY KEY (id);

/***** Add Table: pharmapluz.user *****/

/* Build Table Structure */
CREATE TABLE pharmapluz.user
(
    id serial NOT NULL,
    first_name TEXT NOT NULL,
    last_name TEXT NOT NULL,
    contact_number TEXT NOT NULL,
    address1 TEXT NOT NULL,
    address2 TEXT NULL,
    city TEXT NOT NULL,
    email TEXT NOT NULL,
    username TEXT NOT NULL,
    password TEXT NOT NULL,
    created_date TIMESTAMP NOT NULL,
    created_by INTEGER NOT NULL,
    modified_date TIMESTAMP NOT NULL,
    modified_by INTEGER NOT NULL,
    user_type_id INTEGER NOT NULL
);

/* Add Primary Key */
ALTER TABLE pharmapluz.user ADD CONSTRAINT pkuser
    PRIMARY KEY (id);

```

```
/****** Add Table: phmapluz.user_type *****/
```

```
/* Build Table Structure */
```

```
CREATE TABLE phmapluz.user_type  
(  
    id serial NOT NULL,  
    created_date TIMESTAMP NOT NULL,  
    created_by INTEGER NOT NULL,  
    modified_date TIMESTAMP NOT NULL,  
    modified_by INTEGER NOT NULL  
);
```

```
/* Add Primary Key */
```

```
ALTER TABLE phmapluz.user_type ADD CONSTRAINT pkuser_type  
    PRIMARY KEY (id);
```

```
/****** Add Foreign Keys *****/
```

```
/* Add Foreign Key: fk_customer_region */
```

```
ALTER TABLE phmapluz.customer ADD CONSTRAINT fk_customer_region  
    FOREIGN KEY (region_id) REFERENCES phmapluz.region (id)  
    ON UPDATE NO ACTION ON DELETE NO ACTION;
```

```
/* Add Foreign Key: fk_drug_drug_type */
```

```
ALTER TABLE phmapluz.drug ADD CONSTRAINT fk_drug_drug_type  
    FOREIGN KEY (drug_type_id) REFERENCES phmapluz.drug_type (id)  
    ON UPDATE NO ACTION ON DELETE NO ACTION;
```

```
/* Add Foreign Key: fk_drug_manufacture */
```

```
ALTER TABLE phmapluz.drug ADD CONSTRAINT fk_drug_manufacture  
    FOREIGN KEY (manufacture_id) REFERENCES phmapluz.manufacture (id)  
    ON UPDATE NO ACTION ON DELETE NO ACTION;
```

```
/* Add Foreign Key: fk_drug_inventory_drug */
```

```
ALTER TABLE phmapluz.drug_inventory ADD CONSTRAINT fk_drug_inventory_drug  
    FOREIGN KEY (drug_id) REFERENCES phmapluz.drug (id)  
    ON UPDATE NO ACTION ON DELETE NO ACTION;
```

```
/* Add Foreign Key: fk_drug_inventory_purchase */
```

```
ALTER TABLE phmapluz.drug_inventory ADD CONSTRAINT fk_drug_inventory_purchase  
    FOREIGN KEY (purchase_id) REFERENCES phmapluz.purchase (id)  
    ON UPDATE NO ACTION ON DELETE NO ACTION;
```

```
/* Add Foreign Key: fk_order_customer */
```

```
ALTER TABLE phmapluz.order ADD CONSTRAINT fk_order_customer  
    FOREIGN KEY (customer_id) REFERENCES phmapluz.customer (id)
```

ON UPDATE NO ACTION ON DELETE NO ACTION;

/* Add Foreign Key: fk_order_user */

ALTER TABLE pharmaapluz.order ADD CONSTRAINT fk_order_user
FOREIGN KEY (created_by) REFERENCES pharmaapluz.user (id)
ON UPDATE NO ACTION ON DELETE NO ACTION;

/* Add Foreign Key: fk_purchase_supplier */

ALTER TABLE pharmaapluz.purchase ADD CONSTRAINT fk_purchase_supplier
FOREIGN KEY (supplier_id) REFERENCES pharmaapluz.supplier (id)
ON UPDATE NO ACTION ON DELETE NO ACTION;

/* Add Foreign Key: fk_sale_customer */

ALTER TABLE pharmaapluz.sale ADD CONSTRAINT fk_sale_customer
FOREIGN KEY (customer_id) REFERENCES pharmaapluz.customer (id)
ON UPDATE NO ACTION ON DELETE NO ACTION;

/* Add Foreign Key: fk_sale_order */

ALTER TABLE pharmaapluz.sale ADD CONSTRAINT fk_sale_order
FOREIGN KEY (order_id) REFERENCES pharmaapluz.order (id)
ON UPDATE NO ACTION ON DELETE NO ACTION;

/* Add Foreign Key: fk_user_user_type */

ALTER TABLE pharmaapluz.user ADD CONSTRAINT fk_user_user_type
FOREIGN KEY (user_type_id) REFERENCES pharmaapluz.user_type (id)
ON UPDATE NO ACTION ON DELETE NO ACTION;

Appendix B - Design Documentation

Activity Diagram

Activity diagram of Inventory

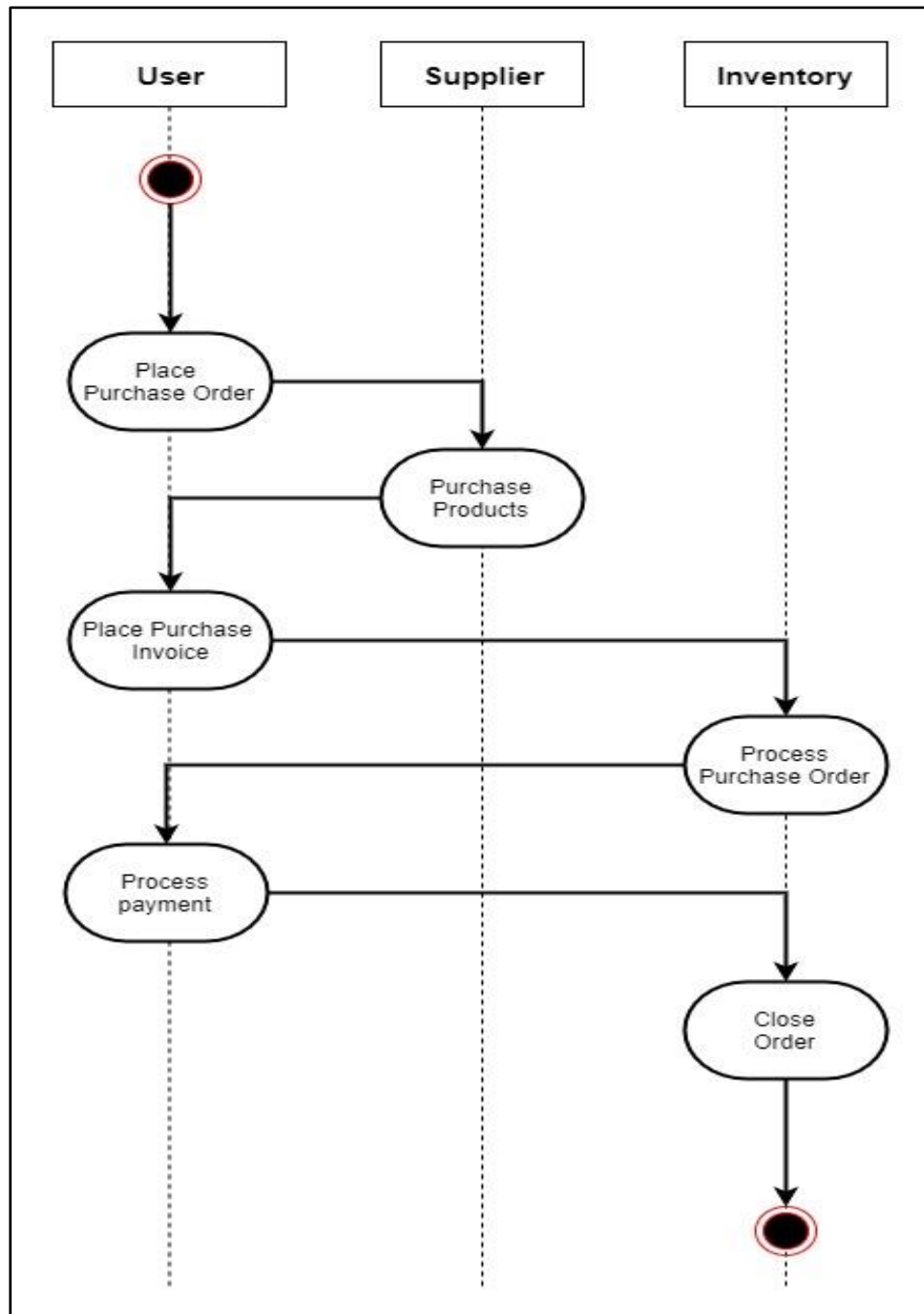


Figure B.1 Activity Diagram - Inventory

Activity Diagram - Order

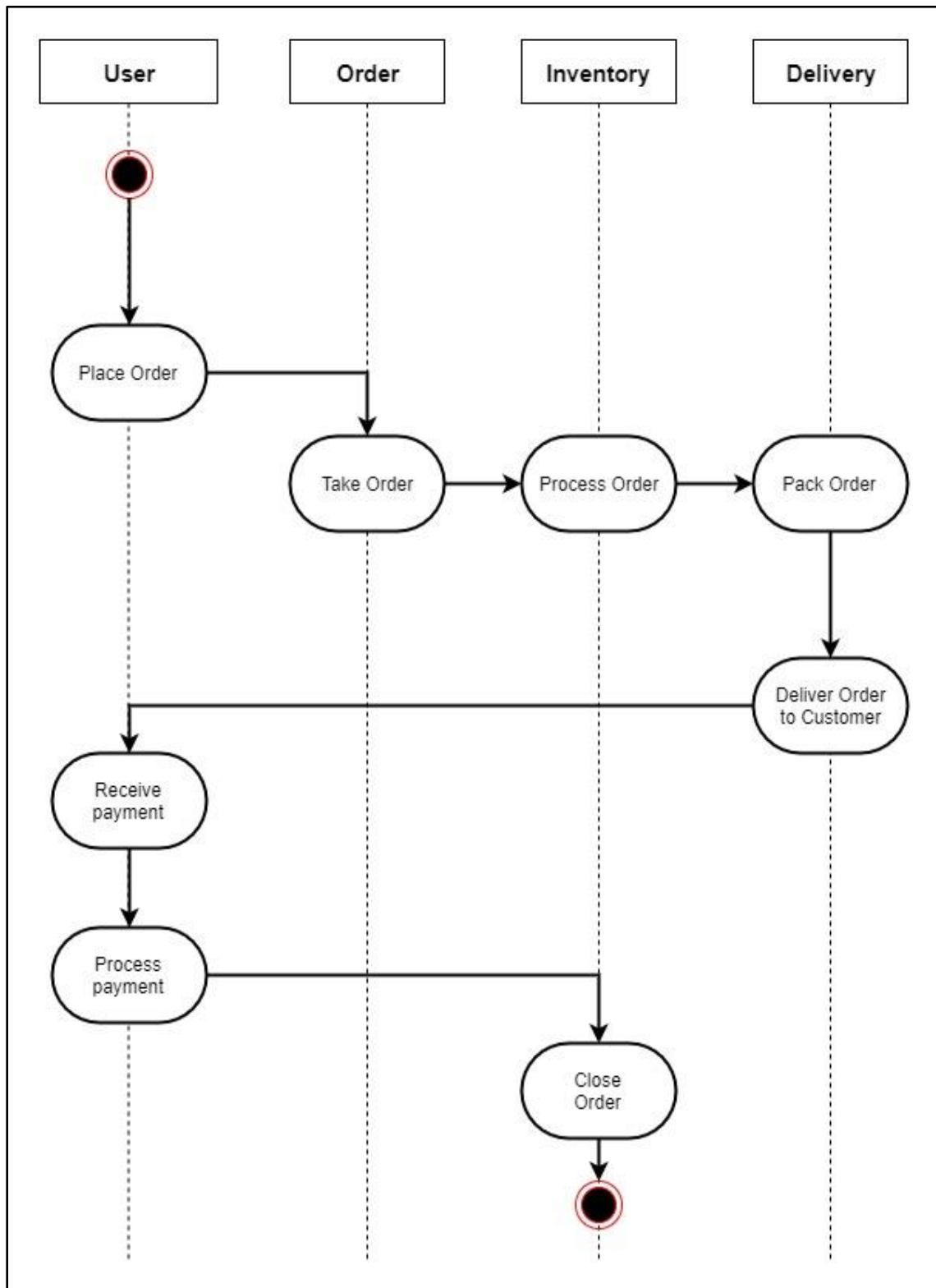


Figure B.2 Activity Diagram -Order

Activity Diagram - Arrears Payment

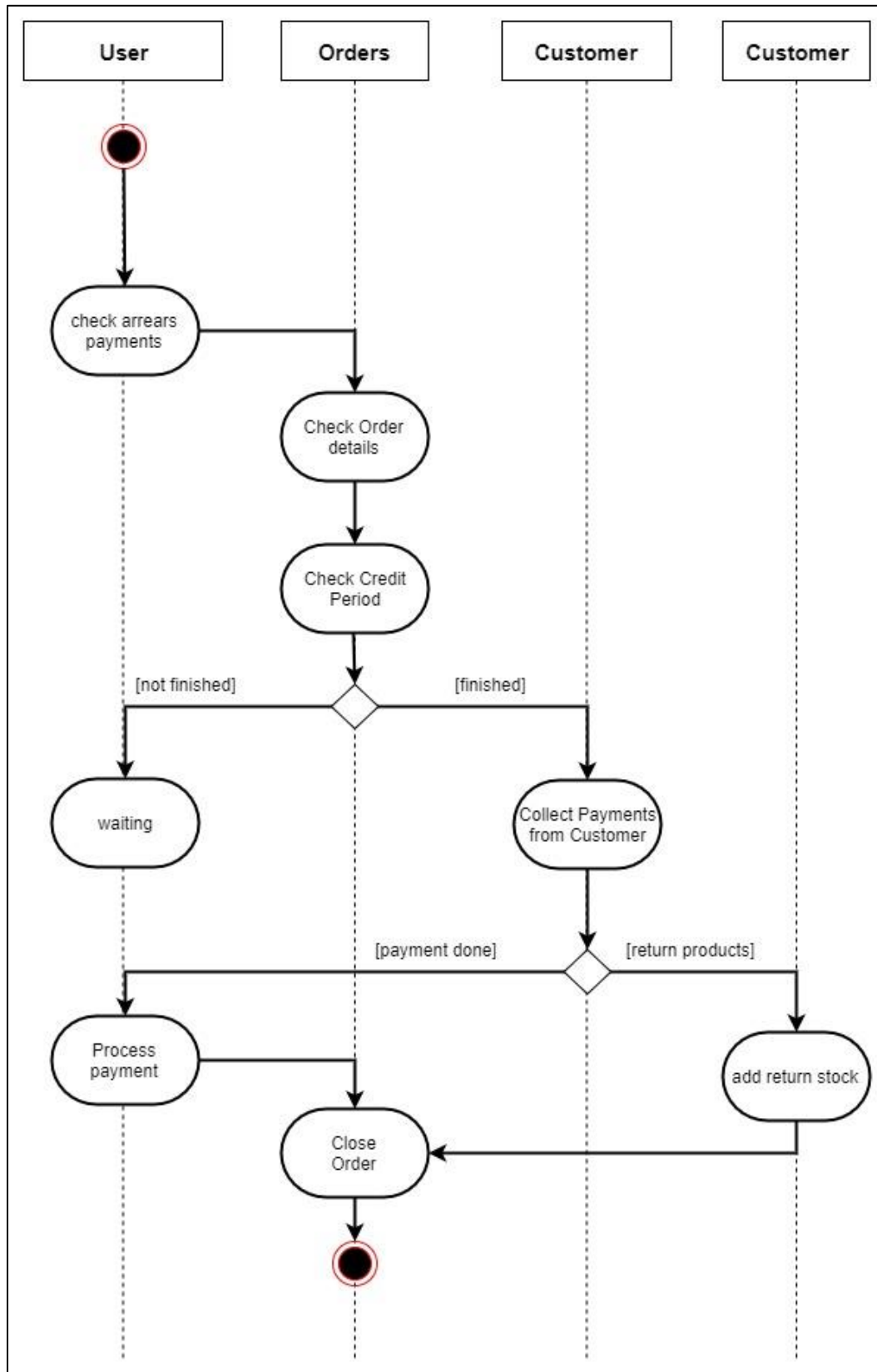


Figure B.3 Activity Diagram - Arrears Payment

Sequence Diagram

Sequence Diagram - Order

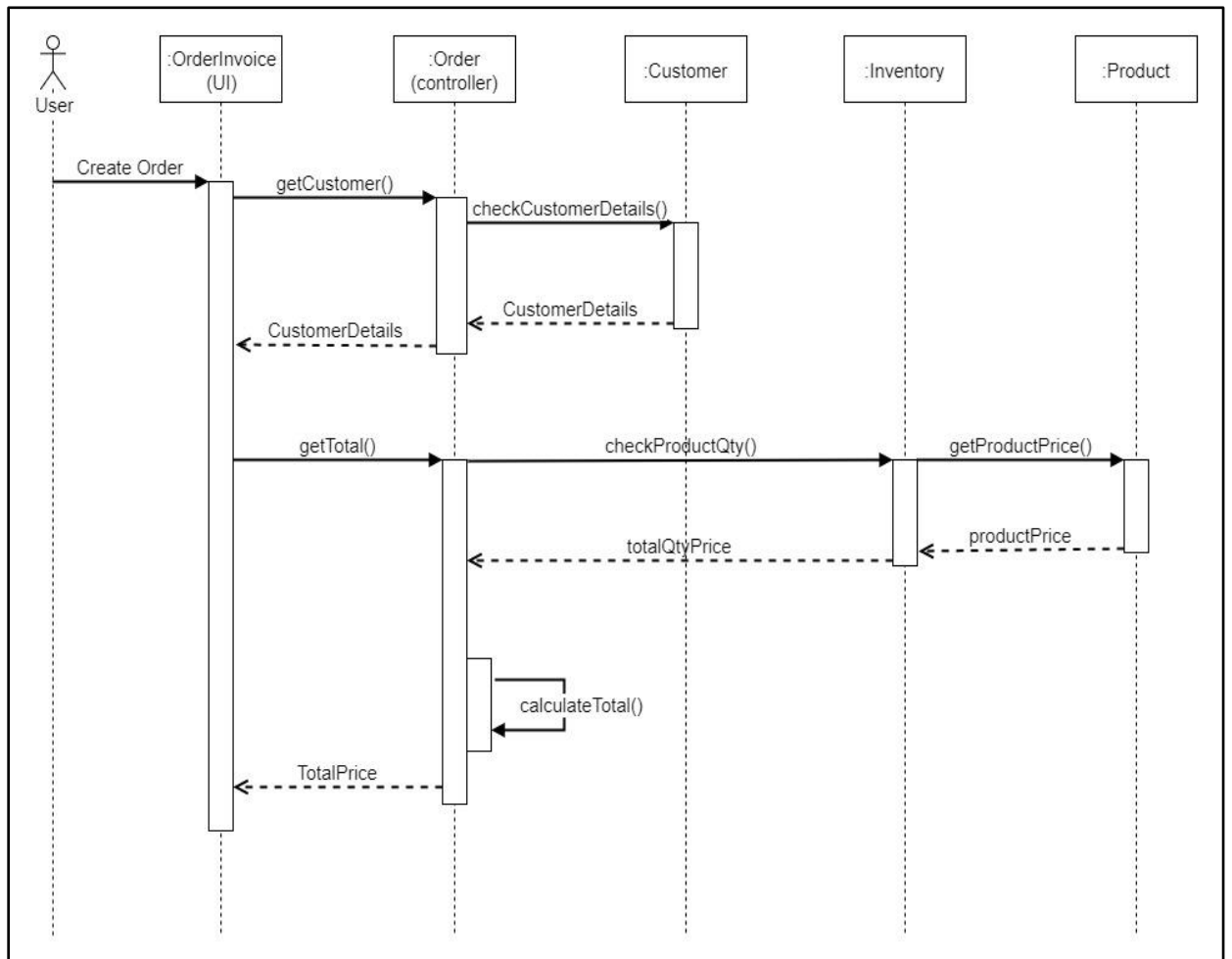


Figure B.4 Sequence Diagram -Order

Sequence Diagram – Order Reports

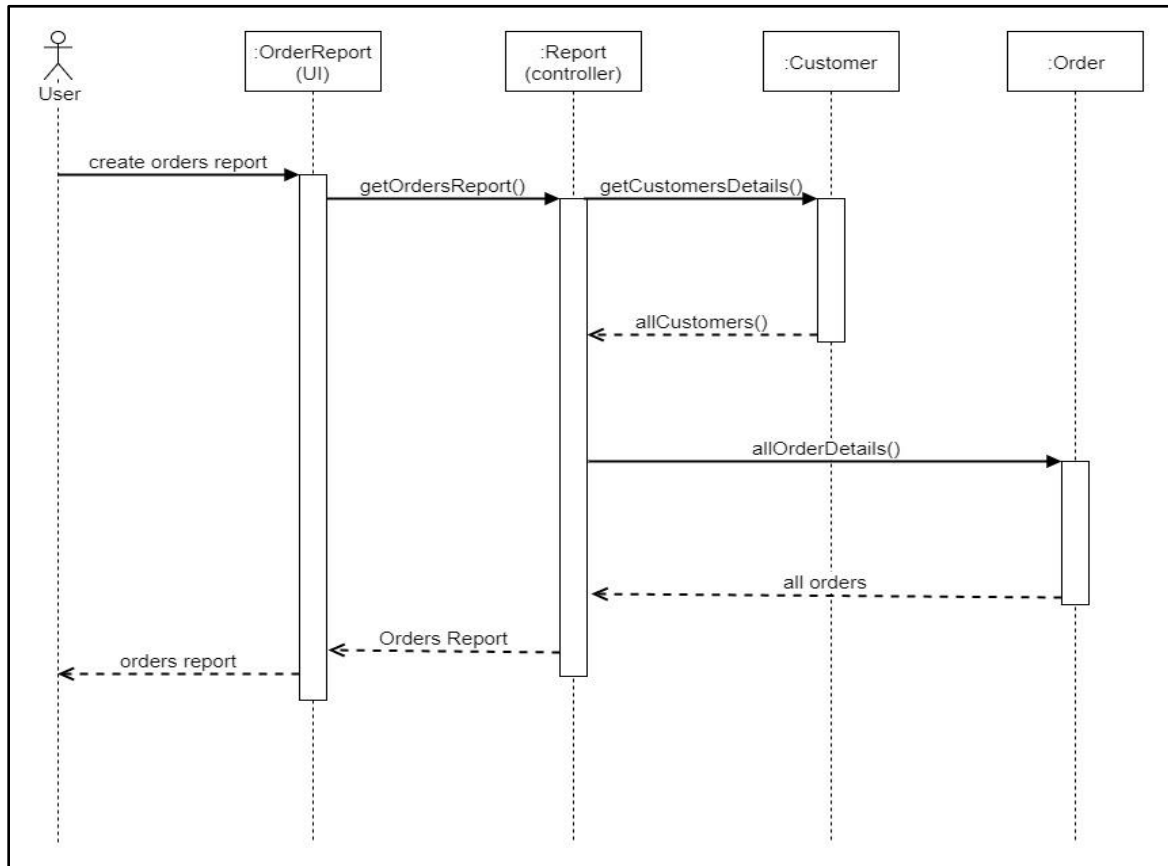


Figure B.5 Sequence Diagram -Order Reports

Sequence Diagram – Customer Create

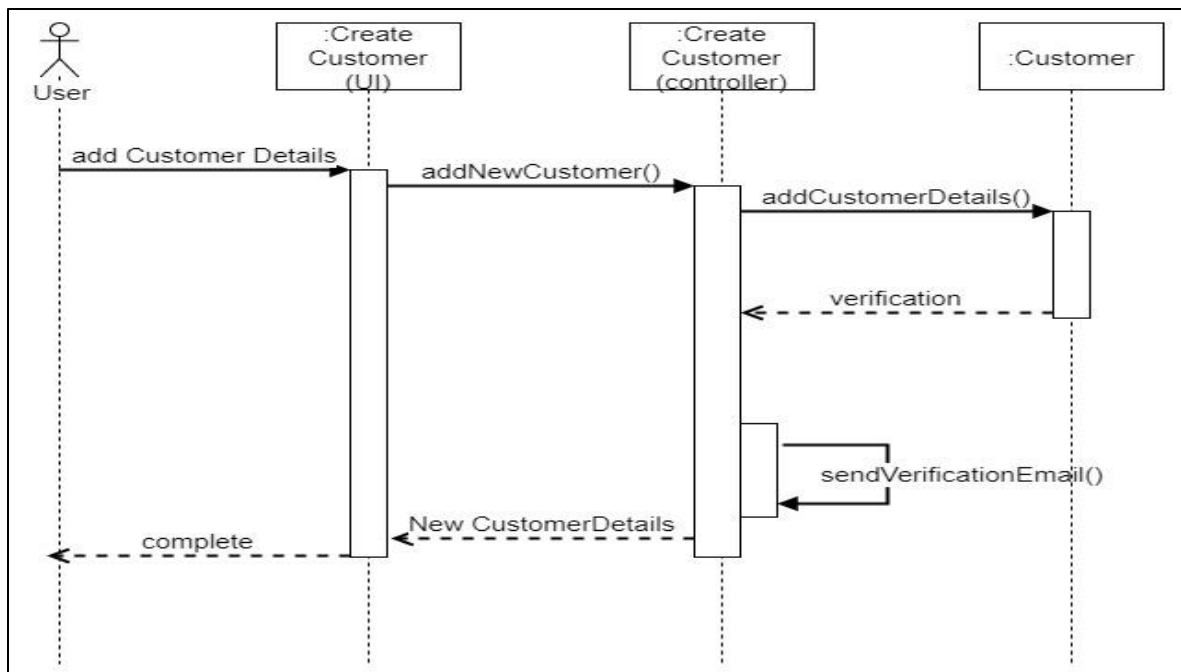


Figure B.6 Sequence Diagram -Customer Create

Sequence Diagram – Product Stock

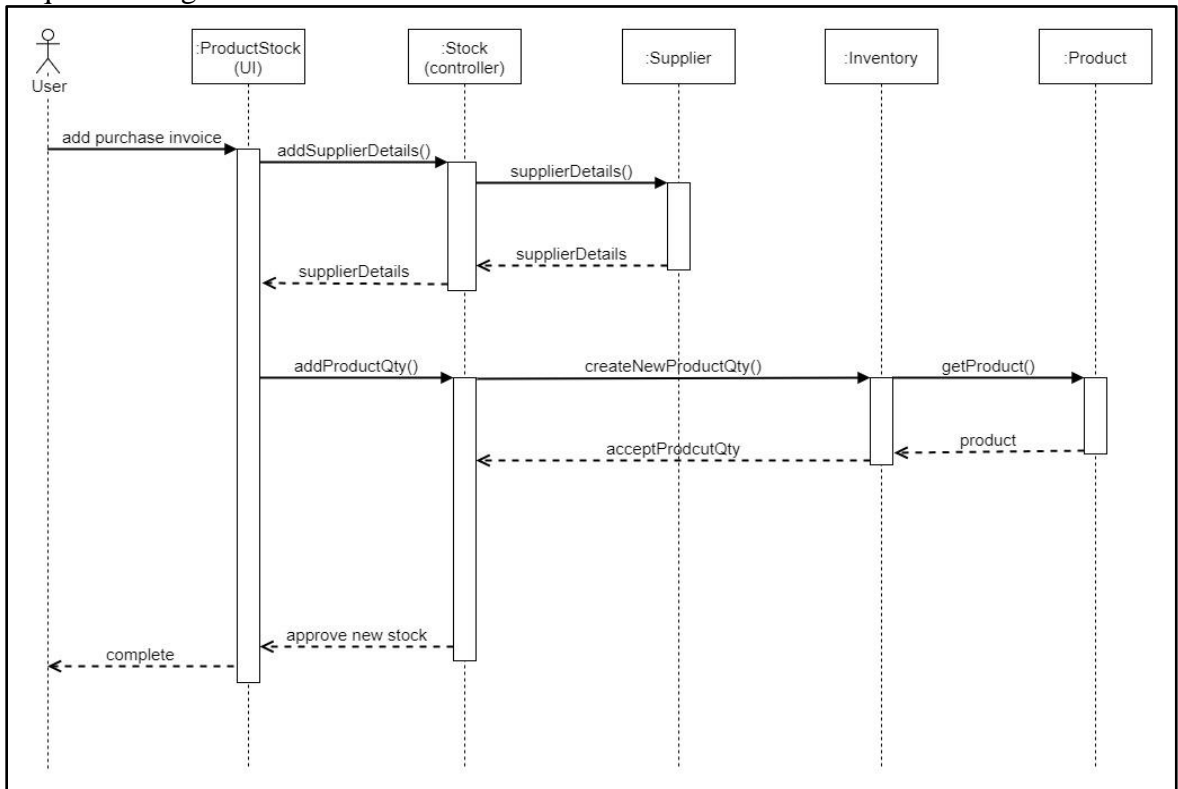


Figure B.7 Sequence Diagram -Product Stock

Sequence Diagram – Payment

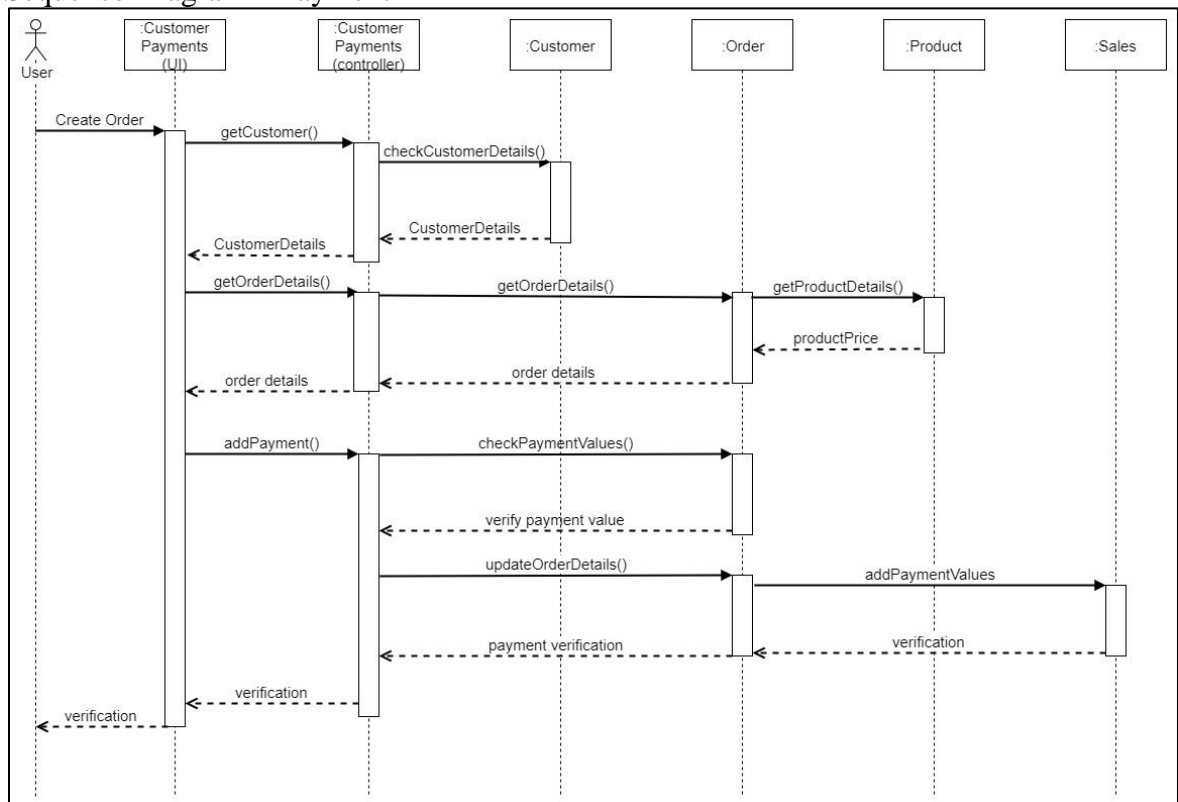


Figure B.8 Sequence Diagram -Payment

Appendix C - User Documentation

The user documentation is a part of the overall product delivered to the client and it provides assistance for the users who are supposing to use the stock and sales management system developed by the developer. Documentation for a product or service provided to the end users refers as user documentation. User documentation designed to assist end users to use the product or service. In order to perform day today activities with the implemented system user need to be aware about the functionalities of the system and user documentation will help on that. There are different types of system users in this system and system will grant privileges based on their user responsibilities.

User Login

An Authorized person who can login to the system using their credentials. After successfully login the system they can appear the main menu with the option with their user role in the company.

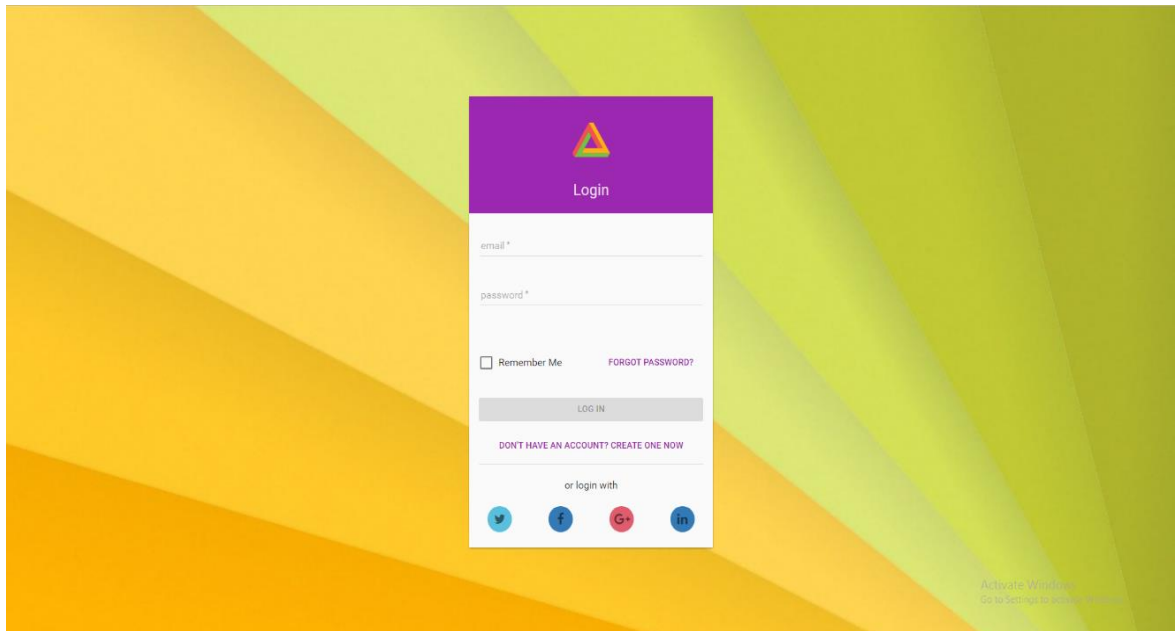


Figure C.1 Login Page

Create user accounts

Several users working this system and they are playing different roles in the system. They have different privileges. So, the admin can grant the privileges from creating different user accounts

The screenshot shows the 'Update User' form in the Pharmapluz system. The left sidebar contains a menu with options: Dashboards, User, Register New User, Update User, Customers, Suppliers, Stock, Order, Accounts, Products, and Permissions. The main content area is titled 'Update User' and shows a progress bar at 83% Complete. The form is divided into two sections: 'USER DETAILS' and 'UPDATE'. The 'USER DETAILS' section contains the following fields: User Type (Admin), First name (Ranjith), Last name (Bandara), Contact No (123456789), Email (asd@asd), Address 1 (asd), Address 2 (asd), and City (asd). An 'UPDATE' button is located at the bottom right of the form.

Figure C.2 Update user

Add new product to inventory

Before use the system need to add the products to the system.

Go to Main Menu -> Products -> Add New Product

The screenshot shows the 'Add New Product' form in the Pharmapluz system. The left sidebar contains a menu with options: Dashboards, User, Customers, Suppliers, Stock, Order, Accounts, Products, Add New Product, and Permissions. The main content area is titled 'Add New Product' and shows a progress bar at 71% Complete. The form is divided into two sections: 'PRODUCT DETAILS' and 'CATEGORY DETAILS'. The 'PRODUCT DETAILS' section contains the following fields: Product Code (PP042), Product Name (VERMOX SUSPENSION), Generic Name (VERMOX), Pack Size (6), and Manufacture (GSK product company). A 'NEXT' button is located at the bottom right of the form.

Figure C.3 Add New Product

- Enter All the mandatory (*) fields -> Click Next

Figure C.4 Add New Product

- Fill the category details -> Click Send

Once added the product to the system, it's appear on products Inventory

#	Name	Generic Name	Product Type	Manufacture
1	pandol	Amoxicillin	pill	GSK product company
2	amoxicillin	Amoxicillin	Syrup	GSK product company
3	VERMOX	Amoxicillin	Syrup	GSK product company
4	VERMOX TABLETS 500MG	VERMOX	Syrup	GSK product company
5	VERMOX	VERMOX	Syrup	GSK product company

Figure C.5 All Products

Adding the Products to Stock

Add invoice details of buying medicines from suppliers

- Go to the Stock on Main Menu -> Products Stock -> Click the NEW Icon on left-hand side

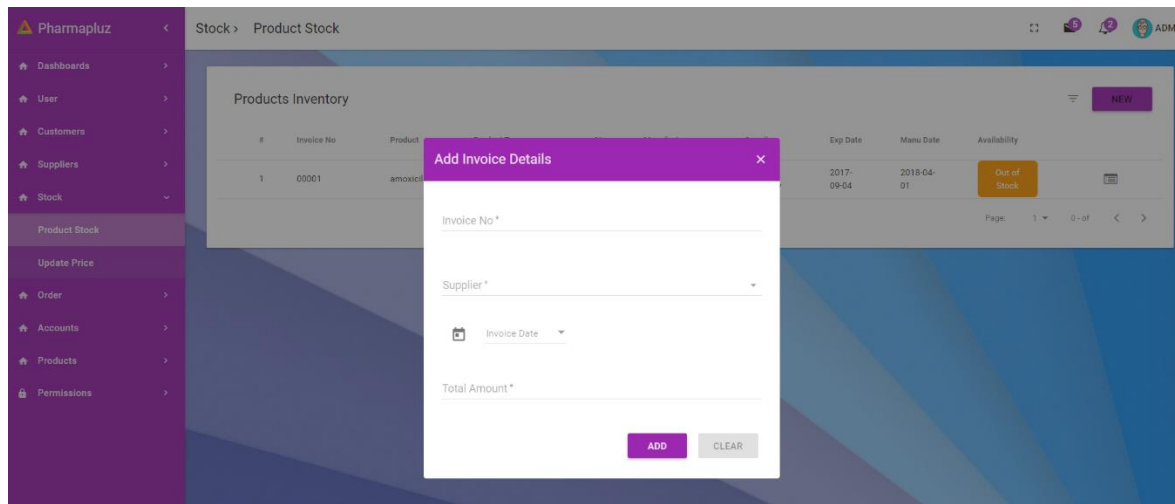


Figure C.6 Add invoice details

- Fill the all mandatory fields -> ADD

After Clicking the add button and popup stock window like below

The 'New Stock' modal form is shown with the following pre-filled information: 'Invoice No : P0050', 'Invoice Date : Aug 1, 2017', and 'Supplier : pettah pharmacy (pvt)'. Below this is a table with columns: '#', 'Drug', 'Drug Type', 'Qty', 'Manufacture', 'Manu Date', 'Exp Date', 'Purchase Price', and 'Sale Price'. The table contains one row with the following values: 'VERMOX TABLETS 500MG' for Drug, 'tablet' for Drug Type, '100' for Qty, 'GSK product company' for Manufacture, '7/1/2017' for Manu Date, '1/1/2018' for Exp Date, '1500' for Purchase Price, and '1600' for Sale Price. There is a checkmark icon and an 'ADD' button at the bottom right of the form.

Figure C.7 New Stock

- Select the product on “Product Name” drop down and fill other fields correctly

New order place

Order place mean selling medicines for the customers

- Go to the Order on Main Menu -> New Order -> Click the NEW INVOICE Icon on left-hand side

The 'Order Invoice Details' modal form is shown. It has a 'NEW INVOICE' button at the top right. Below it is a section titled 'Order Items' with a 'Total Amount: 0' on the right. The 'Order Items' section contains a table with columns: '#', 'Product', 'Product Type', 'Qty', 'Free Qty', 'Sale Price', and 'Total Price'. The table has one row with dropdown menus for 'Product Name *' and 'Product Type *', and input fields for 'Qty *', 'Free Qty', 'Unit Price', and 'Total Price'. At the bottom right of the form are 'COMPLETE ORDER' and 'CANCEL' buttons, along with a checkmark icon.

Figure C.8 Order Invoice Details

- Filled the invoice details

Add Invoice Details

Invoice No *

00023

Invoice Date

2017-11-08

Customer *

New Manel Pharmacy

ADD

CLEAR

Figure C.9 Order Invoice Details

- Filled the all field and select all the filed accordingly
- After that click COPMPLETE ORDER

Order Invoice Details

NEW INVOICE

Invoice Number : 0023 , Invoice Date : 2017-11-08

Customer : pettah pharmacy (pvt)

Order Items

Total Amount: 0

#	Product	Product Type	Qty	Free Qty	Sale Price	Total Price
	Product Name *	Product Type *	Qty *	Free Qty	Unit Price	Total Price
	VERMOX TABLETS 500MG	tablet	50	0	1500	75000

COMPLETE ORDER

CANCEL

Figure C.10 Order Invoice Details

Appendix D - Management Reports

System Reporting is most important part in the system and reports enables company to do strong reporting to understand how their system is being used. Below mention some sample reports in the proposed system.

Customer Information reports provide all customers in the system

PARA TRADING (PVT) LTD			CUSTOMER INFORMATION
			<i>Printed on 3-Sep-17 at 9:03 pm</i>
Code	Customer	Address	Telephone #s
D002	ASHA PHARMACY	MATHALE ROAD,, DAMBULLA	
D013	ASHA PHARMACY	NO: 633/1, ANURADHAPURA ROAD,, DAMBULLA	0662283690
D006	DAMBULLA OSUSALA	KURUNEGALA JUNCTION,, DAMBULLA	
D001	DARSHANA PHARMACY	NO. 68, KURUNEGALA ROAD,, DAMBULLA	0662283799
D012	DR W J M V PERERA	NO:34,MARKET ROAD,, DAMBULLA	0773216481
D017	DR. DAYANTHA WEERASEKARA	KURUNEGALA ROAD,, DAMBULLA	
D007	HARCOURTS PHARMACY (PVT) LTD	NO. 60, KURUNEGALA RD,, DAMBULLA	0667755400
D015	MAHAJANA PHARMACY	KANDY ROAD,, DAMBULLA	0712987788
D008	N I T PHARMACY	NO 102, KANDALAMA ROAD,, DAMBULLA	0719005888
D005	RANGIRI OSUSALA	38 A, KURUNEGALA RD,, DAMBULLA	
D014	SAMPATH PHARMACY	NO. 60, KURUNEGALA ROAD,, DAMBULLA	0777186850

Figure D.1 Customer Information Report

PARA TRADING (PVT) LTD			Supplier INFORMATION
			<i>Printed on 3-Sep-17 at 9:02 pm</i>
Code	Customer	Address	Telephone #s
AL	ALARIS LANKA (PVT) LTD.	NO. 9/3, HORANA ROAD,, KESBEWA,, PILIYANDALA	0112620167
CP	CHAMEE PHARMACEUTICALS (PVT) LTD	NO. 14, NEW BUS STAND,, ANURADHAPURA	
DE	DELKA ENTERPRISES (PVT) LTD	NO. 4, 3rd FLOOR, WEAVERS SHOPPING COMPLEX,, KURUNEGALA	0372267072
Del	Delmage Forsyth & Co. Ltd	101, Vinayalankara Mawatha,, P.O. Box 45,, Colombo 10	0117729330
DP	DHAMMIKA PHARMACEUTICALS	No. 42B, Technical Zone, Dahalyagama, Anuradhapura	0252237692
HD	HASITHA DISTRIBUTORS	NO. 18, YAKALLA ROAD,, KEKIRAWA	0766280125

Figure D.2 Supplier Information Report

PARA TRADING (PVT) LTD
PRODUCT PRICE- with margin

Supplier: PETTAH PHARMACY (PRIVATE) LIMITED
COLOMBO 10

Printed on 3-Sep-17 at 9:06 pm

Item Code	Description	Pack Size	Cost Price	W/S Price	Margin%
PP001	RETIN A CREAM	30 GM	575.75	575.75	0.00
PP002	RETIN A GEL	30 GM	689.75	689.75	0.00
PP003	EPREX INJECTION 2000 IU	1 S	1,189.95	1,189.95	0.00
PP004	EPREX INJ 4000IU	1	2,232.00	1,682.75	-32.64
PP005	GYNO PEVARYL DEPO OVULES 150MG	3 s	541.85	541.85	0.00
PP007	PEVARYL CREAM	15 GM	480.00	519.00	7.51
PP092	Maczee 250mg capsules	6 s	0.01	99.50	99.99
PP093	Maczee 500mg capsules	6 s	0.01	196.80	99.99
PP094	Olmesar 20mg	7 s	813.00	879.00	7.51

Figure D.3 Product Price Report

INVOICE No : 4690
INVOICE Date : 14-Aug-17
PRINTED ON : 3-Sep-17 9:00:16PM
SAVED ON : 28-Aug-2017 10:59 pm

Supplier PETTAH PHARMACY (PRIVATE) LIMITED

Item Code	Product	ExpD	Pack Size	Qty	Free	Dis %	W/S Price	Amount	Free Value
PP022	DAKTARIN CREAM	Jan-19	15 GM	150	0	0.00	428.00	63,900.00	0.00
PP035	DAKTARIN ORAL GEL	Oct-19	40g	50	0	0.00	687.00	34,850.00	0.00
PP028	IMODIUM CAPSULES loperamide capsule	Aug-21	60	30	0	0.00	1,885.00	55,950.00	0.00
PP030	MOTILLIUM SUSPENSION	Mar-20	100 ML	30	0	0.00	570.00	17,100.00	0.00
PP032	NIZORAL CREAM	Aug-19	15 GM	50	0	0.00	539.00	26,950.00	0.00
PP029	MOTILLIUM TABLETS	Jan-22	100	50	0	0.00	433.00	21,650.00	0.00
PP013	PEVISONONE CREAM	May-19	15 GM	50	0	0.00	499.00	24,950.00	0.00
PP050	STUGERON TAB 50MG	Jul-19	mg	50	0	0.00	583.00	29,150.00	0.00
PP042	VERMOX SUSPENSION	Jan-22	30 ML	30	0	0.00	285.00	7,950.00	0.00
PP041	VERMOX TABLETS 500MG Mebendazole	Dec-19	1	1200	0	0.00	117.00	140,400.00	0.00
PP007	PEVARYL CREAM	Jul-18	15 GM	50	0	0.00	480.00	24,000.00	0.00

No of Items 11

Gross Total 446,850.00
Dis Amount 0.00 % 0.00
Net Total 446,850.00
Net Free Value 0.00

Figure D.4 Invoice Report

Appendix E - Test Result

User Authentication

No.	Scenario	Output Expected	Status
1	User Login with correct user name and correct password	Login to main menu	Pass
2	User Login with correct user name and wrong password	Not allow login to main menu	Pass
3	User Login with wrong user name and correct password	Not allow login to main menu	Pass
4	User Login with wrong user name and wrong password	Not allow login to main menu	Pass

Table E.1 User Authentication

Register new system user

No.	Scenario	Output Expected	Status
1	Without enter the data to mandatory (*) fields	Not allowed to Save the record. Button not enable to save	Pass
2	Enter the data to mandatory (*) fields correctly	Allowed to Save the record. Button enable to save	Pass
3	Enter a contact number more than 10 digit	Showing under the input field “Invalid Number”	Pass
4	Enter a contact number less than 10 digit	Showing under the input field “Invalid Number”	Pass
5	Enter an email address without proper email address	Showing under the input field “Invalid email format”	Pass
6	Enter an email address correctly	Allowed to Save the record. Enable the save button.	Pass

Table E.2 Register New User

Update the System User

No.	Scenario	Output Expected	Status
1	Erase the data on mandatory (*) fields	Not allowed to update the record. Button not enable to update.	Pass
2	Update the data to mandatory (*) fields correctly	Allowed to update the record. Button enable to update.	Pass
3	Update a contact number more than 10 digit	Showing under the input field "Invalid Number"	Pass
4	Update a contact number less than 10 digit	Showing under the input field "Invalid Number"	Pass
5	Update an email address without proper email address	Showing under the input field "Invalid email format"	Pass
6	Update an email address correctly	Allowed to update the record. Button enable to update.	Pass
7	User Account Delete	Pop up Alerting and ask, "Do you want to remove this user permanently"	Pass

Table E.3 Update the System User

Adding Product to the stock

No.	Scenario	Output Expected	Status
1	Without enter the data to mandatory (*) fields	Not allowed to add the record. Button not enable to update.	Pass
2	Enter the data to mandatory (*) fields correctly	Allowed to Save the record. Button enable to save	Pass
3	Enter the letter to total amount	Showing under the input field "Invalid Amount"	Pass
4	Enter the numbers to total amount	Allowed to add the record. Button enable to add.	Pass
5	Expire date older than or equal manufacture date	Showing under the input field "Invalid Expired date"	Pass
6	Expire date equal to manufacture date	Showing under the input field "Invalid Expired date"	Pass

7	Enter the letter to Purchase price and sales price	Showing under the input field “Invalid Amount”	Pass
8	Delete a product from stock	Pop up Alerting and ask, “Do you want to remove this product from stock permanently”	Pass

Table E.4 Adding Product to the stock

Add Product module

No.	Scenario	Output Expected	Status
1	Visible all the adding products on the drop-down menu	Display all the product details	Pass
2	When placing new order, display the available item quantity	Display the available item quantity of the stock	Pass
3	Enter the items more than available in the stock	Pop up Alerting “not enough stock “	Pass
4	Without entering Quantity of the form	Showing under the input field “This filed required”	Pass
5	Cancelling order	Pop up Alerting and ask, “Do you want to cancelling order”	Pass
6	Adding an Order without invoice details	Not allowed to add the record. Button not enable to update.	Pass

Table E.5 Add Product module

New Order Place

No.	Scenario	Output Expected	Status
1	Without enter the data to mandatory (*) fields	Not allowed to add the record. Button not enable to update.	Pass
2	Add an existing product code to the Product Inventory	Showing under the input field “Already Exist the product code”	Pass
3	Add an existing product Name to the Product Inventory	Showing under the input field “Already Exist the product Name”	Pass
4	Delete a product	Pop up Alerting and ask, “Do you want to remove this product permanently”	Pass

Table E.6 New Order Place

Adding Product to the stock

No.	Scenario	Output Expected	Status
1	Without enter the data to mandatory (*) fields	Not allowed to add the record. Button not enable to update.	Pass
2	Enter the data to mandatory (*) fields correctly	Allowed to Save the record. Button enable to save	Pass
3	Enter the letter to total amount	Showing under the input field “Invalid Amount”	Pass
4	Enter the numbers to total amount	Allowed to add the record. Button enable to add.	Pass
5	Expire date older than or equal manufacture date	Showing under the input field “Invalid Expired date”	Pass
6	Expire date equal to manufacture date	Showing under the input field “Invalid Expired date”	Pass
7	Enter the letter to Purchase price and sales price	Showing under the input field “Invalid Amount”	Pass
8	Delete a product from stock	Pop up Alerting and ask, “Do you want to remove this product from stock permanently”	Pass

Table E.7 Adding Product to the stock

Goods returning Module

No.	Scenario	Output Expected	Status
1	Find an existing goods returning records	Search the relevant record and display	Pass
2	Without enter the data to mandatory (*) fields	Not allowed to add the record. Button not enable to update	Pass
3	Existing record again enter	Showing under the input field “Already Exist”	Pass

Table E.8 Goods Returning Module

Appendix F – Code Listening

```
(function() {
  'use strict';
  angular
    .module('app.pharmapluz.inventory')
    .config(moduleConfig);

  function moduleConfig($stateProvider, triMenuProvider) {

    $stateProvider
      .state('triangular.addDrugs', {
        url: '/adddrugs',
        templateUrl: 'app/pharmapluz/inventory/add-drug/add-drug.tmpl.html',
        controller: 'AddDrugController',
        controllerAs: 'ad',
        data: {
          layout: {
            contentClass: 'layout-column full-image-background mb-bg-fb-02 background-
overlay-static',
            innerContentClass: 'overlay-gradient-20'
          }
        }
      })
      .state('triangular.productStock', {
        url: '/productstock',
        templateUrl: 'app/pharmapluz/inventory/products-stock/products-stock.tmpl.html',
        controller: 'productStockController',
        controllerAs: 'vm',
        data: {
          layout: {
            contentClass: 'layout-column full-image-background mb-bg-fb-02 background-
overlay-static',
            innerContentClass: 'overlay-gradient-20'
          }
        }
      })
      .state('triangular.updatePrice', {
        url: '/updateprice',
        templateUrl: 'app/pharmapluz/inventory/update-price/update-price.tmpl.html',
        controller: 'UpdatePriceController',
        controllerAs: 'up',
```

```

        data: {
            layout: {
                contentClass: 'layout-column full-image-background mb-bg-01 background-overlay-
static',
                innerContentClass: 'overlay-gradient-20'
            }
        }
    });

    triMenuProvider.addMenu({
        name: 'Stock',
        icon: 'zmdi zmdi-home',
        type: 'dropdown',
        priority: 5.1,
        children: [
            {
                name: 'Product Stock',
                state: 'triangular.productStock',
                type: 'link'
            }, {
                name: 'Update Price',
                state: 'triangular.updatePrice',
                type: 'link'
            }
        ]
    });
}
})();

```

Figure F.1 Inventory module

```

(function(){
    'use strict';

    angular
        .module('app.pharmapluz.order')
        .controller('NewOrderController', NewOrderController);

    NewOrderController.$inject = ['$window', '$mdDialog', '$filter', 'OrderService', 'InventoryService',
    '$mdToast', '$state'];
    function NewOrderController($window, $mdDialog, $filter, OrderService, InventoryService,
    $mdToast, $state) {
        var noc = this;

        noc.isInvoiceDetails = false;
        noc.orderInvoiceDetails = null;
        var invoiceData = false;

        noc.totalAmount = 0.00;

        var productList = [];
        var priceList = [];

        if(productList.length == 0){
            var allProduct = new InventoryService.ProductList();
            allProduct.getProductList();
            allProduct.completed.then(function(){
                angular.forEach(allProduct.ProductList, function(val){
                    productList.push(val);
                });
            }, function(err){
                console.log(err);
            });
            noc.product = productList;
        }

        if(priceList.length == 0){ // this is happened several times :(
            var allPrices = new OrderService.ItemPriceList();
            allPrices.getAllItemsPrices();
            allPrices.completed.then(function(){
                angular.forEach(allPrices.ItemPriceList, function(val){
                    priceList.push(val);
                });
            }, function(err){
                console.log(err);
            });
        }
    }
}

```



```

    noc.getUnitPriceList = function($event, productId, typeId, qty, freeQty){
        invoiceData = true;
        var totalQty = (!!qty ? parseInt(qty) : 0) + (!!freeQty ? parseInt(freeQty) : 0);

        if (!!productId && !!typeId){

            var allUnitprices = [];

            angular.forEach(priceList, function(val){
                if(val.inventory.productId == productId){
                    allUnitprices.push(val);
                }
            });
            $mdDialog.show({
                controller: 'SelectedItemPriceListController',
                controllerAs: 'pl',
                templateUrl: 'app/pharmapluz/order/new-order/selected-item-price-list-
dialog.tmpl.html',
                locals: {
                    itemPrices: allUnitprices,
                    requiredQty: totalQty
                },
                targetEvent: $event
            }).then(function(results){
                noc.item.price = results.price;
                noc.item.inventoryId = results.inventory.id;
                if(noc.item.stockQuantity > 0){
                    noc.calTotalPrice(noc.item.stockQuantity);
                }
            });
        }
    }

    noc.newOrderInvoice = function($event){
        invoiceData = true;

        $mdDialog.show({
            controller: 'orderInvoiceDetailController',
            controllerAs: 'oinvd',
            templateUrl: 'app/pharmapluz/order/new-order/order-invoice-detail.tmpl.html',
            locals: {
            },
            targetEvent: $event
        }).then(function(results){
            console.log(results);
            noc.orderInvoiceDetails = results;
            noc.isInvoiceDetails = true;
        });
    }
}

```

```

    noc.getUnitPrice = function($event, id){
        var priceById = priceList.filter(function(val){
            return val.inventory.id == id;
        });
        noc.item.price = priceById[0].price;

        if(noc.item.stockQuantity > 0){
            noc.calTotalPrice(noc.item.stockQuantity);
        }
    }

    noc.calTotalPrice = function(qty){
        var total = noc.item.price * qty;
        noc.item.totalPrice = total;
    }

    noc.allItems = [];

    noc.productTypes = [{
        id: 1,
        type: "Syrup"
    }, {
        id: 2,
        type: "pill"
    }, {
        id: 3,
        type: "tablet"
    }
    ];

    noc.customers = [{
        id: 1,
        name: "xyz (pvt)"
    }
    ];

    noc.addNewItem = function(){

        angular.forEach(priceList, function(val){
            if(val.inventory.id == noc.item.inventoryId){
                val.inventory.stockQuantity -= (!!noc.item.stockQuantity ?
                parseInt(noc.item.stockQuantity) : 0) + (!!noc.item.freeQuantity ?
                parseInt(noc.item.freeQuantity) : 0));
            }
        });
        noc.totalAmount += noc.item.totalPrice;
        noc.allItems.push(noc.item);
        noc.item = {};
    }
}

```

```

    noc.

    getProductType = function (productTypeid) {
        var getProductType = $filter('filter')(noc.productTypes, {id:productTypeid})[0];
        return getProductType.type;
    }

    noc.getCustomer = function(id){
        var customer = $filter('filter')(noc.customers, {id:id})[0];
        return customer.name;
    }

    noc.completeOrder = function(){
        var invoiceOrder = {
            invoiceNumber: noc.orderInvoiceDetails.invoiceNumber,
            invoiceDate: noc.orderInvoiceDetails.invoiceDate,
            customer: noc.orderInvoiceDetails.customers,
            order: noc.allItems,
            totalAmount: noc.totalAmount
        }

        console.log(invoiceOrder);

        var newOrder = new OrderService.Order();
        newOrder.createNewOrder(invoiceOrder);
        newOrder.completed.then(function(){
            $mdToast.show({
                template: '<md-toast><span flex>Order Completed !!</span></md-toast>',
                position: 'top right',
                hideDelay: 2000,
            });
            $state.reload();

        }, function(err){
            console.log(err);
        });
    }

    noc.getProductName = function (productId) {
        var getProduct = $filter('filter')(noc.product, {id:productId})[0];
        return getProduct.name;
    }

}

```

Figure F.2 Order module

```

(function(){
  'use strict';

  angular
    .module('app.pharmapluz.products')
    .controller('AllProductsController', AllProductsController);

  AllProductsController.$inject = ['ProductsService', '$state', '$mdToast', '$scope', '$filter'];

  function AllProductsController(ProductsService, $state, $mdToast, $scope, $filter) {
    var alp = this;

    alp.allProducts = [];
    getAllProducts();

    alp.productType = [{
      id: 1,
      type: "Syrup"
    }, {
      id: 2,
      type: "pill"
    }, {
      id: 3,
      type: "tablet"
    }
  ];

    alp.manufactures = [{
      id: 1,
      name: "ABC company"
    }
  ];

  function getAllProducts(){
    var allProducts = new ProductsService.ProductList();
    allProducts.getProductList();
    allProducts.completed.then(function(){
      angular.forEach(allProducts.ProductList, function(val){
        alp.allProducts.push(val);
      });
    }, function (err) {
      console.log(err);
    });
  };
};

```

```

alp.query = {
  filter: "",
  limit: '10',
  order: '-id',
  page: 1
};
alp.selected = [];
alp.columns = {
  avatar: 'Avatar',
  login: 'Login',
  id: 'ID'
};
alp.filter = {
  options: {
    debounce: 500
  }
};
alp.removeFilter = removeFilter;

activate();

function activate() {
  var bookmark;
  $scope.$watch('alp.query.filter', function (newValue, oldValue) {
    if(!oldValue) {
      bookmark = alp.query.page;
    }

    if(newValue !== oldValue) {
      alp.query.page = 1;
    }

    if(!newValue) {
      alp.query.page = bookmark;
    }

  });
}

function removeFilter() {
  alp.filter.show = false;
  alp.query.filter = "";

  if(alp.filter.form.$dirty) {
    alp.filter.form.$setPristine();
  }
}

```

```

alp.convertToDate = function(jsonDate){
    return $filter('date')(jsonDate, 'yyyy-MM-dd');
}

alp.getProductType = function (id) {
    var type = $filter('filter')(alp.productType, {id:id})[0];
    return type.type;
}

alp.getManufacture = function (id) {
    var manufa = $filter('filter')(alp.manufactures, {id:id})[0];
    return manufa.name;
}

alp.updateProduct = function ($event, product) {
    $state.go('triangular.updateProduct', { redirectingFromPreviousPage: true,
productDetails: product});
}

}

})();

```

Figure F.3 All Product module

Appendix G – CLIENT CERTIFICATION

PARA Trading (Pvt) Ltd.

Tele : 037 - 3763240 Fax :037- 4502402 E-mail : paratrading.pvt@gmail.com

No. 25/2, Wijayaba Mawatha,
Negombo Road,
Kurunegala.

5th November 2017

Project examination Board,
BIT,
University of Colombo school of computing,
Colombo 7.
Dear Sir / Madam,

LETTER OF CERTIFICATION

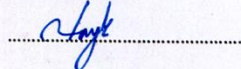
This is to inform you that Mr. G.K.D.K.Gajanayeka (R092721) has successfully developed " Para Trading Stock & Sales management System " for Para Trading (PVT) Ltd, DS Senanayeka Mawatha, Anuradhapura.

We strongly believe that this system will support to us to carry our business smoothly.

I wish all the success in his future endeavors.

Thank you,

Yours Faithfully,



Ranjith J. Bandara,

Director,

Para Trading (pvt) Ltd.

PARA TRADING (PVT) LTD.
25/2, Wijayaba Mawatha,
Negombo Road, Kurunegala Tel: 037 3763240

Index

A		I	
Abstract	iii, v	Idea	22, 40
Acknowledgement	iv	Implement	37
Agile	vi, viii, 9, 10, 11, 39	Implementation	iii, 1, 19, 20
Angular	iii, 20, 21	Inquiries	1, 7
Apache	21, 22, 40	Interface	v, 15
		Interview	4
C		Inventories	iii
Computing	22		
Conclusion	36	J	
Customer	vii, 6, 7, 8, 10, 14, 62, 63	Java	iii, 20, 22, 42
D		L	
Database	v, vi, vii, 13, 14, 29	Learnt	vi, 38
Design	v, 4, 10, 11, 12, 13, 15, 19, 22, 30, 31, 39, 40	Lessons	38
Design	50, 57	List	vi, viii
Diagram	vi, vii, 6, 23, 50, 51, 53, 54, 55, 56	Loading	vii
Documentation	41, 50, 57, 62, 65, 71, 72, 81, 82	Login	v, vi, vii, 16, 24, 25, 26, 27, 28, 31, 57, 65
E		M	
Encouraging	12	Management	vi, 2, 62
Environment	36	Manual process	1, 5, 9
Established	iii, 1, 3, 36	Methodology	v, 9
Evaluation	19, 30	Modifications	14
Existing	34, 68	N	
Existing	69	Normalization	v, 14, 40
F		O	
Figure	vi, vii, viii, 6, 11, 13, 15, 16, 17, 18, 19, 21, 23, 24, 26, 27, 28, 29, 31, 32, 33, 34, 35, 51, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 72, 77, 80	Object-oriented	12
Focused	iii, 36	Observation	5
Framework	iii, 20, 21	Order	vii, viii, 2, 7, 18, 34, 35, 51, 53, 54, 55, 60, 61, 62, 68, 77
Framework	vii	P	
Functional requirements	v, 8, 9	Pharmaceutical	iii, 1, 3, 36
Future	37	Planning	iii, 1, 3, 31
G		Postgresql	iv, 20, 21, 40, 42
Google	iii, 20	Principle	10
		Process	iii, v, 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 14, 30, 31, 37, 38, 39
		Processing	vii
		Product	34, 68
		Products	iii, 3, 7, 8, 30, 34, 36, 37, 58, 59, 67

Project	v, 1	System	iv, vi, vii, 31, 36, 41
Proposed	v, vi, 5, 11, 13, 15, 31	Systems	vi
	R		T
Raw material	viii	Test plan	31
Reducing	iii, 2, 3, 10		
Reports	vi, 55, 62		U
Result	vi, 65	Ucsc	37
	S	Unambiguous	14
Scope	v, 2	User	iii, viii, 7, 9, 15, 20, 30, 31, 32, 33, 36, 38, 41, 47, 48, 49, 57, 58, 65, 66
Sequence	vi, 54, 55, 56	User	35
Spring boot	iii, 20, 42		
Sql server	vii		V
Stakeholder	12	Validation	30
Stock	vii, viii, 2, 8, 9, 56, 59, 60		
Supplier	vii, 6, 7, 17, 63		