Inventory Management System

Sajan Rajbhandari

Table of Contents

1. Introduction to the System

1.1 Problem Definition

This software, “Inventory Management System”, is used for recording the information about the day to day transaction of stocks of an organization. It stores purchase information of the products with credit/debit information from the supplier. Similarly, it stores sales information with credit/debit about the customers. If a product is purchased, then the related information is stored in stocks, that is, stocks are up to date. Another part is it prepares sales report after product is sold. In the sales information, the information about who sold the product is also kept, so there is no problem for misunderstandings in future.

1.2 Objectives and scope of the project

The project is a remarkable chance to experience the real world working environment and culture where the knowledge learned during the BIM course can be implemented. This project not only makes us familiar with the real working environment but also make us more mature in the way we deal with the real world problem and try to solve those problem in the best way possible by applying the knowledge we have acquired throughout the BIM course.

The main objective of the project is to analyze the existing system under study and give necessary suggestions or solutions to improve it. To implement the theoretical knowledge acquired from college in real working environment.

To enable us to understand how theory knowledge differs from practical life thus helping us to understand the complexity and unforeseen nature of problem and opportunity that exist in the country

As its name implies, the main objective of this software is to record the information about the stocks of an organization and perform basic operations,

purchase and sales of the products. It is developed to increase the efficiency of an organization as it can perform tasks quickly and accurately.

This software can be useful for small to medium size organization where stocks are required to be managed day to day. This software can be useful for handling the inventory as compared with traditional paper system.

1.3 Benefits of the project

- Developed teamwork
- Developed coding and documentation skills
- Increased knowledge about how organization manages its inventory
- Learn to automate the existing system as far as practically within limited time constraints
- To enable the students gain better understanding of different aspects of the working environment and working condition.
- To help the students gain experience of professional working environment.
- To analyze the real world problem and find the solution using knowledge obtained.
- To enhance knowledge and skills necessary to be an effective manager.
- To develop and enhance research skills, report writing skills along with presentation and communication skills.
- To get career insights existing in the country.

1.4 Limitations of the project

Due to the constraints with time, there are certain limitations of this project; some of them are highlighted below:

- Does not consist accounting features

- Not suitable for large organization
- Main focus was given in the functional requirement of the system
- Time period was not enough for a comprehensive study and development of the software of Inventory Management System.

1.5 Feasibility Assessment

The **feasibility study** is an evaluation and analysis of the potential of a proposed project which is based on extensive investigation and research to support the process of decision making.

- **Economic Feasibility**

  As we need not to perform high level researches on our project, we did not spend any amount while preparing this project.

- **Technical Feasibility**

  There is no problem in technical feasibility as it supports basic hardware and software. Other software required for this project can be easily available in the internet. To use this software, no technical person is required. It is user friendly and can be used by any non-technical person.

- **Operation Feasibility**

  Our software runs smoothly in the given software and hardware requirements. It does not consist extra requirements.

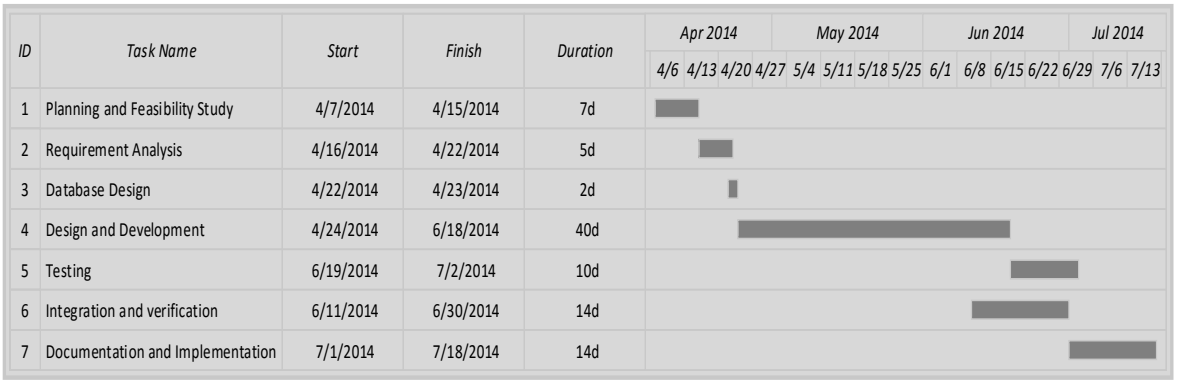
- **Legal Feasibility**

  This software does not hamper any legal matters, so it is legally feasible.

- **Schedule Feasibility**

  We had about 4-5 months time to prepare this software and we have completed it within

Gantt chart for Inventory Management System:



ID	Task Name	Start	Finish	Duration
1	Planning and Feasibility Study	4/7/2014	4/15/2014	7d
2	Requirement Analysis	4/16/2014	4/22/2014	5d
3	Database Design	4/22/2014	4/23/2014	2d
4	Design and Development	4/24/2014	6/18/2014	40d
5	Testing	6/19/2014	7/2/2014	10d
6	Integration and verification	6/11/2014	6/30/2014	14d
7	Documentation and Implementation	7/1/2014	7/18/2014	14d

1.6 Tools used

For the development IMS system a variety of tools and techniques are used. For the development of the system the user requirements has to be written in the understandable form. The use of different graphical representation of the system process has been implemented in development of system for non technical users to understand the system working process. We used the mostly used Tools and techniques for system development as:

- Net Beans IDE (Coding)
- JDK 1.7_3
- MySQL server
- MS Visio (Designing)
- MS Word (Documentation)
- Adobe Photoshop (Graphic Designing)

2. Analysis of the System

For the development of IMS we have follow a traditional system development procedure which is generally known as system development life cycle (SDLC). The general steps that we have followed for development of system can be shown below:

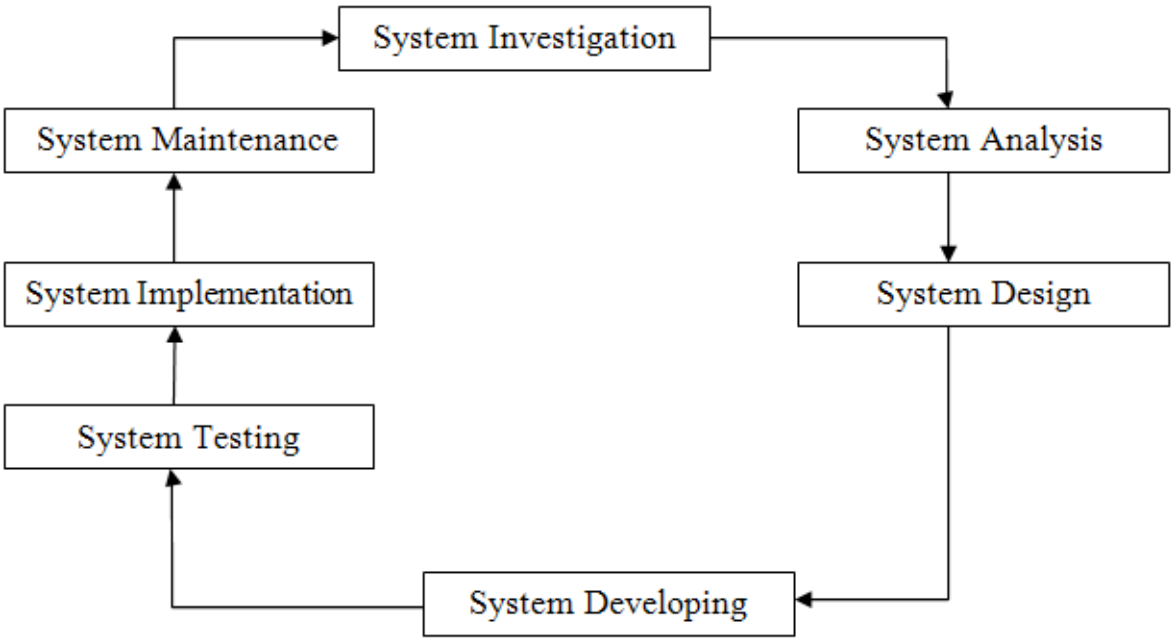


Fig: System Development Life Cycle

System Investigation

In this stage we analyzed Existing system and its defects were found out viewing those defects, we developed the software which addresses those defects. Once the investigation of the system is finish the stage of the system analysis starts

System Analysis

System analysis is the analysis of the problem that the organisation wants to solve with the information system. It consists of defining the problem, identifying its

causes, specifying the solution, and identifying the information requirements that must be met by a system solution.

System Design

After identifying the user requirements, specifications for the hardware, software, people and data resources were developed. The software module that satisfies the functional requirements of the proposed system was also developed. In this phase both logical and physical design of the system was done.

System Development

After design the system the development of the system starts. We develop the system using different modules. Once the different modules are been developed they are integrated the whole system. There modules are developed using different tools and techniques.

System Testing

Various tests were conducted at different times. Some of which are:

- Functional testing
- Module testing
- System testing

Functional and module testing has been conducted after creating function and module respectively to assure correct operation. Functional testing is done for the purpose of checking whether the functions used in system are operational or not. To check whether the modules function appropriately, module testing is done. After the operational system is built up, system testing is done to identify if there are any bugs in the system or not.

System Implementation

After designing the system it was implemented and maintained. It involves installing the new system and changeover from the existing system to the new one.

System Maintenance

For the smooth functioning of the system during its working life, maintenance is necessary. Small errors and system inconsistency may arise in the system which must be debugged and brought into regular operation.

Plans for System Implementation

System implementation involves installation of the new system and switch from the existing system to the new one. System installation is not a major problem, as the system can be installed on a machine supporting certain minimum requirements. The changeover from the old system to a fully computerized new robust system requires in depth training familiarity of the users with the various aspect of the new system and making certain specific adjustments.

System Installation

To install the IMS application on a certain machine, it requires the machine to fulfill minimal requirements and additionally, the backend software My SQL server needs to be installed.

With the successful installation of the IMS application, it will run as any other software present earlier in the machine.

Requirement of User's Training

Any user with basic computer knowledge can operate the system in an efficient and effective way but since this system is completely new software, some general guidelines needs to be provided. This will help them to counter face the difficulties while dealing with the new software. The interface of the system is

quite user friendly such that any user who can operate the operating system like Windows 98 and Windows XP can easily run the system.

2.1 System Requirement

- **Hardware requirement**
  - Pentium 3 or above
  - 512 MB RAM

- **Software Requirement**
  - Microsoft/Linux/Mac Operating System
  - Java Virtual Machine
  - Java Development Kit
  - Java core/unofficial API
    - JTatoo.jar
    - jCalendar.jar

- **User Requirement**
  - Basic Computer knowledge
  - File Browsing Skills

2.2 Context Diagram

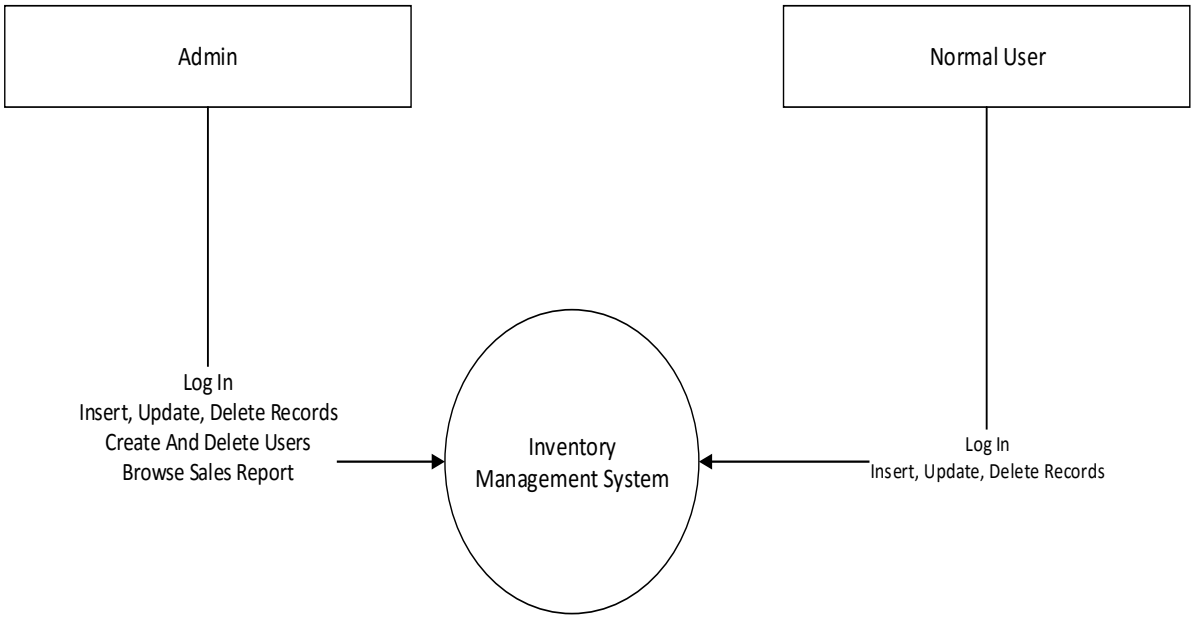
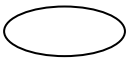
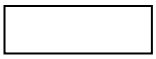
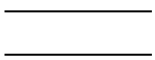


Fig: Context Diagram for IMS

2.3 Level-0 DFD

After the development of the ERD the Data Flow Diagram for the IMS is created. It defines how the data actually flows in the system and the sources of data in the system

TBL 3.1: DFD symbols and their descriptions

Symbols	***Description***
	*Process*
	Data Flow
	*Entity*
	*Database*

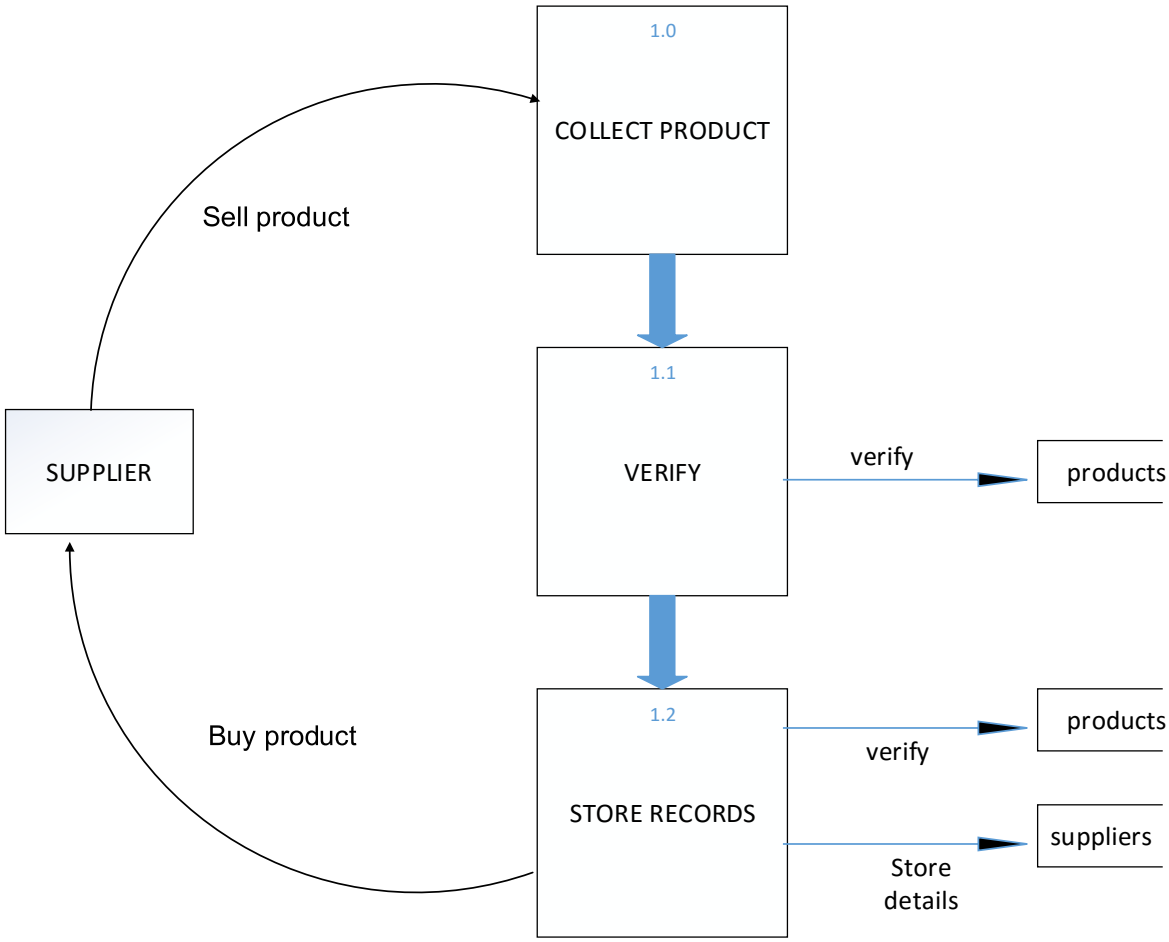


Fig: Collecting products from suppliers

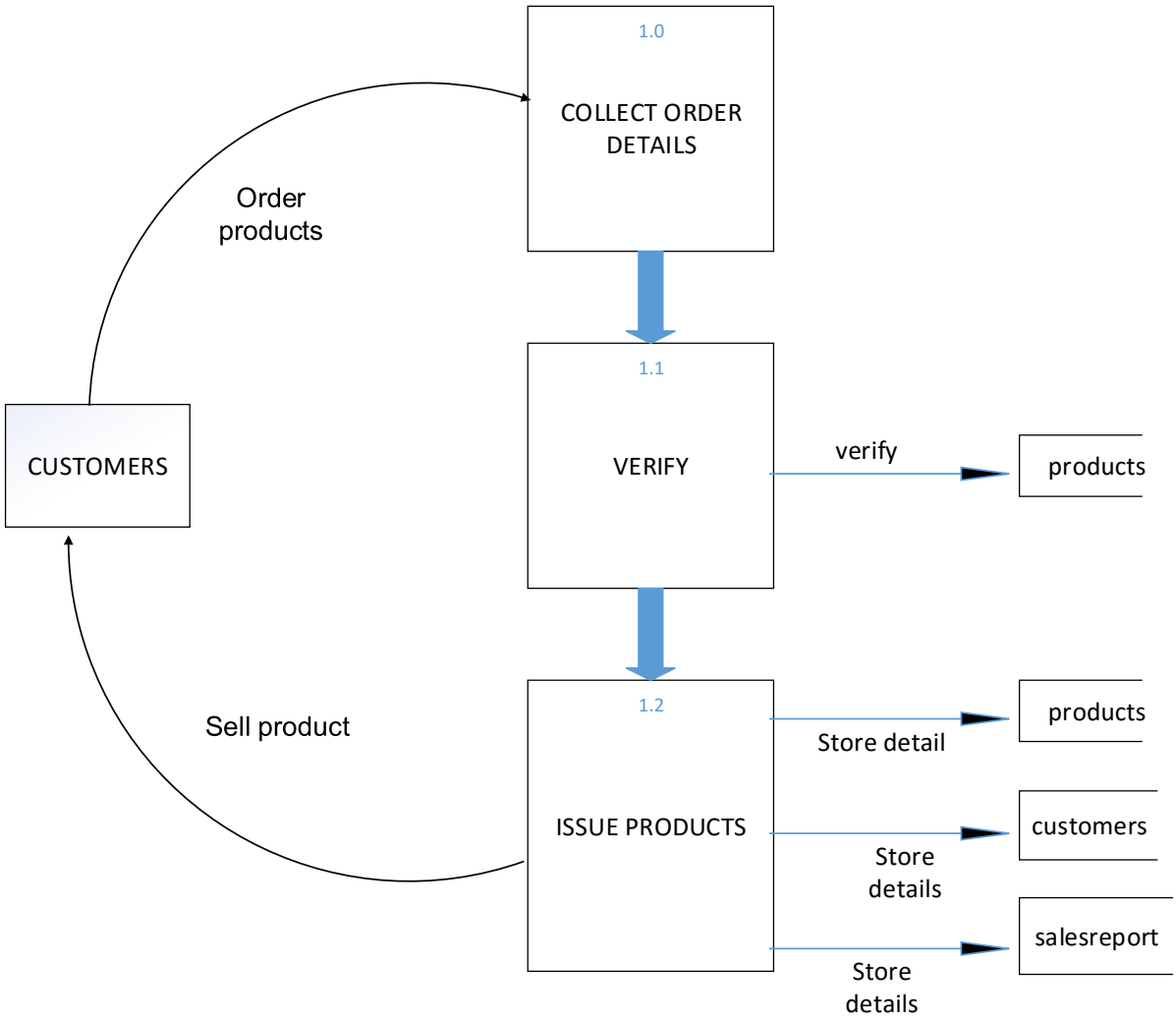


Fig: Selling products to customers

2.3 Use Case Diagram

A use case diagram is a type of behavioral diagram defined by the Unified Modeling Language (UML) whose aim is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. It is used to identify the primary elements (actors) and processes (use cases) that form the system. The use case technique is used in software and systems engineering to capture the functional requirements of a system from the user's perspective. The use case shows how the different actor will be performing what activity within and application the following use case diagram show how different user will be performing the different activity within the application

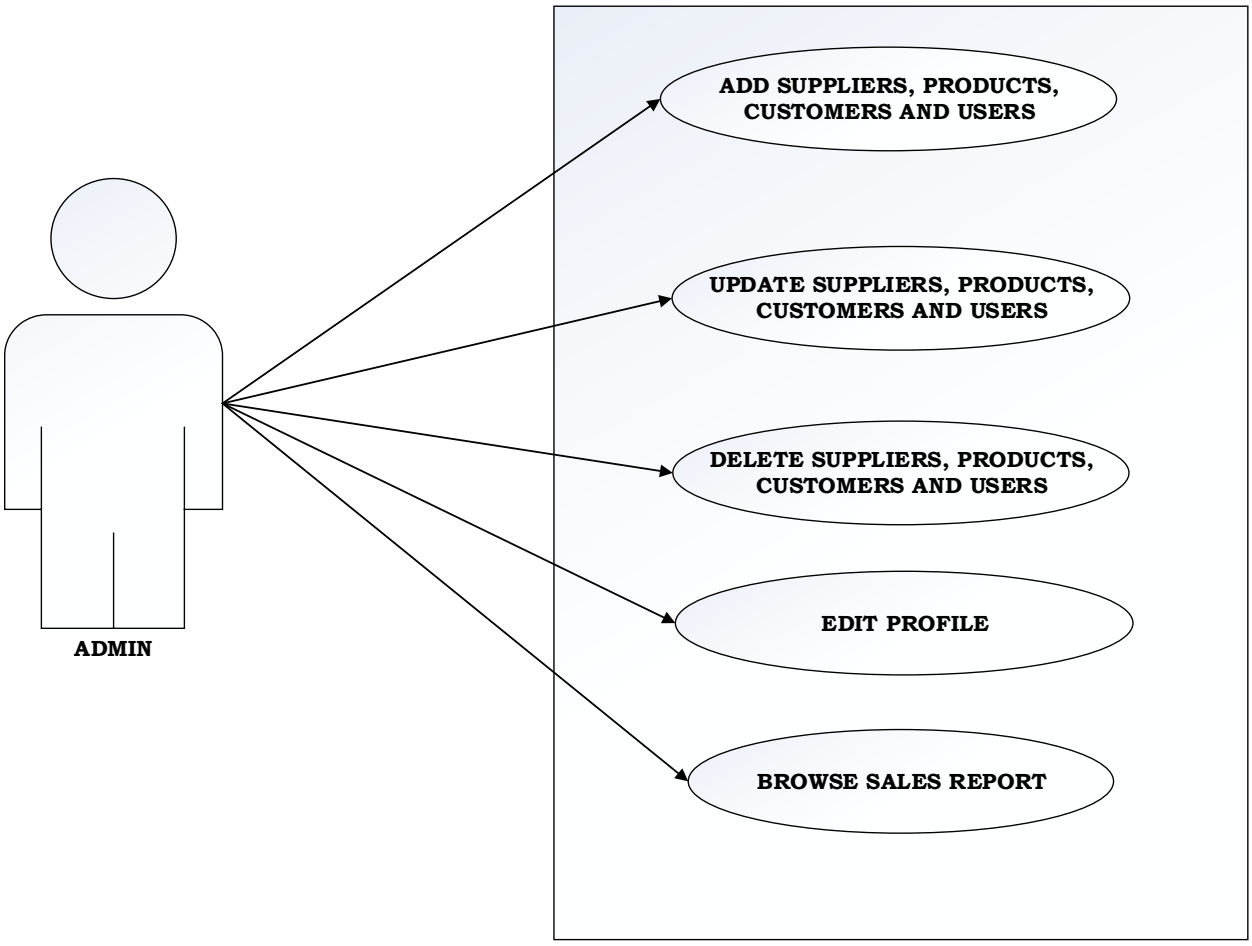


Fig: Use Case Diagram for Admin

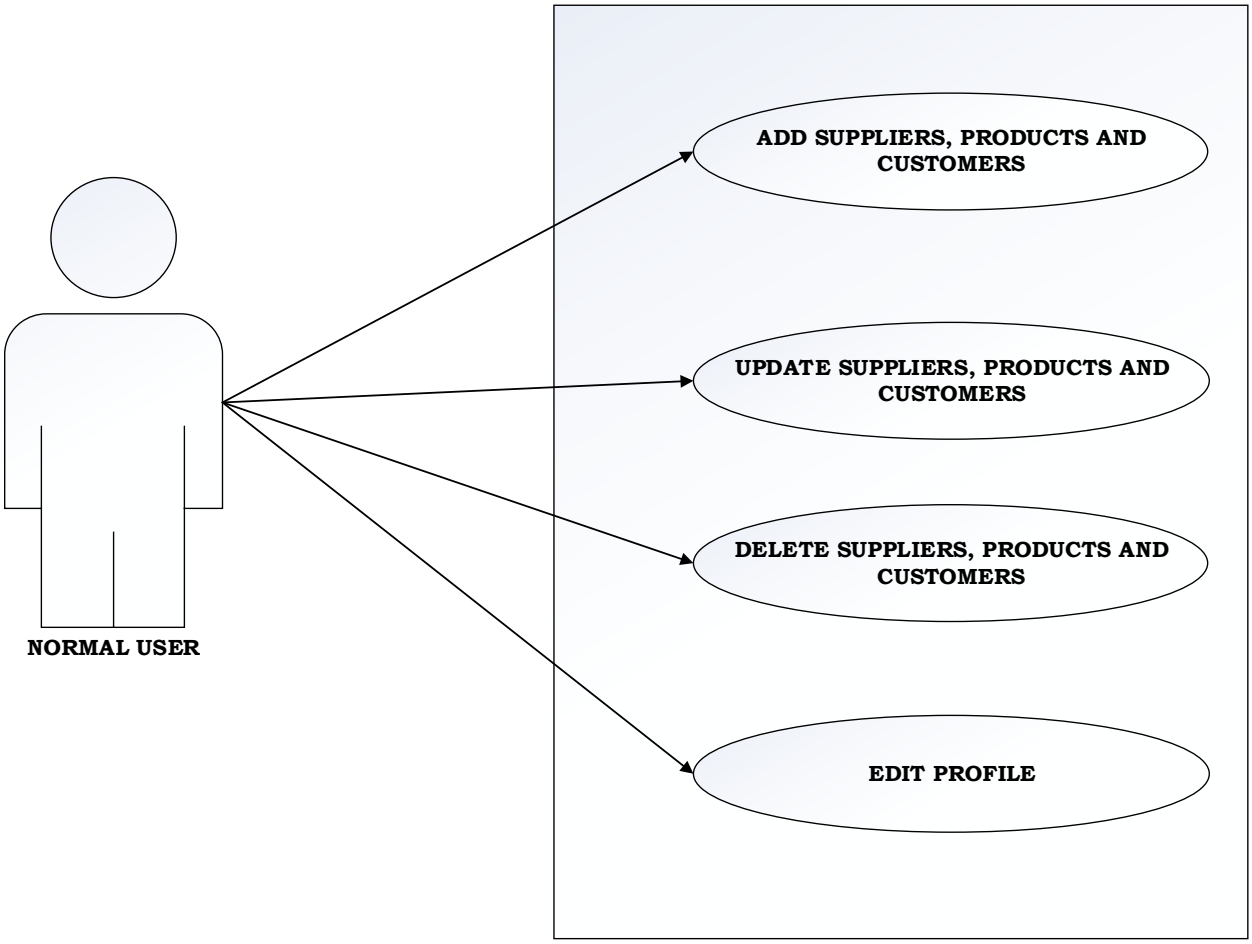


Fig: Use Case Diagram for Normal user

2.4 Sequence Diagram

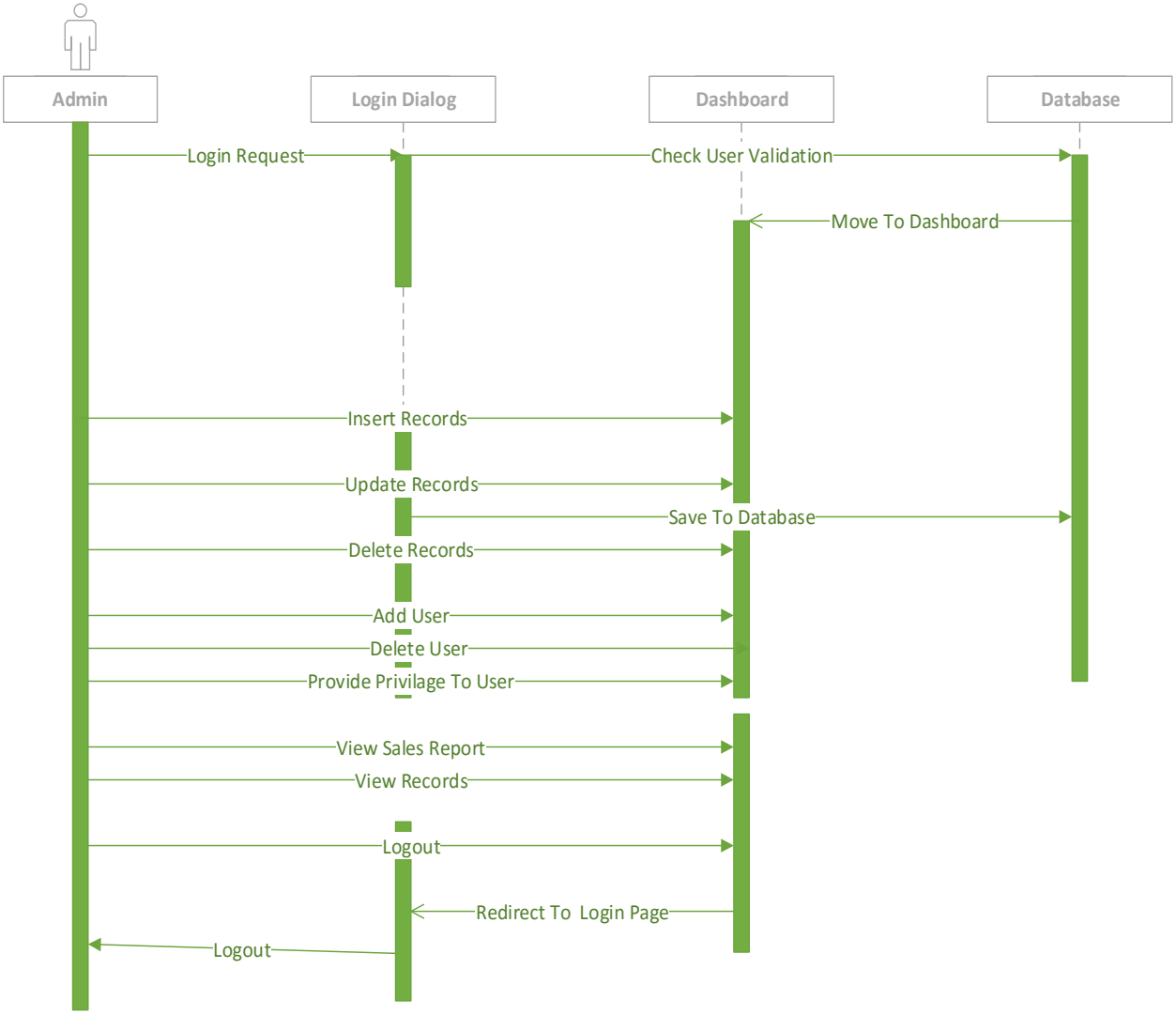


Fig: Sequence diagram for Admin

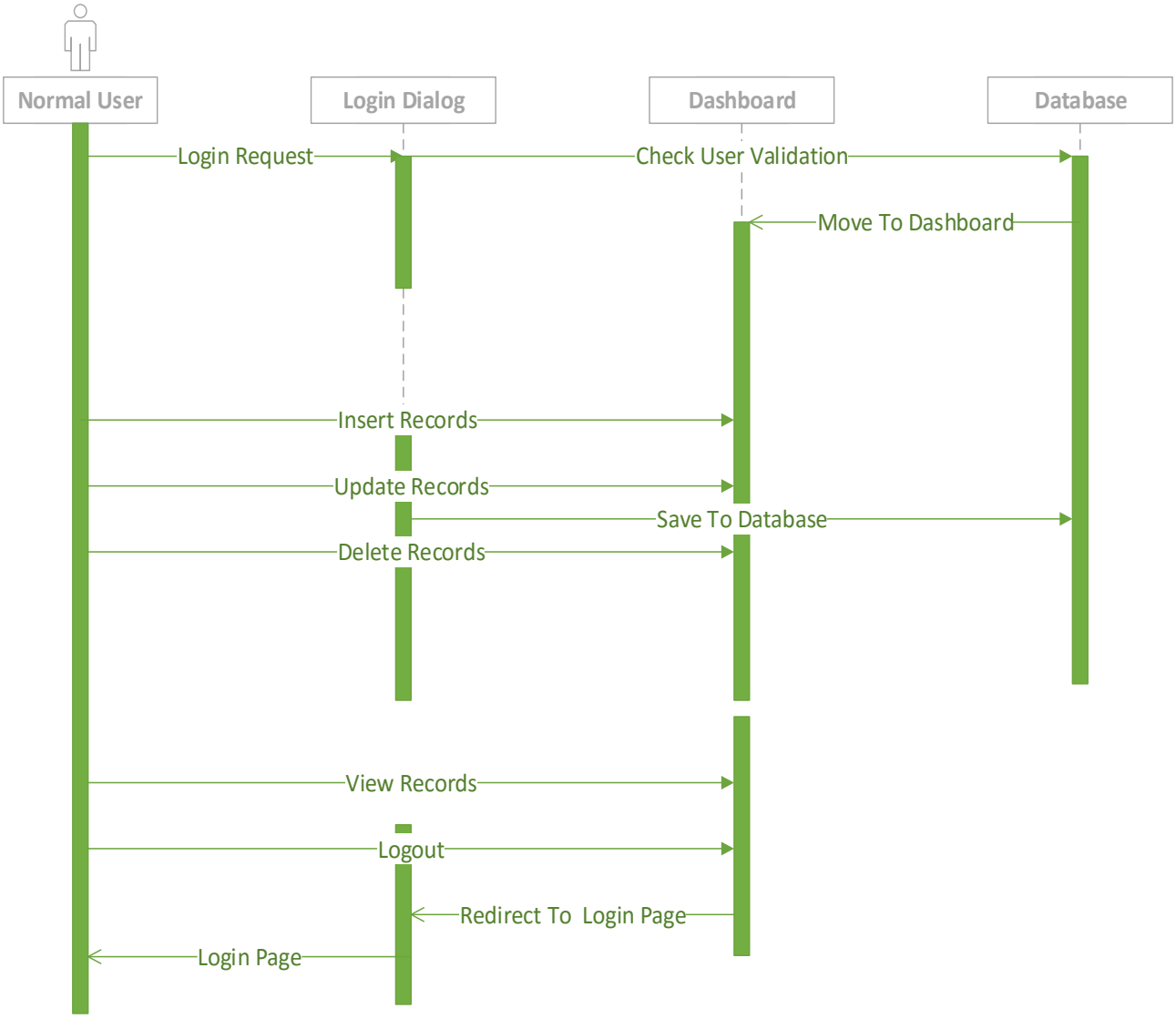


Fig: Sequence diagram for Normal user

2.5 System Flowchart

Flowcharts are a modeling technique. It is typically used to describe the detailed logic of a business process or business rule. A chart that traces the movement of data in a computer system and shows how the data is to be processed. It is also known as bubble chart; system flowchart. The chart is read from top to bottom, and left to right. Flow charts diagrams use a universal set of symbols. Ovals are beginning and ending points. Squares and rectangles are activities or steps. When a series of steps are condensed together, a rectangle with

vertical lines on each end is used to indicate a sub-process. Diamonds are decisions. Circles connect parts of the diagram together.

The Activity Flow Chart for individual Actors of the system are separately created and listed as:

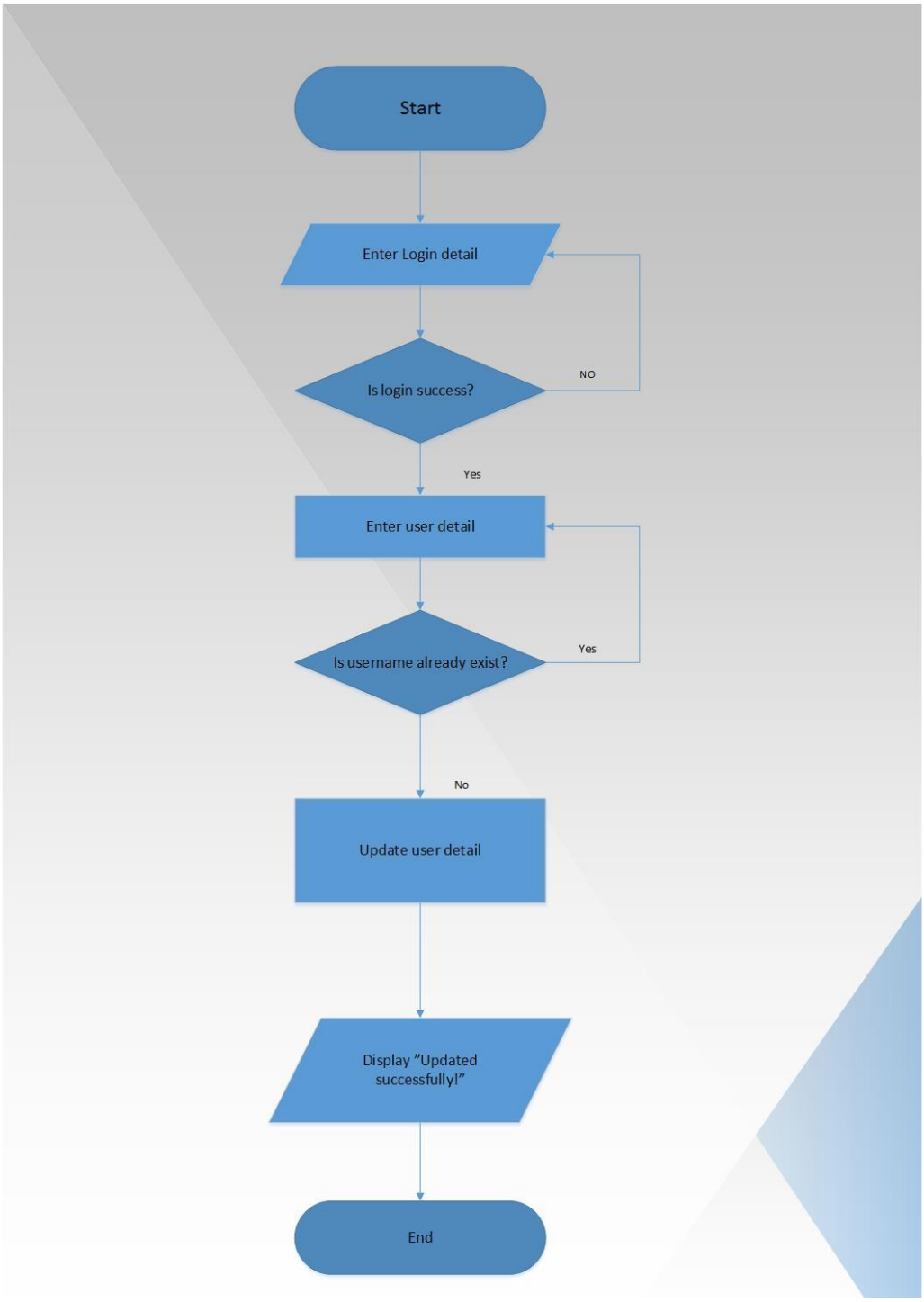


Fig: Change user details

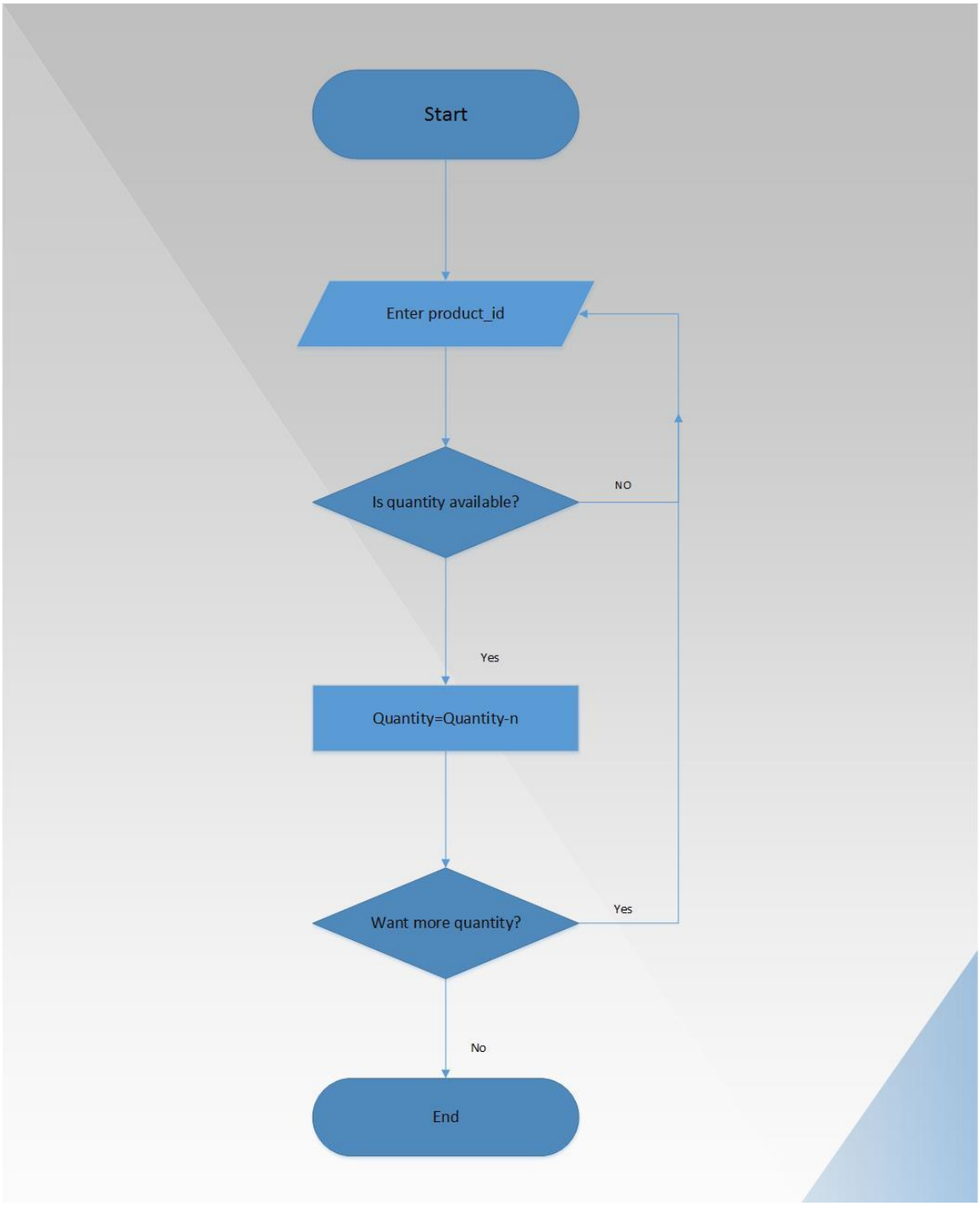


Fig: Selling products to customers

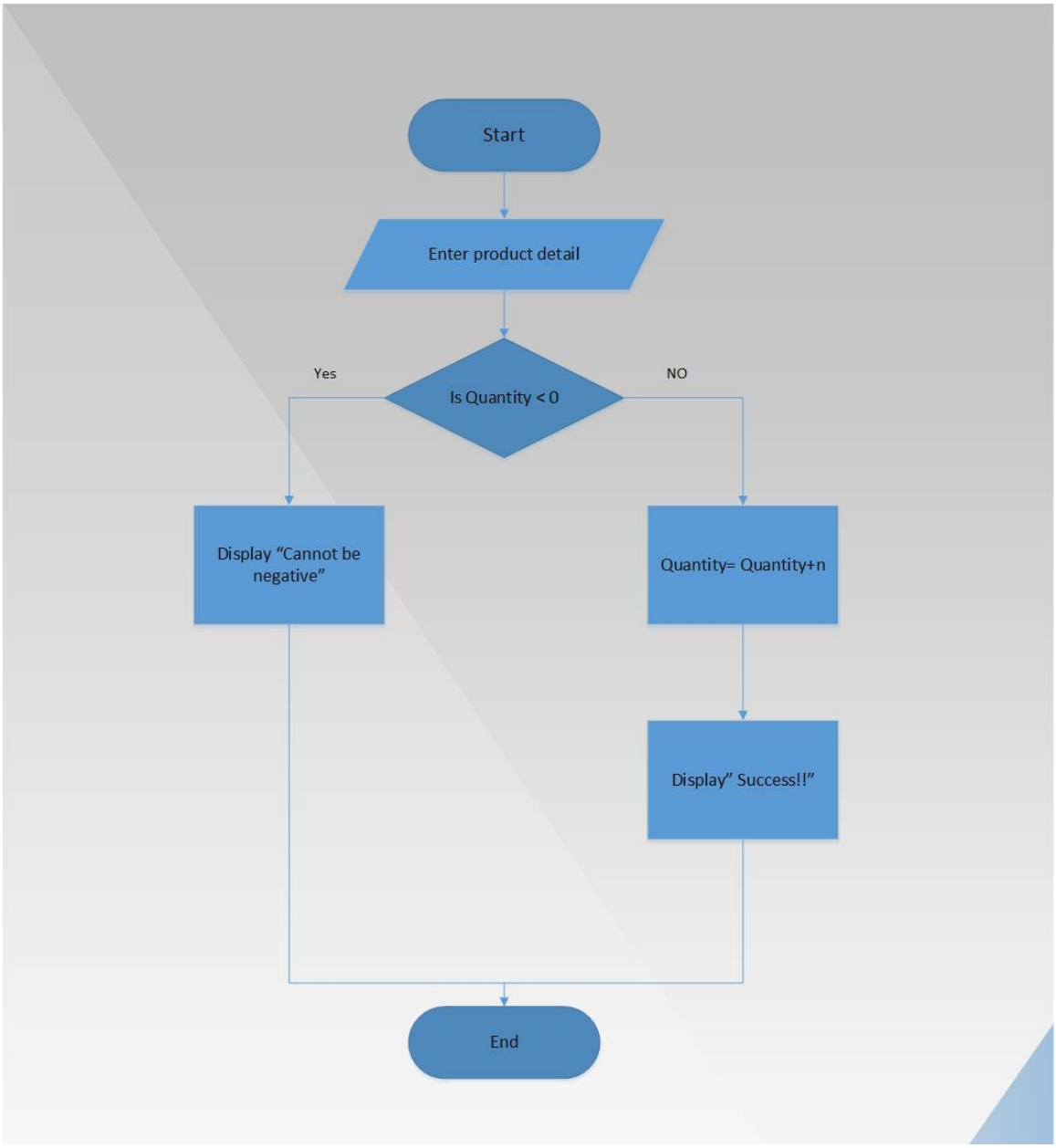


Fig: Receiving products from suppliers

3. System Design

The goal of systems design is to build a system that is effective, reliable, and maintainable. To be effective, the system must satisfy the defined requirements

and constraints. Users who use it to support the organizations business objectives also must accept the system.

A system is reliable if it adequately handles errors, such as input errors, processing errors, hardware failures, or human mistakes. Ideally, all errors can be prevented. A more realistic approach to building a reliable system is to plan for errors, detect them as early as possible, allow for their correction, and prevent them from damaging the system.

A system is maintainable if it is well design, flexible, and developed with future modifications in mind. No matter how well a system is designed and implemented, at some point it will need to be modified. Modifications will be necessary to correct problems, to adapt to changing user requirements, to enhance the system, or to take advantage

3.1 System Architecture

Two-tier architecture has been implemented in the system development. Three-tier is a client–server architecture in which consists of presentation logic and database logic.

Two-tier architecture then will have two processing nodes. Layers refer to a logical grouping of components which may or may not be physically located on one processing node.

IMS application will be a desktop application. It will consist of a database server. These server components can exist in the same sever computer or can be different computers such as Database Server.

Component Architecture

The IMS application will have 2 layers which are as:

1. Presentation Layer
2. Data Access Layer

1. Presentation Layer

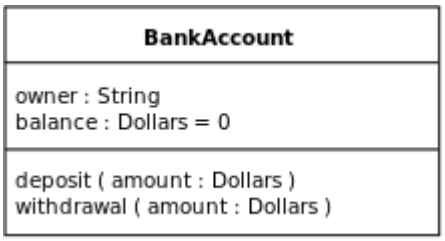
This layer is responsible for presenting the data and information to the users in the format that the user can identify with i.e., in high level language. It is also liable for collecting events, responses and data from the various users. The presentation layer responds to the user request by receiving the required result from application layer and presenting in predetermined format. This layer will be developed by using JAVA.

2. Data Access Layer

This layer is responsible for communicating with the presentation layer to get the data along with other information and with database for storage and retrieval of the data. It receives the information from the presentation layer, process that information as requested and if required, communicates with the database for further processing. The result received from the database is again presented to presentation layer.

3.2 Design Class Diagram

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.[1] The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed.



A class with three sections.

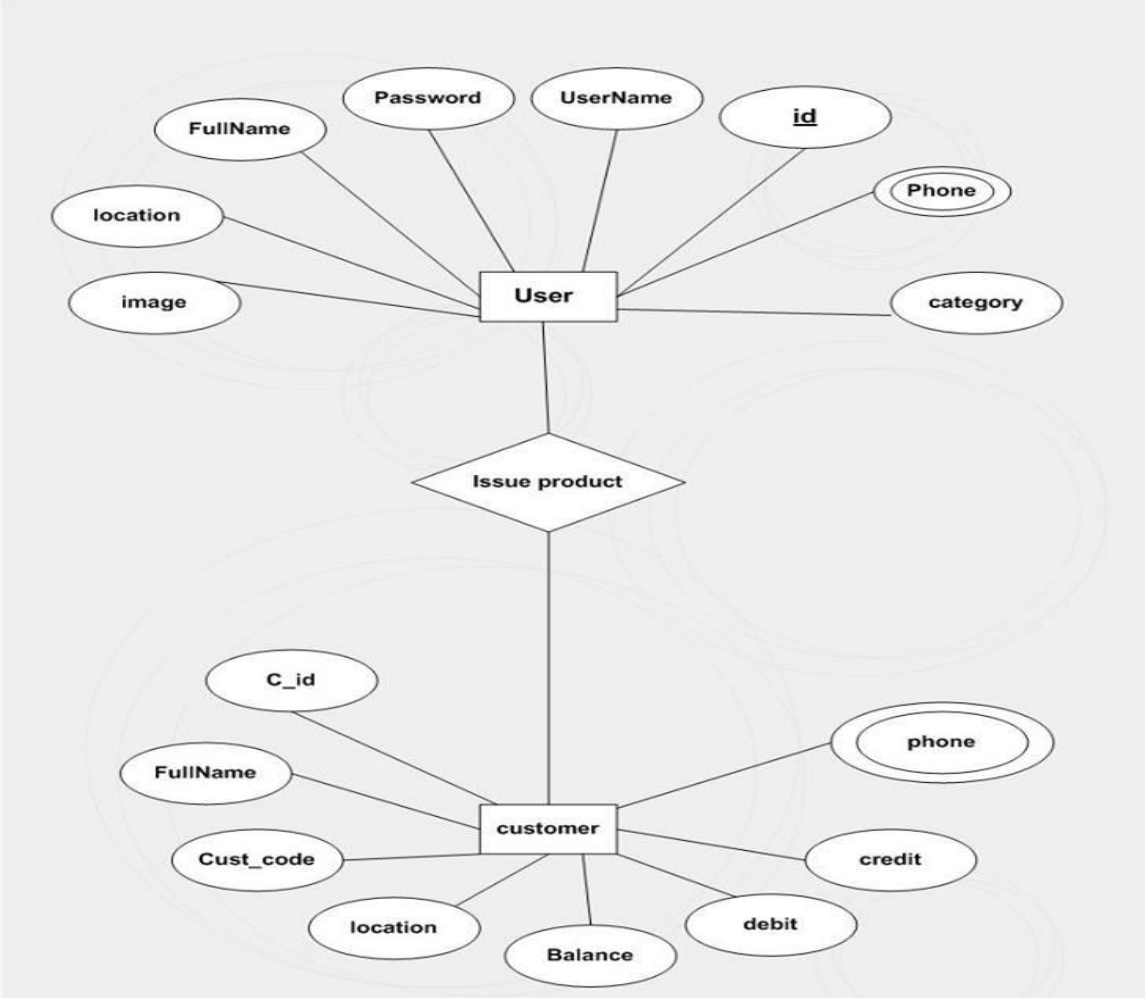
In the diagram, classes are represented with boxes which contain three parts:

- The top part contains the name of the class. It is printed in Bold, centered and the first letter capitalized.
- The middle part contains the attributes of the class. They are left aligned and the first letter is lower case.
- The bottom part gives the methods or operations the class can take or undertake. They are also left aligned and the first letter is lower case.

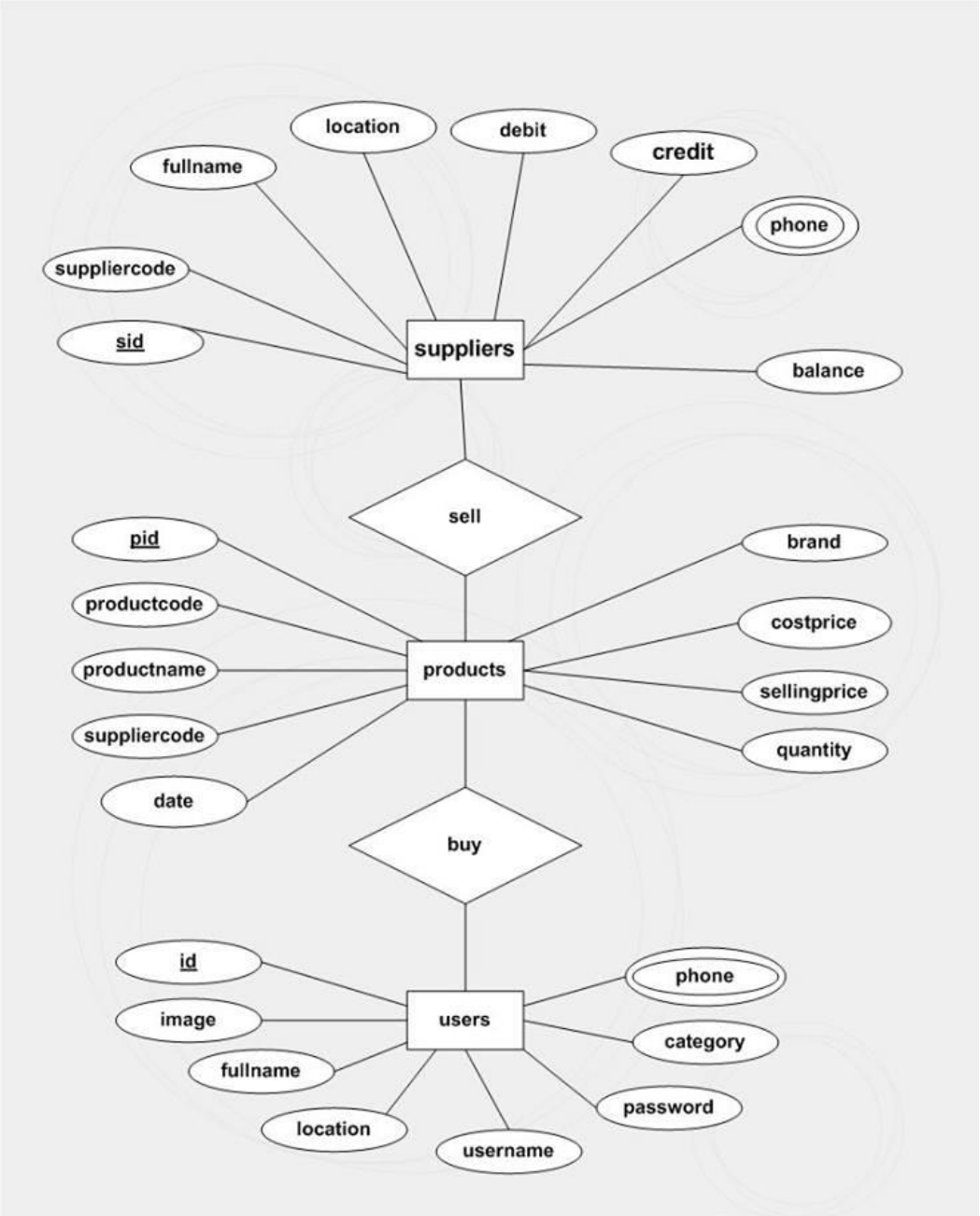
3.3 Entity Relation Diagram (ERD)

For the development of the system, initially all the entities and there interrelation were identified. On the basis of that an ERD was developed. ERD consists of these components:

Rectangles	Represent entity sets
Ellipses	Represent attributes.
Underline Ellipse	Represent Primary Key
Diamonds	Represent relationship sets.
Lines	Link attributes to entity sets, entity sets to relationship sets



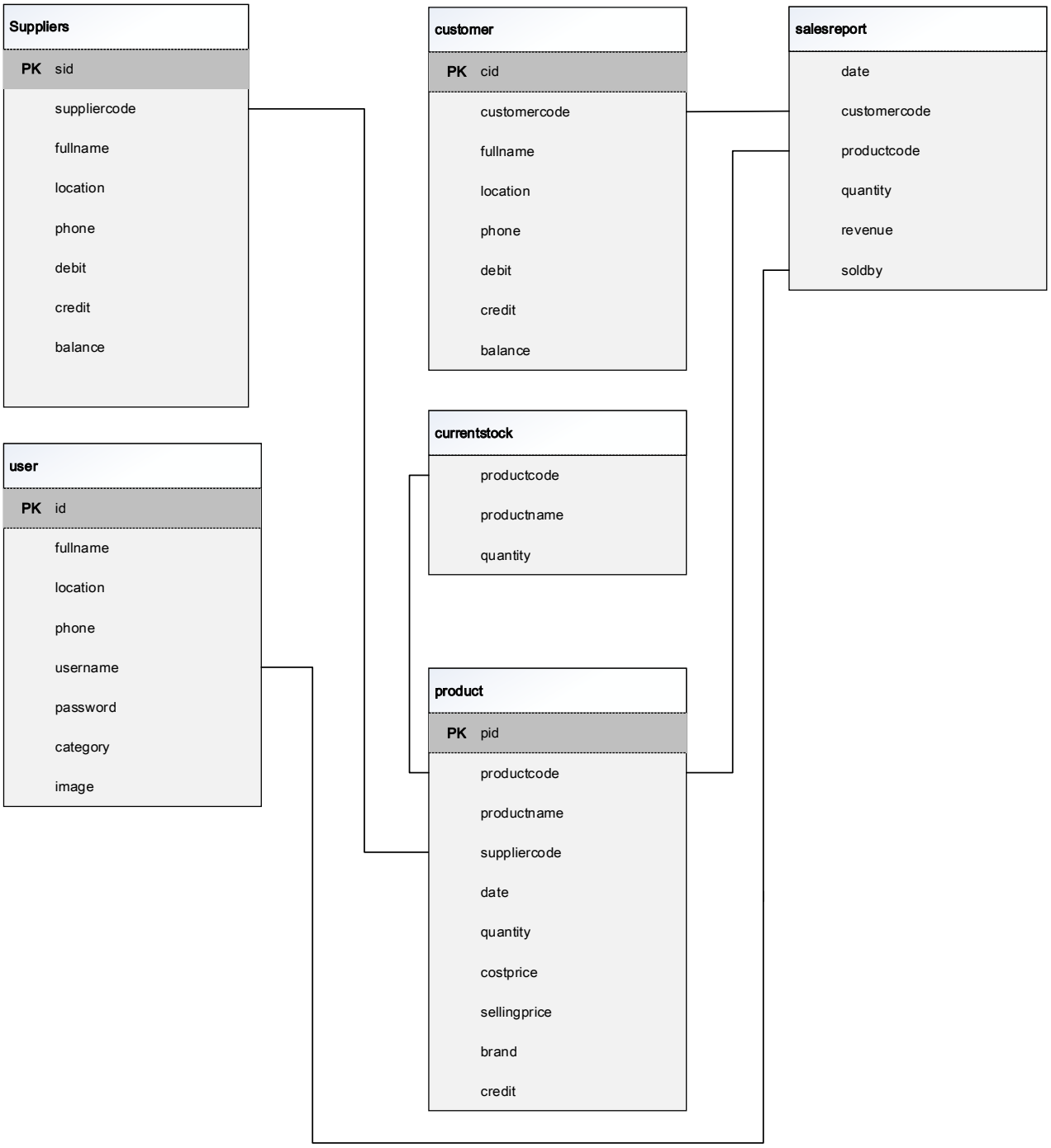
Selling products to customers



Buying product from suppliers

3.3 Schema Diagram

With the reference of DFD and ERD, schema diagram was constructed. The schema diagram actually shows the tables, fields and relation between them. The diagram is constructed before actual coding of database. On the basis of this diagram, database is realized. Schema diagram shows which data will be stored where and what will be the relation between tables and fields.


Suppliers
PK sid
suppliercode
fullname
location
phone
debit
credit
balance
customer
PK cid
customercode
fullname
location
phone
debit
credit
balance
salesreport
date
customercode
productcode
quantity
revenue
soldby
currentstock
productcode
productname
quantity
user
PK id
fullname
location
phone
username
password
category
image
product
PK pid
productcode
productname
suppliercode
date
quantity
costprice
sellingprice
brand
credit

3.4 Database Tables

Suppliers	
sid(PK)	int
suppliercode	varchar
fullname	varchar
location	varchar
debit	double
credit	double
balance	double

customers	
cid(PK)	int
suppliercode	varchar
fullname	varchar
location	varchar
debit	double
credit	double
balance	double

products	
pid(PK)	int
productcode	varchar
productname	varchar
suppliercode	varchar
date	varchar
quantity	int
costprice	double
sellingprice	double
brand	varchar

currentstocks	
productcode	varchar
productname	varchar
quantity	int

salesreport	
date	varchar
customercode	varchar
productcode	varchar
quantity	int
revenue	double
soldby	varchar

4. System Implementation

System implementation involves installation of the new system and switch from the existing system to the new one. System installation is not a major problem, as the system can be installed on a machine supporting certain minimum requirements. The changeover from the old system to a fully computerized new robust system requires in depth training familiarity of the users with the various aspect of the new system and making certain specific adjustments.

To install the IMS application on a certain machine, it requires the machine to fulfill minimal requirements and additionally, the backend software MySQL server needs to be installed and JVM. With the successful installation of the IMS application, it will run as any other software present earlier in the machine.

4.1 Coding

```
public static void main(String[] args) {
        try{
            Properties p=new Properties();
            p.put("logoString","IMS");
            HiFiLookAndFeel.setCurrentTheme(p);
            UIManager.setLookAndFeel("com.jtattoo.plaf.hifi.HiFiLookAndFeel");
        }catch(ClassNotFoundException | InstantiationException |
IllegalAccessException | UnsupportedLookAndFeelException e){
            e.printStackTrace();
        }
        LoginDialog ld=new LoginDialog();
    ld.setLocationRelativeTo(null);
    ld.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    ld.setVisible(true);
  }
```

4.2 Testing Strategies

Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and code generation. Once source code has been generated, software must be tested to uncover and correct as many errors as possible before delivery to customer. Our goal is to design a series of test case that have a likelihood of finding errors. So, testing technique provide systematic guidance for designing tests that:

- ✓ Excise the internal logic of software component.
- ✓ Excise the input and output domains of the program to uncover errors in program function, behavior and performance.

A rich variety of test case design methods has evolved for software. This method provides the developer with a systematic approach to testing.

Any software can be tested in two ways:

1) White box testing:
2) Black box testing

1. White box testing

White box testing is a test case design method that uses the control structure of the procedural design to derive test cases. White-box testing of software is predicated on close examination of procedural details. Logical paths through the software are tested by providing test cases that exercise specific sets of conditions and loops. White box testes focus on the program control structure.

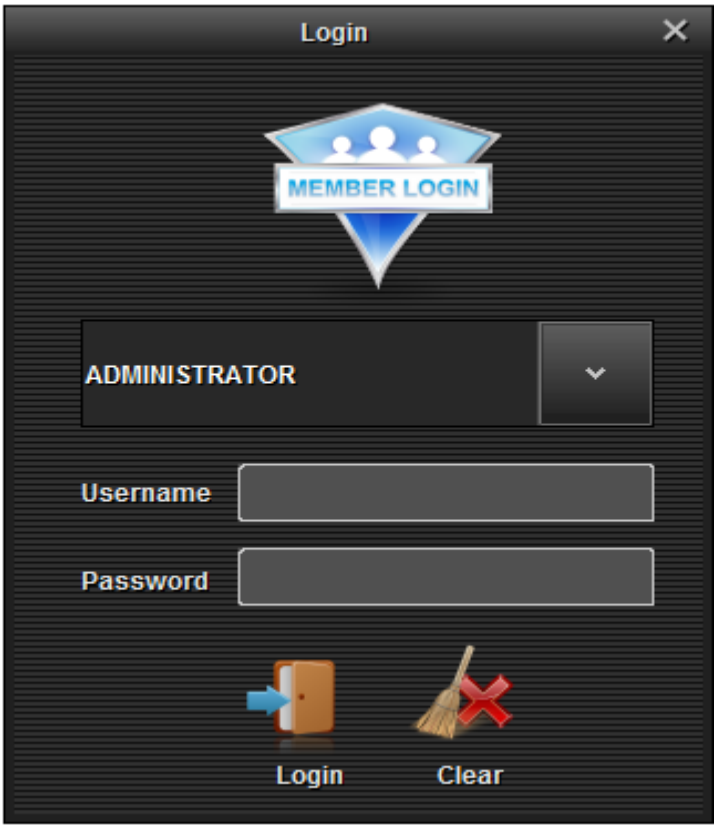
2. Black box testing

Black box testing focuses on the functional requirements of the software. It is user acceptance testing that has the objective of selling the user on the validity and the reliability of the system.

Black box testing is not an alternative to white box testing rather it is a complementary approach that is likely to uncover different class of error than white box method.

4.3 Execution Snapshot

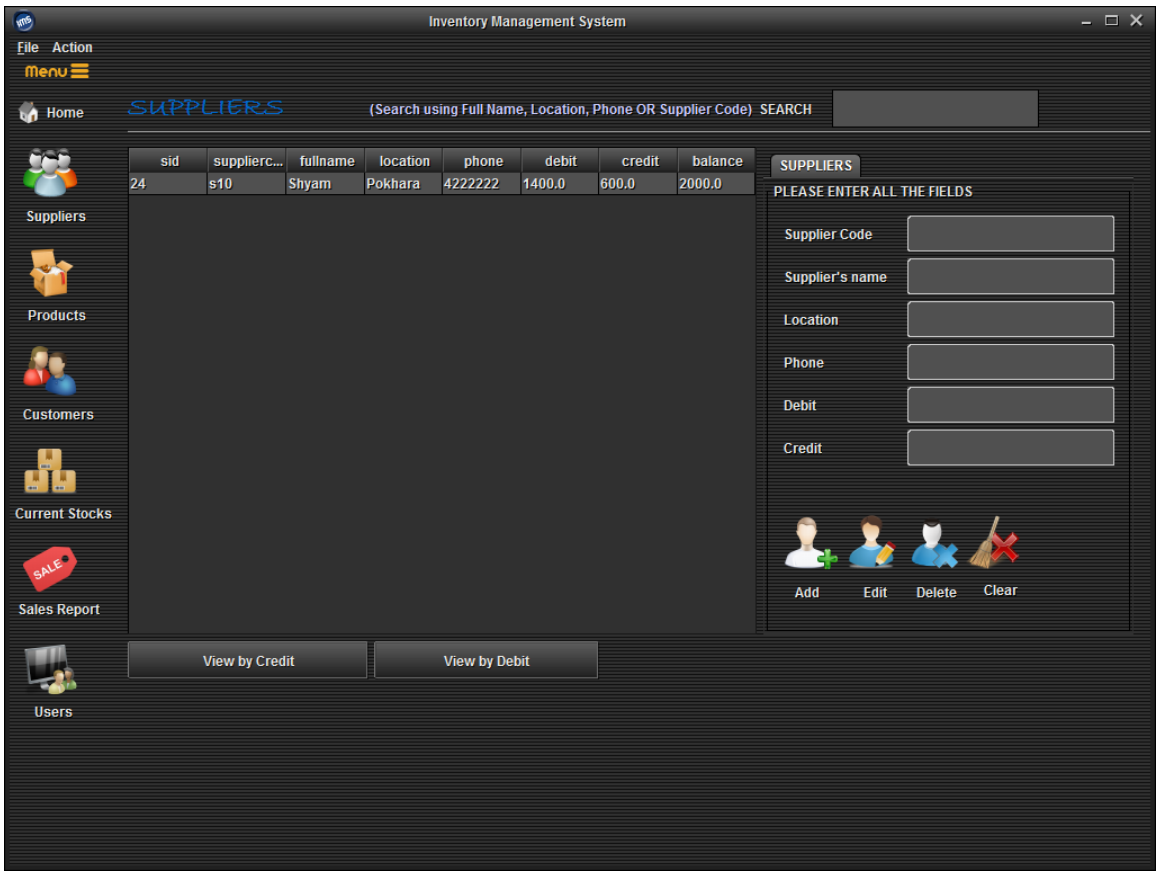
Login Dialog:



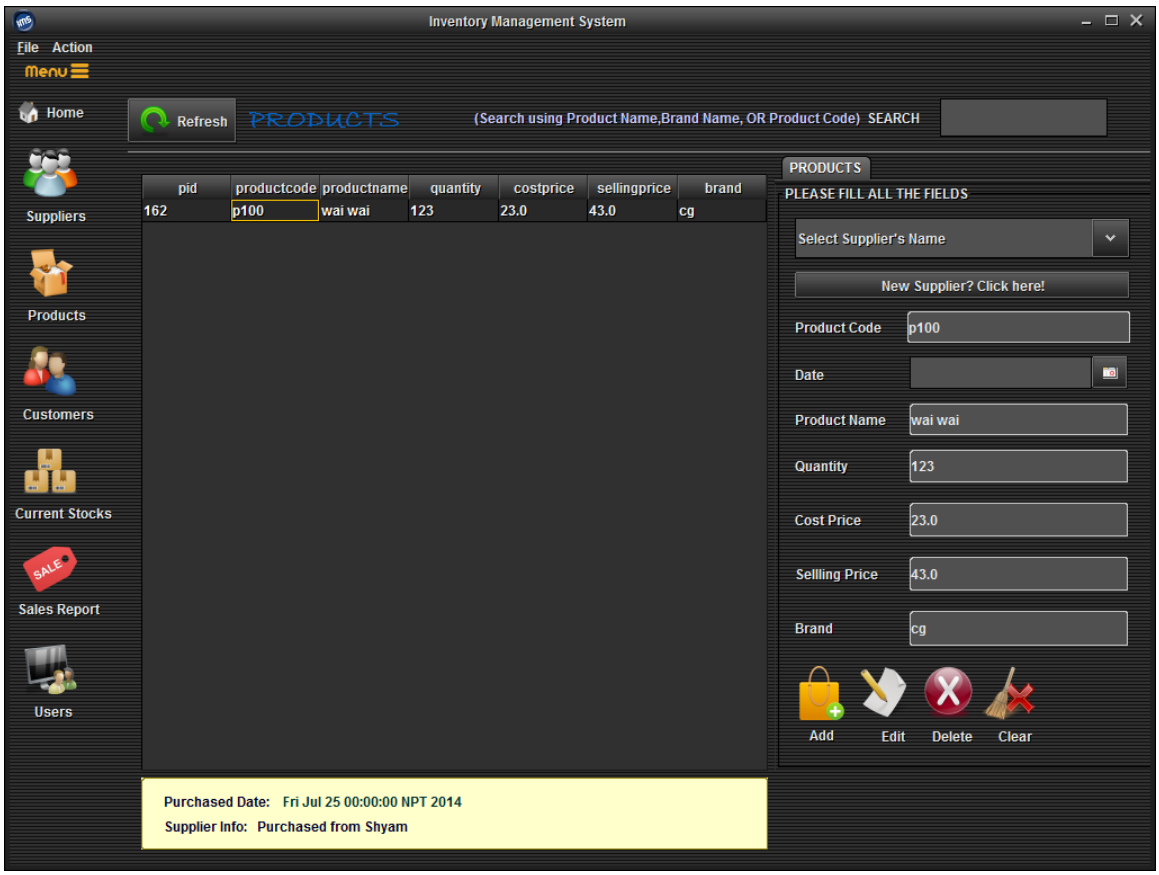
Dashboard:


Inventory Management System
File Action
Menu
Home
Suppliers
Products
Customers
Current Stocks
Sales Report
Users
WELCOME
TO
NCCS
Store

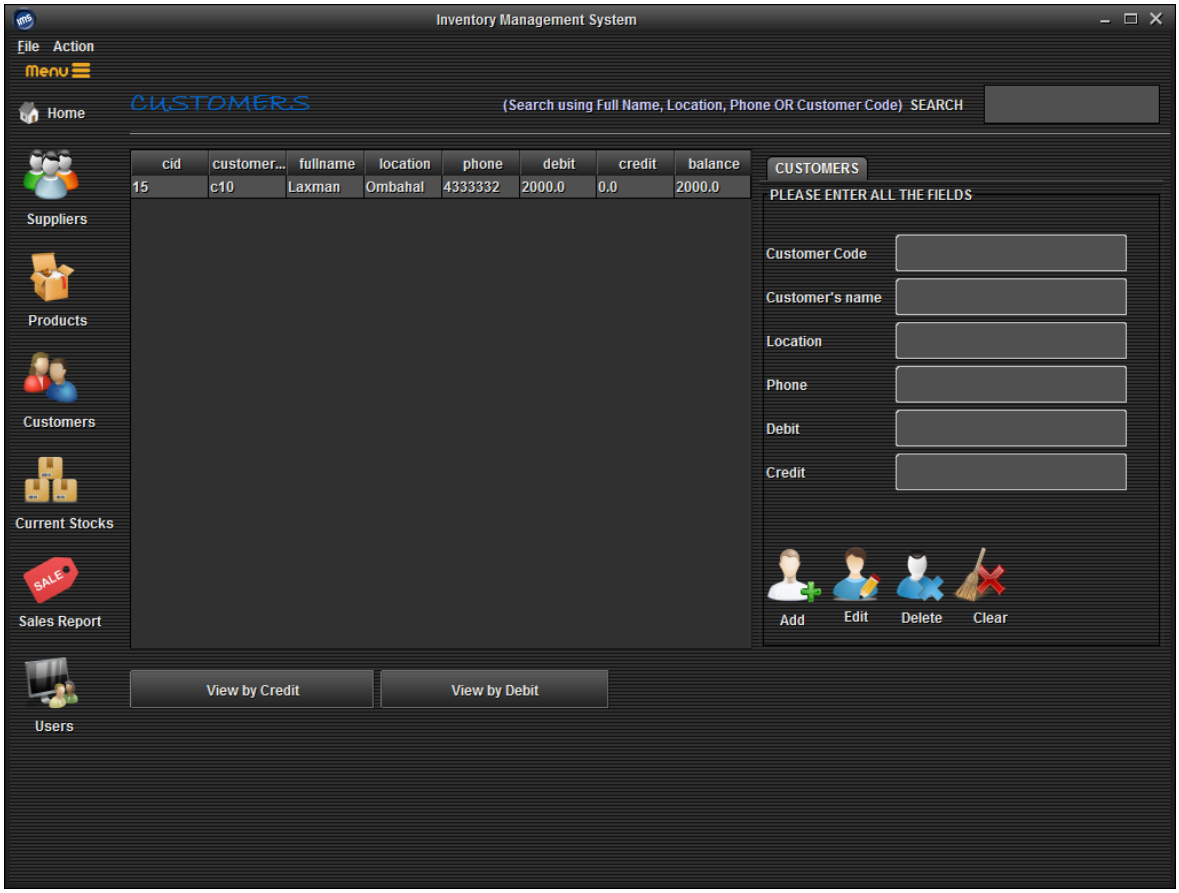
Suppliers:


Inventory Management System
File Action
Menu
Home
SUPPLIERS
(Search using Full Name, Location, Phone OR Supplier Code) SEARCH
sid supplierc... fullname location phone debit credit balance
24 s10 Shyam Pokhara 4222222 1400.0 600.0 2000.0
SUPPLIERS
PLEASE ENTER ALL THE FIELDS
Supplier Code
Supplier's name
Location
Phone
Debit
Credit
Add
Edit
Delete
Clear
Suppliers
Products
Customers
Current Stocks
Sales Report
Users
View by Credit
View by Debit

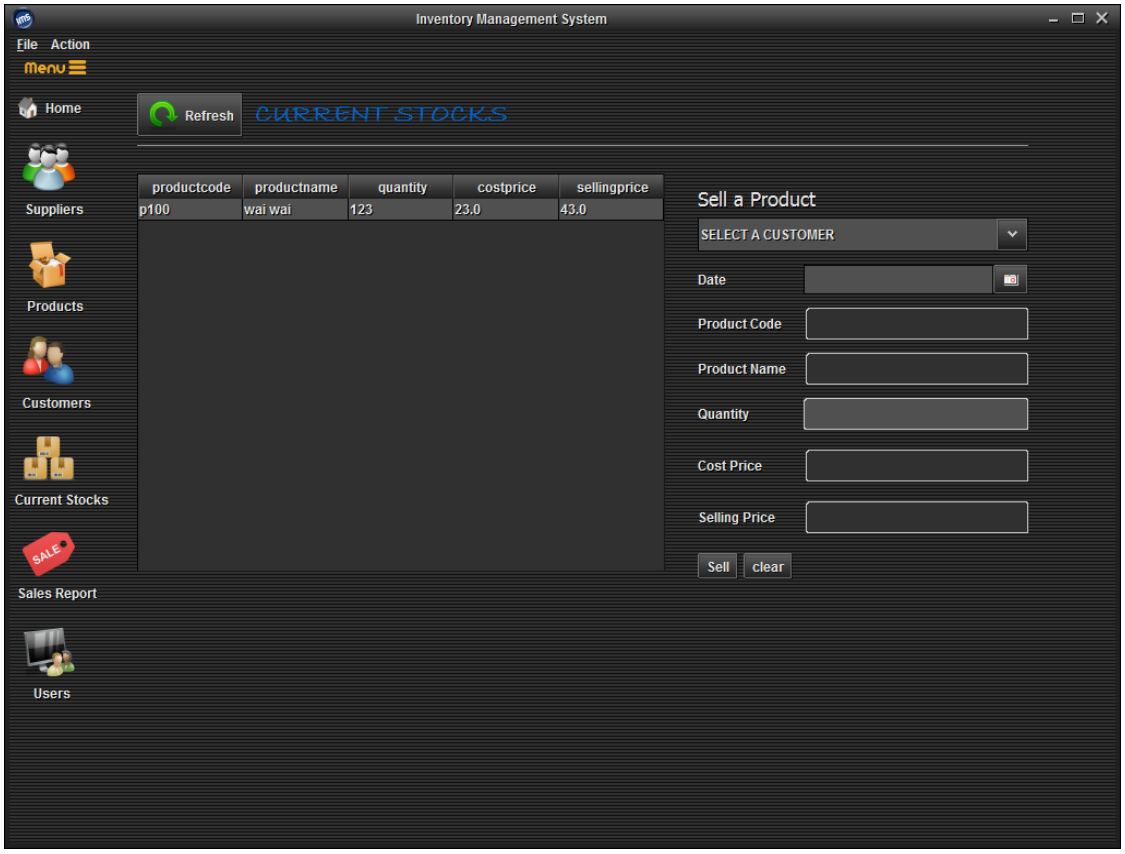
Products:


Inventory Management System
File Action
Menu
Home
Refresh
PRODUCTS
(Search using Product Name,Brand Name, OR Product Code) SEARCH
Suppliers
Products
Customers
Current Stocks
Sales Report
Users
pid
productcode
productname
quantity
costprice
sellingprice
brand
162
p100
wai wai
123
23.0
43.0
cg
PRODUCTS
PLEASE FILL ALL THE FIELDS
Select Supplier's Name
New Supplier? Click here!
Product Code
p100
Date
Product Name
wai wai
Quantity
123
Cost Price
23.0
Sellling Price
43.0
Brand
cg
Add
Edit
Delete
Clear
Purchased Date: Fri Jul 25 00:00:00 NPT 2014
Supplier Info: Purchased from Shyam

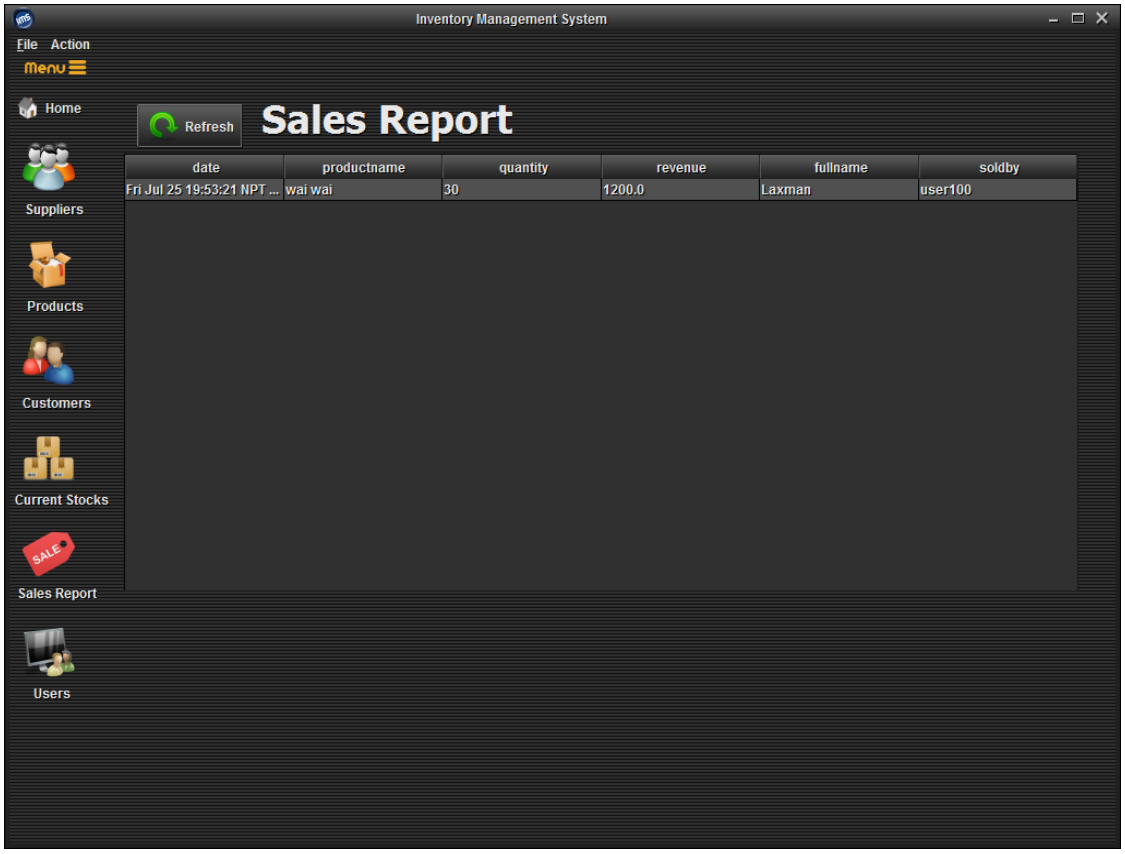
Customers:


Inventory Management System
File Action
Menu
Home
CUSTOMERS
(Search using Full Name, Location, Phone OR Customer Code) SEARCH
Suppliers
Products
Customers
Current Stocks
Sales Report
Users
cid customer... fullname location phone debit credit balance
15 c10 Laxman Ombahal 4333332 2000.0 0.0 2000.0
CUSTOMERS
PLEASE ENTER ALL THE FIELDS
Customer Code
Customer's name
Location
Phone
Debit
Credit
Add
Edit
Delete
Clear
View by Credit
View by Debit

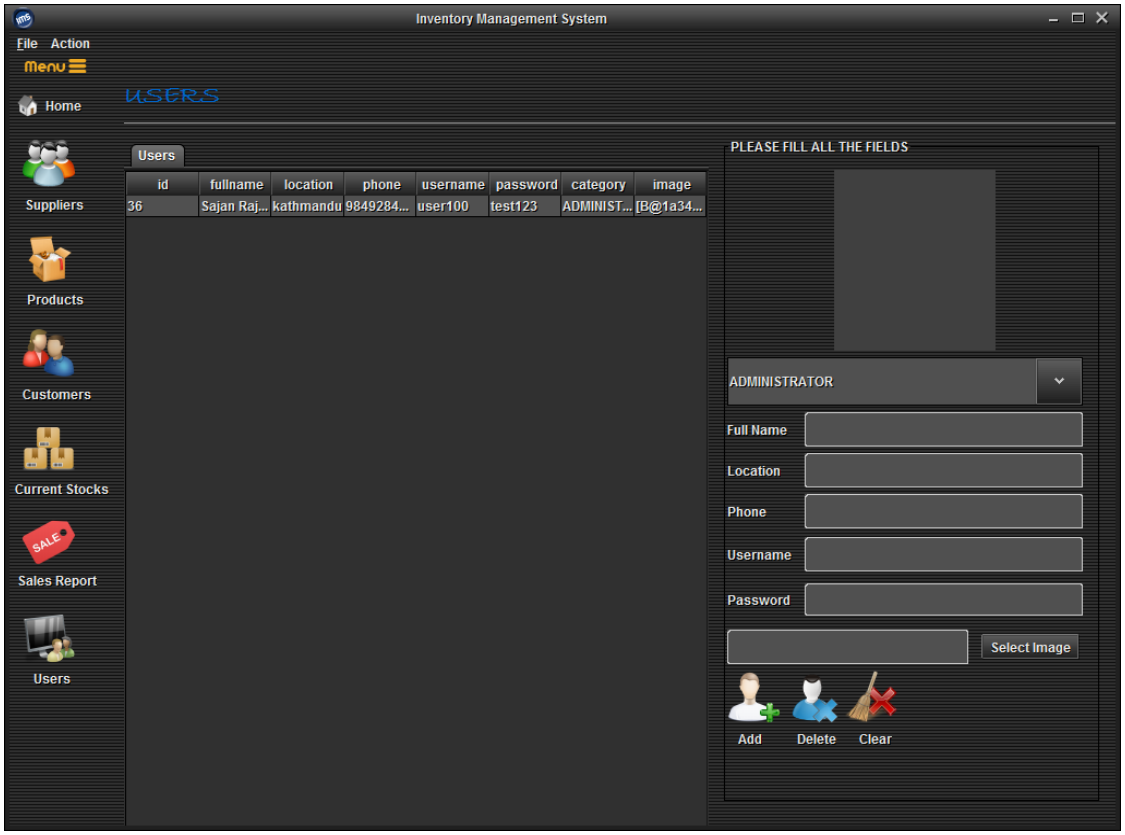
Current Stocks:



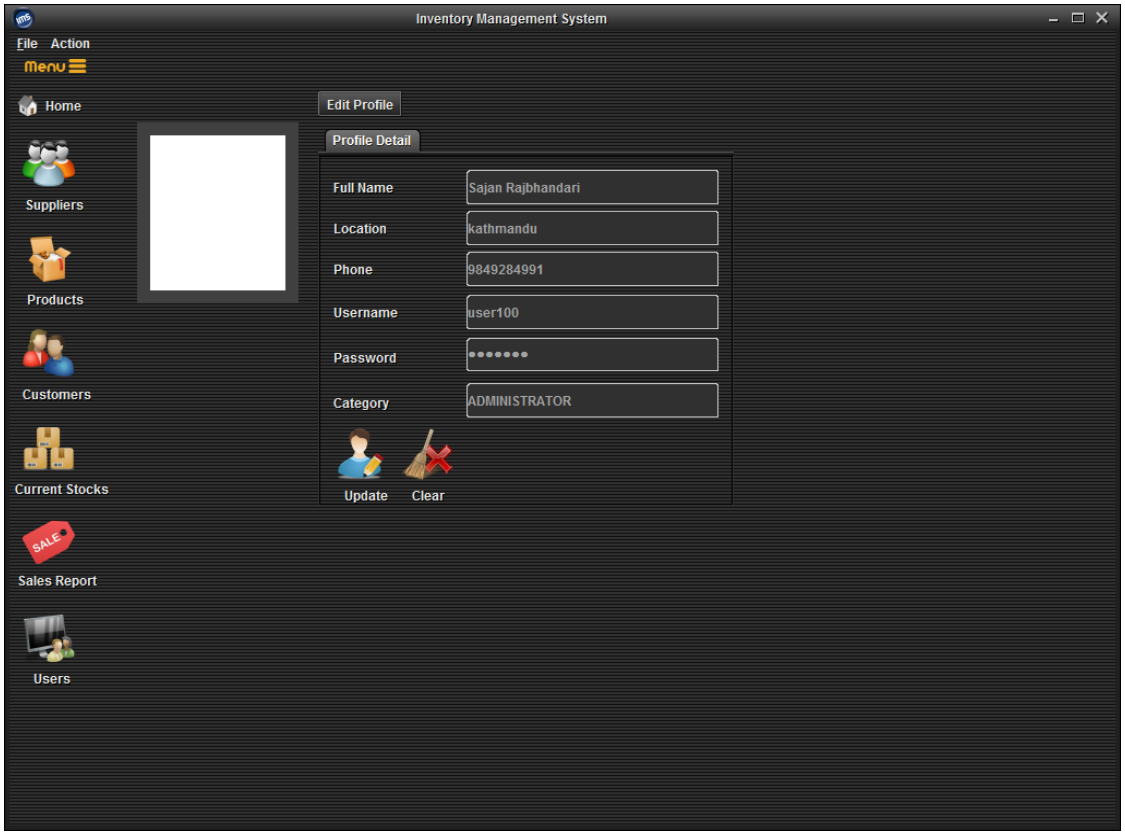
Sales Report:



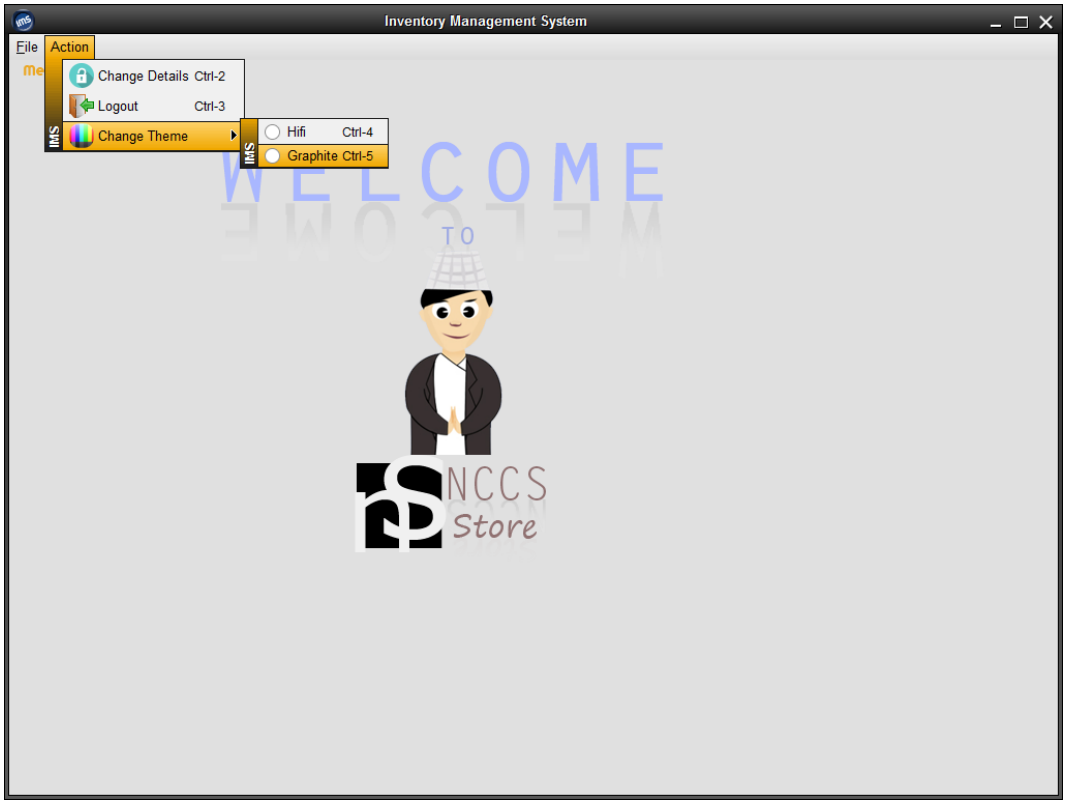
Users:


Inventory Management System
File Action
Menu
Home
USERS
Suppliers
Products
Customers
Current Stocks
Sales Report
Users
Users
id fullname location phone username password category image
36 Sajan Raj... kathmandu 9849284... user100 test123 ADMINIST... [B@1a34...
PLEASE FILL ALL THE FIELDS
ADMINISTRATOR
Full Name
Location
Phone
Username
Password
Select Image
Add
Delete
Clear

Change Details:



Change theme:



5. Conclusion

The Inventory Management System is developed and designed for recording and managing the inventory of an organization. It can also be used for different institution with fewer modifications as per requirement. The system can be easily updated as the other institutional requirement may not be integrated on our project. After the continuous effort, testing and debugging the current system is ready to be implemented in an organization.

The system development project has developed the ability on us to implement the theoretical knowledge we have gained during BIM study in the real life scenario.

Some of the lesson that we had learned from the project are:-

- Sharpen the knowledge of working cooperatively in working organizational environment and work place.
- Know the value of time and discipline.
- Work in group and make group decision.
- Learnt communication skill, leadership quality and to make good public relation.

6. Reference and Bibliography

- Jimmy Wales, online encyclopedia Wikipedia, http://www.wikipedia.org
- James Gosling, Java (programming language) , http://www.java.com
- Names Allaire, Netbeans- Fully-featured Java IDE, http:// www.netbeans.org
- James Gosling, Welcome to JavaWorld.com: how-to features and columns by Java experts; news; Java applets; sample code; tips, http:// www.javaworld.com
- Pressman, Roger S. “Software Engineering A Practitioner’s Approach”
- John Osborn, JavaBeans: Developing Component Software in Java
- Doug Lea, Concurrent Programming in Java: Design Principles and Patterns, Addison-Wesley , November, 1996