GENERAL STORE MANAGEMENT SYSTEM

By

P.PRABHAKARAN (511913621323)

Of

PRIYADARSHINI ENGINEERING COLLEGE, VANIYAMBADI.

A PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In Partial fulfilment of the requirements
For the award of the degree

Of

MASTER OF COMPUTER APPLICATIONS

ANNA UNIVERSITY
CHENNAI – 600 025.

JUNE - 2016

CERTIFICATE OF EVALUATION

COLLEGE NAME **:** 5119, PRIYADARSHINI ENGINEERING COLLEGE VANIYAMBADI.

BRANCH & SEMESTER : MCA, 6th SEMESTER

SUB.CODE & NAME **:** MC7611, PROJECT WORK

MONTH & YEAR **:** JUNE 2016

Name of the Student who have Done the Project	Register Number	Title of the Project	Name of the Supervisor With Designation
P.PRABHAKARAN	**511913621323**	**GENERAL STORE MANAGEMENT SYSTEM**	**Mr.N.SARAVANAN, MCA,** **Assistant Professor,** **Department of MCA**

The reports of the project work submitted of the fulfillment of Master of Computer Applications of Anna University was evaluated and confirmed to be reports of the work done by the above student.

Signature of Internal Guide

Mr.N.SARAVANAN MCA .,
Assistant. Professor,
Priyadarshini Engg. College,
Vaniyambadi – 635 751.

Signature of HOD

Mr.S.VIJAYAKUMAR MCA,
M.Phil.Ph.D
Associate Prof & HOD
Priyadarshini Engg. College,
Vaniyambadi – 635 751.

ANNA UNIVERSITY

CHENNAI

PRIYADARSHINI ENGINEERING COLLEGE

VANIYAMBADI

DEPARTMENT OF COMPUTER APPLICATIONS

BONAFIDE CERTIFICATE

Certified to Main project report titled "**GENERAL STORE MANAGEMENT SYSTEM"** is the bonafide work of Mr.P.PRABHAKARAN (Reg No: 511913621323) who carried out the research under my supervision. Certified further, that do the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SUPERVISOR **HEAD OF THE DEPARTMENT**

Mr.N.SARAVANAN, MCA, Mr.S.VIJAYAKUMAR, MCA, M.Phil Ph.D.,
Assistant Professor, Associate Prof & HOD Priyadarshini

Engg. College, Priyadarshini Engg.College, Vaniyambadi – 635

751. Vaniyambadi – 635 751. Submitted to project and

Viva Examination held on ______________ at

Priyadarshini engineering college vaniyambadi-635751.

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

It is obvious that the development of this project needs the support of many people. First of all I wish to express my deepest love and gratitude to my dear parent's brother and Friends for their unwavering support. I would like to extend my sincere thanks to our principal **Dr. P. NATARAJAN, M.E., Ph.D**., for exposing me to do this project.

I am grateful to **Mr. S. VIJAYAKUMAR, MCA, M. Phil**.,**(Ph.D)** Associate. Prof & HOD of Computer Applications who narrowed me in my journey of this project with valuable support and encouragement.

I thank **Mr. N.SARAVANAN, MCA.**, Asst. Prof. Department of MCA, who spent numerous hours guiding me with patience throughout the tenure of this project. I would like to thanks all faculties of MCA department for the support and encouragement for the last three years.

(P.PRABHAKARAN)

ABSTRACT

The General Store Management System is a project that deals with general store automation and it includes both purchasing a selling of items. This project is designed with a goal to making the existing system more informative, reliable, fast and easier. There are many reasons for the starting of the project because in the selling of items through the manual system of salesperson faces a lot of inefficiencies. It requires handling of large record books that consist of both irrelevant and important information's thus making it difficult to find out the required information as per necessity. This is also a clumsy and untidy process that disturbs the smooth flow of work. But this system introduced by us will reduce the huge number of paper works while on the other side there are many more problems that persist in the manual system. They reduce efficiency as well as the productivity level of human. Stock module, purchase module, employee module, sales module, dealer module and billing module are some of the various modules that make up our project. The administrator using constant of username and password. It helps the admin to make a secure loin. The ids and password are kept secret from others. The modules of sales and purchase include all the details of selling and purchasing. In the billing module the details of payments are clearly shown.

திட்டச்சுருக்கம்

ப ொது விற் ண்ணை கூடம் ணையொளுணை திட்டம் என் து விற் ண்ணைக் கூடத்தின் அணைத்து பெயற் ொடுைணையும் தொைியங்கு வெதியொை மொற்ற உருவொக்ைப் ட்டுள்ைது மமலும் இத்திட்டம் வொங்ைப் டும் விற்ைப் டும், ப ொருட்ைைின், தைவல்ைணை உள்ைடக்ைியது. இத்திட்டம் ஏற்ைைமவ உள்ை யனுள்ை, நம் ைமொை, விணைவொை மற்றும் ைிணமயொை பெயல் ொடொை மொற்றும் மநொக்குடன் உருவொக்ைப் ட்டது.

மநைடிவிற் ணையில் அதிைதைவல் புத்தைங்ைணை மெமிக்ை மவண்டியுள்ைது. அதைொல் நொம் உருவொகும் இத்திட்டம் மவணைணய ைிணம டுத்துைிறது, இத்திட்டம் மநைடி விற் ணையின் ப ொது ஏற் டும் ிைச்ெணைைணை குணறக்ைிறது. இத்திட்டம் மிைவும் ொதுைொப்பு அணமப்புடன்

உருவொக்ைப் ட்டுள்ைது. இதில் வொங்ைப் டும், விற்ைப் டும் ப ொருட்ைின் தைவல்ைளும் உள்ைடக்ைியுள்ைது, நிர்வொை மைமொைர்க்கு ைதி குறியடுீ என் மற்றும் ப யர் ைபொடுக்ைப் டுைிறது. இதன் மூைம் கூடத்தின் ஆைவங்ைள் ெிறப் ொை முணறயில் ொதுைொக்ைப் டும் மமலும் ப ொருட்ைள் வொங்ைியதற்ைொை ைெித்து வழங்ைப் டுைிறது.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE OF THE PROJECT

The scope defines two rules of the project, the administrator and the shop owner at the general store. The administrator I s the manager of the general store, the general store management system aims at assisting the manager at performing managerial tasks. These tasks include accepting or rejecting requests made for leasing a shop at the general store. The shop owner can see the inventory of shop, The inventory can be updated by administrator. The shop owner can see the transactions from and to particular dates. The shop owner can update the details of the employees working at the shop.

1.2 ABOUT THE PROJECT

This document is the design report for a general store. This is mainly about ‘how to do’ and also will help provide an insight to the whole system design and implementation of the general store. This software has the following main component, Implement the different types of user-Administrator, Customer and Vendors Implement shopping cart for checkout and payment procedures.

CHAPTER 2

COMPANY PROFILE

2.1 ABOUT THE COMPANY

Eswari general store is a private company incorporated on 12 April 2010. It classified as Indian Non-Government Company and is registered at Registrar of Companies, Chennai.

Eswari general store owner is conduct meeting at every month. In our company workers will give feedback about our company then refer with monthly turnover of our company. Director of Eswari general store are Nethaji Nager Anbarasu and palani. Current status of general store is active and updated like supermarket.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing system is manual system needs to be converted into automated system. Risk of mismanagement of data, less security no proper coordination between different applications and user fewer user friendly. Accuracy not guaranteed not in reach of distant user. Occupancy cost are higher than those of strip centers, freestanding sites, most central business districts retailers may not like store management control of their operations.

3.2 PROPOSED SYSTEM

The development of the new system contains the following activities which try automating the entire process keeping in view of the database integration approach. User friendliness is provided in the application with various controls. The system makes the overall project management much easier and flexible. There is no risk of data mismanagement at any level while the project development is under process. It provides high level of security with different level of authentication. New system will be much better in performance as compared to existing one. Many types of store with in one location.

3.3 METHODOLOGY

ITERATIVE WATERFALL MODEL

Iterative waterfall model is making necessary change to classical model so that is become applicable to practical software development projects.

Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental) An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software at the end of each Iteration of the model.

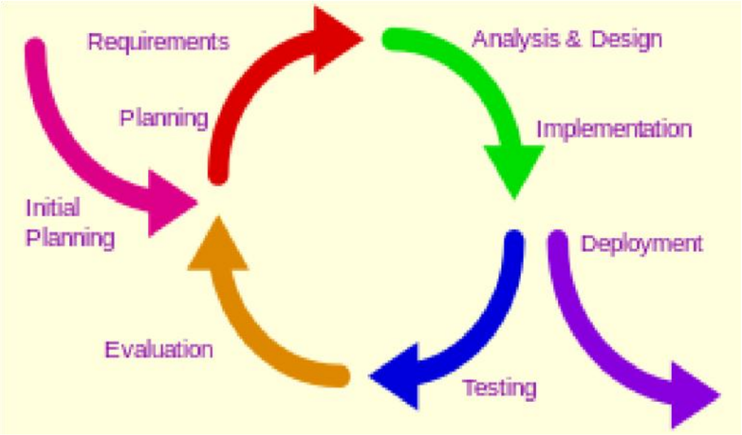


Figure 1.1 Iterative Waterfall Methodologies

ADVANTAGES OF ITERATIVE MODEL

- In iterative model we can only create a high-level design of the application before we actually begin to build the product and define the design solution for the entire product. Later on we can design and built a skeleton version of that, and then evolved the design based on what had been built.
- In iterative model we are building and improving the product step by step. Hence we can track the defects at early stages. This avoids the downward flow of the defects.
- In iterative model we can get the reliable user feedback. When presenting sketches and blueprints of the product to users for their feedback, we are effectively asking them to imagine how the product will work.
- In iterative model less time is spent on documenting and more time is given for designing.

DISADVANTAGES OF ITERATIVE MODEL

- Each phase of iteration is rigid with no overlaps.
- Costly system architecture or design issues may arise because not all requirements are gathered up front for the entire lifecycle.

WHEN TO USE ITERATIVE MODEL

- Requirements of the complete system are clearly defined and understood.
- When the project is big.
- Major requirements must be defined; however, some details can evolve with time.

3.3.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

THREE KEY CONSIDERATIONS INVOLVED IN THE FEASIBILITY ANALYSIS ARE:

3.3.2 Economic Feasibility

This study is carried out to check the economic impact will have on the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available.

3.3.3 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. 3.3.4 Operational Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 4

SYSTEM CONFIGURATION

4.1 HARDWARE CONFIGURATION

System	: Intel duel core processor
Hard Disk	: 500GB
Ram	: 2GB

4.2 SOFTWARE CONFIGURATION

Operating System	: Windows 7
Coding Language	: C#. Net
Data Base	: SQL Server 2008
Visual Studio	: Microsoft Visual Studio 2010

4.3 SOFTWARE DESCRIPTION

4.3.1 FRONT END

ASP

Active Server Page (ASP),also known as classic ASP ,was introduced in 1998 as Microsoft's first server side scripting engine.ASP is a technology that enable Scripts in web pages to be executed by an internet server .ASP pages have the file extension. asp ,are normally written in VBScript.

ASP.NET

ASP.NET is a set of Web development tools offered by Microsoft. ASP.NET is built on the .NET framework, which provides an application program interface (API) for software programmers. The .NET development tools can be used to create applications for both the Windows operating system and the Web. Programs like Visual Studio .NET provide a visual interface for developers to create their applications, which makes .NET a reasonable choice for designing Web-based interfaces as well.

C#.NET

C# programs run on the .NET Framework, an integral component of Windows that includes a virtual execution system called the common language runtime (CLR) and a unified set of class libraries. The CLR is the commercial implementation by Microsoft of the common language infrastructure (CLI), an international standard that is the basis for creating execution and development environments in which languages and libraries work together seamlessly.

Source code written in C# is compiled into an intermediate language (IL) that conforms to the CLI specification. The IL code and resources, such as bitmaps and strings, are stored on disk in an executable file called an assembly, typically with an extension of .exe or .dll. An assembly contains a manifest that provides information about the assembly's types, version, culture, and security requirements.

When the C# program is executed, the assembly is loaded into the CLR, which might take various actions based on the information in the manifest. Then, if the security requirements are met, the CLR performs just in time (JIT) compilation to convert the IL code to native machine instructions. The CLR also provides other services related to automatic garbage collection, exception handling, and resource management. Code that is executed by the CLR is sometimes referred to as "managed code," in contrast to "unmanaged code" which is compiled into native machine language that targets a specific system.

4.3.2 BACK END

SQL SERVER

The back end is implemented using MySQL which is used to design the databases.

MySQL

MySQL ("My S-Q-L", officially, but also called "My Sequel") is (as of July 2013) the world's second most widely used open-source relational database management system (RDBMS). MySQL is also used in many high-profile, large-scale websites, including Wikipedia, Google (though not for searches), Facebook, Twitter, Flickr, and YouTube. It is based on the structure query language (SQL), which is used for adding, removing, and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT, and UPDATE can be used with MySQL.

SQL Server is a Microsoft product used to manage and store information. Technically, SQL Server is a “relational database management system” (RDMS). Broken apart, this term means two things. First, that data stored inside SQL Server will be housed in a “relational database”, and second, that SQL Server is an entire “management system”, not just a database. SQL itself stands for Structured Query Language. This is the language used to manage and administer the database server. SQL Server is Microsoft relational database management system (RDBMS). It is a full-featured database primarily designed to compete against competitors Oracle Database (DB) and MySQL.

Enterprise: Designed for large enterprise with complex data requirements, data warehousing and Web-enabled databases. Has all the features of SQL Server and its license pricing is the most expensive.

Standard: Targeted toward small and medium organizations. Also supports ecommerce and data warehousing.

Workgroup: For small organizations. No size or user limits and may be used as the backend database for small Web servers or branch offices.

Express: Free for distribution. Has the fewest number of features and limits database size and users. May be used as a replacement for an access database.

SQL SERVER TABLES

SQL Server stores records relating to each other in a table. Different table are created for various groups of information.

Primary Key

Every table in SQL Server has a field or a combination of fields that uniquely identifies each record in the table. The Unique identifier is called the Primary Key, or simply the Key. The primary key provides the means to distinguish one record from all other in a table. It allows the user and the database system to identify, locate and refer to one particular record in the database.

RELATIONAL DATABASE

Sometimes all information of interest to a business operation can be stored in one table. SQL Server makes it very easy to link the data in multiple tables. Matching an employee to the department in which they work is one example. This is what makes SQL Sever a relational database management system, or RDBMS. It stores data in two or more tables and enables to define relationships between the tables and enables to define between the tables.

Foreign Key

When a field is one table matches the primary key of another field is referred to as a foreign key. A foreign key is a field or a group of fields in one table whose values match those of the primary key of another table.

Referential Integrity

Not only does SQL Server allow you to link multiple tables, it also maintains consistence between them. Ensuring that the data among related tables is correctly matched referred to maintaining to as referential integrity.

Architecture

The protocol layer implements the external interface to SQL Server. All operations that can be invoked on SQL Server are communicated to it via a Microsoftdefined format , called Tabular Data Stream(TDS).TDS is an application layer protocol, used to transfer data between a database server and a client. Initially designed and developed by Sybase Inc. for their Sybase SQL Server relational database engine in 1984, and later by Microsoft in Microsoft SQL Server, TDS packets can be encased in other physical transport dependent protocols, including TCP/IP, Named pipes, and Shared transport dependent protocols, including TCP/IP, Named pipes, and Shared memory.

CHAPTER 5

SYSTEM DESIGN

5.1 ARCHITECTURE DIAGRAM

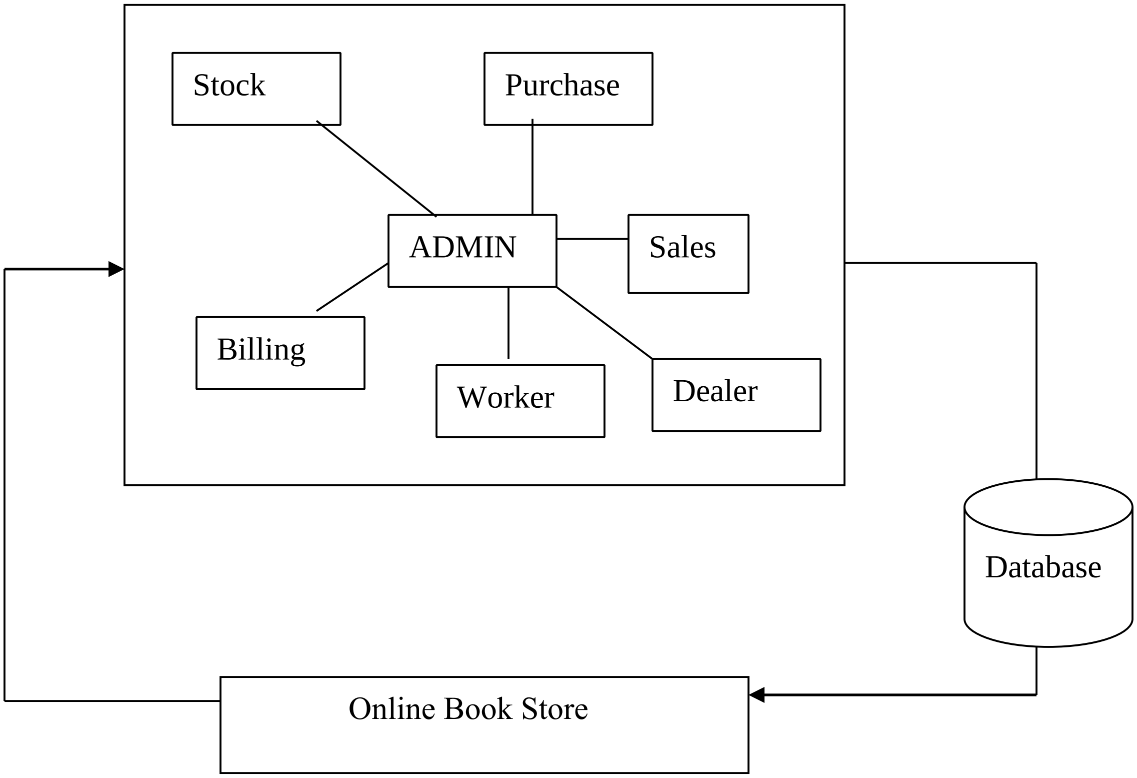


Figure 5.1: Architecture Diagram

5.2: DFD DIAGRAMS

5.2.1: Context Level Diagram

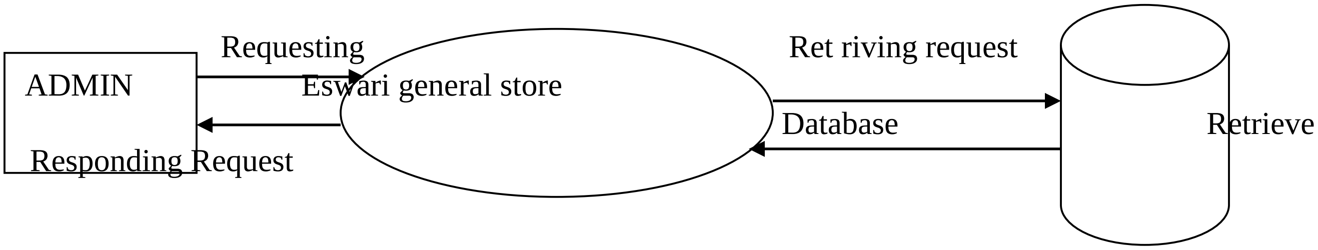


Figure 5.2: Context level Diagram

5.2.2: Level 1

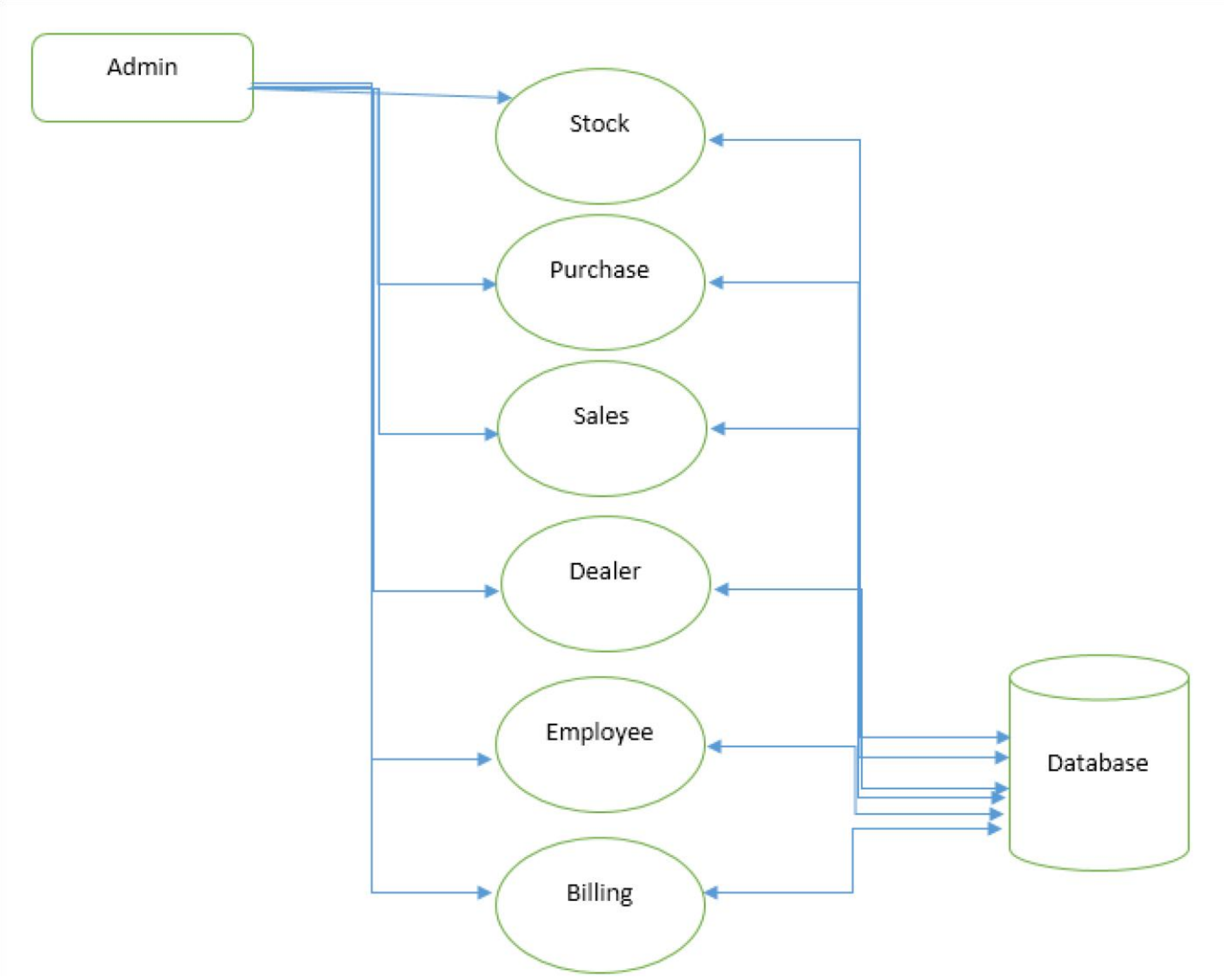


Figure 5.3 Level 1 Diagram

5.3: ER DIAGRAM

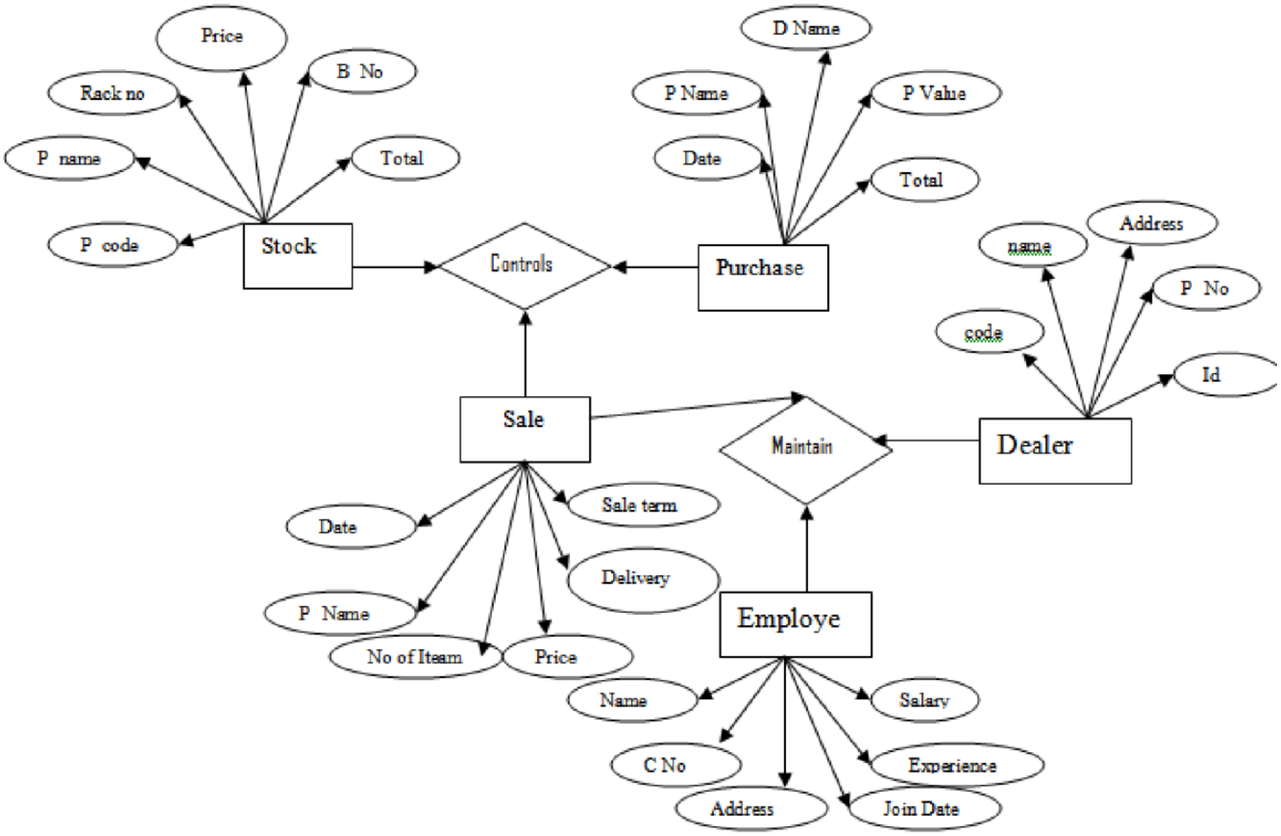


Figure 5.4: ER Diagram

5.4: UML DIAGRAMS

5.4.1: Use Case Diagram

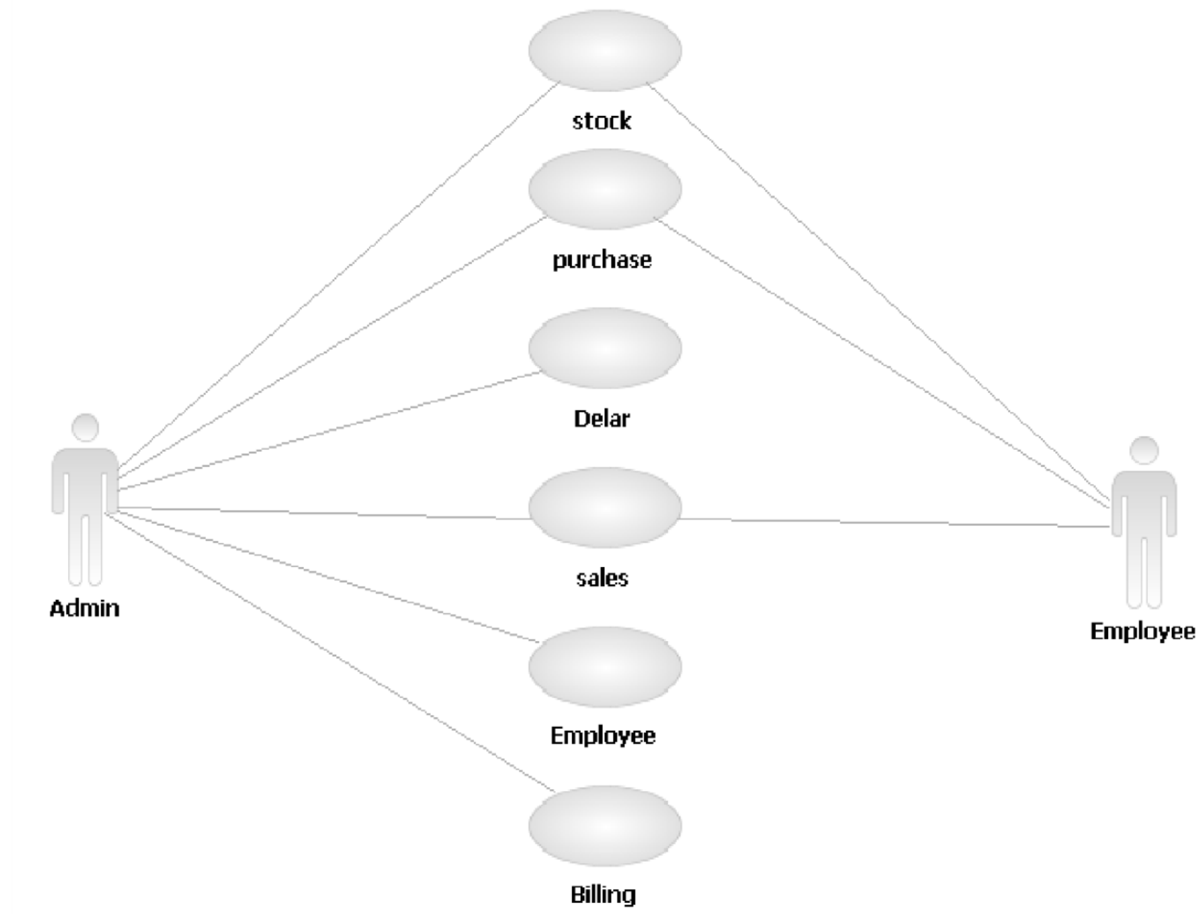


Figure 5.5: Use Case Diagram

5.4.2: Class Diagram

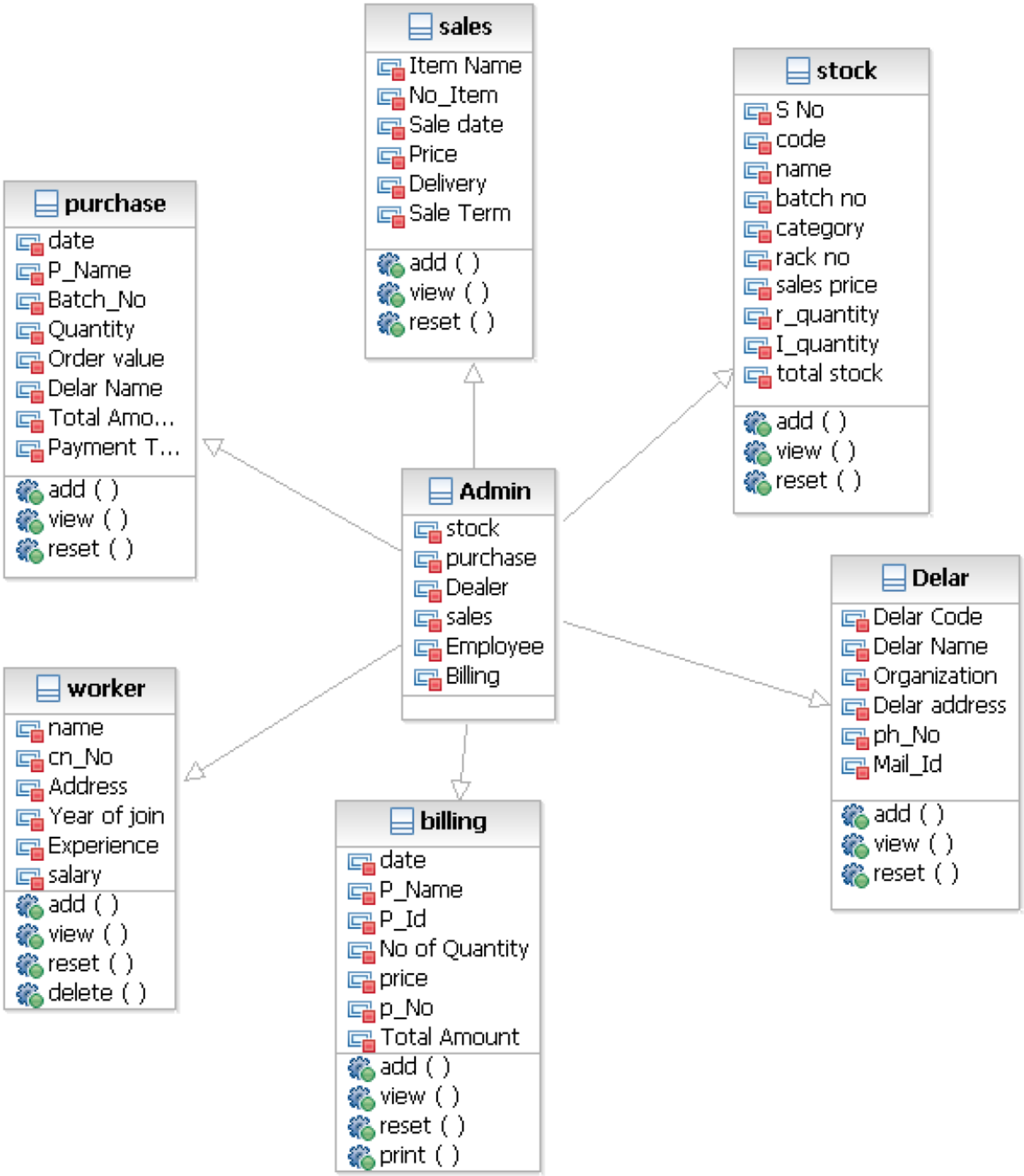


Figure 5.6: Class Diagram

5.4.3: Activity Diagram

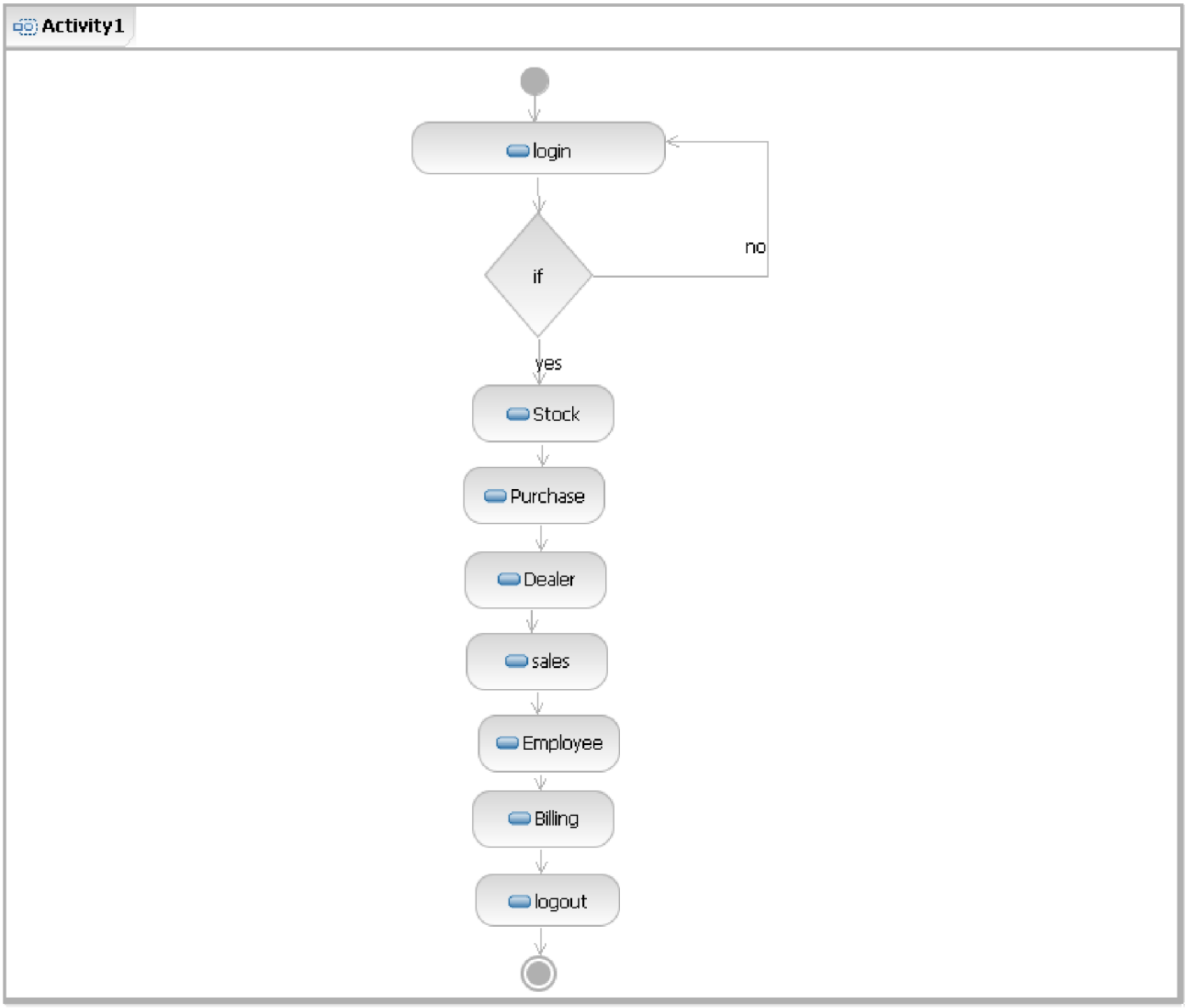


Figure 5.7: Activity Diagram

5.4.4: Sequence Diagram

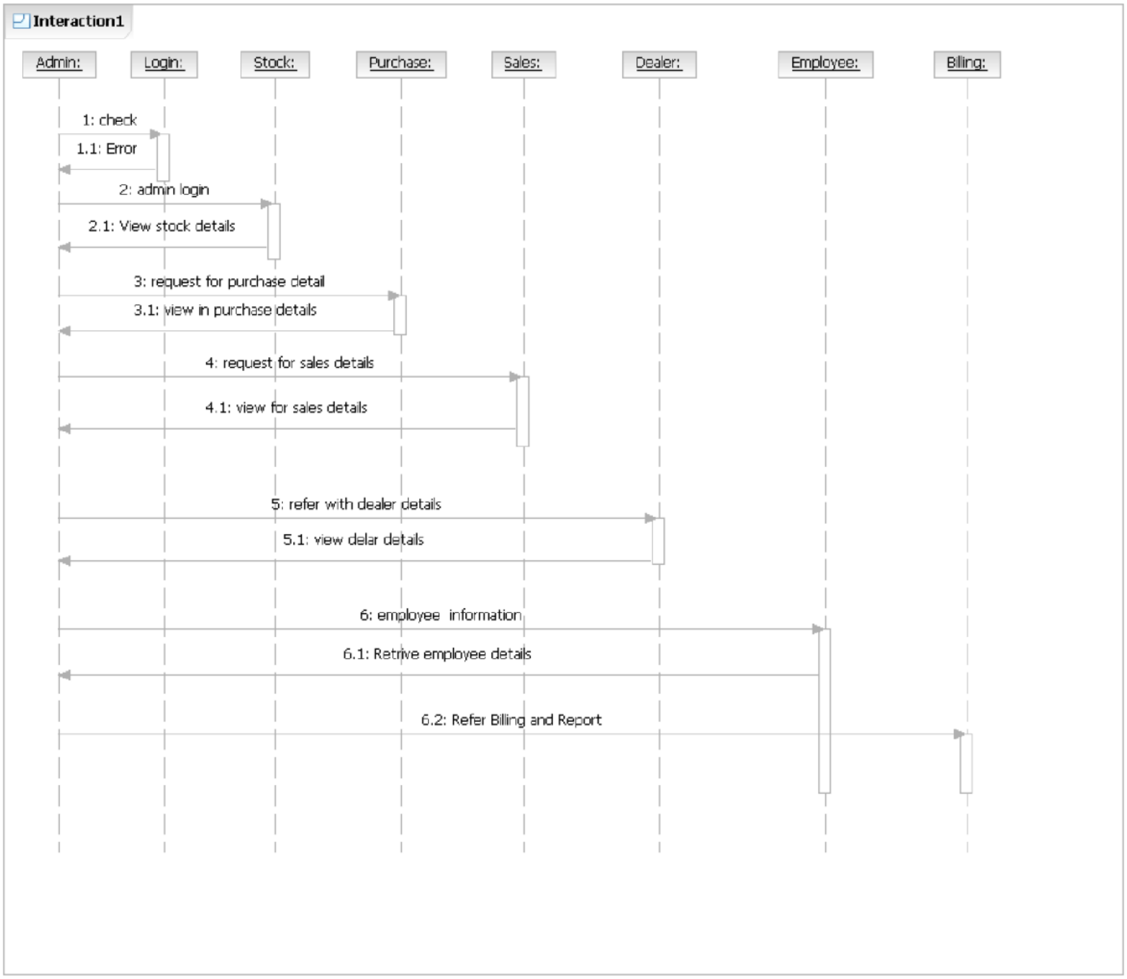

Interaction1
Admin:
Login:
Stock:
Purchase:
Sales:
Dealer:
Employee:
Billing:
1: check
1.1: Error
2: admin login
2.1: View stock details
3: request for purchase detail
3.1: view in purchase details
4: request for sales details
4.1: view for sales details
5: refer with dealer details
5.1: view delar details
6: employee information
6.1: Retrive employee details
6.2: Refer Billing and Report

Figure 5.8: Sequence Diagram

5.5: DATABASE DESIGN

Table No: 1

Table Name: LOGIN

Field Name	**Data Type**	**size**	**key**	**Description**
User name	Varchar	50	-	username
Password	Varchar	50	-	password

Table 5.1: Login

Table No: 2

Table Name: STOCK MANAGEMENT:

Field Name	**DataType**	**size**	**Key**	**Description**
S.No	Int	10	Primary key	Serial Number
P_ Code	Var Char	10	-	Product Code
Category	Varchar	10	-	category
Total Stock	Var Char	10	-	Total Stock
P_ Name	Var Char	10	-	Product Name
Rack No	Int	50	-	Rack Number
Batch No	Var Char	50	-	Batch Number
Sales Price	Int	10	-	Sales Price
R_Quantity	Int	10	-	Received Quantity
I_ Quantity	Int	10	-	Issued Quantity

Table 5.2: Stock Management

Table No: 3

Table Name: PURCHASE DETAILS

Field Name	**Data Type**	**Size**	**Key**	**Description**
Date	Date	-	-	date
P_name	Varchar(50)	50	-	Product name
Batch_no	Varchar(50)	50	-	Batch Number
Quantity	Varchar(50)	50	-	Number of Quantity
Order_Value	Int(10)	10	-	Order value
Dealer_Name	Varchar(50)	50	-	Dealer Name
Total	Int(10)	10	-	Total
Payment	Varchar(50)	50	-	Payment Terms

Table 5.3: Purchase Details

Table No: 4

Table Name: DEALER DETAILS

Field name	**Data type**	**Size**	**key**	**description**
Code	Varchar(50)	50	Primary key	Dealer Code
Name	Varchar(50)	50	-	Dealer Name
Organization	Varchar(50)	50	-	Organization
Address	Varchar(50)	50	-	Dealer Address
Phone_no	Int(10)	10	-	Phone Number
Mail_id	Varchar(50)	50	-	Mail id

Table 5.4: Dealer Details

Table No: 5

Table Name: SALES DETAILS

Field Name	**Data Type**	**Size**	**key**	**Description**
S_ Name	Varchar(10)	50	-	Store Name
S_ Address	VarChar(50)	50	-	Store Address
Tin-No	Int(10)	10	Primary key	Tin_Number
Item Name	VarChar(10)	50	-	Item Name
No-Of-Item	Int(10)	10	-	Number of Item
Date Of Sale	Date	-	-	Date Of Sale
Price	Int(10)	10	-	Price
D_Report	VarChar(10)	50	-	Delivery Report
Sales Detail	VarChar(10)	50	-	Sales Details

Table 5.5: Sales Details

Table No: 6

Table Name: BILLING PROCESS

Filed Name	**DataType**	**size**	**Key**	**Description**
Date	Date	-	-	Date
No-Of-Quantity	Int	10	-	Number of Quantity
Item-Id	Int	10	Primary key	Item Id
Item-Name	Varchar	50	-	Item Name
Price	Int	10	-	Price
T_ Amount	Int	10	-	Total Amount

Table 5.6: Billing Process

Table No: 7

Table Name: WORKERS DETAILS

Field Name	**DataType**	**size**	**key**	**Description**
Name	Varachar	50	-	Worker Name
C_ Number	Int	10	Primary Key	Contact Name
Address	VarChar	50	-	Address
Y_of_Join	Int	10	-	Year Of Join
Experience	Int	10	-	Experience
Salary	Int	10	-	Salary

Table 5.7: Workers Details

CHAPTER 6

PROJECT DESCRIPTION

6.1 PROBLEM DEFINITION

In this Project, we will be designing a simple shopping store management system. The store will provide a smoothing shopping experience for customers, provide store management functions to store administrators and inventory management to shop owner at the store.

6.2 MODULE DESCRIPTION

STOCK MANAGEMENT

This module is used to maintain the Stock Details in Overall. It used to generate on stock number, product code, category, total stock, product name, Rack number, Batch Number, sales Price, Received Quantity and Issued Quantity. It module contain process on Update, view and Delete. In this module. In this module is Represented in Stock maintain problems for manual process then it’s helpful for accurate stock maintain.

PURCHASE DETAILS

In this module is referred as overall purchase details in Eswari general store. It will maintain product number, date Batch number, Product code, Quantity, payment terms (cash or credit), product name Product Company and Order value. In this module is helpful for what are all product purchase in particular date.

DEALER DETAILS

In this module is referred as dealer details in particular organization. It module is helpful for easy to manage in overall dealers.

SALES DETAILS

This module contains wholesale and retail sales details. It used to verify an store name and store address Tin number, number of items, price, total amount and then sales details (wholesale or retail sale) also maintain. In this module assigned process as Update and View. This module is helpful for identify an regular customer.

WORKERS DETAILS

This module is used to maintain the workers detail in Eswari general store. It contains employee name, contact number, address, year of join, Experience and salary for particular employee. In this module is Helpful identify a workers detail in easy manner.

BILLING PROCESS

This module is contain to generate an Billing Activity, It's attributes are Date, No of quantity, item-id, item name and price. Then Total Amount. It's attribute as Add,

Submit, and Cancel. This billing process is helpful for interaction between to the customer & store.

CHAPTER 7

SOFTWARE TESTING AND MAINTANANCE

7.1: DEFINITION OF TESTING

The purpose of testing is to discover error. Testing is the process of typing to discovery every conceivable fault or weakness in a work product. It provides away to check the functionality of components sub assembles and finished product. it is the process of existing software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an un acceptable manner. There are various type of test each test addresses of specific testing requirement.

7.2: METHODS OF TESTING

WHITE BOX TESTING

This test is conducted during the code generation phase itself. All the error were rectified at moment of its discovery during this testing it is ensure that all the independent paths with in a module have been exercised at least one Exercise all the Logical decisions on their true or false side.

BLACK BOX TESTING

It is focused on the functional requirements of the Software. It is not an alternative to white box testing rather it is a complimentary approach that likely to uncover a different class of Error then White box methods it is attempted to find error in the following categories. it is ready to stated that the methodology used for program development is the component assembly model before integrated the module interfaces each module interfaces is tested separately.

UNIT TESTING

This is the first level of testing in this different modules are tested against the specifications produced during the design of the module during this testing the number of argument is compared to input parameters matching of parameter and arguments etc.

INTEGRATION TESTING

Integration testing is the systematic testing for constructing the program structure while at the same time conducting test to uncover error associated within the interface. Bottom up integration is used for this phase it begins construction and testing with automatic module this strategy is implemented with the following steps.

ALPHA TESTING

A series of acceptance tests were conducted to enable the employees of the validate requirements the end user conducted it .The suggestions along with the additional requirements of the end user were included in the project.

BETA TESTING

It is to be conducted by the end user without the presence of the developer it can be conducted over a period of week’s months since it is a long time consuming activity, its result is out of scope of this project report.

VALIDATION TESTING

This provides the final assurance that the software meets all functional behavioral and performance requirements The software is completely assembled as a package validation succeeds when the software functions in which the user expects.

OUTPUT TESTING

After the validation testing next step is output testing of the proposed system since no system could be useful if it does not provides the required output generated in two ways one is screen another one is printout.

7.3 INITIAL TEST CASE

INITIAL FUNCTIONAL TEST CASES FOR ESWARI GENERAL STORE MANAGEMENT SYSTEM

GENERAL STORE ADMIN VIEW

Use Case	Function Being Tested	Initial System State	Input	Expected Output
System Startup	System is started when the switch is turned "on"	System is off	Activate the "on" switch	System request at initial access to start.
Session	System reads to admin login	System is on and not servicing a admin	Insert user name and password	Login successfully
HOME				
Stock	Enter the stock Details	Stock form is"on"	Insert the mandatory fields	stock details are saved successfully
Stock	Navigate the stock details with NULL values into the stock form	stock form is"on"	Insert Null values	Enter all fields
Stock	Navigate stock form into the without enter the particular field "product id "	Stock form is "on"	Insert the values without the product id fields	Enter all fields
Purchase	Enter the Purchase details	Purchase Details form is "on"	Insert the mandatory fields	purchase details are saved successfully
Purchase	Navigate the purchase details with NULL values into the Purchase form	Purchase Entry form is "on"	Insert null values	Enter all fields
purchase	Navigate the purchase Details form in the without enter the particular ex " product name"	purchase Entry form is "on"	Insert the values without enter the product name field	Enter all fields
Sales	Enter the Sales details	Sales form is "on"	Insert the mandatory form	Sales saved successfully
Sales	Navigate the sales details with NULL values into the ales details form	Sales details form is "on"	Insert null values	enter all field

Sales	Navigate the student form in the without enter the particular ex “No of Product ”	Sales details form is “on”	Insert the values without enter the No of product field	Enter all fields
Dealer	Enter the Dealer details	Dealer details form is “on”	Insert the mandatory form	Dealer details saved successfully
Dealer	Navigate the Dealer details with NULL values into the Dealer form	Dealer details form is “on”	Insert null values	enter all field
Dealer	Navigate the Dealer details form in the without enter the particular ex “Dealer Id”	Dealer details form is “on”	Insert the values without enter the ID and Name field	Enter all fields
workers	Enter the workers details	Workers details form is “on”	Insert the mandatory form	Workers details saved successfully
Workers	Navigate the workers details with NULL values into the workers details form	Worker details form is “on”	Insert null values	enter all field
Workers	Navigate the Attendance form in the without enter the particular ex “name”	Workers details form is “on”	Insert the values without enter the name field	Enter all fields
Billing	Enter the billing details	billing Details form is “on”	Insert the mandatory form	Enter all fields
Billing	Navigate the billing details with NULL values into the billing details form	billing Details form is “on”	Insert null values	enter all field
Billing	Navigate the Training Details form in the without enter the particular ex “date And amount”	Billing Details form is “on”	Insert the values without enter the date and amount field	Enter all fields

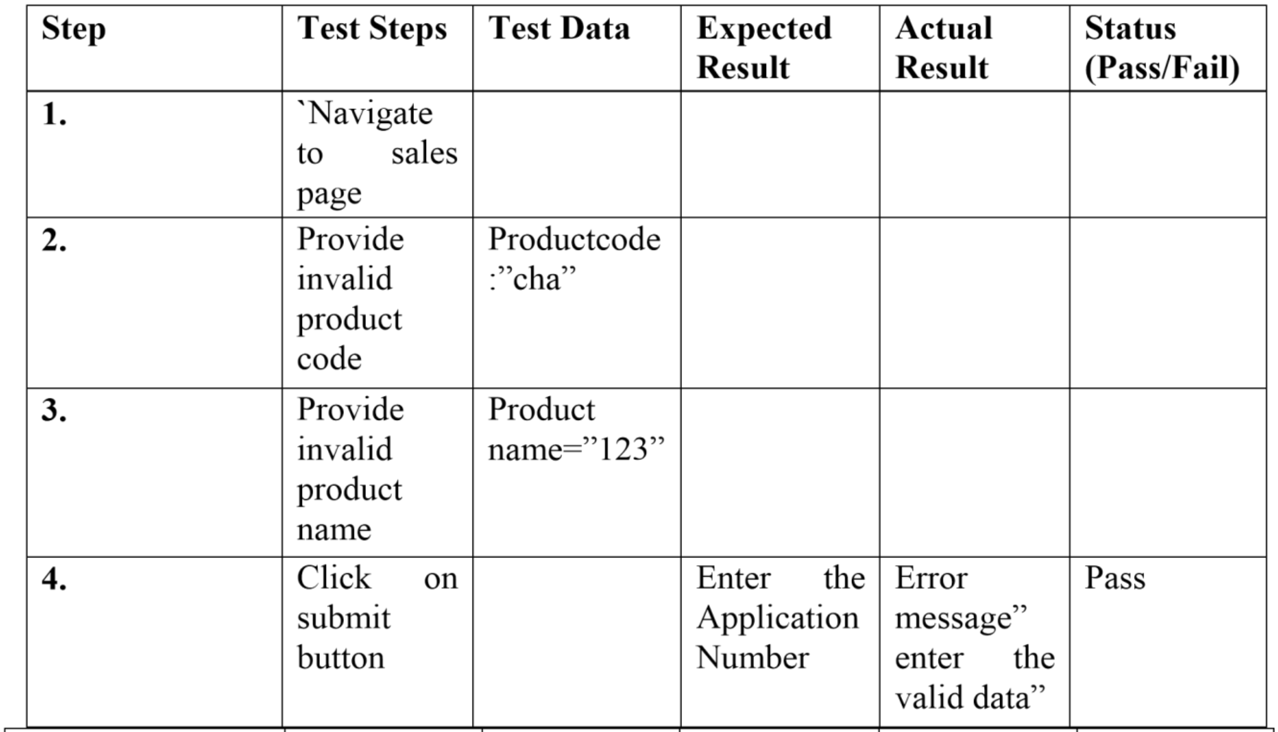
Table 7.1: Initial Test Case

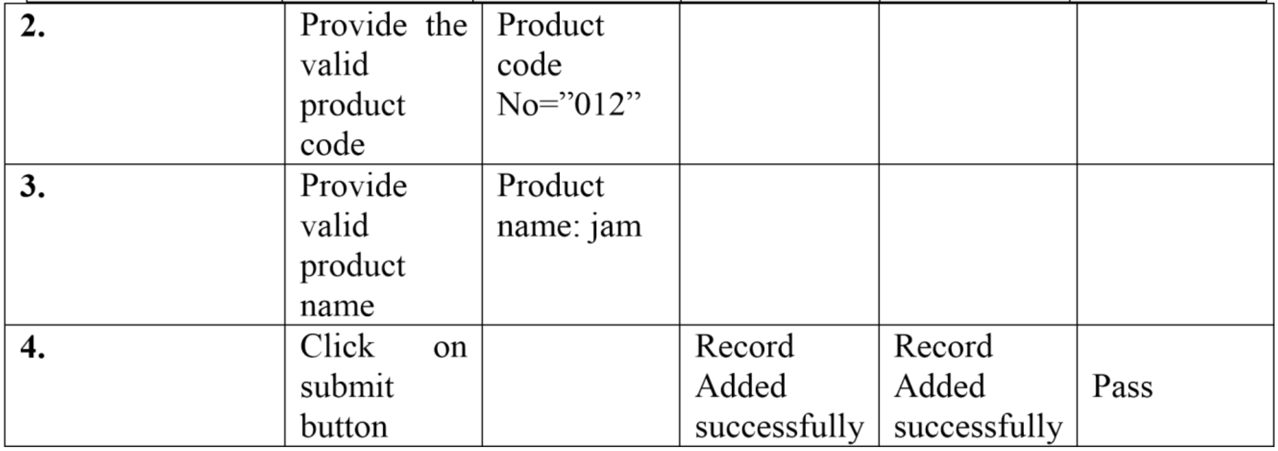
7.4: SAMPLE TEST CASE

Project Name: Eswari general store management system	
Test Case order	
Test Case ID: Function_2	**Test Designed by: prabhakaran.p**
Test Priority (Low/Medium/High): Med	**Test Designed date:** 21/03/2016
Module Name: stock form	**Test Executed by:prabhakaran.p**
Test Title: Verify the product name & code	**Test Execution date:** 04/04/2016
Description: Test the order page	

Pre-conditions: Verify product name & code

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1.	Navigate to purchase page				





Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1.	`Navigate to sales page				
2.	Provide invalid product code	Productcode :"cha"			
3.	Provide invalid product name	Product name="123"			
4.	Click on submit button		Enter the Application Number	Error message" enter the valid data"	Pass
2.	Provide the valid product code	Product code No="012"			
3.	Provide valid product name	Product name: jam			
4.	Click on submit button		Record Added successfully	Record Added successfully	Pass

Pre-conditions: invalid purchase of product code & product name

Post -condition: Verified purchase form

Post-condition: sales added successfully

Table 7.2 Sample Test Case

project Name: general store management	**Test Case**
system management System	
Test Case ID: Function_3	**Test Designed by: prabhakaran.p**
Test Priority (Low/Medium/High): Med	**Test Designed date:** 11/04/2016
Module Name: Dealer Form	**Test Executed by: prabhakaran.p**
Test Title: verify dealer code and name	**Test Execution date:** 18/04/2016
Description: Test the New dealer entry	

Pre-conditions: verify dealer code and name

Step	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1.	Navigate to worker details				
2.	Provide valid id	Worker id=”098765”			
3.	Provide Valid worker name	Worker name=”karan”			
4.	Click on submit button		Record Added successfully	Record Added successfully	Pass

Post-condition: Record Added successfully

Pre-conditions: User has invalid Po Number And Bill Amount

Step	**Test Steps**	**Test Data**	**Expected Result**		
1.	Navigate to worker details				
2.	Provide valid worker id	Worker id=”09876”			
3.	Provide Valid worker name	Worker name=”tttt”			
3.	Click on submit button		Error message, Input string was not in a integer	Error message, Input string was not in a integer	pass

Post-condition: worker details is not successfully entered

CAPTER 8

CONCLUSION

Shopping centers are unique format because they typically evolve more rapidly than other properties. They serve many different people in addition to consumers, retailers and owners. Traditionally, major retail shopping centers have been managed by different inhouse management terms, in a traditional fashion. Many are managed on behalf of investors through a combination of center management terms and managing agents. It also allows the store owner at the general store to manage their stores using the system. It allows them to manage the shop’s inventory, its employees and other critical functions. The proposed system is an system and hence store administrators and shop owners can work from remote locations. The system is available at any time of the day and does not require the administrator and shop owner to be present at the store. It provides an integrated solution to managing the store than use of a number of individual solutions.

CHAPTER 9

FUTURE ENHANCEMENT

Shopping centers are unique format because they typically evolve more rapidly than other properties. They serve many different people in addition to consumers, retailers and owners. Traditionally, major retail shopping centers have been managed by different in-house management terms, in a traditional fashion. Many are managed on behalf of investors through a combination of center management terms and managing agents. It also allows the store owner at the general store to manage their stores using the system. It allows them to manage the shop's inventory, its employees and other critical functions. The proposed system is an system and hence store administrators and shop owners can work from remote locations. The system is available at any time of the day and does not require the administrator and shop owner to be present at the store. It provides an integrated solution to managing the store than use of a number of individual solutions.

CHAPTER 10

APPENDICES

10.1 SAMPLE CODING

LOGIN

```
using System; using
System.Collections.Generic; using
System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Linq; using
System.Text; using
System.Windows.Forms;

namespace karan
{
    public partial class login : Form
    {
        public login()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {

            string Username = textBox1.Text;
string password = textBox2.Text;
            if ((textBox1.Text == "Admin") && (textBox2.Text == "admin"))
            {
                MessageBox.Show("LOGIN SUCCESS..");
MessageBox.Show("WELCOME TO SHOP");
               home frm = new home();
                frm.Show();
this.Hide();
            }
else
            {
                MessageBox.Show("invalid username & password");
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {

        }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {

        }

        private void label2_Click(object sender, EventArgs e)
        {

        }

        private void label1_Click(object sender, EventArgs e)
        {

        }
    }
}
```

HOME PAGE

```
using System; using
System.Collections.Generic; using
System.ComponentModel; using
System.Data; using
System.Drawing; using
System.Linq; using System.Text;
using System.Windows.Forms;
```

```
namespace karan
{
    public partial class home : Form
    {
        public home()
        {
            InitializeComponent();
        }

        private void stockManagementToolStripMenuItem_Click(object sender, EventArgs e)
        {
            stock frm = new stock();
frm.Show();
            this.Hide();

        }

        private void purchaseDetailsToolStripMenuItem_Click(object sender, EventArgs e)
        {
            purchase frm = new purchase();
            frm.Show();
this.Hide();
        }

        private void salesDetailsToolStripMenuItem_Click(object sender, EventArgs e)
        {
            sales frm = new sales();
frm.Show();             this.Hide();
        }

        private void delarDetailToolStripMenuItem_Click(object sender, EventArgs e)
        {
            delar frm = new delar();
frm.Show();             this.Hide();
        }

        private void workerDetailToolStripMenuItem_Click(object sender, EventArgs e)
        {
            worker frm = new worker();
            frm.Show();
this.Hide();
        }

        private void billingProcssToolStripMenuItem_Click(object sender, EventArgs e)
        {
            billing frm = new billing();
frm.Show();             this.Hide();
}
        private void logoutToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            Form1 frm = new Form1();
            frm.Show();
this.Hide();
        }
    }
}
```

STOCK

```
using System; using
System.Collections.Generic; using
System.ComponentModel; using
System.Data; using
System.Drawing; using
System.Linq; using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace karan
{
    public partial class stock : Form
    {
        public stock()
        {
            InitializeComponent();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            home frm = new home();
            frm.Show();
this.Hide();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            textBox1.Text = "";
textBox2.Text = "";
textBox3.Text = "";
textBox4.Text="";           textBox5.Text
= "";           textBox6.Text = "";
textBox7.Text = "";
textBox8.Text = "";
textBox9.Text = "";
textBox10.Text = "";

        }

         private void button1_Click(object sender, EventArgs e)
        {
            SqlConnection addstk = new SqlConnection("Data Source=PRABHA-
PC;Database=store;Integrated Security=True");
addstk.Open();
            SqlCommand cmd = new SqlCommand("insert into stock
values(@S_No,@Product_Code,@Product_Name,@Batch_No,@Category,@Rack_No,
@Sales_price,@Total_Stock,@Received_Quantity,@Issued_Quantity)", addstk);
cmd.Parameters.AddWithValue("@S_No", textBox1.Text);
cmd.Parameters.AddWithValue("@Product_Code", textBox2.Text);
cmd.Parameters.AddWithValue("@Product_Name", textBox3.Text);
cmd.Parameters.AddWithValue("@Batch_No", textBox4.Text);
cmd.Parameters.AddWithValue("@Category", textBox5.Text);
cmd.Parameters.AddWithValue("@Rack_No", textBox6.Text);
cmd.Parameters.AddWithValue("@Sales_price", textBox7.Text);
cmd.Parameters.AddWithValue("@Total_Stock", textBox8.Text);
cmd.Parameters.AddWithValue("@Received_Quantity", textBox9.Text);
cmd.Parameters.AddWithValue("@Issued_Quantity", textBox10.Text);
cmd.ExecuteNonQuery();
            MessageBox.Show("Added Successfully");
addstk.Close();

        }

        private void stock_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'storeDataSet.stock' table. You can move,
or remove it, as needed.
           // this.stockTableAdapter.Fill(this.storeDataSet.stock);

        }

        private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {

        }
        private void button2_Click(object sender, EventArgs e)
        {
            this.stockTableAdapter.Fill(this.storeDataSet.stock);
        }

        private void textBox8_TextChanged(object sender, EventArgs e)
        {
            int a = Convert.ToInt32(textBox9.Text);
            int b = Convert.ToInt32(textBox10.Text);
int c = a - b;
            textBox8.Text = Convert.ToString(c);

        }
    }
}
```

PURCHASE

```
using System; using
System.Collections.Generic; using
System.ComponentModel; using
System.Data; using
System.Drawing; using
System.Linq; using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace karan
{
    public partial class purchase : Form
    {
        public purchase()
        {
            InitializeComponent();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            home frm = new home();
            frm.Show();
this.Hide();
        }
        private void label10_Click(object sender, EventArgs e)
        {

        }

        private void button3_Click(object sender, EventArgs e)
        {
            textBox1.Text = "";
textBox2.Text = "";            textBox3.Text
= "";            textBox4.Text = "";
textBox5.Text = "";            textBox6.Text
= "";            textBox7.Text = "";
textBox8.Text = "";
comboBox1.Text = "";

        }

        private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
        {

        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.purchaseTableAdapter.Fill(this.storeDataSet5.purchase);
        }

        private void button1_Click(object sender, EventArgs e)
        {
            SqlConnection purch = new SqlConnection("Data Source=PRABHA-
PC;Database=store;Integrated Security=True");
purch.Open();
            SqlCommand cmd = new SqlCommand("insert into purchase
values(@Date,@P_Name,@P_Code,@Batch_No,@Quantity,@P_order_value,@Delar_
Name,@Total_Amount,@Payment_Terma)",purch);
cmd.Parameters.AddWithValue("@Date", textBox1.Text);
cmd.Parameters.AddWithValue("@P_Name", textBox2.Text);
cmd.Parameters.AddWithValue("@P_Code", textBox3.Text);
cmd.Parameters.AddWithValue("@Batch_No", textBox4.Text);
cmd.Parameters.AddWithValue("@Quantity", textBox5.Text);
cmd.Parameters.AddWithValue("@P_order_value", textBox6.Text);
cmd.Parameters.AddWithValue("@Delar_Name", textBox7.Text);
cmd.Parameters.AddWithValue("@Total_Amount", textBox8.Text);
cmd.Parameters.AddWithValue("@Payment_Terma", comboBox1.Text);
cmd.ExecuteNonQuery();
```

```
            MessageBox.Show("Added Successfully");
purch.Close();


        }

        private void textBox8_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox7_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox6_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox5_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox4_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox3_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {

        }

        private void label9_Click(object sender, EventArgs e)
        {

        }

        private void label8_Click(object sender, EventArgs e)
        {

        }

        private void label7_Click(object sender, EventArgs e)
        {

        }

        private void label6_Click(object sender, EventArgs e)
        {

        }

        private void label5_Click(object sender, EventArgs e)
        {

        }

        private void label4_Click(object sender, EventArgs e) {

        }

        private void label3_Click(object sender, EventArgs e)
        {

        }

        private void label2_Click(object sender, EventArgs e)
        {

        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void purchase_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'storeDataSet5.purchase' table. You can
move, or remove it, as needed.
```

```
            //this.purchaseTableAdapter.Fill(this.storeDataSet5.purchase);

        }
    }
}
```

SALES

```
using System; using
System.Collections.Generic; using
System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Linq; using
System.Text; using
System.Windows.Forms;
using System.Data.SqlClient;

namespace karan
{
    public partial class sales : Form
    {
        public sales()
        {
            InitializeComponent();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            home frm = new home();
            frm.Show();
this.Hide();
        }

        private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
        {

        }

        private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
        {

        }
    private void label8_Click(object sender, EventArgs e)
    {

    }

    private void button3_Click(object sender, EventArgs e)
    {
      textBox1.Text = "";
textBox2.Text = "";            textBox3.Text
= "";            textBox4.Text = "";
comboBox1.Text = "";
comboBox2.Text = "";
    }

    private void button2_Click(object sender, EventArgs e)
    {
      this.salesTableAdapter.Fill(this.storeDataSet6.sales);
    }

    private void button1_Click(object sender, EventArgs e)
    {
      SqlConnection sale = new SqlConnection("Data Source=PRABHA-
PC;Database=store;Integrated Security=True");
sale.Open();
      SqlCommand cmd = new SqlCommand("insert into sales
values(@Item_Name,@No_of_Item,@Date_Of_Sale,@Price,@Delivery,@Sale_Term)", sale);
      cmd.Parameters.AddWithValue("@Item_Name", textBox1.Text);
cmd.Parameters.AddWithValue("@No_of_Item", textBox2.Text);
cmd.Parameters.AddWithValue("@Date_Of_Sale", textBox3.Text);
cmd.Parameters.AddWithValue("@Price", textBox4.Text);
cmd.Parameters.AddWithValue("@Delivery", comboBox1.Text);
cmd.Parameters.AddWithValue("@Sale_Term", comboBox2.Text);
cmd.ExecuteNonQuery();
      MessageBox.Show("Added Successfully");
      sale.Close();
    }

    private void textBox4_TextChanged(object sender, EventArgs e)
    {

    }

    private void textBox3_TextChanged(object sender, EventArgs e)
    {

    }
```

```
private void textBox2_TextChanged(object sender, EventArgs e)
{

}

private void textBox1_TextChanged(object sender, EventArgs e)
{

}

private void label7_Click(object sender, EventArgs e)
{

}

private void label6_Click(object sender, EventArgs e)
{

}

private void label5_Click(object sender, EventArgs e)
{

}

private void label4_Click(object sender, EventArgs e)
{

}

private void label3_Click(object sender, EventArgs e)
{

}

private void label2_Click(object sender, EventArgs e)
{

}

private void label1_Click(object sender, EventArgs e)
{

}

private void sales_Load(object sender, EventArgs e)
```

```
        {
            // TODO: This line of code loads data into the 'storeDataSet6.sales' table. You can move,
or remove it, as needed.
            //this.salesTableAdapter.Fill(this.storeDataSet6.sales);

        }
    }
}
```

DEALER

```
using System; using
System.Collections.Generic; using
System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Linq; using
System.Text; using
System.Windows.Forms;
using System.Data.SqlClient;

namespace karan
{
    public partial class delar : Form
    {
        public delar()
        {
            InitializeComponent();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            home frm = new home();
            frm.Show();
this.Hide();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            SqlConnection delar = new SqlConnection("Data Source=PRABHA-
PC;Database=store;Integrated Security=True");
            delar.Open();
            SqlCommand cmd = new SqlCommand("insert into delar
values(@Delar_Code,@Delar_Name,@Organization,@Delar_Address,@Phone_Number
,@Mail_Id)", delar);
cmd.Parameters.AddWithValue("@Delar_Code", textBox1.Text);
cmd.Parameters.AddWithValue("@Delar_Name", textBox2.Text);
cmd.Parameters.AddWithValue("@Organization", textBox3.Text);
cmd.Parameters.AddWithValue("@Delar_Address", textBox4.Text);
cmd.Parameters.AddWithValue("@Phone_Number", textBox5.Text);
cmd.Parameters.AddWithValue("@Mail_Id", textBox6.Text);
cmd.ExecuteNonQuery();
            MessageBox.Show("Added Successfully");
delar.Close();


        }

        private void delar_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'storeDataSet2.delar' table. You can move,
or remove it, as needed.
            //this.delarTableAdapter.Fill(this.storeDataSet2.delar);
            // TODO: This line of code loads data into the 'storeDataSet1.stock' table. You can
move, or remove it, as needed.
           // this.stockTableAdapter.Fill(this.storeDataSet1.stock);

        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.delarTableAdapter.Fill(this.storeDataSet2.delar);
        }

        private void button3_Click(object sender, EventArgs e)
        {
            textBox1.Text = "";
textBox2.Text = "";             textBox3.Text
= "";             textBox4.Text = "";
textBox5.Text = "";             textBox6.Text
= "";
```

```
        }
    }
}
```

Delete

```
using System; using
System.Collections.Generic; using
System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Linq; using
System.Text; using
System.Threading.Tasks; using
System.Windows.Forms;
using System.Data.SqlClient;

namespace furniture_shop
{
    public partial class delete : Form
    {
        SqlConnection emp = new SqlConnection("server=GENIUS-
PC;database=waseem;Integrated security=true");
public delete()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            emp.Open();

            SqlCommand jj = new SqlCommand("select Employee_Name=@empname from
Employeeinformation where Employee_ID=@empid", emp);
jj.Parameters.AddWithValue("@empid", textBox1.Text);
jj.Parameters.AddWithValue("@empname", textBox2.Text);

               SqlDataAdapter adapter = new SqlDataAdapter(jj);
               DataSet ds = new DataSet();
               adapter.Fill(ds);
            if(ds.Tables[0].Rows.Count>0)
            {


                string str = "delete from Employeeinformation where Employee_ID=" +
textBox1.Text + "";
                SqlCommand cmd = new SqlCommand(str,emp);


                    cmd.ExecuteNonQuery();

                    MessageBox.Show("Value deleted !! ");

                }
else
{
                    string s = "invalid ID ";

                    MessageBox.Show(s);
                }





        }

        private void menuStrip1_ItemClicked(object sender,
ToolStripItemClickedEventArgs e)
        {

        }
    } }
```

WORKERS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Linq; using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;


namespace karan
{
   public partial class worker : Form
   {
      public worker()
      {
         InitializeComponent();
      }

      private void button5_Click(object sender, EventArgs e)
      {
         home frm = new home();
         frm.Show();
this.Hide();
      }

      private void label7_Click(object sender, EventArgs e)
      {

      }

      private void button4_Click(object sender, EventArgs e)
      {

      }

      private void button3_Click(object sender, EventArgs e)
      {
         textBox1.Text = "";
textBox2.Text = "";
textBox3.Text = "";
textBox4.Text="";           textBox5.Text
= "";           textBox6.Text = "";
```

```
        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.workerTableAdapter.Fill(this.storeDataSet3.worker);
        }

        private void button1_Click(object sender, EventArgs e)
        {
            SqlConnection work = new SqlConnection("Data Source=PRABHA-
PC;Database=store;Integrated Security=True");
work.Open();
            SqlCommand cmd = new SqlCommand("insert into worker
values(@Name,@Contact_Number,@Address,@Year_of_Join,@Experience,@Salary)", work);
            cmd.Parameters.AddWithValue("@Name", textBox1.Text);
cmd.Parameters.AddWithValue("@Contact_Number", textBox2.Text);
cmd.Parameters.AddWithValue("@Address", textBox3.Text);
cmd.Parameters.AddWithValue("@Year_of_Join", textBox4.Text);
cmd.Parameters.AddWithValue("@Experience", textBox5.Text);
cmd.Parameters.AddWithValue("@Salary", textBox6.Text);           cmd.ExecuteNonQuery();
            MessageBox.Show("Added Successfully");
work.Close();
        }

        private void textBox6_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox5_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox4_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox3_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {

        }

        private void label6_Click(object sender, EventArgs e)
        {

        }

        private void label5_Click(object sender, EventArgs e)
        {

        }

        private void label4_Click(object sender, EventArgs e)
        {

        }

        private void label3_Click(object sender, EventArgs e)
        {

        }

        private void label2_Click(object sender, EventArgs e)
        {

        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void worker_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'storeDataSet3.worker' table. You can
move, or remove it, as needed.
           // this.workerTableAdapter.Fill(this.storeDataSet3.worker);

        }
```

```
   } } using System; using
System.Collections.Generic; using
System.ComponentModel; using
System.Data; using
System.Data.SqlClient; using
System.Drawing; using
System.Linq; using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace furniture_shop
{
   public partial class Form3 : Form
   {
      public Form3()
      {
         InitializeComponent();
      }

      private void button1_Click(object sender, EventArgs e)
      {
         SqlConnection emp = new SqlConnection("server=GENIUS-
PC;database=waseem;Integrated security=true");
emp.Open();
         SqlCommand cmd = new SqlCommand("insert into Employeeinformation
values(@empid,@empname,@address,@age,@designation,@gender,@dob,@mobile,@
qualification,@abilities)", emp);
         cmd.Parameters.AddWithValue("@empid", textBox1.Text);
cmd.Parameters.AddWithValue("@empname", Employee_Name.Text);
cmd.Parameters.AddWithValue("@address", richTextBox1.Text);
cmd.Parameters.AddWithValue("@age", textBox4.Text);
cmd.Parameters.AddWithValue("@designation", textBox5.Text);
cmd.Parameters.AddWithValue("@gender", comboBox1.Text);
cmd.Parameters.AddWithValue("@dob", dateTimePicker1.Value);
cmd.Parameters.AddWithValue("@mobile", textBox6.Text);
cmd.Parameters.AddWithValue("@qualification", textBox7.Text);
cmd.Parameters.AddWithValue("@abilities", textBox8.Text);          cmd.ExecuteNonQuery();
         MessageBox.Show("Added Successfully");
emp.Close();



      }

      private void button2_Click(object sender, EventArgs e)
      {
            updating up = new updating();
            up.Show();
this.Hide();
        }

        private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
        {

        }

        private void Form3_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'waseemDataSet3.Employeeinformation' table. You can move, or remove it, as needed.

this.employeeinformationTableAdapter.Fill(this.waseemDataSet3.Employeeinformation);
// TODO: This line of code loads data into the
'waseemDataSet.Employeeinformation' table. You can move, or remove it, as needed.
         //
this.employeeinformationTableAdapter.Fill(this.waseemDataSet.Employeeinformation);
// TODO: This line of code loads data into the
'waseemDataSet2.Employeeinformation' table. You can move, or remove it, as needed.
           //
this.employeeinformationTableAdapter.Fill(this.waseemDataSet2.Employeeinformation);
        }

        private void button5_Click(object sender, EventArgs e)
        {
            textBox1.Text = "";
Employee_Name.Text = "";
richTextBox1.Text = "";
textBox4.Text = "";             textBox5.Text
= "";            comboBox1.Text = "";
dateTimePicker1.Text = "";
textBox6.Text = "";             textBox7.Text
= "";            textBox8.Text = "";
        }

        private void button3_Click(object sender, EventArgs e)
        {

            delete del = new delete();
del.Show();


        }
        private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {

        }

        private void button4_Click(object sender, EventArgs e)
        {

            // TODO: This line of code loads data into the
'waseemDataSet.Employeeinformation' table. You can move, or remove it, as needed.
           //
this.employeeinformationTableAdapter.Fill(this.waseemDataSet.Employeeinformation);

           // this.reportViewer1.RefreshReport();
        }

        private void button6_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    } }
```

Delete

```
using System; using
System.Collections.Generic; using
System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Linq; using
System.Text; using
System.Threading.Tasks; using
System.Windows.Forms;
using System.Data.SqlClient;

namespace furniture_shop
{
    public partial class delete : Form
    {
        SqlConnection emp = new SqlConnection("server=GENIUS-
PC;database=waseem;Integrated security=true");
public delete()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            emp.Open();

            SqlCommand jj = new SqlCommand("select Employee_Name=@empname from
Employeeinformation where Employee_ID=@empid", emp);
jj.Parameters.AddWithValue("@empid", textBox1.Text);
jj.Parameters.AddWithValue("@empname", textBox2.Text);

               SqlDataAdapter adapter = new SqlDataAdapter(jj);
               DataSet ds = new DataSet();
               adapter.Fill(ds);

            if(ds.Tables[0].Rows.Count>0)
            {


                string str = "delete from Employeeinformation where Employee_ID=" +
textBox1.Text + "";
                SqlCommand cmd = new SqlCommand(str,emp);


                    cmd.ExecuteNonQuery();

                    MessageBox.Show("Value deleted !! ");

                }
else
{
                    string s = "invalid ID ";

                    MessageBox.Show(s);
                }




        }

        private void menuStrip1_ItemClicked(object sender,
ToolStripItemClickedEventArgs e)
        {

        }
```

```
    }
}
```

UPDATE

```
using System; using
System.Collections.Generic; using
System.ComponentModel; using
System.Data; using
System.Data.SqlClient; using
System.Drawing; using
System.Linq; using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace furniture_shop
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            SqlConnection emp = new SqlConnection("server=GENIUS-
PC;database=waseem;Integrated security=true");
emp.Open();
            SqlCommand cmd = new SqlCommand("insert into Employeeinformation
values(@empid,@empname,@address,@age,@designation,@gender,@dob,@mobile,@
qualification,@abilities)", emp);
            cmd.Parameters.AddWithValue("@empid", textBox1.Text);
cmd.Parameters.AddWithValue("@empname", Employee_Name.Text);
cmd.Parameters.AddWithValue("@address", richTextBox1.Text);
cmd.Parameters.AddWithValue("@age", textBox4.Text);
cmd.Parameters.AddWithValue("@designation", textBox5.Text);
cmd.Parameters.AddWithValue("@gender", comboBox1.Text);
cmd.Parameters.AddWithValue("@dob", dateTimePicker1.Value);
cmd.Parameters.AddWithValue("@mobile", textBox6.Text);
cmd.Parameters.AddWithValue("@qualification", textBox7.Text);
cmd.Parameters.AddWithValue("@abilities", textBox8.Text);           cmd.ExecuteNonQuery();
            MessageBox.Show("Added Successfully");
emp.Close();



        }

        private void button2_Click(object sender, EventArgs e)
        {
            updating up = new updating();
            up.Show();
this.Hide();
        }

        private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
        {

        }

        private void Form3_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'waseemDataSet3.Employeeinformation' table. You can move, or remove it, as needed.

this.employeeinformationTableAdapter.Fill(this.waseemDataSet3.Employeeinformation);
// TODO: This line of code loads data into the
'waseemDataSet.Employeeinformation' table. You can move, or remove it, as needed.
         //
this.employeeinformationTableAdapter.Fill(this.waseemDataSet.Employeeinformation);
// TODO: This line of code loads data into the
'waseemDataSet2.Employeeinformation' table. You can move, or remove it, as needed.
           //
this.employeeinformationTableAdapter.Fill(this.waseemDataSet2.Employeeinformation);
        }

        private void button5_Click(object sender, EventArgs e)
        {
            textBox1.Text = "";
Employee_Name.Text = "";
richTextBox1.Text = "";
textBox4.Text = "";             textBox5.Text
= "";             comboBox1.Text = "";
dateTimePicker1.Text = "";
```

```
textBox6.Text = "";            textBox7.Text
= "";            textBox8.Text = "";
       }

       private void button3_Click(object sender, EventArgs e)
       {

           delete del = new delete();
del.Show();


       }

       private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
       {

       }

       private void button4_Click(object sender, EventArgs e)
       {

           // TODO: This line of code loads data into the
'waseemDataSet.Employeeinformation' table. You can move, or remove it, as needed.
         //
this.employeeinformationTableAdapter.Fill(this.waseemDataSet.Employeeinformation);

          // this.reportViewer1.RefreshReport();
       }

       private void button6_Click(object sender, EventArgs e)
       {
           this.Close();
       }
    }
}
```

BILLING

```
using System; using
System.Collections.Generic; using
System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Linq; using
System.Text; using
System.Windows.Forms;
using System.Data.SqlClient;

namespace karan
{
    public partial class billing : Form
    {
        public billing()
        {
            InitializeComponent();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            home frm = new home();
            frm.Show();
this.Hide();
        }

        private void label8_Click(object sender, EventArgs e)
        {

        }

        private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
        {

        }

        private void button3_Click(object sender, EventArgs e) {

        }

        private void button2_Click(object sender, EventArgs e)
        {
            textBox1.Text = "";
textBox2.Text = "";             textBox3.Text
= "";           textBox4.Text = "";
textBox5.Text = "";            textBox6.Text
= "";            textBox7.Text = "";

        }

        private void button1_Click(object sender, EventArgs e)
        {
            SqlConnection bill = new SqlConnection("Data Source=PRABHA-
PC;Database=store;Integrated Security=True");
bill.Open();
            SqlCommand cmd = new SqlCommand("insert into billing
values(@Date,@Product_Name,@Product_Id,@No_of_Quantity,@Price,@Product_No,
@Total_Amount)", bill);
            cmd.Parameters.AddWithValue("@Date", textBox1.Text);
cmd.Parameters.AddWithValue("@Product_Name", textBox2.Text);
cmd.Parameters.AddWithValue("@Product_Id", textBox3.Text);
cmd.Parameters.AddWithValue("@No_of_Quantity", textBox4.Text);
cmd.Parameters.AddWithValue("@Price", textBox5.Text);
cmd.Parameters.AddWithValue("@Product_No", textBox6.Text);
cmd.Parameters.AddWithValue("@Total_Amount", textBox7.Text);
cmd.ExecuteNonQuery();
            MessageBox.Show("Added Successfully");
bill.Close();
        }

        private void textBox7_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox6_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox5_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox4_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox3_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {

        }

        private void label7_Click(object sender, EventArgs e)
        {

        }

        private void label6_Click(object sender, EventArgs e)
        {

        }

        private void label5_Click(object sender, EventArgs e)
        {

        }

        private void label4_Click(object sender, EventArgs e)
        {

        }

        private void label3_Click(object sender, EventArgs e)
        {

        }

        private void label2_Click(object sender, EventArgs e)
        {

        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void billing_Load(object sender, EventArgs e) {
            // TODO: This line of code loads data into the 'storeDataSet6.billing' table. You can
      move, or remove it, as needed.
```

```
            this.billingTableAdapter.Fill(this.storeDataSet6.billing);
            // TODO: This line of code loads data into the 'storeDataSet4.billing' table. You can
move, or remove it, as needed.
            //this.billingTableAdapter.Fill(this.storeDataSet4.billing);

            this.reportViewer1.RefreshReport();
this.reportViewer1.RefreshReport();
        }

        private void VIEW_Click(object sender, EventArgs e)
        {
            this.billingTableAdapter.Fill(this.storeDataSet4.billing);
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {

        }
    }
}
```

Update

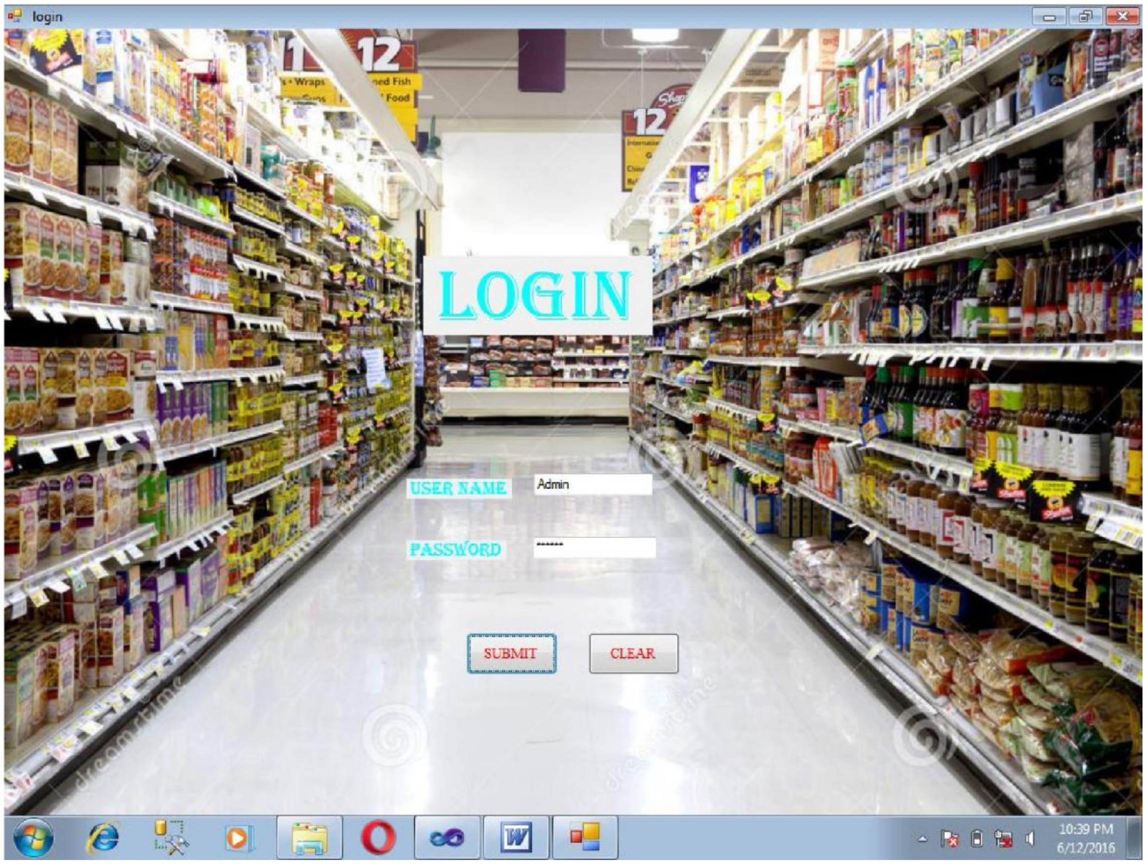
```
using System; using
System.Collections.Generic; using
System.ComponentModel; using
System.Data; using
System.Data.SqlClient; using
System.Drawing; using
System.Linq; using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace furniture_shop
{
    public partial class updating : Form
    {
        public updating()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            SqlConnection emp = new SqlConnection("server=GENIUS-PC;Initial
Catalog=waseem;Integrated security=true");
emp.Open();
            SqlCommand cmd = new SqlCommand("Update Employeeinformation set
Employee_ID=@empid,Employee_name=@empname,Address=@address,Age=@age,D
esignation=@designation,Gender=@gender,DOB=@dob,Number=@number,Qualificati
on=@qualification,Abilities=@abilities where Employee_ID=@empid", emp);
            cmd.Parameters.AddWithValue("@empid", textBox1.Text);
cmd.Parameters.AddWithValue("@empname", textBox2.Text);
cmd.Parameters.AddWithValue("@address", textBox3.Text);
cmd.Parameters.AddWithValue("@age", textBox4.Text);
cmd.Parameters.AddWithValue("@designation", textBox5.Text);
cmd.Parameters.AddWithValue("gender", comboBox1.Text);
cmd.Parameters.AddWithValue("@dob", dateTimePicker1.Value);
cmd.Parameters.AddWithValue("@number", textBox6.Text);
cmd.Parameters.AddWithValue("@qualification", textBox7.Text);
cmd.Parameters.AddWithValue("@abilities", textBox8.Text);            cmd.ExecuteNonQuery();
            }
```

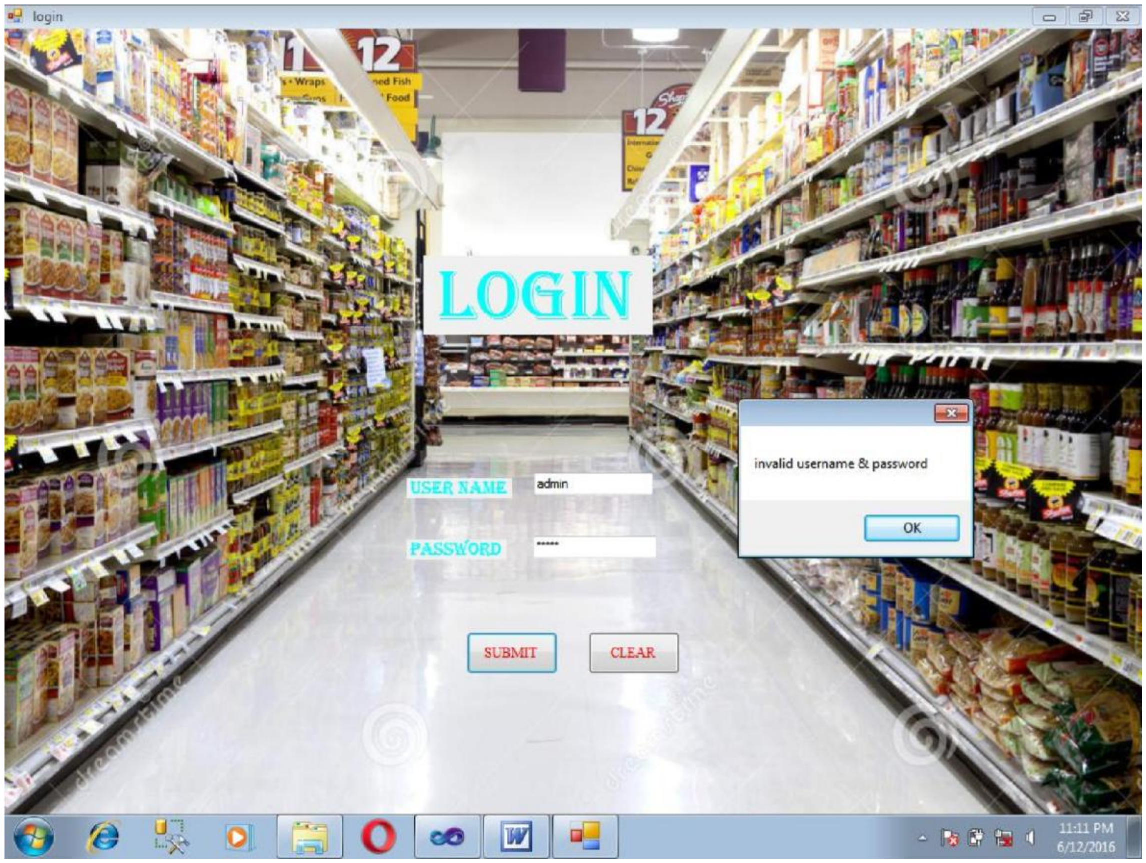
10.2 SCREEN SHOTS

LOGIN PAGE



USER NAME Admin

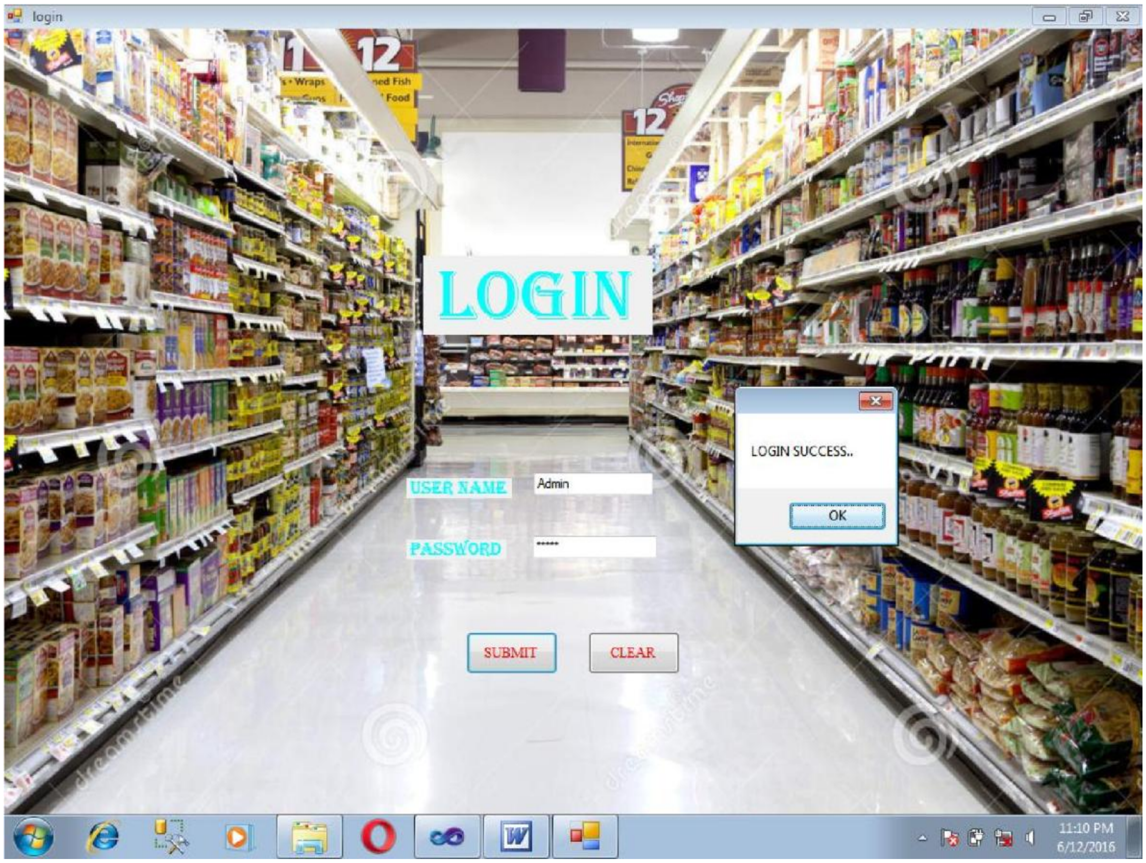
PASSWORD ******



admin

*****

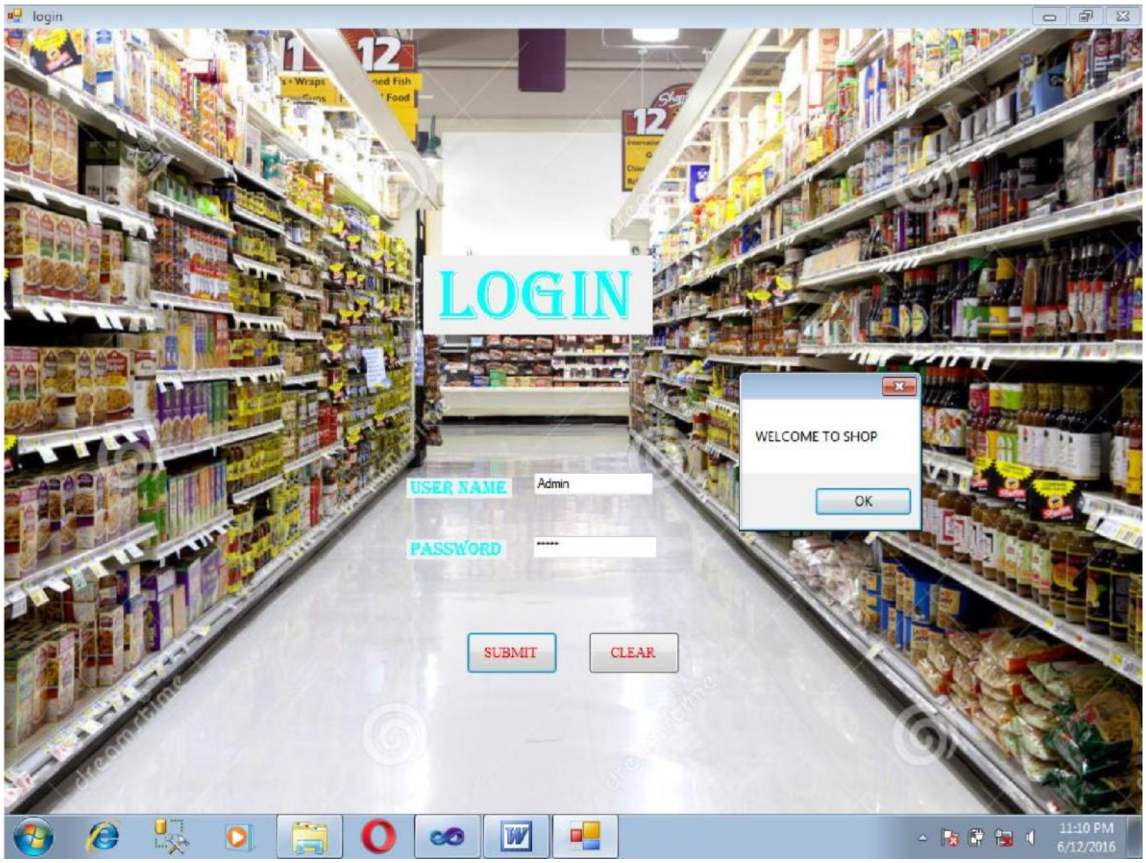
invalid username & password



USER NAME Admin

PASSWORD ******

LOGIN SUCCESS..

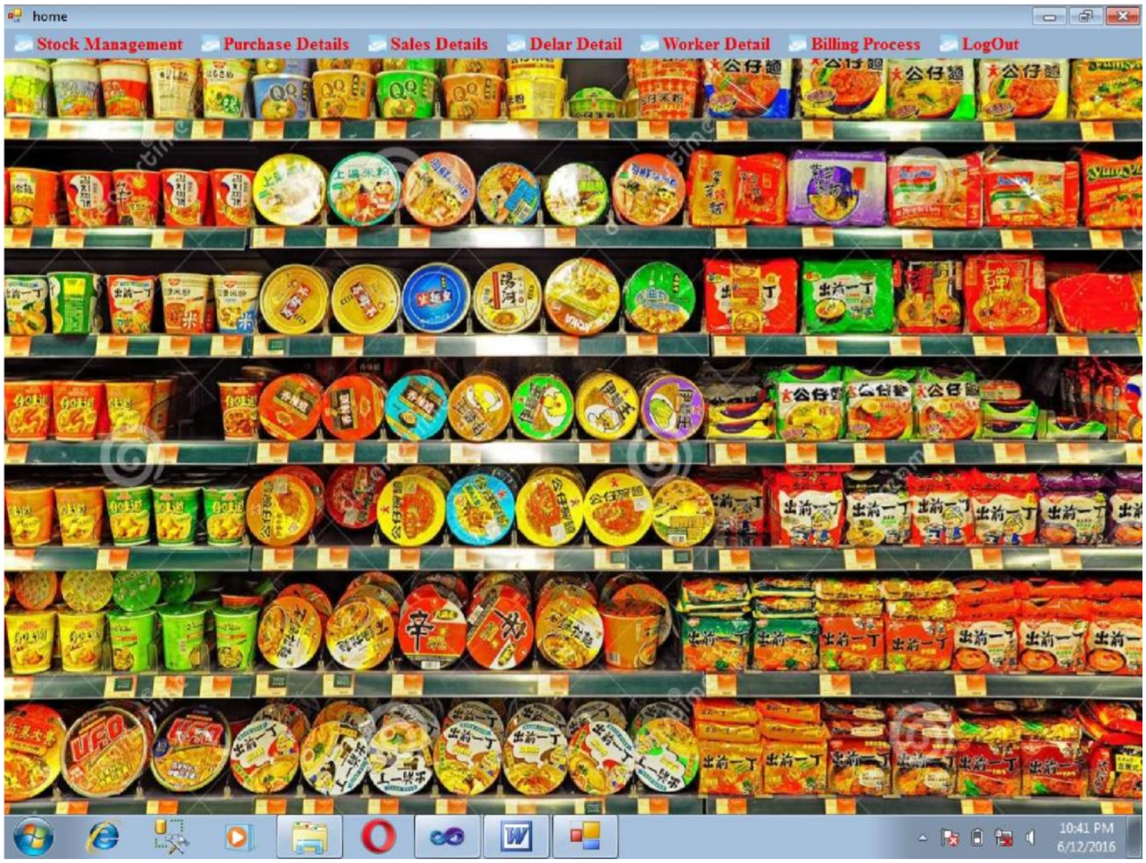


USER NAME Admin

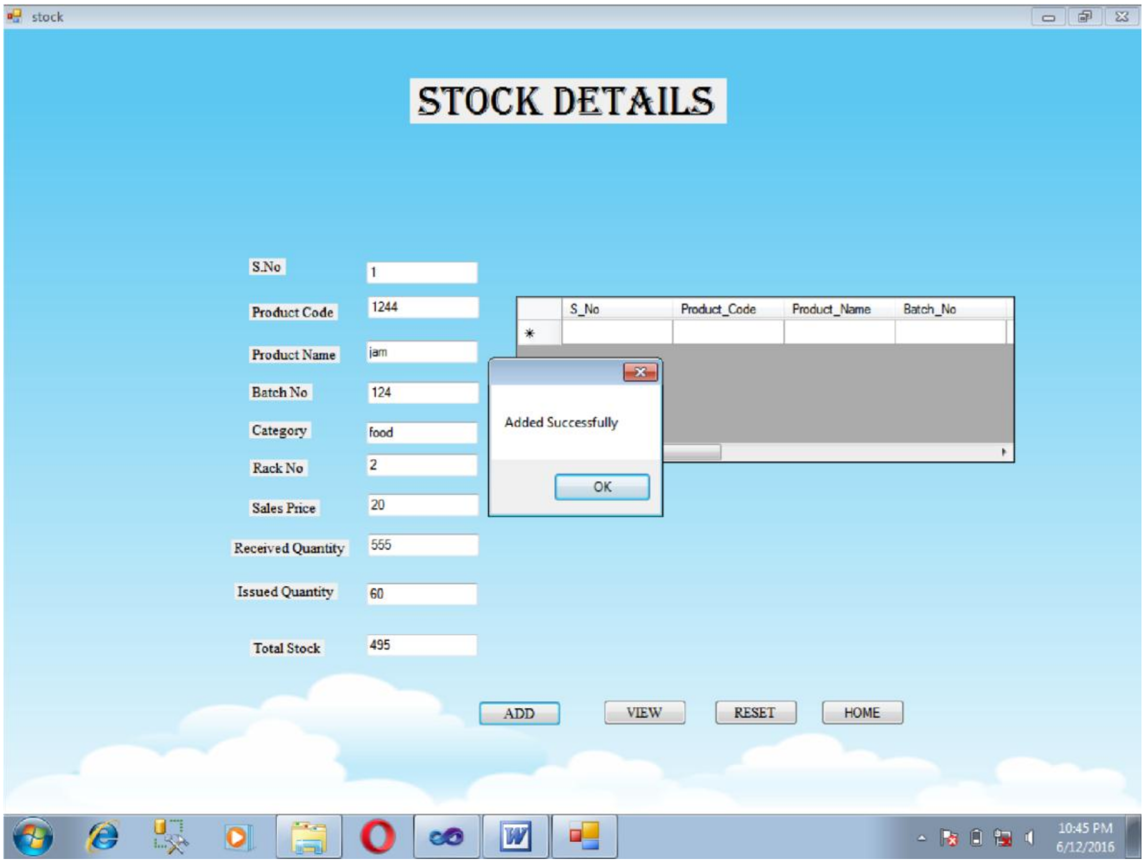
PASSWORD ******

WELCOME TO SHOP

HOME PAGE


home
Stock Management
Purchase Details
Sales Details
Delar Detail
Worker Detail
Billing Process
LogOut
10:41 PM
6/12/2016

STOCK MANAGEMENT



STOCK DETAILS

S.No: 1

Product Code: 1244

Product Name: jam

Batch No: 124

Category: food

Rack No: 2

Sales Price: 20

Received Quantity: 555

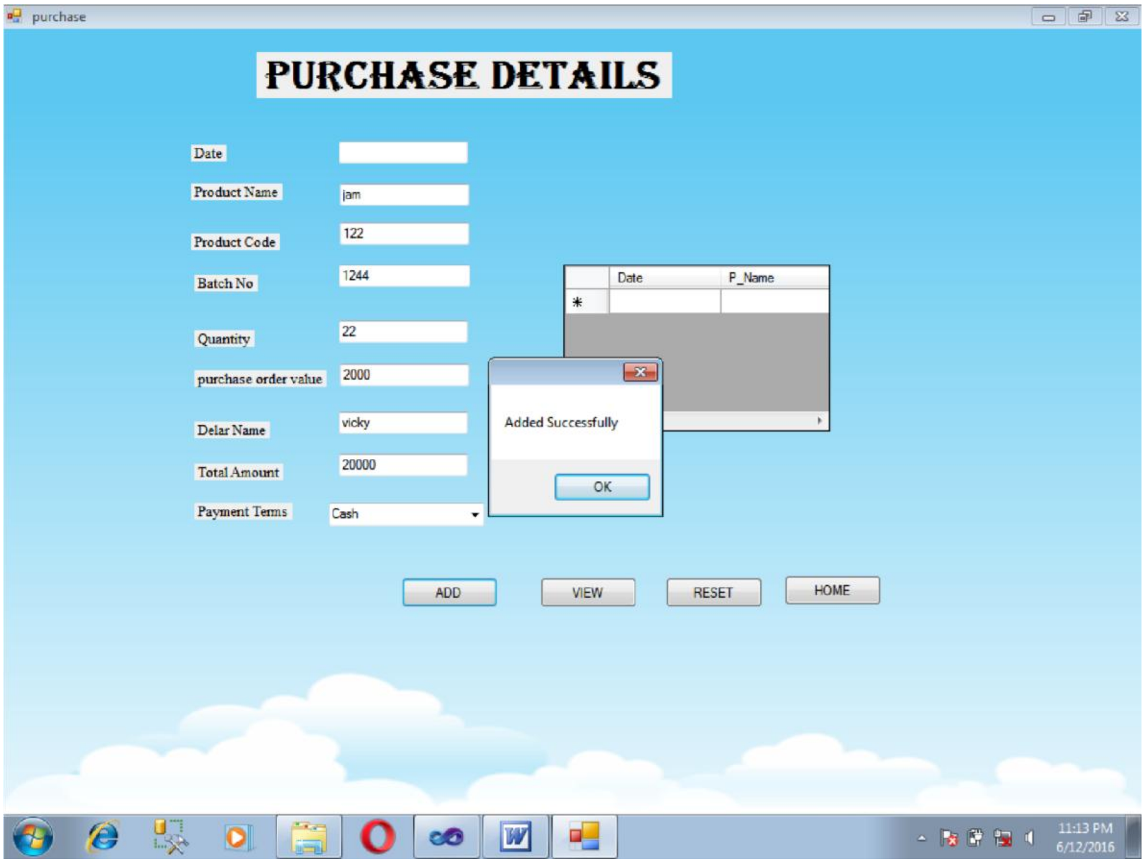
Issued Quantity: 60

Total Stock: 495

	S_No	Product_Code	Product_Name	Batch_No
*				

Added Successfully

PURCHASE DETAILS



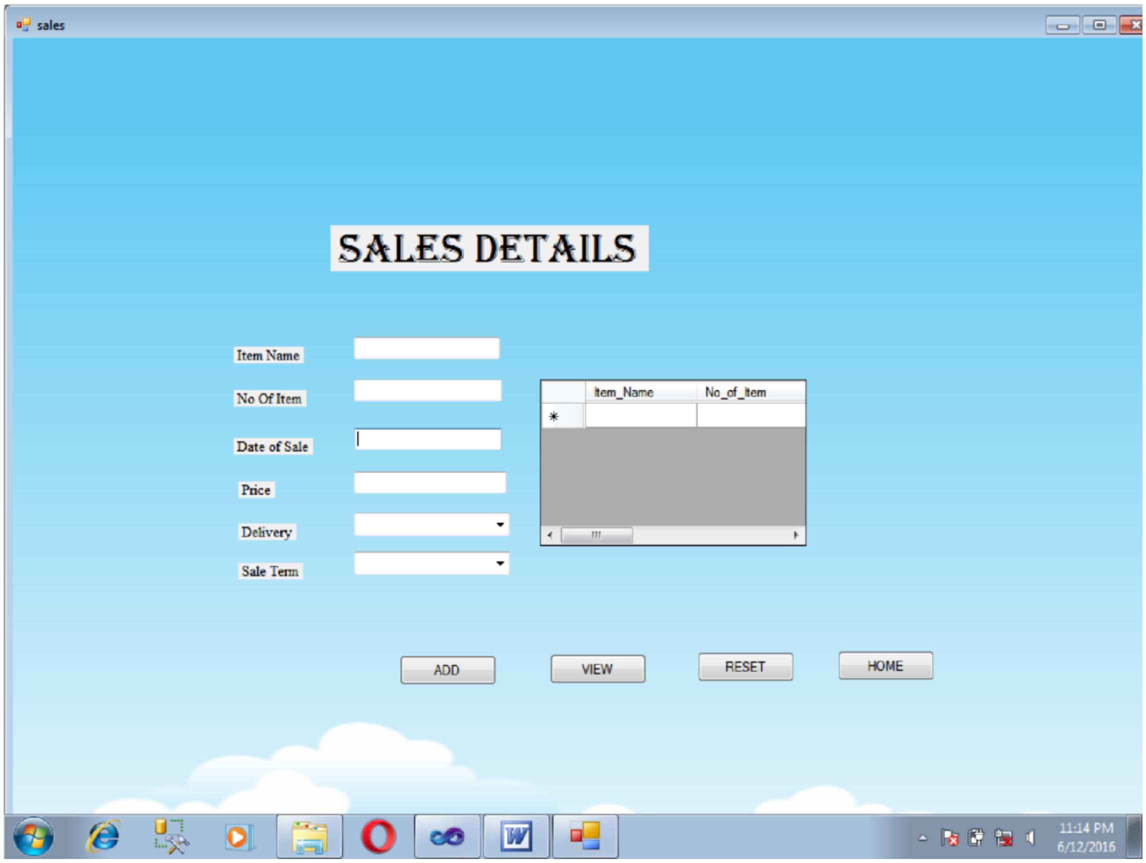
PURCHASE DETAILS

Date	
Product Name	jam
Product Code	122
Batch No	1244
Quantity	22
purchase order value	2000
Delar Name	vicky
Total Amount	20000
Payment Terms	Cash

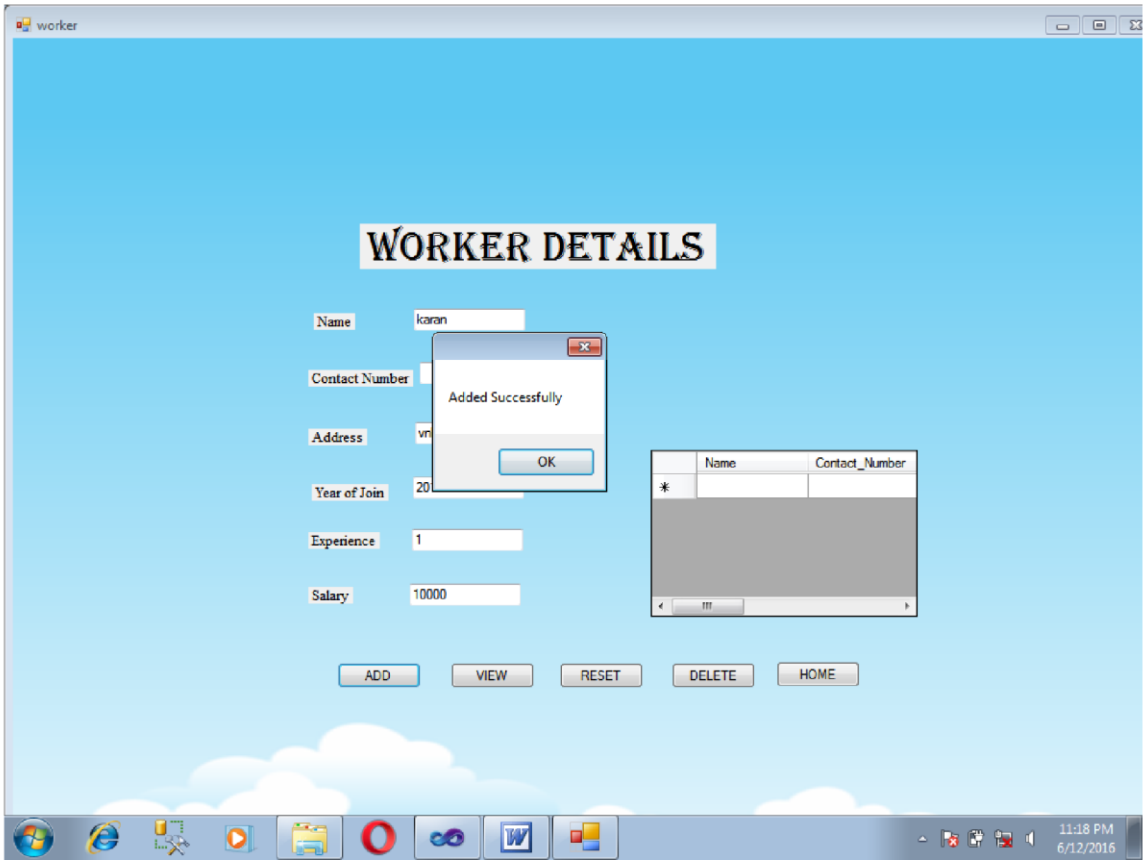
	Date	P_Name
*		

Added Successfully

SALES DETAILS



WORKER DETAILS



Name: karan

Contact Number:

Address: vn

Year of Join: 20

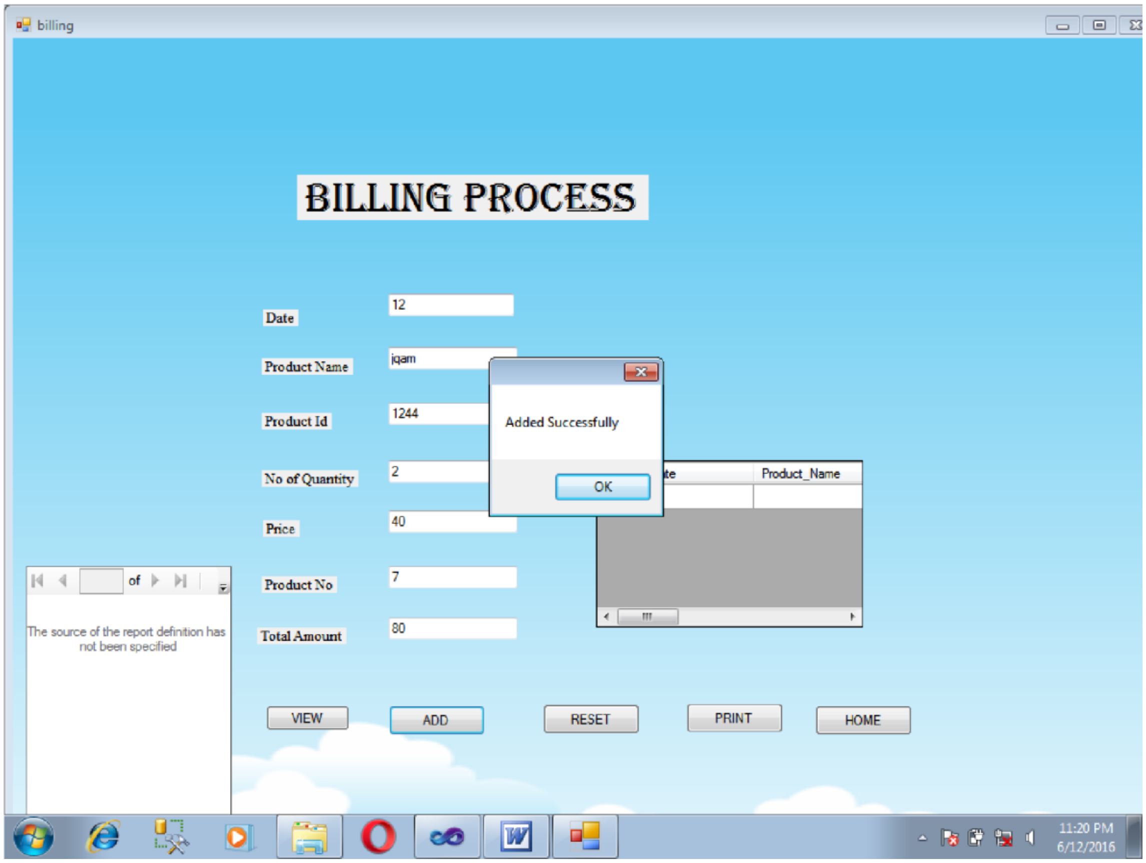
Experience: 1

Salary: 10000

Added Successfully

	Name	Contact_Number
*		

BILLING PROCESS



BILLING PROCESS

Date: 12

Product Name: jqam

Product Id: 1244

No of Quantity: 2

Price: 40

Product No: 7

Total Amount: 80

Added Successfully

The source of the report definition has not been specified

CHAPTER 11

REFERENCES

Web Sites

➢ www.tcs.com

-TCS official website

➢ www.msdn.microsoft.com

-Microsoft official website

➢ http://sourceforge.net/

-Largest repository of open source code and applications available on the internet ➢ http://www.gnu.org

-Web site of the Free Software Foundation (FSF) which supports the open source community

➢ http://www.thefreecountry.com

-Large collection of good programming resources

➢ http://www.1001tutorials.com/

-Large collection of good tutorials for reference.