BCPL

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING

BY

Shantanu Malviya(EN16CS301242)

Shruti Goyal(EN16CS301249)

Sonal Balwani(EN16CS301266)

Under the Guidance of

Prof. (Dr.) Ruchi Patel

Mr. Nischay Goyal

Department of Computer Science & Engineering

Faculty of Engineering

MEDI-CAPS UNIVERSITY, INDORE- 453331

MAY, 2020

BCPL

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING

BY

Shantanu Malviya(EN16CS301242)

Shruti Goyal(EN16CS301249)

Sonal Balwani(EN16CS301266)

Under the Guidance of

Prof. (Dr.) Ruchi Patel

Mr. Nischay Goyal

Department of Computer Science & Engineering

Faculty of Engineering

MEDI-CAPS UNIVERSITY, INDORE- 453331

MAY, 2020

Report Approval

The project work **BCPL** is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the "Project Report" only for the purpose for which it has been submitted.

Internal Examiner: Prof. (Dr.) Ruchi Patel
Name:
Designation:
Affiliation:

External Examiner
Name:
Designation:
Affiliation:

Declaration

We hereby declare that the project entitled **BCPL** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering completed under the supervision of **Mr. Nischay Goyal, HR, Bajrang Cotton Pvt. Ltd. & Prof. (Dr.) Ruchi Patel, Professor, Computer Science department,** Faculty of Engineering, Medi-Caps University Indore is an authentic work.

Further, we declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

Shantanu Malviya(EN16CS301242)
Shruti Goyal(EN16CS301249)
Sonal Balwani(EN16CS301266)

Certificate

We, **Dr. Ruchi Patel, Professor, Computer Science and Engineering & Mr. Nischay Goyal** certify that the project entitled **BCPL** submitted in partial fulfillment for the award of the degree of Bachelor of Technology by **Shantanu Malviya, Shruti Goyal and Sonal Balwani** is the record carried out by them under our guidance and that the work has not formed the basis of award of any other degree elsewhere.

Prof. (Dr.) Ruchi Patel	Mr. Nischay Goyal
Computer Science and Engineering	Human Resource
Medi-Caps University, Indore	BCPL, Barwaha

Prof. (Dr). Suresh Jain

Head of the Department

Computer Science and Engineering

Medi-Caps University, Indore

Acknowledgements

We would like to express our deepest gratitude to Honorable Chancellor, **Shri R C Mittal,** who has provided us with every facility to successfully carry out this project, and our profound indebtedness to **Prof. (Dr.) Sunil K Somani,** Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up our morale. We also thank **Prof. (Dr.) D K Panda,** Dean, Faculty of Engineering, Medi-Caps University, for giving us a chance to work on this project. We would also like to thank our Head of the Department **Prof. (Dr.) Suresh Jain** for his continuous encouragement for betterment of the project.

We express our heartfelt gratitude to our External Guide**, Mr. Nischay Goyal, Bajrang Cotton Pvt. Ltd.** as well as to our Internal Guide, **Prof. (Dr.) Ruchi Patel**, **Department of Computer Science and Engineering, MU**, without whose continuous help and support, this project would ever have reached to the completion.

It is their help and support, due to which we were able to complete the design and technical report.

Without their support this report would not have been possible.

Shantanu Malviya

Shruti Goyal

Sonal Balwani

B.Tech. IV Year
Department of Computer Science and Engineering
Faculty of Engineering
Medi-Caps University, Indore

Abstract

This paper deals with the design of BCPL, in which it is possible to manage pay invoices electronically. The system entitled "BCPL" is a web based application software designed for all type of companies to maintain the bills as per the recent trending of filing tax in the name of Goods and Service Tax (GST) by the new government. This approach is implemented for managing the Sales and Purchase System. In other hand many applications can be realize such as; Customer Creation, Purchase Order Creation and Payment Status can easily be retrieved by the user. Regular Validations are used to apply authentication rules, security and privacy. This system ensures the complete procedure of delivering commodity through a systematic CRM process.

User can input client information that means client details and when its need to generate bill for Client it's just create invoice for client and client receive a bill. When client will pay the bill then its invoice status will show success otherwise pending. There were many problems related to billing system which were responsible for this. The major loophole was not having a systematic customer relationship process. Users were doing paper work for processing the Orders and the billing process wasn't at this level. We have gone through their requirement and created all the modules as per that. They were checking payments as per the invoice one by one, but this process has made the solution for following. It was very important to review the payment status for an organization to make a fruitful organization, so that we have built this module and implemented a process, where user can check all the remaining payments with the Invoice detail. This customer relationship management has made the process easier and executable. So a proper and well defined plan is proposed in this project. Motivated by the above mentioned reasons, here in this project is addressed which aims on modeling and integration of automated billing and order creation system. The system designed here will be connected with the already developed organizations to make their internal process of managing financials stronger and in a very organized way. Customers are the assets of an organization and in this system; we have taken care of managing customers as well. User can create and manager customers at one place with all the required information. Customer code will be unique for every customer.

Each customer connected to each charging station will also have a unique customer code in itself. Each purchase order will have a unique PO No., through which the system can identify the tracking of the order. Every Invoice generated against a dedicated order number, which will

be used to track the actual data, which has been generated through a particular order. The designed system will allow any user to design the billing software based on their needs. This system vanquishes the old stand-alone computer systems and replaces them into unified single software divided into number of modules that roughly approximate the old standalone systems. Thus making it a smart system which can be implemented by big firms and it can be a major breakthrough in deciding the future of customer relationship management.

Table of Contents

		Page No.
	Report Approval	iii
	Declaration	iv
	Certificate	v
	Acknowledgement	vi
	Abstract	vii
	Table of Contents	ix
	List of figures	xi
	List of tables	xii
	Abbreviations	xiii
Chapter 1	Introduction	1
	1.1 Introduction	1
	1.2 About the organization	2
	1.3 Objectives	2
	1.4 Significance	3
	1.5 Source of data	3
Chapter 2		5
	2.1 Experimental Set-up	5
	2.2 Procedures Adopted	6
	2.3 System Feasibility	7
	2.3.1 Economical	7
	2.3.2 Technical	7
	2.3.3 Behavioral	8
	2.4 Platform Specification (Development & Deployment)	9
	2.4.1 Hardware	9
	2.4.2 Software Implementation Language/ Technology	9
Chapter 3	System Analysis	10
	3.1 Information Flow Representation	10
	3.1.1 ER Diagram/ Object Diagram	10

	3.1.2 Data Dictionary	11
	3.1.3 Data Flow/Control Flow/ Process Chart/ Activity Diagram	13
	3.1.4 Use Cases/ Swim lane Diagram/ State Chart Diagram/ Sequence Diagram/Collaboration Diagram	15
	3.1.5 Class Diagram	16
Chapter 4	Design	17
	4.1 Architectural Design	17
	4.1.1 Architectural Context Diagram	17
	4.1.2 Architectural Behavioral Diagram	18
	4.1.3 Description of Architectural Diagram	18
	4.1.4 Control Hierarchy	18
	4.2 Procedural and Modular Approach	19
	4.3 Data Design	19
	4.3.1 Data objects and Resultant Data Structures	20
	4.4 Interface Design	21
	4.4.1 Human-Machine Interface Design Specification	21
	4.4.2 I/O Forms	25
Chapter 5	Testing	27
	5.1 Testing Objective	27
	5.2 Testing Scope	27
	5.3 Testing Principles	29
	5.4 Testing Methods Used	30
	5.5 Test Cases	31
	5.6 Sample Test Data & Results	32
Chapter 6	Limitations	34
Chapter 7	Future Scope	35
Chapter 8	Learning	36
	8.1 Learning during training	36
Chapter 9	Appendix	39
Chapter 10	Bibliography	41

List of Figures

Figure No.	**Figure Name**	**Page No.**
Figure 2.1	Experimental Setup	5
Figure 3.1	ER Diagram	10
Figure 3.2	DFD Diagram	14
Figure 3.3	UML Diagram	15
Figure 3.4	Class Diagram	16
Figure 4.1	Architectural Diagram	17
Figure 4.2	Architectural Behavioral Diagram	18
Figure 4.3	Control Hierarchy	18
Figure 4.4	Data Design Diagram	19
Figure 4.5	Normalization	21
Figure 4.6	Human Machine Interface Design	21
Figure 4.7	Customers Module	22
Figure 4.8	Products Module	23
Figure 4.9	Purchase Order Module	23
Figure 4.10	Invoice Module	24
Figure 4.11	Customer Details Filling	25
Figure 4.12	Product Details Filling	25
Figure 4.13	Purchase Order Filling	26
Figure 4.14	User Details Filling	26
Figure 5.1	Login Form	33
Figure 5.2	Dashboard	33

List of Tables

Table No.	Table Name	Page No.
3.1	Admin Details	11
3.2	Company Details	11
3.3	Product Details	11
3.4	Customer Details	12
3.5	Purchase Order Details	12
3.6	Supplier Details	12
3.7	Bill Details	13
5.1	Sample Test Data Results	32

Abbreviations

Word	Abbreviation
ER	Entity Relationship
DFD	Data flow Diagram
UML	Unified Modeling Language
SQL	Structured Query Language
IIS	Internet Information Sever
OS	Operation System
GB	Giga Byte
PHP	Hypertext Preprocessor
XAMPP	Apache + MariaDB + PHP + Perl
WAMP	Windows + Apache + MySQL + PHP
DB	Database

Chapter 1

Introduction

1.1 Introduction

This project is very popular and widely used in many big organizations for creating various types of Purchase Orders. This is an online system and is used maintain most of the activities happening in big organizations. This project is aimed at developing a system by which the user can create Customers, Purchase Order and Products within the system and can integrate all at one place. The billing system involves receiving billing records from various networks, determining the billing rates associated with the billing records, calculating the cost for each billing record, aggregating these records periodically to generate invoices, sending invoices to the customer, and collecting payments received from the customer. The researchers decided to use the said software, since it is considered as an object-oriented programming tool in developing the system. It will allow the creation of the different files and permit the entry of records into a database designed for the computation of generated Invoices. The language had very powerful database management system, which is necessary in proposed system.

A billing system is a combination of software and hardware that receives call detail and service usage information, grouping this information for specific accounts or customers, produces invoices, creating reports for management, and recording (posting) payments made to customer accounts. Billing systems are composed of interfaces (Network, Marketing, Customer Care, Finance, etc.), computers, software programs and databases of information. Computers are the hardware (computer servers) and operating systems are used to run the programs and process. Network interfaces are the hardware devices that gather accounting information (usage) from multiple networks, convert it into detailed billing records, and pass it on to the billing system.

Billing system use databases to hold customer information; detail records, rate tables, and billing records that is ready to be invoiced. The key functional parts of a billing system include creating usage records, event processing, bill calculation, customer care, payment processing, bill rendering and management reporting. In addition to the basic billing system functions, billing systems share information with many other business functions such as sales, marketing, customer care, finance and operations.

1.2 About the organization

Bajrang Cotton Private Limited is an approved Ginning Pressing Company founded in 2011. The Company is engaged in manufacturing and export of best quality **cotton bales**, **cotton seeds, cotton seed oil and cotton seed cakes** in all principal world market. The company is professionally managed company having large network and infrastructure. It is equipped with state-of-art infrastructure backed by large and fully automatic plant for cotton to cotton bales. Company has its roots since, the beginning of **Cotton Ginning & Pressing** in this region.

They lead the market in their region and have a large workforce. They sell their goods to the spinning industries and process the CCI cotton as they get tenders. All the administrative work is carried out there manually by their administrative team.

Address: Jaimalpura, Maheshwar Road, Barwaha (M.P.)

Company CIN: U17121MP2011PTC025896

Directors: Mr. Sanjay Goyal, Mr. Dilip Goyal

Email: bcplbarwaha@gmail.com

Contact: 07280-290175

1.3 Objectives

We apply technology with innovation and responsibility to achieve these broad objectives:

- Effectively address the business issues our customers face today
- Generate new opportunities that will help them stay ahead in the future
- Time saving process for the organizations to increase their growth rate
- Systematic & digital approach to maintain & execute all the documents at one place
- Users can retrieve any attribute within some clicks, which can save their time

1.4 Significance/Scope

As far as the project is developed the functionality is simple, the objective of the proposal is to strengthen the functioning of Status Monitoring and make them effective and better. The proposed software will cover the information needs with respect to each request of the user group viz. accepting the request, providing vulnerability document report and the current status of the Order. The billing system is a system that helps companies in billing their customers, such as through invoices and online orders. It can also help them to integrate payment terminals and gateways to process online payments and credit cards in future amendments. Billing software is installed on the server, and it gets bound with the control panel of the server. All actions that are performed through it have an immediate effect on the server because of the execution of control panel scripts for account suspension, creation, etc, that are run from the software end.

1.5 Source of Data

Our Source of this information gathered from various mediums. Basically, Source of Data can be classified into 2 types:

Following are the two sources of data:

- **Internal Source**

  We have gone through the work structure of the organization, we are working for. We have done analysis on their internal data and get the appropriate data from the resource person. We run oral workflow and understood the basic requirement of fulfilling the need of the system. We have gone through their commodities and created a workflow of modules as per that.

- **External Source**

  Our External resource comes from the process of managing customer relationship that we have maintained. We have gone through some ongoing systems, which are following this type of process. We have gone through the workflow and understood the modules that they have used. Along with that information, we have implemented our ideas and processed workflow, which is advanced and different from the current systems. We have gone through the information about the commodities or products that the organization is manufacturing. Through a research, we came to know that it is a taxable commodity, so as per that we read the GST guidelines and created our system as per the HSN/SAC mentioned in the GST portals and understood the essentials of that.

Chapter 2

Report on Present Investigation

2.1 Experimental Set-up

The experimental setup is as shown in Figure, which consists of

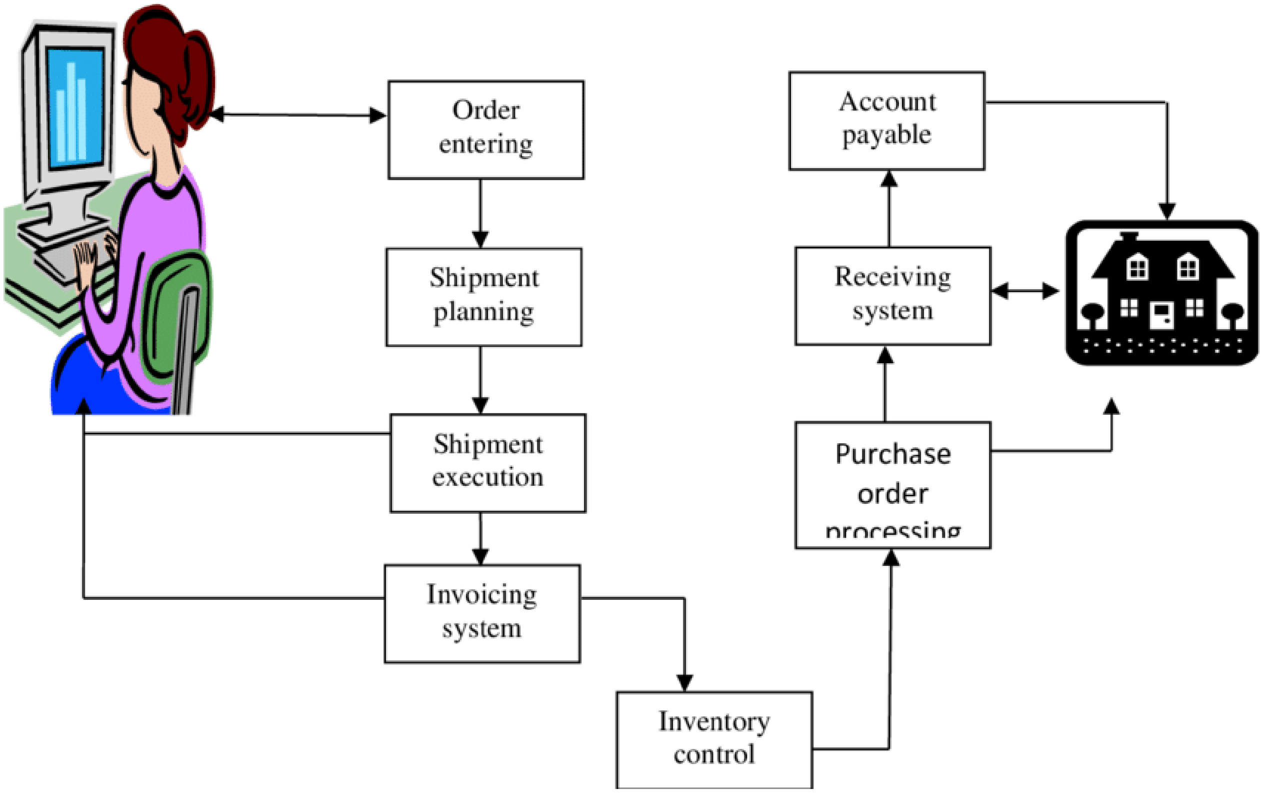


Figure 2.1

Hardware Requirement:

Experimental Setup shows the behavior of the system in real time. For this setup, we need following attributes:

- To make this system, we need a genuine processor based Computer system, on which we can perform the execution
- As we can connect this online, we need an internet connection via Wi-Fi or Mobile Hotspot etc.

- We need commodities that we are going to deal with. As per the commodity based planning, we can create the orders and generate invoice execution
- We should have multiple instances, so that we can check that the system is working online or not
- We need a Printer device as well, so that we can attach the printer to the system and can check whether it is giving printout or not

Software Requirement:

As it is application software, we need to check and ensure all the required software at the time of executing its experimental setup. For the software setup, we need following programs to be checked:

- We need a Xampp and Wamp Server for processing this application
- We need to install the updated Laravel Framework
- PHP Version should be upgraded to avoid the compatibility of the system
- SQL should be installed to implement and execute the data

2.2 Procedures Adopted

To execute the system, we need to adopt some procedures, which are important for a successful implementation.

- **SET YOUR TERMS**

  Before you start working with a client, you should both agree on a set of terms. It’s the basis for good customer relations; it helps you avoid surprises, and increases your chances of getting paid on time.

- **OPTIMIZE THE BILLING PROCESS**

  As it is completely based on an objective of billing, this system needs to get fast billing process and needs to be updated time to time to get better results.

- **AUTOMATE**
  Simple automation features can make your payments faster, and save your, as well as your clients’ time.
- **FORGET SPREADSHEETS, GET IN THE CLOUD!**

In today’s business environment, it’s all about the flexibility and accessibility. If you’re still relying on paper or Excel-based systems to bill your clients, then it’s about time you switched to an online solution and we will try to let our users to get more practice on our system.

2.3 Feasibility Report OF BILLING AUTOMATION SYSTEM PROJECT

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Economical, Technical and Behavioral feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Economic Feasibility
- Technical Feasibility
- Behavioral Feasibility

2.3.1. Economic Feasibility

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economic feasibility for certain.

2.3.2. Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested

- Do the proposed equipment have the technical capacity to hold the data required to use the new system
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users
- Can the system be upgraded if developed
- Are there technical guarantees of accuracy, reliability, ease of access and data security

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

2.3.3 Behavioral Feasibility

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Behavioral feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the behavioral feasibility of a project includes the following: -

- Is there sufficient support for the management from the users
- Will the system be used and work properly if it is being developed and implemented
- Will there be any resistance from the user that will undermine the possible application benefits

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

2.4 Platform Specifications (Development and Deployment)

2.4.1 Hardware Requirements:

Intel Core i3 Processor and Above

RAM 4 GB and Above

HDD 250 GB Hard Disk Space and Above

2.4.2 Software Requirements:

WINDOWS OS (Version 7/8/10)

Xampp and Wamp, PHP Version 7.2

SQL 5.6 and above

Framework: Laravel

Internet Information Server 5.0 (IIS)

Visual Studio .Net Framework (Minimal for Deployment)

Chapter 3

System Analysis

3.1 Information Flow Representation

In system feasibility or system analysis, we first study the feasibility of the system, to find whether the system is beneficial to user and organization or not. The feasibility study is carried out to select the best system that meets performance requirements. Once a relevant member recognizes a need, then the proposed system is with reference to certain parameters relates to our organization.

3.1.1 ER Diagram/ Object Diagram

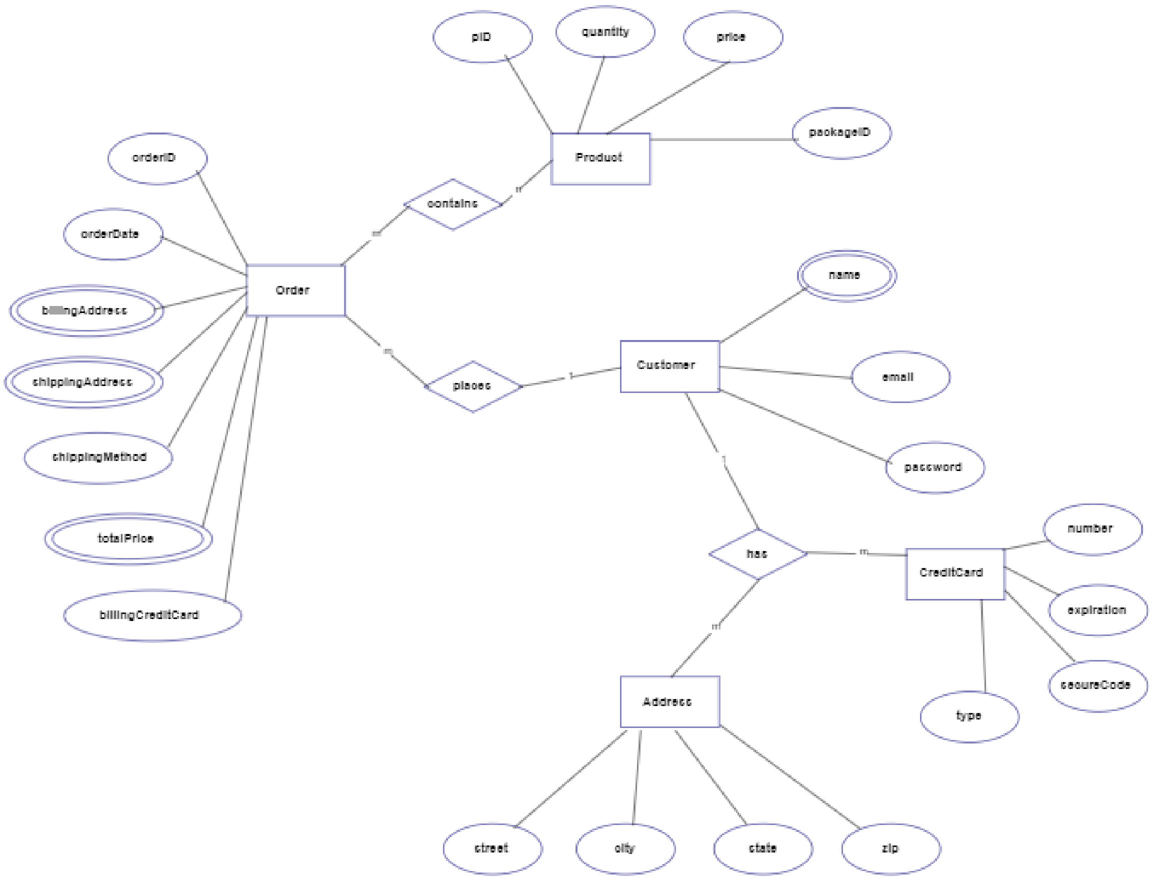


Figure 3.1

3.1.2 Data Dictionary

Admin Details

Field Name	Size	Data-Type	Constraint	Property
Username	10	Varchar	Primary key	Username
Password	10	Varchar	Not null	Password

Company Details

Field Name	Size	Data-Type	Constraint	Property
Co_Id	5	number	Primary key	Company Id
Coname	20	Varchar	Not null	Company Name
Addr	100	Varchar	Not null	Address
Mobile	10	Number	Not null	Mobile number

Product Details

Field Name	Size	Data-Type	Constraint	Property
Pro_Id	5	number	Primary key	Product Id
Proname	20	Varchar	Not null	Product Name
SACCode	15	Varchar	Foreign key	Company name
Batch_No	10	varchar	Not null	Batch number
Rate	10	Number	Not null	Price of product
Pro_type	20	Varchar	Not null	Product type
Mfg_Date	–	Date	Not null	Product Manufacture Date
Exp_Date	–	Date	Not null	Product Expire Date

Customer Details

Field Name	Size	Data-Type	Constraint	Property
Cust_Id	5	number	Primary key	Dealer Id
Custname	20	Varchar	Not null	Dealer’s Name
Addr	100	Varchar	Not null	Address
Mobile	10	Number	Not null	Mobile number
City	15	Varchar	Not null	City
E-mail	30	Varchar	Not null	Email Id
Co_Id	5	Number	Foreign key	Company Id

Purchase Details

Field Name	Size	Data-Type	Constraint	Property
Pur_Id	5	number	Primary key	Purchase Id
Dea_Id	5	number	Foreign key	Dealer Id
Pro_Id	5	number	Foreign key	Product Id
Pur_Date	–	Date	Not null	Purchase Date
Quantity	10	Number	Not null	Quantity
Rate	10	Number	Not null	Price
Total	15	Number	Not null	Total amount

Sales Details

Field Name	Size	Data-Type	Constraint	Property
Sale Id	10	Varchar	Primary key	Sales Id
Pro Id	10	Varchar	Foreign key	Product Id
Quantity	10	Number	Not null	Quantity
Price	10	Number	Not null	Price
Total	10	Number	Not null	Total

Bill Details

Field Name	Size	Data-Type	Constraint	Property
Bill_Id	10	Varchar	Primary key	Bill Id
Pro_Id	10	Varchar	Foreign key	Product Id
Cus_Name	25	Varchar	Not null	Customer Name
Quantity	10	Number	Not null	Quantity
Price	10	Number	Not null	Price
Total	10	Number	Not null	Total
Date	–	Date	Not null	Billing Date

3.1.3 Data Flow

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD'S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists of a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this lead to the modular design.

A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

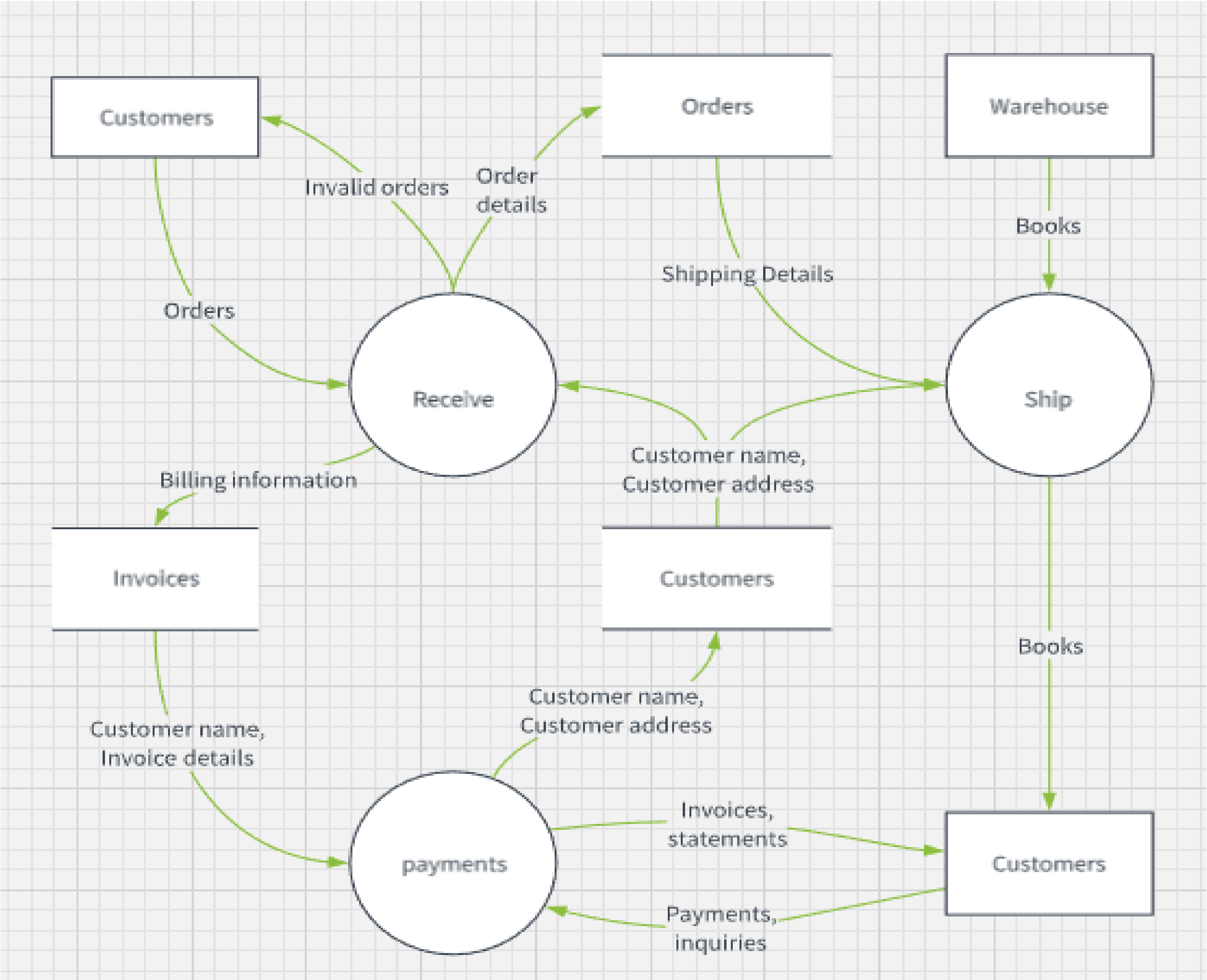


Figure 3.2

3.1.4 Use Case Diagram

User view represents the goal and objective of the system from user point. Our project is based on Billing System, so the main Objective is to manage Sales and Purchase Invoices properly and in a frequent manner.

Use case diagrams show actors and use cases together with their relationships. The use cases represent functionality of a system or a classifier, like a subsystem or a class, as manifested to external integrators' with the system or the classifier.

A use case diagram is a graph of actors, a set of use cases, possibly some interfaces, and the relationships between these elements. The relationship is associations between the actors and the use cases, generalizations between the actors, and generalizations, extends, and includes among the use cases. The use cases may optionally be enclosed by a eclipse that represents the boundary of the containing system or classifier.

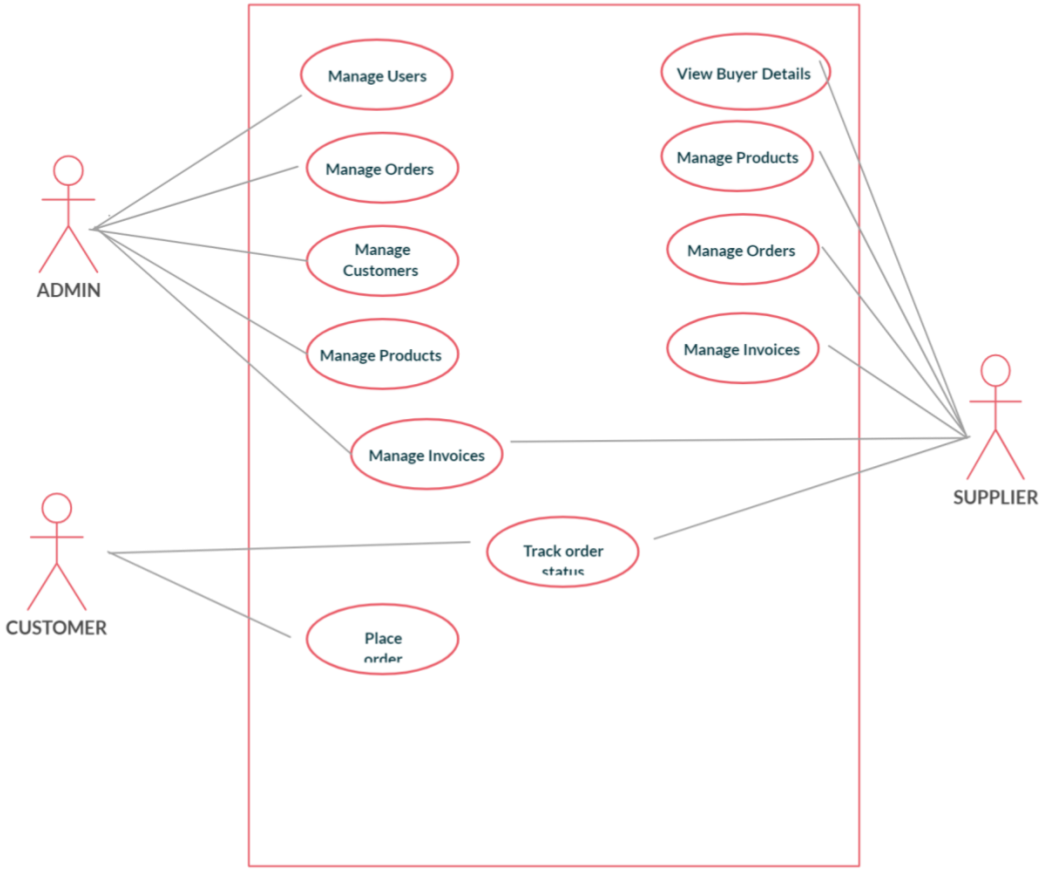


Figure 3.3

3.1.5 Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. In the diagram, classes are represented with boxes that contain three compartments:

The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.

The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.

The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase. In the design of a system, a number of classes are identified and grouped together in a class diagram that helps to determine the static relations between them.

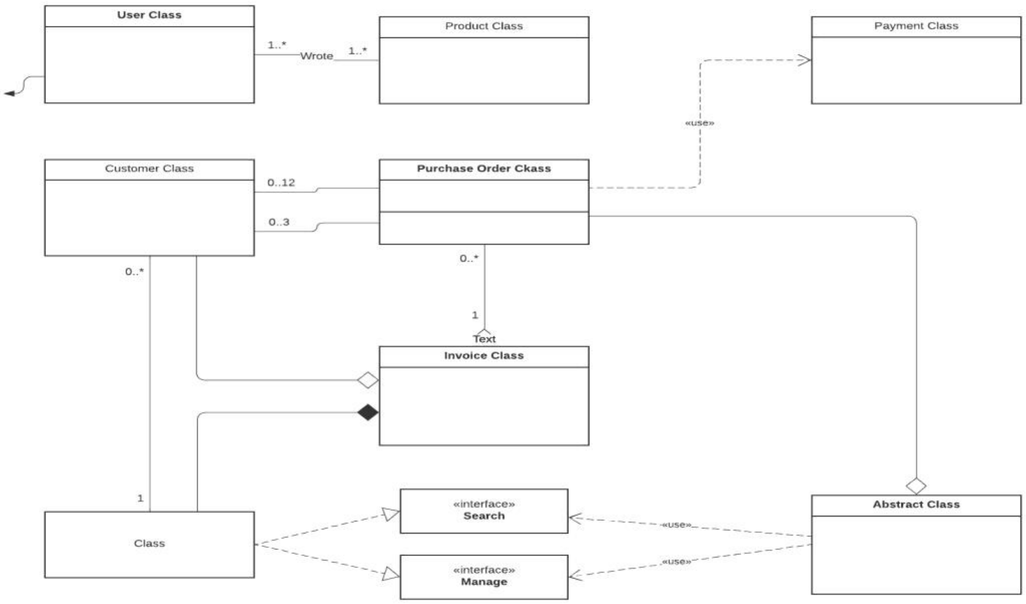


Figure 3.4

Chapter 4

Design

4.1 System Design

System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, procedural design, data design and interface design.

4.1.1 Architectural Design

The Customer Support System application will follow a Four Layer Architecture so that the objects in the system as a whole can be organized to best separate concerns and prepare for distribution and reuse. A principal advantage to this design is the relative stability of the components as seen by the applications developer. Implementations may change considerably to enhance the performance or in response to changes in the architecture.

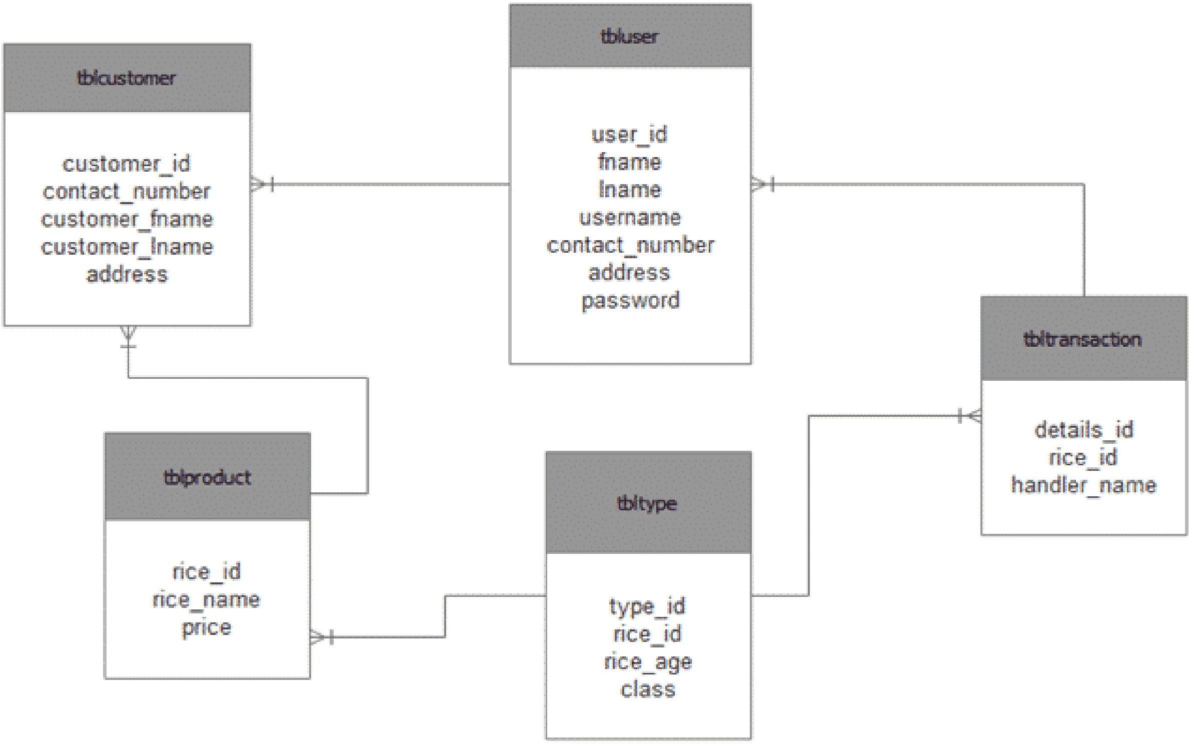


Figure 4.1

4.1.2 Architectural Behavioral diagram

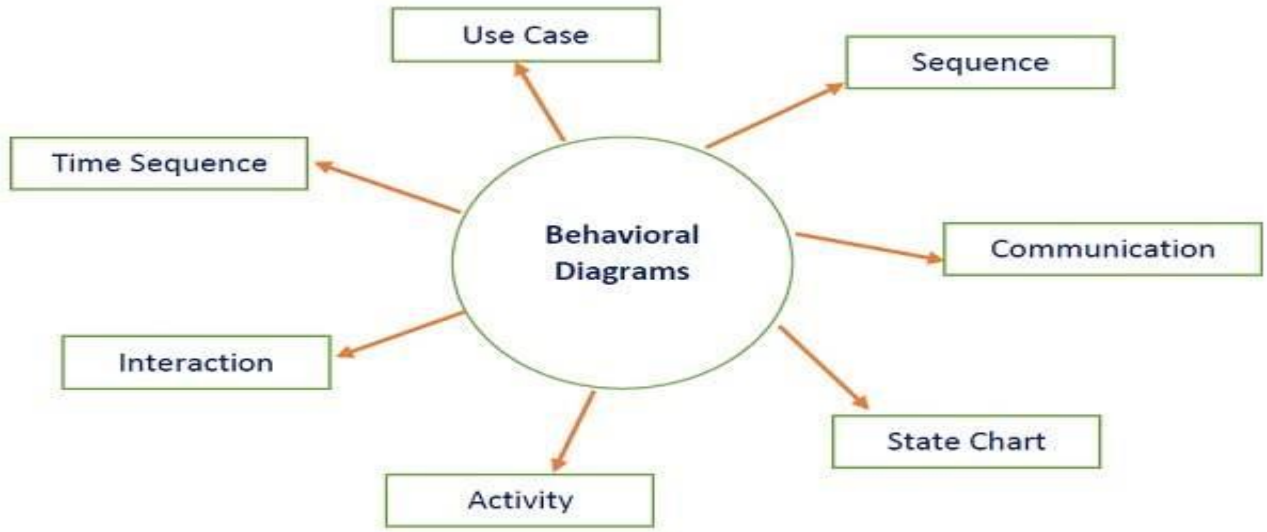


Figure 4.2

4.1.3 Description of Architectural Diagram

Architecture defines the designed process of integrating modules. These modules are responsible for translating data file, byte for byte, into the data fields required for the database. The information parser component interfaces with the biller's existing billing system. The parser extracts the relevant billing information from the legacy billing systems data formats, including fixed length, data tagged, Xerox Line Mode or other file formats.

4.1.4 Control Hierarchy

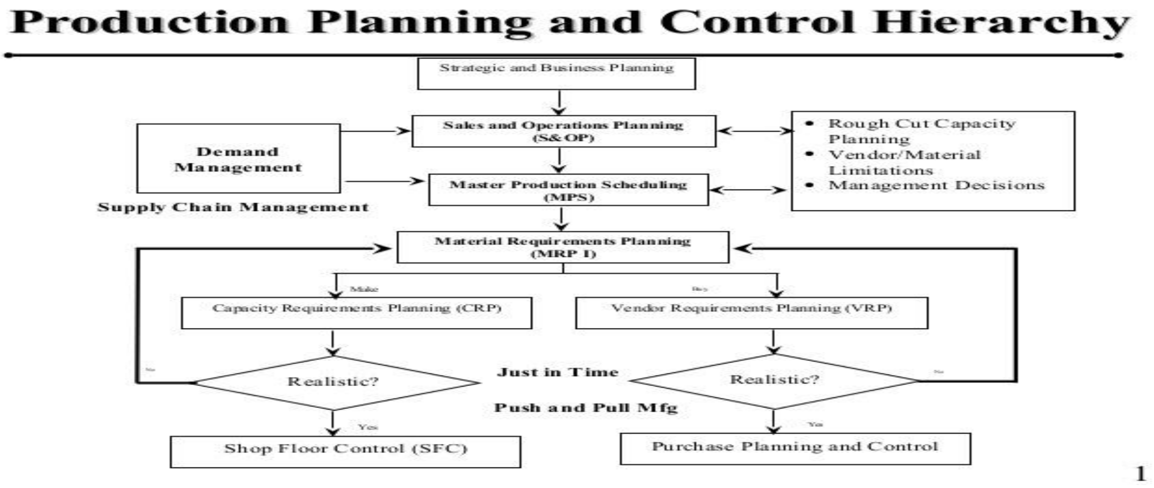

Production Planning and Control Hierarchy
Strategic and Business Planning
Sales and Operations Planning (S&OP)
Demand Management
Master Production Scheduling (MPS)
Supply Chain Management
Rough Cut Capacity Planning
Vendor/Material Limitations
Management Decisions
Material Requirements Planning (MRP I)
Make
Buy
Capacity Requirements Planning (CRP)
Vendor Requirements Planning (VRP)
No
Realistic?
Just in Time
Realistic?
No
Push and Pull Mfg
Yes
Yes
Shop Floor Control (SFC)
Purchase Planning and Control
1

Figure 4.3

4.2 Procedural and Modular Approach

Procedural Approach is a whole life-cycle process. It must be applied at each stage in the Project designing of software process.

OBJECTIVE

It regulates the system development process and the maintenance and modification of systems.

This standard helps to achieve the following objectives:

- Improvement and guarantee of the quality
- The discovery of defects in a system. The assessment of whether or not the system is usable in an operational situation. The completeness of the results to be delivered can best be guaranteed by a standardized procedure
- Project Modules Management
- Defined interim results make early assessment procedures possible. Uniform product contents alleviate the readability of the products and the assessment procedures.
- Checking the costs for the whole lifecycle
- The generation of relevant project-specific development standards and its assessment will be simplified

The standardized procedure makes the cost calculation more transparent.

4.3 Data Design

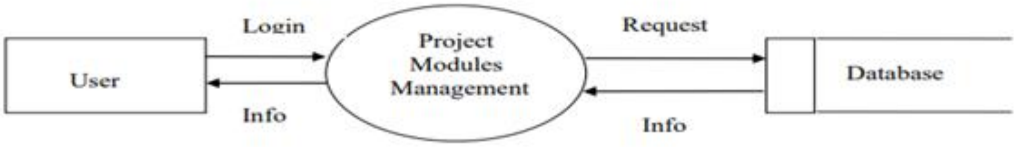


Figure 4.4

4.3.1 Resultant Data Objects and Resultant Data Structures

This is one, which transforms the information domain model created during analysis into one, data structures that will implement the software. The primary activity during this design is to select logical representations of data objects identified during the requirement definition and specification phase. The main tables required for the software are identified during the analysis phase. The tables are designed in such a way so as to store information efficiently and avoid unnecessary redundancy, yet making the retrieval of data easier. Above all, catering to the needs of the application, the database is normalized. The advantages of normalization are: To structure the data so that any pertinent relationships between the entities can be represented. To permit retrieval of data in response to query and report requests. To minimize the need to restructure or recognize data when a new application requirements arise. The database is to be protected from accidental destruction. The organization of the database should be such that it achieves data integrity and data independence.

Normalization Technique:

Normalization provides for table optimization through the investigation of entity relationships. Main purpose of normalization is to avoid Data redundancy and some unforeseen scalability factors. Normalization is done to remove Insertion, Updating and Modification anomalies and redundancy of data. A certain level of normalization of tables in database gives a particular normal form based of particulars steps followed. Database can be normalized up to any defined normal forms according as the need of application and its effectiveness. Database of this project is normalized up to Second Normal Form. Further normalization of database was not considered taking into account the need of application and ease of working with database.

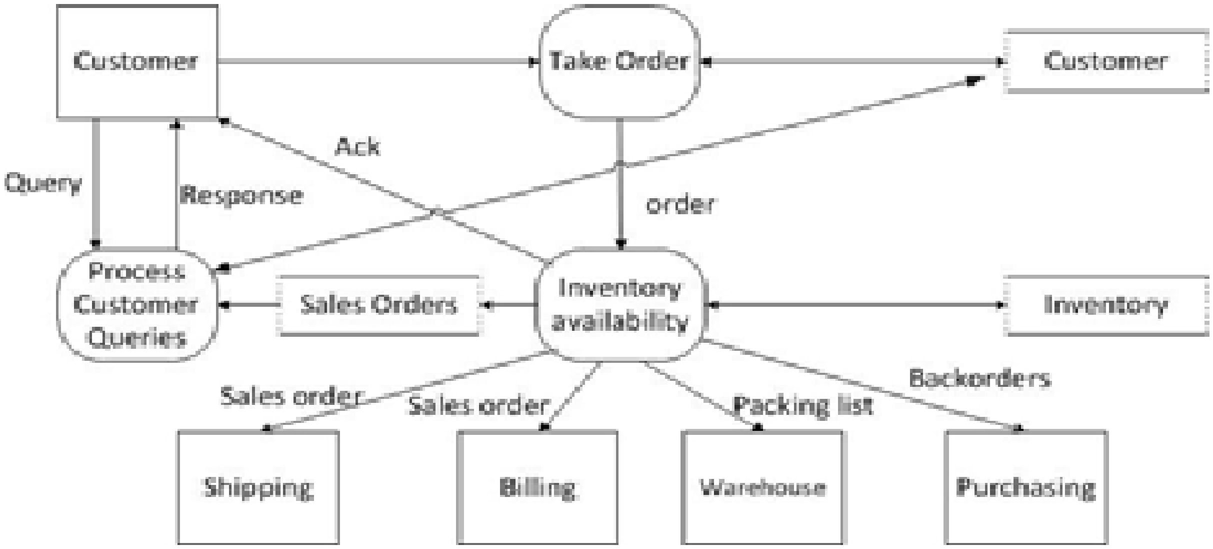
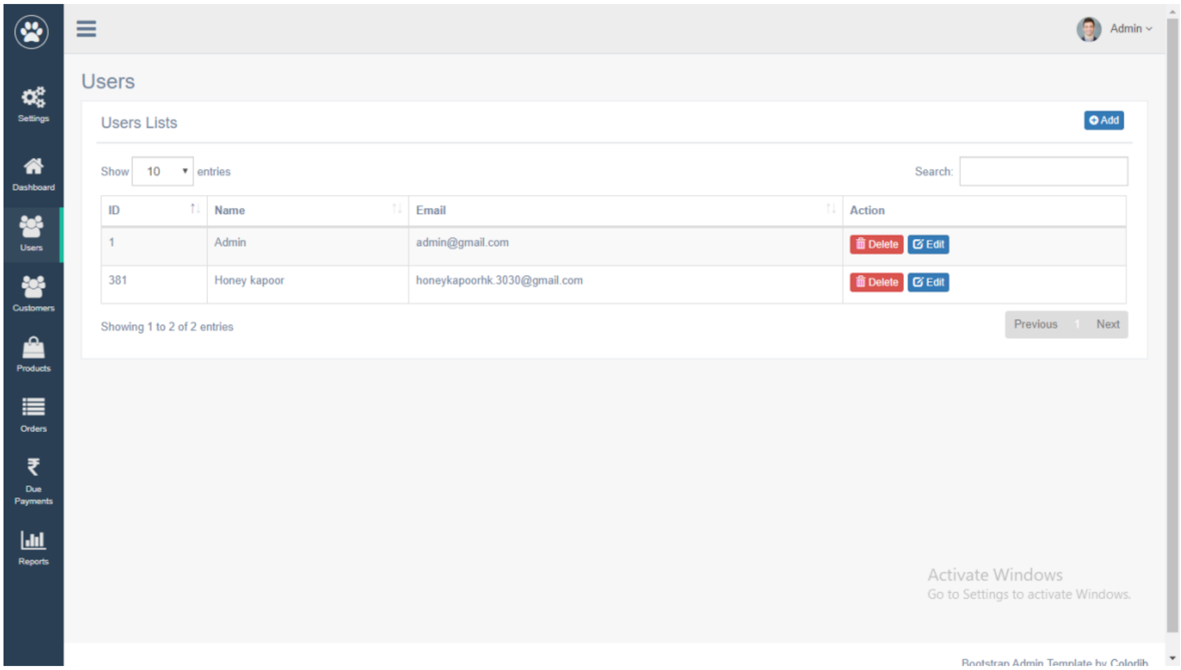


Figure 4.5

4.4 Interface Design

Interface design or refers to the visual layout of the elements that a user might interact with in a website, or technological product. User interface designs must not only be attractive to potential users, but must also be functional and created with users in mind.

4.4.1 Human Machine Interface Design Specification



Users

Users Lists

Show 10 entries

ID	Name	Email	Action
1	Admin	admin@gmail.com	Delete Edit
381	Honey kapoor	honeykapoorhk.3030@gmail.com	Delete Edit

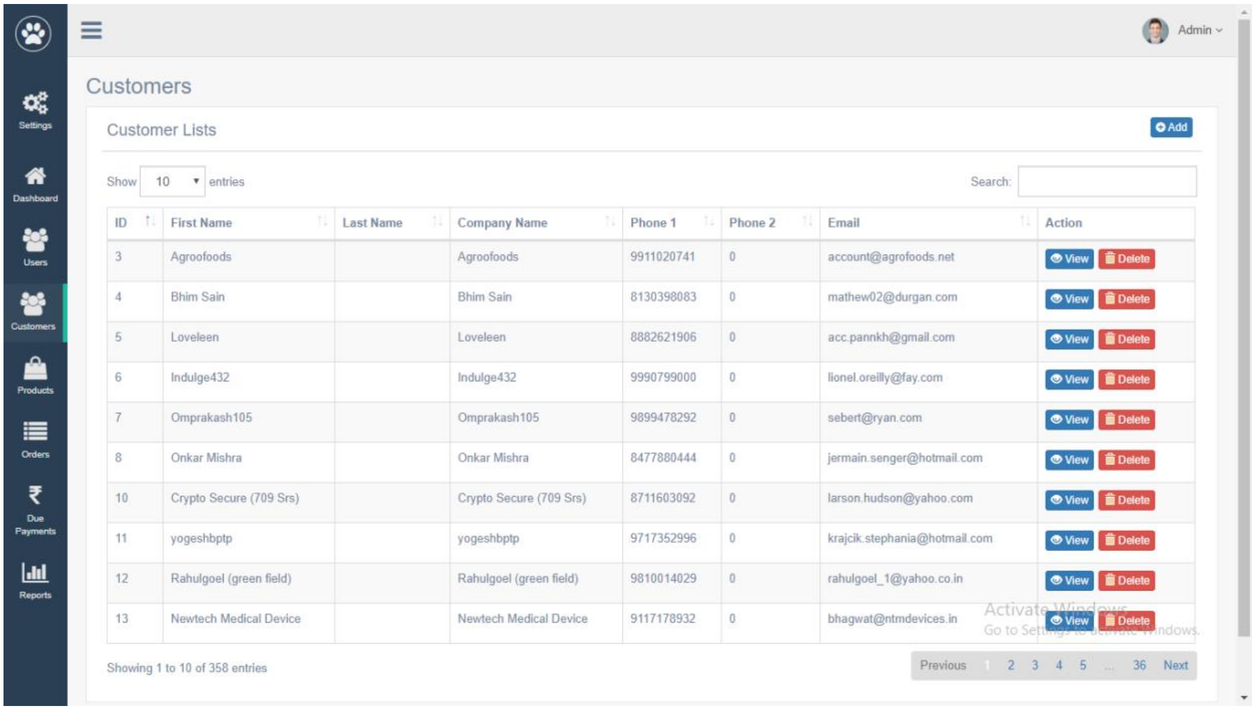
Showing 1 to 2 of 2 entries

Activate Windows
Go to Settings to activate Windows.

Bootstrap Admin Template by Colorlib

Figure 4.6

This figure shows the list of Users, who can login and access the system. We can add, edit and delete the users from here.



Customers

Customer Lists

Show 10 entries Search:

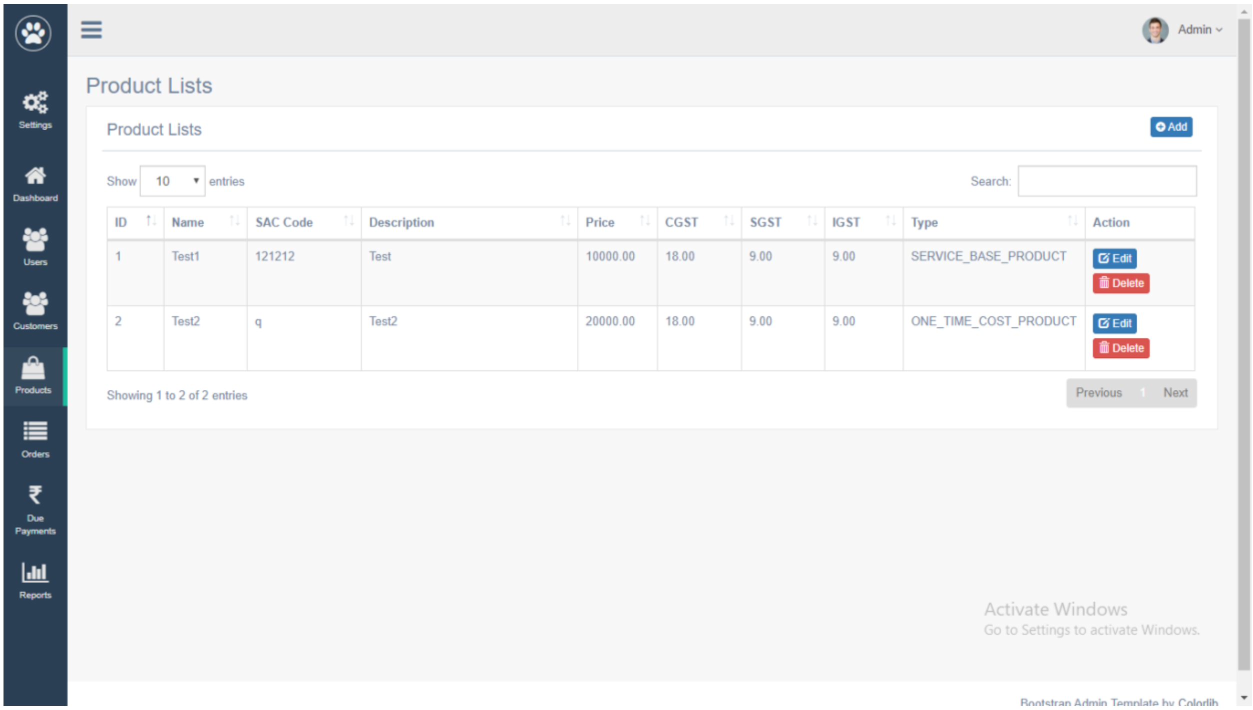
ID	First Name	Last Name	Company Name	Phone 1	Phone 2	Email	Action
3	Agroofoods		Agroofoods	9911020741	0	account@agrofoods.net	View Delete
4	Bhim Sain		Bhim Sain	8130398083	0	mathew02@durgan.com	View Delete
5	Loveleen		Loveleen	8882621906	0	acc.pannkh@gmail.com	View Delete
6	Indulge432		Indulge432	9990799000	0	lionel.oreilly@fay.com	View Delete
7	Omprakash105		Omprakash105	9899478292	0	sebert@ryan.com	View Delete
8	Onkar Mishra		Onkar Mishra	8477880444	0	jermain.senger@hotmail.com	View Delete
10	Crypto Secure (709 Srs)		Crypto Secure (709 Srs)	8711603092	0	larson.hudson@yahoo.com	View Delete
11	yogeshbptp		yogeshbptp	9717352996	0	krajcik.stephania@hotmail.com	View Delete
12	Rahulgoel (green field)		Rahulgoel (green field)	9810014029	0	rahulgoel_1@yahoo.co.in	View Delete
13	Newtech Medical Device		Newtech Medical Device	9117178932	0	bhagwat@ntmdevices.in	View Delete

Showing 1 to 10 of 358 entries

Previous 1 2 3 4 5 ... 36 Next
Activate Windows
Go to Settings to activate Windows.

Figure 4.7

This figure shows the list of customers, along with their Company Name and Contact details. These are the list of the customers, who are going to purchase the product. We can manage the Customers from here. We can add, edit and delete the customers easily.



Product Lists

Product Lists

Show 10 entries Search:

ID	Name	SAC Code	Description	Price	CGST	SGST	IGST	Type	Action
1	Test1	121212	Test	10000.00	18.00	9.00	9.00	SERVICE_BASE_PRODUCT	Edit Delete
2	Test2	q	Test2	20000.00	18.00	9.00	9.00	ONE_TIME_COST_PRODUCT	Edit Delete

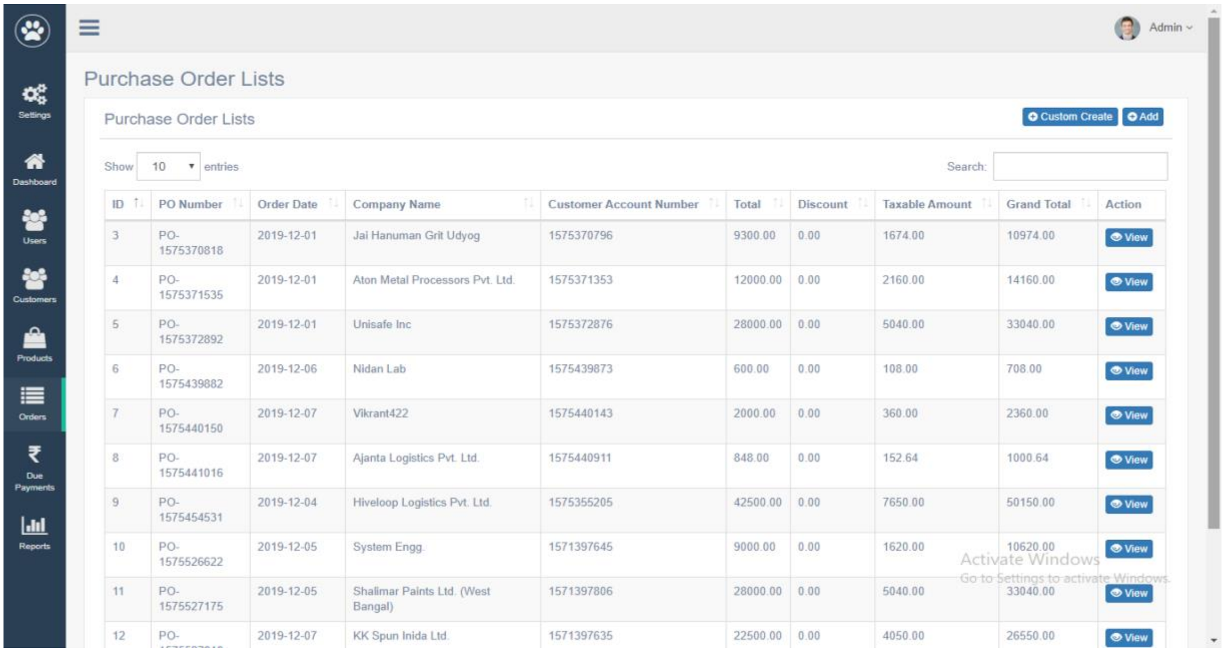
Showing 1 to 2 of 2 entries

Previous 1 Next
Activate Windows
Go to Settings to activate Windows.
Bootstrap Admin Template by Colorlib

Figure 4.8

The above figure shows the list of products

As these are GST based Products, that’s why we have created field for its SAC Code along with the associated Tax details.



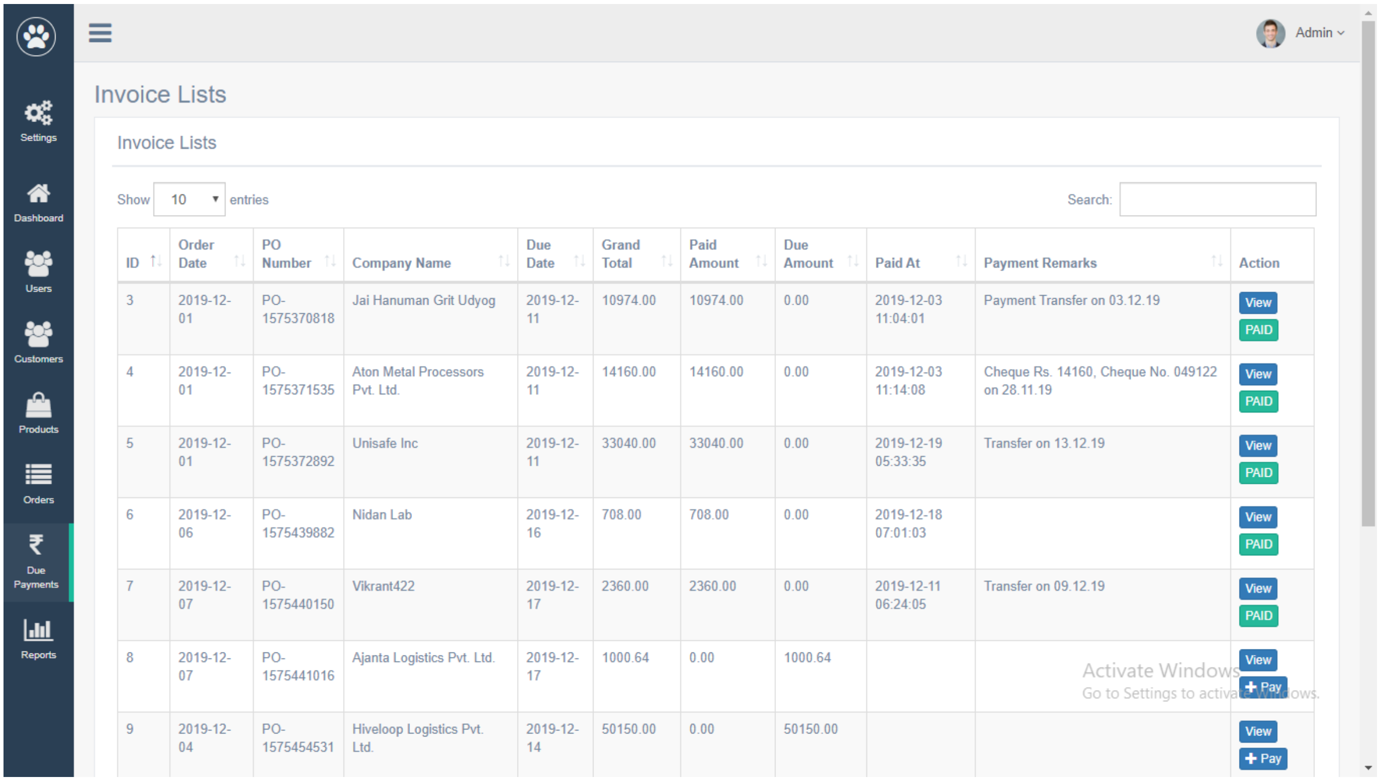
Purchase Order Lists

Purchase Order Lists

ID	PO Number	Order Date	Company Name	Customer Account Number	Total	Discount	Taxable Amount	Grand Total	Action
3	PO-1575370818	2019-12-01	Jai Hanuman Grit Udyog	1575370796	9300.00	0.00	1674.00	10974.00	View
4	PO-1575371535	2019-12-01	Aton Metal Processors Pvt. Ltd.	1575371353	12000.00	0.00	2160.00	14160.00	View
5	PO-1575372892	2019-12-01	Unisafe Inc	1575372876	28000.00	0.00	5040.00	33040.00	View
6	PO-1575439882	2019-12-06	Nidan Lab	1575439873	600.00	0.00	108.00	708.00	View
7	PO-1575440150	2019-12-07	Vikrant422	1575440143	2000.00	0.00	360.00	2360.00	View
8	PO-1575441016	2019-12-07	Ajanta Logistics Pvt. Ltd.	1575440911	848.00	0.00	152.64	1000.64	View
9	PO-1575454531	2019-12-04	Hiveloop Logistics Pvt. Ltd.	1575355205	42500.00	0.00	7650.00	50150.00	View
10	PO-1575526622	2019-12-05	System Engg.	1571397645	9000.00	0.00	1620.00	10620.00	View
11	PO-1575527175	2019-12-05	Shalimar Paints Ltd. (West Bangal)	1571397806	28000.00	0.00	5040.00	33040.00	View
12	PO-	2019-12-07	KK Spun Inida Ltd.	1571397635	22500.00	0.00	4050.00	26550.00	View

Figure 4.9

Above Figure shows the list of Purchase Orders, we get. The users can add/edit/delete the Purchase Orders. It is the Purchase Orders, generated by the Customers, which means that it works like a Sales Order for our Internal Process.



Invoice Lists

Invoice Lists

Show 10 entries

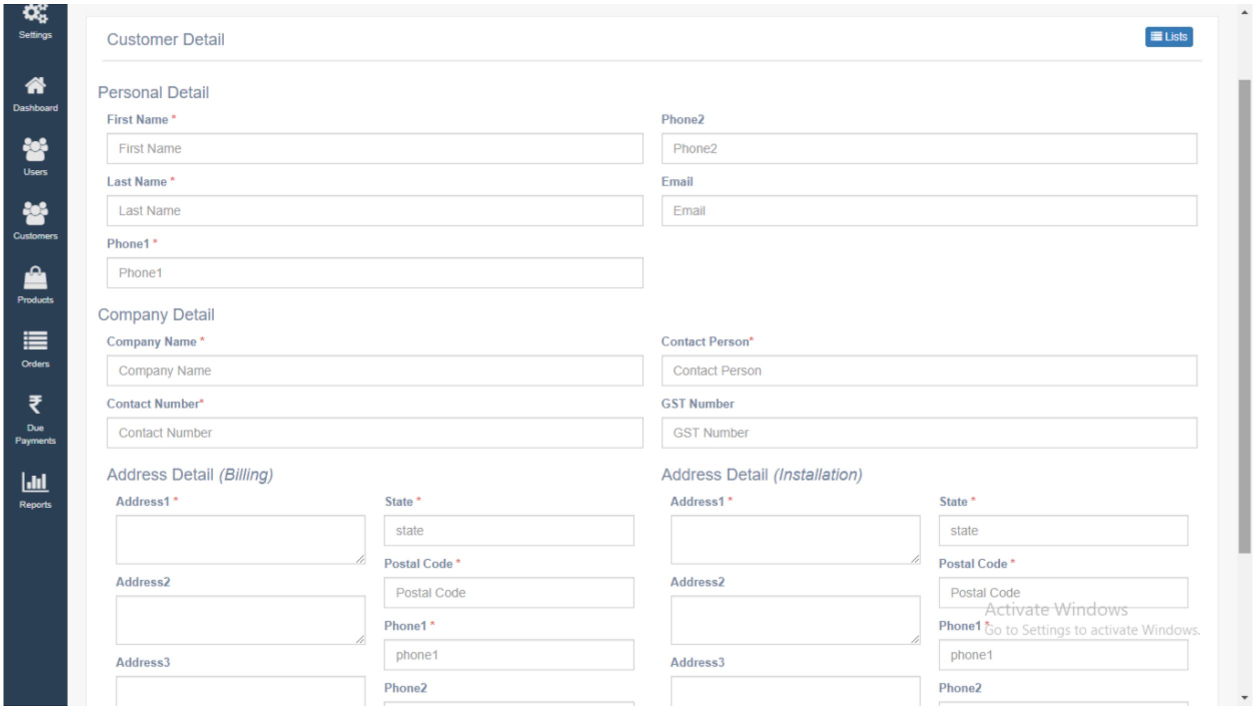
Search:

ID	Order Date	PO Number	Company Name	Due Date	Grand Total	Paid Amount	Due Amount	Paid At	Payment Remarks	Action
3	2019-12-01	PO-1575370818	Jai Hanuman Grit Udyog	2019-12-11	10974.00	10974.00	0.00	2019-12-03 11:04:01	Payment Transfer on 03.12.19	View PAID
4	2019-12-01	PO-1575371535	Aton Metal Processors Pvt. Ltd.	2019-12-11	14160.00	14160.00	0.00	2019-12-03 11:14:08	Cheque Rs. 14160, Cheque No. 049122 on 28.11.19	View PAID
5	2019-12-01	PO-1575372892	Unisafe Inc	2019-12-11	33040.00	33040.00	0.00	2019-12-19 05:33:35	Transfer on 13.12.19	View PAID
6	2019-12-06	PO-1575439882	Nidan Lab	2019-12-16	708.00	708.00	0.00	2019-12-18 07:01:03		View PAID
7	2019-12-07	PO-1575440150	Vikrant422	2019-12-17	2360.00	2360.00	0.00	2019-12-11 06:24:05	Transfer on 09.12.19	View PAID
8	2019-12-07	PO-1575441016	Ajanta Logistics Pvt. Ltd.	2019-12-17	1000.64	0.00	1000.64			View + Pay
9	2019-12-04	PO-1575454531	Hiveloop Logistics Pvt. Ltd.	2019-12-14	50150.00	0.00	50150.00			View + Pay

Figure 4.10

Using above Module, we can see the list of all Invoices at one glance and also, we can see the Payment Status of the following, whether it is paid or unpaid.

4.4.2 I/O Forms



Customer Detail

Personal Detail

First Name * : First Name

Phone2 : Phone2

Last Name * : Last Name

Email : Email

Phone1 * : Phone1

Company Detail

Company Name * : Company Name

Contact Person* : Contact Person

Contact Number* : Contact Number

GST Number : GST Number

Address Detail *(Billing)*

Address1 *

State * : state

Postal Code * : Postal Code

Address2

Phone1 * : phone1

Address3

Phone2

Address Detail *(Installation)*

Address1 *

State * : state

Postal Code * : Postal Code

Address2

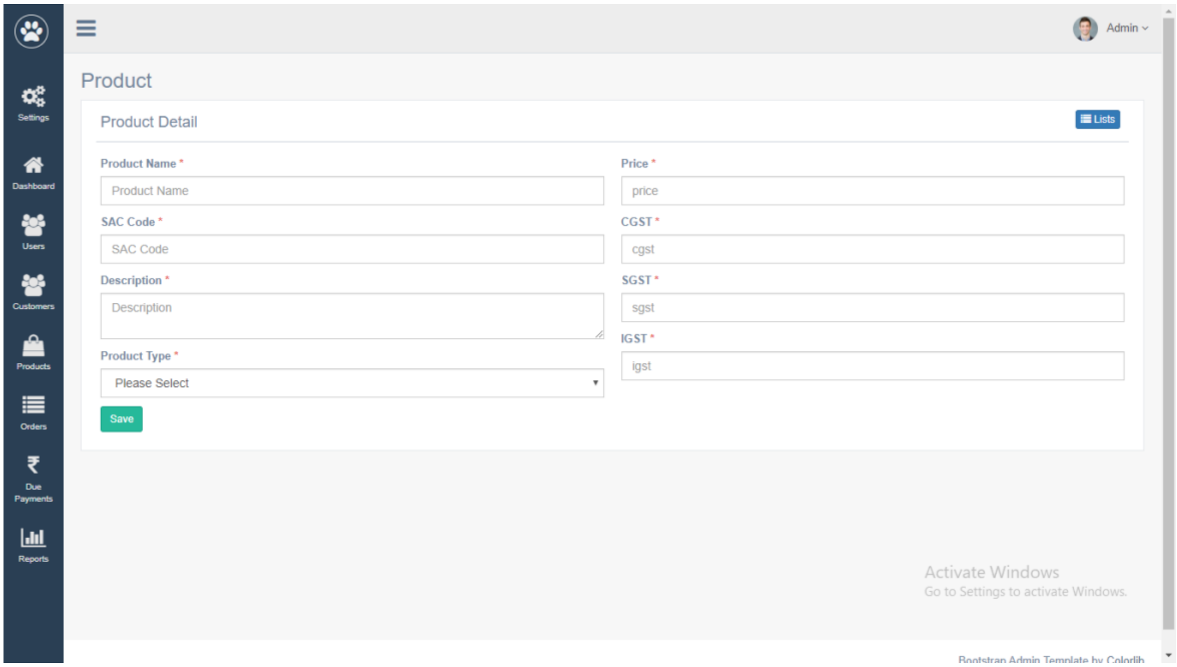
Phone1 * : phone1

Address3

Phone2

Figure 4.11

Above Input Form is designed to Input the details of the Customer.



Product

Product Detail

Product Name * : Product Name

Price * : price

SAC Code * : SAC Code

CGST * : cgst

Description * : Description

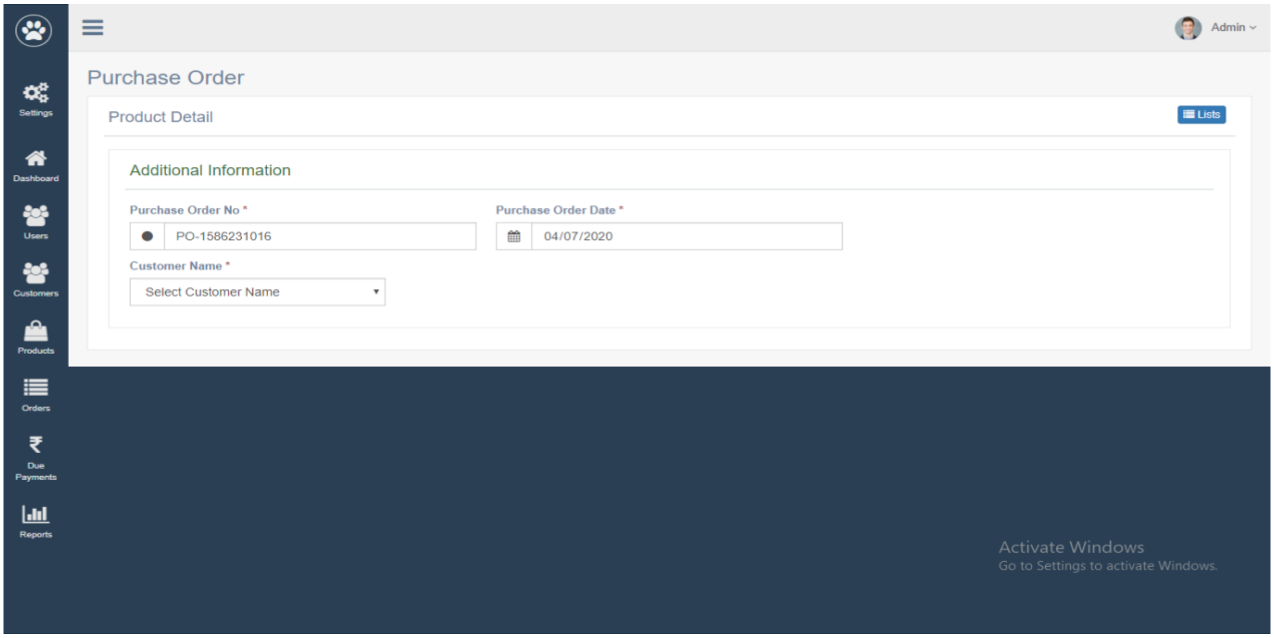
SGST * : sgst

IGST * : igst

Product Type * : Please Select

Figure 4.12

Above Form is the Input form for adding a Product.



Purchase Order

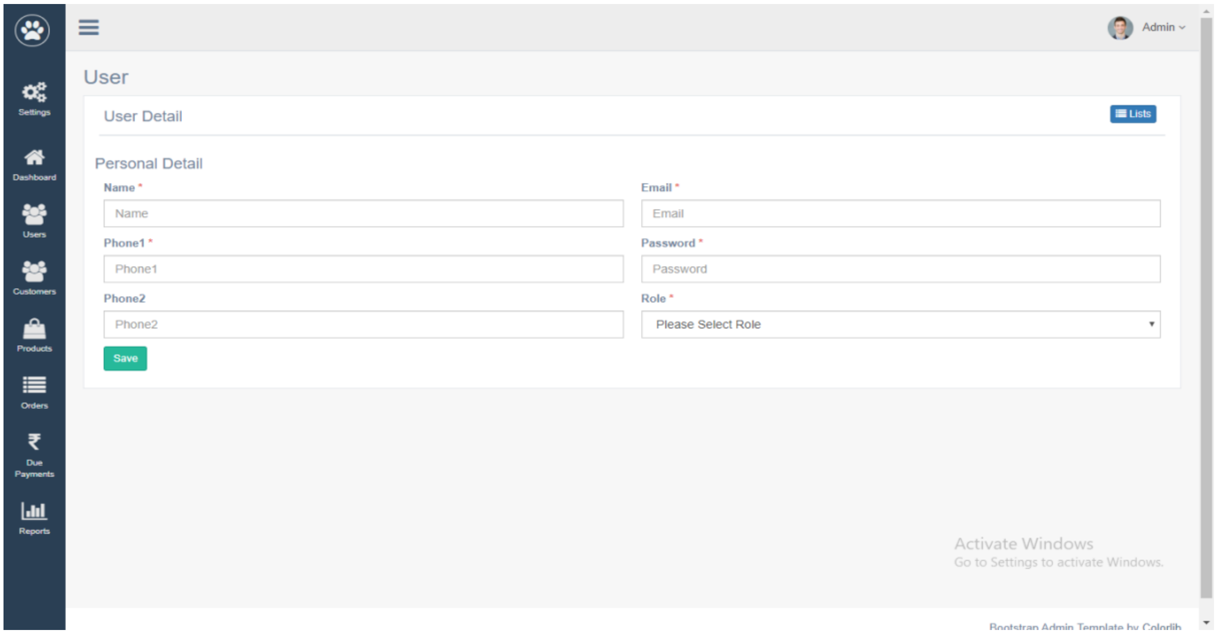
Product Detail

Additional Information

Purchase Order No *	Purchase Order Date *
● PO-1586231016	04/07/2020
Customer Name *	
Select Customer Name ▾	

Figure 4.13

From the above Form, User can create purchase order by selecting the Customer directly. It is linked form, based on the list of Customers



User

User Detail

Personal Detail

Name *	Email *
Name	Email
Phone1 *	Password *
Phone1	Password
Phone2	Role *
Phone2	Please Select Role ▾

Figure 4.14

Above form is simply an input form for registering the User.

Chapter 5

Testing

5.1 Testing objective

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

5.2 Testing Scope

Testing is a process of executing a program with the intent of finding error. Debugging is the process of locating the exact cause of an error and removing that cause. Testing area is concerned with verifying that the correct solution to the problem, embodied in the statement of the requirements, has been developed .Testing is a multi-stage process that consists of activities for validating the software product, from the most primitives elements up to the fully integrated system.

This area includes activities such as:

- Unit testing
- Integration testing
- System testing
- Performance testing
- Module Testing
- Black Box Testing

- White Box Testing

Verification is the process of proving that the developed code can be derived formally from the Specification .Validation is the process of demonstrating to client that the software performs the process required by the client.Verification may be done in a formal way, using the prepositional calculus to prove that code implements the specification correctly. Verification may also be done in a less rigid way, by reasoning gets difficult, this may be a sign that the code as it is being developed.The objectives is to write correct readable code in the first place .if the reasoning gets difficult this may be a sign that the code can be improved .Validation unfortunately cannot be completed without operational software .

Prior to implementation, the best I can do is communicated to client and user our understanding of the problem and what our solution will achieve .We can do this to some extent by providing incomplete models through rapid prototyping or incomplete implementation through incremental development.

Software testing is a critical element of software quality assurance & represents the ultimate review of specification, design & code generation. Once source has been generated, software must be tested to uncover & correct as many errors as possible before delivery to the customer. Our goal is to design a series of test cases that have a high likelihood of finding errors. That's why software-testing techniques come into the picture. These techniques provide systematic guidance for designing test that:

1. Exercise the internal logic of software components

2. Exercise the input & output domains of the program to uncover errors in program function, behavior & performance.

Testing of the software leads to the uncovering of errors in the software functional & performance requirements are met. Testing also provides a good indication of software reliability as software quality as a whole. The result of the different phases of testing are evaluated & then compared with the expected results. If the errors are uncovered they are debugged & corrected.

A strategy approach to software testing has the generic characteristics:

1. Testing begins at the module level & works “outwards” towards the integration of the entire computer based system

2. Different testing techniques are appropriate at different points of time

3. Testing and Debugging are different activities, but debugging must be accommodated in the testing strategy

A strategy for the software testing must accommodate low level tests that are necessary to verify that a small source code segment is performing correctly according to the customer’s requirements and that of developer’s expectations. We test our project, we had first gone for “unit testing” strategy in which we have tested the functionality of each functions, after that we performed “Integration Testing” where we integrate them all and tested them together.

5.3 Testing Principle

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing proceeds by moving outwards along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.

5.4 Testing Methods Used

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

1. WHITE BOX TESTING

This type of testing ensures that

· All independent paths have been exercised at least once

·All logical decisions have been exercised on their true and false sides

·All loops are executed at their boundaries and within their operational bounds

·All internal data structures have been exercised to assure their validity

To follow the concept of white box testing we have tested each form we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, all loops are executed on their boundaries.

2. BASIC PATH TESTING

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were: Use the design of the code and draw correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula:

$V(G)=E-N+2$ or

$V(G)=P+1$ or

$V(G)=$Number Of Regions

Where $V(G)$ is Cyclomatic complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

3. CONDITIONAL TESTING

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

4. DATA FLOW TESTING

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested

5. LOOP TESTING

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

- All the loops were tested at their limits, just above them and just below them
- All the loops were skipped at least once
- For nested loops test the inner most loop first and then work outwards
- For concatenated loops the values of dependent loops were set with the help of connected loop
- Unstructured loops were resolved into nested loops or concatenated loops and tested as above

5.5 Test case design

Test case design for Admin Log in page:

Input:-The admin will enter the user name and password.

Expected Output:- If password is correct then it will perform operation and if not then error message will show up.

Test case design for Adding New User /Customer/Product:

Input:-The user will enter the required details for creating User or Customer or Product.

Expected Output:- The user will get the list of all User or Customer or Product in a list view and user can add/edit/delete from the same particular module.

Test case design for Purchase Order Creation:

Input:-The Order details will be entered and will submit the Purchase Order

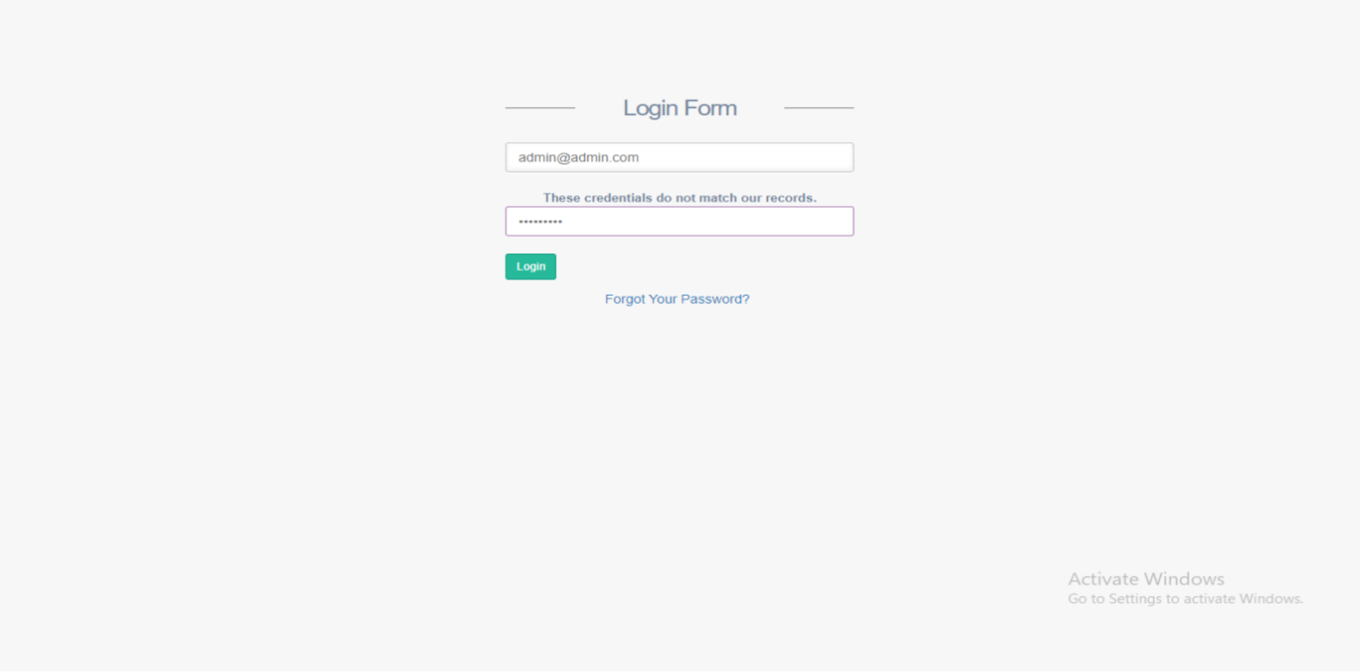
Expected Output:- It will get shown in the list of Purchase Orders and user can easily add/edit and delete it.

5.6 Sample Test Data & Results

Data for Admin log in

Input	Expected Output	Result
Entered id	Entered the account	Entered the account
Wrong id	Invalid id	Invalid id
No Id and password given	Username and Password is required	Username and Password is required

If you entered wrong Id or password, check below figure.



Login Form

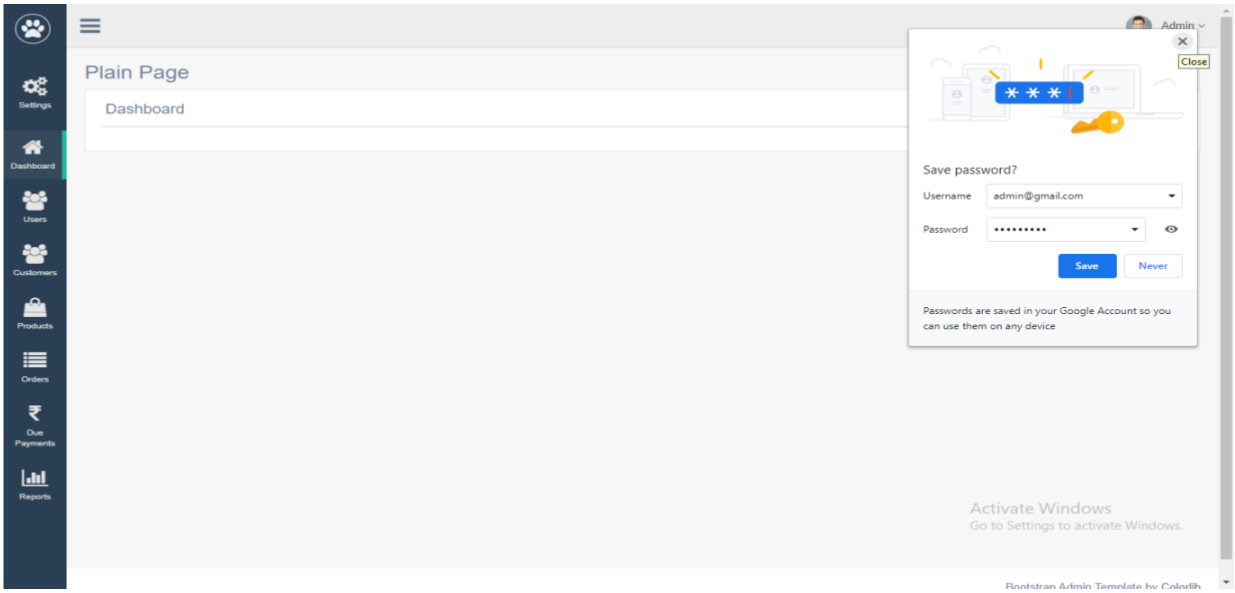
admin@admin.com

These credentials do not match our records.

•••••••••

Figure 5.1

If you entered correct Id and password, check below figure.



Plain Page

Dashboard

Save password?

Username admin@gmail.com

Password •••••••••

Passwords are saved in your Google Account so you can use them on any device

Figure 5.2

Chapter-6

Limitations

Limitations:

- The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity
- Training for simple computer operations is necessary for the users working on the system
- Understanding of Sales and Purchase is very necessary to run this System
- Customer’s information should be correct and more appropriate to avoid the incomplete detailed system

Chapter 7

Future Scope

Future Scope:

- This system can be converted to ” Text to speech ”
- It can be converted into Mobile Application
- It can be implemented as the Professional POS System for any Organization, where dealing of Sales and Purchase take place

Chapter 8

Learning

8.1 Learning during training:

The initial stages of the internship involved learning about the language PHP, it’s framework Laravel and their implementation.

Laravel is an open-source PHP framework, which is robust and easy to understand. It follows a model-view-controller design pattern. Laravel reuses the existing components of different frameworks which helps in creating an application. The application thus designed is more structured and pragmatic. Laravel is accessible, yet powerful, providing powerful tools needed for large, robust applications. A superb inversion of control container, expressive migration system, and tightly integrated unit testing support give you the tools you need to build any application with which you are tasked.

Key learning areas-

1. MVC Support and Object-Oriented Approach: The first and best advantage of using the Laravel framework is that it follows - Model, View, and Controller-based architectural pattern and it has an expressive beautiful syntax which makes it object-oriented.

2. Built-In Authentication and Authorization: Laravel provides an out-of-the-box configuration for the Authentication and Authorization system. That is, in just a few artisan commands your application will be equipped with secure Authentication and Authorization.

3. Packaging System: packaging system deals with the multiple support software or libraries that help the web application to automate the process. Laravel uses a composer as a dependency manager, which manages all the information needed to manage packages. Packages are a great way to accelerate development is to provide the functionality we need out of the box. Image, Laravel Debug bar and Laravel IDE helper are some of the best Laravel packages.

4. Multiple File System: Laravel also has a built-in support for the cloud storage system such as Amazon S3 and Rack space Cloud Storage and of course for local storage. It's amazingly simple to switch between these storage options as the API remains the same for each system. One can use all three systems in one application to serve files from multiple locations like in a distributed environment.

5. Artisan Console: Laravel has its own command line interface called as Artisan. Common uses of Artisan include publishing package assets, managing database migrations, seeding and generating boilerplate code for new controllers, models, and migrations. This feature frees the developer from creating proper code skeletons. One can extend the functionality and capabilities of Artisan by implementing new custom commands.

6. Eloquent ORM: The Eloquent ORM is Laravel's built-in ORM implementation. Laravel has the best Object-relational Mapper as compared to the other frameworks out there. This Object-relational mapping allows you to interact with your database objects and database relationships using expressive syntax.

7. Templating engine: Laravel comes with the inbuilt template engine known as Blade Template Engine. Blade templating engine combines one or more templates with a data model to produce resulting views, doing that by transpiring the templates into cached PHP code for improved performance. Blade also provides a set of its own control structures such as conditional statements and loops, which are internally mapped to their PHP counterparts.

8. Task Scheduling: Scheduler, introduced in Laravel 5.0, is an addition to the Artisan command-line utility that allows programmatic scheduling of periodically executed tasks. Internally, scheduler relies on the cron daemon to run a single Artisan job that, in turn, executes the configured tasks.

9. Events and Broadcasting: Laravel has a concept named broadcasting which is useful in the modern web application to implement real-time data, showing live feeds,etc. Broadcasting allows you to share same event name between your server-side and client-side, so you will able to pull real-time data from the application.

10. Testing: When it comes to the testing of the application Laravel by default provides the unit test for the application, which itself contains tests that detect and prevent regressions in the framework. Integration of PHP unit such as a testing framework is very easy in Laravel application. In addition to that unit tests can be run through the provided artisan command-line utility.

Chapter 9

Appendix

Appendix A:

BCPL with the purpose of integrating various business processes. Within the Organization and to provide the pat, which move towards the direction of integrated system.

The System analyses current application capability against business, technology needs and best practices.

- Identify present and future needs
- Identify wins and successes (strengths)
- Identify pain-points and challenges (weaknesses)
- What are the risks and opportunities
- Identify areas where the current application is capable of meeting needs
- Identify the pros and cons of using the current application to fulfill these needs
- Estimate the time and costs to implement and support
- Compare these costs to alternatives
- Conduct an overall evaluation to determine Solution providers
- Benchmark support and maintenance costs
- Determine if the current application allows the ability to respond quickly and cost effectively to changing business and technical needs

In Scope

- Selecting an evaluation consulting firm
- Perform an analysis of our current enterprise environment
- Perform gap analysis of current application
- Deliver a comprehensive evaluation report

Out of Scope

- Selecting a new Billing Solution for the existing environment

Appendix B:

There are some Mandatory Technical Specifications, as follows:

- **B.1** The solution shall be designed with Service Oriented Architecture. The solution shall be designed based on Component-based approach. It shall be highly granular and loosely coupled to ensure that the failure of one component does not cascade to others
- **B.2** The solution shall be designed on web based architecture
- **B.3** The solution shall be horizontally and vertically scalable and also have virtualization capability
- **B.4** The solution shall be designed with Open Industry Standards and not with Supplier's proprietary protocol
- **B.5** The directory services shall be based on commonly accepted application protocol like LDAP
- **B.6** The proposed solution should be based on WS-* specifications (Web services specifications) & unified access framework compliant to W3C (World Wide Web Consortium) specifications
- **B.7** The application shall provide the functionality to configure the parameters to define the business rules with the application. These parameters shall not be hard-coded in the application
- **B.8** The solution shall provide the functionality to configure the roles & responsibilities and grant role based access to the users. Also, the system shall have the capability to integrate with various standard SSO (Single Sign-On) or IDM (Identity Management) applications
- **B.9** The solution shall use an integration middleware layer so that all required external systems shall be integrated on a continuous basis. The solution shall conform to the requirements specified
- **B.10** The solution shall provide the functionality to encrypt the data stored or transmitted data

Chapter 10

Bibliography

- https://tutorialspoint.com

- https://www.w3schools.com/
- https://stackoverflow.com/
- http://www.google.com/
- https://laravel.com/