CHAPTER 1
INTRODUCTION

1.1. INTRODUCTION

The Mobile Shop Management System is developed for desktop systems to facilitate mobile shop owner's management of customer details and inventory data, which will include mobile phones. It can be used efficiently for physically separated shops in different locations. This application will provide in a simple and easy to operate user interface, which can be managed by any user without having prior in-depth knowledge of the computer system. One can use this application to get a sales report. This application is a complete package for small organizations which will allow them to keep track of their sales and inventory, and provide a computerized billing system. There are various applications with more complex implementation and features available in the market, but they are generally very expensive. Therefore, creating an application with the basic requirement of low cost is essential for small organizations. This application will allow Shops to manage customer details, keep inventory of all products and purchase information, in a very simple way, using a state-of the-art application..

1.2 BACKGROUND AND MOTIVATION

The concept of the Mobile Store Management system has been around for a long time, but it is still in the phase of discussion and design. Initially, all inventory and billing reports were managed manually by shop owners/employers using ledger- based systems. This requires a significant amount of time due to repeated access of the data. There is a high risk of lost or stolen data in that system. Storing old data is also one big factor. Stores have to spare one separate room to store this information. Paper- based documents might lose their information with time, and after some years we can't really read them at all. So the Mobile Shop Management System is designed to reduce paper- based data storage system and provide digital touch to billing and inventory system.

1.3 MARKET

This application application is targeted for small and medium retail stores who want to transform their paper- based inventory, sales, and procurement system to a computer- based system. This is an inexpensive and easy-to-use application application for easy transition to digital media. Also, this system is simple to install and maintain in PC/Laptops, thus avoiding huge investments on enterprise or other types of servers. Currently, there is only one user for this application, who will

also be the administrator. The system administrator will have complete access to the system configuration and data. The system administrator will also have access to other partner stores. The benefits drawn from the system and low cost for installation and maintenance come as a huge advantage.

1.4 Objectives

In today's market, retailers and wholesale outlets should quickly adapt to the ever-changing technology to minimize overhead, lower cost of operation, and help to stay competitive. Everybody needs application, which can facilitate Shop operations and make their day-to-day lives much easier.

Mobile Shop Management System is application designed to take advantage of today's technology and reduce or avoid the burden of storing data on paper and in files. This facilitates moving purchase, sales, and customer information, as well as supplier and company data, from paper to digital media on a secured server. Sales and purchase bills can be generated as needed.

- To minimize the cost of manpower and paper work this leads to minimum wastage of time in the organisation. The paperwork in the organisation can be reduced due to the computerisation of the various departments
- To do the processing of the application forms speedily and to help in decision making whether to accept the form or not, due to the automation the information regarding the acceptance of the form can be retrieved in the fastest manner as compared to manual system and the information provided the system is accurate and to the point which helps in faster decision making.
- To give management accurate and updated reports about cash collection, products available etc.
- To give the details and solutions for the queries asked by the system timely and to give the precise and confirm information to them.
- To keep the document or records and other valuable information in safety so that unauthorized access of the information can be prevented. Unauthorized people cannot modify the records.

CHAPTER 2
LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 Overview of Visual Basic

VISUAL BASIC (VB) is a powerful application tool. It is a Microsoft windows programming language. VB uses different data types and Shops data differently when compare to C/C ++. VB is a high level language which evolved from the earlier DOS version called BASICS and visual refers to creating a GUI-(graphical user interface) application, it enables the rapid application development (RAD) of GUI applications. It is not only a programming language but also a GUI. It access to the database using data access objects, remote data objects or ActiveX data objects and creation of ActiveX control and objects, scripting languages such as VB script.

VB is an event driven, because end users may click on certain object randomly, so each object has to be programmed independently to response to those actions or events(which means that, object are called automatically when the end user clicks the mouse, moves the objects in the screen etc.).

VB helps in developing complicated application very quickly. It helps in developing complicated applications very quickly. It can develop programs that can be used as front end applications to the database system. It allows the programmer to create good graphical programs with less coding, unlike the other languages.

VB is used to develop applications in a number of different fields like

- Education
- Medicine
- Research
- Science
- Law
- Marketing
- Bank

2.2 Introduction to Visual Basic 6.0

Visual basic evolved from BASIC language which was developed by a professor John keenly and Thomas Kurtz of Dartmouth College in 1960's. Since 1991, 6 versions have been released, with VB 6.0 released in 1998.VB 6.0 provides a vital link to graphical environment and allows you to develop applications based on standard windows features: Dialog boxes, Command buttons, Pull down menus, Scroll Bars, Selection lists etc.

Visual: Refers to the methods used to create the graphical user interface (GUI) by using pre-built objects provided by VB.

Basic: Refer to the basic language used as its basic syntax of statements. However VB 6.0 contains several statements, functions etc. , which are related directly to windows GUI.

Visual Basics comes in three editions:

- **Standard:** It includes some basic controls such as setup wizard, icons and help files. Simple windows application can be created easily using the standard edition.

- **Professional:** It is normally used by computer professionals. It includes all the features of standard edition plus many more such as crystal report, ActiveX and internet controls.

- **Enterprise:** It is the most advanced edition. It includes all the features of professional edition plus features such as remote OLE automation, component manager and removes database access tools. It is used to build distributed application in a team environment.

FEATURES

- ActiveX Data Objects (ADO) and OLE DB replace the Open database connectivity (ODBC) API as the preferred method for accessing shared file and client/server databases.
- A multitude of wizards and other graphical tools aid developers new to Visual Basic.
- Visual Basic is an event driven programming language.
- Visual Basic allows you to adopt more of parallel approach, with independent sections of code for each option that the user may select. This is known as Event driven programming language.
- Data environment designer
- Drag and drop form generation.
- ADO data control (ADODC).
- ADO compliant data bound controls.
- Hierarchical record sets and the flex grid control.
- Data report design.
- Data form wizards.
- Format objects.
- Data repeater control.
- Data source classes and data building.
- Visual data tools.
- SQL editor.
- Component creation.
- Language.
- Packaging and development wizard.
- Data object wizard.

Advantages of Visual Basic

- Visual development of GUI which are simple and easy to learn.
- Programmers need not write code to display the required component.
- The component can be moved, resized, or even deleted if required.
- There is no restriction on the number of controls that can be placed in a form.
- The interface provided by the Visual Programming environment has some inbuilt code.

2.3 Introduction to Oracle 11g

This package is not cheap at all. It is very complex and it can be difficult to administer. But its complexity makes it very robust. There is almost nothing that you cannot do in Oracle. Oracle is specifically designed for multi-user applications. In fact, thousands of users can access the same data without any problems due to Oracle's very good transactional control but applications are not developed nearly as quickly. Oracle can also handle large amounts of data. In the end, oracle is a true DBMS and access is not. This does not mean that access does not have its place. If I'm designing a quick database that only I'll use, I would do it in access over oracle. If I'm designing a robust, scalable application used throughout my enterprise with many users, then oracle is my choice!

Information Management with Oracle Database 11g

Release 2 February 2010

Author: James Steiner

Contributing Authors: Melliyal Annamalai, Mark Drake, Roger Ford, Jean Ihm, Kevin Jernigan, Xavier Lopez, MaryBeth Pierantoni, Jinyu Wang

Oracle Corporation World Headquarters 500 Oracle Parkway Redwood Shores, CA 94065 U.S.A.

The dramatic performance and functional improvements in Oracle Database 11g make the two essential elements for better information management; the ability to manage, secure, query, and administer information with the highest levels of performance, and the ability to derive understanding and knowledge in an open, standard manner from data which had previously been dependent upon proprietary application or device logic. Over a decade of development, research, and close collaboration with customers and application providers have resulted in these unique capabilities found only in Oracle Database 11g.

FEATURES

- Most data management system based on the relational model has a built-in support for query languages like ANSI SQL or QBE (Query by Example). These queries are simple English constructs that allow adodc data manipulation from a table.
- Deferred writing at commits to improve transaction performance.
- Security and control.
- It has a rollback command for recreating the database to its most recent safe point. Grant and revoke limits access to information down to row and column level. Views are valuable features for limiting access to the primary tackles in the database.
- Professional ORACLE starts the DBMS (Database Management System) in the extended memory, so more main memory is available for other applications.
- Relational model of data management is based on set theory. Built-in query language is designed in the RDBMS, so that it can manipulate sets of data (one or more tuples).
- User interface used with relational models is non-procedural because only what needs to be done is specified and not how it has to be done. Using any of the other methods, you have not only to specify what need to be done but how it has to be done as well.

Database Connectivity

Most of the business applications Shop large volumes of data, organized in a format that provides easy access to data. Database Management System (DBMS) provides this mechanism to manipulate data Shopd in Tables, with high-level commands.

By using a database, the programmer can spend more time designing the interface, rather than worrying about how to save, retrieve and sort the actual data. A database basically is a group of related information, organized for easy processing and retrieval.

In visual basic, three data access interfaces are available to you: ActiveX data objects (ADO), Remote Data Objects (RDO), and Data Access Objects (DAO). A data access interface is an object model that represents various facets of accessing data. Using visual basic, you can

programmatically control the connection, statements builders, and returned data for use in any application.

To create a data connection to an Oracle database

1. In the choose Data Source dialog box, select Oracle Database, click OK.
2. If the Add connection dialog box opens, and the data source is not oracle database, click change to open the choose/change data source dialog box.
3. Type the name of the server where the database you want to access is located.
4. Type the username and password used to access the database.
5. Click OK.

DATA CONTROL PROPERTIES

- **ALIGN:** Determine where data control is displayed.
- **CAPTION:** Phrase displayed on the data control.
- **CONNECTIONSTRING:** contains the information used to establish a connection to a database.
- **RECORD SET:** A set of record defined by a data controls connection string and record source properties.
- **RECORD SOURCE:** Determines the table(or virtual table) the data control is attached to.

CHAPTER 3
SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The current system has several shortcomings that are as follows:-

- The existing system was not very effective & was highly time consuming.
- The current system works manually.
- The existing system was somewhat paper-based (paper-work) which takes time?
- The rate at which the work done is very slow due to non-utilization of computerized system.
- It does not provide better data facilities as well as data availability on fingertip.
- Inconsistency was the major problem in the existing system as there is no proper facility was provided to update the data.
- In case user wants to find out details of particular Mobiles whole database records are displayed.
- It is difficult to remember all data
- Consumes large volume of paper work
- Large storage space is required to keep to the files and register in proper coordination's.
- In the present system the report generation becomes very difficult.

3.2 PROPOSED SYSTEM

- This system computerizes all data is storing all data of the Mobiles Sales, Customer Details etc.
- Validation at the time of entering data, so no chance of duplication of data.
- Extensive Validation on input record so that no changeable record are Prone to error.
- Response time for the query should be minimized.
- To keep data base up to date by regular updating.
- Reduce the cost of maintaining system.
- To provide various report facility.
- Cost effective and less manpower required.

ADVANTAGES OF PROPOSED SYSTEM

- The new system would easily overcome most of the short coming of the current system.

- Owner can see the fine report.
- Not much manual work is involved.
- Ensure data accuracy.
- Security of data is done.
- Save a lot of time and effort.
- Optimize processing time.
- User friendly system.

3.3 ANALYSIS

An examination of data and facts to understand cause effect relationships, thus providing basis for problem solving and decision making.

3.3.1 SYSTEM ANALYSIS

System analysis is the evaluation of an activity to identify its desired objectives and determine procedures for efficiently attaining them. Computers the methodical study of the data processing needs of a business or department, together with recommendations for specific hardware and application installation.

3.3.2 FEASIBILITY STUDY

Feasibility study is to check the viability of the project under consideration. Theoretically various types of feasibilities are conducted, but we have conducted three type of feasibilities explained as below. Feasibility study aims to objectively and rationally uncover the strengths and weakness of the existing business or proposed adventure, opportunities and threats as presented by the environment, the resources required to carry through and ultimately the prospects of success.

Feasibility study is most important and it concerns great to the passenger, for which new system is to be developed. It includes the study of success and level of satisfaction that the passenger and the user will get from the system. Feasibility means possibilities (to some extent), i.e. this study explores the possibility of implementation of new system as a substitute to old system removing all shortcomings and including all new requirements.

Feasibility study is a report directed management. It evaluates the impact of the proposed changes in the area(s) in question. The report is a formal document for management, brief enough and sufficiently, non-technical to be understandable, yet detailed enough to provide the basis for system design.

An initial determine in a proposal that whether an alternative system is feasible or not. To determine feasibility of candidate system in all respect I need to consider following feasibility factors:

- ✓ Technical feasibility.
- ✓ Economic feasibility.
- ✓ Operational feasibility.

3.3.2.1 TECHNICAL FEASIBILITY

Technical feasibility centres around the existing system (hardware, application, etc) into what it can sort the proposed addition. A study of resource availability that may affect the ability to achieve an acceptable system. This evaluation determines whether the technology needed for the proposed system is available or not.

- It is determined by the hardware and supporting application.
- Hardware requirement: stand-alone computer with enough main memory space & backup storage more than 2 GB to support RDBMS and huge database.
- Application requirement: ORACLE.
- Can the work for the project be done with current equipment existing technology & available personal?
- Can the system be upgraded if developed?

- If new technology with specifying equipment and application that will successfully satisfy the user requirement front end and back end.

3.3.2.2 OPERATIONAL FEASIBILITY

The proposed system will automated the existing manual system and make it user friendly. With the required training the users will find the system easier to operate. The systems cuts down the time delay of the existing manual system.

3.3.2.3 ECONOMIC FEASIBILITY

This feasibility checks whether the system can be developed with the available funds. The Mobile Shop Management System does not require enormous amount of money to be developed. This can be done economically if planned judicially, so it is economically feasible. The economic feasibility of the system looks upon the financial aspects of the system. It determines whether the project is economically feasible. In other words, It determines whether the investment that goes into the implementation of the project is recoverable. The cost benefit analysis is a commonly used method in evaluating the effectiveness of the system. As the hardware is already available and no investment is to be made in that direction, the only cost involved is that of implementing the system and application.

CHAPTER 4
APPLICATION REQUIREMENTS SPECIFICATIONS

4 APPLICATION REQUIREMENTS

4.1 INTRODUCTION

SRS is a document that completely describes what the proposed application should do without describing how application will do it. The basic goal of the requirement phase is to produce the SRS, which describe the complete behaviour of the proposed application. SRS is also helping the clients to understand their own needs.

Application SRS establishes the basic for agreement the client and supplier on what the application product will do.

1. A SRS provides a reference for validation of the final product.
2. A high-quality SRS is a prerequisite to high-product application.
3. A high-quality SRS reduces the development cost.

4.2 FUNCTIONAL REQUIREMENTS

It defines functions of a system or its components. A function is described as a set of inputs, the behaviour, and the outputs. Behavioural requirements describing all the cases where the system uses the functionality requirements are captured in use cases.

The user shall be able to search either all the initial set of database. The system shall provide appropriate viewers for the user to read documents in document Shop. Every reservation shall be allocated a unique identifier which shall be copy to account permanent Storage area.

Some of the functional requirements of our project are as follows

- ✓ The user shall be able to search either all the initial set of database.
- ✓ The system shall provide appropriate viewers for the user to read documents in document Shop.
- ✓ Every reservation shall be allocated a unique identification.

Some of the other functional requirements are:

- ✓ Administrative functions.
- ✓ Authentication.

- ✓ Authorization levels.
- ✓ Audit tracking.
- ✓ External interfaces.
- ✓ Certification requirements.
- ✓ Reporting requirements.
- ✓ Historical data.
- ✓ Legal or Regulatory requirements.

4.3 NON- FUNCTIONAL REQUIREMENTS

The requirements that specifies criteria that can be used to judge the operation of a system, rather than specific behaviour. This should be contracted with functional requirements that define specific behaviour or functions.

Response time should be less then specified time, input screen be self-explanatory, user friendly, attractive, accurate.

Some non-functional requirements are:

- ✓ Fast (Response time be less than specified time)
- ✓ Accurate (Up three places after decimal)
- ✓ User friendly (Input screen be Self-Explanatory)
- ✓ Attractive (Aesthetically appealing)

Some of the other Non-Functional requirements are:

- ✓ Capacity
- ✓ Availability
- ✓ Reliability
- ✓ Recoverability
- ✓ Maintainability
- ✓ Serviceability
- ✓ Regulatory
- ✓ Manageability

- ✓ Environmental
- ✓ Data Integrity
- ✓ Usability
- ✓ Interoperability

4.4 DOMAIN REQUIREMENTS

Domain requirement is the requirement that comes from the application domain of the system that refers the characteristics of that domain. Therefore, as our system is mobile shop management system, the domain requirement of this system should concern about the requirements that reflect characteristics of is mobile shop management system. It must have basic functions: storing, tracking, updating, and must be able to generate reports.

1. Every operation that occurs in the system must concern of data integrity
2. Every operation that occurs in inventory must be recorded, and the system should generate report from time to time.
3. Regarding security issue, the system must have an authorization module to prevent unauthorized access.
4. Authorized person must be able to access the system 24/7 except the system is under maintenance.
5. Back up unit is required for unexpected system failure event.

4.5 SYSTEM REQUIREMENTS

The most common set of requirement defined by any operating system or application application is the physical computer resources, also known as hardware, a hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating system.

The Hardware and Application Configuration for the proposed system is given below.

Hardware Configuration

For the successful run of the proposed system the required standalone Personal Computer with, minimum hardware required to run the system is as below.

Hardware	Minimum Requirement
Processor	Pentium 2
Hard disk	4 GB
RAM	512 MB

Application Configuration

The minimum Application required by the system is follows.

Front End	Visual basic 6.0
Back End	Oracle 11g
Platform	Windows XP,7 & 10

CHAPTER 5
SYSTEM DESIGN

5 SYSTEM DESIGN

5.1 INTRODUCTION

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. System design could be seen as the application of the systems theory to product development. There are mainly three types of application model:

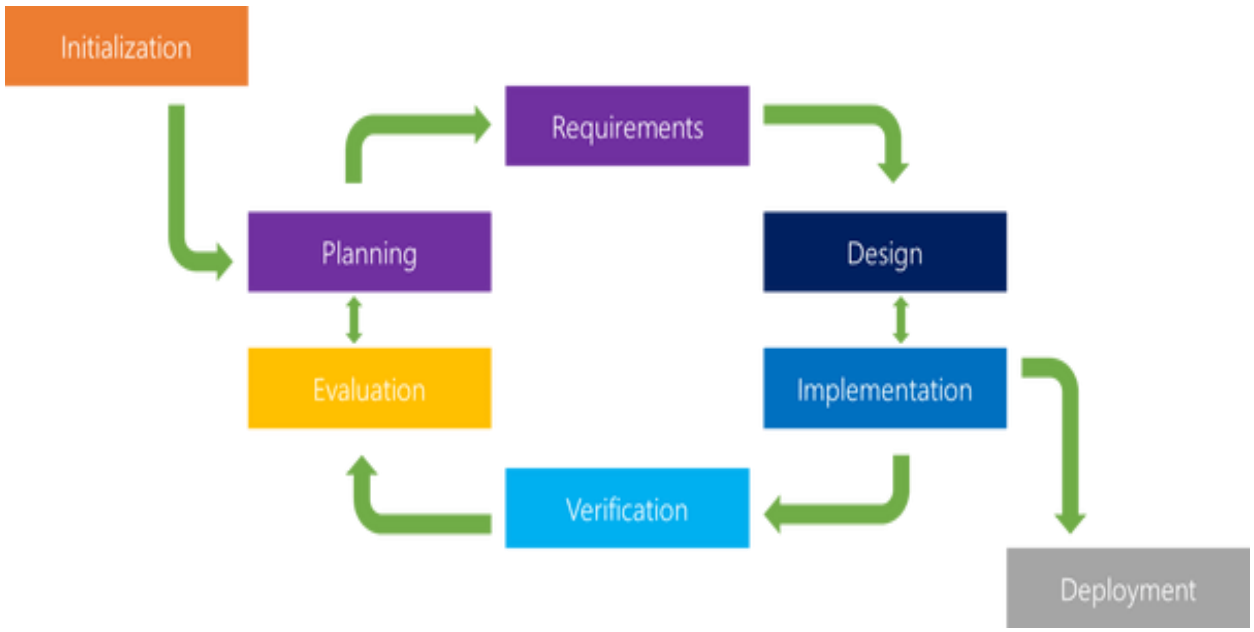
1. Waterfall model.
2. Iterative Enhancement model.
3. Spiral model.

In our project we are implementing Iterative model for developing our project.

Iterative Model:

In the Iterative model, iterative process starts with a simple implementation of a small set of the application requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed.

.



Why iterative model?

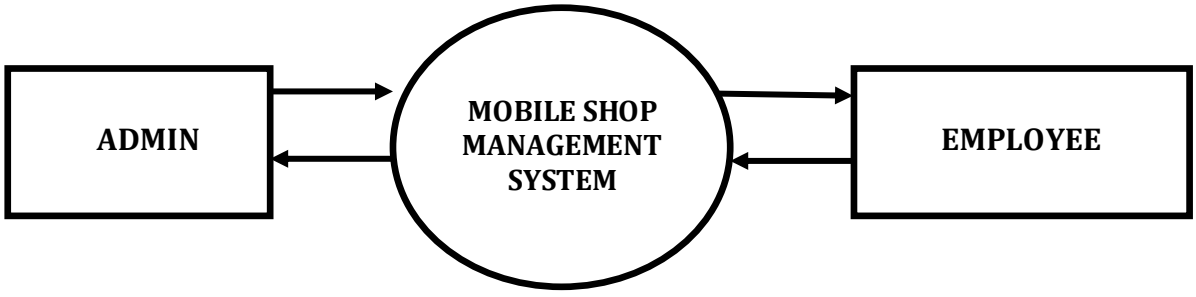
The advantages of the Iterative and Incremental SDLC Model are as follows −

- Some working functionality can be developed quickly and early in the life cycle.
- Results are obtained early and periodically.
- Parallel development can be planned.
- Progress can be measured.
- Less costly to change the scope/requirements.
- Testing and debugging during smaller iteration is easy.
- Risks are identified and resolved during iteration; and each iteration is an easily managed milestone.
- With every increment, operational product is delivered.
- During the life cycle, application is produced early which facilitates customer evaluation and feedback.

5.2 CONTEXT LEVEL DIAGRAM

A Level 0 DFD also called as fundamental system model or a context model represents the entire application element as a single bubble with input and output data indicated by incoming and outgoing arrows respectively. Additional processes and information flow paths are represented as the level 0 is partitioned to reveal details. Each of that proves represents at level 1 is sub function of the overall system depicted in the context model. Each of the process may be refined are layered to depict more detail. Information continuity must be maintained in every layer, that is input and output to each refinement must remain the same.

0 Level Diagram



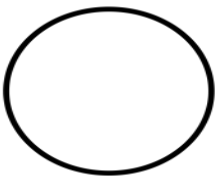
5.3 DATA FLOW DIAGRAM

As information moves through application, it is modified by a series of transformations. A Data Flow Diagram (DFD) is a graphical technique that depicts information flow and the transformations that are applied as data move from input to output. The data flow diagram is known as a data flow graph or a bubble chart.

The Data Flow Diagram may be used to representation a system or application at any level of abstraction. In fact, DFDs may be used partitioned into levels that represent increasing information flow and functional detail. Therefore, the DFD provides a mechanism for functional modelling as information flow modelling.
Data Flow Diagram is a way of expressing system requirements in a graphical form. A DFD also known as bubble chart shows the flow of data through a system and also the movement of data through the different transformations or processes in the system

Data flow diagrams (DFD) are commonly used during problem analysis. DFD's are very useful in understanding a system and can be effectively used for partitioning during analysis.

The basic symbols used to draw a DFD are shown below

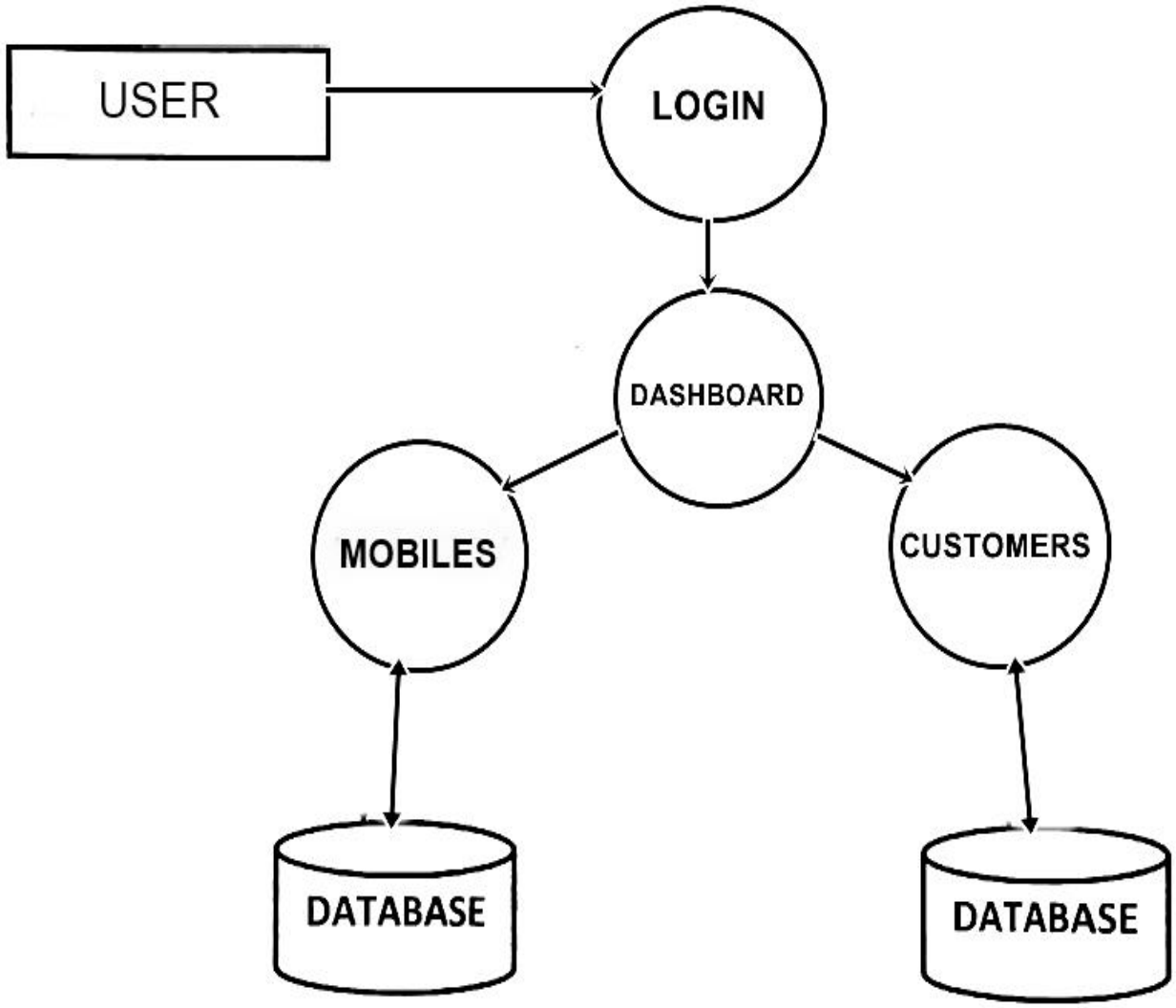


Function

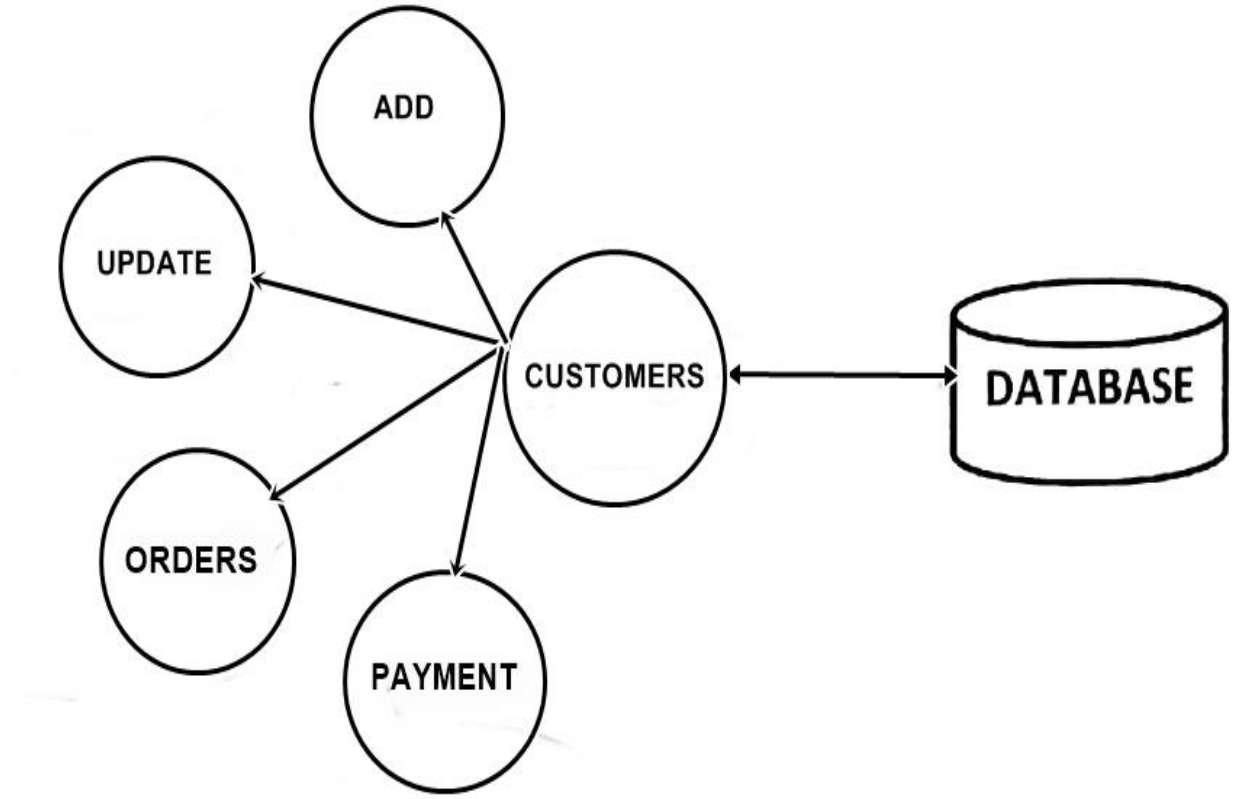
File/Database

Input/Output

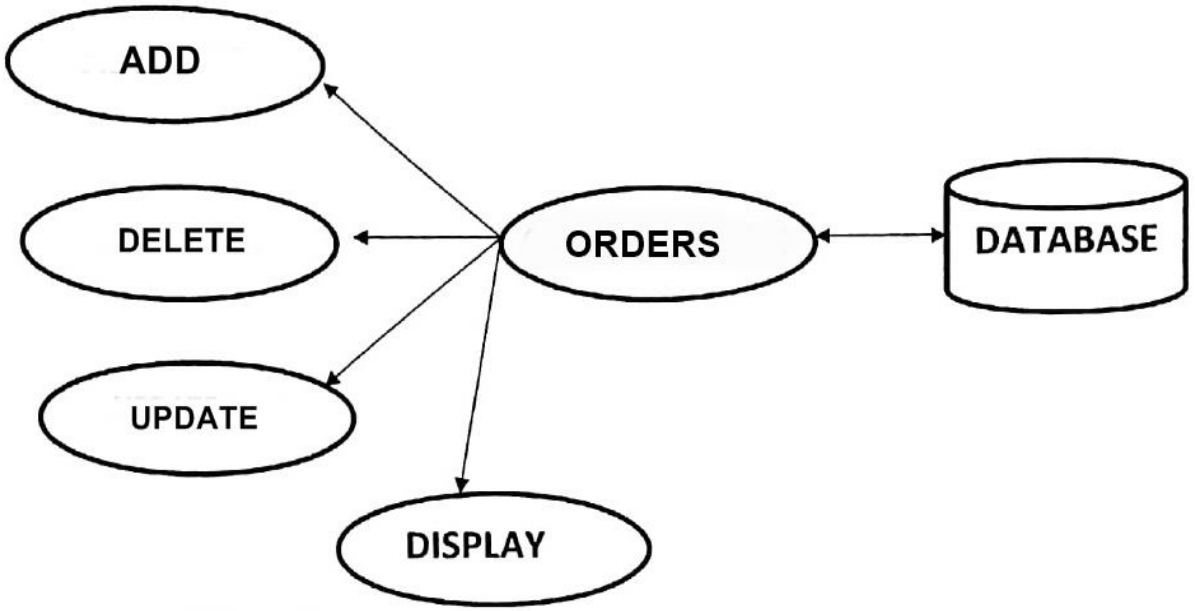
LEVEL 1 DIAGRAM FOR PROCESS

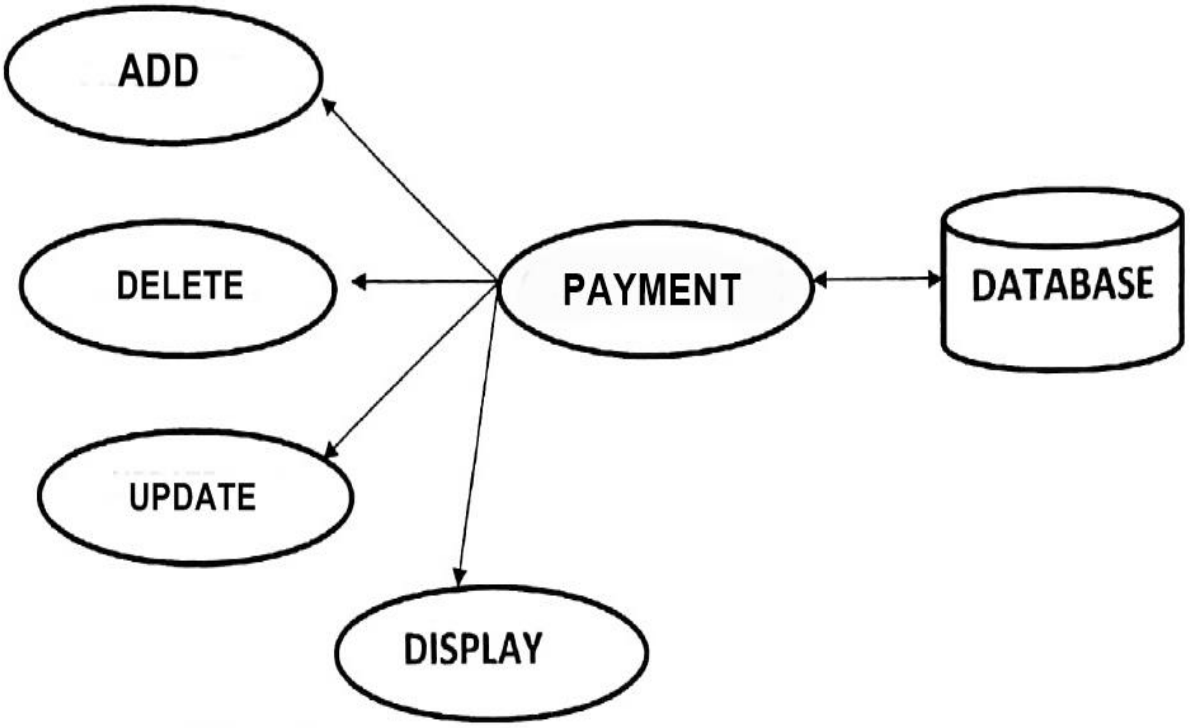


LEVEL 2 DIAGRAM FOR PROCESS



LEVEL 3 DIAGRAM FOR PROCESS





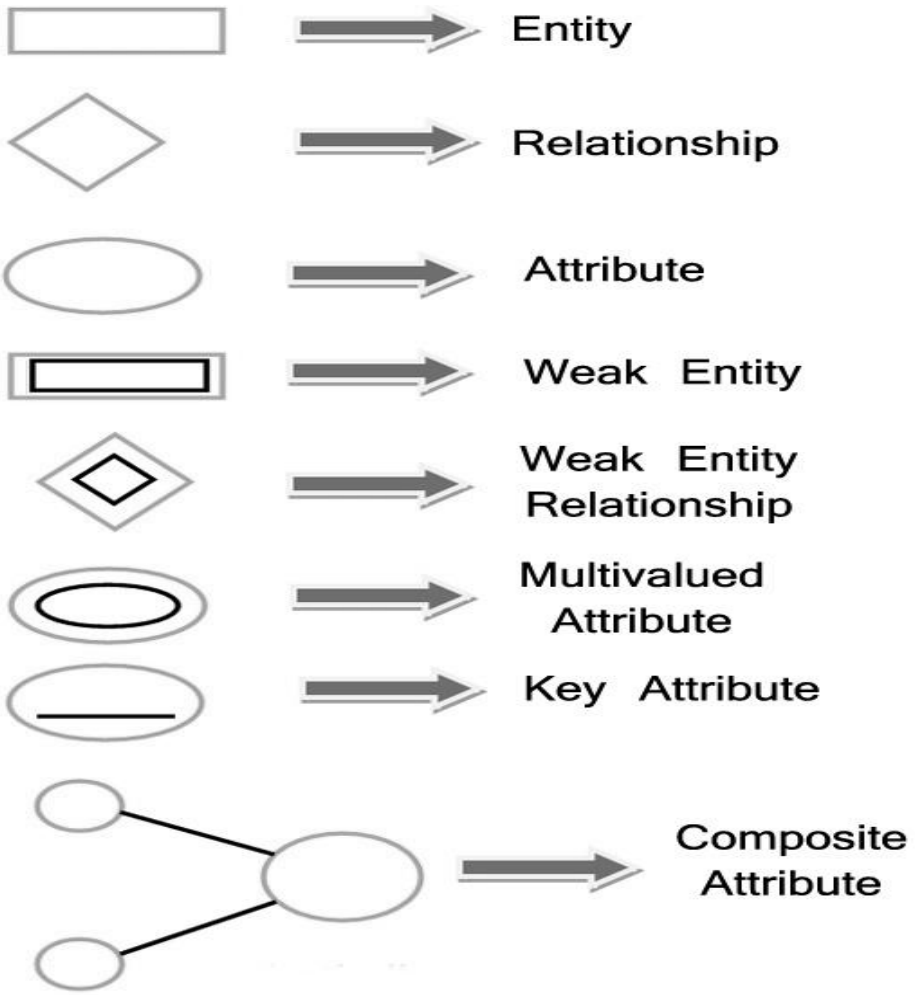
INTRODUCTION TO E-R MODEL:

An ER model is an abstract way to describe a database. Describing a database usually starts with a relational database, which Shops data in tables. Some of the data in these tables point to data in other tables - for instance, your entry in the database could point to several entries for each of the phone numbers that are yours. The ER model would say that you are an entity, and each phone number is an entity, and the relationship between you and the phone numbers is 'has a phone number'. Diagrams created to design these entities and relationships are called entity– relationship diagrams or ER diagrams.

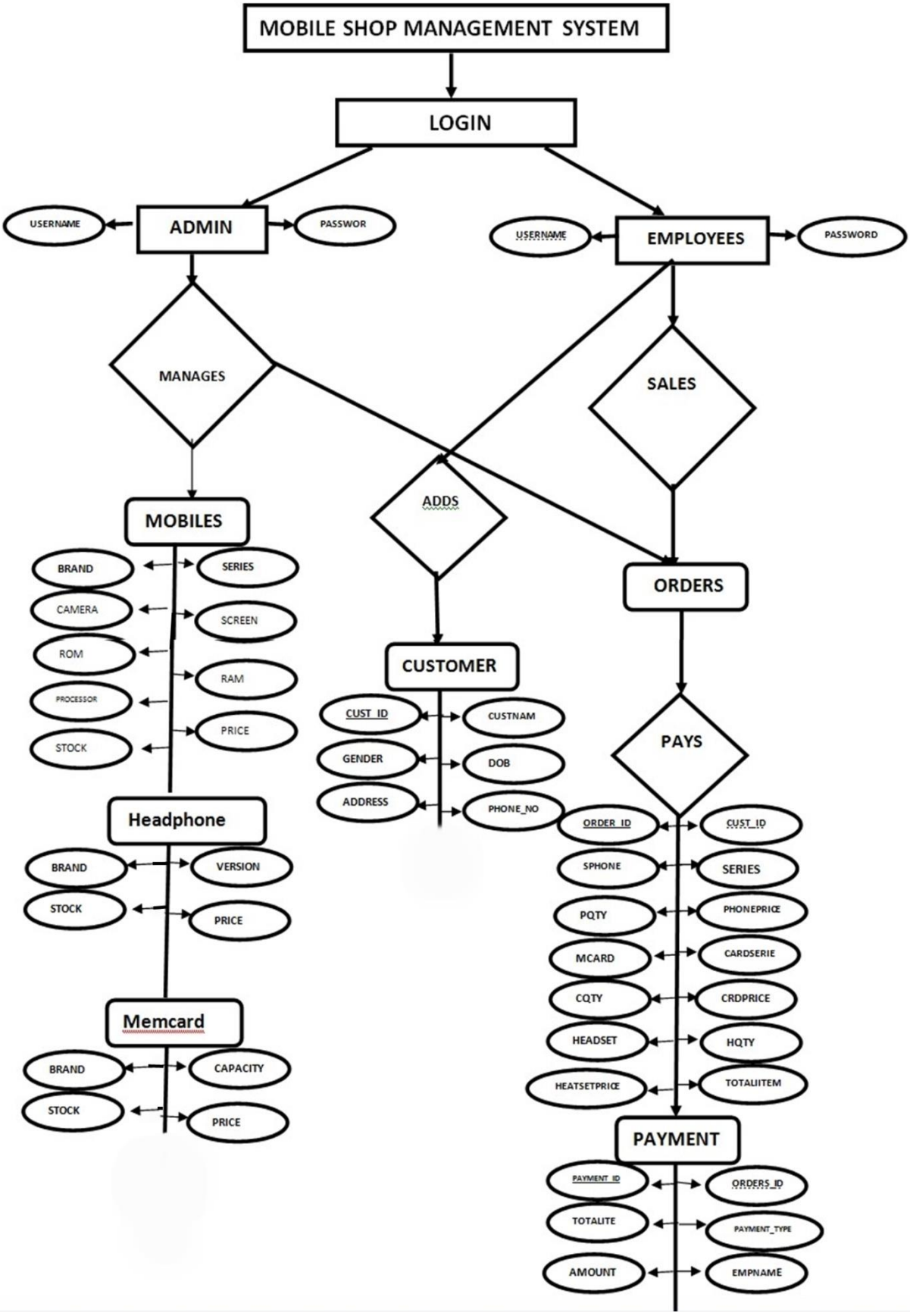
INTRODUCTION TO E-R DIAGRAM:

Entity relationship diagram is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data-an object or concept. A relationship is how the data is shared between entities.

ER DIAGRAM SYMBOLS



ER DIAGRAM:


MOBILE SHOP MANAGEMENT SYSTEM
LOGIN
ADMIN
USERNAME
PASSWOR
EMPLOYEES
USERNAME
PASSWORD
MANAGES
SALES
ADDS
MOBILES
BRAND
SERIES
CAMERA
SCREEN
ROM
RAM
PROCESSOR
PRICE
STOCK
ORDERS
CUSTOMER
CUST_ID
CUSTNAM
GENDER
DOB
ADDRESS
PHONE_NO
PAYS
Headphone
BRAND
VERSION
STOCK
PRICE
ORDER_ID
CUST_ID
SPHONE
SERIES
PQTY
PHONEPRICE
MCARD
CARDSERIE
CQTY
CRDPRICE
HEADSET
HQTY
HEATSETPRICE
TOTALIITEM
Memcard
BRAND
CAPACITY
STOCK
PRICE
PAYMENT
PAYMENT_ID
ORDERS_ID
TOTALITE
PAYMENT_TYPE
AMOUNT
EMPNAME

5.5 DATABASE DESIGN

NORMALIZATION

Database normalization is the process of removing redundant data from the tables to improve storage efficiency, data integrity and scalability. The classification is called normal forms (or NF's) and they show how algorithms are converting a given database between them. Normalization generally involves splitting existing tables into multiple ones, which must be re-joined or linked each time a query is issued.

1 NF (FIRST NORMAL FORM)

A table is in 1 NF if**:**

- There are no duplicate rows in a relation**.**
- Each data value stored is single valued
- Entities in the column are of same type

Consider the following example**:**

ORDER ID	PRODUCT NAME
OR-001	APPLE , SAMSUNG
OR-002	SAMSUNG , NOKIA
OR-003	NOKIA , SAMSUNG
OR-004	ONE PLUS , APPLE

Rule 1**:**
A column with atomic data cannot have several values of same type in it**.** In the above table there are multiple values for item name value, this can be overcome by creating multiple columns for multiple values**.**

ORDER ID	PRODUCT NAME 1	PRODUCT NAME 2
OR-001	APPLE	SAMSUNG
OR-002	SAMSUNG	NOKIA
OR-003	NOKIA	SAMSUNG
OR-004	ONE PLUS	APPLE

It satisfies rule 1 but the table is not in 1 NF
Rule 2**:**
A table with atomic data cannot have multiple columns with same type of data**.**

ORDER ID	PRODUCT NAME
OR-001	APPLE
OR-002	SAMSUNG
OR-002	SAMSUNG
OR-003	NOKIA

Second Normal Form (2NF):

A table is in 2NF if & only if it is in 1NF and every non-key attribute is fully functionally dependent on the whole of the primary key.

Tables are said to be in 2NF when

- The tables meet the criteria for 1NF.
- If the primary key is a composite of attributes (contains multiple columns), the non key attributes (columns) must depend on the whole key.

Any table with a primary key that is composed of a single attribute (column) is automatically in 2NF.

- Our project supports both 1NF and 2NF.
- The primary key in our project is " CUSTOMER ID "

A table is in 2 NF if it is in 1 NF and every non-key attribute is fully functionally dependent on each candidate key of the relation i**.**e**.**, every non-key column depends on the key but not on the subset of key**.**

Consider the following example**:**

ORDERS DETAILS

ORDER ID	CUST_ID	SPHONE	SERIES	PQTY	PHONEPRICE	MCARD	CARDSERIES
0R-001	0000	APPLE	iPhone X	1	95000	SONY	16GB
0R-002	0002	Samsung	Galaxy S9	1	59999	SanDisk	128GB
0R-003	0003	Nokia	6.1	1	15000	HP	8GB

CQTY	CARDPRICE	HEADSET	HEADSETVER	HQTY	HEADSETPRICE	TOTALITEMS
1	650	JBL	C300	1	800	3
1	1300	Sony	MDXR	1	2300	3
1	500	Sennheiser	CX90	1	2349	3

CUSTOMER DETAILS

CUST_ID	CUSTNAME	GENDER	DOP	ADDRESS	PHONE_NO
0001	SHAIK	MALE	17-10-2018	BTM	9022457784
0002	MURAD	MALE	24-10-2018	Indiranagar	8446322104
0003	AMRIT	MALE	25-10-2018	HAL	7004558131

PAYMENT DETAILS

PAYMENT_ID	ORDER_ID	TOTALITEMS	PAYMENT_TYPE	AMOUNT	EMPNAME
PY-000	SHAIK	3	Cash	96450	Shaik
PY-001	MURAD	3	Paytm	635999	Murad
PY-002	AMRIT	3	Paytm	17849	Shaik

5.4 Table St ructure:

- **Admin Details:**

FIELD NAME	DATATYPE	SIZE
USERNAME	Varchar2	10
PASSWORD	Varchar2	30

- **Customer Details:**

Field name	Datatype	Size
Cust_id	Varchar2 primary key	10
custname	Varchar2	30
Gender	Varchar2	10
Dop	Varchar2	20
address	Varchar2	100
Series	Varchar2	20
Phone_no	Varchar2	15

- **Orders details:**

Field name	Datatype	Size
Order_id	Varchar2 primary key	10
Cust_id	Varchar2 foreign key	10
Sphone	Varchar2	20
Series	Varchar2	20
Pqty	Varchar2	20
phoneprice	Varchar2	20
Mcard	Varchar2	20
cardseries	Varchar2	20
Cqty	Varchar2	20
cardprice	Varchar2	20
headset	Varchar2	20
headsetver	Varchar2	20

hqty	Varchar2	20
headsetprice	Varchar2	20

- **Payment Details:**

Field name	Datatype	Size
Payment_id	Varchar2 primary key	10
Order_id	Varchar2	10
totalitems	int foreign key	
Payment_type	Varchar2	40
amount	number	10
empname	Varchar2	20

- **Mobile Details:**

Field name	Datatype	Size
Brand	Varchar2	29
series	Varchar2	29
price	number	22
stock	number	22
processor	Varchar2	38
RAM	Varchar2	32
ROM	Varchar2	33
camera	Varchar2	32
screen	Varchar2	30

- **Emp_detail :**

Field name	Datatype	Size
Emp_id	number	22
username	Varchar2	20
password	Varchar2	22
phone	number	22
salary	Varchar2	38
adhaar	Varchar2	32
Join_date	Varchar2	33

- **Headphone details:**

Field name	Datatype	Size
Brand	Varchar2	10
Version	Varchar2	10
Stock	Varchar2	20
Price	Varchar2	20

- **Memcard details:**

Field name	Datatype	Size
Brand	Varchar2	10
capacity	Varchar2	10
Stock	Varchar2	20
Price	Varchar2	20

CHAPTER 6
IMPLEMENTATION

1. Implementation

6.1 Introduction :

Implementation is the carrying out, execution, or practice of a plan, a method, or any design for doing something. As such, implementation is the action that must follow any context, implementation encompasses all the processes involved in getting new application or hardware operating properly in its environment, including installation, configuration, and running, testing, & making necessary changes. The word deployment used to mean the same thing.

Implementation Procedure

We have used Oracle 11G as back end & visual basic 6.0 as front end. We have used ADODB for back end connections in our project & we have used Microsoft tabbed dialog control 6.0 for handling admin login pages.

We have also used combo boxes & many other components which make our project a user friendly project.

6.2 Source Code:

MODULE CONNECTION

```
Public con As ADODB.Connection
Public rs As ADODB.Recordset
Public com As ADODB.Command

Public Sub connect()
Set con = New ADODB.Connection
Set rs = New ADODB.Recordset
Set com = New ADODB.Command
With con
.ConnectionString="Provider=MSDAORA.1;Password=password;User ID=Mudassir;Persist
Security Info=True"
```

```
.CursorLocation = adUseClient
.Open
com.ActiveConnection = con
End With
End Sub
```

Register Code:

```
Private Sub cmdregister_Click()
If txtuser.Text = "" Then
MsgBox "enter username", vbCritical
ElseIf txtpass.Text = "" Then
MsgBox "enter password", vbCritical
ElseIf txtphone.Text = "" Then
MsgBox "enter phone number", vbCritical
ElseIf txtadhaar.Text = "" Then
MsgBox "enter adhaar number", vbCritical
Else
rs.AddNew
rs.Fields("empid").Value = txtid.Text
rs.Fields("username").Value = txtuser.Text
rs.Fields("password").Value = txtpass.Text
rs.Fields("phone").Value = txtphone.Text
rs.Fields("adhaar").Value = txtadhaar.Text
rs.Fields("join_date").Value = DTPicker1.Value
rs.Fields("salary").Value = txtsalary.Text
MsgBox "REGISTRATION SUCESSFULLY", vbInformation
rs.Update
End If
End Sub
```

Login Code:

```
Private Sub cmdemplog_Click()
If txtuser.Text = "" Then
MsgBox "enter the username", vbInformation
ElseIf txtpass.Text = "" Then
MsgBox "enter the password", vbInformation
Else
sql = "select * from emp_detail where username='" + txtuser.Text + "' and password='" + txtpass.Text + "'"
rs.Open sql, con, adOpenKeyset, adLockPessimistic, adCmdText
If rs.EOF = False Then
sales.txtempid.Text = txtuser.Text
dash.lblname = txtuser.Text
Unload Me
Mobiles.cmdadd.Enabled = False
Mobiles.cmdupdate.Enabled = False
dash.Label6.Enabled = False
success.Show
Else
MsgBox "Invalid Credentials", vbCritical
End If
rs.Close
End If
End Sub

Private Sub cmdadminlog_Click()
If txtuser.Text = "" Then
MsgBox "enter the username", vbInformation
ElseIf txtpass.Text = "" Then
MsgBox "enter the password", vbInformation
Else
sql = "select * from admin where username='" + txtuser.Text + "' and password='" + txtpass.Text + "'"
```

```
rs.Open sql, con, adOpenKeyset, adLockPessimistic, adCmdText
If rs.EOF = False Then
dash.lblname.Visible = False
dash.lblee = "Administrator"
Unload Me
success.Show
Else
MsgBox "Invalid Credentials", vbCritical
End If
rs.Close
End If
End Sub

Private Sub Command2_Click()
End
End Sub

Private Sub Command3_Click()
Unload Me
register.Show
End Sub

Private Sub Form_Load()
Module1.connect
Me.Left = (Screen.Width - Me.Width) / 2
Me.Top = (Screen.Height - Me.Height) / 2
End Sub
```

Password Show/Hide Code:

```
Private Sub Check1_Click()
If Check1.Value = 1 Then
txtpass.PasswordChar = ""
Else
txtpass.PasswordChar = "*"
```

```
End If
End Sub

Private Sub txtpass_KeyPress(KeyAscii As Integer)
txtpass.PasswordChar = "*"
End Sub

Private Sub txtuser_GotFocus()
txtuser.Text = ""
End Sub
```

Forgot Password Form:

```
Private Sub checkbtn_Click()
sql = "select * from emp_detail where USERNAME= '" + txtuserid + "' "
rs.Open sql, con
If rs.EOF Then
lblmsg1.Caption = "Username does'nt exist."
lblmsg1.ForeColor = &HFF&
Else
lblmsg1.Caption = "Username Found."
lblmsg1.ForeColor = &H8000&
Label2.Visible = True
txtnumber.Visible = True
cmdverify.Visible = True
checkbtn.Enabled = False
txtuserid.Enabled = False
End If
rs.Close
End Sub

Private Sub cmdchange_Click()
If txtnewpass.Text = txtconfirm.Text Then
sql = "select * from emp_detail where username='" + txtuserid + "'"
```

```
rs.Open sql, con, adOpenKeyset, adLockPessimistic, adcmdtxt
rs!Password = txtconfirm.Text
rs.Update
MsgBox "Password changed sucessfully", vbInformation, "password change success"
Unload Me
login.Show
Else
MsgBox "Password does not match,please enter correct details", vbCritical, "change password failed"
txtnewpass.Text = ""
txtconfirm.Text = ""
End If
End Sub

Private Sub cmdchangepass_Click()
If txtnewpass.Text = txtconfirm.Text Then
sql = "select * from emp_detail where username='" + txtuserid + "'"
rs.Open sql, con, adOpenKeyset, adLockPessimistic, adcmdtxt
rs!Password = txtconfirm.Text
rs.Update
MsgBox "Password Changed Sucessfully", vbInformation, "Password Change Sucess"
Unload Me
login.Show
Else
MsgBox "Passwod does not match, Please enter correct details", vbCritical, "change password failed"
txtnewpass.Text = ""
txtconfirm.Text = ""
End If
End Sub

Private Sub cmdverify_Click()
Dim str As String
sql = "select * from emp_detail where phone= '" + txtnumber + "' "
```

```
rs.Open sql, con
If rs.EOF Then
lblmsg2.Caption = "Contact not found.....sorry can't reset the password!!!"
lblmsg2.ForeColor = &HFF&
Else
lblmsg2.Caption = "Contact found."
lblmsg2.ForeColor = &H8000&
Label3.Visible = True
Label4.Visible = True
txtnewpass.Visible = True
txtconfirm.Visible = True
cmdverify.Visible = True
cmdchangepass.Visible = True
cmdverify.Enabled = False
txtnumber.Enabled = False
End If
rs.Close
End Sub

Private Sub Command1_Click()
Unload Me
End Sub

Private Sub Form_Load()
Label2.Visible = False
Label3.Visible = False
Label4.Visible = False
txtnumber.Visible = False
txtnewpass.Visible = False
cmdverify.Visible = False
cmdchangepass.Visible = False
txtconfirm.Visible = False
Module1.connect
Me.Left = (Screen.Width - Me.Width) / 2
```

```
Me.Top = (Screen.Height - Me.Height) / 2
End Sub

Private Sub Form_Load()
Me.Left = (Screen.Width - Me.Width) / 2
Me.Top = (Screen.Height - Me.Height) / 2
End Sub

Private Sub Label1_Click()
headphone.Show
End Sub

Private Sub Label3_Click()
memorycard.Show
End Sub

Private Sub dash_Click()
Unload Me
dashboard.Show
End Sub

Private Sub Form_Load()
Me.Left = (Screen.Width - Me.Width) / 2
Me.Top = (Screen.Height - Me.Height) / 2
End Sub

Private Sub Label1_Click()
Nokia.Show
Unload Me
End Sub

Private Sub Label2_Click()
Unload Me
samsung.Show
```

```
End Sub

Private Sub Label3_Click()
Unload Me
google.Show
End Sub

Private Sub Label6_Click()
Unload Me
out.Show
End Sub

Private Sub Label4_Click()
Unload Me
oneplus.Show
End Sub

Private Sub Label5_Click()
MsgBox "Coming Soon", vbInformation

End Sub

Private Sub Label7_Click()
MsgBox "Coming Soon", vbInformation
End Sub
```

Cart Form:

```
Private Sub cart_Click()
sales.txtb1.Text = cart.txtb1.Text
sales.txts1.Text = cart.txts1.Text
sales.txtp1.Text = cart.txtp1.Text
sales.txtb2.Text = cart.txtb2.Text
sales.txts2.Text = cart.txts2.Text
```

```
sales.txtp2.Text = cart.txtp2.Text
sales.txtb3.Text = cart.txtb3.Text
sales.txts3.Text = cart.txts3.Text
sales.txtp3.Text = cart.txtp3.Text
dash.lblstatus.Visible = False
sales.txtcname.Text = ""
sales.txtaddress.Text = ""
sales.txtpayid.Text = ""
sales.txtcid.Text = ""
sales.txtnumber.Text = ""
sales.txtitems.Text = ""
sales.txtdate.Text = ""
sales.cmbgender.Text = ""
sales.txtq1.Text = ""
sales.txtq2.Text = ""
sales.txtq3.Text = ""
sales.txtoid.Text = ""
sales.cmbapay.Text = ""
sales.txttotal.Text = ""
Unload Me
sales.ShowEnd Sub

Private Sub Form_Load()
Me.Left = (Screen.Width - Me.Width) / 2
Me.Top = (Screen.Height - Me.Height) / 2
End Sub
```

Dashboard Form Code:

```
Private Sub Form_Load()
WebBrowser1.Visible = False
connect
If rs.State = 1 Then rs.Close
rs.Open "select count(*),sum(amount),sum(totalitems) from payment", con
```

```
GoTo last
last:
lblcus = rs(0)
lblprice = rs(1)
lblqty = rs(2)
End Sub

Private Sub Label1_Click()
newo.Show
End Sub

Private Sub Label10_Click()
accessories.Show
End Sub

Private Sub lblsale_Click()
sales.Show
End Sub

Private Sub Label12_Click()
WebBrowser1.GoBack
End Sub

Private Sub Label13_Click()
WebBrowser1.GoForward
End Sub

Private Sub Label14_Click()
WebBrowser1.Visible = False
End Sub


Private Sub lbldash_Click()
dash.Show
```

```
End Sub

Private Sub Label3_Click()
Mobiles.Show
End Sub

Private Sub Label4_Click()
cart.Show
End Sub

Private Sub Label6_Click()
admincontrol.Show
End Sub

Private Sub Label7_Click()
End
End Sub

Private Sub Label8_Click()
WebBrowser1.Visible = True
WebBrowser1.Navigate "google.com"
End Sub

Private Sub Label9_Click()
WebBrowser1.Navigate "google.com"
End Sub
```

Splash:

```
Private Sub time1_Timer()
If time1.Interval = 5000 Then
login.Show
Unload Me
time1.Enabled = False
```

```
End If
End Sub

Private Sub Timer1_Timer()
If Timer1.Interval = 3500 Then
Unload Me
dash.Show
Timer1.Enabled = False
End If
End Sub
```

Sales Code:

```
Private Sub cmdpurchase_Click()
If txtcid.Text = "" Then
MsgBox "enter Customer ID", vbCritical
ElseIf txtdate.Text = "" Then
MsgBox "mention the date", vbCritical
ElseIf txtcname.Text = "" Then
MsgBox "enter customer name", vbCritical
ElseIf txtnumber.Text = "" Then
MsgBox "enter the number", vbCritical
ElseIf cmbgender.Text = "" Then
MsgBox "mention gender", vbCritical
ElseIf txttotal.Text = "" Then
MsgBox "enter the total amount", vbCritical
ElseIf txtoid.Text = "" Then
MsgBox "enter order ID", vbCritical
ElseIf txtpayid.Text = "" Then
MsgBox "enter payment ID", vbCritical
Else
Adodc1.Recordset.Update
customers
orders
payment
```

```
com.CommandText = "update mobiles set stock=stock - '" + txtq1.Text + "' where series ='" + txts1.Text + "'"
com.ActiveConnection = con
com.Execute
com.CommandText = "update memcard set stock=stock - '" + txtq2.Text + "' where brand='" + txtb2.Text + "' AND capacity = '" + txts2 + "'"
com.ActiveConnection = con
com.Execute
com.CommandText = "update headphone set stock=stock - '" + txtq3.Text + "' where version='" + txts3.Text + "'"
com.ActiveConnection = con
com.Execute
End If
End Sub


Sub orders()
com.CommandText = "insert into orders values('" & txtoid.Text & "','" & txtcid.Text & "','" & txtb1.Text & "','" & txts1.Text & "','" & txtq1.Text & "','" & txtp1.Text & "','" & txtb2.Text & "','" & txts2.Text & "','" & txtq2.Text & "','" & txtp2.Text & "','" & txtb3.Text & "','" & txts3.Text & "','" & txtq3.Text & "','" & txtp3.Text & "',' " & txtitems.Text & " ')"
com.ActiveConnection = con
com.Execute
MsgBox "Order Details added "
End Sub

Sub payment()
com.CommandText = "insert into payment values('" & txtpayid.Text & "','" & txtoid.Text & "','" & txtitems.Text & "','" & cmbapay.Text & "','" & txttotal.Text & "','" & txtempid.Text & "')"
com.ActiveConnection = con
com.Execute
MsgBox "Payment added successfully"
End Sub
```

```
Sub customers()
com.CommandText = "insert into customers values('" & txtcid.Text & "','" & txtcname.Text & "','" & cmbgender.Text & "','" & DTPicker1.Value & "','" & txtaddress.Text & "','" & txtnumber.Text & "')"
com.ActiveConnection = con
com.Execute
MsgBox "Customer Details Added Successfully..!"
End Sub
```

Validations

```
Private Sub txtnumber_KeyPress(KeyAscii As Integer)
If (KeyAscii >= 48 And KeyAscii <= 57) Or KeyAscii = 8 Or KeyAscii = 32 Then
txtnumber.Locked = False
Else
txtnumber.Locked = True
MsgBox "enter only numbers", vbCritical
End If
End Sub

Private Sub txtcid_KeyPress(KeyAscii As Integer)
If (KeyAscii >= 48 And KeyAscii <= 57) Or KeyAscii = 8 Or KeyAscii = 32 Then
txtadhaar.Locked = False
Else
txtadhaar.Locked = True
MsgBox "enter only numbers", vbCritical
End If
End Sub

Private Sub txtitems_KeyPress(KeyAscii As Integer)
If (KeyAscii >= 48 And KeyAscii <= 57) Or KeyAscii = 8 Or KeyAscii = 32 Then
txtadhaar.Locked = False
Else
txtadhaar.Locked = True
MsgBox "enter only numbers", vbCritical
```

```
End If
End Sub

Private Sub txtcname_KeyPress(KeyAscii As Integer)
If (KeyAscii >= 65 And KeyAscii <= 90) Or (KeyAscii >= 97 And KeyAscii <= 122) Or KeyAscii = 8 Or KeyAscii = 32 Then
txtcname.Locked = False
Else
txtcname.Locked = True
MsgBox "enter only alphabets", vbCritical
End If
End Sub
```

Auto Increment Code:

```
Private Sub txtcid_GotFocus()
Set rs = New ADODB.Recordset
rs.Open "select * from customers order by cust_id desc", con
Dim num As String * 6
Dim nums As Long
With rs
If rs.EOF Then
nums = "1" + "111"
txtcid.Text = num
Else
nums = Right(!cust_id, 3) + 1
num = "1" + Right("111" & nums, 3)
End If
txtcid.Text = num
End With
rs.Close
End Sub
```

```
Private Sub txtoid_GotFocus()
Set rs = New ADODB.Recordset
rs.Open "select * from orders order by order_id desc", con
Dim num As String * 6
Dim nums As Long
With rs
If rs.EOF Then
nums = "OR-" + "000"
txtoid.Text = num
Else
nums = Right(!order_id, 3) + 1
num = "OR-" + Right("000" & nums, 3)
End If
txtoid.Text = num
End With
rs.Close
End Sub

Private Sub txtpayid_GotFocus()
Set rs = New ADODB.Recordset
rs.Open "select * from payment order by payment_id desc", con
Dim num As String * 6
Dim nums As Long
With rs
If rs.EOF Then
nums = "PY-" + "000"
txtpayid.Text = num
Else
nums = Right(!payment_id, 3) + 1
num = "PY-" + Right("000" & nums, 3)
End If
txtpayid.Text = num
End With
rs.Close
```

End Sub

Clear Code:

```
Private Sub cmdclear_Click()
txtcname.Text = ""
txtaddress.Text = ""
txtb1.Text = ""
txtb2.Text = ""
txtb3.Text = ""
txtpayid.Text = ""
txtcid.Text = ""
txtnumber.Text = ""
txtitems.Text = ""
txtp1.Text = ""
txtp2.Text = ""
txtp3.Text = ""
txtp1.Text = ""
txtp2.Text = ""
txtdate.Text = ""
txtp3.Text = ""
cmbgender.Text = ""
txtq1.Text = ""
txtq2.Text = ""
txtq3.Text = ""
txtoid.Text = ""
cmbapay.Text = ""
txttotal.Text = ""
txts3.Text = ""
txts2.Text = ""
txts1.Text = ""
End Sub
```

Print Code:

```
Private Sub cmdprint_Click()
DataEnvironment1.Command1 txtcid
DataReport1.Show
DataReport1.Refresh
DataEnvironment1.rsCommand1.Close
End Sub
```

Admin control code:

```
Private Sub Form_Load()
Module1.connect
paygrid.Visible = False
empgrid.Visible = False
custgrid.Visible = False
ordergrid.Visible = False
mobgrid.Visible = False
DTPicker1.Value = Date
DTPicker2.Value = Date
Me.Left = (Screen.Width - Me.Width) / 2
Me.Top = (Screen.Height - Me.Height) / 2
End Sub
```

Between Dates

```
Private Sub searchdatebtn_Click()
If DTPicker2.Value < DTPicker1.Value Then
MsgBox "select correct date", vbCritical, "warning"
Else
Adodc1.RecordSource = "select * from sold where dop between '" & Text1.Text & "'  and '" &
Text2.Text & "'"
Adodc1.Refresh
If Adodc1.Recordset.EOF Then
MsgBox "record not found", vbCritical
```

```
End If
End If
End Sub
```

Admin control buttons code:

```
Private Sub cmdfind_Click()
If custgrid.Visible = True Then
Adodc3.RecordSource = "select * from customers where cust_id like '" & txtid.Text & "'"
Adodc3.Refresh
If Adodc3.Recordset.EOF Then MsgBox "Customer ID not found", vbCritical
ElseIf ordergrid.Visible = True Then
Adodc1.RecordSource = "select * from sold where cust_id like '" & txtid.Text & "'"
Adodc1.Refresh
If Adodc1.Recordset.EOF Then MsgBox "Customer ID not found", vbCritical
ElseIf paygrid.Visible = True Then
Adodc2.RecordSource = "select * from payment where payment_id like '" & txtid.Text & "'"
Adodc2.Refresh
If Adodc2.Recordset.EOF Then MsgBox "Payment ID not found", vbCritical
End If
End Sub

Private Sub cmdgo_Click()
If mobgrid.Visible = True Then
Adodc4.RecordSource = "select * from mobiles where brand like '" & Text4.Text & "'"
Adodc4.Refresh
If Adodc4.Recordset.EOF Then MsgBox "Brand not found", vbCritical
ElseIf ordergrid.Visible = True Then
Adodc1.RecordSource = "select * from sold where sphone like '" & Text4.Text & "'"
Adodc1.Refresh
If Adodc1.Recordset.EOF Then MsgBox "Brand not found", vbCritical
End If
End Sub

Private Sub cmdlook_Click()
```

```
Adodc1.RecordSource = "select * from sold where empname like '" & Text3.Text & "'"
Adodc1.Refresh
If Adodc1.Recordset.EOF Then MsgBox "Employee name not found", vbCritical
End Sub

Private Sub cmdmobile_Click()
empgrid.Visible = False
custgrid.Visible = False
ordergrid.Visible = False
paygrid.Visible = False
mobgrid.Visible = True
End Sub

Private Sub cmdorder_Click()
empgrid.Visible = False
paygrid.Visible = False
custgrid.Visible = False
ordergrid.Visible = True
mobgrid.Visible = False
End Sub

Private Sub cmdemp_Click()
empgrid.Visible = True
paygrid.Visible = False
custgrid.Visible = False
ordergrid.Visible = False
mobgrid.Visible = False
End Sub

Private Sub Command1_Click()
Adodc3.RecordSource = "select * from customers where cust_id like '" & txtid.Text & "'"
Adodc3.Refresh
If Adodc3.Recordset.EOF Then MsgBox "Customer ID not found", vbCritical
End Sub
```

```
Private Sub cmdrefresh_Click()
Adodc1.Refresh
End Sub

Private Sub cmdpay_Click()
empgrid.Visible = False
custgrid.Visible = False
ordergrid.Visible = False
mobgrid.Visible = False
paygrid.Visible = True
End Sub

Private Sub cmdreg_Click()
register.Show
End Sub

Private Sub Command11_Click()
ordergrid.Refresh
End Sub

Private Sub dashboard_Click()
dash.Show
End Sub

Private Sub DTPicker1_Change()
Text1 = DTPicker1.Value
End Sub

Private Sub DTPicker2_Change()
Text2 = DTPicker2.Value
End Sub
```

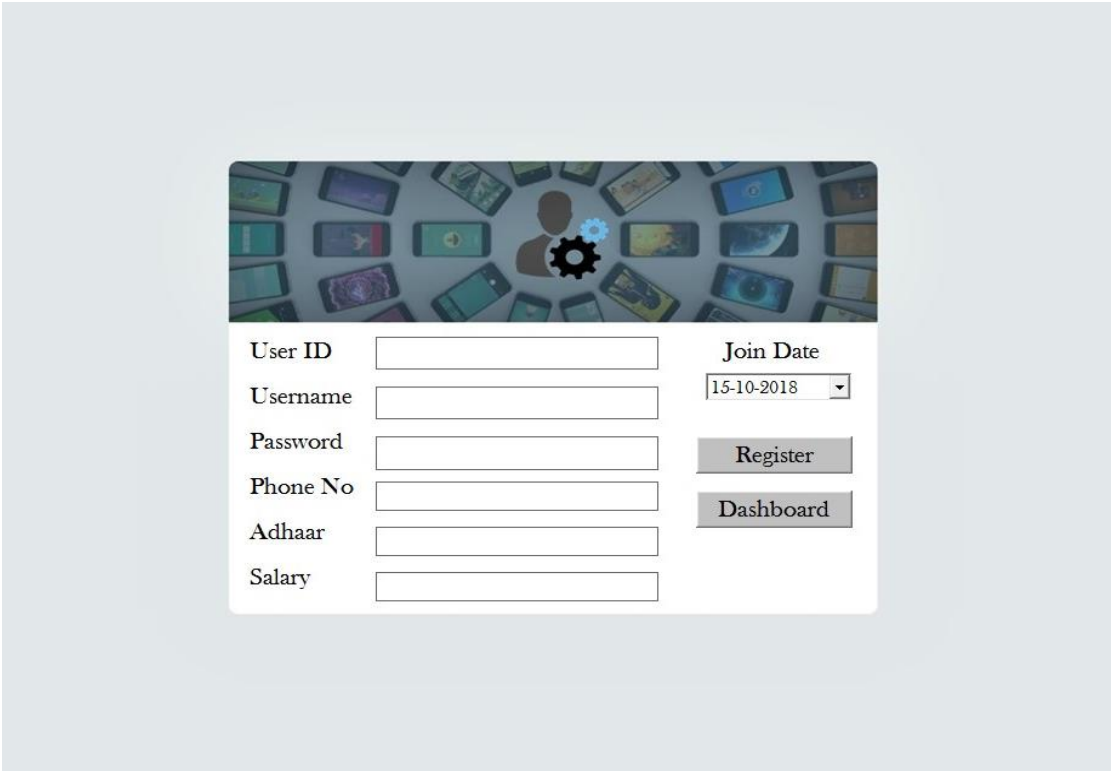
```
Private Sub ex_Click()
Unload Me
End Sub

Private Sub Form_Load()
Module1.connect
paygrid.Visible = False
empgrid.Visible = False
custgrid.Visible = False
ordergrid.Visible = False
mobgrid.Visible = False
DTPicker1.Value = Date
DTPicker2.Value = Date
Me.Left = (Screen.Width - Me.Width) / 2
Me.Top = (Screen.Height - Me.Height) / 2
End Sub
```

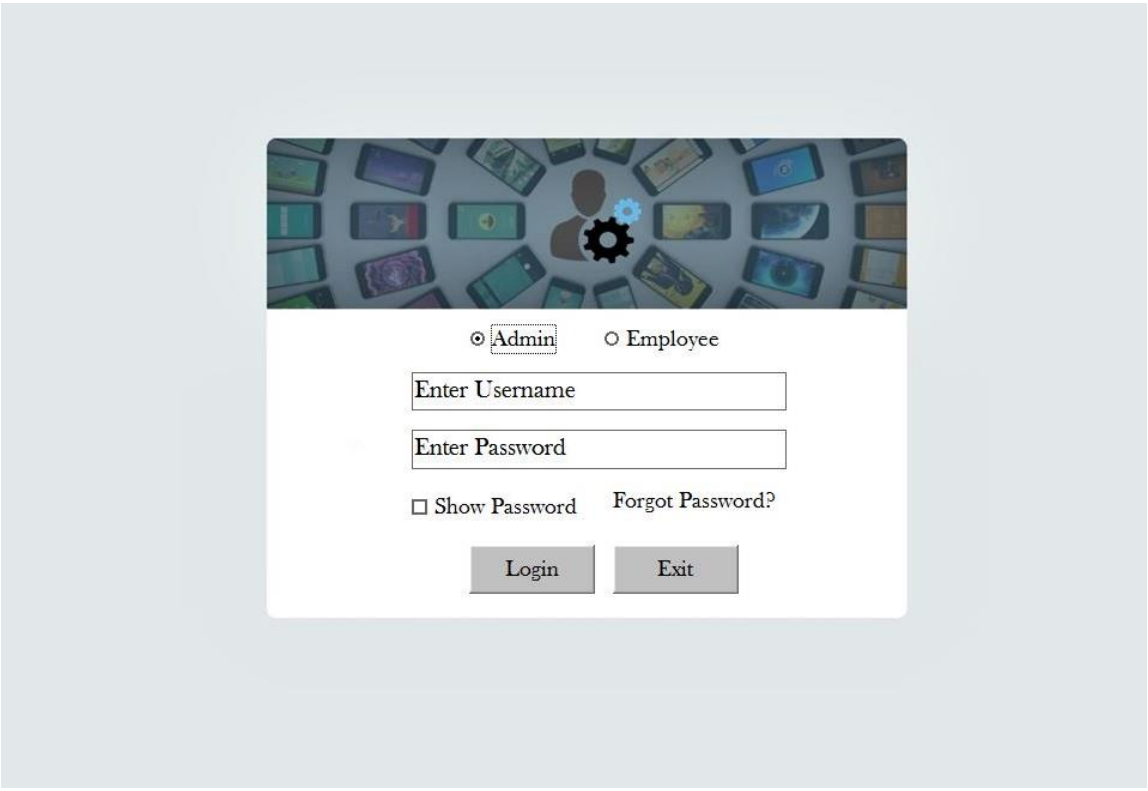
Intro Splash Screen

Login Success Splash

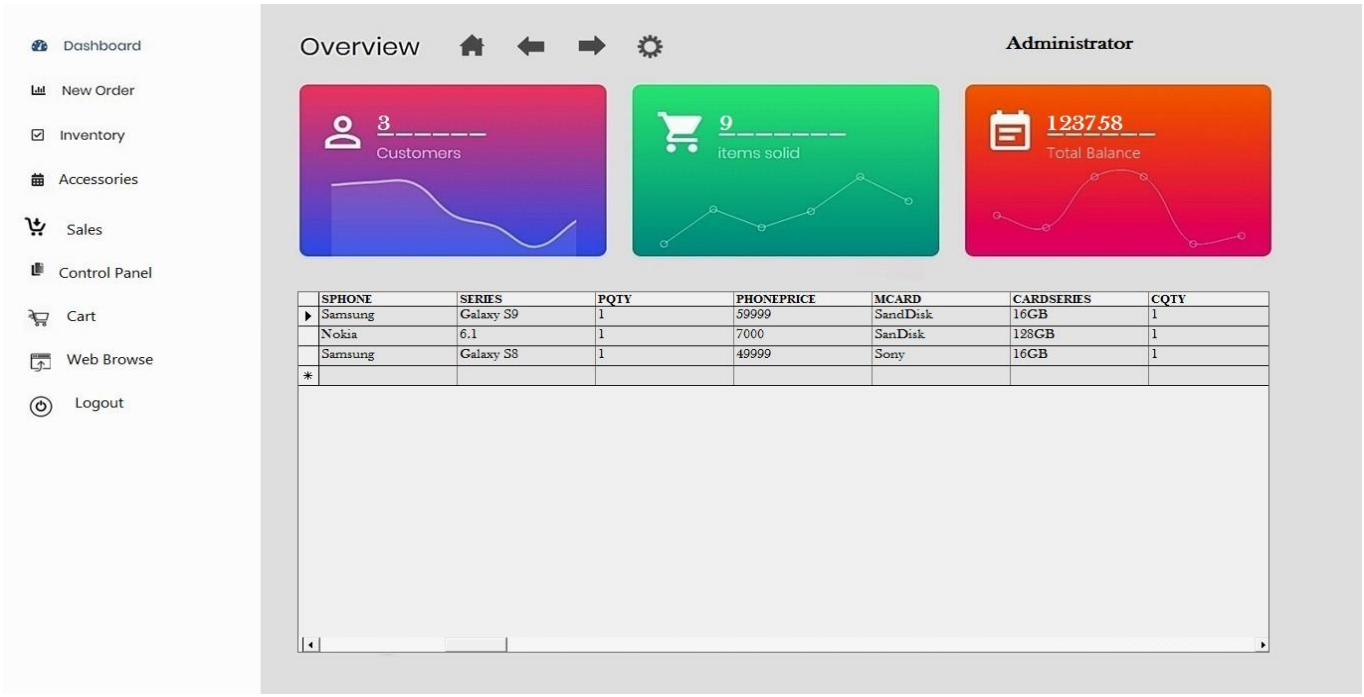
Register Screen



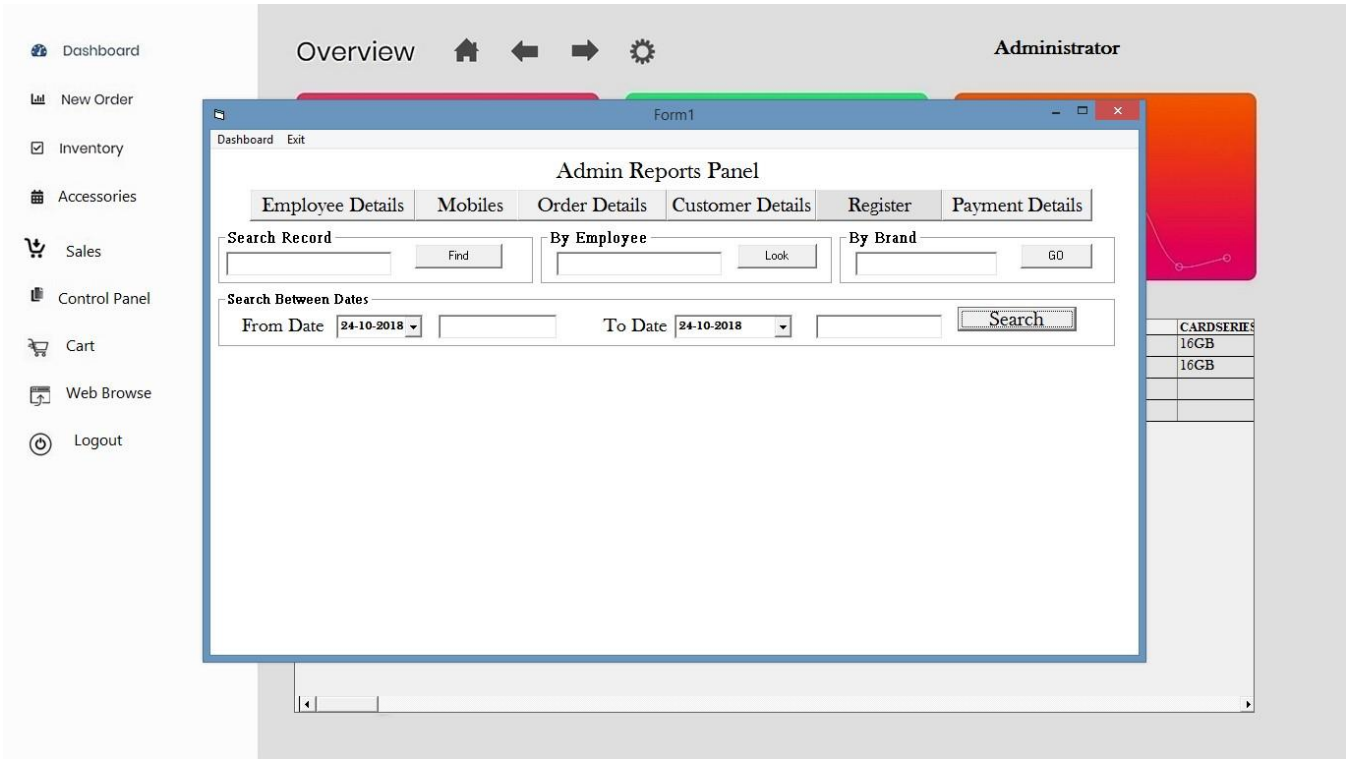
Login Screen



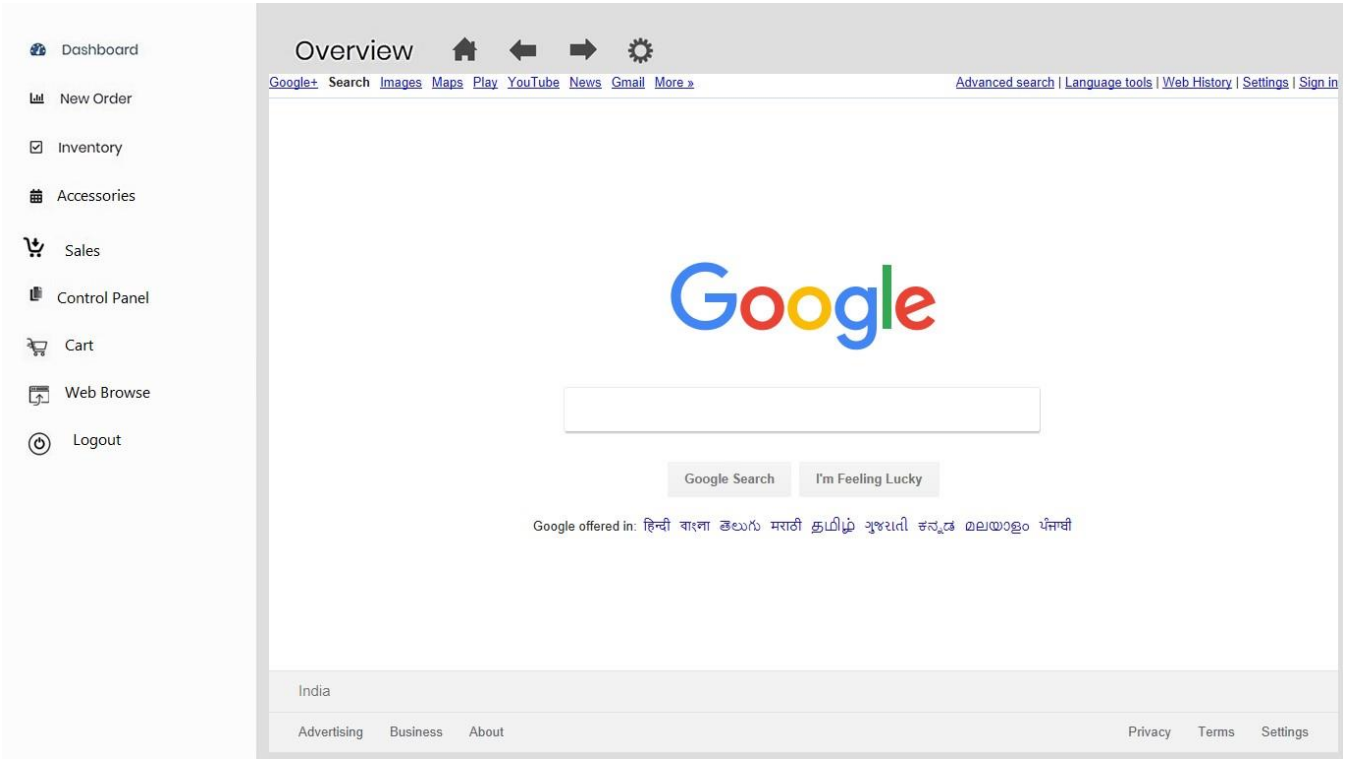
Dashboard



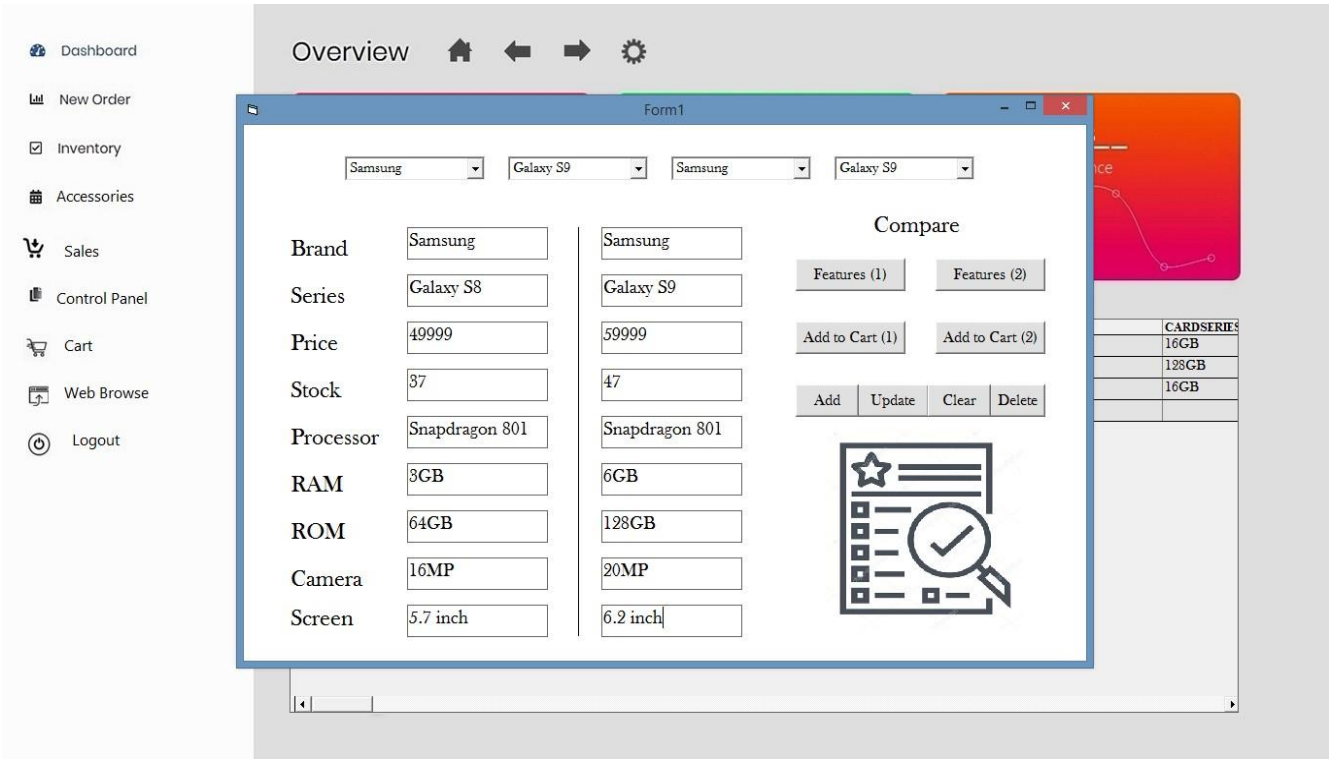
Admin Control Panel



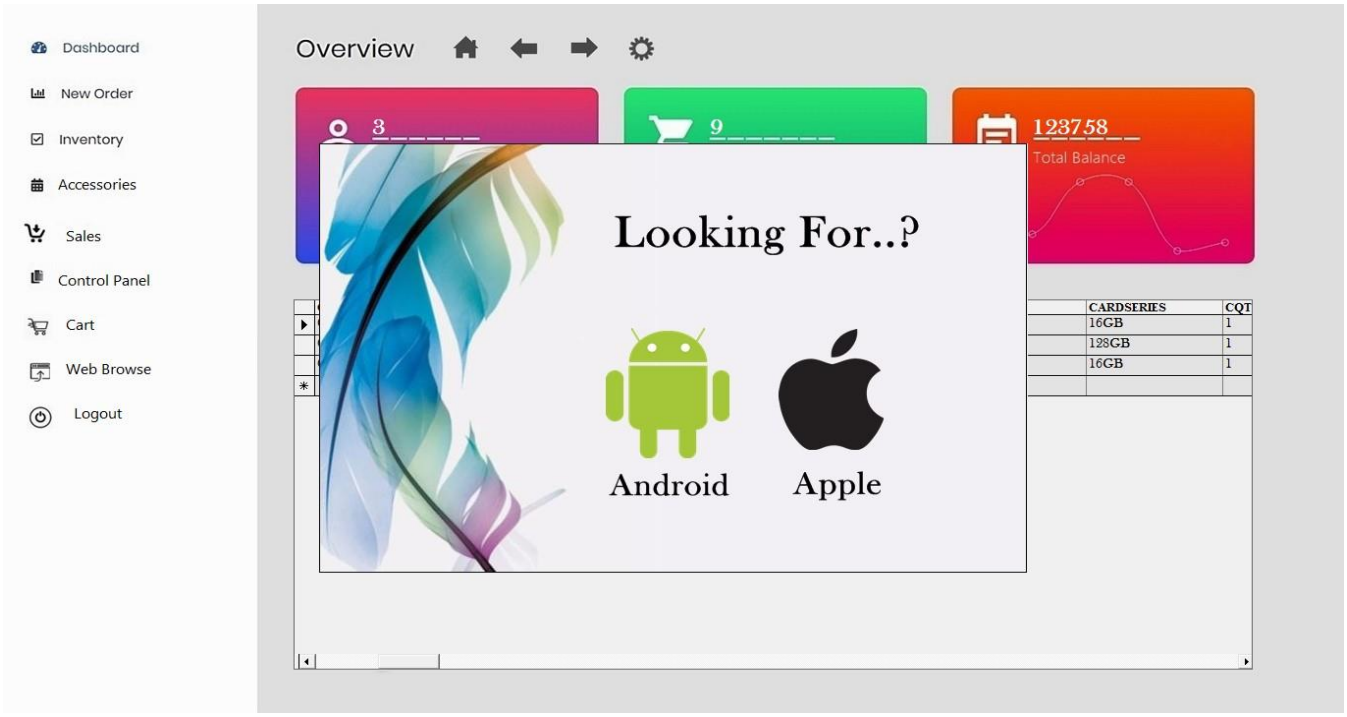
Web Browser



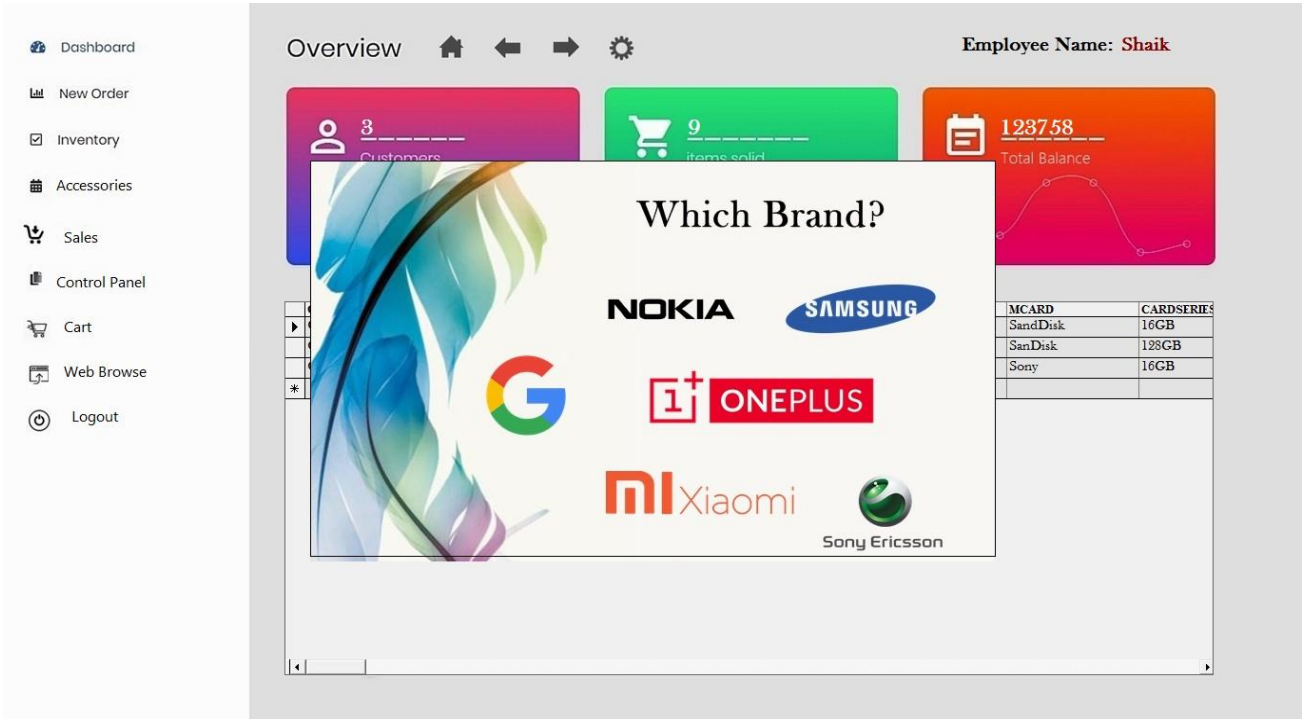
Inventory



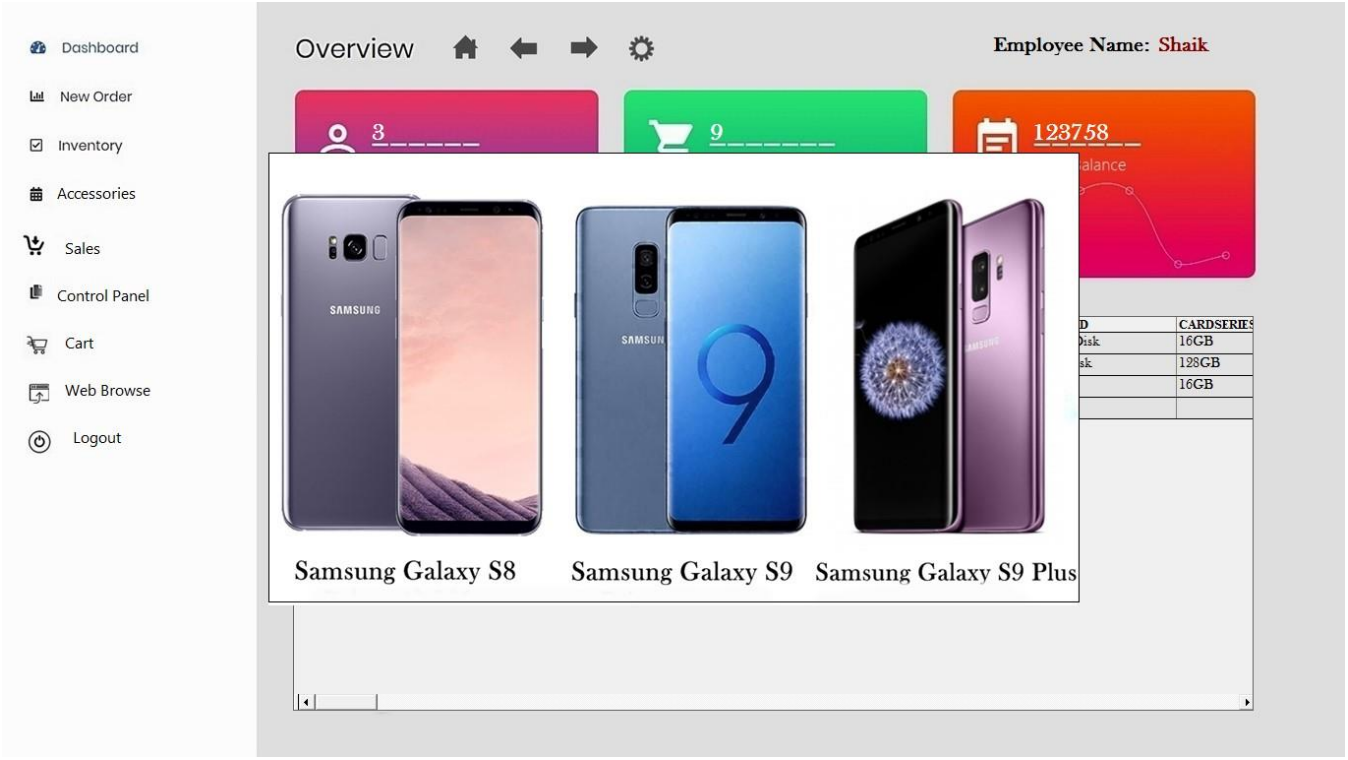
Option Selection



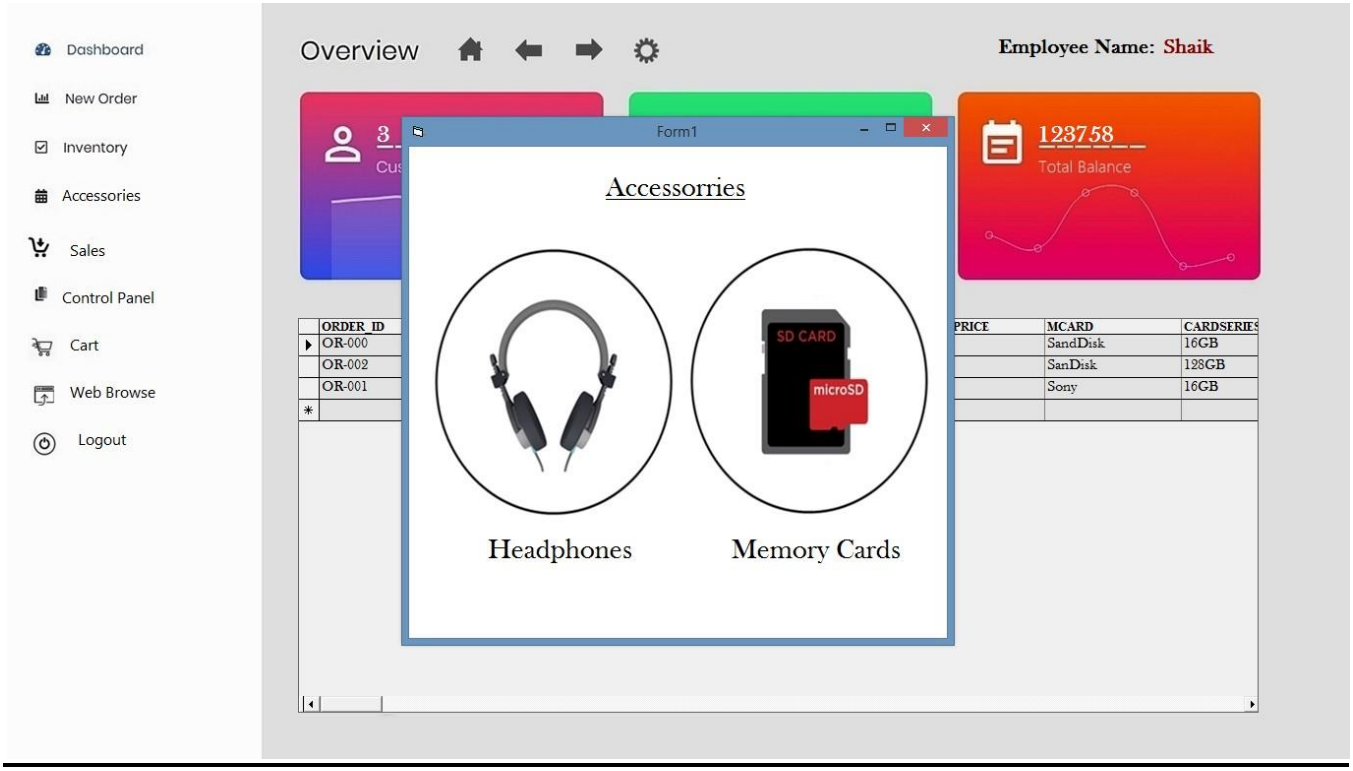
Brands



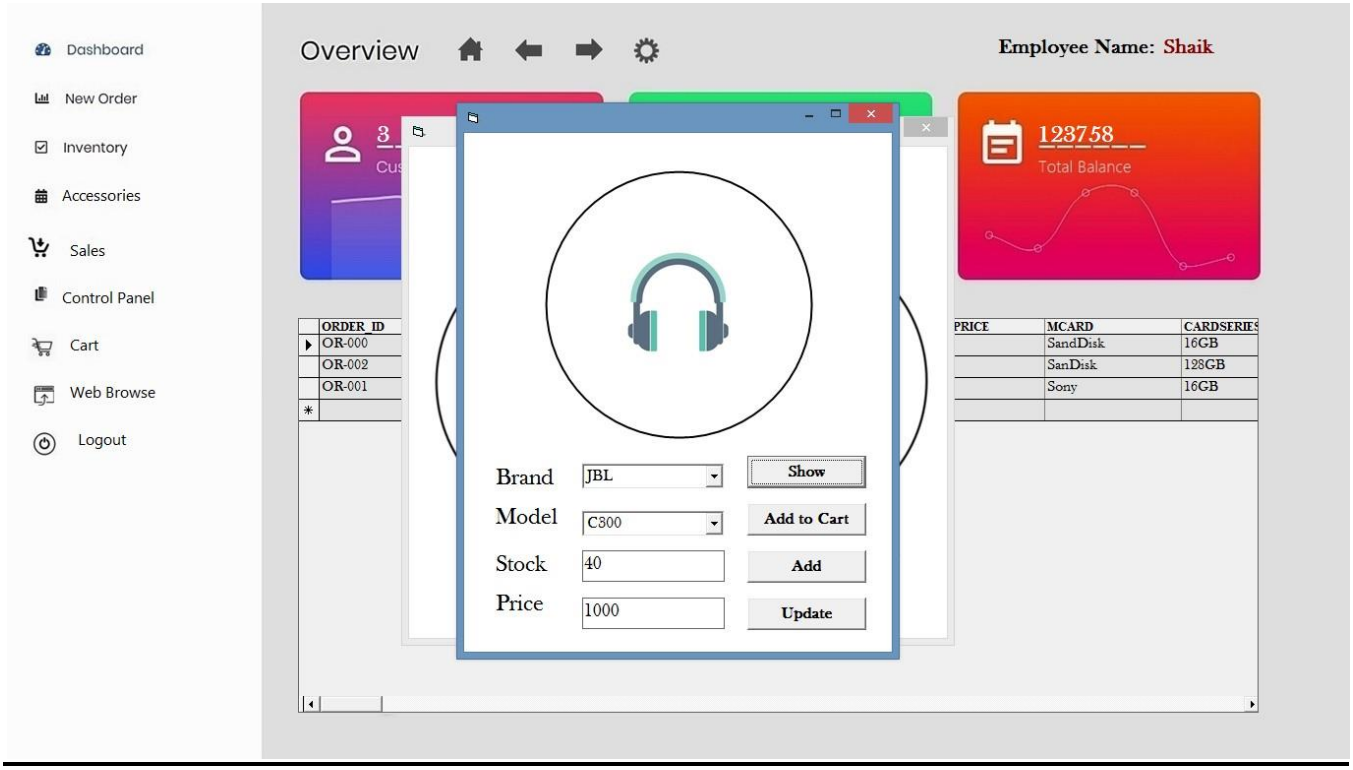
Selection



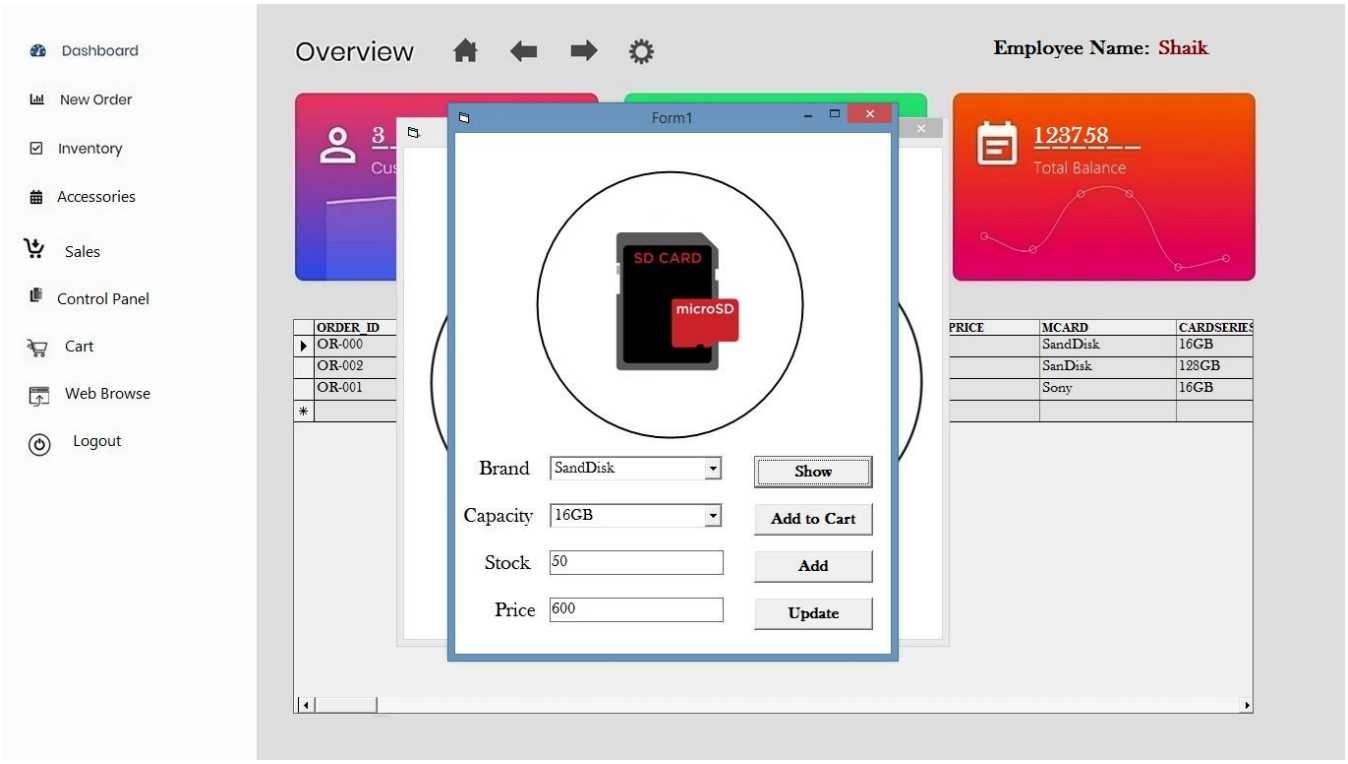
Accessories



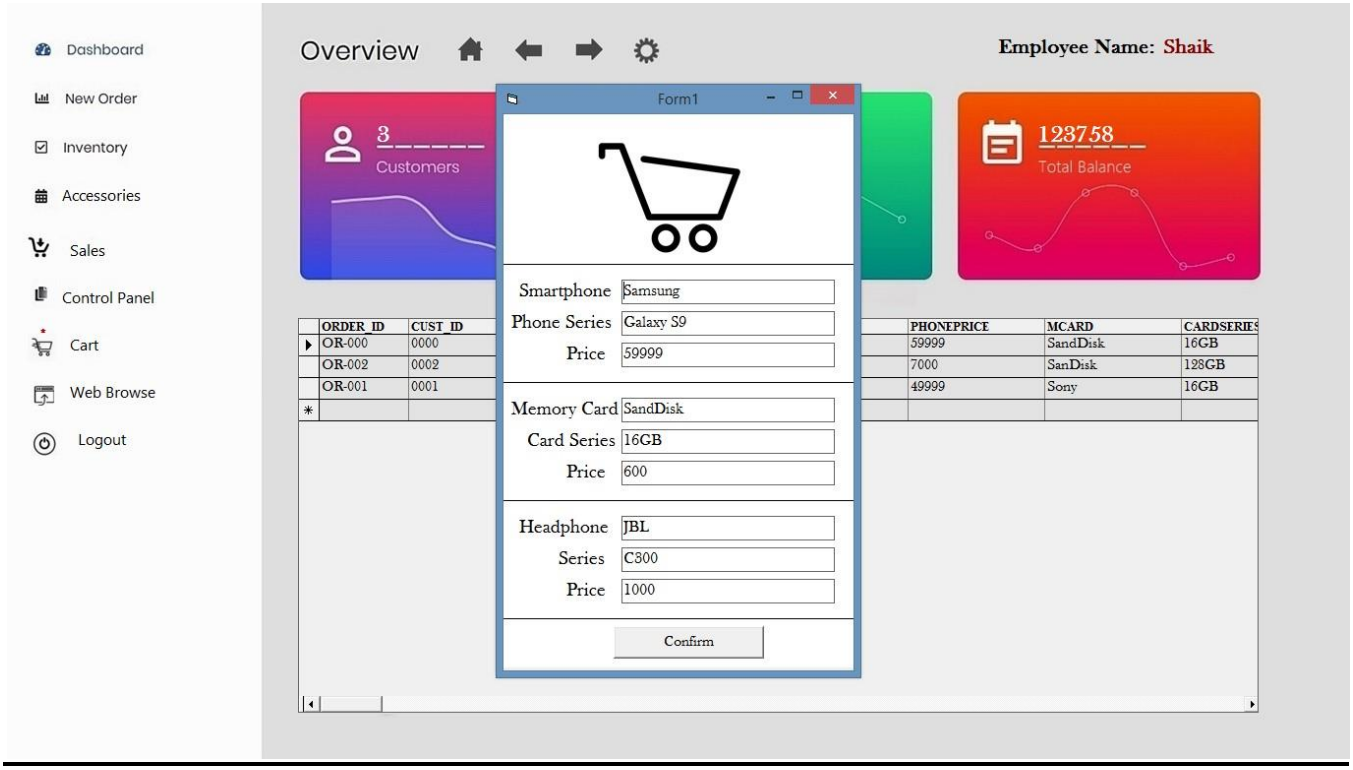
Headphones



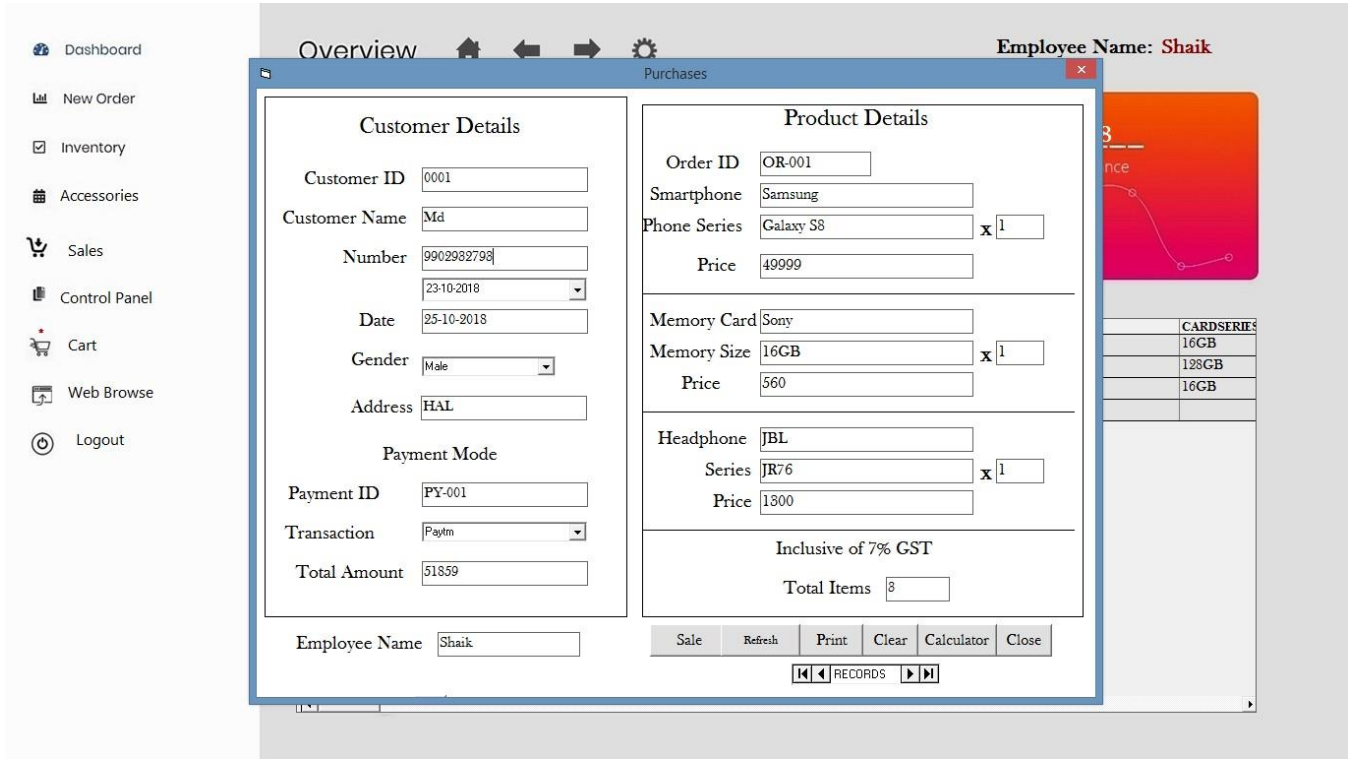
Memory Cards



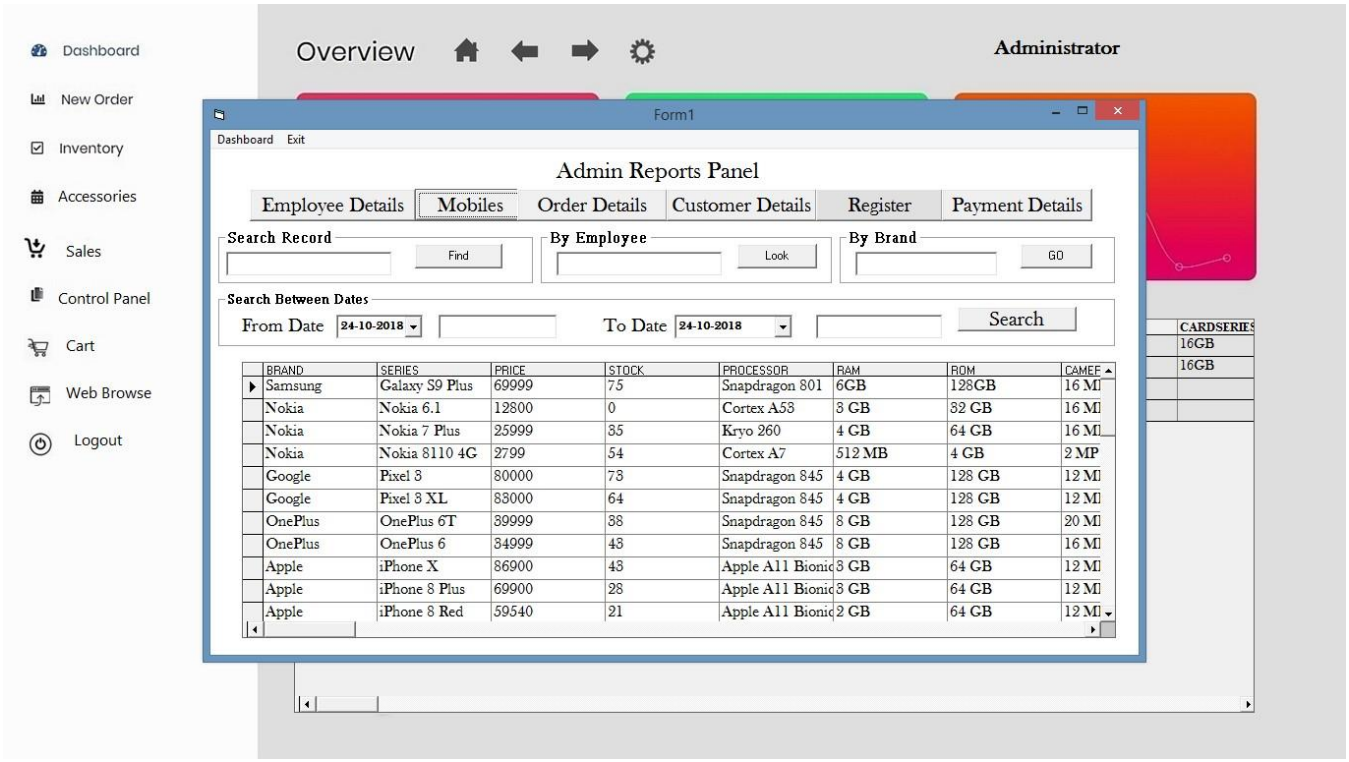
Sales Cart



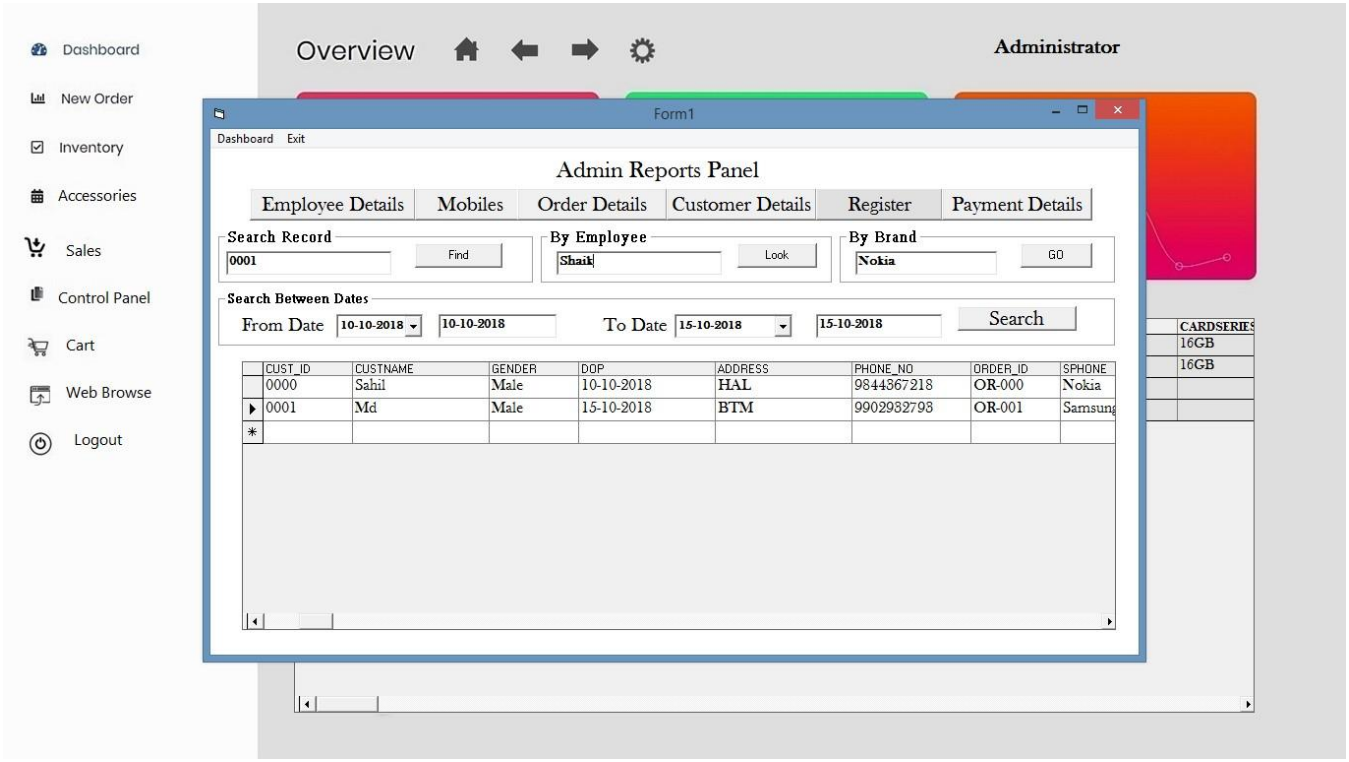
Sales



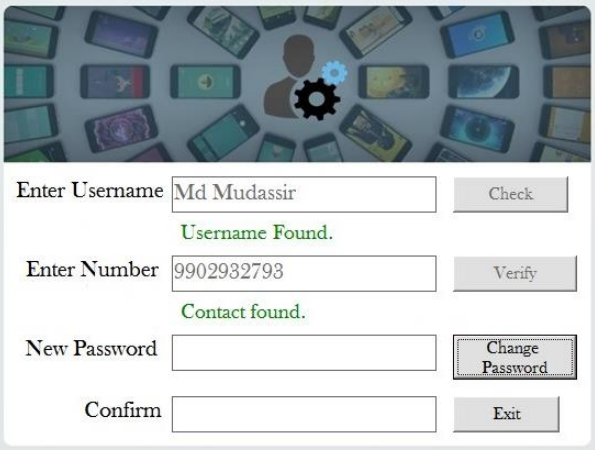
Mobiles Stock



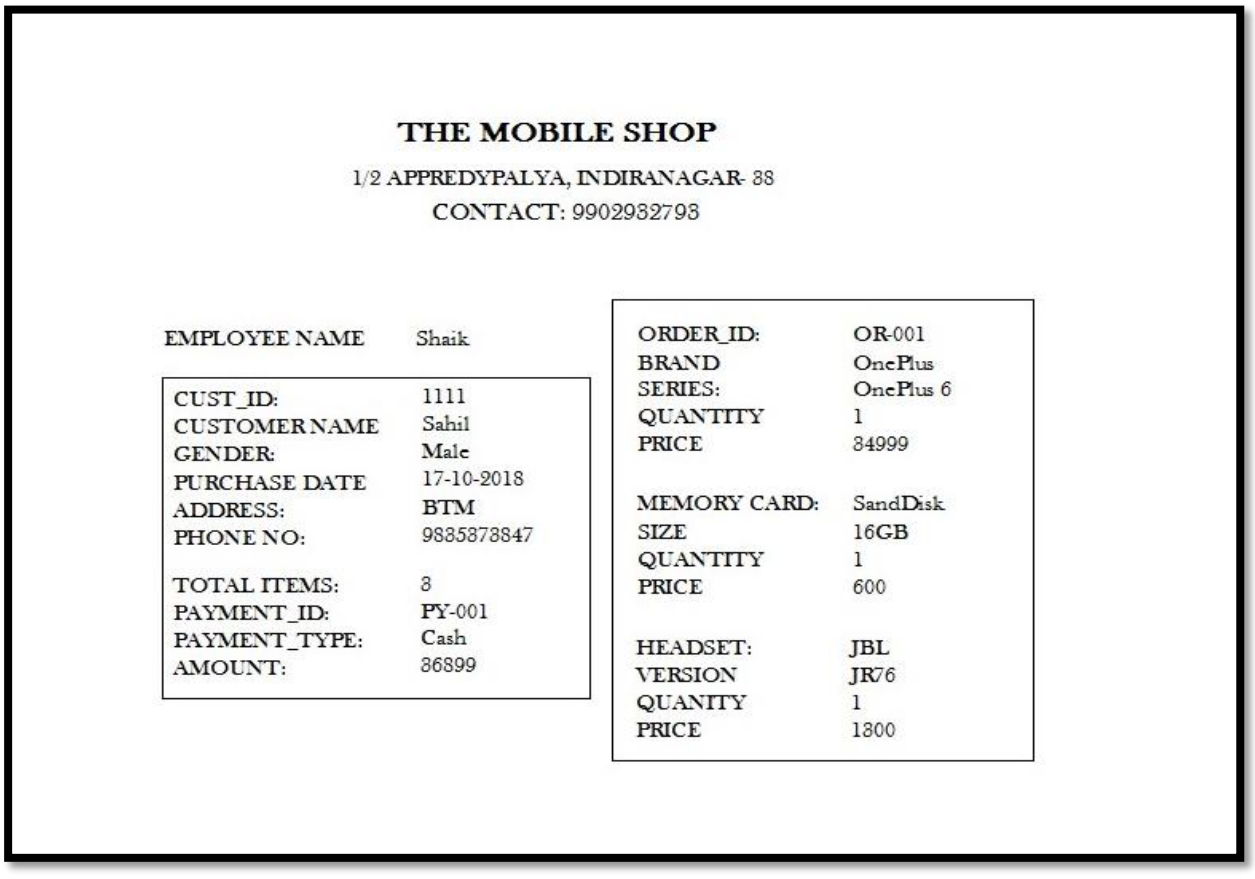
Search between dates



Forgot Password



Printed Bill



THE MOBILE SHOP

1/2 APPREDYPALYA, INDIRANAGAR- 88

CONTACT: 9902932793

EMPLOYEE NAME Shaik

CUST_ID:	1111
CUSTOMER NAME	Sahil
GENDER:	Male
PURCHASE DATE	17-10-2018
ADDRESS:	BTM
PHONE NO:	9885378847
TOTAL ITEMS:	3
PAYMENT_ID:	PY-001
PAYMENT_TYPE:	Cash
AMOUNT:	86899

ORDER_ID:	OR-001
BRAND	OnePlus
SERIES:	OnePlus 6
QUANTITY	1
PRICE	84999
MEMORY CARD:	SandDisk
SIZE	16GB
QUANTITY	1
PRICE	600
HEADSET:	JBL
VERSION	JR76
QUANITY	1
PRICE	1300

CHAPTER 8
TESTING

2. TESTING

2.1 Introduction

Testing:

Testing is the process of validation of evaluation a application item to detect difference between given input & expected output. Also to access the feature of a application item. Testing assesses the quality of the product. Application testing is a process that should be done during the development process. In other words application testing is a verification & validation process.

Types of Testing:

There are many types of testing:

- Unit Testing.
- Integration Testing.
- Functional Testing.
- System Testing.
- Stress Testing.
- Performance Testing.
- Usability Testing.
- Acceptance Testing.
- Regression Testing.
- Beta Testing.

Unit Testing:

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

Integration Testing:

Integration testing is a testing in which a group of components are combined to produce output. Also, the interaction between application & hardware is tested in integration testing if application & hardware components have any relation. It may fall under both white box & black box testing.

Functional Testing:

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

System Testing:

System testing is the testing to ensure that by putting the application in different environments (e.g. operating Systems) it still works. System testing is done with full system implementations & environment. It falls under the class of black box testing.

Stress Testing:

Stress testing is the testing to evaluate how the system behaves under unfavourable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

Performance Testing:

Performance testing is the testing to assess the speed & effectiveness of the system & to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

Usability Testing:

Usability testing is performed to the perspective of the client, to evaluate how the GUI is use-friendly. How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

Acceptance Testing:

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements & works as the customer expected. It falls under the class of black box testing.

Regression Testing:

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly & is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

Beta Testing:

Beta testing is the testing which is done by end users, a team outside development, or publicly releasing full pre-version of the product which is known as beta version. The aim of beta testing is to cover unexpected errors. It falls under the class of black box testing.

2.2 Black Box Testing:

Black Box testing, also known as Behavioural Testing is a application testing method in which the internal structure/design/implementations of the item being tested is not known to the tester. These tests can be functional or non-functional, through usually functional.

This method is named so because the application program, in the eyes of the tester, is like a black box, inside which one cannot see.

Black box testing is contrasted with white box testing. View differences between black box & white box testing.

- Incorrect or missing functions.
- Interfaces errors.
- Errors in data structures or external database access.
- Behaviour or performance errors.
- Initialization & termination errors.

Advantages:

- ✓ Tests are done from user's point of view & will help in exposing discrepancies in the specifications.
- ✓ Tester need not know programming languages or how the application has been implemented.
- ✓ Tests can be conducted by a body independent from the developers, allowing for an objective perspective & the avoidance of developer-bias.
- ✓ Test cases can be designed as soon as the specifications are complete.

Disadvantages:

- ✓ Only a small number of possible inputs can be tested & many program paths will be left untested.

- ✓ Without a clear specification which is the situation in many projects, test cases will be difficult to design.
- ✓ Tests can be redundant if the application designer/developer has already run a test case.
- ✓ Ever wondered why a soothsayer closes the eyes when foretelling events? So is almost the case in Black box testing?

2.3 White Box Testing:

White Box Testing also known as clear box testing, open box testing, is a application testing method in which the internal structure/design/implementation of the item being tested is known as to the tester. The tester chooses inputs to exercise paths through the code & determines the appropriate outputs. Programming know how & the implementation knowledge is essential. White box testing is testing beyond the user interfaces & into the nitty-gritty of a system/

The method is named so because the application program, in the eyes of the tester, is like a white/transparent box, inside which one clearly sees.

Advantages:

- ✓ Testing can be commenced at an earlier stage. One need not wait for the GUI to be available.
- ✓ Testing is more through, with the possibility of covering most paths.

Disadvantages:

- ✓ Since tests can be very complex, highly skilled resources are required, with through knowledge of programming & implementation.
- ✓ Test script maintenance can be a burden if the implementation changes too frequently.
- ✓ Since this method of testing it closely tied with the application being testing, tools to cater to every kind of implementation/platform may not be readily available.
- ✓ It is like the work of a mechanic who examines the engine to see why the car is not moving.

8.4 Testing Strategies:

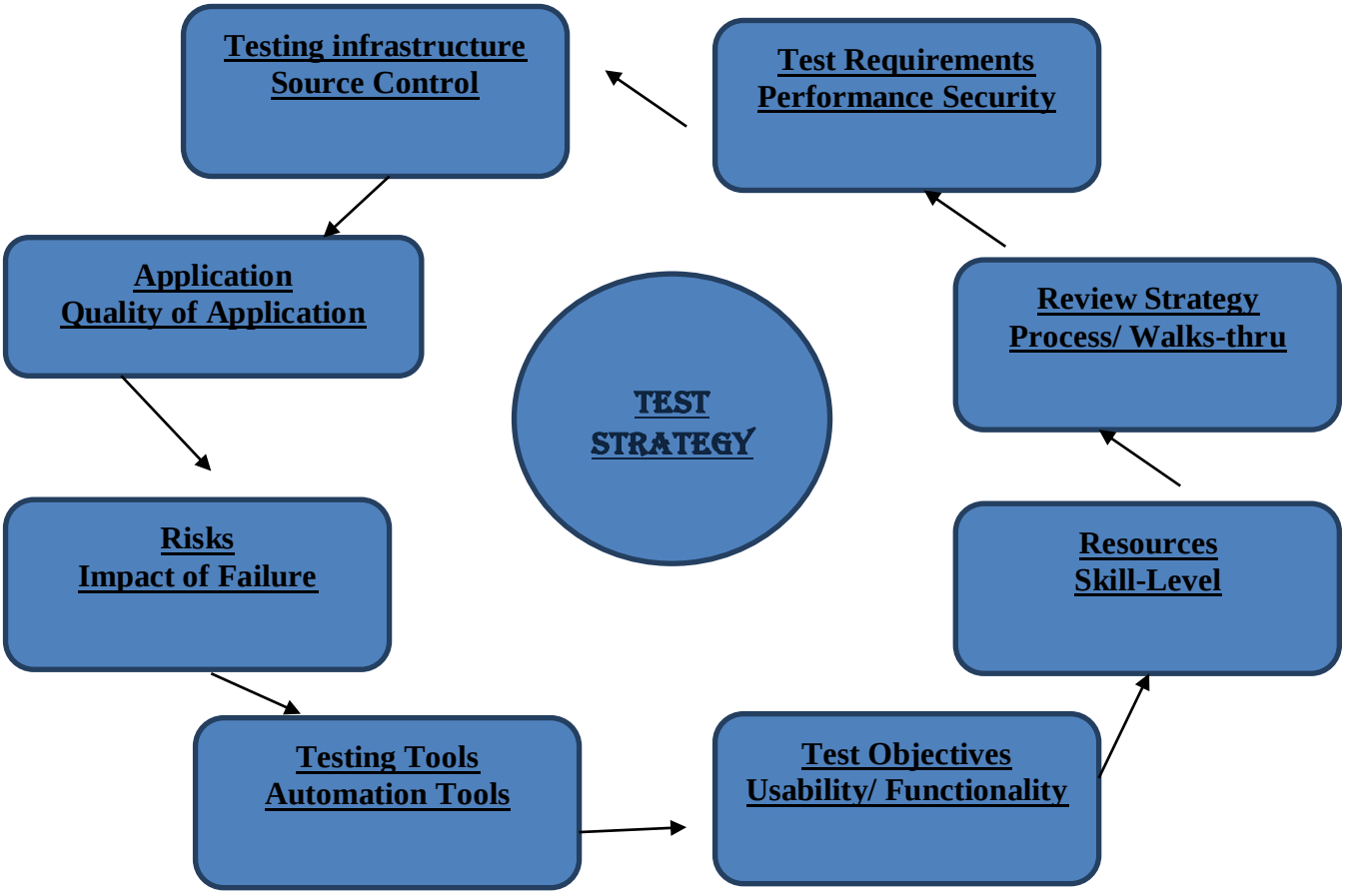
A test strategy is a high level document & normally developed by project manager. This document defines "Application Testing Approach" to achieve testing objectives. The test strategy is a static

document meaning that it is not updated too often. It sets the standards for testing processes & activities & other documents such as the test plan draws its contents from those standards set in the test strategy document.

A test strategy will typically cover the following aspects:

- ✓ Definition of test objectives.
- ✓ Strategy to meet the specified objective.
- ✓ Overall testing approach.
- ✓ Test environment & etc.

A Block Diagram of Test Strategy is shown below:



Test Strategy:

A test strategy is a high level document & normally developed by project manager. This document defines "Application Testing Approach" to achieve testing objectives. The test strategy is a static document meaning that it is not updated too often. It sets the standards for testing

processes & activities & other documents such as the test plan draws its contents from those standards set in the test strategy document.

Some companies include the "Test Approach" or "Strategy" inside the test plan, which is fine & it is usually the case for small projects. However, for larger projects, there is one test strategy document & different number of test plan for each phase or level of testing.

Components of the Test Strategy document:

- Scope & Objectives.
- Business issues.
- Roles & responsibilities.
- Communication & status reporting.
- Test deliverability.
- Industry standards to follow.
- Test automation & tools.
- Testing measurements & metrics.
- Risks & mitigation.
- Defect reporting & tracking.
- Change & configuration management.
- Training plan.

8.4.1 Unit Testing:

Unit testing is a level of the application testing process where individual units/components of a application/system are tested. The purpose is to validate that each unit of the application performs as designed.

A unit is the smallest testable part of application. It usually has one or a few inputs & usually a single output. In procedural programming a unit may be an individual program, function, procedure, etc. in object oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. Unit testing frameworks, drivers, stubs & mock or fake objects are used to assist in unit testing.

Method:

Unit testing is performed by using the white box testing method.

When is it performed?

Unit testing is the first level of testing & is performed prior to integration testing.

Who performs it?

Unit testing is normally performed by application developers themselves or their peers. In rare cases it may also be performed by independent application testers.

Tasks:

- Unit test plan
- Prepare
- Review
- Rework

8.4.2 Integration Testing:

Integration testing is a level of the application testing process where individual units are combined & tested as a group.

The purpose of this level of testing is to expose faults in the interaction between integrated units.

Test drivers & test stubs are used to assist in integration testing.

Tasks:

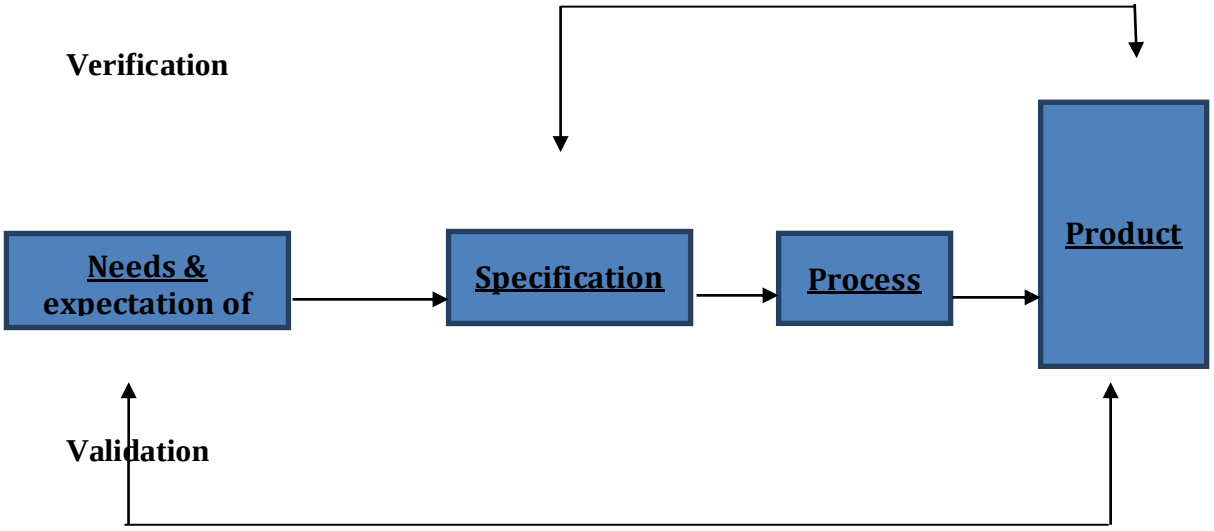
- Integration Test plan
- Prepare
- Review
- Rework
- Baseline

- Integration test cases/ scripts
- Prepare
- Review
- Rework
- Baseline
- Integration test

- perform

8.4.3 Validation Testing:

- Determining if the system complies with the requirements & performs functions for which it is intended & meet the organization's goals & user needs.
- Validation is done at the end of the development process and takes place after verification are completed.
- It answers the question like: AM I BUILDING THE RIGHT PRODUCT?
- Am I accessing the right data (in terms of the data required to satisfy the requirements).
- It is a high level activity.
- Performed after a work product is produced against established criteria ensuring that the product integrates correctly into the environment.
- Determination of correctness of the final application product by a development project with respect to the user needs & requirements.



According to the capability maturity model (CMMI-SW v1.1) we can also define validation as the process of evaluating application during or at the end development process to determine whether it satisfies specified requirements.

A product can pass while verification, as it is done on the paper & no running or functional application is required. But, when same points which were verified on the paper is actually developed then the running application or product can fail while validation. This may happen because when a product or application is built as per the specification but these specifications is not up to the mark hence they fail to address the user requirements.

Advantages of Validation:

- During verification if some defects are missed then during validation process it can be caught as failure.
- If during verification some specification is misunderstood and development had happen then during validation process while executing that functionality the difference between the actual result and expected result can be understood.
- Validation is done during testing like feature testing, integration testing, system testing, load testing, compatibility testing, stress testing, etc.
- Validation helps in building the right product as per the customer's requirements and helps in satisfying their needs.

Validation is basically done by the testers during the testing. While validating the product if some deviation is found in the actual result from the expected result then a bug is reported or an incident is raised. Not all incidents are bugs. But all bugs are incidents. Incidents can also be of type 'Question' where the functionality is not clear to the tester.

Hence, validation helps in unfolding the exact functionality of the features and helps the testers to understand the product in much better way. It helps in making the product more users friendly.

8.5 Test Cases:

A test case, in application engineering, is a set of conditio0ns or variables under which a tester will determine whether an application, application system or one of its features is working as it was originally established for it to do. The mechanism for determining whether a application program or system has passed or failed such a test is known as test oracle.

In some settings, an oracle could be a requirement or use case, while in others it could be a heuristic. It may take many test cases it determine that a application program or is considered sufficiently scrutinized to be released. Test cases are often referred to as test scripts, particularly when written when they are usually collected into test suite.

8.5.1 Login:

Test Case Number	Check Item	Test case objective	Step to Execute	Test Data	Excepted Result	Actual Result
1.	Login Page	Leave all fields empty	Click login	----------	Prompt error message	Yes
2.	Login Page	Enter invalid username	Click Login	17757 Admin	Prompt error message	Yes
3.	Username	Enter valid username & password	Click login	User password	Open success form	Yes
4.	Login Page	Enter a valid username & wrong password	Click login	User 0000	Error Message	Yes

8.5.2 Sales:

Test case Number	Check Item	Test Case Objectives	Step to Execute	Test data	Excepted Result	Actual Result
1.	Customer ID	empty	Click sale	----------	Message "enter customer id"	Yes
2.	Gender	empty	Click sale	----------	Message "menion Gender"	No
3.	Mobile Number	Enter text	----------	hfsdh	Message "Text not allowed"	Yes
4.	Payment ID	empty	Click sale	----------	Message "enter payment id"	yes
5.	Total amount	empty	Click sale	----------	Message "enter total amount"	Yes
6.	Date	empty	Click sale	----------	Message "Select the date"	Yes
7.	Customer Name	Enter numbers	----------	34635	Message "number not allowed"	Yes

8.5.5 Admin control:

Test case Number	Check item	Test case objectives	Step to execute	Test data	Excepted result	Actual result
1.	Between dates	Add date2 < date1	Click search	---------	Enter correct date	No
2.	Record find	blank	Click find	0001	Show record	Yes
3.	Employee name	Enter wrong name.	Click look	jay	Error ”employee not found”	Yes
4.	brand	Show available brand	Click go	Nokia	Show Nokia fields	Yes
5.	Employee	Enter correct name.	Click look	Shaik	Show the record	Yes

8.5.6 Forgot Password

Test case Number	Check item	Test case objectives	Step to execute	Test data	Excepted result	Actual result
1.	username	Invalid username	Click check	user	Username doesn’t exist	Yes
2.	username	valid name	Click check	Shaik	Username found	Yes
3.	Phone no	Verify number	Click verify	4566753	Contact not found.....sorry can't reset the password!!	Yes
4.	Password reset	If password match then change	Click change pass	pass	Passwod does not match, Please enter correct details	Yes

CHAPTER 9
CONCLUSION

CONCLUSION

The Mobile Shop Management System has been a way of minimizing the clerical work, which is almost a routine and consumes the most precious time. This **MOBILE SHOP MANAGEMENT SYSTEM** has been an attempt to help the user to minimize his workload along with minimizing the paper works and saving of time.

The system has been developed in a way to make it very user friendly. Any person having a little bit of window based can run this system without any pain. Almost all the difficulties of manual reservation have been removed by this system.

The project entitled "**MOBILE SHOP MANAGEMENT SYSTEM"** was completed on time with total satisfaction after testing with possible sample data. The performance was found to be efficient and error free. This is a user friendly packaged application which is very easy to access and understand. Anyone with the knowledge of computer will find it very easy to use this application and perform various operations in it.

The main aim of making this application is to make the booking of orders easy fast and efficient. A conscious effort has been made to develop a application packages, making use of available tools, techniques and resources that would generate a proper system.

An eye has been kept on making it as user friendly and as flexible as possible. Therefore, one may hope that the system will be acceptable to any user and will adequately meet his/her needs. There have been limitations and short comings in the system development as well such as time constraints and limited exposure towards the respected field, and hence the project is still under modification.

In this project, first an attempt has been conducted to find the need of the system. To fulfil the needs, a detailed study had been conducted to find the various requirements of the system. This particular system has been designed in an attractive manner so that even a user with minimum knowledge can be able to operate the system easily.

This application combines the best of the world's i.e. programming language VISUAL BASICS 6.0 and database ORACLE 11g providing easy accessibility and security. It was developed to benefit the organizations and the customers. Finally the system was tested with real data and everything worked successfully. Thus the system has fulfilled all the objectives identified and is able to replace the existing system.

Chapter 10
Future Enhancement

10.Future Enhancement

Our "MOBILE SHOP MANAGEMENT SYSTEM" project has only one source place which is a major drawback of our company and it is restricted to only small vendors. Our system is standalone system which means it runs only on one computer so in future our application will be developed in such a way that it can be run on many computers with a decentralized database.

Chapter 11
Bibliography

11. BIBLIOGRAPHY

A bibliography is a list of the sources we used to get information for our project. It is included at the end of our report, on the last page.

Books Referred:

- ✓ Visual Basics Programming
- ✓ Database management System

Web Links:

- ✓ http://www.codeproject.com/Tips/351122/application-testing
- ✓ https://www.youtube.com/user/GetInCity
- ✓ https://www.youtube.com/user/sandydehrian