ABSTRACT

Jewellery management system is developed in ASP.NET, which can keep track of all your business activity in a jewellery shop from small segments to large and very large segments.

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE ORGANIZATION:

1.2 ABOUT THE PROJECT:

But maintaining the same quality or upgrading the present one is not an easy task because quality is the ultimate picture of the entire business. Good quality of a product depends on many factors e.g. sound infrastructure, better management control, etc. So to obtain the optimum quality, jewellers have to upgrade those ingredients by which the quality is affected. To upgrade those ingredients the jewellers have to depend on some types of data. So, if the decision making person of the business wants to have a grip on the total business, he/she will have to have a knowledge of the entire flow of data and information within the organisation .It cannot be done

without the help of a Business Related Software. Jewellery management system is developed in Asp.Net, which can keep track of all your business activity in a jewellery shop from small segments to large and very large segments.As we all know the jewellery trade can be divided into three major categories i.e.

1) Retail

2) Wholesale

3) Export

Main Features Of Jewellery Management System:

Creation of unlimited types of purity

Each purity can be divided into 50 grades depending on percentage of alloys.

- **Creation of Artisan／Dealer Master**

Every single information regarding the artisan/dealer can be stored here e.g. name & address of the artisan, making charge of an ornament etc.

- **Creation of Customer Master**

Every single information regarding the customer can be stored here e.g. name & address of the customer etc.

- **Creation of Stone Master**

There are many types of stones in the business which are categorised according to:

1) Diamond : It can be divided into many categories e.g. round, square, Marquise etc.

2) Colour Stones : Ruby, Pearl, Emerald, etc.

Module:

User

My Account

Product Details

Category

Order

PrintOrderForm

Admin

OrderCustomerDetails

OrderProductDetails

Category

OrderProcessed

OrderCancelled

Special Product

Product Reports

CHAPTER 2

2. SYSTEM REQUIREMENTS

As a Result of careful analysis of the requirements of developing this project and as per the needs of the project, the requirements are determined to the company. The Requirements are

being classified as Hardware and Software Requirements respectively. They are summarized in the form of tables as follows:

2.1 HARDWARE REQUIREMENTS:

Hardware interface describe the logical and physical characteristics of each interface between the software product and the hardware components of the system.

PROCESSOR	:	Dual core
HARD DISK	:	40GB
RAM	:	512MB
MONITOR	:	15”SVGA Digital Color Monitor
PEN DRIVE	:	512 MB
CD-ROM DRIVE	:	52X
MODEM	:	D Link 56Kbps
KEY BOARD	:	104 keys
MOUSE	:	Optical Mouse

2.2 SOFTWARE REQUIREMENTS:

Software interface describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries

and integrated commercial components. It describes the services needed and the nature of communications.

Front-End Design	:	Asp.Net with C# 2008
Back-End Database	:	Sql Server 2005
Operating System	:	Windows XP, Vista, Windows 7

CHAPTER 3

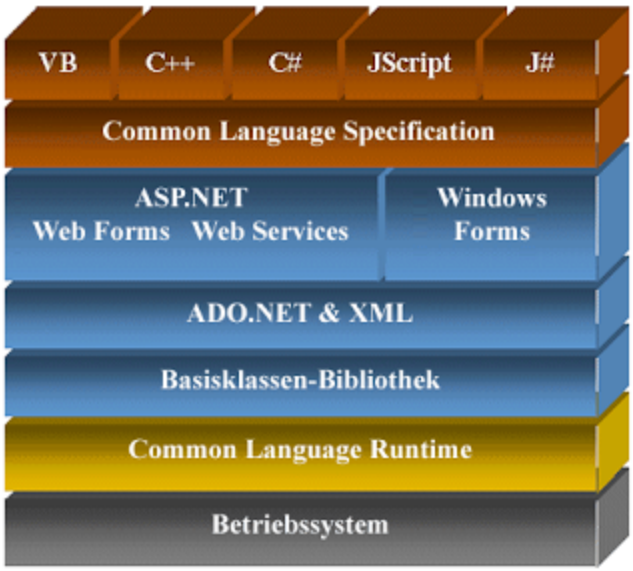
SYSTEM ENVIRONMENT

Introduction To .Net Framework

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

The .NET Framework has two main components: the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and Remoting, while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.



Figuter 3.3.0

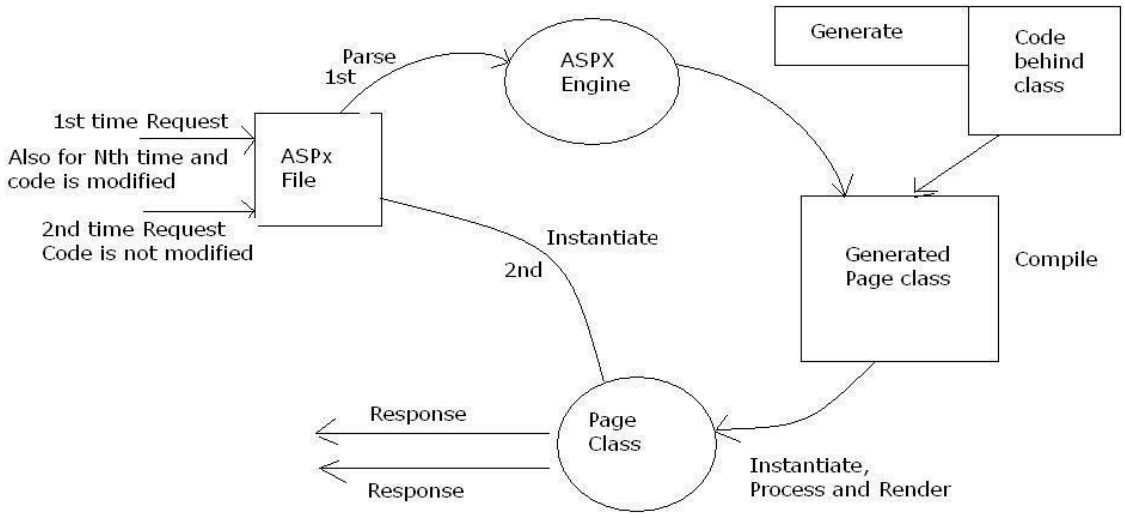
FEATURES OF THE COMMON LANGUAGE RUNTIME

The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registry-access operations, or other sensitive functions, even if it is being used in the same active application.

The runtime enforces code access security. For example, users can trust that an executable embedded in a Web page can play an animation on screen or sing a song, but cannot access their personal data, file system, or network. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally featuring rich.

The runtime also enforces code robustness by implementing a strict type- and code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The various Microsoft and third-party language compilers



Figuter 3.3.1

While the runtime is designed for the software of the future, it also supports software of today and yesterday. Interoperability between managed and unmanaged code enables developers to continue to use necessary COM components and DLLs.

The runtime is designed to enhance performance. Although the common language runtime provides many standard runtime services, managed code is never interpreted. A feature called just-in-time (JIT) compiling enables all managed code to run in the native machine language of the system on which it is executing. Meanwhile, the memory manager removes the possibilities of fragmented memory and increases memory locality-of-reference to further increase performance.

Finally, the runtime can be hosted by high-performance, server-side applications, such as Microsoft® SQL Server™ and Internet Information Services (IIS). This infrastructure enables you to use managed code to write your business logic, while still enjoying the superior performance of the industry's best enterprise servers that support runtime hosting.

.NET FRAMEWORK CLASS LIBRARY

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which your own managed code can derive functionality. This not only makes the .NET

Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework.

For example, the .NET Framework collection classes implement a set of interfaces that you can use to develop your own collection classes. Your collection classes will blend seamlessly with the classes in the .NET Framework.

As you would expect from an object-oriented class library, the .NET Framework types enable you to accomplish a range of common programming tasks, including tasks such as string management, data collection, database connectivity, and file access. In addition to these common tasks, the class library includes types that support a variety of specialized development scenarios. For example, you can use the .NET Framework to develop the following types of applications and services:

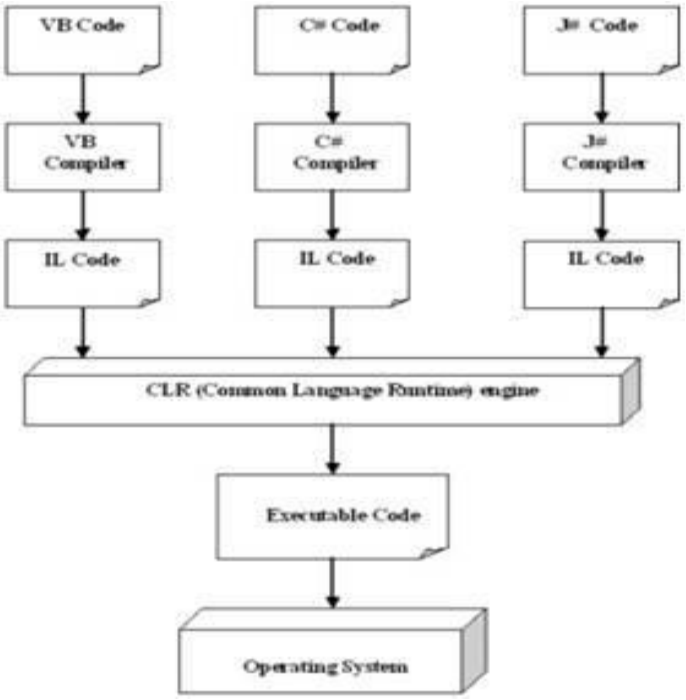
- Console applications.
- Scripted or hosted applications.
- Windows GUI applications (Windows Forms).
- ASP.NET applications.
- XML Web services.
- Windows services.

For example, the Windows Forms classes are a comprehensive set of reusable types that vastly simplify Windows GUI development. If you write an ASP.NET Web Form application, you can use the Web Forms classes.

CLIENT APPLICATION DEVELOPMENT

Client applications are the closest to a traditional style of application in Windows-based programming. These are the types of applications that display windows or forms on the desktop, enabling a user to perform a task. Client applications include applications such as word processors and spreadsheets, as well as custom business applications such as data-entry tools, reporting tools, and so on. Client applications usually employ windows, menus, buttons, and

other GUI elements, and they likely access local resources such as the file system and peripherals such as printers.



Figuter 3.3.2

The Windows Forms classes contained in the .NET Framework are designed to be used for GUI development. You can easily create command windows, buttons, menus, toolbars, and other screen elements with the flexibility necessary to accommodate shifting business needs.

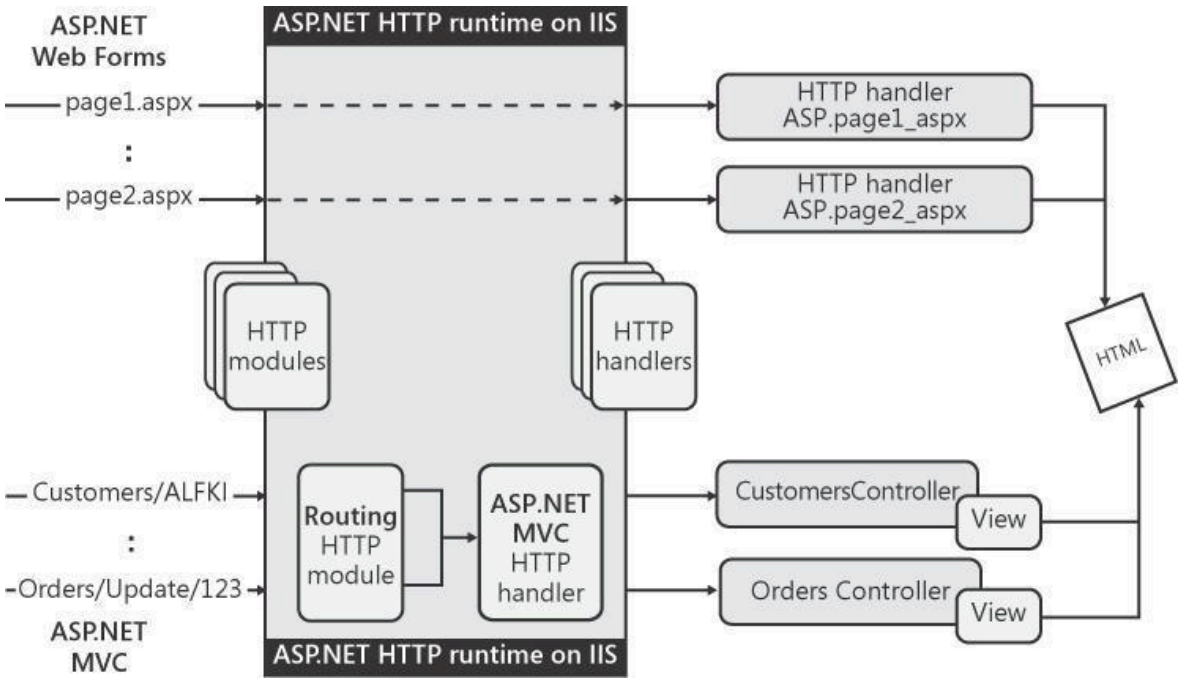
For example, the .NET Framework provides simple properties to adjust visual attributes associated with forms. In some cases the underlying operating system does not support changing these attributes directly, and in these cases the .NET Framework automatically recreates the forms. This is one of many ways in which the .NET Framework integrates the developer interface, making coding simpler and more consistent.

ASP.NET

Server Application Development

Server-side applications in the managed world are implemented through runtime hosts. Unmanaged applications host the common language runtime, which allows your custom managed code to control the behavior of the server. This model provides you with all the features of the common language runtime and class library while gaining the performance and scalability of the host server.

The following illustration shows a basic network schema with managed code running in different server environments. Servers such as IIS and SQL Server can perform standard operations while your application logic executes through the managed code.



Figuter 3.3.3

SERVER-SIDE MANAGED CODE

ASP.NET is the hosting environment that enables developers to use the .NET Framework to target Web-based applications. However, ASP.NET is more than just a runtime host; it is a complete architecture for developing Web sites and Internet-distributed objects using managed code. Both Web Forms and XML Web services use IIS and ASP.NET as the publishing

mechanism for applications, and both have a collection of supporting classes in the .NET Framework.

XML Web services, an important evolution in Web-based technology, are distributed, server-side application components similar to common Web sites. However, unlike Web-based applications, XML Web services components have no UI and are not targeted for browsers such as Internet Explorer and Netscape Navigator. Instead, XML Web services consist of reusable software components designed to be consumed by other applications, such as traditional client applications, Web-based applications, or even other XML Web services. As a result, XML Web services technology is rapidly moving application development and deployment into the highly distributed environment of the Internet.

If you have used earlier versions of ASP technology, you will immediately notice the improvements that ASP.NET and Web Forms offers. For example, you can develop Web Forms pages in any language that supports the .NET Framework. In addition, your code no longer needs to share the same file with your HTTP text (although it can continue to do so if you prefer). Web Forms pages execute in native machine language because, like any other managed application, they take full advantage of the runtime. In contrast, unmanaged ASP pages are always scripted and interpreted. ASP.NET pages are faster, more functional, and easier to develop than unmanaged ASP pages because they interact with the runtime like any managed application.

ACTIVE SERVER PAGES.NET

ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET offers several important advantages over previous Web development models:

- **Enhanced Performance.** ASP.NET is compiled common language runtime code running on the server. Unlike its interpreted predecessors, ASP.NET can take advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box. This amounts to dramatically better performance before you ever write a line of code.
- **World-Class Tool Support.** The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing,

drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool provides.

- **Power and Flexibility.** Because ASP.NET is based on the common language runtime, the power and flexibility of that entire platform is available to Web application developers. The .NET Framework class library, Messaging, and Data Access solutions are all seamlessly accessible from the Web. ASP.NET is also language-independent, so you can choose the language that best applies to your application or partition your application across many languages. Further, common language runtime interoperability guarantees that your existing investment in COM-based development is preserved when migrating to ASP.NET.
- **Simplicity.** ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration. For example, the ASP.NET page framework allows you to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic - like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.

LANGUAGE SUPPORT

The Microsoft .NET Platform currently offers built-in support for three languages: C#, Visual Basic, and JScript.

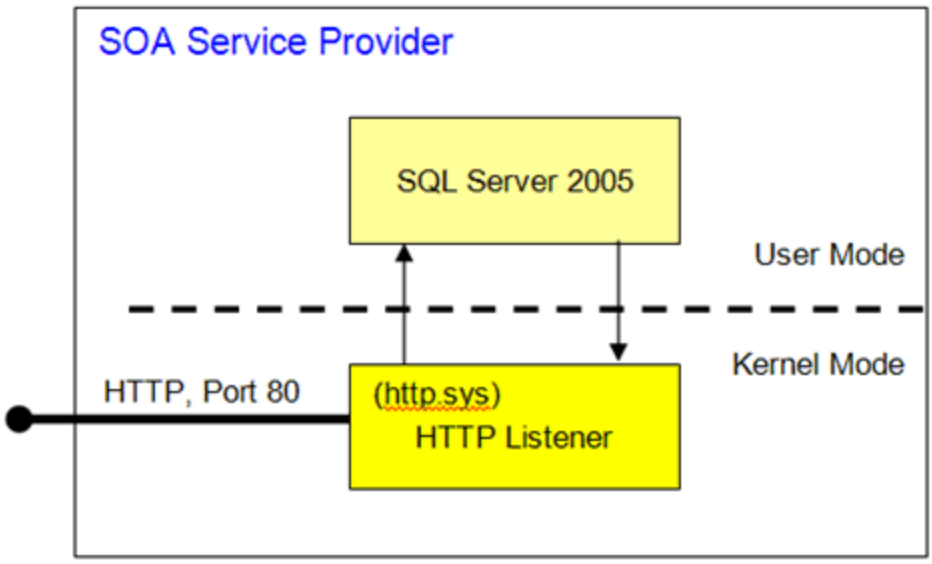
SQL SERVER

A database management, or DBMS, gives the user access to their data and helps them transform the data into information. Such database management systems include dBase, paradox, IMS, SQL Server and SQL Server. These systems allow users to create, update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. SQL Server stores each data item in its own fields. In SQL Server, the fields relating to a particular person, thing or event are bundled together to form a single complete unit

of data, called a record (it can also be referred to as raw or an occurrence). Each record is made up of a number of fields. No two fields in a record can have the same field name.

During an SQL Server Database design project, the analysis of your business needs identifies all the fields or attributes of interest. If your business needs change over time, you define any additional fields or change the definition of existing fields.



Figuter 3.3.4

SQL SERVER TABLES

SQL Server stores records relating to each other in a table. Different tables are created for the various groups of information. Related tables are grouped together to form a database.

PRIMARY KEY

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

RELATIONAL DATABASE

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

FOREIGN KEY

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

REFERENTIAL INTEGRITY

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

ADVANTAGES OF RDBMS

- Redundancy can be avoided
- Inconsistency can be eliminated
- Data can be Shared
- Standards can be enforced
- Security restrictions ca be applied

- Integrity can be maintained
- Conflicting requirements can be balanced
- Data independence can be achieved.

DISADVANTAGES OF DBMS

A significant disadvantage of the DBMS system is cost. In addition to the cost of purchasing of developing the software, the hardware has to be upgraded to allow for the extensive programs and the workspace required for their execution and storage. While centralization reduces duplication, the lack of duplication requires that the database be adequately backed up so that in case of failure the data can be recovered.

FEATURES OF SQL SERVER (RDBMS)

SQL SERVER is one of the leading database management systems (DBMS) because it is the only Database that meets the uncompromising requirements of today's most demanding information systems. From complex decision support systems (DSS) to the most rigorous online transaction processing (OLTP) application, even application that require simultaneous DSS and OLTP access to the same critical data, SQL Server leads the industry in both performance and capability

SQL SERVER is a truly portable, distributed, and open DBMS that delivers unmatched performance, continuous operation and support for every database.

SQL SERVER RDBMS is high performance fault tolerant DBMS which is specially designed for online transactions processing and for handling large database application.

SQL SERVER with transactions processing option offers two features which contribute to very high level of transaction processing throughput, which are

3.4 BENEFITS OF THE SOFTWARE

- Any application can talk to a host of other applications, running on diverse technology and hardware, in turn lowering the operation costs. Consider this one case, wherein the stock management system is connected to the accounting system, resulting in ample savings.
- Bring on one platform all your internal applications, your partners as well as your customers.
- Microsoft Visual Studio® .NET and the .NET Framework supports varied languages, in turn helping developers to focus on work at hand instead of trying to learn a new languagethatcandothejob.
- Employees can look for updated information on desktop applications, internet browsers or even mobile devices.
- Optimum speed of development
- Ability of cross platform migration
- High Reliability
- Rigorous Security
- Easy configurations of applications
- Vast and enriched Class library, features, controls
- ASP.NET Framework supports varied languages

12 important advantages ASP.NET offers over other Web development models:

1. ASP.NET drastically reduces the amount of code required to build large applications.

2. With built-in Windows authentication and per-application configuration, your applications are safe and secured.

3. It provides better performance by taking advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box.

4. The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing, drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool provides.

5. Provides simplicity as ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration.

6. The source code and HTML are together therefore ASP.NET pages are easy to maintain and write. Also the source code is executed on the server. This provides a lot of power and flexibility to the web pages.

7. All the processes are closely monitored and managed by the ASP.NET runtime, so that if process is dead, a new process can be created in its place, which helps keep your application constantly available to handle requests.

8. It is purely server-side technology so, ASP.NET code executes on the server before it is sent to the browser.

9. Being language-independent, it allows you to choose the language that best applies to your application or partition your application across many languages.

10. ASP.NET makes for easy deployment. There is no need to register components because the configuration information is built-in.

11. The Web server continuously monitors the pages, components and applications running on it. If it notices any memory leaks, infinite loops, other illegal activities, it immediately destroys those activities and restarts itself.

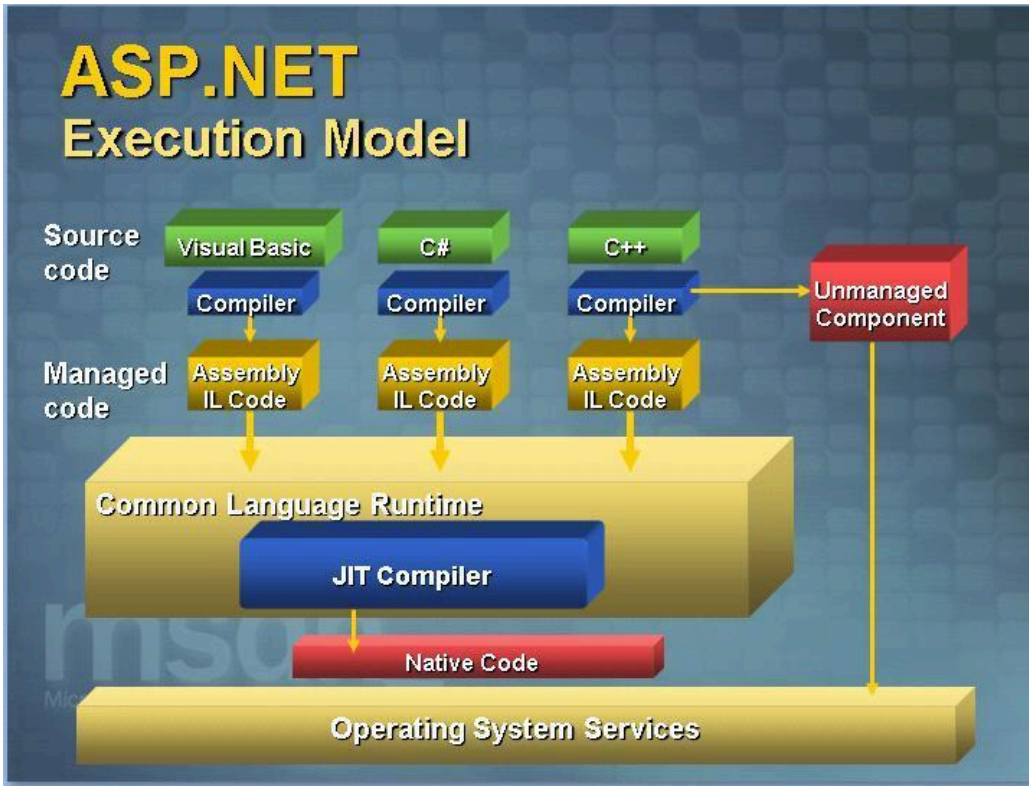
12. Easily works with ADO.NET using data-binding and page formatting features. It is an application which runs faster and counters large volumes of users without having performance problems

In short ASP.NET, the next generation version of Microsoft's ASP, is a programming framework used to create enterprise-class web sites, web applications, and technologies. ASP.NET developed applications are accessible on a global basis leading to efficient information management. Whether you are building a small business web site or a large corporate web application distributed across multiple networks, ASP.NET will provide you all the features you could possibly need...and at an affordable cost.

ASP.NET: Advantages

When choosing a programming framework to use, the most important things are the benefits it brings, the support it offers and its reliability. The **ASP.NET framework** incorporates all these characteristics, aiming every time for performance.

The first advantage that ASP.NET has over other frameworks, such as PHP and J2EE, is that it is developed on the Microsoft platform, and this confers the programmers access to the most **updated documentation**, reliable customer support from Microsoft through the MSDN service, innovative features and secured applications, thanks to the Windows built-in authentication. Meant as a propeller for dynamic web sites, web services and web applications, the .NET framework proves to be more than that, offering end users rich, easy to use and reliable web tools.



Figuter 3.4.0

Being a real object oriented (OOP) framework, ASP.NET offers better code management and a clean code structure. It also produces faster web applications using optimized compiled code, in comparison to the PHP language, which runs as interpreted code. Also, many project managers find the partition between code and markup (between logic and design) to be very effective and to allow a more organized and efficient work inside a team.

Another big plus that the ASP.NET web application framework brings is that it supports more than 25 mainstream coding languages, such as: **Visual Basic .Net, C++, C#**, JScript and others.

CHAPTER 4

SYSTEM ANALYSIS

EXISTING SYSTEM

At present all the activities in transaction are handled manually. Manual data processing system, whole providing economy, flexibility and adaptability at low data volumes become more complex when the volume of data becomes large. As an organization expands in size and function, a stage is reached when manual procedures become inadequate and inefficient. No matter how many clerks are employed a stage is reached then it becomes impossible to systemize such a large amount of information. What is required then is an upgrading in the class of information processing technology.

The present system is not sufficient to hold all the information that is necessary for the processing. So the library is in need of new computerized system, which is very flexible, user-friendly and capable of holding the system in a robust manner.

LIMITATIONS OF EXISTING SYSTEM

There were a lot of reasons for the introduction of the new system. They are mainly due to the drawbacks and efficiency of the existing system.

- Physical volume of the data is very large.
- The delay in information search and retrieval.
- Problems in updating and backup.
- Damage of papers containing the information.
- Considerable time taken for report generation.
- Accuracy of data is very lower in manual system.

PROPOSED SYSTEM

The system study phase studies the problem, identifies alternate solution, evaluates these solutions and finally recommends best solution. The system gives the structure and function of the system. A detailed system study is essential for developing an efficient system. The proposed system provides a better user interface. The system is a menu driven program.

ADVANTAGES OF PROPOSED SYSTEM

- The proposed system can be utilized for easy documenting and accessing various data carriers such as forms, reports, records etc.
- Automation makes the system to be user-friendly and hastily in manipulation and generation of valuable reports providing menu driven facilities.
- Accuracy and security of data will be more comfortable for the organization.
- Computerization will avoids human errors due to inexperience in data entry, manipulation etc.
- The paper work occurred in the manual system can be completely avoided.

FEASIBILITY STUDY

During system analysis, a feasibility study of the proposed system is carried out to see whether it is beneficial to the organization.

The integration unit is currently manual. To get the detailed information on production, bagging etc large bundles of files have to be looked into. It is very time consuming affair. An operator has to keep in mind or search a file for the details of department for the data. So working with the existing system is quite tedious. Whereas considering the merits of the new system it is very beneficial. The results of the feasibility study are given below:

TECHNICAL FEASIBILITY STUDY

It is a study of resource availability that may affect the availability to achieve an acceptable system. It is essential that the process of analysis and definition be conducted in parallel with an assessment of technical feasibility.

It centers on the existing computer system and to what extent it can support the proposed system. Though information in manual system is enormous, it is easily handled by the Access (which is a RDBMS software). It is easy to find and buy a system, which support this software. So it is technically feasible.

ECONOMIC FEASIBILITY STUDY

Tremendous is the range of changes that accompanies the new technology. Introduction of a computerized system has some merits and demerits can lead to monetary gains. The cost to buy a computer system for running this software is quite cheap. We get benefit because we serve more borrowers quickly and easily. So this system is economically feasible.

BEHAVIORAL FEASIBILITY STUDY

The hierarchy of the new system is much better than the old. The new system is very much user friendly and the operational cost is bearable. The maintenance and working of the new system needs less human effort.

CHAPTER 5

OBJECT ORIENTED ANALYSIS

LOGICAL DESIGN:

The logical flow of a system and define the boundaries of a system. It includes the following steps:

Reviews the current physical system - its data flows, file content, volumes , frequencies etc.

Prepares output specifications - that is, determines the format, content and frequency of reports.

Prepares input specifications - format, content and most of the input functions.

Prepares edit, security and control specifications.

Specifies the implementation plan.

Prepares a logical design walk through of the information flow, output, input, controls and implementation plan.

Reviews benefits, costs, target dates and system constraints.

PHYSICAL DESIGN:

Physical system produces the working systems by define the design specifications that tell the programmers exactly what the candidate system must do. It includes the following steps.

Design the physical system.

Specify input and output media.

Design the database and specify backup procedures.

Design physical information flow through the system and a physical design Walk through.

Plan system implementation.

Prepare a conversion schedule and target date.

Determine training procedures, courses and timetable.

Devise a test and implementation plan and specify any new hardware/software.

Update benefits , costs , conversion date and system constraints

Design/Specification activities:

Concept formulation.

Problem understanding.

High level requirements proposals.

Feasibility study.

Requirements engineering.

Architectural design.

MODULE DESIGN

ADMIN

The Administrator logs in using the admin login. In this module two operations are done. During login the Login and Password is verified with that in the database

INPUT DESIGN

The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

o What data should be given as input

o How the data should be arranged or coded

o The dialog to guide the operating personnel in providing input.

o Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in a maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUTDESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should :

Identify the specific output that is needed to meet the requirements.

Select methods for presenting information.

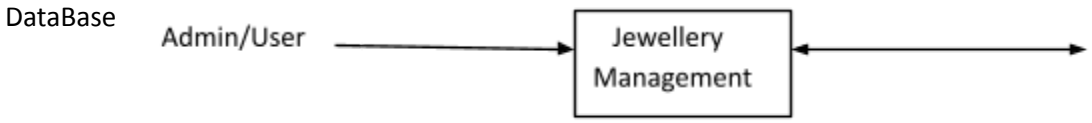
Create document, report, or other formats that contain information produced by the system.

Analysis is the process of extracting the needs of the system and what the system must do to satisfy user's requirements. Object Oriented Analysis is made to develop a series of solution models that describes computer software, which works to satisfy the users. The goal of Object Oriented Analysis is first to understand the domain of the problem and the system responsibilities by understanding hoe the users use or will use the system. This is accomplished by constructing several models of the system. OOA process consists of the following steps:

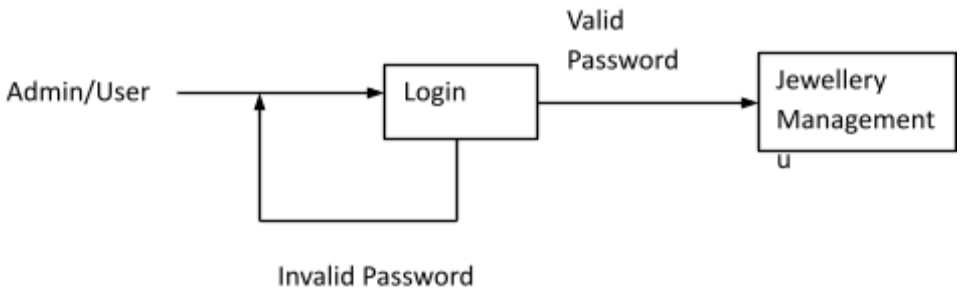
1. Identify the actors.
2. Develop a simple business process model using UML activity diagram.
3. Develop a Use Case.
4. Develop interaction diagrams
5. Identify classes.

DFD Diagram:

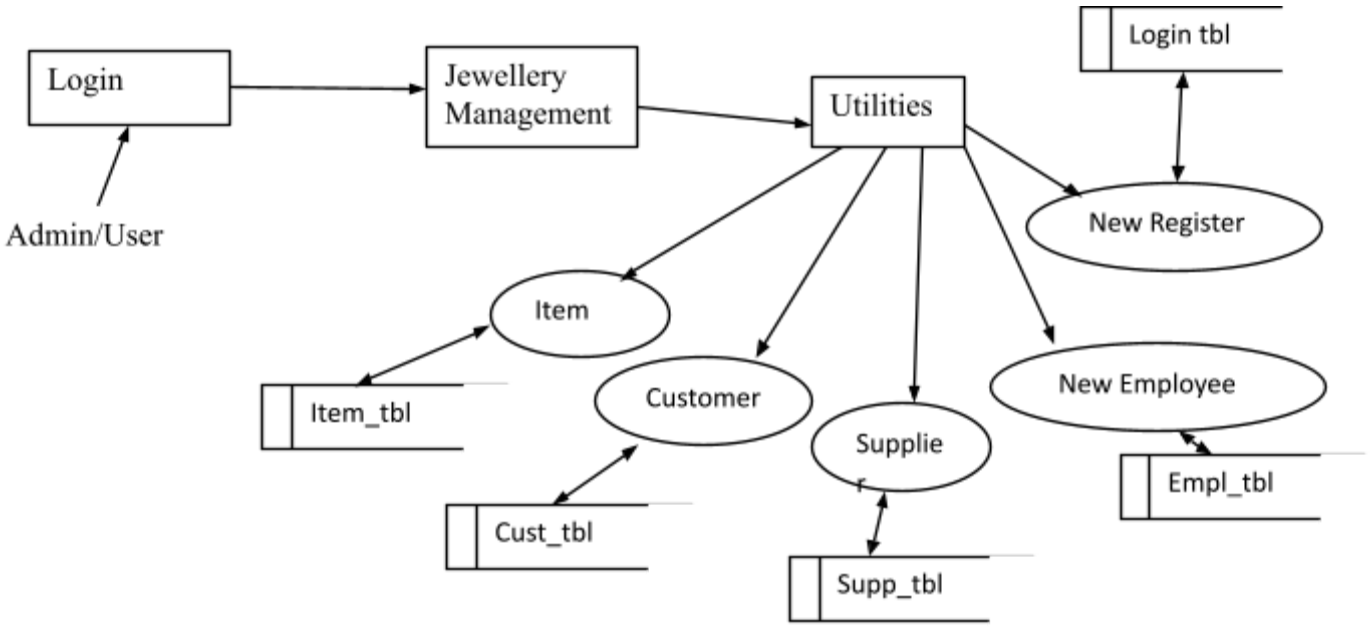
DFD Level 0:



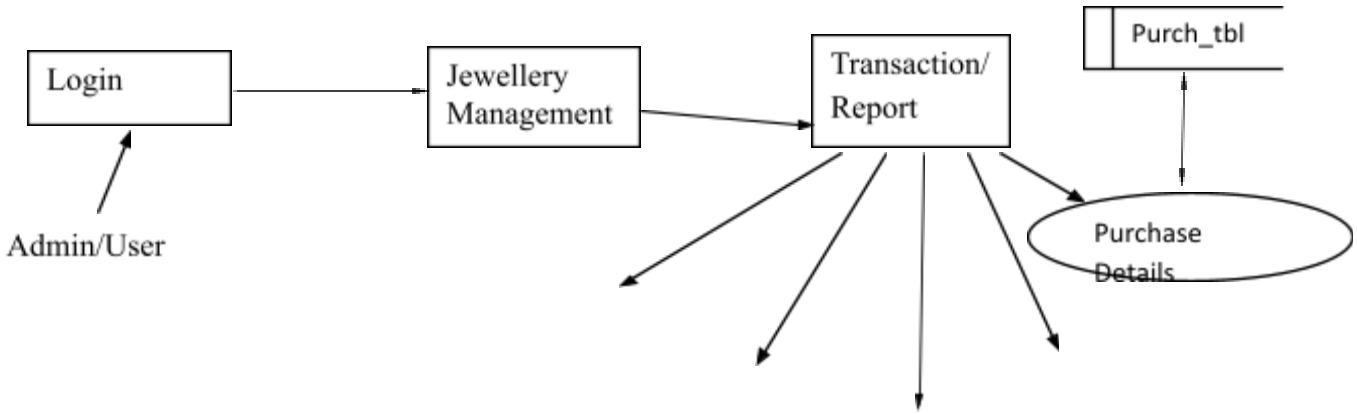
DFD Level 1:

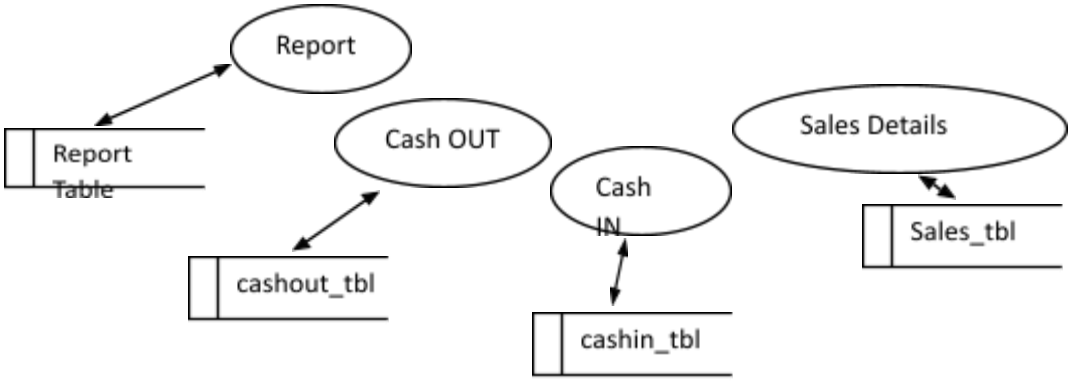


DFD Level 2:

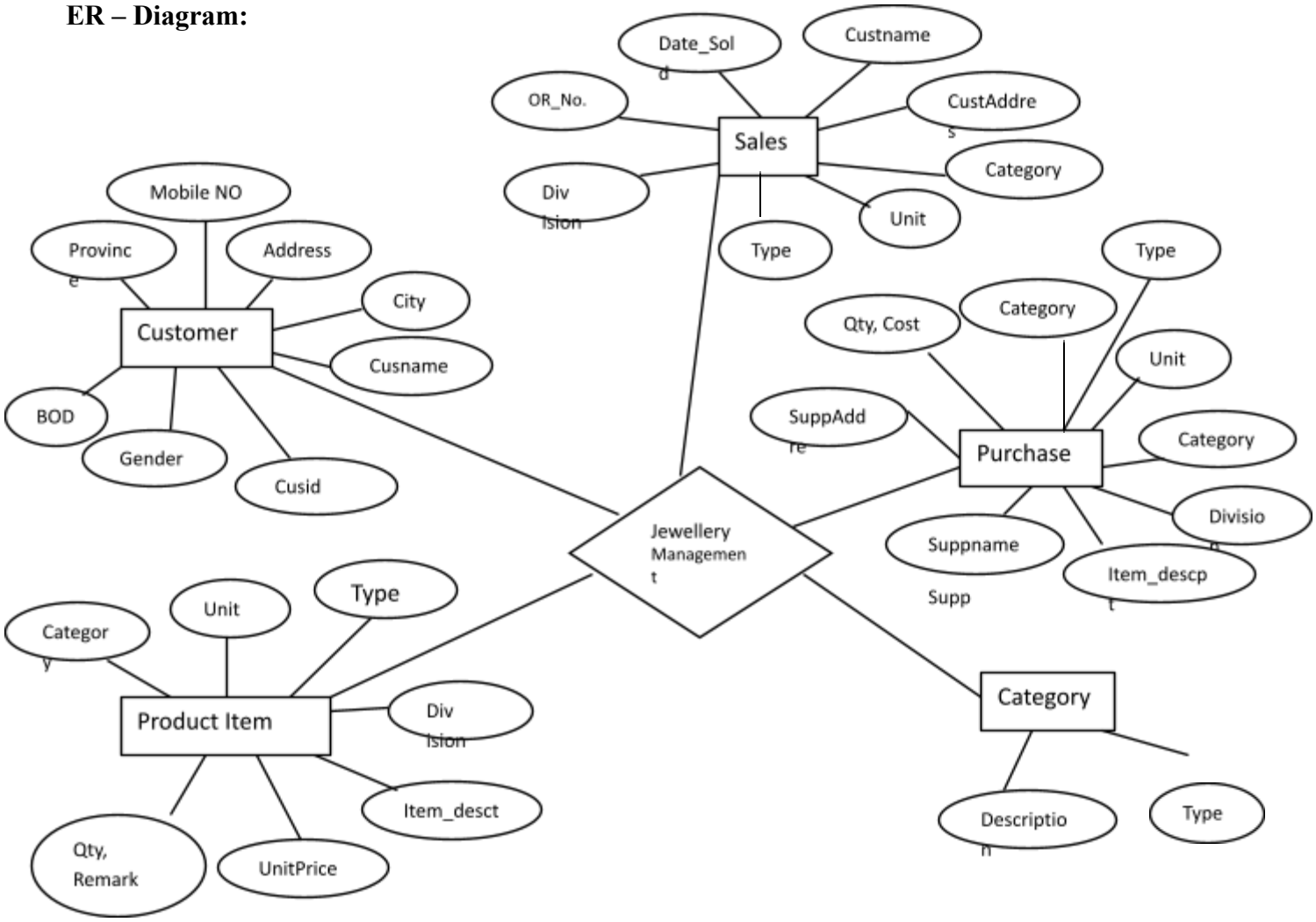


DFD Level 3:




Report
Report Table
Cash OUT
cashout_tbl
Cash IN
cashin_tbl
Sales Details
Sales_tbl

ER – Diagram:


Date_Sold
Custname
OR_No.
CustAddress
Sales
Category
Division
Unit
Type
Mobile NO
Province
Address
City
Customer
Cusname
BOD
Gender
Cusid
Type
Qty, Cost
Category
Unit
SuppAddre
Purchase
Category
Division
Suppname
Item_descpt
Supp
Jewellery Management
Type
Unit
Category
Product Item
Division
Item_desct
Qty, Remark
UnitPrice
Category
Description
Type

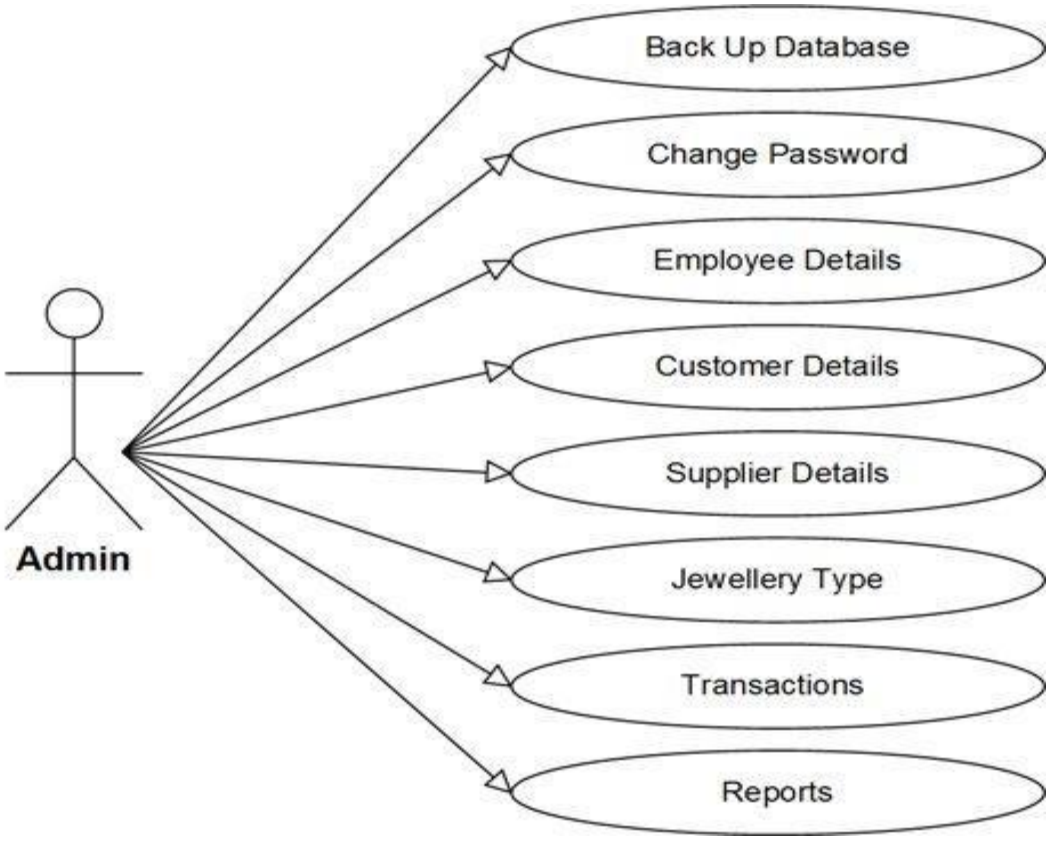
4.1 USE CASES AND USE CASE DIAGRAMS:

A use-case is a UML modeling element that describes how a user of the proposed system will interact with the system to perform a discrete unit of work. It describes and signifies a single interact with the system to perform a discrete unit of work. It describes and signifies a single interaction over time that has meaning for the end user (person, machine or other system), and is required to leave the system in a complete state: either the interaction completed or was rolled back to the initial state.

- A use case typically has requirements and constraints that describe the essential features and rules under which it operates.
- A use case may have an associated Sequence diagram illustrating behavior over time – who does what and to whom, when.

- A use case typically has scenarios associated with it that describe the work flow over time that produces the end result. Alternate work flows (to capture exceptions, etc.) are also allowed.

A use case diagram captures use cases and actor interactions. It describes the functional requirements of the system, the manner that outside things (actors) interact at the system boundary and the response of the system.



4.2 SEQUENCE DIAGRAM:

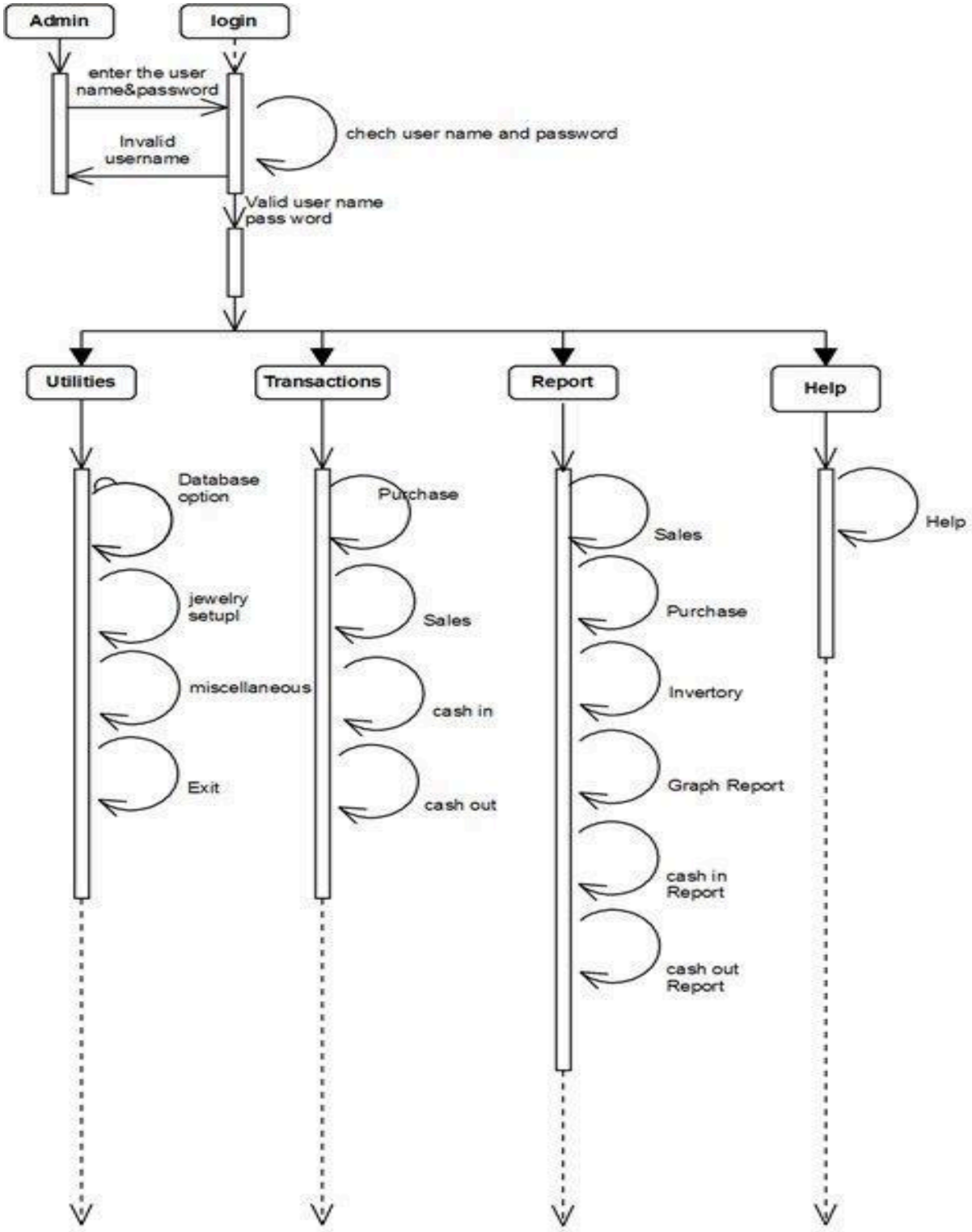
A sequence diagram is a structures representation of behavior as a series of sequential steps over time. It is used to depict work flow, message passing and how elements in general cooperate over time to achieve results.

A sequence diagram models a dynamic view of the interactions between model elements at runtime. Sequence diagrams are commonly used as explanatory models for use case scenarios.

- Each sequence element is arranged in a horizontal sequence, with messages passing back and forward between elements.

- An actor element may be used to represent the user initiating the flow of events.
- Stereotyped elements, such as boundary, control and entity, may be used to illustrate screens, controllers and database items respectively.
- Each element has a dashed stem called a lifeline, where that element that exists and potentially takes part in the illustrations.

Sequence diagram in UML, indicates how events cause transitions from object to object. Once examining a use-case has identified events, the modeler creates a sequence diagram- a representation of how events cause flow from one object to another as a function of time. The sequence diagram is a shorthand version of the use-case. It represents key classes and the events that cause behavior to flow from class to class.


Admin
login
enter the user name&password
Invalid username
chech user name and password
Valid user name pass word
Utilities
Transactions
Report
Help
Database option
jewelry setupl
miscellaneous
Exit
Purchase
Sales
cash in
cash out
Sales
Purchase
Invertory
Graph Report
cash in Report
cash out Report
Help

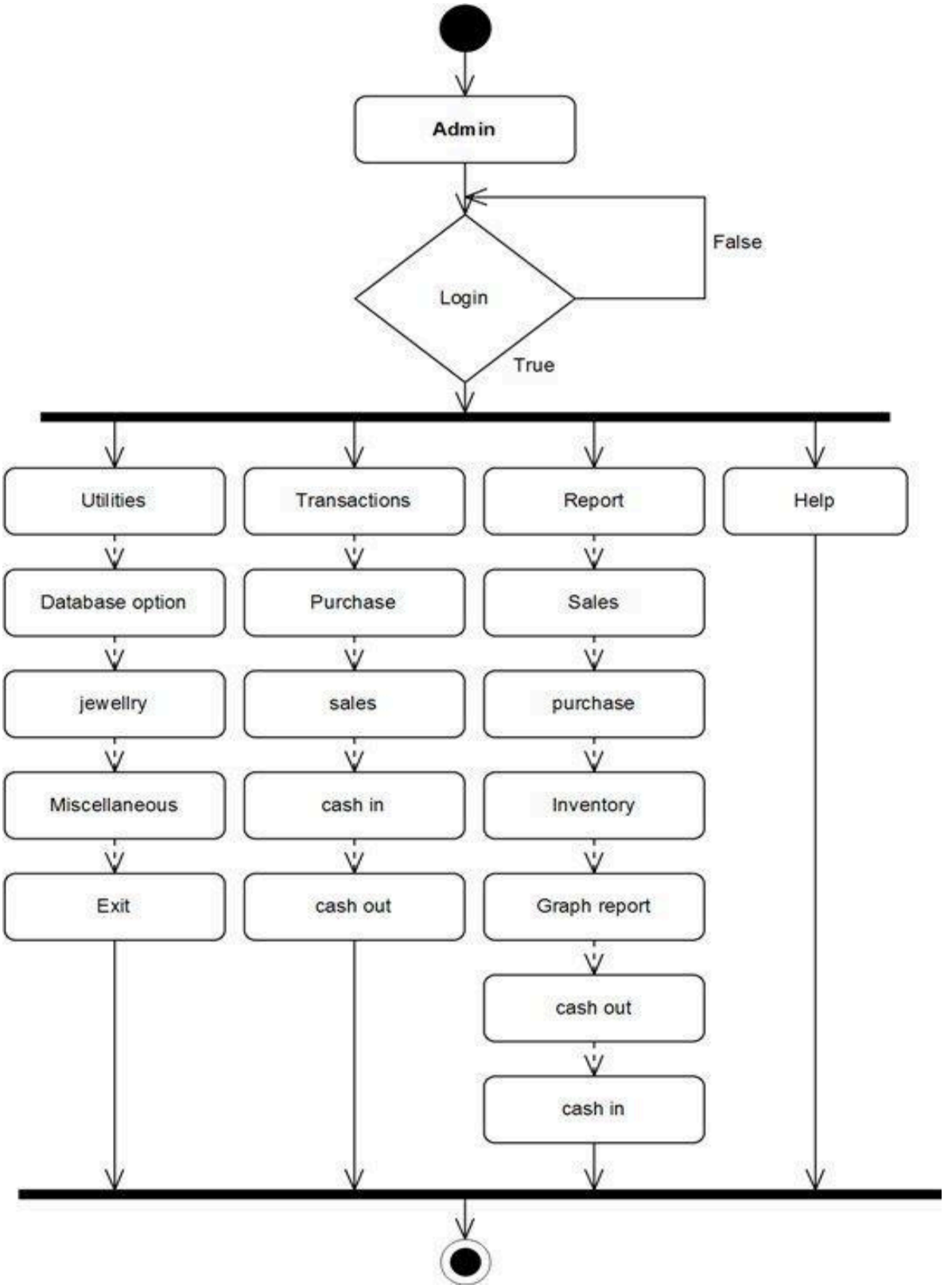
4.3 ACTIVITY DIAGRAM:

The UML activity diagram is used to indicate the flow of the interaction within a specific scenario by means of graphical representation. An activity diagram uses rounded rectangles to imply a specific system function, arrow to represent flow through the system, decision diamonds to deflect a branching decision and solid horizontal lines to indicate that parallel activities are occurring.

Some of the elements of Activity diagrams as follows

- **Activity:** an activity organizes and specifies the participation of subordinate behaviors, such as sub-activities or actions, to reflect the control and data flow of a process.
- **Initial Node:** The initial element is used by the Activity and State Machine diagrams. In Activity diagrams, it defines the start of flow when an activity is invoked.
- **Final Node:** There are two nodes used to define a final state in an activity. The final element indicates the completion of an activity diagram is aborted. The other type of final node, flow final, depicts an exit from the system but has no effect on other executing flows in the activity.
- **Decision Nodes:** A decision is an element of an activity diagram that indicates a point of conditional progression: if a condition is true, then processing continues one way, if not, then another. This can also be used as a merge node in that multiple alternate flows can be merged (but not synchronized) to form one flow. The following are examples of these modes of decision element.
- **Fork:** The fork/join elements have different modes of use. They are as follows:
    1. To fork or split the flow into number of concurrent flows.
    2. To join the flow of number of concurrent flows.

3. To both join and fork a number of incoming flows to a number of outgoing flows.


Admin
Login
False
True
Utilities
Transactions
Report
Help
Database option
Purchase
Sales
jewellry
sales
purchase
Miscellaneous
cash in
Inventory
Exit
cash out
Graph report
cash out
cash in

5.4 DATA DESIGN:

Data design creates a model of data and information that is represented at a high level of abstraction. The structure of data has always been an important part of software design. Data design plays a vital role at the program component level, application level and business level.

In program component level, the design of data structures and algorithms are manipulated. At the application level, the data model is translated into a database and at the business level, the collection of information in the database are recognized into "data warehouse".

In this project, "Jewellery management system", the tables and database creation are carried out in designing model phase according to the data elements. All the tables that have been used in Estate Builder are described in Appendix-A.

CHAPTER 6

SYSTEM IMPLEMENTATION

Implementation includes all those activities that take place to convert from the old system to the new. The old system consists of manual operations, which is operated in a very different manner from the proposed new system. A proper implementation is essential to provide a reliable system to meet the requirements of the organizations. An improper installation may affect the success of the computerized system.

IMPLEMENTATION METHODS:

There are several methods for handling the implementation and the consequent conversion from the old to the new computerized system.

The most secure method for conversion from the old system to the new system is to run the old and new system in parallel. In this approach, a person may operate in the manual older processing system as well as start operating the new computerized system. This method offers high security, because even if there is a flaw in the computerized system, we can depend upon the manual system. However, the cost for maintaining two systems in parallel is very high. This outweighs its benefits.

Another commonly method is a direct cut over from the existing manual system to the computerized system. The change may be with in a week or with in a day. There are no parallel activities. However, there is no remedy in case of a problem. This strategy requires careful planning.

A working version of the system can also be implemented in one part of the organization and the personnel will be piloting the system and changes can be made as and when required. But this method is less preferable due to the loss of entirety of the system.

IMPLEMENTATION PLAN:

The implementation plan includes a description of all the activities that must occur to implement the new system and to put it into operation. It identifies the personnel responsible for the activities and prepares a time chart for implementing the system. The implementation plan consists of the following steps.

- List all files required for implementation.
- Identify all data required to build new files during the implementation.
- List all new documents and procedures that go into the new system.

The implementation plan should anticipate possible problems and must be able to deal with them. The usual problems may be missing documents; mixed data formats between current and files, errors in data translation, missing data etc.

POST IMPLEMENTATION REVIEW:

After the system is implemented, a review should be conducted to determine whether the system is meeting expectations and where improvements are needed. System quality, user confidence and operating systems statistics are accessed through such technique event logging, impact evaluation and attitude surveys. The review not only assesses how well the proposed system is designed and implemented, but also is a valuable source of information that can be applied to a critical evaluation of the system.

The reviews are conducted by the operating personals as well as the software developers in order to determine how well the system is working, how it has been accepted and whether adjustments are needed. The review of the system is highly essential to determine the future enhancements required by the system. The system can be considered successful only if information system has met it objectives. The review analyses the opinion of the employees and identifies the attitudes towards the new computerized system. Only when the merits and demerits of the implemented system are known, one can determine what all additional features it requires are. The following are the issues to be considered in the evaluation of the system.

- The change in the cost of operation after the installation of the computerized system.
- The basic change that has been effected after the introduction of the system.
- The improvement in the accuracy of the computations.
- The acceptance of the new system by the staff and the convenience it brought to them.
- The change in the effectiveness caused by the implementation of the new system.

A study of the system has revealed that the employees due to the user friendliness have accepted the system, reduced the number of errors, increased accuracy and decreased cost of operations. The system also pays for efficient and speedy execution of operations compared to the earlier system.

CHAPTER 7

TESTING

7.1 INTRODUCTION:

Testing is the set of activities that can be planned in advanced and s conducted systematically. Testing requires that the developer discard preconceived notions of the "correctness" of the software just developed and overcome a conflict of interest that occurs when errors are encounterd.Testing principles are

- All tests should be traceable to customer requirements
- Testing should be planned long before the testing begins
- Testing should begin "in the small" and progress towards testing "in the large".
- Exhaustive testing is not possible
- To be most effective, testing should be conducted by an independent third party.

Testing objective are

- Testing is the process of executing a program within the intent of finding an error.
- A good test case is one that has high probability of finding an as-yet-undiscovered error.
- A successful test is one that un covers an as yet-undiscovered error

There are various testing strategies available to accommodate from low-level testing to high-level testing as discussed below.

7.2 TEST PLAN

Testing is the major quality control measure employed during software development. In the project, the first test considered is the unit testing. In this unit testing, each modules of the system are tested separately. This is carried out during programming stage itself. Each module should work satisfactory as regard from the module.

After the entire module are checked independently and completed then the integration testing is performed to check whether there is any interface errors. Then those errors are verified and corrected.

And also the security test is performed to allow only authorized persons to this system. Finally, the validation testing is performed to validate whether the customer requirements are stratified are not.

7.3 UNIT TESTING

The unit testing is carried out on coding. Here different modules are tested against the specifications produced during design for the modules. Unit testing mainly focused first in the smallest and low level modules, proceeding one at a time. Each module was tested against required functionally and test cases were developed to test the boundary values.

Unit testing focuses verification effort on the smallest unit of software design the software component or module. The unit test focuses on the internal processing logic and data structures within the boundary of the component. This type of testing can be conducted in parallel for multiple components.

7.4 INTEGRATION TESTING:

Integration testing is a systematic technique for consulting the software architecture while at the same time conducting test to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design.

7.5 VALIDATION TESTING:

Validation testing is that validation succeeds when software functions in a manner that can be reasonably expected by the user. Validation testing begins after the culmination of integration testing, software is completely assembled as a package; interfacing errors have been uncovered and corrected.

The error detecting during this testing is

- Incorrect Function
- Input Condition Errors
- Database Error
- Performance Error
- Initialization and Interface Error

7.6 SECURITY TESTING:

Security testing verifies that protection mechanisms built into a system will, in fact, protect it from improper penetration. The system security must, of course, be tested for invulnerability from flank or rear attack.

Test case

- The system provides authentication by means of validating the username and password. It won't allow the user gives the exact password and username.

CHAPTER 8

Future Enhancement

This system is developed such a way that additional enhancement can be done without much difficulty. The renovation of the project would increase the flexibility of the system. Also the features are provided in such a way that the system can also be made better and efficient functionality The programs were coded in an easier and more structured manner so that may further modifications may be incorporated easily. The processing time in this system is very lesser compared to existing system. This system has good flexibility of accommodating any more changes that might arise in the future also. In this system, data integrity is maintained and data redundancy is avoided and it increase system efficiency. The database is designed in such a way that it will be also helpful for enhancement of the system

CHAPTER 9

Conclusion

The system “Jewellary Management system” deals with purchase and sales processing of a Jewellary shop. This system has been developed to satisfy all the proposed requirements. The process of recording details about supplier, item, Billing andcustomers is more simple and easy. The system reduces the possibility of errors to a greatextent and maintains the data in an efficient manner. User friendliness is the uniquefeature of this system. The system generates the reports as and when required. Thesystem is highly interactive and flexible for further enhancement.The

coding is done in a simplified and easy to understandable manner so thatother team trying to enhance the project can do so without facing much difficulty. Thedocumentation will also assist in the process as it has also been carried out in a simplifiedand concise way.

BIBLIOGRAPHY

VISUAL BASIC 6.0

1. Visual Basic 6.0 from the Ground up, Gary Cornell, Tata McGraw Hill Edition
2. The Complete Reference Visual Basic 6.0 , Noel Jerke, Tata Mcgraw HillEdition
3. Visual Basic 6.0 Programming Black Book, Sten Holzner, Dream tech Press,Dreamtech Press, New Delhi-2004

4. MS-Office 2000, Michael Busy and Rossell a Stultz, BPB Publication, NewDelhi.
5. Micro soft Office Access Bible by Groh
6. Software Engineering a practioner's approach- Roger S. Pressman, TataMcGraw Hill Edition
7. Working with Access by RON Mansfield, Tata McGraw Hill Publication

APPENDIX – A

TABLES

DATABASE DESIGN

TblAdmin

Column Name	Data Type	Allow Nulls
Admin_ID	int	☐
Login_ID	varchar(50)	☑
Password	varchar(50)	☑
First_Name	varchar(50)	☑
Last_Name	varchar(50)	☑
Address	varchar(100)	☑
Company_Name	varchar(50)	☑
Company_Address	varchar(100)	☑
Contact_No	varchar(20)	☑
Fax_No	varchar(30)	☑
AccountCreated_Date	datetime	☑
LastLogin_Date	datetime	☑

TblCategory

Column Name	Data Type	Allow Nulls
Category1_ID	int	☐
Category_Name	varchar(100)	☑
		☐

Column Name	Data Type	Allow Nulls
Category2_ID	int	☐
Category_Name	varchar(100)	☑
Category1_ID	int	☑
Final_Product	bit	☑
Image	varchar(MAX)	☑
		☐

Column Name	Data Type	Allow Nulls
Category3_ID	int	☐
Category_Name	varchar(100)	☑
Category2_ID	int	☑
Final_Product	bit	☑
Image	varchar(MAX)	☑
		☐

Customer Information

Column Name	Data Type	Allow Nulls
Customer_ID	int	☐
First_Name	varchar(30)	☑
Last_Name	varchar(30)	☑
Occupation	varchar(50)	☑
Street	varchar(50)	☑
City	varchar(50)	☑
State_ID	varchar(50)	☑
Country_ID	varchar(50)	☑
Pin_Code	varchar(10)	☑
Contact_No	varchar(15)	☑
Mobile_No	varchar(15)	☑
Email_ID	varchar(40)	☑
User_Name	varchar(20)	☑
Password	varchar(15)	☑
Date	datetime	☑

Product Item

Column Name	Data Type	Allow Nulls
Product_ID	int	☐
Product_Code	varchar(30)	☑
Product_Name	varchar(60)	☑
Category1_ID	int	☑
Category2_ID	int	☑
Category3_ID	int	☑
Category4_ID	int	☑
Image	varchar(MAX)	☑
Price	money	☑
Weight	float	☑
Weight_Unit	varchar(50)	☑
Short_Desc	varchar(250)	☑
Long_Desc	varchar(MAX)	☑
Special	bit	☑
DisplayOrderGroup	int	☑
Home_Option	bit	☑

Purchase table

Field Name	Data Type
Category	Text
Unit	Text
Type	Text
Division	Text
Item_Description	Text
Qty	Text
Cost	Text
Supplier_Name	Text
Supplier_Address	Text
Date_Purchased	Date/Time
Amount	Text
Purchase_Code	Text

TblOrder

Column Name	Data Type	Allow Nulls
Order_ID	int	☐
Order_Code	varchar(20)	☑
Customer_ID	int	☑
Processed	bit	☑
Cancelled	bit	☑
Order_Date	varchar(30)	☑
Processed_Date	varchar(30)	☑
Order_Total	money	☑
No_Of_Products	int	☑
Remark	varchar(300)	☑

SOURCE CODE

Customer Form

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class Admin_Customers : System.Web.UI.Page
{
    clsCustomers objCustomer = new clsCustomers();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
```

```
        {
            loadGridView();
        }

    }
    protected void btnGo_Click(object sender, EventArgs e)
    {
        if (txtcustSearch.Text != "")
        {
            Session["search"] = 1;
            DataSet ds = new DataSet();
            ds=objCustomer.SearchByName(txtcustSearch.Text);
            GridView1.DataSource = ds;
            GridView1.DataBind();
        }
        else
        {
            loadGridView();
        }
    }
    protected void GridView1_SelectedIndexChanging(object sender,
GridViewSelectEventArgs e)
    {

    }
    protected void GridView1_PageIndexChanging(object sender,
GridViewPageEventArgs e)
    {
        GridView1.PageIndex=e.NewPageIndex ;
        if (Convert.ToInt32(Session["search"]) == 1)
        {
            DataSet ds = new DataSet();
            ds = objCustomer.SearchByName(txtcustSearch.Text);
            GridView1.DataSource = ds;
            GridView1.DataBind();
        }
        else
        {
            loadGridView();
        }
    }
    public void loadGridView()
    {
        DataSet ds = new DataSet();
        ds = objCustomer.GetAllCustomers();
        GridView1.DataSource = ds;
        GridView1.DataBind();
    }
    protected void GridView1_RowCommand(object sender, GridViewCommandEventArgs
e)
    {
        if (e.CommandName == "History" && e.CommandArgument != null)
        {
            int customerID = Convert.ToInt32(e.CommandArgument);
```

```
            Response.Redirect("CustomerHistory.aspx?ci=" + customerID);
        }
    }

}
```

Order Form

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class Admin_Orders : System.Web.UI.Page
{
    clsOrders objOrder = new clsOrders();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            //Binding data with Grid View
            DataSet ds = new DataSet();
            ds = objOrder.GetAllOrders();
            if (ds.Tables[0].Rows.Count > 0)
            {
                GridView1.DataSource = ds;
                GridView1.DataBind();
                PanelData.Visible = true;
                PanelError.Visible = false;
            }
            else
            {
                PanelData.Visible =false;
                PanelError.Visible = true;
            }
            //Filling the drop down list for date
            for (int i = 1; i <= 31; i++)
            {
                ddlstartDay.Items.Add(i.ToString());
                ddlendDay.Items.Add(i.ToString());
            }
            for (int j = 2000; j <= System.DateTime.Now.Year; j++)
            {
                ddlstartYear.Items.Add(j.ToString());
                ddlendYear.Items.Add(j.ToString());
            }
        }
```

```
    }
    protected void btnGo_Click(object sender, EventArgs e)
    {

    }
    protected void btnshowall_Click(object sender, EventArgs e)
    {
        DataSet ds = new DataSet();
        ds = objOrder.GetAllOrders();
        if (ds.Tables[0].Rows.Count > 0)
        {
            GridView1.DataSource = ds;
            GridView1.DataBind();
            PanelData.Visible = true;
            PanelError.Visible = false;
        }
        else
        {
            PanelData.Visible = false;
            PanelError.Visible = true;
        }

    }
    protected void btnsubmitdate_Click(object sender, EventArgs e)
    {
        if (ddlstartDay.SelectedItem.Text != "Day")
        {
            if (ddlstartMonth.SelectedItem.Text != "Month")
            {
                if (ddlstartYear.SelectedItem.Text != "Year")
                {
                    if (ddlendDay.SelectedItem.Text != "Day")
                    {
                        if (ddlendMonth.SelectedItem.Text != "Month")
                        {
                            if (ddlstartDay.SelectedItem.Text != "Year")
                            {
                                DateTime startDate, endDate;
                                string stringStartDate, stringEndDate;
                                stringStartDate =
ddlstartMonth.SelectedItem.Text + "/" + ddlstartDay.SelectedItem.Text + "/" +
ddlstartYear.SelectedItem.Text;
                                stringEndDate = ddlendMonth.SelectedItem.Text +
"/" + ddlendDay.SelectedItem.Text + "/" + ddlendYear.SelectedItem.Text;
                                startDate =
Convert.ToDateTime(stringStartDate);
                                endDate = Convert.ToDateTime(stringEndDate);
                                DataSet ds = new DataSet();
                                ds = objOrder.GetOrderBetweenDate(startDate,
endDate);
                                if (ds.Tables[0].Rows.Count > 0)
                                {
                                    GridView1.DataSource = ds;
                                    GridView1.DataBind();
```

```
                                PanelData.Visible = true;
                                PanelError.Visible = false;
                            }
                            else
                            {

                                PanelData.Visible = false;
                                PanelError.Visible = true;
                            }

                        }
                        else
                        {
                            string str1 = "<script
language='javascript'>alert('Select proper date');</script>";
                            Page.RegisterStartupScript("PopUp", str1);
                        }
                    }
                    else
                    {
                        string str2 = "<script
language='javascript'>alert('Select proper date');</script>";
                        Page.RegisterStartupScript("PopUp", str2);
                    }
                }
                else
                {
                    string str3 = "<script
language='javascript'>alert('Select proper date');</script>";
                    Page.RegisterStartupScript("PopUp", str3);
                }
            }
            else
            {
                string str4 = "<script language='javascript'>alert('Select
proper date');</script>";
                Page.RegisterStartupScript("PopUp", str4);
            }
        }
        else
        {
            string str5 = "<script language='javascript'>alert('Select
proper date');</script>";
            Page.RegisterStartupScript("PopUp", str5);
        }
    }
    else
    {
        string str6 = "<script language='javascript'>alert('Select proper
date');</script>";
        Page.RegisterStartupScript("PopUp", str6);

    }
}
```

```
    protected void GridView1_RowCommand(object sender, GridViewCommandEventArgs
e)
    {
        int orderId=Convert.ToInt32(e.CommandArgument);
        if (e.CommandName == "Customer" && e.CommandArgument != null)
        {
            Response.Redirect("OrderCustomerDetails.aspx?oi=" + orderId);
            Session["redirect"] = 1;


        }
        if (e.CommandName == "Product" && e.CommandArgument != null)
        {
            Response.Redirect("OrderProductDetails.aspx?oi=" + orderId);
            Session["redirect"] = 1;
        }
    }
    protected void GridView1_PageIndexChanging(object sender,
GridViewPageEventArgs e)
    {
        GridView1.PageIndex = e.NewPageIndex;
        DataSet ds = new DataSet();
        ds = objOrder.GetAllOrders();
        GridView1.DataSource = ds;
        GridView1.DataBind();


    }
    protected void GridView1_SelectedIndexChanging(object sender,
GridViewSelectEventArgs e)
    {

    }
}
```

Product Form

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class Admin_Products : System.Web.UI.Page
{
    clsProducts objProduct = new clsProducts();
    clsCategories objCategory = new clsCategories();
```

```
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            DataSet dsProducts = new DataSet();
            dsProducts = objProduct.GetAllProducts();
            if (dsProducts.Tables[0].Rows.Count >= 1)
            {
                GridView1.DataSource = dsProducts;
                GridView1.DataBind();
                PanelError.Visible = false;
                PanelProducts.Visible = true;
            }
            else
            {
                PanelProducts.Visible = false;
                PanelError.Visible = true;
            }

            DataSet dsRootCategory = new DataSet();
            dsRootCategory = objCategory.getRootCategories();
            ddlrootcategory.Items.Clear();
            ddlrootcategory.Items.Add("Select");
            ddlrootcategory.DataSource = dsRootCategory;
            ddlrootcategory.DataTextField = "Category_Name";
            ddlrootcategory.DataValueField = "Category1_ID";
            ddlrootcategory.DataBind();

        }
    }
    protected void lnkAddProduct_Click(object sender, EventArgs e)
    {
        Response.Redirect("ProductAE.aspx");
    }
    protected void txtsearch_TextChanged(object sender, EventArgs e)
    {

    }

    protected void btnGo_Click(object sender, ImageClickEventArgs e)
    {
        DataSet ds = new DataSet();
        ds = objProduct.SearchByName(txtsearch.Text);
        GridView1.Visible = false;
        GridView2.Visible = true;
        GridView2.DataSource = ds;
        GridView2.DataBind();
    }
    protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
    {

    }
    protected void GridView1_RowCommand(object sender, GridViewCommandEventArgs
e)
```

```
    {
        int productID = Convert.ToInt32(e.CommandArgument);
        //To remove a particular product from database
        if (e.CommandName == "Remove" && e.CommandArgument!=null)
        {
            objProduct.RemoveProduct(productID);
            DataSet ds = new DataSet();
            ds = gridViewDataSource();
            GridView1.DataSource = ds;
            GridView1.DataBind();
            string str = "<script language='javascript'>alert('Product was
removed from the database.');</script>";
            Page.RegisterStartupScript("PopUp", str);

        }

        //To update the details of a particular product
        if (e.CommandName == "Edit" && e.CommandArgument != null)
        {
            Response.Redirect("ProductAE.aspx?a="+ productID);
        }
    }
    protected void ddlrootcategory_SelectedIndexChanged(object sender,
EventArgs e)
    {
        if (ddlrootcategory.SelectedItem.Text != "Select")
        {
            int RootCategoryID =
Convert.ToInt32(ddlrootcategory.SelectedValue);

            //To fill ddlCategory
            DataSet dsCategory = new DataSet();
            dsCategory = objCategory.getCategories(RootCategoryID);
            ddlcategory.Items.Clear();
            ddlcategory.Items.Add("Select");
            if (dsCategory.Tables[0].Rows.Count > 0)
            {
                //lblCat1.Visible = true;
                //ddlcategory.Visible = true;
                //img1.Visible = true;
                ddlcategory.DataSource = dsCategory;
                ddlcategory.DataTextField = "Category_Name";
                ddlcategory.DataValueField = "Category2_ID";
                ddlcategory.DataBind();
            }
            else
            {
                //img1.Visible = false;
                //lblCat1.Visible = false;
                //ddlcategory.Visible = false;
            }

            //To display Products for a particular root category
            DataSet dsProducts = new DataSet();
```

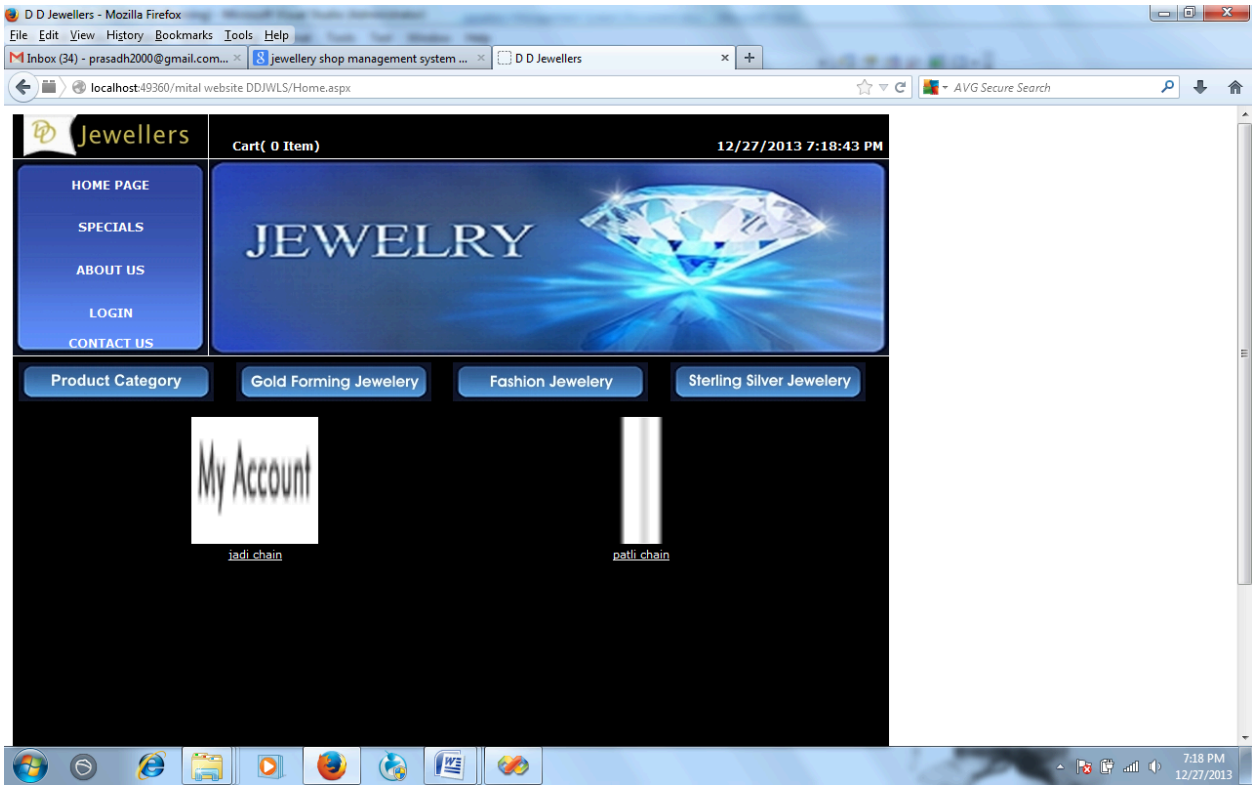
```
            dsProducts = objProduct.GetRootCategoryProducts(RootCategoryID);
            if (dsProducts.Tables[0].Rows.Count >= 1)
            {
                GridView1.Visible = true;
                GridView2.Visible = false;
                GridView1.DataSource = dsProducts;
                GridView1.DataBind();
                PanelProducts.Visible = true;
                PanelError.Visible = false;
            }
            else
            {
                PanelProducts.Visible = false;
                PanelError.Visible = true;
            }
        }
    }
    protected void ddlsubcategory_SelectedIndexChanged(object sender, EventArgs
e)
    {
        if (ddlsubcategory.SelectedItem.Text != "Select")
        {
            int subCategoryID = Convert.ToInt32(ddlsubcategory.SelectedValue);
            //To display Products for a particular sub category
            DataSet dsProducts = new DataSet();

            dsProducts = objProduct.GetSubCategoryProducts(subCategoryID);
            if (dsProducts.Tables[0].Rows.Count >= 1)
            {
                GridView1.Visible = true;
                GridView2.Visible = false;
                GridView1.DataSource = dsProducts;
                GridView1.DataBind();
                PanelError.Visible = false;
                PanelProducts.Visible = true;
            }
            else
            {
                PanelProducts.Visible = false;
                PanelError.Visible = true;
            }
        }

    }
    protected void ddlcategory_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (ddlcategory.SelectedItem.Text != "Select")
        {
            int CategoryID = Convert.ToInt32(ddlcategory.SelectedValue);

            //To fill ddlCategory
            DataSet dssubCategory = new DataSet();
            dssubCategory = objCategory.getSubCategories(CategoryID);
            ddlsubcategory.Items.Clear();
```

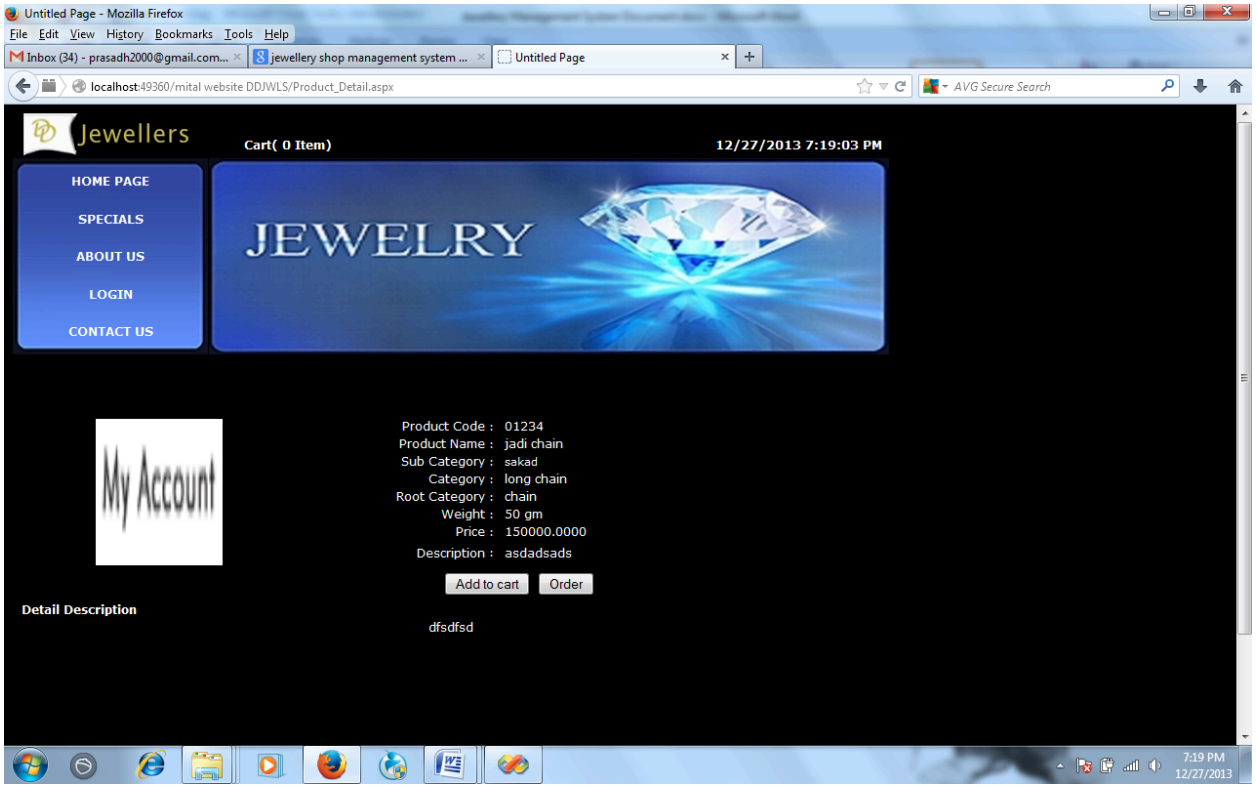
```
            ddlsubcategory.Items.Add("Select");
            if (dssubCategory.Tables[0].Rows.Count >0)
            {
                lblCat2.Visible = true;
                ddlsubcategory.Visible = true;
                img2.Visible = true;
                ddlsubcategory.DataSource = dssubCategory;
                ddlsubcategory.DataTextField = "Category_Name";
                ddlsubcategory.DataValueField = "Category3_ID";
                ddlsubcategory.DataBind();
            }
            else
            {
                img2.Visible = false;
                lblCat2.Visible = false;
                ddlsubcategory.Visible = false;
            }

            //To display Products for a particular category
            DataSet dsProducts = new DataSet();
            dsProducts = objProduct.GetCategoryProducts(CategoryID);
            if (dsProducts.Tables[0].Rows.Count >0)
            {
                GridView1.Visible = true;
                GridView2.Visible = false;
                GridView1.DataSource = dsProducts;
                GridView1.DataBind();
                PanelError.Visible = false;
                PanelProducts.Visible = true;
            }
            else
            {
                PanelProducts.Visible = false;
                PanelError.Visible = true;
            }
        }
    }
    protected void GridView1_PageIndexChanging(object sender,
GridViewPageEventArgs e)
    {

        GridView1.PageIndex = e.NewPageIndex;
        DataSet ds=new DataSet();
        ds=gridViewDataSource();
        GridView1.DataSource = ds;
        GridView1.DataBind();

    }

    public DataSet gridViewDataSource()
    {
        DataSet ds = new DataSet();
        if (ddlrootcategory.SelectedItem.Text != "Select")
        {
```

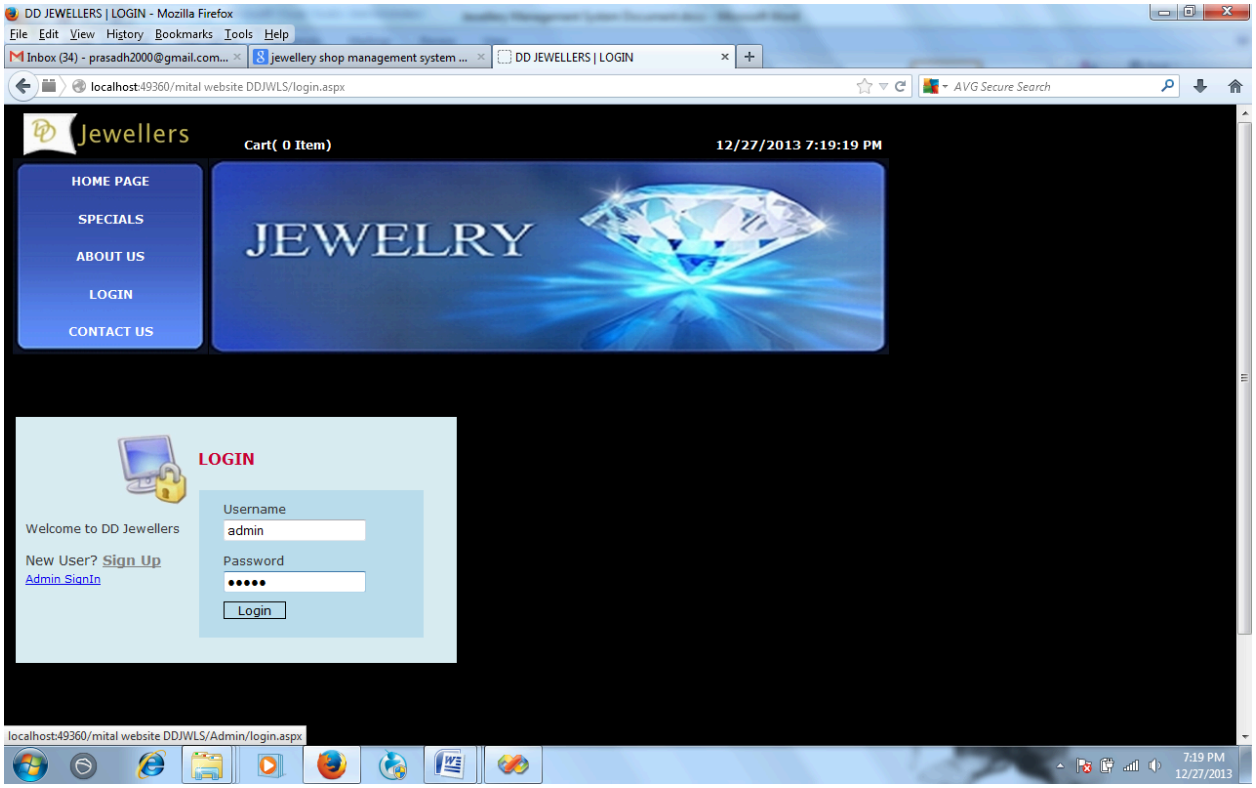
```
            if (ddlcategory.SelectedItem.Text != "Select")
            {
                if (ddlsubcategory.SelectedItem.Text != "Select")
                {
                    ds =
objProduct.GetSubCategoryProducts(Convert.ToInt32(ddlsubcategory.SelectedValue)
);
                }
                else
                {
                    ds =
objProduct.GetCategoryProducts(Convert.ToInt32(ddlcategory.SelectedValue));
                }
            }
            else
            {
                ds =
objProduct.GetRootCategoryProducts(Convert.ToInt32(ddlrootcategory.SelectedValu
e));
            }
        }
        else
        {
            ds = objProduct.GetAllProducts();
        }
        return ds;
    }
    protected void lnkCategory_Click(object sender, EventArgs e)
    {

    }
    protected void GridView2_RowCommand(object sender, GridViewCommandEventArgs
e)
    {
        int productID = Convert.ToInt32(e.CommandArgument);
        //To remove a particular product from database
        if (e.CommandName == "Remove" && e.CommandArgument != null)
        {
            objProduct.RemoveProduct(productID);
            DataSet ds = new DataSet();
            ds = objProduct.SearchByName(txtsearch.Text);
            GridView2.DataSource = ds;
            GridView2.DataBind();
            string str = "<script language='javascript'>alert('Product was
removed from the database.');</script>";
            Page.RegisterStartupScript("PopUp", str);

        }

        //To update the details of a particular product
        if (e.CommandName == "Edit" && e.CommandArgument != null)
        {
            Response.Redirect("ProductAE.aspx?a=" + productID);
        }
```

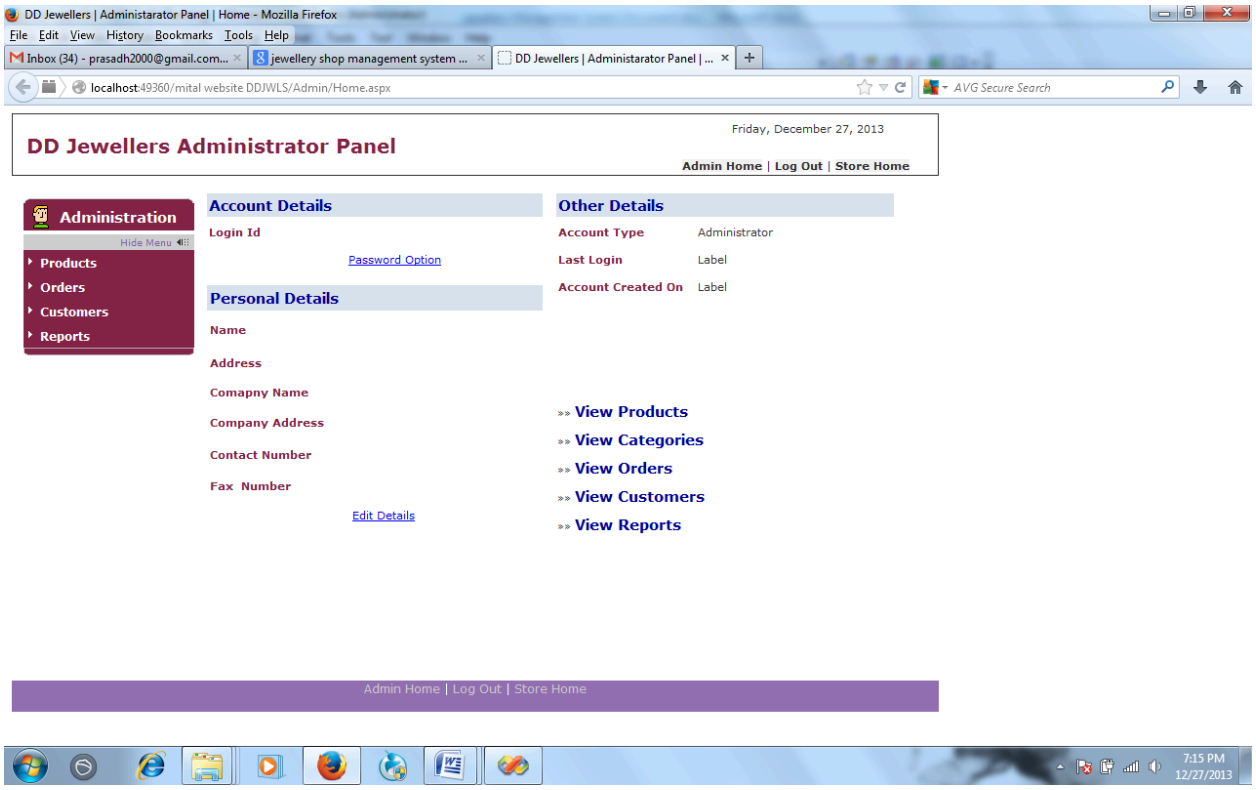
}

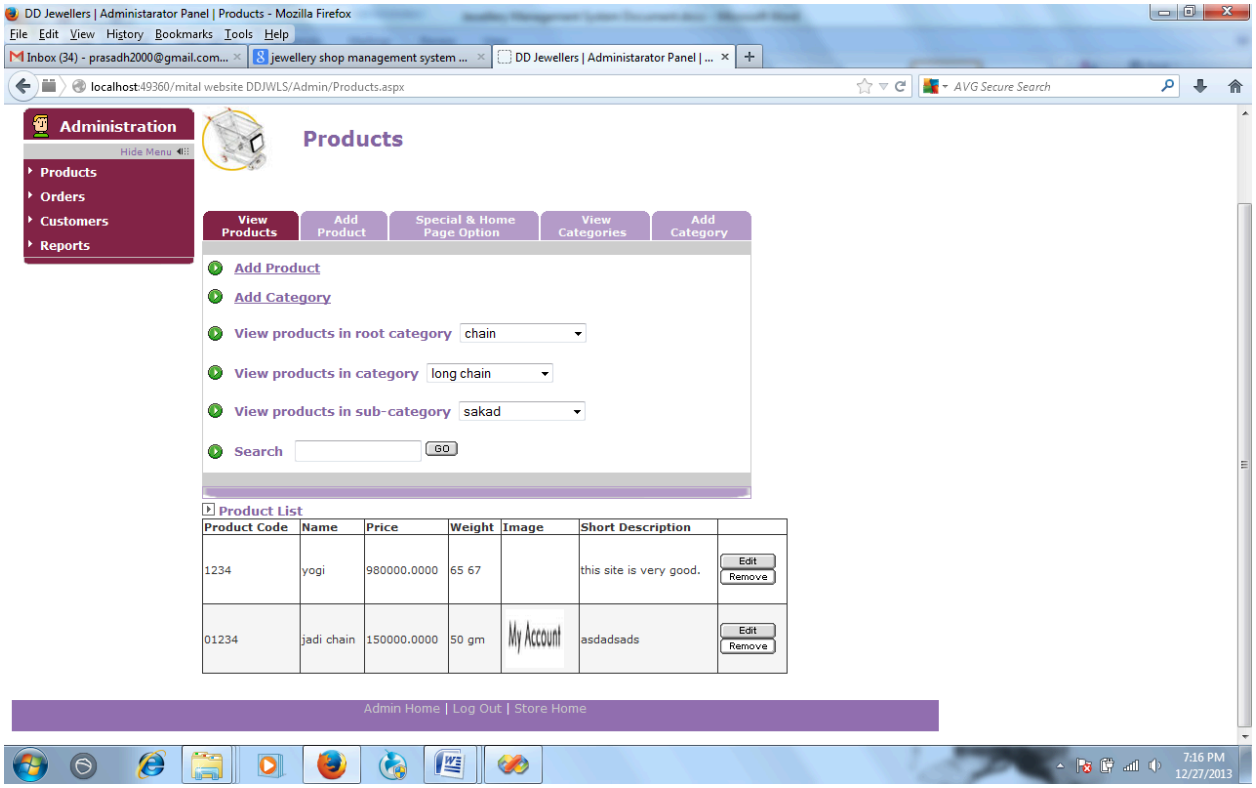
SCREEN SHORT

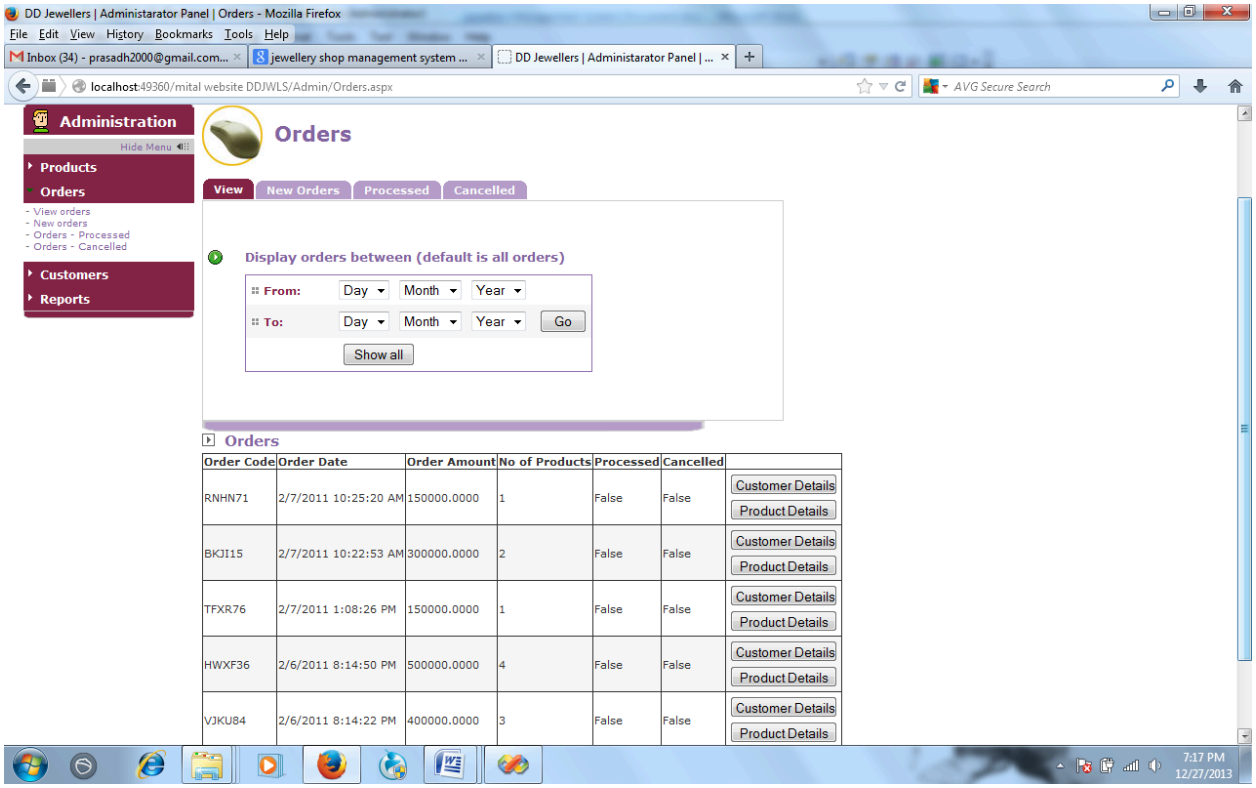


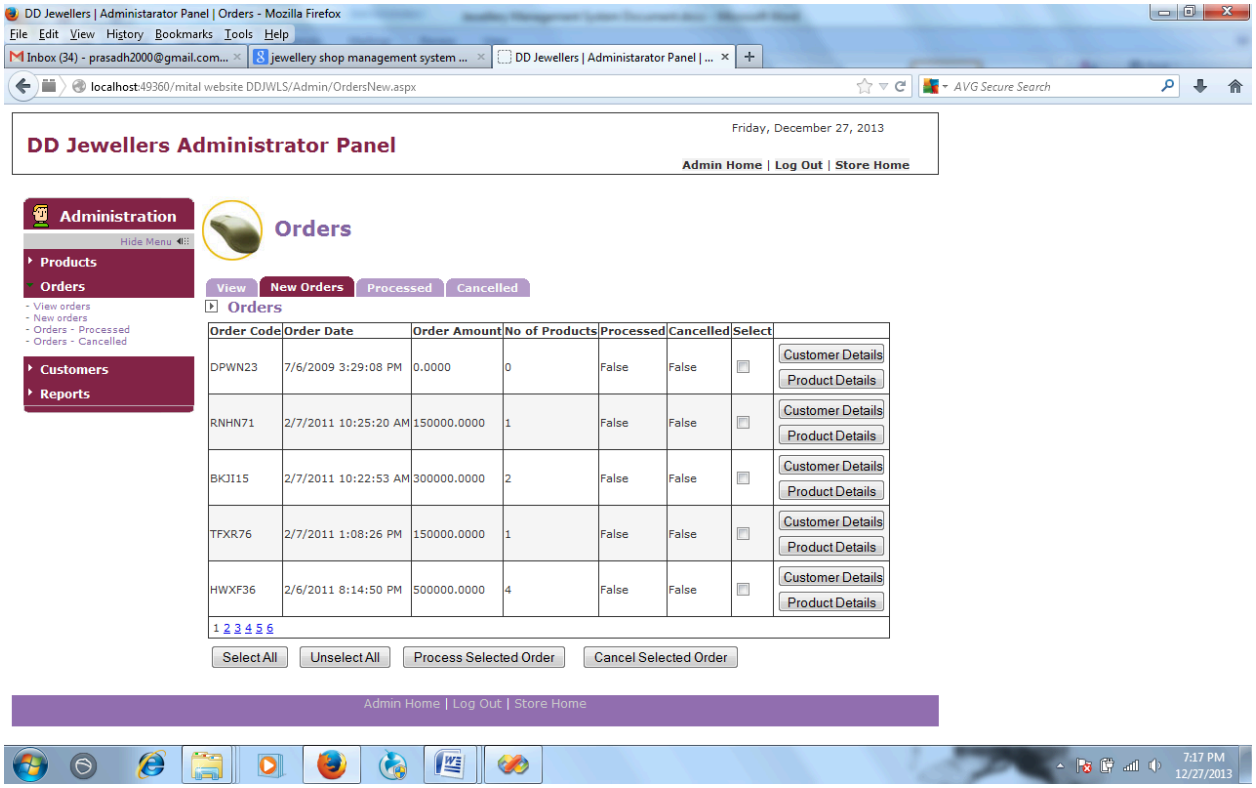

Untitled Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Inbox (34) - prasadh2000@gmail.com...
jewellery shop management system ...
Untitled Page
localhost:49360/mital website DDJWLS/Product_Detail.aspx
AVG Secure Search
Jewellers
Cart(0 Item)
12/27/2013 7:19:03 PM
HOME PAGE
SPECIALS
ABOUT US
LOGIN
CONTACT US
JEWELRY
My Account
Product Code : 01234
Product Name : jadi chain
Sub Category : sakad
Category : long chain
Root Category : chain
Weight : 50 gm
Price : 150000.0000
Description : asdadsads
Add to cart
Order
Detail Description
dfsdfsd
7:19 PM
12/27/2013

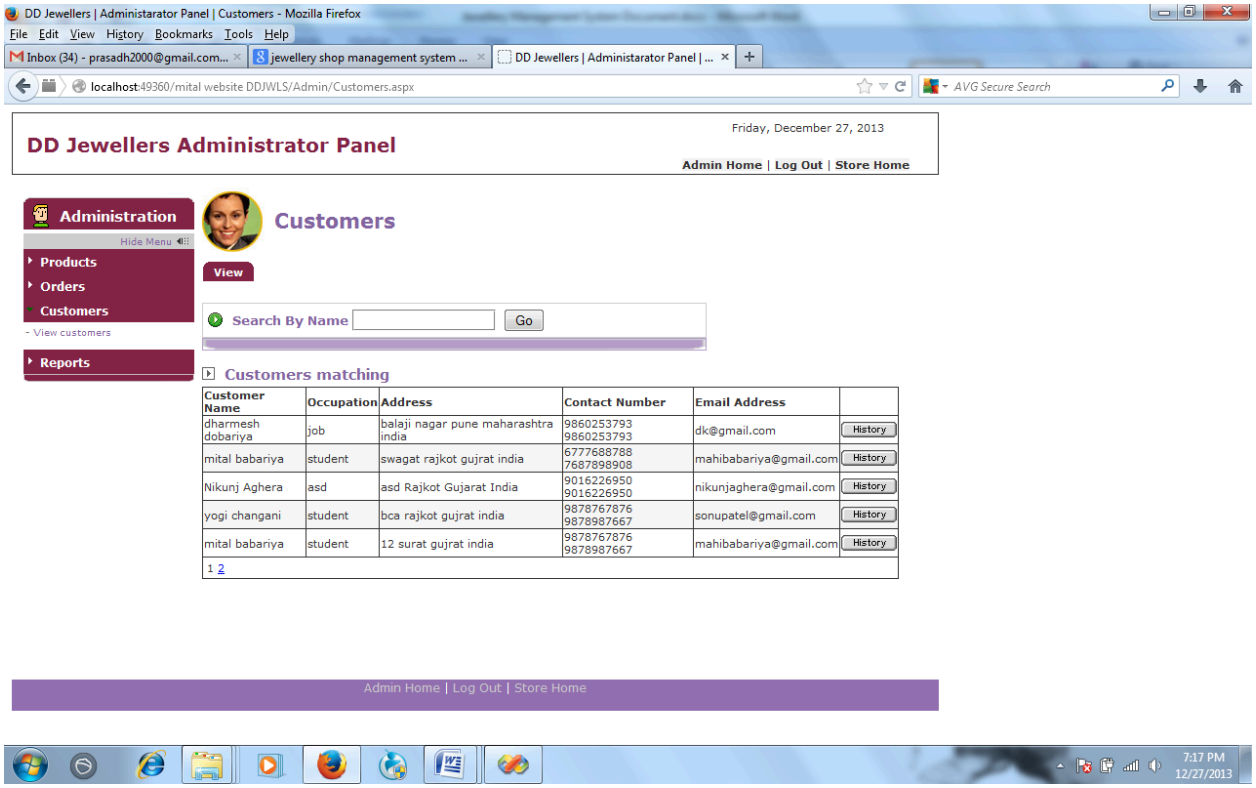

DD JEWELLERS | LOGIN - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Inbox (34) - prasadh2000@gmail.com...
jewellery shop management system ...
DD JEWELLERS | LOGIN
localhost:49360/mital website DDJWLS/login.aspx
AVG Secure Search
Jewellers
Cart(0 Item)
12/27/2013 7:19:19 PM
HOME PAGE
SPECIALS
ABOUT US
LOGIN
CONTACT US
JEWELRY
LOGIN
Welcome to DD Jewellers
New User? Sign Up
Admin SignIn
Username
admin
Password
Login
localhost:49360/mital website DDJWLS/Admin/login.aspx
7:19 PM
12/27/2013

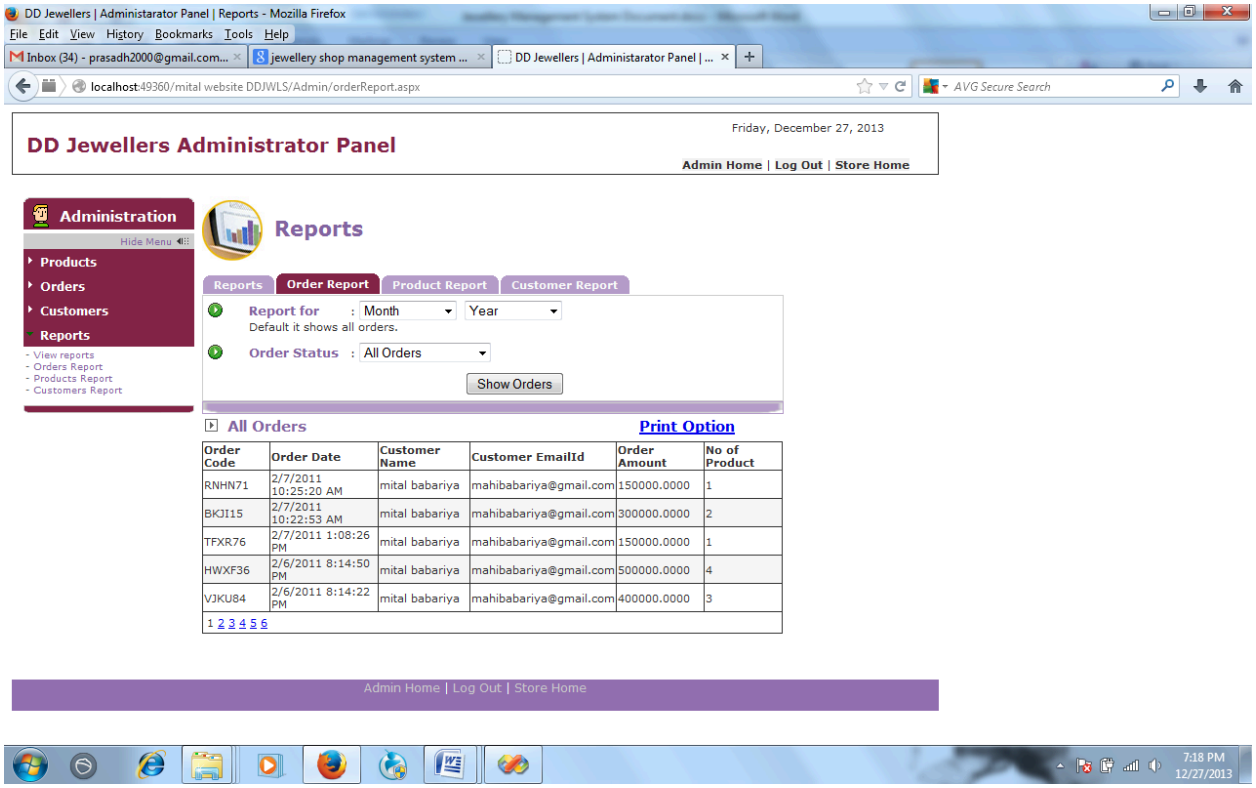

DD Jewellers | Administarator Panel | Home - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Inbox (34) - prasadh2000@gmail.com...
jewellery shop management system ...
DD Jewellers | Administarator Panel | ...
localhost:49360/mital website DDJWLS/Admin/Home.aspx
AVG Secure Search
DD Jewellers Administrator Panel
Friday, December 27, 2013
Admin Home | Log Out | Store Home
Administration
Hide Menu
Products
Orders
Customers
Reports
Account Details
Login Id
Password Option
Personal Details
Name
Address
Comapny Name
Company Address
Contact Number
Fax Number
Edit Details
Other Details
Account Type Administrator
Last Login Label
Account Created On Label
View Products
View Categories
View Orders
View Customers
View Reports
Admin Home | Log Out | Store Home
7:15 PM
12/27/2013

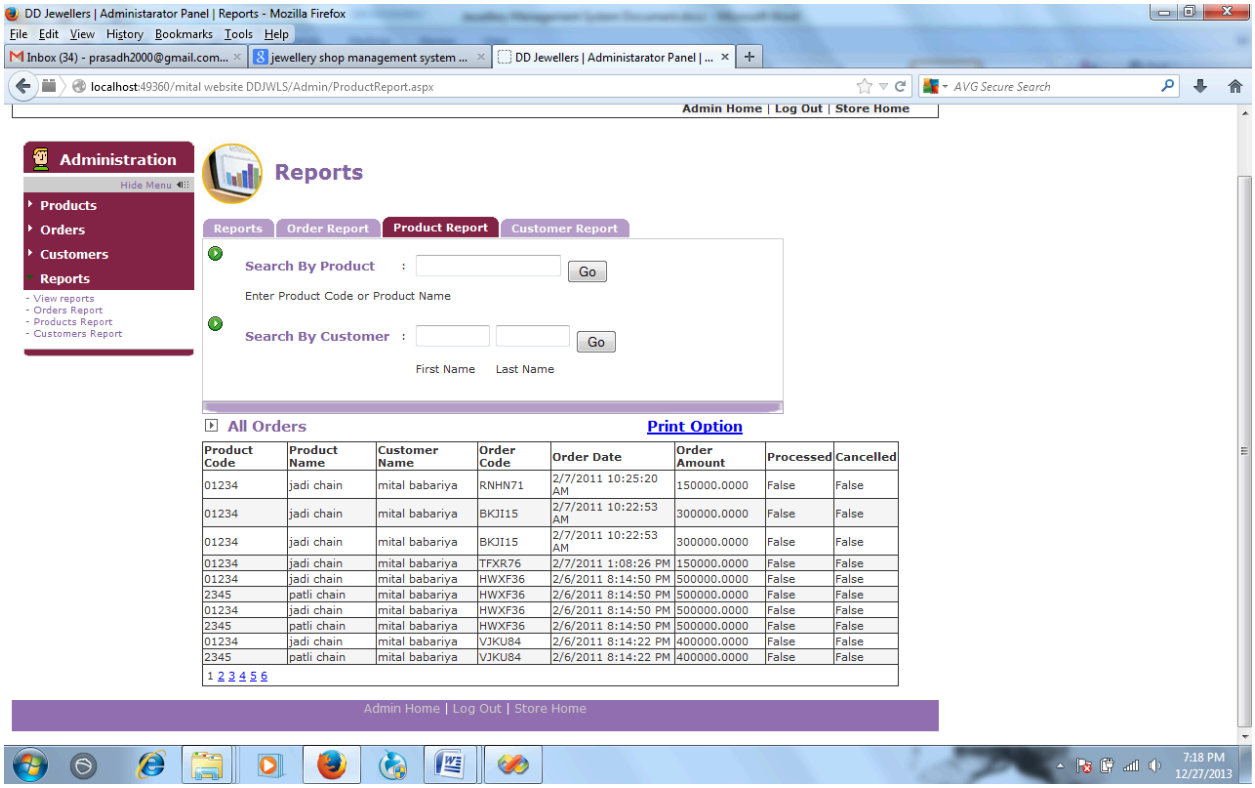

DD Jewellers | Administarator Panel | Products - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Inbox (34) - prasadh2000@gmail.com...
jewellery shop management system ...
DD Jewellers | Administarator Panel | ...
localhost:49360/mital website DDJWLS/Admin/Products.aspx
AVG Secure Search
Administration
Hide Menu
Products
Orders
Customers
Reports
Products
View Products
Add Product
Special & Home Page Option
View Categories
Add Category
Add Product
Add Category
View products in root category chain
View products in category long chain
View products in sub-category sakad
Search
GO
Product List
Product Code
Name
Price
Weight
Image
Short Description
1234
yogi
980000.0000
65 67
this site is very good.
Edit
Remove
01234
jadi chain
150000.0000
50 gm
My Account
asdadsads
Edit
Remove
Admin Home | Log Out | Store Home
7:16 PM
12/27/2013


DD Jewellers | Administarator Panel | Orders - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Inbox (34) - prasadh2000@gmail.com...
jewellery shop management system ...
DD Jewellers | Administarator Panel | ...
localhost:49360/mital website DDJWLS/Admin/Orders.aspx
AVG Secure Search
Administration
Hide Menu
Products
Orders
- View orders
- New orders
- Orders - Processed
- Orders - Cancelled
Customers
Reports
Orders
View
New Orders
Processed
Cancelled
Display orders between (default is all orders)
From: Day Month Year
To: Day Month Year Go
Show all
Orders
Order Code | Order Date | Order Amount | No of Products | Processed | Cancelled
RNHN71 | 2/7/2011 10:25:20 AM | 150000.0000 | 1 | False | False | Customer Details | Product Details
BKJI15 | 2/7/2011 10:22:53 AM | 300000.0000 | 2 | False | False | Customer Details | Product Details
TFXR76 | 2/7/2011 1:08:26 PM | 150000.0000 | 1 | False | False | Customer Details | Product Details
HWXF36 | 2/6/2011 8:14:50 PM | 500000.0000 | 4 | False | False | Customer Details | Product Details
VJKU84 | 2/6/2011 8:14:22 PM | 400000.0000 | 3 | False | False | Customer Details | Product Details
7:17 PM
12/27/2013


DD Jewellers | Administarator Panel | Orders - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Inbox (34) - prasadh2000@gmail.com...
jewellery shop management system ...
DD Jewellers | Administarator Panel | ...
localhost:49360/mital website DDJWLS/Admin/OrdersNew.aspx
AVG Secure Search
DD Jewellers Administrator Panel
Friday, December 27, 2013
Admin Home | Log Out | Store Home
Administration
Hide Menu
Products
Orders
- View orders
- New orders
- Orders - Processed
- Orders - Cancelled
Customers
Reports
Orders
View
New Orders
Processed
Cancelled
Orders
Order Code | Order Date | Order Amount | No of Products | Processed | Cancelled | Select
DPWN23 | 7/6/2009 3:29:08 PM | 0.0000 | 0 | False | False
RNHN71 | 2/7/2011 10:25:20 AM | 150000.0000 | 1 | False | False
BKJI15 | 2/7/2011 10:22:53 AM | 300000.0000 | 2 | False | False
TFXR76 | 2/7/2011 1:08:26 PM | 150000.0000 | 1 | False | False
HWXF36 | 2/6/2011 8:14:50 PM | 500000.0000 | 4 | False | False
Customer Details
Product Details
1 2 3 4 5 6
Select All
Unselect All
Process Selected Order
Cancel Selected Order
Admin Home | Log Out | Store Home
7:17 PM
12/27/2013


DD Jewellers | Administarator Panel | Customers - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Inbox (34) - prasadh2000@gmail.com...
jewellery shop management system ...
DD Jewellers | Administarator Panel | ...
localhost:49360/mital website DDJWLS/Admin/Customers.aspx
AVG Secure Search
DD Jewellers Administrator Panel
Friday, December 27, 2013
Admin Home | Log Out | Store Home
Administration
Hide Menu
Products
Orders
Customers
- View customers
Reports
Customers
View
Search By Name
Go
Customers matching
Customer Name | Occupation | Address | Contact Number | Email Address
dharmesh dobariya | job | balaji nagar pune maharashtra india | 9860253793 9860253793 | dk@gmail.com
mital babariya | student | swagat rajkot gujrat india | 6777688788 7687898908 | mahibabariya@gmail.com
Nikunj Aghera | asd | asd Rajkot Gujarat India | 9016226950 9016226950 | nikunjaghera@gmail.com
yogi changani | student | bca rajkot gujrat india | 9878767876 9878987667 | sonupatel@gmail.com
mital babariya | student | 12 surat gujrat india | 9878767876 9878987667 | mahibabariya@gmail.com
History
1 2
Admin Home | Log Out | Store Home
7:17 PM
12/27/2013


DD Jewellers | Administarator Panel | Reports - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Inbox (34) - prasadh2000@gmail.com...
jewellery shop management system ...
DD Jewellers | Administarator Panel | ...
localhost:49360/mital website DDJWLS/Admin/orderReport.aspx
AVG Secure Search
DD Jewellers Administrator Panel
Friday, December 27, 2013
Admin Home | Log Out | Store Home
Administration
Hide Menu
Products
Orders
Customers
Reports
- View reports
- Orders Report
- Products Report
- Customers Report
Reports
Reports
Order Report
Product Report
Customer Report
Report for : Month Year
Default it shows all orders.
Order Status : All Orders
Show Orders
All Orders
Print Option
Order Code | Order Date | Customer Name | Customer EmailId | Order Amount | No of Product
RNHN71 | 2/7/2011 10:25:20 AM | mital babariya | mahibabariya@gmail.com | 150000.0000 | 1
BKJI15 | 2/7/2011 10:22:53 AM | mital babariya | mahibabariya@gmail.com | 300000.0000 | 2
TFXR76 | 2/7/2011 1:08:26 PM | mital babariya | mahibabariya@gmail.com | 150000.0000 | 1
HWXF36 | 2/6/2011 8:14:50 PM | mital babariya | mahibabariya@gmail.com | 500000.0000 | 4
VJKU84 | 2/6/2011 8:14:22 PM | mital babariya | mahibabariya@gmail.com | 400000.0000 | 3
1 2 3 4 5 6
Admin Home | Log Out | Store Home
7:18 PM
12/27/2013


DD Jewellers | Administarator Panel | Reports - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Inbox (34) - prasadh2000@gmail.com...
jewellery shop management system ...
DD Jewellers | Administarator Panel | ...
localhost:49360/mital website DDJWLS/Admin/ProductReport.aspx
AVG Secure Search
Admin Home | Log Out | Store Home
Administration
Hide Menu
Products
Orders
Customers
Reports
- View reports
- Orders Report
- Products Report
- Customers Report
Reports
Reports
Order Report
Product Report
Customer Report
Search By Product :
Go
Enter Product Code or Product Name
Search By Customer :
Go
First Name
Last Name
All Orders
Print Option
Product Code | Product Name | Customer Name | Order Code | Order Date | Order Amount | Processed | Cancelled
01234 | jadi chain | mital babariya | RNHN71 | 2/7/2011 10:25:20 AM | 150000.0000 | False | False
01234 | jadi chain | mital babariya | BKJI15 | 2/7/2011 10:22:53 AM | 300000.0000 | False | False
01234 | jadi chain | mital babariya | BKJI15 | 2/7/2011 10:22:53 AM | 300000.0000 | False | False
01234 | jadi chain | mital babariya | TFXR76 | 2/7/2011 1:08:26 PM | 150000.0000 | False | False
01234 | jadi chain | mital babariya | HWXF36 | 2/6/2011 8:14:50 PM | 500000.0000 | False | False
2345 | patli chain | mital babariya | HWXF36 | 2/6/2011 8:14:50 PM | 500000.0000 | False | False
01234 | jadi chain | mital babariya | HWXF36 | 2/6/2011 8:14:50 PM | 500000.0000 | False | False
2345 | patli chain | mital babariya | HWXF36 | 2/6/2011 8:14:50 PM | 500000.0000 | False | False
01234 | jadi chain | mital babariya | VJKU84 | 2/6/2011 8:14:22 PM | 400000.0000 | False | False
2345 | patli chain | mital babariya | VJKU84 | 2/6/2011 8:14:22 PM | 400000.0000 | False | False
1 2 3 4 5 6
Admin Home | Log Out | Store Home
7:18 PM
12/27/2013