**E-Commerce Website**

**(Amazon)**

By

Abhinav Sharma

***INDEX***

**CONTENTS**

1. INTRODUCTION

2. ANALYSIS

2.1 SYSTEM ANALYSIS

3. DESIGN APPROACH

3.1 INTRODUCTION TO DESIGN

3.2 UML DIAGRAMS

3.3 DATA FLOW DIAGRAMS

3.4 E-R DIAGRAMS

4. PROJECT MODULES

4.1 TESTING

4.2 TEST CASES

7. CONCLUSION

8. FUTURE ENHANCEMENTS

9. UML DIGRAM

9.1 CLASS

9.2 OBJECT

9.3 DEPLOYMENT

9.4 PACKAGE

**INTRODUCTION:**

**OBJECTIVE:**

The Shopping cart is mainly useful for who haven’t time to go to shopping, those are just entered into this website and bought what ever they want. Even it is night or morning they entered into this site, and chosen different items like fruits, books, toys etc.. ‘Customer is our god’ mainly this website is based on this formula. After chosen items he bought into Pay pal process like VISA or MASTER credit cards or any Debit cards are accepted in this website. Customer is happily shopping at his rest place.

**SYSTEM ANALYSIS**:

1. Existing System

.Existing system is a manual one in which users are maintaining books to store the information like product details, Distributors details, purchases, sales details and accounts for every month. It is very difficult to maintain historical data.

DISADVANTAGES:

The following are the disadvantages of the existing system

* It is difficult to maintain important information in books.
* More manual hours need to generate required reports.
* It is tedious to manage historical data which needs much space to keep all the previous years’ ledgers, books etc.
* Daily sales and purchases details must be entered into books are very difficult to maintain.

2. Proposed System

The DISTRIBUTORS MANAGEMENT TOOL is a software application which avoids more manual hours that need to spend in record keeping and generating reports. This application keeps the data in a centralized way which is available to all the users simultaneously. It is very easy to manage historical data in database. No specific training is required for the distributors to use this application. They can easily use the tool that decreases manual hours spending for normal things and hence increases the performance. It is very easy to record the information of online sales and purchases in the databases.

3. Objective of the System

            The objective of the Distributors Management Tool is to provide better information for the users of this system for better results for their maintainence in the product details that is sales, purchases and stock.

INTRODUCTION TO DESIGN:-

                                       Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

                                          Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

                                          The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer’s requirements into finished software or a system.

                                                   Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

UML Diagrams:

Actor:  
            A coherent set of roles that users of use cases play when interacting with the use `cases.

 Use case:

                              A description of sequence of actions, including variants, that a system performs that yields an observable result of value of an actor.  
  
UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

                        There are various kinds of methods in software design:

They are as follows:

**Ø  Use case Diagram**

**Ø  Sequence Diagram**

**Ø  Collaboration Diagram**

**Ø  Activity Diagram**

**Ø  State chat Diagram**

**USECASE DIAGRAMS:**

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what’s called  an actor.

                        Use case diagram can be useful for getting an overall view of the system and clarifying who can do and more importantly what they can’t do.

                        Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

·         The purpose is to show the interactions between the use case and actor.

·         To represent the system requirements from user’s perspective.

·         An actor could be the end-user of the system or an external system.

**USECASE DIAGRAM:**

A Use case is a description of set of sequence of actions.  Graphically it is rendered as an ellipse with solid line including only its name.  Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship.  It is an association between the use cases and actors.  An actor represents a real-world object.  Primary Actor – Sender, Secondary ActorReceiver.

**SEQUENCE DIAGRAM:**

Sequence diagram and collaboration diagram are called INTERACTION DIAGRAMS. An interaction diagram shows an interaction, consisting of set of objects and their relationship including the  messages that may be dispatched among them.

            A sequence diagram is an introduction that empathizes the time ordering of messages. Graphically a sequence diagram is a table that shows objects arranged along the X-axis and messages ordered in increasing time along the Y-axis

**COLLABORATION DIAGRAM:**

A collaboration diagram is an introduction diagram that emphasizes the structural organization of the objects that send and receive messages. Graphically a collaboration diagram is a collection of vertices and arcs.

**CLASS DIAGRAM:**

            Class is nothing but a structure that contains both variables and methods.  The Class Diagram shows a set of classes, interfaces, and collaborations and their relating ships.  There is most common diagram in modeling the object oriented systems and are used to give the static view of a system.  It shows the dependency between the classes that can be used in our system.

The interactions between the modules or classes of our projects are shown below.  Each block contains Class Name, Variables and Methods.

**CLASS:**

   A description of set of objects that share the same attributes, operations, relationships, and semantics

**DATA FLOW DIAGRAMS**:

 The DFD takes an input-process-output view of a system i.e. data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software.

              Data objects represented by labeled arrows and transformation are represented by circles also called as bubbles. DFD is presented in a hierarchical fashion i.e. the first data flow model represents the system as a whole. Subsequent DFD refine the context diagram (level 0 DFD), providing increasing details with each subsequent level.

             The DFD enables the software engineer to develop models of the information domain & functional domain at the same time. As the DFD is refined into greater levels of details, the analyst perform an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of the data as it moves through the process that embody the applications.

             A context-level DFD for the system the primary external entities produce information for use by the system and consume information generated by the system. The labeled arrow represents data objects or object hierarchy.

**RULES FOR DFD:**

·         Fix the scope of the system by means of context diagrams.

·         Organize the DFD so that the main sequence of the actions

·         Reads left to right and top to bottom.

·          Identify all inputs and outputs.

·         Identify and label each process internal to the system with Rounded   circles.

·         A process is required for all the data transformation and Transfers. Therefore, never connect a data store to a data Source or the destinations or another data store with just a Data flow arrow.

·         Do not indicate hardware and ignore control information.

·         Make sure the names of the processes accurately convey everything the process is done.

·         There must not be unnamed process.

·         Indicate external sources and destinations of the data, with        Squares.

·         Number each occurrence of repeated external entities.

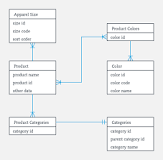
·         Identify all data flows for each process step, except simple Record retrievals.

·         Label data flow on each arrow.

·         Use details flow on each arrow.

·         Use the details flow arrow to indicate data movements.

**ER Digram :-**

[[](https://www.google.com/search?q=What+is+a+ER+diagram+with+example?&rlz=1C1GCEU_en-GBIN967IN967&sxsrf=AOaemvKjVMB8EZS8Rxh_jvHfGmIx2WUy1Q:1630342273077&tbm=isch&source=iu&ictx=1&fir=35td1vDr6E-b6M%252CNLCIPMXuNM1MrM%252C_&vet=1&usg=AI4_-kS4Mc4cz3UD90bE1Xe1fueAm4y3EQ&sa=X&ved=2ahUKEwjE57GgmtnyAhWG4XMBHbtZB2gQ9QF6BAgfEAE#imgrc=35td1vDr6E-b6M)](https://www.google.com/search?q=What+is+a+ER+diagram+with+example?&rlz=1C1GCEU_en-GBIN967IN967&sxsrf=AOaemvKjVMB8EZS8Rxh_jvHfGmIx2WUy1Q:1630342273077&tbm=isch&source=iu&ictx=1&fir=35td1vDr6E-b6M%252CNLCIPMXuNM1MrM%252C_&vet=1&usg=AI4_-kS4Mc4cz3UD90bE1Xe1fueAm4y3EQ&sa=X&ved=2ahUKEwjE57GgmtnyAhWG4XMBHbtZB2gQ9QF6BAgfEAE" \l "imgrc=35td1vDr6E-b6M)

ER diagrams are created based on three basic concepts: entities, attributes and relationships. ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships. At first look, an ER diagram looks very similar to **the flowchart**.

**TESTING:-**

Testing is a process of executing a program with the indent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding.  
  
System Testing is an important phase. Testing represents an interesting anomaly for the software.  Thus a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

A good test case is one that has a high probability of finding an as undiscovered error. A successful test is one that uncovers an as undiscovered error.

Testing Objectives:

1.      Testing is a process of executing a program with the intent of finding an error

2.      A good test case is one that has a probability of finding an as yet undiscovered error

3.      A successful test is one that uncovers an undiscovered error

Testing Principles:

·                     All tests should be traceable to end user requirements

·                     Tests should be planned long before testing begins

·                     Testing should begin on a small scale and progress towards testing in large

·                     Exhaustive testing is not possible

·                     To be most effective testing should be conducted by a independent third party

        The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used. They are

§  White box testing.

§  Black box testing.

**White-box testing:**

White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

**Block-box testing:**

Black box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing mainly focuses on the information domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing functions, interface errors, errors in data structures, error in functional logic are the errors falling in this category.

Testing strategies:

A strategy for software testing must accommodate low-level tests that are necessary to verify that all small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements.

Testing fundamentals:

Testing is a process of executing program with the intent of finding error. A good test case is one that has high probability of finding an undiscovered error. If testing is conducted successfully it uncovers the errors in the software. Testing cannot show the absence of defects, it can only show that software defects present.

Testing Information flow:

Information flow for testing flows the pattern. Two class of input provided to test the process. The software configuration includes a software requirements specification, a design specification and source code.

Test configuration includes test plan and test cases and test tools. Tests are conducted and all the results are evaluated. That is test results are compared with expected results. When erroneous data are uncovered, an error is implied and debugging commences.

Unit testing:

Unit testing is essential for the verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules.  Using the detailed design description as a guide, important paths are tested to uncover errors with in the boundary of the modules.  These tests were carried out during the programming stage itself. All units of ViennaSQL were successfully tested.

Integration testing :

Integration testing focuses on unit tested modules and build the program structure that is dictated by the design phase.

System testing:

System testing tests the integration of each module in the system. It also tests to find discrepancies between the system and it’s original objective, current specification and system documentation. The primary concern is the compatibility of individual modules. Entire system is working properly or not will be tested here, and specified path ODBC connection will correct or not, and giving output or not are tested here these verifications and validations are done by giving input values to the system and by comparing with  expected output. Top-down testing implementing here.

**ACCEPTANCE TESTING:**

This testing is done to verify the readiness of the system for the implementation. Acceptance testing begins when the system is complete. Its purpose is to provide the end user with the confidence that the system is ready for use. It involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements.

Tools to special importance during acceptance testing include:

Test coverage Analyzer – records the control paths followed for each test case.

Timing Analyzer – also called a profiler, reports the time spent in various regions of the code are areas to concentrate on to improve system performance.

Coding standards – static analyzers and standard checkers are used to inspect code for deviations from standards and guidelines.

Test Cases:

Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

Using White-Box testing methods, the software engineer can drive test cases that

·         Guarantee that logical decisions on their true and false sides.

·         Exercise all logical decisions on their true and false sides.

·         Execute all loops at their boundaries and with in their operational bounds.

·         Exercise internal data structure to assure their validity.

The test case specification for system testing has to be submitted for review before system testing commences.

**CONCLUSION:**

The package was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project.

Ø  Automation of the entire system improves the efficiency

Ø  It provides a friendly graphical user interface which proves to be better when compared to the existing system.

Ø  It gives appropriate access to the authorized users depending on their permissions.

Ø  It effectively overcomes the delay in communications.

Ø  Updating of information becomes so easier.

Ø  System security, data security and reliability are the striking features.

Ø  The System has adequate scope for modification in future if it is necessary.

**FUTURE ENHANCEMENTS:**

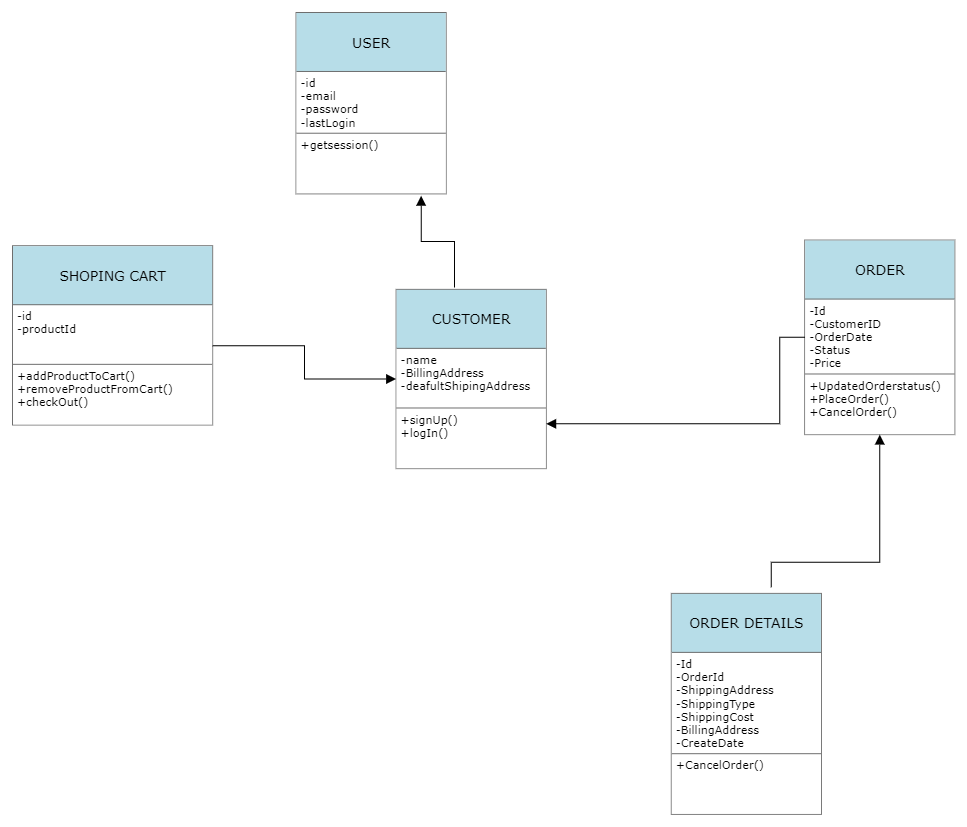
                   This application avoids the manual work and the problems concern with it. It is an easy  way to obtain the information regarding the various products information that are present in the Super markets.

Well I and my team members have worked hard in order to present an improved website better than  the existing one’s regarding the information about the various activities. Still ,we found out that the project can be done in a better way. Primarily, when we request information about a particular product it just shows the company, product id, product name and no. of quantities available. So, after getting the information we can get access to the product company website just by a click on the product name .

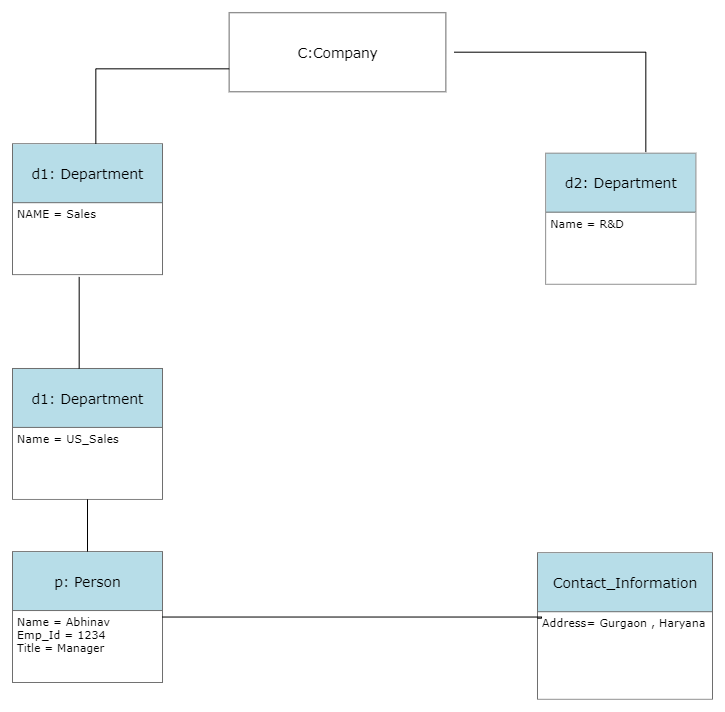
                                           The next enhancement that we can add the searching option. We can directly search to the particular product company  from this site .These are the two enhancements that we could think of at present.

UML DIGRAM :-

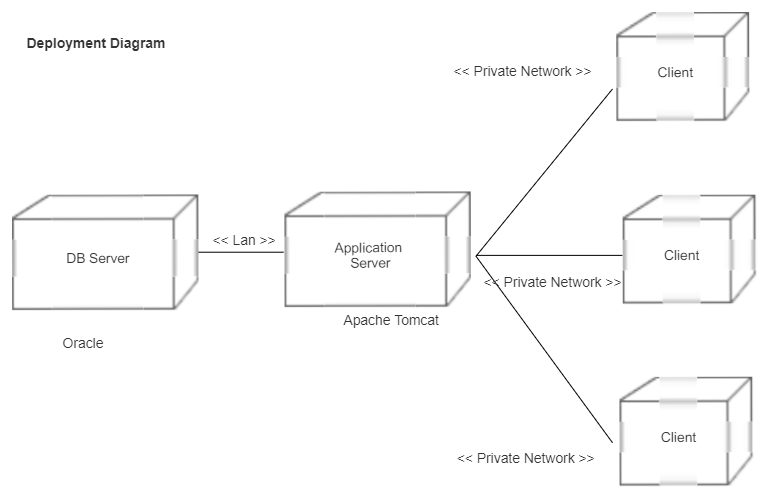
1. CLASS

****

1. Object

****

1. Deployment



1. Package

