A PROJECT REPORT

ON

Quick shop – A shoping website application

The Partial Fulfillment of the Requirements for the Award of the Degree

"BACHELOR OF COMPUTER APPLICATION"

KIRODIMAL GOVT. ARTS & SCIENCE COLLEGE RAIGARH

YEAR-2021

PROJECT GUIDE	SUBMITTED BY
Kamni	
KAMINI PATEL	NATASHA BHATPAHARE
Signature	

ATAL BIHARI VAJPAYEE VISHWAVIDYALAYA

BILASPUR(C.G.)

CERTIFICATE

This is to certify that project work entitled **"Quick shop - A shopping website Application"** for **KIRODIMAL GOVT. ARTS & SCIENCE COLLEGE RAIGARH** has been
prepared by **Natasha bhatpahre** is student of **B.C.A. 6 SEM** of our College. This Project is Submitted in partial fulfillment of requirement of (C.G.) for **B.C.A.** Course and completed under the guidance of our Lecturer.

[illegible]
Signature of Student

[illegible]
Signature of Guide

ACKNOWLEDGEMENTS

At every outset I express my gratitude to almighty lord for showering his grace and blessings upon me to complete this project. Although our name appears on the cover of this book, many people had contributed in some form or the other form to this project Development. We could not done this project without the assistance or support of each of the following we thank you all.

I wish to place on my record my deep sense of gratitude to my project guide, **MISS. KAMINI PATEL GUEST LECTURER** of **KIRODIMAL GOVT.ARTS & SCIENCE COLLEGE RAIGARH** for her constant motivation and valuable help through the project work. Express my gratitude to her valuable suggestions and advices throughout the **B.C.A.** course. I also extend my thanks to other Faculties for their Cooperation during my Course.

CERTIFICATE BY THE EXAMINERS

This is to certify that the project entitled

"Quick shop - A shopping website Application"

Submitted by

NATASHA NHATPAHARE

Has been examined as a part of Examination for the award of " Bachelor of Computer Application" on Atal Bihari Vajpayee University Bilaspur (C.G) India.

$\boxed{\frac{79}{100}}$

[illegible]

Internal Examiner	External Examiner
Name: G.P. Banaj	Name:
Date: 01.10.2021	Date:

Table Content

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.

PROJECT OVERVIEW:

Once customer entered with his own username and password, at that time automatically one shopping cart will be created, once user select an item it will add to cart. In case user thinks the selected item is not useful for me, then deleted that item from shopping cart.

Customer selected some items, but in his credit or debit cart haven't that much balance, then he was logout from the website, the selected items are stored at cart with specific users with his allotted carts, after some days he bought those items then automatically deleted from the cart.

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.

System Specifications

Hardware Requirements:-

- Pentium-IV(Processor).
- 256 MB Ram
- 512 KB Cache Memory
- Hard disk 10 GB
- Microsoft Compatible 101 or more Key Board

Software Requirements: -

Web Technologies	: ASP.NET 2.0
Language	: C#
Database	: SQL SERVER 2000,05
Web Server	: IIS
Operating System	: WINDOWS XP/7/10

INTRODUCTION OF 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.

E-R Diagrams:

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represents data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design For the database designer, the utility of the ER model is:

- it maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
- it is simple and easy to understand with a minimum of training.
- Therefore, the model can be used by the database designer to communicate the design to the end user.
- In addition, the model can be used as a design plan by the database developer to implement a data model in a specific database management software.

Connectivity and Cardinality

The basic types of connectivity for relations are: one-to-one, one-to-many, and many-to-many. A *one-to-one* (1:1) relationship is when at most one instance of a entity A is associated with one instance of entity B. For example, "employees in the company are each assigned their own office. For each employee there exists a unique office and for each office there exists a unique employee.

A *one-to-many* (1:N) relationships is when for one instance of entity A, there are zero, one, or many instances of entity B, but for one instance of entity B, there is only one instance of entity A. An example of a 1:N relationships is

a department has many employees

each employee is assigned to one department

A *many-to-many* (M:N) relationship, sometimes called non-specific, is when for one instance of entity A, there are zero, one, or many instances of entity B and for one instance of entity B there are zero, one, or many instances of entity A. The connectivity of a relationship describes the mapping of associated

ER Notation

There is no standard for representing data objects in ER diagrams. Each modeling methodology uses its own notation. The original notation used by Chen is widely used in academics texts and journals but rarely seen in either CASE tools or publications by non-academics. Today, there are a number of notations used, among the more common are Bachman, crow's foot, and IDEFIX.

All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. The notation used in this document is from Martin. The symbols used for the basic ER constructs are:

- **entities** are represented by labeled rectangles. The label is the name of the entity. Entity names should be singular nouns.
- **relationships** are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs
- **attributes,** when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should be singular nouns.
- **cardinality** of many is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.
- **existence** is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional

PROJECT MODULES

MODULES : This project contains 3 modules, those are

- **Admin**
- **Products**
- **User**

MODULES DESCRIPTION:

Admin:-
When admin login, he saw the customer's database, means how many users are authenticated to this website and how many users are transact everyday, and newly items are inserting into products.

Products:-
This module contains product name, and related image, and cost of its. Like toys, books, furniture, gold items, etc.. Whatever customer wants from the shopping cart.

User:-
User entered into with his username and password, when he entered into this, he saw what items are available today, this facility is available for this site. Chosen different items from website get those through door delivery.

Database Tables:

Admin Table:

Column Name	Type	Computed	Length
uname	varchar	no	20
password	varchar	no	20

Products Table:

Column Name	Type	Computed	Length
Pid	int	no	4
ProductName	varchar	no	100
ProductType	varchar	no	50
image	varchar	no	100
price	int	no	4

ABOUT INTERNET AND INTRANET

Technologically, the Internet is network of computers. Not just a few special Computers, but over nine million of all kinds of computers. Similarly it is not just a network, but a network of networks hence the name and using TCP/IP (transmission control protocol and internet protocol).

Internet is the name for a vast, worldwide system consisting of people, information and computers. Internet is global communication system of diverse, INTER connected computer NETWORK for the exchange of information of virtually every conceivable topic known to man. Internet is not just one thing. It is a lot of things to lot of people. In today's world it is one of the most important commodity of life. The Internet is more important in what it enables than what it is, more of a phenomenon than fact.

Intranet

The classical definition of Intranet is the application of the Internet technologies to the internal business applications media most refer to the Intranet in terms of applying web technologies to information systems in the organization.

Functions:

Functions are bet declared between the tag of HTML page. Functions are called by user-initiated events. Seems reasonable to keep the functions between the tags. They are loaded first before a user can do anything that might call a function. Scripts can be placed between inside comment fields to ensure that older browser do not display the script itself.

If we want to test this one immediately and you are using a Java Script enabled browser then please go ahead and push the button.
This script will create a button and when you press it a window will pop up saying "hello!". In fact we have a lot of possibilities just by adding functions to our scripts.
The common browsers transmit the form information by either method: here's the complete tag including the GET transmission method attribute for the previous form

Example

....

Input elements.

Use the tag to define any one of a number of common form elements including text fields multiple choice lists click able images and submission buttons. There are many attributers for this tag only that types and name attributes are required for each element, each type of input element uses only a subset of the followed attributes. Additional attributes may be required based upon which type of the form element you specify.

Submit button:

The submit button (

) does what its name implies, settings in motion the form's submission to the server from the browser. We many have more than submit buttons will be added to the parameter list the browser sends along to the server.

Example

< Input type ="submit'>

Reset button:

The reset button if firm button is nearly self- explanatory; it lets the user reset erase or set to some default value all elements in the form. By default the browser displays a reset button worth the label "reset". We can change that by specifying a value attribute with tour own button label.

FEASIBILITY STUDY:

Feasibility study is conducted once the problem is clearly understood. Feasibility study is a high level capsule version of the entire system analysis and design process. The objective is to determine quickly at a minimum expense how to solve a problem. The purpose of feasibility is not to solve the problem but to determine if the problem is worth solving.

The system has been tested for feasibility in the following points.

1. Technical Feasibility
2. Economical Feasibility
3. Operational Feasibility.

1. Technical Feasibility

The project entitles "Courier Service System" is technically feasibility because of the below mentioned feature. The project was developed in Java which Graphical User Interface.

It provides the high level of reliability, availability and compatibility. All these make Java an appropriate language for this project. Thus the existing software Java is a powerful language.

2. Economical Feasibility

The computerized system will help in automate the selection leading the profits and details of the organization. With this software, the machine and manpower utilization are expected to go up by 80-90% approximately. The costs incurred of not creating the system are set to be great, because precious time can be wanted by manually.

3. Operational Feasibility

In this project, the management will know the details of each project where he may be presented and the data will be maintained as decentralized and if any inquires for that particular contract can be known as per their requirements and necessaries.

IMPLEMENTAYION:

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively.

The system can be implemented only after thorough testing is done and if it is found to work according to the specification.

It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system.

The more complex the system being implemented, the more involved will be the systems analysis and design effort required just for implementation.

The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then changes over to his new fully tested system and the old system is discontinued.

INTRODUCTION TO HTML4.0

What is the World Wide Web?

The World Wide Web is a network of information resources. The Web relies on three mechanisms to make these resources readily available to the widest possible audience.

1. A uniform naming scheme for locating resources on the Web (e.g. URLs)
2. Protocols, for access to named resources over the Web (e.g. HTTP)
3. Hypertext, for easy navigation among resources (e.g.HTML)

The ties between the three mechanisms are apparent throughout this specification.

What is HTML?

To publish information for global distribution, one needs a universally understood language, a kind of publishing mother tongue that all computers may potentially understand. The publishing language used by the World Wide Web is HTML (from Hyper Text Markup Language). HTML gives authors the means to

- Publish online documents with headings, text, tables, lists, photos, etc.
- Retrieve online information via hypertext links, at the click of a button
- Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products etc.
- Include spread - sheets, video clips, sound clips, and other applications directly in their documents.

A brief history of HTML:

HTML was originally developed by Tim Berners-Lee while at CERN, and popularized by the Mosaic browser developed at NCSA. During the course of the 1990s it has blossomed with the explosive growth of the Web during this time. HTML has been extended in a number of ways. The Web depends on Web page authors and vendors sharing the same conventions for HTML. This has motivated joint work on specifications for HTML.

HTML 2.0 (November 1995) was developed under the aegis of the Internet Engineering Task Force (IETF) to codify common practice in late 1994. HTML (1993) and ([HTML.30]) (1995) proposed much richer versions of HTML, despite never receiving consensus in standards discussions, these drafts led to the adoption of a range new features. The efforts of the World Wide Web Consortium's HTML working group to codify common in 1996 resulted in HTML 3.2 (January 1997). Most people agree that HTML documents should work well across different browsers and platforms. Achieving interoperability lowers costs to content providers since they must develop only one version of a document. If the effort is not made, there is much greater risk that the Web will devolve into a proprietary world of incompatible formats, ultimately reducing the Web's commercial potential for all participants.

SOFTWARE METHODOLOGY

The software methodology followed in this project includes the object-oriented methodology and the application system development methodologies. The description of these methodologies is given below.

Application System Development – A Life cycle Approach

Although there are a growing number of applications (such as decision support systems) that should be developed using an experimental process strategy such as prototyping, a significant amount of new development work continue to involve major operational applications of broad scope. The application systems are large highly structured. User task comprehension and developer task proficiency is usually high. These factors suggest a linear or iterative assurance strategy. The most common method for this stage class of problems is a system development life cycle modal in which each stage of development is well defined and has straightforward requirements for deliverables, feedback and sign off. The system development life cycle is described in detail since it continues to be an appropriate methodology for a significant part of new development work.

The basic idea of the system development life cycle is that there is a well-defined process by which an application is conceived and developed and implemented. The life cycle gives structure to a creative process. In order to manage and control the development effort, it is necessary to know what should have been done, what has been done, and what has yet to be accomplished. The phrases in the system development life cycle provide a basis for management and control because they define segments of the

flow of work, which can be identified for managerial purposes and specifies the documents or other deliverables to be produced in each phase.

The phases in the life cycle for information system development are described differently by different writers, but the differences are primarily in the amount of necessity and manner of categorization. There is a general agreement on the flow of development steps and the necessity for control procedures at each stage.

The information system development cycle for an application consists of three major stages.

1) Definition.
2) Development.
3) Installation and operation.

The first stage of the process, which defines the information requirements for a feasible cost effective system. The requirements are then translated into a physical system of forms, procedures, programs etc., by the system design, computer programming and procedure development. The resulting system is test and put into operation. No system is perfect so there is always a need for maintenance changes. To complete the cycle, there should be a post audit of the system to evaluate how well it performs and how well it meets the cost and performance specifications. The stages of definition, development and installation and operation can therefore be divided into smaller steps or phrases as follows.

Definition

Proposed definition	: preparation of request for proposed applications.
Feasibility assessment	: evaluation of feasibility and cost benefit of proposed system.
Information requirement analysis	: determination of information needed.

Design

Conceptual design	:	User-oriented design of application development.
Physical system design	:	Detailed design of flows and processes in applications processing system and preparation of program specification.

Development

Program development	:	coding and testing of computer programs.
Procedure development	:	design of procedures and preparation of user instructions.

Installation and operation

Conversion	:	final system test and conversion.
Operation and maintenance	:	Month to month operation and maintenance

Post audit : Evaluation of development process,application system and results of use at the completion of the each phase, formal approval sign-off is required from the users as well as from the manager of the project development.

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.

- Automation of the entire system improves the efficiency
- It provides a friendly graphical user interface which proves to be better when compared to the existing system.
- It gives appropriate access to the authorized users depending on their permissions.
- It effectively overcomes the delay in communications.
- Updating of information becomes so easier.
- System security, data security and reliability are the striking features.
- 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.

BIBLIOGRAPHY

The following books were referred during the analysis and execution phase of the project

Common Language Runtime
-By Steven Pratschner

SOFTWARE ENGINEERING
-By Roger S. Pressman

IMAGES
-Google Search

HTML PUBLISHING BIBLE
- Alan Simpson.

C# 2008
-Andrew Troelson

WEBSITES:
www.google.com
www.KCPWholesale.com

Thank you