**1. PROJECT INTRODUCTION**

**Casey Camera Club project** is an e-commerce site which deals with the selling of all the varieties of Camera and their accessories. The main point regarding this project is that the user can see the products but to buy the product they must be a registered user. This Project can be used by any person who is having general knowledge about internet. All the users will be first considered as anonymous user later if he needs any service then he will be treated as registered user.

**1.1 Features of the Project:**

The Proposed System provides online Selling of Cameras and their accessories. It also provides additional services to the registered user. The development of this new system contains the following activities, which try to automate the entire process keeping in the view of database integration approach.

* User Friendliness is provided in the application with various controls provided by system Rich User Interface.
* The system makes the overall project management much easier and flexible.
* It can be accessed over the Internet.
* The Product information files can be stored in centralized database which can be maintained by the system.
* This project also provides a special feature of comparing two different Camera which helps user to make decision easily.
* Product reviews are also available to clarify the user’s perspective of the product.

**1.2 Main parts of the Project:-**

1. **Front View:-**
2. Register: - First of all anyone new to this portal will have to sign up to register him/her to the site. After his/her registration, the user will be able to buy the product available on this site. It will be absolutely free.
3. Log In: - After their complete registration, they will be able to login their account. The user must login his/her account with their username and password.
4. Dashboard:- After complete login a dashboard will be open which contains the following important links:-

* User Control Panel: - user can also check their order status and their reviews give about the products.
* Edit your profile:-The login person can also edit their person details.
* Change password: - She/he can also change their password by simply typing their old password.
* Log out: - The user can also log out from his account.

1. **Admin Control Panel**: - admin can modify all users account and add new account in the database. Only admin can access database. Admin can add new items, modify items and also can check reviews given by the users.
2. **Back end View:-**
3. At back end we will use asp.net website application to make the forms for add a Main Page, Register, login, Product Description, view Reviews and search items with us.
4. For Sign up/ Login we will use the concept of Session available in asp.net.
5. In database we will make tables as:-

* Users:- Contains columns of the signup forms
* Order : - Contain all detail about the order given by users
* Reviews: - Contains detail about the reviews given by user about the products
* Product:-Contain the detail about the products.
* Product Category: - Contain detail about the Category of Product.
* Shopping Cart: - Contain the detail of buying products by user.

**1.3 Advantages/Scope:-**

The main problem with the city people is that they even don’t know what they want. So, the main advantage / scope of this project are that it will help the people’s to get all they want in just single click. We think its best idea to bring the city people under one roof and it will also helps to compare two different products which makes user to take decision easily.

**2. System Development Life Cycle**

**2.1 Life Cycle Phases:-**

**Initiation Phase**

The initiation of a system (or project) begins when a business need or opportunity is identified. A Project Manager should be appointed to manage the project. This business need is documented in a Concept Proposal. After the Concept Proposal is approved, the System Concept Development Phase begins.

**System Concept Development Phase**

Once a business need is approved, the approaches for accomplishing the concept are reviewed for feasibility and appropriateness. The Systems Boundary Document identifies the scope of the system and requires Senior Official approval and funding before beginning the Planning Phase.

**Planning Phase**

The concept is further developed to describe how the business will operate once the approved system is implemented, and to assess how the system will impact employee and customer privacy. To ensure the products and /or services provide the required capability on-time and within budget, project resources, activities, schedules, tools, and reviews are defined. Additionally, security certification and accreditation activities begin with the identification of system security requirements and the completion of a high level vulnerability assessment.

**Requirements Analysis Phase**

Functional user requirements are formally defined and delineate the requirements in terms of data, system performance, security, and maintainability requirements for the system. All requirements are defined to a level of detail sufficient for systems design to proceed. All requirements need to be measurable and testable and relate to the business need or opportunity identified in the Initiation Phase.

**Design Phase**

The physical characteristics of the system are designed during this phase. The operating environment is established, major subsystems and their inputs and outputs are defined, and processes are allocated to resources. Everything requiring user input or approval must be documented and reviewed by the user. The physical characteristics of the system are specified and a detailed design is prepared. Subsystems identified during design are used to create a detailed structure of the system. Each subsystem is partitioned into one or more design units or modules. Detailed logic specifications are prepared for each software module.

**Development Phase**

The detailed specifications produced during the design phase are translated into hardware, communications, and executable software. Software shall be unit tested, integrated, and retested in a systematic manner. Hardware is assembled and tested.

**Integration and Test Phase**

The various components of the system are integrated and systematically tested. The user tests the system to ensure that the functional requirements, as defined in the functional requirements document, are satisfied by the developed or modified system. Prior to installing and operating the system in a production environment, the system must undergo certification and accreditation activities.

**Implementation Phase**

The system or system modifications are installed and made operational in a production environment. The phase is initiated after the system has been tested and accepted by the user. This phase continues until the system is operating in production in accordance with the defined user requirements.

**Operations and Maintenance Phase**

The system operation is ongoing. The system is monitored for continued performance in accordance with user requirements, and needed system modifications are incorporated. The operational system is periodically assessed through In-Process Reviews to determine how the system can be made more efficient and effective. Operations continue as long as the system can be effectively adapted to respond to an organization’s needs. When modifications or changes are identified as necessary, the system may reenter the planning phase.

**Disposition Phase**

The disposition activities ensure the orderly termination of the system and preserve the vital information about the system so that some or all of the information may be reactivated in the future if necessary. Particular emphasis is given to proper preservation of the data processed by the system, so that the data is effectively migrated to another system or archived in accordance with applicable records management regulations and policies, for potential future access.

**2.2 SDLC Objectives:-**

This guide was developed to disseminate proven practices to system developers, project managers, program/account analysts and system owners/users throughout the DOJ. The specific objectives expected include the following:

* To reduce the risk of project failure
* To consider system and data requirements throughout the entire life of the system
* To identify technical and management issues early
* To disclose all life cycle costs to guide business decisions
* To foster realistic expectations of what the systems will and will not provide
* To provide information to better balance programmatic, technical, management, and cost aspects of proposed system development or modification
* To encourage periodic evaluations to identify systems that are no longer effective
* To measure progress and status for effective corrective action
* To support effective resource management and budget planning
* To consider meeting current and future business requirements

**2.3 Key Principles:-**

This guidance document refines traditional information system life cycle management approaches to reflect the principles outlined in the following subsections. These are the foundations for life cycle management.

***Life Cycle Management Should be used to Ensure a Structured Approach to Information Systems Development, Maintenance, and Operation***

This SDLC describes an overall structured approach to information management. Primary emphasis is placed on the information and systems decisions to be made and the proper timing of decisions. The manual provides a flexible framework for approaching a variety of systems projects. The framework enables system developers, project managers, program/account analysts, and system owners/users to combine activities, processes, and products, as appropriate, and to select the tools and methodologies best suited to the unique needs of each project.

***Support the use of an Integrated Product Team***

The establishment of an Integrated Product Team (IPT) can aid in the success of a project. An IPT is a multidisciplinary group of people who support the Project Manager in the planning, execution, delivery and implementation of life cycle decisions for the project. The IPT is composed of qualified empowered individuals from all appropriate functional disciplines that have a stake in the success of the project. Working together in a proactive, open communication, team oriented environment can aid in building a successful project and providing decision makers with the necessary information to make the right decisions at the right time.

***Each System Project must have a Program Sponsor***

To help ensure effective planning, management, and commitment to information systems, each project must have a clearly identified program sponsor. The program sponsor serves in a leadership role, providing guidance to the project team and securing, from senior management, the required reviews and approvals at specific points in the life cycle. An approval from senior management is required after the completion of the first seven of the SDLC phases, annually during Operations and Maintenance Phase and six-months after the Disposition Phase. Senior management approval authority may be varied based on dollar value, visibility level, congressional interests or a combination of these.

The program sponsor is responsible for identifying who will be responsible for formally accepting the delivered system at the end of the Implementation Phase.

***A Single Project Manager must be Selected for Each System Project***

The Project Manager has responsibility for the success of the project and works through a project team and other supporting organization structures, such as working groups or user groups, to accomplish the objectives of the project. Regardless of organizational affiliation, the Project Manager is accountable and responsible for ensuring that project activities and decisions consider the needs of all organizations that will be affected by the system. The Project Manager develops a project charter to define and clearly identify the lines of authority between and within the agency’s executive management, program sponsor, (user/customer), and developer for purposes of management and oversight.

***A Comprehensive Project Management Plan is Required for Each System Project***

The project management plan is a pivotal element in the successful solution of an information management requirement. The project management plan must describe how each life cycle phase will be accomplished to suit the specific characteristics of the project. The project management plan is a vehicle for documenting the project scope, tasks, schedule, allocated resources, and interrelationships with other projects. The plan is used to provide direction to the many activities of the life cycle and must be refined and expanded throughout the life cycle.

***Specific Individuals Must be Assigned to Perform Key Roles throughout the Life Cycle***

Certain roles are considered vital to a successful system project and at least one individual must be designated as responsible for each key role. Assignments may be made on a full- or part-time basis as appropriate. Key roles include program/functional management, quality assurance, security, telecommunications management, data administration, database administration, logistics, financial, systems engineering, test and evaluation, contracts management, and configuration management. For most projects, more than one individual should represent the actual or potential users of the system (that is, program staff) and should be designated by the Program Manager of the program and organization

**2.4 Systems Analysis and Design:-**

Systems analysis and design refers to the process of examining a business situation with the intent of improving it through better procedures and methods. Systems development can generally be thought of as having two major components:

1. Systems Analysis

2. Systems Design.

**Systems Analysis** is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvement to the system. In brief, we can say that analysis specifies what the system should do. Design states hew to accomplish the objective.

**Systems Design** is the process of planning a new system or replace or complement an existing system. But before this planning can be done, we must thoroughly understand the existing system and determine how computers can best be used to make its operation more effective.

**Systems Study** may be defined as "a study of the operations of a set of connected elements and of the interconnections between these elements". It shows clearly that one cannot ignore any part or element of a system without first finding out the effect that element has on the operation of the system as a whole. We can understand this with the help of systems analysis.

There is a difference between "systems approach" and "systems analysis" also. The systems approach shows a set of procedure for solving a particular problem. It applies scientific methods to observe, clarify, identify and solve a problem with special care being taken to understand the inter- relatedness between elements and their system characteristics.

However, systems analysis is a management technique which helps us in designing a new system or improving an existing system.

**System development**, a process consisting of the two major steps of systems analysis and design, starts when management or sometimes system development personnel feel that a new system or an improvement in the existing system is required. The systems development life cycle is classically, thought of as the set of activities- that analysts, designers and users carry out to develop and implement an information system.

The systems development life cycle consists of the following activities:

· Preliminary investigation

· Determination of system requirements

· Design of system

· Development of software

· Systems testing

· Implementation, evaluation and maintenance

**Preliminary investigation**

A request to take assistance from information systems can be made for many reasons, but in each case someone in the organization initiates the request When the request is made. the first systems activity the preliminary investigation begins. This activity has three parts:

i. Request, clarification

ii. Feasibility study

iii. Request approval

**Determination of system requirements**

At the heart of systems analysis is a detailed understanding of all important facets of the business area under investigation. The key questions are

· What is being done?

· How is it being done?

· How frequently does it occur?

· How great is the volume of transactions or decisions?

· How well is the task being performed?

· Does a problem exist?

· If a problem exists, how serious is it? What is the underlying cause?

To answer the above questions, systems analysts discuss with different category of persons to collect the facts about the business process and their opinions of why things happen as they do and their views for changing the existing process. During analysis, data are collected on the available files, decision points and transactions handled by the present system. Some tools are used in analysis like data flow diagrams, interviews, on-site observations and questionnaires. Detail investigations also require the study of manuals and reports. Once the structured analysis is completed, analyst has a firm understanding of what is to be done?

**Design of system**

The design of an information system produces the details that clearly describe how a system will meet the requirements identified during systems analysis. Systems specialists often refer to this stage as logical design, in contrast to the process of developing program software, which is referred to as physical design.

Systems analysts begin the design process by identifying reports and other outputs system will produce. Then the specific data on each are pinpointed. The systems design also describes the data to be input, calculated or stored. Individual data items and calculation procedures are written in detail. Designers select file structures and storage devices, such as magnetic disk, magnetic tape or even paper files. Procedures they write tell how to process the data and produce the output. The documents containing the design specifications portray the design in many different ways-charts, tables, and special symbols. The detailed design information is passed on to the programming staff for the purpose of software development. Designers are responsible for providing programmers with complete and clearly out lined software specifications.

**Development of software**

Software developers may install purchased software or they may develop new, custom- designed programs. The choice depends on the cost of each option, the time available to develop software and the availability of programmers. Generally it has been observed that programmers are part of permanent professional staff in a big organization. In smaller organization, without programmers, outside programming services may be hired or retained on a contractual basis. Programmers are also responsible for documenting the program, providing an explanation of how and why certain procedures are coded in specific ways. Documentation is essential to test the program and carry on maintenance once the application has been installed.

**3. Overview of ASP.Net Technology**

**3.1 Introduction:-**

ASP.Net is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web application for PC, as well as mobile devices.

ASP.Net works on top of the HTTP protocol and uses the HTTP commands and policies to set a browser-to-server two-way communication and cooperation.

ASP.Net is a part of Microsoft .Net platform. ASP.Net applications are complied codes, written using the extensible and reusable components or objects present in .Net framework. These codes can use the entire hierarchy of classes in .Net framework.

The ASP.Net application codes could be written in either of the following languages:

* C#
* Visual Basic .Net
* Jscript
* J#

ASP.Net is used to produce interactive, data-driven web applications over the internet. It consists of a large number of controls like text boxes, buttons and labels for assembling, configuring and manipulating code to create HTML pages.

## 3.2 ASP.Net Web Forms Model:-

ASP.Net web forms extend the event-driven model of interaction to the web applications. The browser submits a web form to the web server and the server returns a full markup page or HTML page in response.

All client side user activities are forwarded to the server for stateful processing. The server processes the output of the client actions and triggers the reactions.

Now, HTTP is a stateless protocol. ASP.Net framework helps in storing the information regarding the state of the application, which consists of:

* Page state
* Session state

The page state is the state of the client, i.e., the content of various input fields in the web form. The session state is the collective obtained from various pages the user visited and worked with, i.e., the overall session state. To clear the concept, let us take up an example of a shopping cart as follows.

User adds items to a shopping cart. Items are selected from a page, say the items page, and the total collected items and price are shown in a different page, say the cart page. Only HTTP cannot keep track of all the information coming from various pages. ASP.Net session state and server side infrastructure keeps track of the information collected globally over a session. The ASP.Net runtime carries the page state to and from the server across page requests while generating the ASP.Net runtime codes and incorporates the state of the server side components in hidden fields.

This way the server becomes aware of the overall application state and operates in a two-tiered connected way.

## ASP.Net Component Model:-

The ASP.Net component model provides various building blocks of ASP.Net pages. Basically it is an object model, which describes:

* Server side counterparts of almost all HTML elements or tags, like <form> and <input>.
* Server controls, which help in developing complex user-interface for example the Calendar control or the Grid view control.

ASP.Net is a technology, which works on the .Net framework that contains all web-related functionalities. The .Net framework is made of an object-oriented hierarchy. An ASP.Net web application is made of pages. When a user requests an ASP.Net page, the IIS delegates the processing of the page to the ASP.Net runtime system.

The ASP.Net runtime transforms the .aspx page into an instance of a class, which inherits from the base class Page of the .Net framework. Therefore, each ASP.Net page is an object and all its components i.e., the server-side controls are also objects.

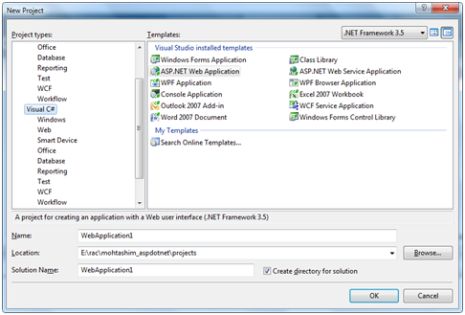
## 3.3 The Visual Studio IDE:-

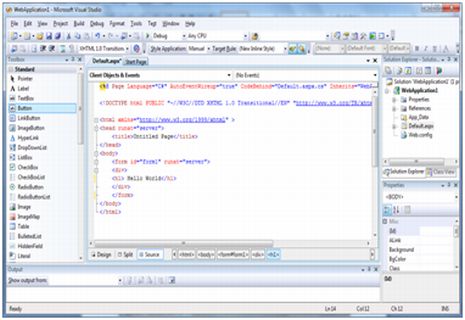
The new project window allows choosing an application template from the available templates.

When you start a new web site, ASP.NET provides the starting folders and files for the site, including two files for the first web form of the site.

The file named Default.aspx contains the HTML and asp code that defines the form, and the file named Default.aspx.cs (for C# coding) or the file named Default.aspx.vb (for vb coding) contains the code in the language you have chosen and this code is responsible for the form's works.

The primary window in the Visual Studio IDE is the Web Forms Designer window. Other supporting windows are the Toolbox, the Solution Explorer, and the Properties window. You use the designer to design a web form, to add code to the control on the form so that the form works according to your need, you use the code editor.





## Ways to work with views and windows:-

The following are the ways to work with different windows:

* To change the Web Forms Designer from one view to another, click on the Design or source button.
* To close a window, click on the close button on the upper right corner and to redisplay, select it from the View menu.
* To hide a window, click on its Auto Hide button; the window changes into a tab, to redisplay again click on the Auto Hide button again.
* To size a wind just drag it.

## Adding folders and files to your web site:-

When a new web form is created, Visual Studio automatically generates the starting HTML for the form and displays it in Source view of the web forms designer. The Solution Explorer is used to add any other files, folders or any existing item on the web site.

* To add a standard folder, right-click on the project or folder under which you are going to add the folder in the Solution Explorer and choose New Folder.
* To add an ASP.Net folder, right-click on the project in the Solution Explorer and select the folder from the list.
* To add an existing item to the site, right-click on the project or folder under which you are going to add the item in the Solution Explorer and select from the dialog box.

## 3.4 Projects and Solutions:-

A typical ASP.Net application consists of many items: the web content files (.aspx), source files (e.g., the .cs files), assemblies (e.g., the .dll files and .exe files), data source files (e.g., .mdb files), references, icons, user controls and miscellaneous other files and folders. All these files that make up the website are contained in a Solution.

When a new website is created VB2008 automatically creates the solution and displays it in the solution explorer.

Solutions may contain one or more projects. A project contains content files, source files, and other files like data sources and image files. Generally the contents of a project are compiled into an assembly as an executable file (.exe) or a dynamic link library (.dll) file.

Typically a project contains the following content files:

* Page file (.aspx)
* Master page (.master)
* Site map (.sitemap)
* Website configuration file (.config)

## Building and Running a Project:-

The application is run by selecting either Start or Start without Debugging from the Debug menu, or by pressing F5 or Ctrl-F5. The program is built i.e. the .exe or the .dll files are generated by selecting a command from the Build menu.

ASP.Net life cycle specifies, how:

* ASP.Net processes pages to produce dynamic output
* The application and its pages are instantiated and processed
* ASP.Net compiles the pages dynamically

The ASP.Net life cycle could be divided into two groups:

* Application Life Cycle
* Page Life Cycle

An ASP.Net page is made of number of server controls along with the HTML controls, text and images. Sensitive data from the page and the states of different controls on the page are stored in hidden fields and forms the context of that page request. ASP.Net runtime controls all association between a page instance and its state. An ASP.Net page is an object of the Page Class or inherited from it.

All the controls on the pages are also objects of the related control class inherited from a parent Control class. When a page is run an instance of the page object is created along with all its content controls. An ASP.Net page is also a server side file saved with the .aspx extension. It is modular in nature and can be divided into the following core sections:

* Page directives
* Code Section
* Page Layout

**3.5 Database Access in Asp.Net:-**

ASP.Net allows the following sources of data to be accessed and used:

* Databases (e.g., Access, SQL Server, Oracle, MySQL)
* XML documents
* Business Objects
* Flat files

ASP.Net hides the complex processes of data access and provides much higher level of classes and objects through which data is accessed easily. These classes hide all complex coding for connection, data retrieving, data querying and data manipulation.

ADO.Net is the technology that provides the bridge between various ASP.Net control objects and the backend data source. We will come to ADO.Net in due time. In this tutorial, we will look at data access and working with the data without going into the details of its inner workings.

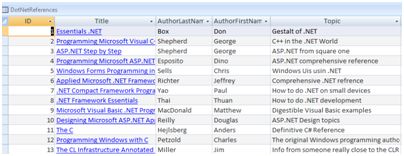
## 3.6 Retrieve and display data:-

It takes two types of data controls to retrieve and display data in ASP.Net:

* **A data source control**. it manages the connection to the data, selection of data and other jobs like paging and caching of data etc.
* **A data view control**. it binds and displays the data and allows data manipulation.

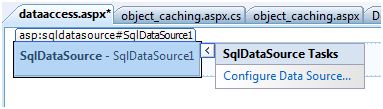
We will discuss the data binding and data source controls in details later. In this section, we will use a SqlDataSource control to access data and a GridView control to display and manipulate data. We will also use an Access database, which has details about .Net books available in the market. Name of our database is ASPDotNetStepByStep.mdb and we will use the data table DotNetReferences.

The table has the following columns: ID, Title, AuthorFirstName, AuthorLastName, Topic, and Publisher. Here is a snapshot of the data table:

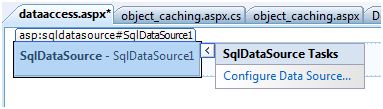


Let us directly move to action, take the following steps:

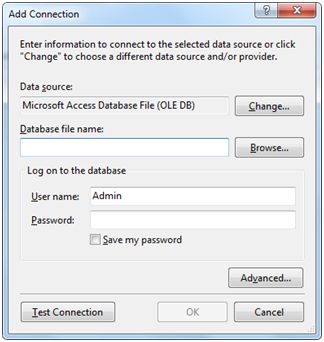
**(1)** Create a web site and add a SqlDataSourceControl on the web form.



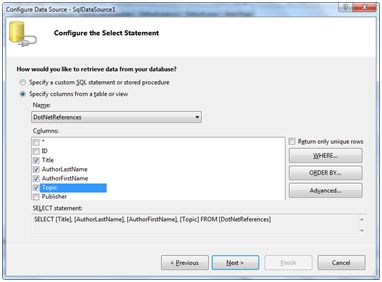
**(2)** Click on the Configure Data Source Link.



**(3)** Click on the New Connection button to establish connection with a database.

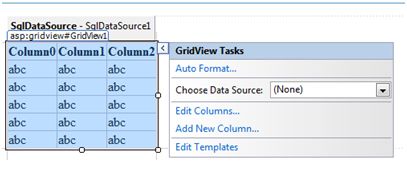


**(4)** Once the connection is set up, you may save it for further use. At the next step, you are asked to configure the select statement:

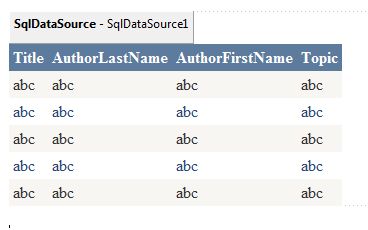


**(5)** Select the columns and click next to complete the steps. Observe the WHERE, ORDER BY, AND the Advanced. Buttons. These buttons allow you to provide the where clause, order by clause and specify the insert, update and delete commands of SQL respectively. This way, you can manipulate the data.

**(6)** Add a Grid View control on the form. Choose the data source and format the control using AutoFormat option.



**(7)** After this the formatted Grid View control displays the column headings, and the application is ready to run.



# 4. System Design

# 4.1 Data Flow Diagram

Data Flow Diagramming is a means of representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data processes, and data sources/destination.

The data flow diagram is analogous to a road map. It is a network model of all possibilities with different detail shown on different hierarchical levels. This processes of representing different details level is called “leveling” or “partitioning” by some data flow diagram advocates. Like a road map, there is no starting point or stop point, no time or timing, or steps to get somewhere. We just know that the data path must exist because at some point it will be needed. A road map shows all existing or planned roads because the road is needed.

Details that is not shown on the different levels of the data flow diagram such as volumes, timing, frequency, etc. is shown on supplementary diagrams or in the data dictionary. For example, data store contents may be shown in the data dictionary.

Data Flow Diagram (DFD) uses a number of symbols to represent the systems. Data Flow Diagram also known as ‘Bubble Chart’ is used to clarify system requirements and identifying the major transformations that will become programs in system design. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the level of details.

**Terms used in DFD**

* **Process**

A process transforms data values. The lowest level processes are pure functions without side effects. An entire data flow graphics high level process.

*Graphical representation:*

**Graphical Representation: **

* **Data flows**

A data flow connects the output of an object or process to input of another object or process. It represents the intermediate data value within a computation. It is represented by an arrow and labeled with a description of data, usually its name or type.

* **Actors**

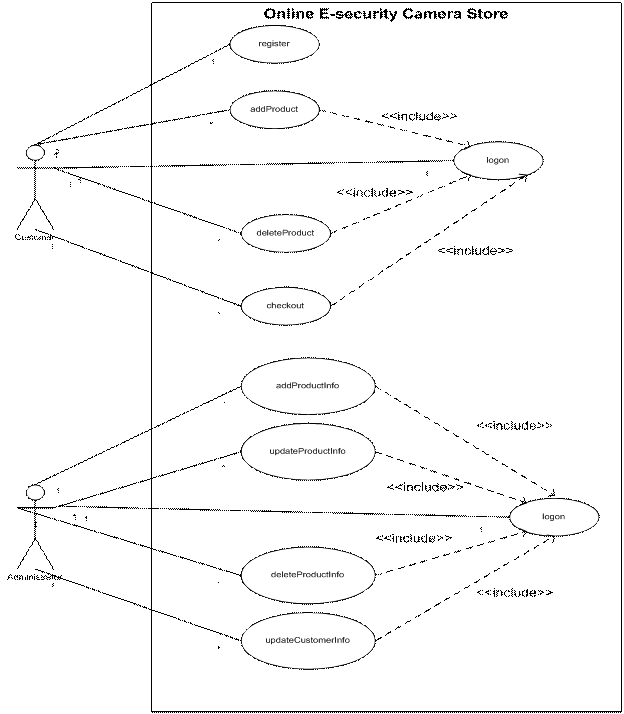
An actor is active object that drives the data flow graph by producing or consuming values.

* **Data store**

A data store is a passive object with in a data flow diagram that stores data for later access.

* **External Entity**

A rectangle represents an external entity such as a librarian ,a library member.



Data Flow Diagram of Casey Camera Club

**4.2 E-R Diagram**

**Definition:**

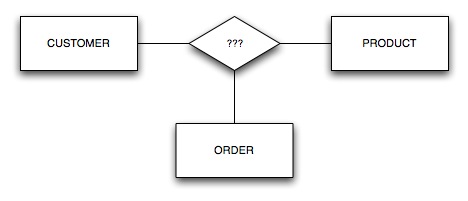
An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

**Entity Relationship (ER) diagram:**

This diagramming technique is used to visually present a database schema or data model and was original proposed by Chen in the 1970s. There are many different data modeling notations; some are very similar to UML class diagrams (with the exception of operations). However, the notation the used here is slightly different, as proposed by Elmasri, et al. The database schema for this system is shown in figure. The table object has been left out of the diagram because the table management feature set had been dropped from the requirements before this stage of the design process.

Some important database design decisions are as follows:

* To store the total price of an order with the order rather than calculating it on the fly when looking at past orders. This is because the price of menu items could change at any time, so the total price at the time of ordering must be stored so that the total price is not incorrectly calculated in future.
* Similar to the previous point, the order receipt is stored as a hard-copy and not regenerated when reviewing past orders because things such as the restaurant name or VAT percentage are subject to change. Receipts stored need to be exactly the same as the customer copy in case of dispute.

****

**5. Database Tables**

1. **Order Table**: This table will store full detail of order given by customer.

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Constraint/Description** |
| SerialNumber | Int | Not Null(Primary Key) |
| UserName | Nvarchar(50) | Not Null |
| Amount | Float | Not Null |

1. **Users:**  This table shall store general information of the users. It contains following fields.

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Constraint/Description** |
| ID | Int | Not Null(Primary Key) |
| Username | nvarchar(50) | Not Null |
| Password | nvarchar(50) | Not Null |
| Email | nvarchar(50) | Not Null |
| Address | nvarchar(50) | Not Null |
| Name | nvarchar(50) | Not Null |
| Idnumber | nvarchar(50) | Not Null |
| User Type | nvarchar(50) | Null |

1. **Product Catogery:** This table will store tell thecatogery of the product.

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Constraint/Description** |
| CatogeryID | Int | Not Null (Primary Key) |
| Catogery | varchar(50) | Not Null |

1. **Product Table:** This table will store details of the product which are present on our website.

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Constraint/Description** |
| ProductID | Int | Not Null (Primary key) |
| ProductName | nvarchar(50) | Not Null |
| ProductDescription | nvarchar(2000) | Not Null |
| ProductShortDescription | nvarchar(500) | Not Null |
| Price | Float | Null |
| Catogery | nvarchar(50) | Null |
| ProductImage | nvarchar(500) | Null |
| ProductImageThumb | nvarchar(50) | Not Null |

(5). **Reviews:** This table stores reviews given by customer about the products.

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Constraint/Description** |
| ReviewID | int | Not Null(Primary key) |
| Username | nvarchar(50) | Not Null |
| Review | nvarchar(500) | Not Null |
| ProductID | nvarchar(50) | Not Null |

(6). **Shpping Cart:** This table contains the information about the product purchased by customer.

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Constraint/Description** |
| ShoppingID | int | Not Null(Primary key) |
| ShoppingIDNo | nvarchar(50) | Not Null |
| UserID | nvarchar(50) | Not Null |
| SalesStatus | nvarchar(50) | Not Null |
| qty | int | Not Null |

**6. IMPLEMENTATION**

**6.1 Introduction**

System implementation is the stage when the user has thoroughly tested the system and approves all the features provided by the system. The various tests are performed and the system is approved only after all the requirements are met and the user is satisfied.

The new system may be totally new, replacing an existing manual or automated system, or it may be a major modification to an existing system. In either case, proper implementation is essential to provide a reliable system to meet organizational requirements. Successful implementation may not guarantee improvement in the organization using the new system (that is a design question), but improper will prevent it.

Implementation is the process of having systems personnel check out and put new equipment into use, train users, install the new application and construct any files of data needed to use it. This phase is less creative than system design. Depending on the size of the organization that will be involved in using the application and the risk involved in its use, systems developers may choose to test the operation in only one area of the firm with only one or two persons. Sometimes, they will run both old and new system in parallel way to com-pare the results. In still other situations, system developers stop using the old system one day and start using the new one the next.

The implementation of the web based or lan based networked project has some extra steps at the time of implementation. We need to configure the system according the requirement of the software.

For the project we need to install and configure Weblogic server 8.1 , database server, and the deployment directory for the project.

# Implementation Tools

The project was implemented using the below mentioned tools. Also, the implementation work was carried out in Windows XP/2000 server platform.

1. Visual Studio 2010
2. IIS Server
3. Ms SQL Server 2005

**Coding**

This means program construction with procedural specifications has finished and the coding for the program begins:

* Once the design phase was over, coding commenced
* Coding is natural consequence of design.
* Coding step translate a detailed design representation of software into a programming language realization.
* Main emphasis while coding was on style so that the end result was an optimized code.

The following points were kept into consideration while coding:

## Coding Style

The structured programming method was used in all the modules the project. It incorporated the following features

* The code has been written so that the definition and implementation of each function is contained in one file.
* A group of related function was clubbed together in one file to include it when needed and save us from the labor of writing it again and again.

## Naming Convention:-

* As the project size grows, so does the complexity of recognizing the purpose of the variables. Thus the variables were given meaningful names; which would help in understanding the context and the purposes of the variable.
* The function names are also given meaningful names that can be easily understood by the user.

**7. HARDWARE & SOFTWARE REQUIREMENTS**

**Hardware Requirements:**

It is recommended that the minimum configuration for clients is as appended below:-

Suggested Configuration of Windows clients:-

Microprocessor : - Pentium-4 class processor, 450 megahertz (MHz)

Ram : - 256 MB of RAM

Hard Disk : - 40 gigabytes (GB) on installation drive, which

Includes 500 MB on system drive.

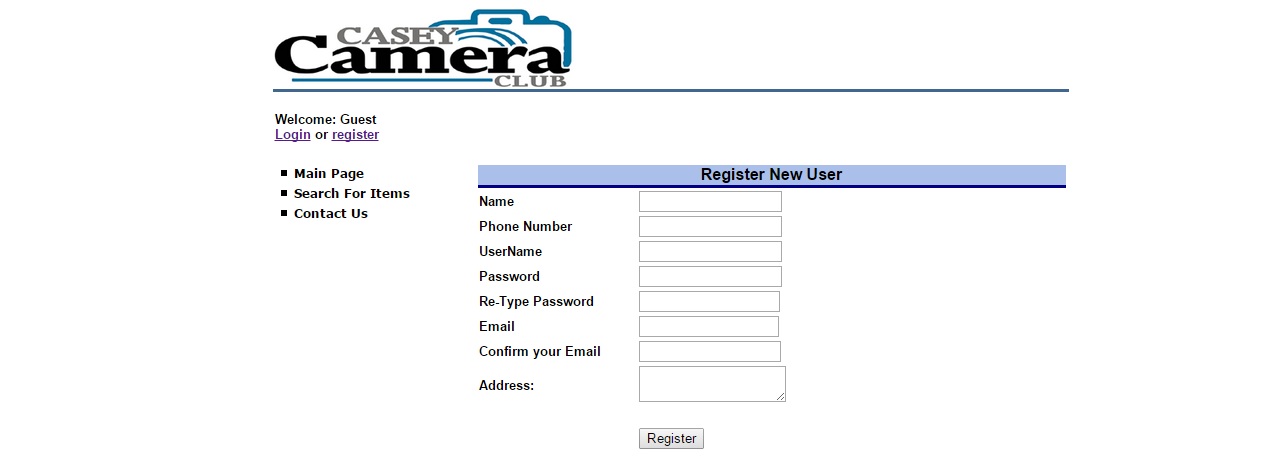
CD ROM Drive : - 52 X CD ROM Drive

**Software requirements:**

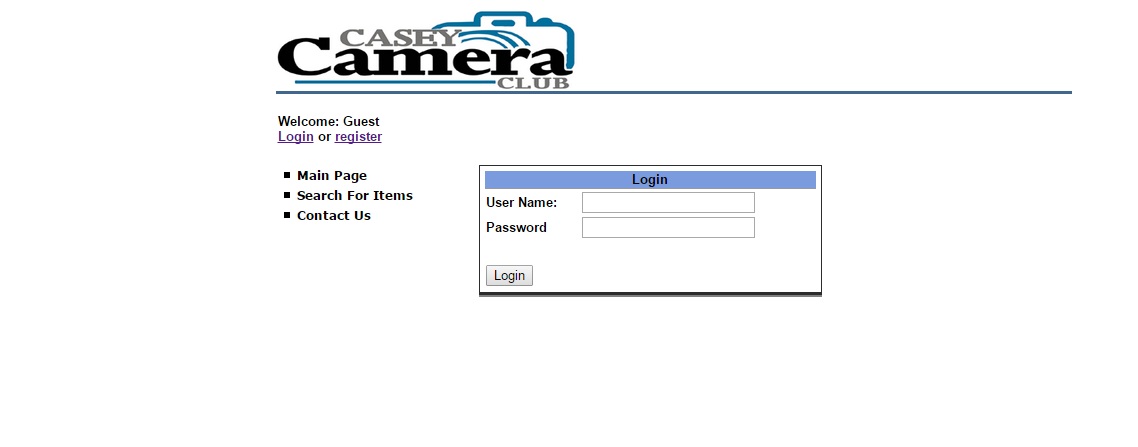
* Windows 98 / XP operating system/MAC OS/Window 7
* Latest Browser
* Visual Studio
* SQL Server

**8. PROJECT SNAPSHOTS:**

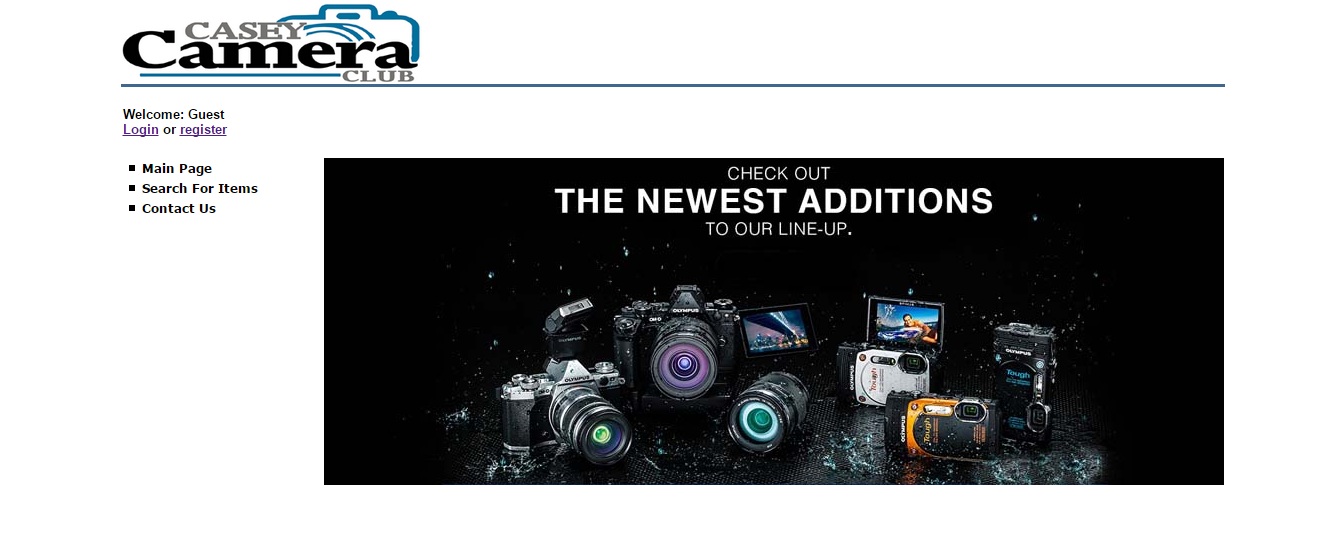
**Sign Up:-**

****

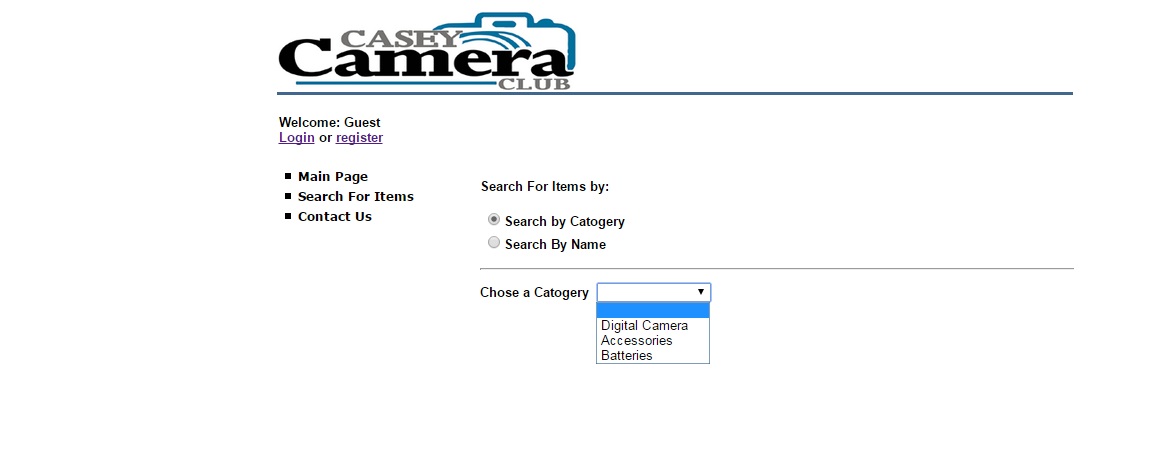
**Login:-**

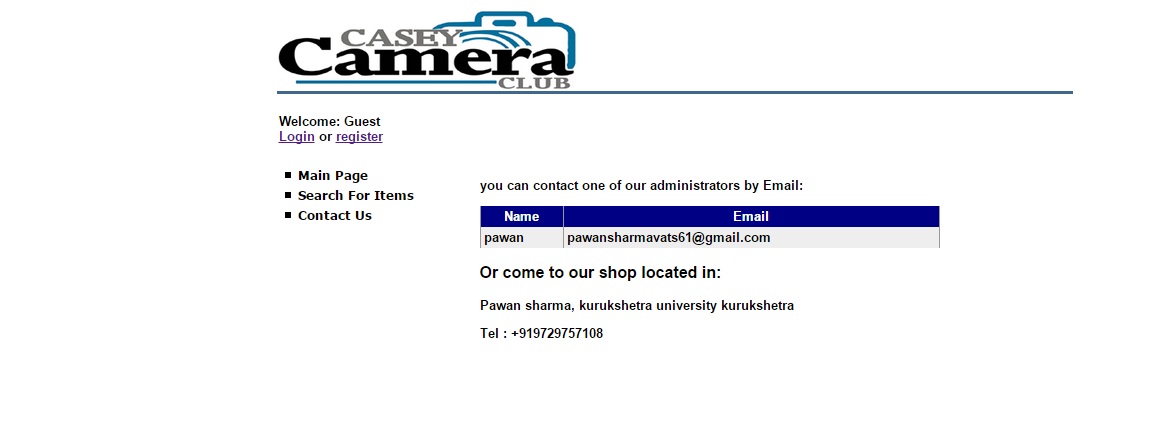
****

**Main Page:-**

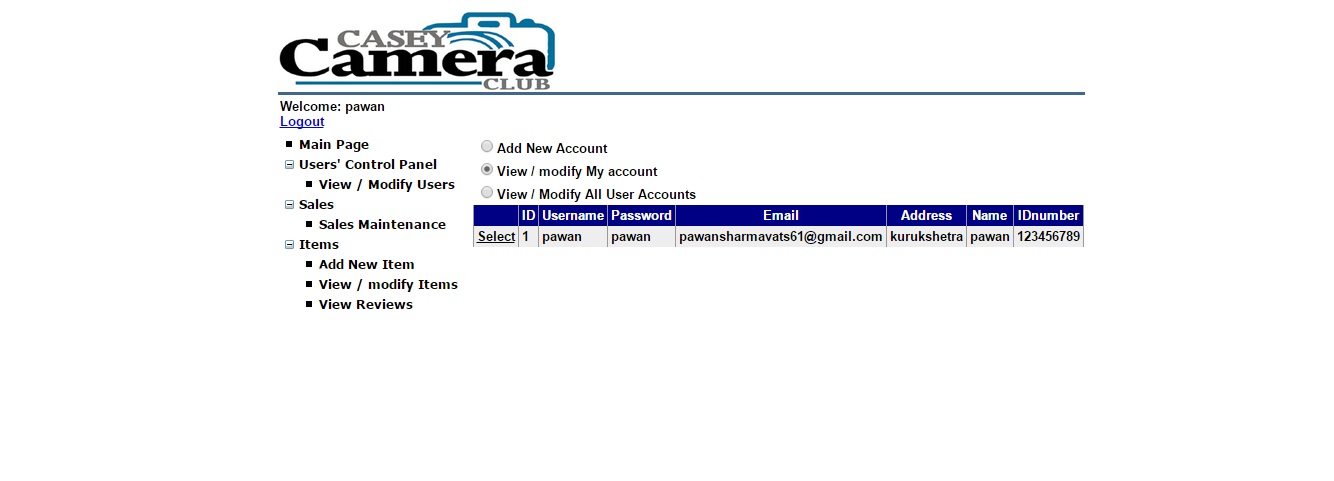


**Search For Items:-**

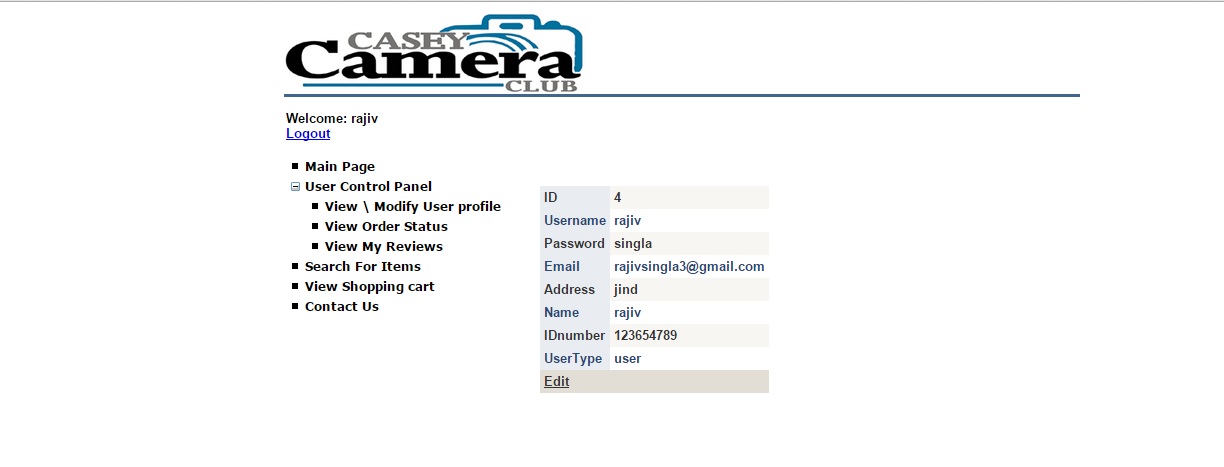
****

**Contact Us:- **

**Admin Controls:-**

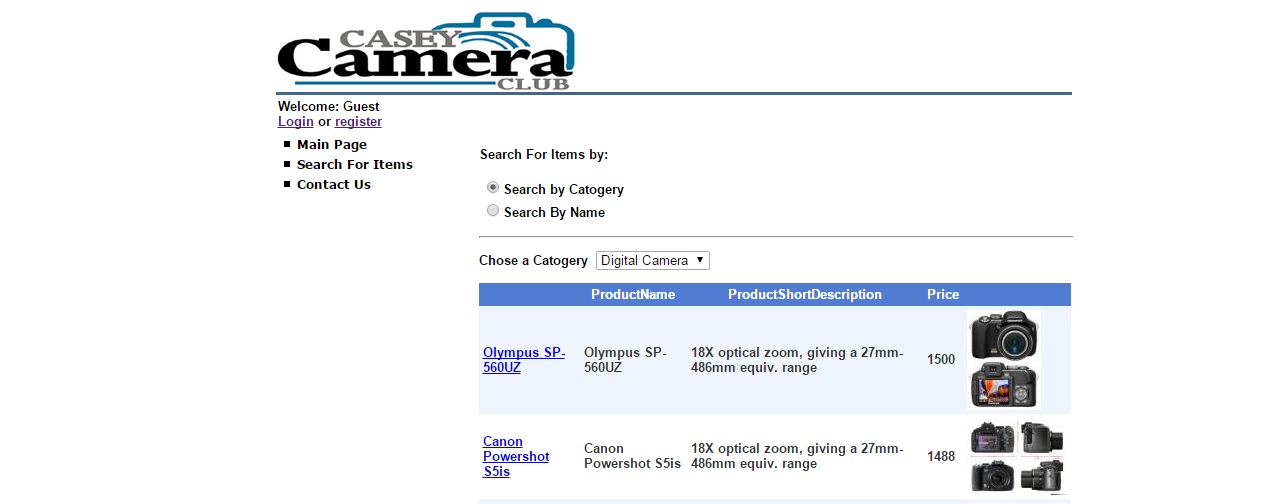
****

**User Controls:-**

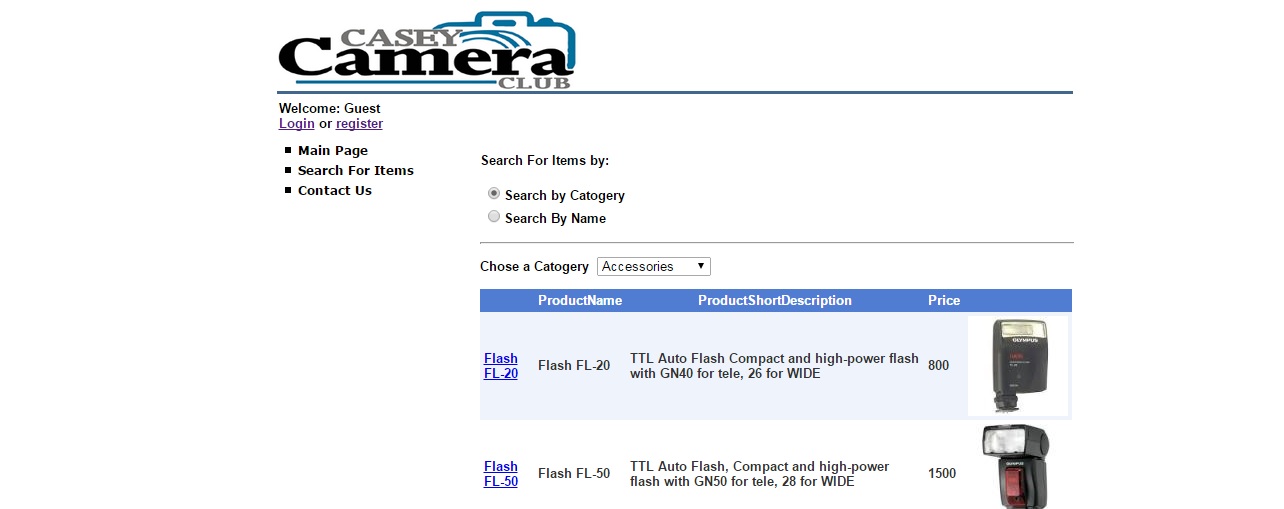
****

**Produsts:-**

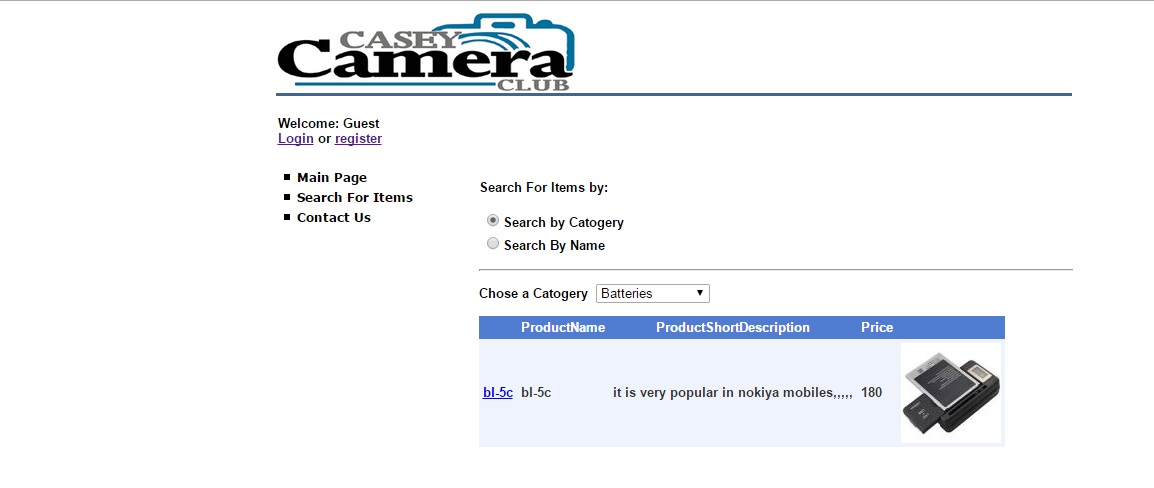
**Digital Camera:-**

****

**Accessories:-**

****

**Batteries:-**

****

**9. TESTING PHASE**

During earlier development phases an attempt is made to build Topology from an  
abstract concept to tangible implementation. Topology testing is a critical element of quality assurance and represents ultimate review of specification;

Design and coding notion of “correctness” of the Topology just developed and overcome a conflict of interest that occurs errors are recovered. A thorough testing of system before any implementation is mandatory. This is because implementing a new system is a major job, which requires a lot of man- hours and other resources, so an error not detected before implementation may cost a lost. Effective testing early in a process is also necessary because in some cases a small error not detected and corrected early before installation may explore into much large problems. Testing refers to the process of exercising and evaluating the system or system components by automated means to verify that it satisfies the specified requirements or to identify differences between expected and actual results. Testing is the critical element of software quality assurance and represents the ultimate review of specification, design and coding. The importance of software testing and its implications with respect to software quality cannot be overemphasized.

**Test Objective:**

Testing is a process of execution a program with the intent of finding error.

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

A successful test is one that uncovers an as- yet- discovered error.

**Integration Testing**

The objective of integration testing is to test the integration of and communication between components. Additionally, it may include testing the integration of subsystems or communication with external systems. On some projects, integration testing may be divided into two levels: Assembly Testing and System Integration Testing.   
During Assembly testing, the integration of the software components is tested.   
During system Integration testing, the communication with external systems is tested.   
In Integration testing, individual modules are put together making a larger application. This can be done repeatedly until the program is assembled in its entirety. The individual components are combined with other components to make sure that necessary communications, links and data sharing occur properly.   It is not truly system testing because the components are not implemented in the operating environment.  The integration phase requires more planning and some reasonable sub-set of production-type data.  Larger systems often require several integration steps.

**System Testing**

The objectives of system testing are to find defects that are attributable to the behavior of the system as a whole, rather than the behavior of individual components, and to test that the software functions as a complete system. This level of testing is different from integration testing in that the tests are concerned with the entire system, not just the interactions between components. Other than system functionality and behavior, system testing may include testing configuration, throughput, security, resource utilization, and performance.

My Project’s **Casey Camera Club** each function and event has been tested individually.

**10. Future Scope:-**

The main problem with the city people is that they even don’t know what they want . So, the main advantage / scope of this project is that it will help the people’s to get all they want in just single click. We think its best idea to bring the city people under one roof and it will also helps to compare two different products which makes user to take decision easily.

**11.References (Bibliography)**

* [**www.asp.net**](http://www.asp.net)
* [**www.asp.net.com**](http://www.asp.net.com)
* [**www.wikipedia.org**](http://www.wikipedia.org)
* [**www.edutechlearners.com**](http://www.edutechlearners.com)