**PROJECT REPORT**

**ON**

**“Carsalon”**

**BACHELOR’S OF SCIENCE**

**IN**

**Computer Science**

***-****SUBMITTED BY-*

**Prof. Shaila Sharin**

- Project Guide –

**COSMOPOLITAN’S**

**VALIA C. L. COLLEGE,**

**D. N. NAGAR, ANDHERI (w), MUMBAI - 400053**

**\*2015-2016\***

# Preface

It gives me great pleasure in presenting this project .This project was developed with a motive to computerize every transaction’s like selling, inventory etc. Aim for better information storage and easily retrieve details for further by Insignia Solutions.

The Front end and Back end used for this project is PHP, HTML and MYSQL Server respectively, as java serves us with all its reliable, dynamic, security, Flexible and more features on the Front end whereas pilling up into various tables at the backend is efficiently handled by SQL Server.

This project report inculcates the basic logic of software development under book the perception of OOAD features enriched with diagrams so that the reader may easily apprehend all the details.

More prominence is given to the perseverance and readability of the user for which this documentation is enhanced with a User’s Manual, to explain the features of the Project.

Though this project satisfies the current requirements of the user to the optimum, but there is always a scope for further enhancement when required.

# Acknowledgement

This research project would not have been possible without the support of many people. The author wishes to express her gratitude to her supervisor, H.O.D Prof. Hrushikesh Panaksar who was abundantly helpful and offered invaluable assistance, support and guidance. Deepest gratitude is also due to the members of the supervisory committee, Assoc. Prof. Shaila Sharin without whose knowledge and assistance this study would not have been successful. The author would also like to convey thanks to the Ministry and Faculty for providing the financial means and laboratory facilities. The author wishes to express her love and gratitude to her beloved families; for their understanding & endless love, through the duration of her studies.

INDEX

|  |  |  |
| --- | --- | --- |
| Sr. No. | Title | Page No. |
| 1.1 | Organizational Overview |  |
| 1.2 | Description Of System |  |
| 1.3 | Present System |  |
| 1.4 | Limitations of the present system |  |
| 1.5 | Proposed system and its advantages |  |
| 1.6 | Feasibility Study |  |
| 1.7 | About The Technology |  |
| 1.8 | Database (MY SQL) - An Overview |  |
| 1.9 | Stakeholders |  |
| 1.10 | Gantt Chart |  |
| 2 | Software Process |  |
| 2.1 | Software Development Process Model(spiral) |  |
| 3 | System Analysis |  |
| 3.1 | Fact Finding Techniques |  |
| 3.2 | Screen Layouts & Report Layouts |  |
| 3.3 | Event Table |  |
| 3.4 | Use Case Diagram and Use Case Description |  |
| 3.5 | Entity Relationship Diagram |  |
| 3.6 | Activity Diagram |  |
| 3.7 | Class Diagram |  |
| 3.8 | Object Diagram |  |
| 3.9 | Sequence Diagram |  |
| 3.10 | Collaboration Diagram |  |
| 3.11 | State Chart Diagram |  |
| 3.12 | Data Flow Diagram(DFD) |  |
| 4 | System Design |  |
| 4.1 | Converting ERD to Tables |  |
| 5 | System Coding |  |
| 5.1 | List of tables with attributes and constraints |  |
| 5.2 | Program Description |  |
| 6 | System Testing |  |
| 6.1 | Test Cases |  |
| 7 | Validations |  |
| 8 | Future Enhancement |  |
| 9 | Conclusion |  |
| 10 | References and Bibliography |  |

Organization overview

**SYNOPSIS**

**TITLE**: “Carsalon”

**INTRODUCTION:-**

It Is An Website Or Web Application Which Can Be Used For Car SERVICING As Well Car MODIFICATION Of Customer’s Car(User).This Website Also Provides Comparison Between Cars Which Helps User While Buying A New Car. It Provides Reviews And Ratings On New Cars. There’s also an Option of Selling As well As Buying Used Car. This Would Be Certified by Our Garage Experts. There Are Registered Garages To Provide Services Which Are Mentioned All Around The City.

**PROBLEM STATEMENT:-**

There Is No Such Website Which Provides Modification of Customization Facility Service Centers Do Not Provide Pickup and Drop Facility

People Find It Difficult to Locate Garage’s That Provide Customization with Unique Ideas and Less Quantity of Parts Which Are Not Unique

**TECHNOLOGY:-**

* + - * Software: Word Press
      * Frontend: Html Design, Php
      * Backend : MySQL

**CURRENT SYSTEM:-**

In Today’s Era There Are Websites Which Provides Review On New Cars, Buying Selling of Used Cars but All of Them Are Not in One Website. Modification And Servicing Facility Is Still Not Available Online. For Which User Has To Visit Specialization Shops Which Are Very Rare In Our Country Because Of It They Are Expensive And Also It Consumes More Time Of User.

**PROPOSED SYSTEM:-**

“THE GARAGE” Is a Solution to Overcome the Current System Problems. It Is An Website Or Web Application Which Can Be Used For Car SERVICING As Well Car MODIFICATION Of Customer’s Car(User).This Website Also Provides Comparison Between Cars Which Helps User While Buying A New Car. It Provides Reviews And Ratings On New Cars. There’s also an Option of Selling As well As Buying Used Car. This Would Be Certified By Our Garage Experts. There Are Registered Garages to Provide Services Which are Mentioned All around the City.

**MODULES:-**

* User and Administrator Login
* Garage System

**Feasibility Study**

An important outcome of preliminary investigation is the determination that the proposed system is feasible. Feasibility is measure of how beneficial or practical the development of an information system will be to a shop.

There are following aspects in feasibility study

a. Operational Feasibility

b. Economical Feasibility

c. Technical Feasibility

**a. Operational Feasibility:-**

Operational feasibility study involves the operational evaluation organizational impact and development performance etc.

1. User friendliness of the proposed system makes the user to use the system without any error.
2. Manual of the proposed system gives complete idea about the system designed to the user. It also helps in maintenance of the system.
3. System will provide timely information and status of various activities to the management. It will facilitate the usage of the system.

**b. Technical Feasibility: -**

As the software is going to developed in vb.net so it can give a much more user friendly as compare to others. Since vb.net an event drove programming. It comprises of graphical display which allows user to work much more freely with the software. It will support printer too for providing customer with quotation or bill. If require internet connection can also be implemented. If user needs it.

**C.Economical Feasibility: -**

Php and Html software is easily available in the market and does not require any long term installation process. Paper cost can be reduced while the transaction details are maintained in the system. It is also an attraction

**ABOUT THE TECHNOLOGY:**

**PHP and HTML:**

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

* PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
* PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
* It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
* PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
* PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
* PHP is forgiving: PHP language tries to be as forgiving as possible.
* PHP Syntax is C-Like.

## Common uses of PHP:

* PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
* PHP can handle forms, i.e. gather data from files, save data to a file, thru email you can send data, return data to the user.
* You add, delete and modify elements within your database thru PHP.
* Access cookies variables and set cookies.
* Using PHP, you can restrict users to access some pages of your website.
* It can encrypt data.

## Characteristics of PHP

Five important characteristics make PHP's practical nature possible −

* Simplicity
* Efficiency
* Security
* Flexibility
* Familiarity

**HTML:**

HTML stands for **H**yper**t**ext **M**arkup **L**anguage, and it is the most widely used language to write Web Pages.

* **Hypertext** refers to the way in which Web pages (HTML documents) are linked together. Thus the links available on a webpage are called Hypertext.
* As its name suggests, HTML is a **Markup Language** which means you use HTML to simply "mark up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

## The <! DOCTYPE> Declaration

The <!DOCTYPE> declaration tag is used by the web browser to understand the version of the HTML used in the document. Current version of HTML is 5 and it makes use of the following declaration:

<!DOCTYPE html>

There are many other declaration types which can be used in HTML document depending on what version of HTML is being used. We will see more details on this while discussing <!DOCTYPE...> tag along with other HTML tags.

**Wordpress:**

**Wordpress** is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software) [content management system](https://en.wikipedia.org/wiki/Content_management_system) (CMS) based on [PHP](https://en.wikipedia.org/wiki/PHP) and [MYSQL](https://en.wikipedia.org/wiki/MySQL). Wordpress is installed on a [web server](https://en.wikipedia.org/wiki/Web_server), which either is part of an [Internet hosting service](https://en.wikipedia.org/wiki/Internet_hosting_service) or is a [network host](https://en.wikipedia.org/wiki/Host_(network)) itself; the first case may be on a service like[WordPress.com](https://en.wikipedia.org/wiki/WordPress.com), for example, and the second case is a computer running the software package Wordpress.org An example of the second case is a local computer configured to act as its own web server hosting Wordpress for single-user testing or learning purposes. Features include a plug-in architecture and a template system. Wordpress was used by more than 23.3% of the top 10 million websites as of January 2015. Wordpress is the most popular blogging system in use on the [Web](https://en.wikipedia.org/wiki/World_Wide_Web), at more than 60 million websites.

**Database (MY SQL Server) - An Overview:**

## MySQL Database:

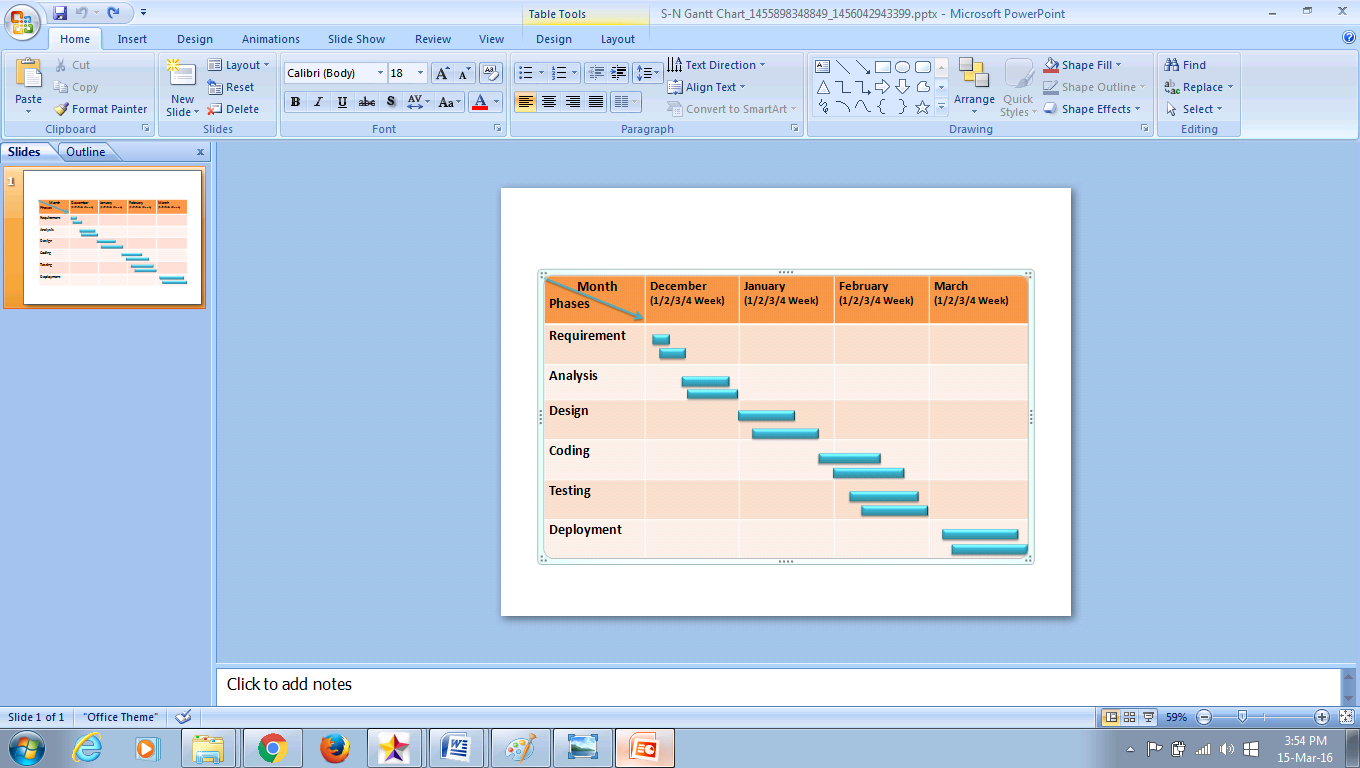
MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
* MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

|  |
| --- |
| **STAKEHOLDERS:** |
| Stakeholders are those persons who are successfully involved in completion of the project at the  Specified time period. Different stakeholders included in the completion of the project are :  The Client or Administrator :  He is the one for whom the project is being made.  He is successfully engaged for the completion of the software as soon as possible.  The Technical Staff:  They are the people who provide guidance in area of programming language to the programmer.  The guidance is given to the student or the programmer so that he can complete the given project  Properly.  The Creator or Programmer:  The student itself is the programmer who wants the project to be completed in the specified time.  He is the main stakeholder for the project. |

Gantt chart

A **Gantt chart** is a type of [bar chart](https://en.wikipedia.org/wiki/Bar_chart), adapted by [Karol Adamiecki](https://en.wikipedia.org/wiki/Karol_Adamiecki) in 1896 and independently by [Henry Gantt](https://en.wikipedia.org/wiki/Henry_Gantt) in the 1910s,[[1]](https://en.wikipedia.org/wiki/Gantt_chart#cite_note-1)that illustrates a [project schedule](https://en.wikipedia.org/wiki/Schedule_(project_management)). Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a [project](https://en.wikipedia.org/wiki/Project). Terminal elements and summary elements comprise the [work breakdown structure](https://en.wikipedia.org/wiki/Work_breakdown_structure) of the project. Modern Gantt charts also show the [dependency](https://en.wikipedia.org/wiki/Dependency_(project_management)) (i.e., precedence network) relationships between activities.

****

System Analysis

SYSTEM ANALYSIS

Fact finding techniques:

**Interview and Questionnaire**

The first step in developing a database system is to clearly define the **mission statement.** For the database project, which defines the major aims of the database system? Once the mission statement is defined, the next activity involves identifying the **mission Objectives**, which should identify the particular tasks that the database must support.

We started the Interview with the manager of the organization City Fashion. Following is some part of important conversation between us:

* **What is the purpose of your company?**
* **We offer a wide range of high quality products i.e Car accessories,Body parts,graphics and other parts of cars including performance tools and also a service for cars and bikes**
* **Why do you feel that you need a database?**
* **With the increasing business the level and volume of data is also increasing. As a result it very difficult to handle this huge data in files. Hence we feel a need for a computer system that could handle our data.**

**Moreover it is very hard and time consuming to retrieve data from files.**

* **What kinds of tasks does your company perform in a typical day?**
* **Our company is a Servicing and Modification Store. We use to sell our customers which we sell, we create an accounts of the products, which is reacquired for the further study of the market and the trends of the market.**
* **What kinds of data do you work with?**
* **We mostly have to maintain the orders from the customers. Our main data involves the sales and the Returns which has to be maintained carefully as it is very important. Last we also have data about our employees, customer, and products.**
* **What types of reports do you use?**
* **Mainly, we generate Appointment Summary reports.**
* **Does any operator use the system?**
* **Yes**
* **How many employees are there in the institution?**
* **60+ Employees**
* **At present how the work is maintained in the institution?**
* **FILES**
* **Does the institution give Salary through Cheque / DD also?**
* **No**
* **Do you want any software that maintains your daily work very easily and without any stress?**
* **Yes**

**Software Process**

A software process is **a set of activities, together with ordering constraints among them, such that if the activities are performed properly and in accordance with the ordering constraints, the desired result is produced**.

**2.1 Software Development Process Model:**

•In the software development process I **focus** on the activity directly related to production of the software, for example, **design, coding, testing**.

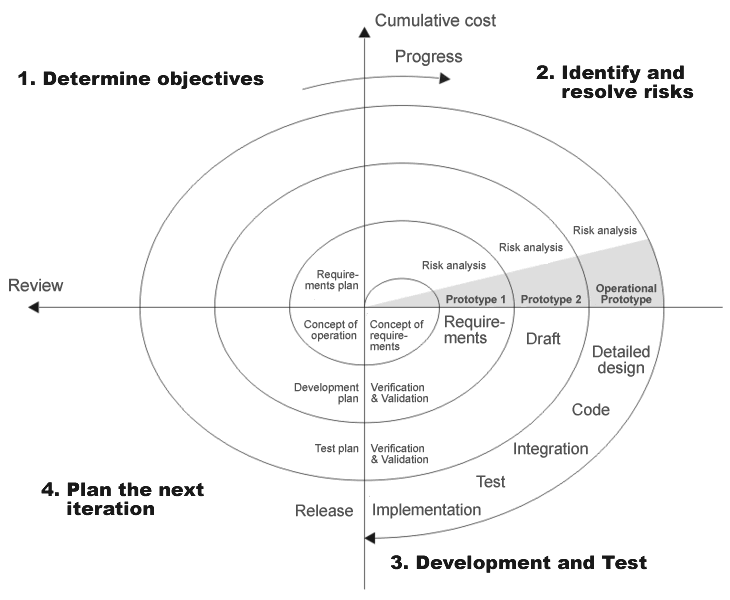
• As the development process **specifies the major development and quality control activities** that need to be performed in the project, the development process really forms the core of the software process.

• The **management process** is decided based on the development process.

• Due to the importance of the development process, **the spiral model** has been proposed.

**Spiral Model**

Figure:



Introduction:

• The spiral model is another **iterative model** that has been proposed.

• As the name suggests, the **activities** in this model can be organized like a spiral that has **manycycles** as shown in Figure.

• Each cycle in the spiral **begins with the identification of objectives** for that cycle, the different.alternatives that are possible **for achieving the objectives and the constraints** that exist.

• The next step in the cycle is to **evaluate these different alternatives** based on the objectives and constraints.

• The focus of evaluation in this step is based on the **risk perception** for the project.

• The next step is **to develop strategies that resolve the uncertainties and risks**. This step may in-volve activities such as benchmarking, simulation, and prototyping.

• Next, the software is developed, keeping in mind the risks.

• Finally the next stage is planned.

Generally, a version of the product is released that contains some capability.

• The first version contains some core capability and more features are added to later versions.

Use:

• One **effective use** of the spiral model is often seen in product development, in which the developers themselves provide the specifications and therefore hava a lot of control on which specifications go in the system and which stay out.

The main reason:

• As businesses are changing very rapidly today, they never really know the **complete** requirements for the software and there is a need to constantly add new capabilities to the software to adapt

the business to changing situations.

• Furthermore, customers do not want to **invest** too much for a long time without seeing returns.

• In the current business scenario, it is **preferable** to see returns continuously of the investment

made. The spiral (another iterative) model **permits** this after each iteration some working software is delivered. This approach to software development is now **widely** used.

Weaknesses:

• It can have **planning overhead**.

• Cost may increase as work done in one iteration may have to be **undone** later.

• System architecture and structure may suffer as frequent **changes** are made.

Types of project:

• In a customized software development, where **the client has to provide and approve the specifications**, this process model is becoming extremely popular.

EVENT TABLES

**Event Table:**

● It is implementing system (one or more) event in a tabular format.

● Event table consists of following columns:

**-** Event

**-** Trigger

**-** Source

**-** Use Case

**-** Response

**-** Destination

● **Event:**

**-** It is action affecting the system. Or

**-** It is one that causes the system to do some thing.

**-** Events are external, temporal and state event.

**-** For example : Customer now an item, Customer order an item, etc.

● **Trigger:**

**-** How the system know that the system occurs?

**-** For example : Item enquiry, New order, etc.

● **Source:**

**-** How is initiating that event (always external)?

**-** For example : Customer, Operator, Administrator, etc.

● **Use Case:**

**-** What does the system do that system occurs?

**-** For example : Look up item availability, Place new order, etc.

● **Response:**

**-** It is availability output.

**-** For example : Item details, Invoice, etc.

● **Destination:**

**-** Whom that going?

**-** For example : Customer, Operator, Administrator, etc.

USECASE DAIGRAM

**USECASE DIAGRAM:**

• Use Case Diagrams model the functionality of system by using Actors and Use Cases:

• Actor is a user of the system.

• Use cases are services or functions provided by the system to its users.

****

****

• Actor:

• An Actor can be described as follows:

• Actor is any entity that is external to the system and directly interacts with the system, thus deriving some benefit from the interaction.

• Actor can be a human being, a machine, or a software.

• Actor is a role that a particular user plays while interacting with the system.

• Examples of Actors are End-user (roles), External systems, and Externalpassive

objects (entities).

• Use Case:

• An Use Case can be defined as a set of activities performed within a system bya User.

• Each Use Case:

• describes one logical interaction between the Actor and the system.

• defines what has changed by the interaction.

• A Use Case diagram has the following elements:

• **Stick figure:** representing an Actor

• **Oval:** representing a Use Case

• **Association lines:** representing communication between Actors and Use Cases

Types of relationships between Use Cases are:

• Include

• Extend

Include relationship:

• «include» stereotype indicates that one use case “includes” the contents of another usecase.

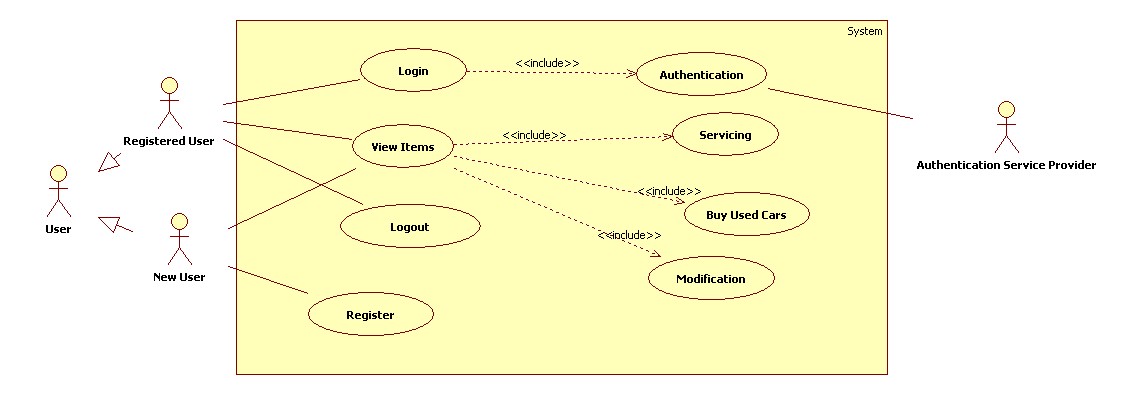
• Include relationship enables factoring out frequent, common behaviour.

• Extend relationship:

• «extend» stereotype indicates that one Use Case is “extended” by another UseCase.

• Extend relationship enables factoring out infrequent behaviour or errorconditions.

• Extend relationship represents optional behavior for a Use Case which will be required only under certain conditions.

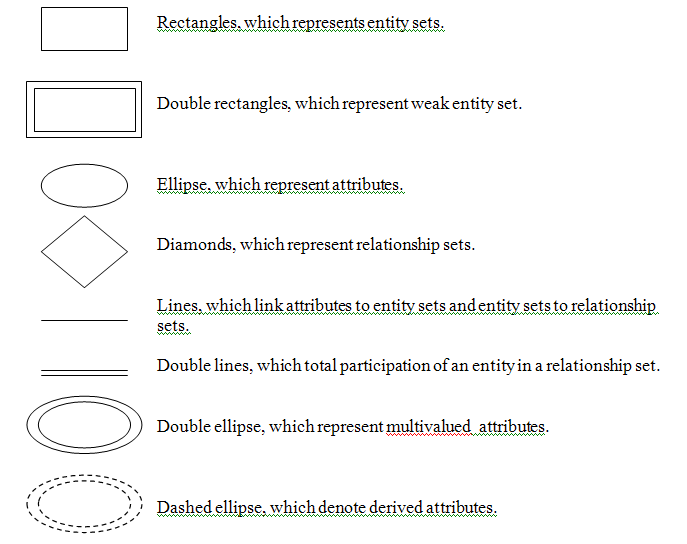


ENTITY RELATIONSHIP DIAGRAM

**ENTITY RELATIONSHIP (ER) DIAGRAM:**

An ER-Diagram (Entity Relationship Diagram) is a graphical model of the data needed by a system. This included the entities about which information is stored and the relationship among them. It is produced in structured analysis methods.

An **entity** is a “thing” or “object” in the real world that is distinguishable from all set of objects. An entity **set** is a set of the same type that share same properties, or attributes.E-R diagram consists of the following major components:



The traditional approach to system development places a great deal of emphasis on data storage requirements for the new system. Data storage requirements include the data entities, their attributes, and the relationships among the data entities. The model used to define the data storage requirements is called the Entity-Relationship Diagram.

**On the Entity-Relationship Diagram:**

* Rectangles represent data entities
* Lines connecting the rectangles show the relationships among data entities.
* Ellipse represents the attributes of the entity.

The traditional approach to system development places a great deal of emphasis on data storage requirements for the new system. Data storage requirements include the data entities, their attributes, and the relationships among the data entities. The model used to define the data storage requirements is called the Entity-Relationship Diagram.

Bill Statement

Recieve

Book

generate

Check schedule

User

Admin

Appointment

ACTIVITY DIAGRAM

**ACTIVITY DIAGRAM**

• An Activity Diagram has the following features:

• It resembles a flow chart.

• It illustrates the dynamic nature of a system by modeling the flow of controlfrom activity to activity.

• An activity can represent an operation on some class in the system thatresults in a change in the state of the system.

• Activity Diagrams are used:

• to model business processes.

• to model internal operation of a Use Case / method.

• to model work flows and computations.

• Notations for Activity Diagrams include:

• Activity (State)

• Name

• Sub-states

• State - Sub state relationship

• Actions

• Entry Action

• Exit Action

• Transition

• Start and End States

• Event and Guard Condition

• Internal activity

• In addition, there are notations for:

• Alternate Paths (Branching and Merging)

• Parallel Paths (Fork and Join)

• Swim lanes

• Alternate Paths:

• Alternate Paths are shown withthefollowing notations:

• Diamond representing a decisionwith alternate paths

• Guard conditions used to labelthe alternates

Parallel Paths are:

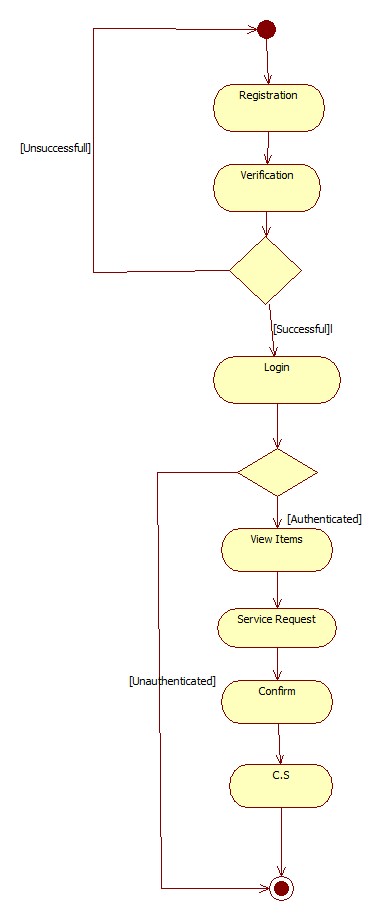
• Fork represents splitting of a singleflow of control into two or moreconcurrent flows of control.

• Join represents the synchronization oftwo or more flows of control into onesequential flow of control.

• Swim lanes:

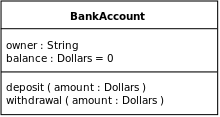
• Swim lanes are used for grouping therelated activities into columns.

• Each column or “Swim lane” denotesdistribution of responsibilities to be handled by different actors / parts of system.



CLASS DIAGRAM

The class diagram is the main building block of [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) modelling. It is used both for general [conceptual modelling](https://en.wikipedia.org/wiki/Conceptual_model) of the systematic of the application, and for detailed modelling translating the models into [programming code](https://en.wikipedia.org/wiki/Programming_code). Class diagrams can also be used for [data modeling](https://en.wikipedia.org/wiki/Data_modeling). The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

[](https://en.wikipedia.org/wiki/File:BankAccount1.svg)

A class with three compartments.

In the diagram, classes are represented with boxes that contain three compartments:

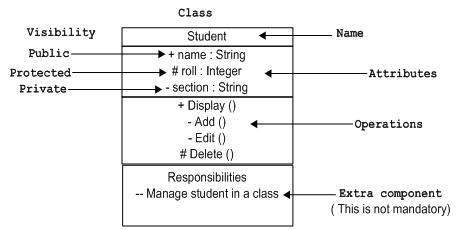
* The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
* The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.
* The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

In the design of a system, a number of classes are identified and grouped together in a class diagram that helps to determine the static relations between them. With detailed modelling, the classes of the conceptual design are often split into a number of subclasses.

### Class Notation:

UML *class* is represented by the diagram shown below. The diagram is divided into four parts.

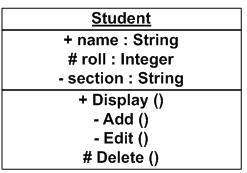
* The top section is used to name the class.
* The second one is used to show the attributes of the class.
* The third section is used to describe the operations performed by the class.
* The fourth section is optional to show any additional components.



Classes are used to represent objects. Objects can be anything having properties and responsibility.

### Object Notation:

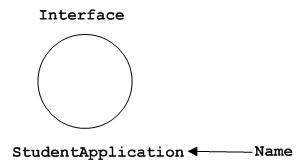
The *object* is represented in the same way as the class. The only difference is the *name* which is underlined as shown below.



As object is the actual implementation of a class which is known as the instance of a class. So it has the same usage as the class.

### Interface Notation:

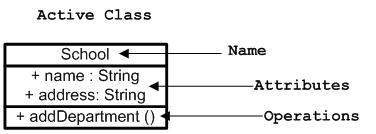
Interface is represented by a circle as shown below. It has a name which is generally written below the circle.



Interface is used to describe functionality without implementation. Interface is the just like a template where you define different functions not the implementation. When a class implements the interface it also implements the functionality as per the requirement.

### Active class Notation:

Active class looks similar to a class with a solid border. Active class is generally used to describe concurrent behaviour of a system.



Active class is used to represent concurrency in a system.

## Relationships

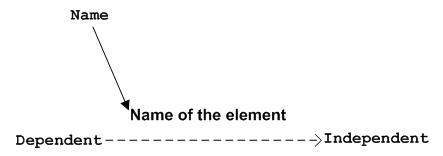
A model is not complete unless the relationships between elements are described properly. The *Relationship* gives a proper meaning to an UML model. Following are the different types of relationships available in UML.

* Dependency
* Association
* Generalization
* Extensibility

### Dependency Notation:

Dependency is an important aspect in UML elements. It describes the dependent elements and the direction of dependency.

Dependency is represented by a dotted arrow as shown below. The arrow head represents the independent element and the other end the dependent element.

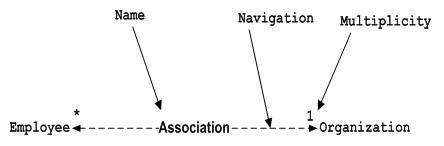


Dependency is used to represent dependency between two elements of a system.

### Association Notation:

Association describes how the elements in an UML diagram are associated. In simple word it describes how many elements are taking part in an interaction.

Association is represented by a dotted line with (without) arrows on both sides. The two ends represent two associated elements as shown below. The multiplicity is also mentioned at the ends (1, \* etc) to show how many objects are associated.



Association is used to represent the relationship between two elements of a system.

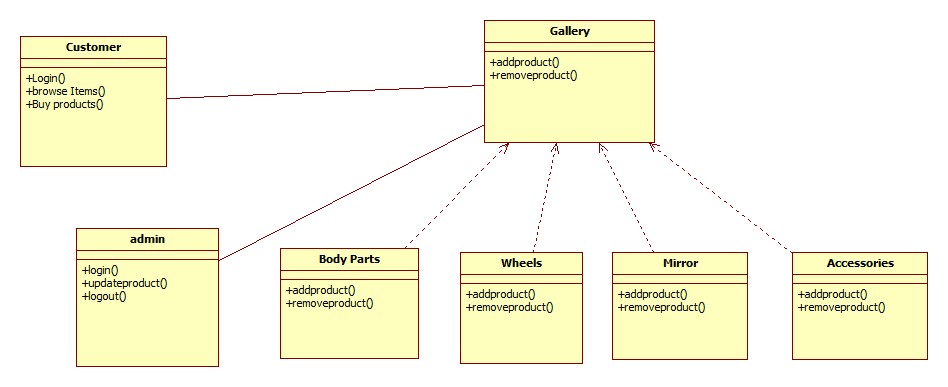
### Generalization Notation:

Generalization describes the inheritance relationship of the object oriented world. It is parent and child relationship.

Generalization is represented by an arrow with hollow arrow head as shown below. One end represents the parent element and the other end child element.

Generalization Notation

Generalization is used to describe parent-child relationship of two elements of a system.



**OBJECT DAIGRAM**

**Object Diagrams:**

• Object Diagrams describe the “static structure” of a system at a particular time.

• Objects and Links are the constituents of Object Diagrams.

• Object Diagrams are different from Class Diagrams

• This is because many objects of same class may exist in the Object Diagram.

• Object Diagrams can be used:

• to test Class Diagrams for accuracy

• to verify system performance at given instance

• to optimize performance (especially useful for server objects)

Admin

+EmailId:Admin@123

+password: root

+manage Appointment ()

User3

+Username: hgf

+Password: 123hd

+Book Appointment ()

User2

+Username: xys

+Password: 123dd

+Book Appointment ()

User1

+Username: abc

+Password: 123

+Book Appointment ()

SEQUENCE DIAGRAM

**SEQUENCE DIAGRAM:**

• Sequence Diagram:

• A Sequence diagram describes interactions among classes in terms of an“exchange of messages over time”.

• Some of the facts related to Sequence Diagrams are:

• Sequence Diagrams are used to depict the time sequence of messagesexchanged between objects.

• Messages can correspond to operation on class or a event trigger.

• Notations of a Sequence diagram include:

• **LifeLine:** It is a vertical dashed line that represents the “lifetime” of anobject.

• **Arrows:** They indicate flow of messages between objects.

• **Activation:** It is a thin rectangle showing period of time, during which anobject is performing an action.

• Direction of arrows:

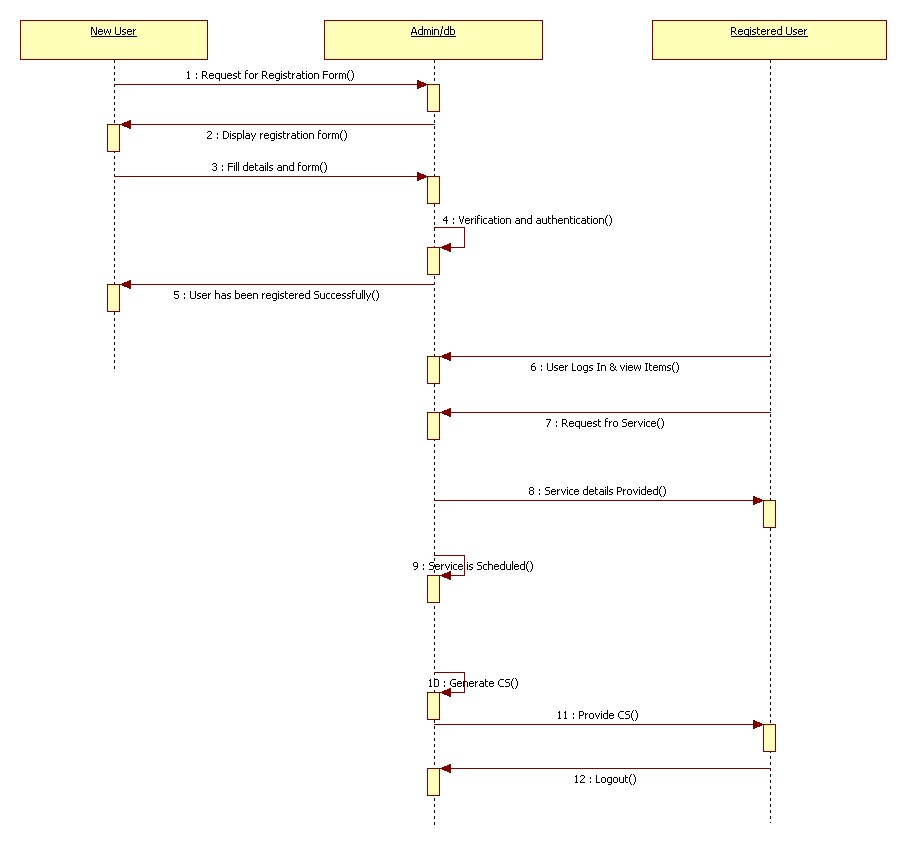
• Direction indicates which object’smethod is being called by whom.

• A circulating arrow on the ObjectLifeline is for a self method– calledwithin the object by itself.

• Branch Conditions:

• Branch Conditions are depicted as“Guard Conditions” within SquareBrackets.

• Repetition or Looping depicted as arectangle, with condition for exitingthe loop placed at the bottom leftcorner.

****

COLLABORATION

DIAGRAM

• A Collaboration Diagram describes interactions among objects in terms ofmessages.

• A Collaboration Diagram:

• Emphasizes on the organization of the objects that participate in theinteraction.

• Is semantically similar to Sequence Diagram.

• While Sequence Diagrams show exchange of messages based on time,Collaboration Diagrams are spatial representations.

• Sequence Numbers can be used for numbering messages to indicate theirtime ordering.

Login

Home

Admin

Feedback

About Us

Products/Servicing

STATE CHART DIAGRAM

• A State Chart Diagram describes the dynamic behaviour of a system inresponse to external stimuli.

• A State Chart Diagram helps:

• to model dynamic behaviour of objects based on states.

• to model reactive objects, whose states are triggered by specific events.

• to describe passive objects, which go through several distinct phases duringtheir life time.

• State Chart Diagrams describe how the objects work:

• Each object is in a given initial state when it is created

• Object may change states (transition) to other states based on some stimuli

• State is the condition of an object

• Transitions indicate relation between the conditions

• Parts of a State are:

• **Name**: Unique name identifying the state

• **Sub-states**: Set of “disjoint sub-states” or “concurrent sub-states”.

• **State – Sub-state relationship**: Useful to understand the modelling of complexbehaviours.

• **Entry action**: An action that happens as a result of transition into a state.

• **Exit action**: An action that has to happen immediately before a state change.

• **Internal activity**: Activities that occur within an object while it is in aparticular state.

• There are some pseudo-states that are used in a State Chart Diagram:

• Start State:

• It is the default start point for an object state.

• Each State Chart Diagram should have exactly one Start State.



• End State:

• It indicates Completion State.

• It may or may not exist for a State Chart Diagram.

• When an object is destroyed, the state is no longer considered.

**Drawing11**

• Transition includes:

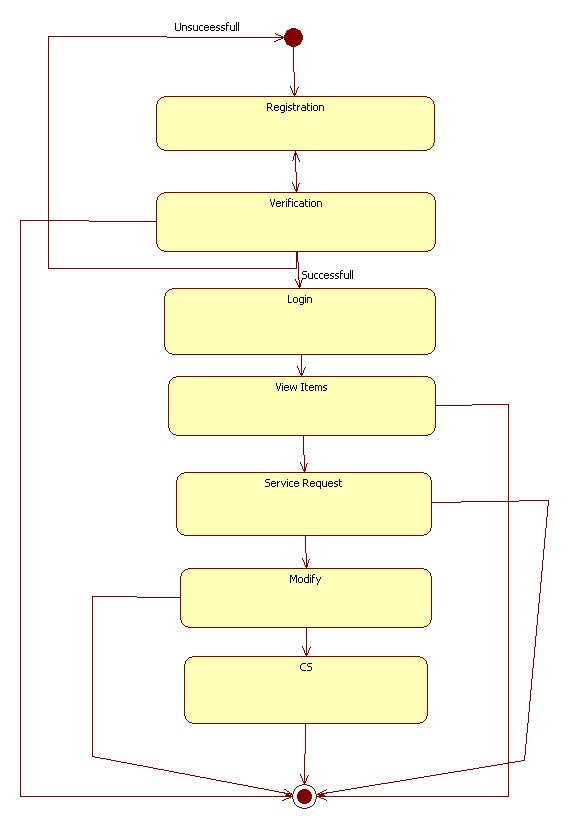
• **Event**: The “trigger”

• **Guard**: A logical condition which returns true or false. It is evaluated at thetime of event

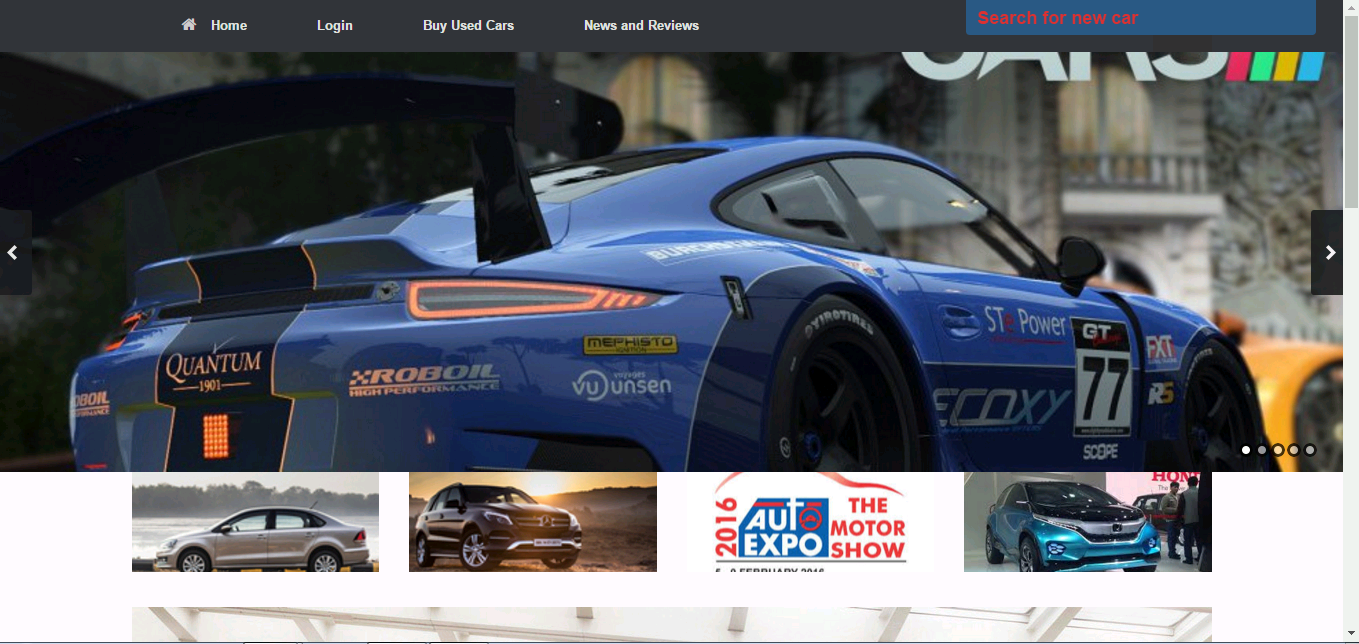
triggering.

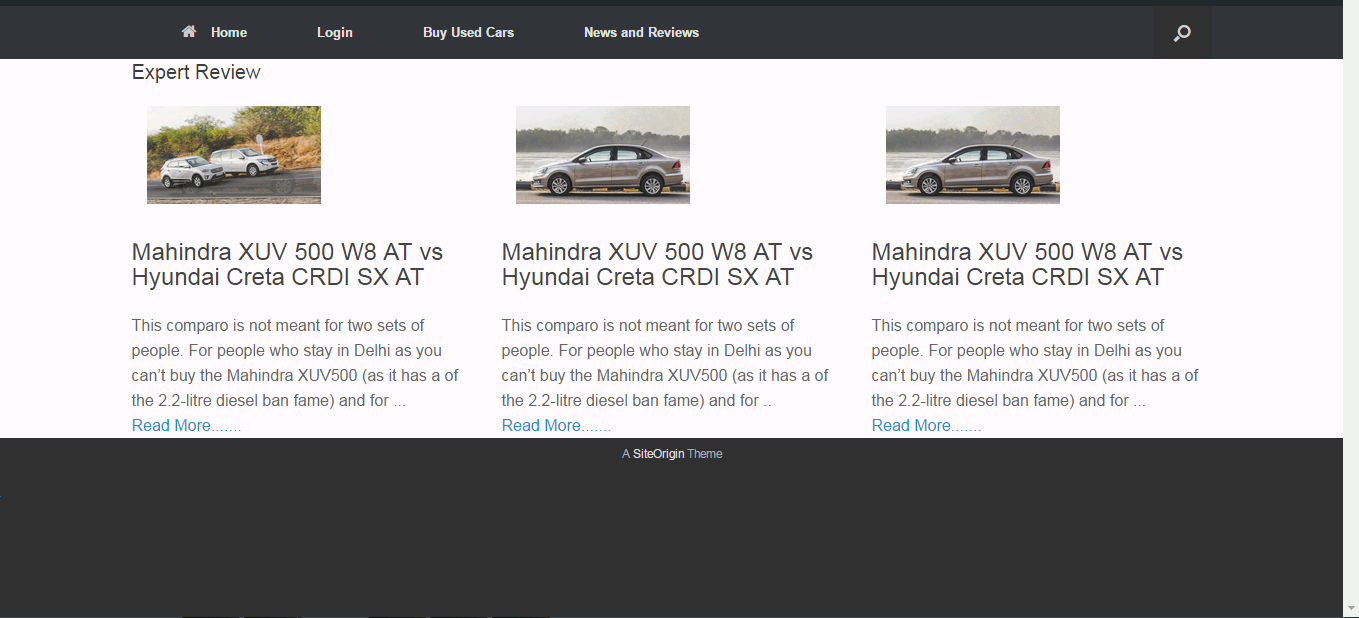
• **Action**: Gets executed when transition is fired.

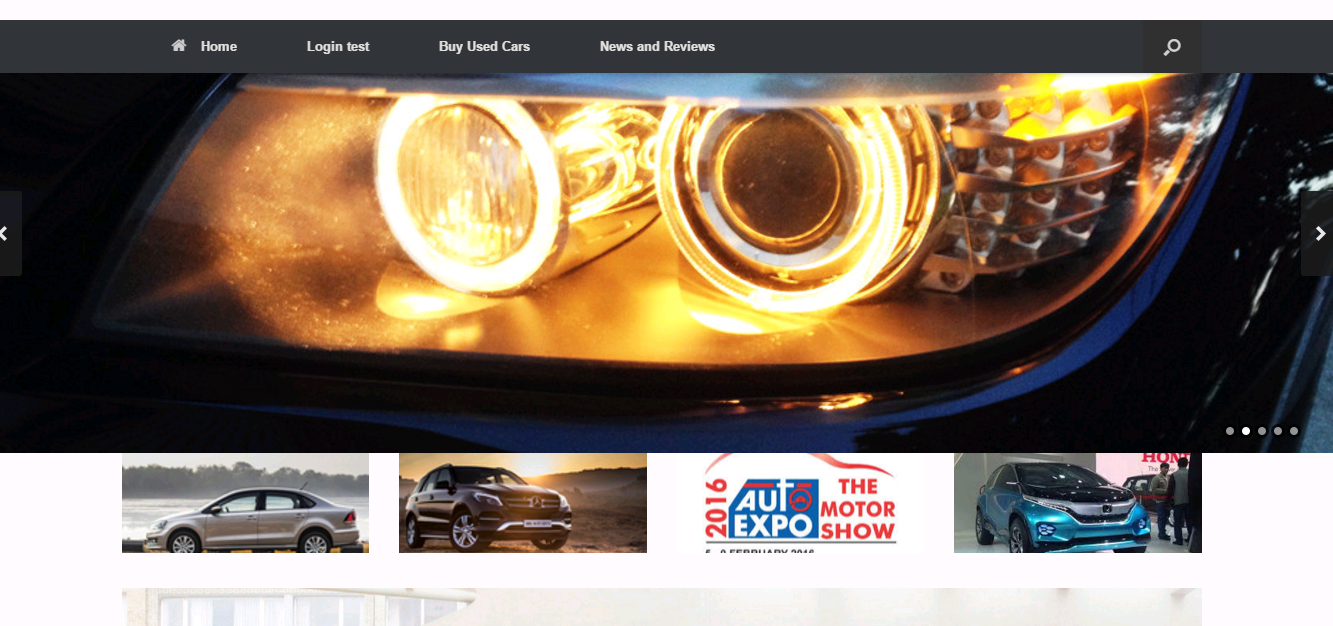
**Event [Guard]/ Action**

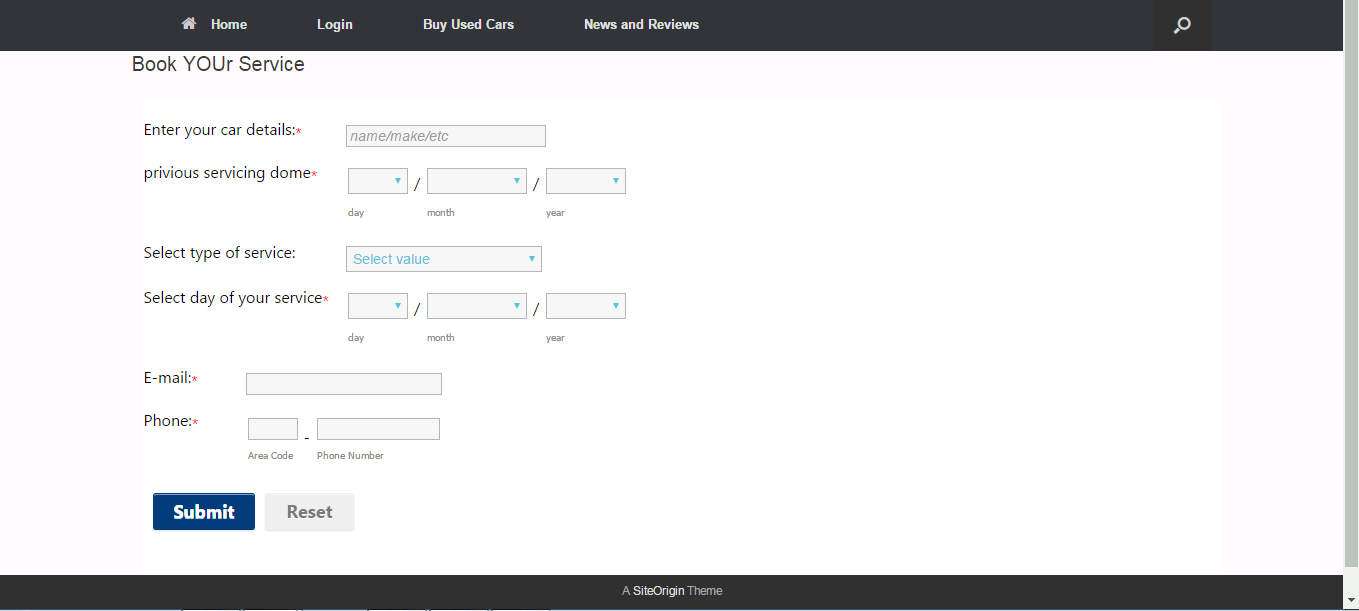


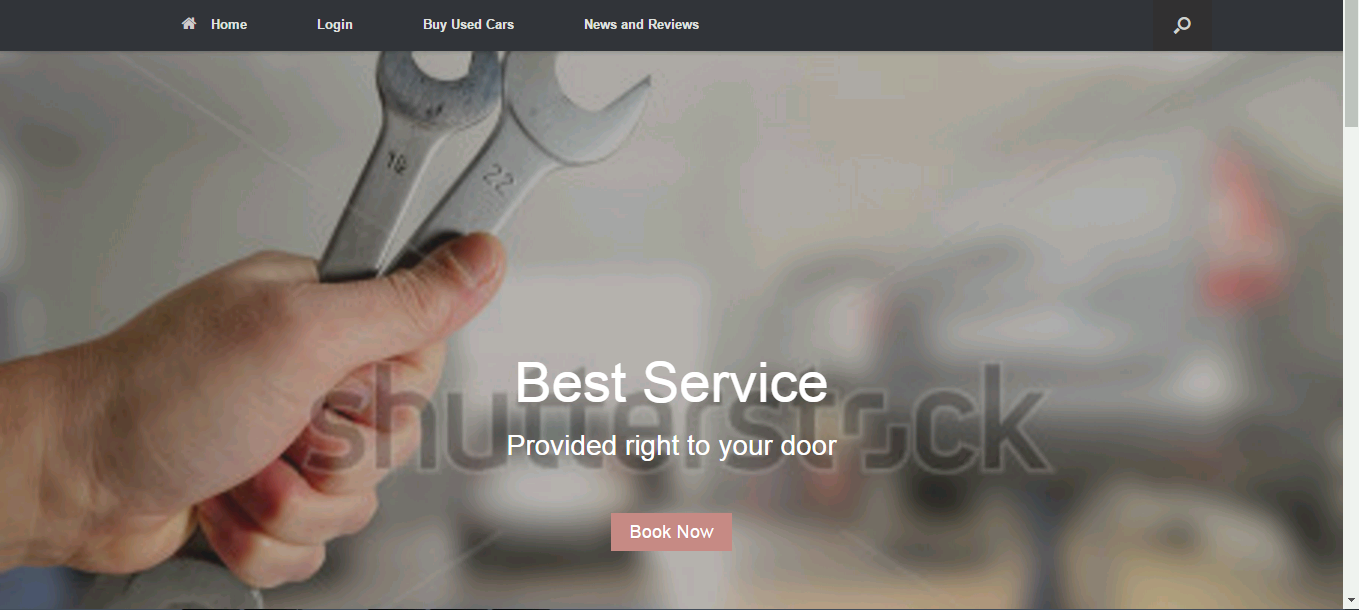
**SCREEN LAYOUTS & REPORT LAYOUTS**

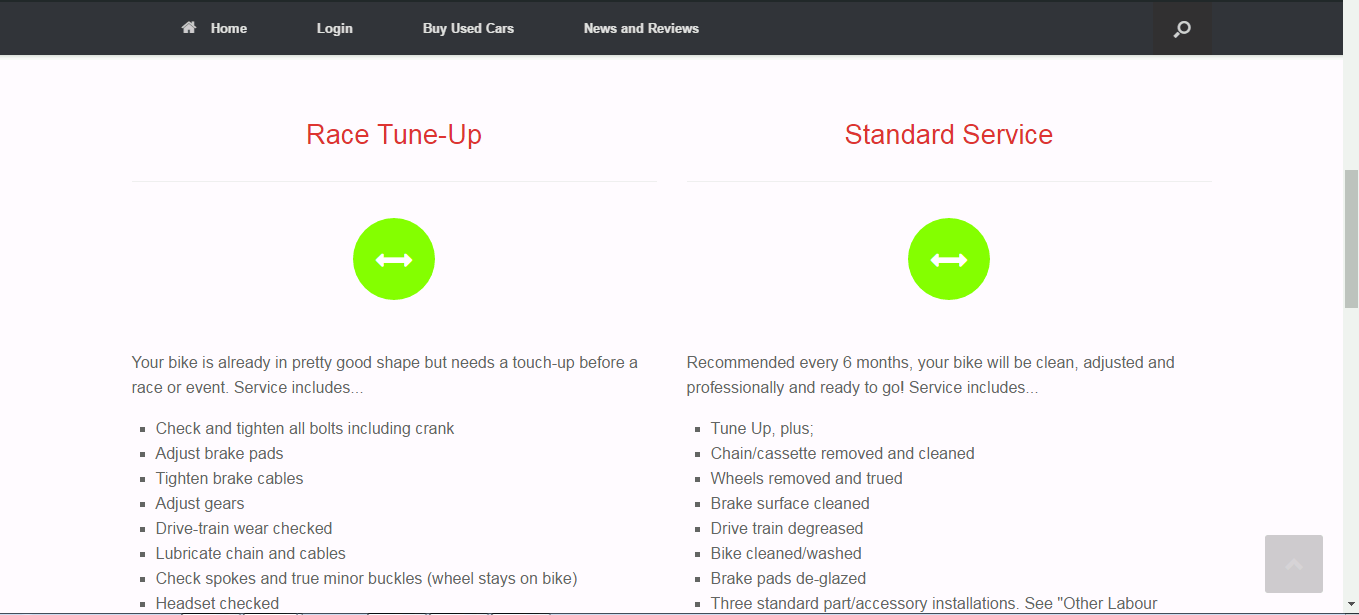
****

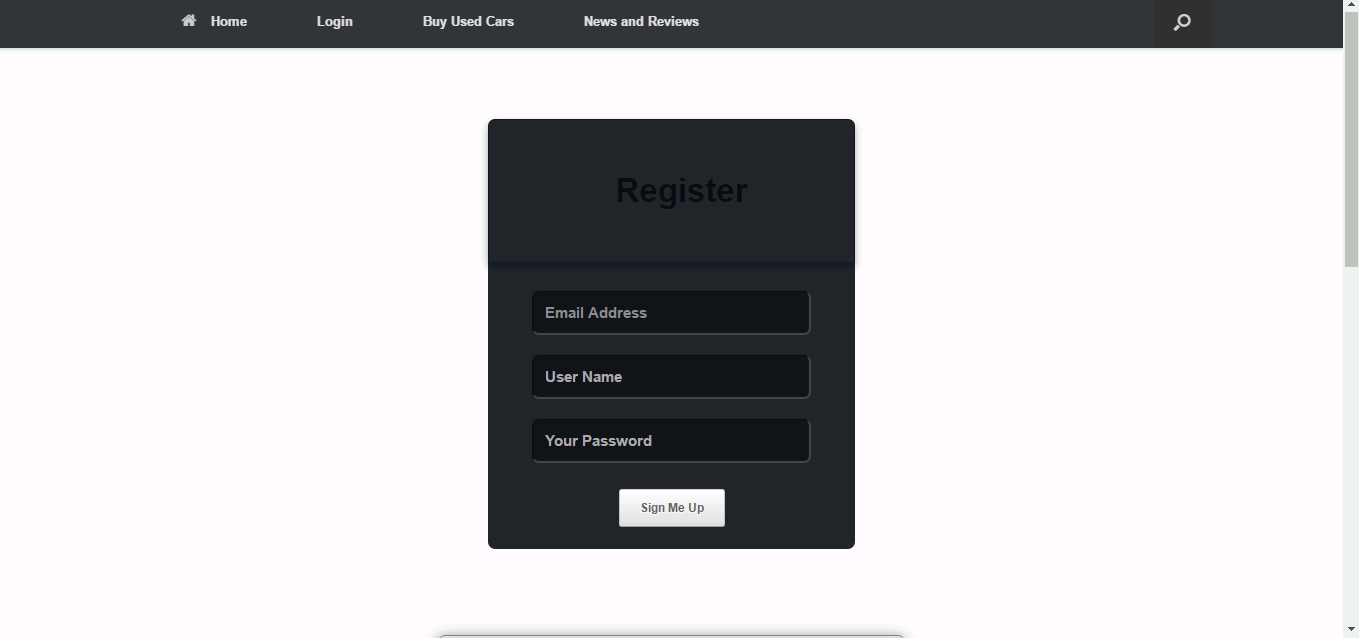
****

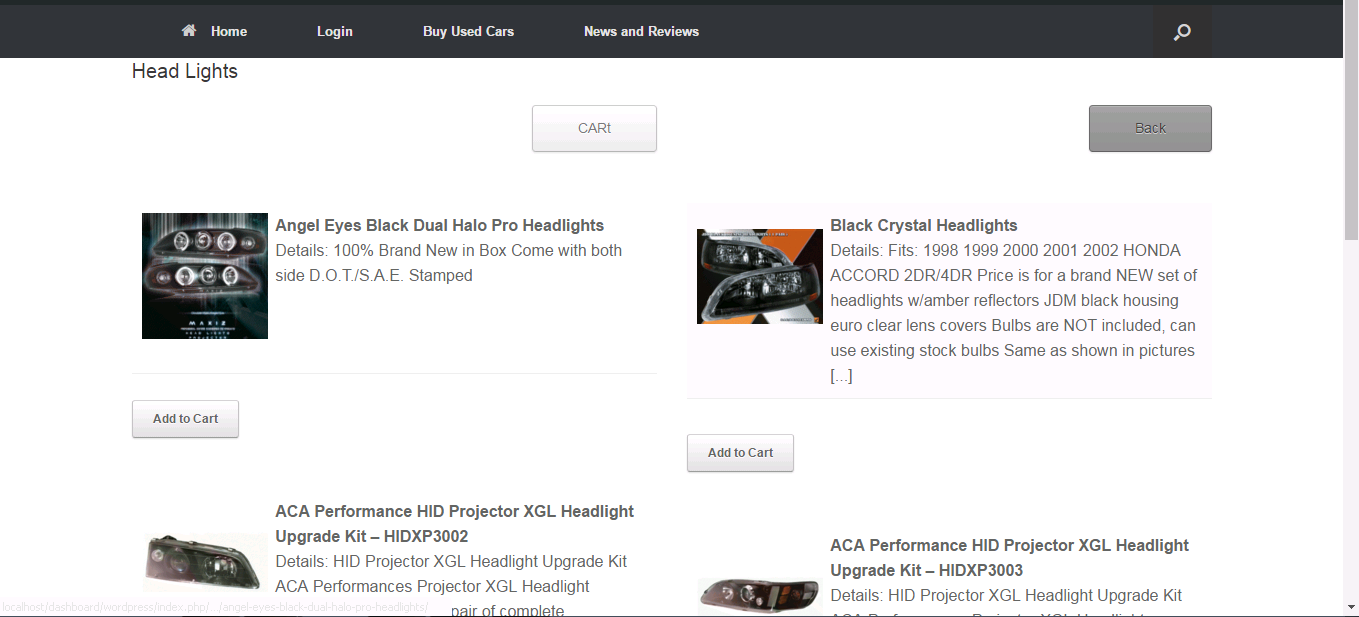
****

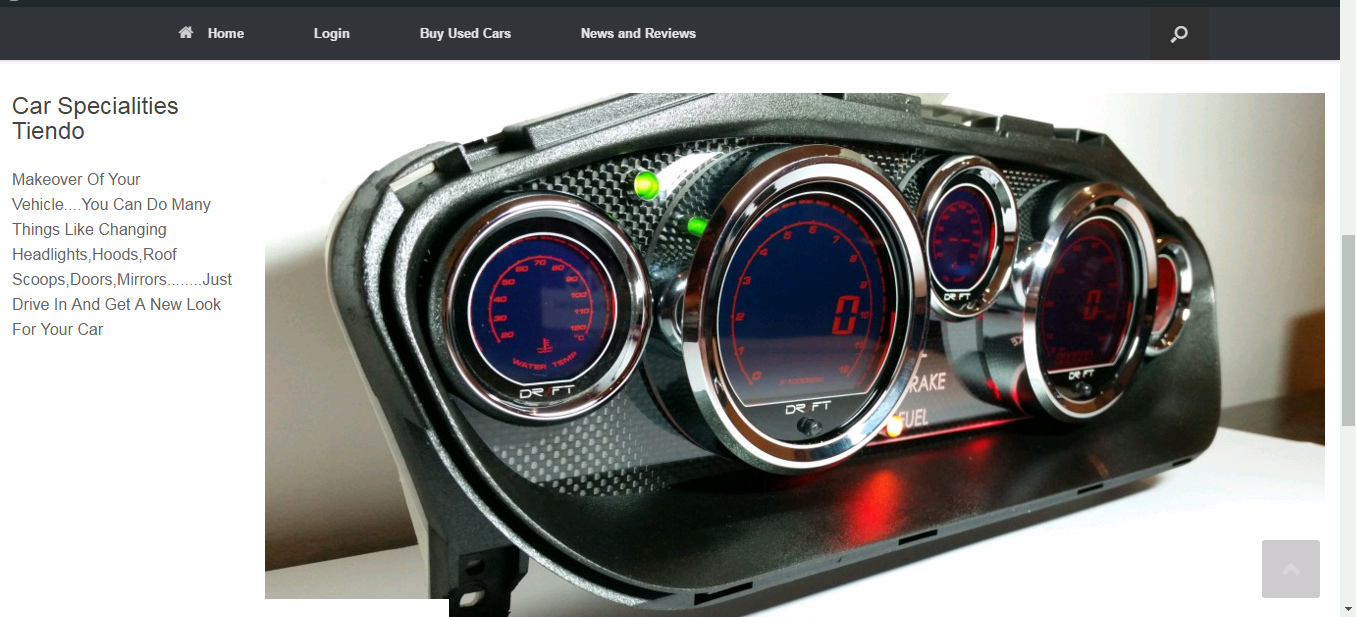
****

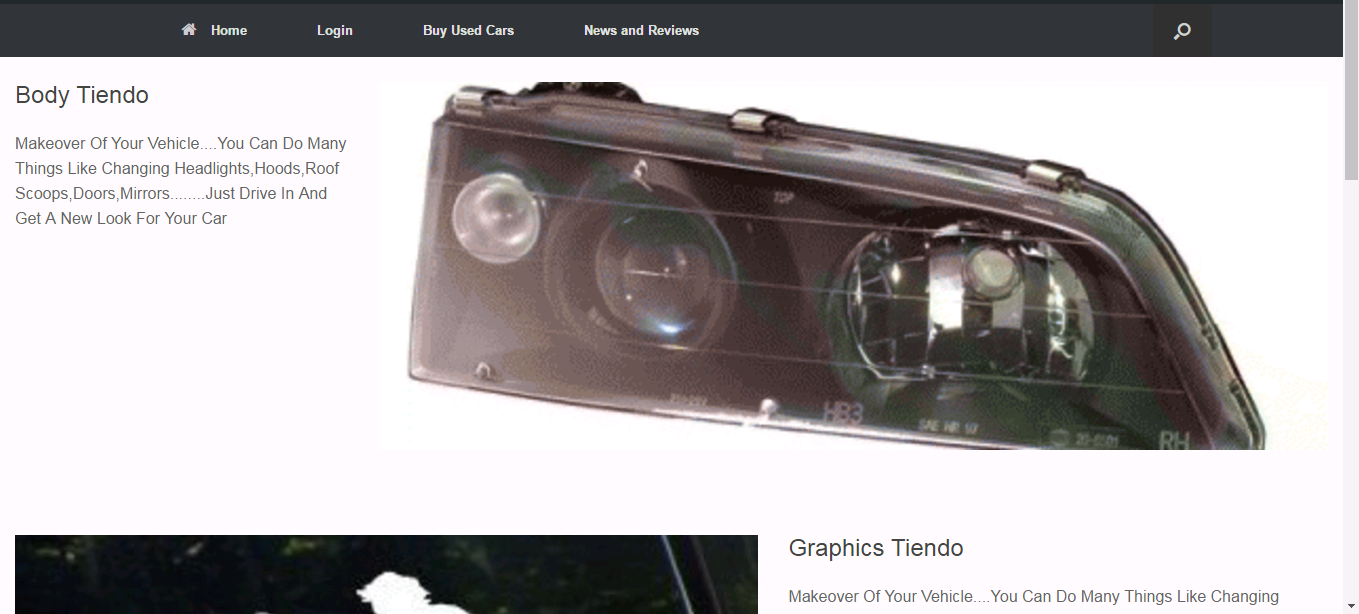
****

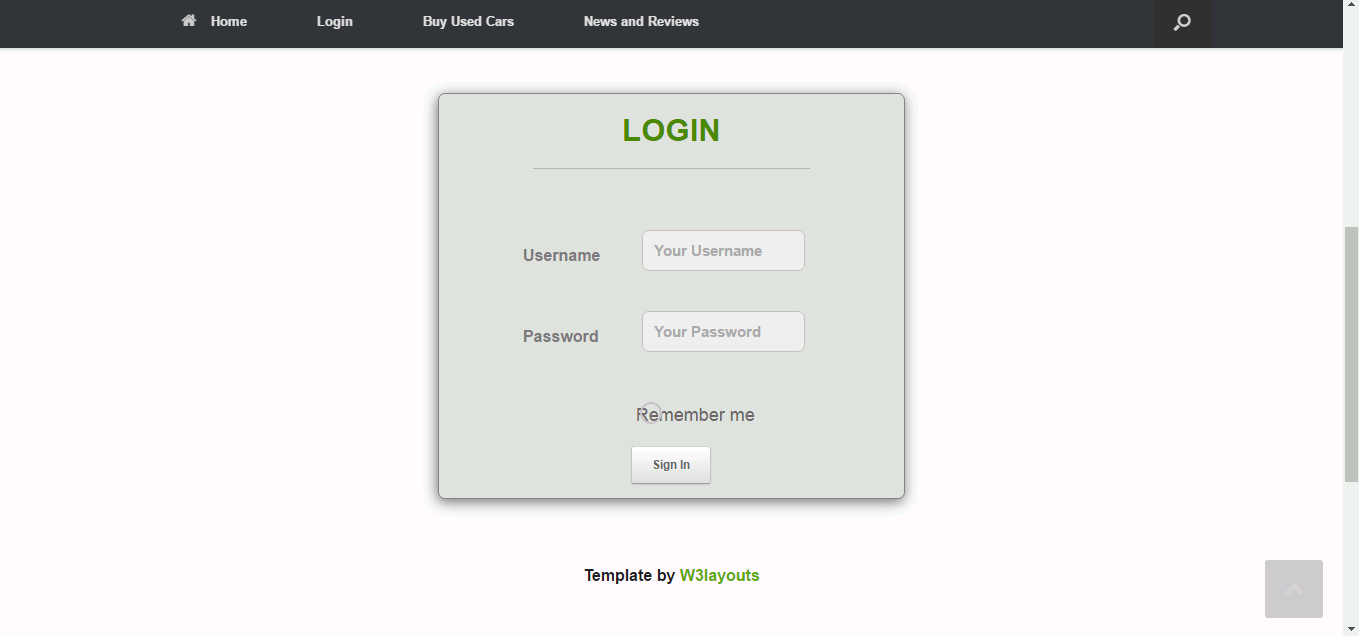
****

****

****

****

****

****

**SYSTEM DESIGN**

**DEPLOYMENT DIAGRAM:**

• Deployment Diagrams depict the physical resources in a system includingnodes, components, and connections.

• Deployment Diagrams help to:

• capture the hardware topology.

• model the distributed systems.

• Node:

• A Node represents:

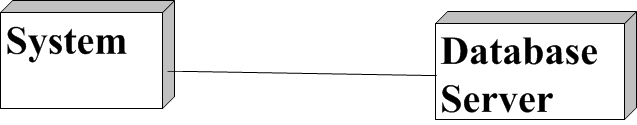
•a computational resource

•a physical resource that executescode components

• A set of components are allocated to a Node.

• Communication Path represents physical connections between nodes.

**Deployment Diagram of CARSALON:**

****

**Data Flow Diagram:**

A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then "exploded" to show more detail of the system being modeled.

Data flow diagrams (DFDs) are one of the three essential perspectives of SSADM. With a dataflow diagram, users are able to visualize how the system will operate, what the system will accomplish and how the system will be implemented. Old system dataflow diagrams can be drawn up and compared with the new systems dataflow diagrams to draw comparisons to implement a more efficient system. Dataflow diagrams can be used to provide the end user with a physical idea of where the data they input, ultimately has an effect upon the structure of the whole system from order to dispatch to restock how any system is developed can be determined through a dataflow diagram.

**There are several common modeling rules that I follow when creating DFDs:**

1. All processes must have at least one data flow in and one data flow out.
2. All processes should modify the incoming data, producing new forms of outgoing data.
3. Each data store must be involved with at least one data flow.
4. Each external entity must be involved with at least one data flow.
5. A data flow must be attached to at least one process.

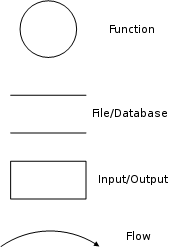
**Data flow diagram notation**:

External Entities/Terminators  are outside of the system being modeled. Terminators represent where information comes from and where it goes. Processes modify the inputs in the process of generating the outputs Data Stores represent a place in the process where data comes to rest. A DFD does not say anything about the relative timing of the processes, so a data store might be a place to accumulate data over a year for the annual accounting process.

Data Flows are how data moves between terminators, processes, and

1. data stores (those that cross the system boundary are known as IO or Input Output Discretions).
2. A data flow diagram (DFD) is a graphical system model that shows all of the main requirements for an information system in one diagram: inputs and outputs, processes, and data storage. Everyone working on a development project can see all aspects of the system working together at once with DFD. That is one reason for its popularity. The DFD is also easy to read because it is graphical model.
3. End Users, management, and all information systems workers typically can read and interpret the DFD with minimal training.

**The four components of a data flow diagram (DFD) are**

[](http://en.wikipedia.org/wiki/Image:DataFlowDiagram_Notation.png)

**Context Diagram**

A DFD (Data Flow Diagram) that summarizes all processing activity within the system in a single process symbol is called Context Diagram. A context Diagram is a DFD that describes the highest-level view of a system are shown in own diagram, with the whole system represented as one process.

The context diagram provides a good overview of the scope of the system, showing the system in “context” but it does not show any detail about the processing that takes place inside the system. The context diagram for the security company management system is shown below. The inputs & outputs of the store are shown in the fig. The diagram shows the external entities of the system & how the data that flows through the system. This helps in determining the basic functionalities.

A context diagram is a top level (also known as Level 0) data flow diagram. It only contains one process node (process 0) that generalizes the function of the entire system in relationship to external entities.

**SYSTEM TESTING**

Testing plays a critical role in quality assurance for software. Due to the limitations of the verification methods for the previous phases, design and requirements faults also appear in the code. Testing is used to detect these errors, in addition to the errors introduced during the coding phase.

Testing is a dynamic method for verification and validation, where the system to be tested is executed and the behavior of the system is observed. Due to This, testing observes the failure of the system, from which the presence of faults can be deduced. However, separate activities have to be performed to identify the faults.

**Approaches:**

**Black-Box Testing**

In the black-box testing, the internal logic of the system under testing is not considered and the test cases are decided from the specification or the requirements. It is often called functional testing. Equivalence class partitioning, boundary value analysis, and cause effecting graphing are examples of methods for selecting test cases for black-box testing. State–based testing is another approach in which the system is modeled as a state machine and then this model is used to select test cases using some transition or path based coverage criteria. State–based testing can also be viewed as grey-box testing in that it often requires more information than just the requirements.

**White-Box Testing**

In white-box testing, the test cases are decided entirely on the internal logic of the program or module being tested. The external specifications are not considered. Often a criterion is specified, but the procedure for selecting test case is left to the tester. The most common control flow-based criteria are statement coverage and branch coverage, and the common data flow-based criteria are All- doffs and all-uses. Mutation testing is another approach for the white-box testing that creates mutants of the original program by changing the original program.

The testing criterion is to kill all the mutants by having the mutant generate a different output from the original program.

**Validation Testing**

Checking to verify that appropriate values have been entered for a text box is called Validation. In this test, if user is trying to enter the invalid data in any field than it will prompt an Message Box showing the “ERROR”

There are following types of error :-

* java.sql.SQlException:[Microsoft][ODBC Driver Manager] Data Source name not found and no default driver specified,
* Please, enter create User Id and Password,
* Please, enter all fields, etc.

**Authentication Testing**

It is use to test that a login user is authorized or not. If a user is not authorizes than it throw an error i.e.” Invalid User”. If user is authorized then system is allow that user to access the system.

**Integrated Testing**

In integrated testing each form is connected to respective program, then each program is connected to its respective menu and the entire system is tested form the top level.

System Implementation

The license under which the Wordpress software is released is the GPLv2 (or later) from the [Free Software Foundation](http://www.fsf.org/). A copy of the license is included with every copy of Wordpress, but you can also read [the text of the license](https://wordpress.org/about/gpl/) here.

Part of this license outlines requirements for derivative works, such as plugins or themes. Derivatives of Wordpress code inherit the GPL license. [Drupal](http://drupal.org/), which has the same GPL license as Wordpress, has an excellent page on [licensing as it applies to themes and modules](http://drupal.org/licensing/faq/) (their word for plugins).

There is some legal grey area regarding what is considered a derivative work, but we feel strongly that plugins and themes are derivative work and thus inherit the GPL license. If you disagree, you might want to consider a non-GPL platform such as [Serendipity](http://www.s9y.org/) (BSD license) or [Habari](http://habariproject.org/en/) (Apache license) instead.

**FUTURE ENHANCEMENTS**

There is a huge scope of enhancement in the project by adding functionalities to the system. The project can be enhanced in the near future by adding more modules and can be implemented for more better use. History of all can be maintained and can be retrieved whenever requires. Progress report can be generated. This all can be modified and can be implemented in the near future.

**CONCLUSION**

While developing the system a conscious effort has been made to create and develop a software package, making use of available tools, techniques and resources – that would generate a proper Technical Institute Support System.

While making the system, an eye has been kept on making it as user-friendly, as cost-effective and as flexible as possible. As such one may hope that the system will be acceptable to any user and will adequately meet his/her needs.

As in case of any system development process where there are a number of shortcomings, there has been some shortcomings in the development of this system also. These have been mentioned in details under the topic Limitations and Future Scope of the System

**REFERENCES AND BIBLOGRAPHY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Book** | **Author** | **Publication** |
| 1. | PHP Reference: Beginner to Intermediate PHP5 | [Mario Lurig](http://www.amazon.in/Mario-Lurig/e/B002L06TDY/ref=dp_byline_cont_book_1) | PaperBack |
| 2. | The Complete Reference SQL | James R. Groff and Paul N. Weinberg TMG | Tata McGraw-Hill Publishing Company Limited |
| 3. | The Complete Reference: HTML & XHTML | THOMAS POWELL | Osborne/McGraw-Hill |
| 4. | HTML & CSS: The Complete Reference, Fifth Edition | THOMAS POWELL | PaperBack |

**Also:**

[WWW.W3SCHOOLS.COM](http://WWW.W3SCHOOLS.COM)

[WWW.TUTORIALPOINT.COM](http://www.tutorialpoint.com)

[WWW.CODEACADEMY.COM](http://www.codeacademy.com)

[WWW.WPBEGINNER.COM](http://WWW.WPBEGINNER.COM)