**Bulbamerica.com**

Module Specification

Version 1.0

August 23, 2011

**Automotive Search**

Mitali Mehta

# 1.0. Introduction

## 1.1. Purpose

The purpose of this document is to present a detailed description of the automotive search module with website. It will explain the purpose and features of the system, the interfaces of the system, what the system will do. This document is intended for both the developers of the system.

## 1.2. Scope of Project

This module is basically developed for automotive search website. This module includes front end as well as back end. The purpose of this website is basically providing automotive bulbs after selecting car manufacturer, year, model, if required type and location of the bulb. Once user selects everything from select boxes of above system gives list of products related to selected criteria. User can go through add to cart, shopping cart and check processes after that.

From back end, admin can add, edit and delete everything like Car Manufacturers, Car years, Car Models, locations and products. All values of them are in product attributes.

## 1.3. Front end files and details

**Blocks**

* **Class Ebulb\_Automativesearch\_Block\_Manufacturer**

This class is for list of manufactures by letters from A-Z using link from home page.

function \_getManufacturerByLetter(){

}

Above function returns list of manufactures when passed letter in this function in query string parameter. So once letter is passed to this function it returns the list of manufactures starting with that letter.

* **Class Ebulb\_Automativesearch\_Block\_Search**

This block is main class of this module which gives users to select from manufactuers to reach product list.

function \_construct(){

}

Above function is constructor of this class. Which sets all basic variables to be used to show on front end depending on parameters passed to it for example if we have passed manufacturer name on this page it sets name of manufacturer and also year list of that particular manufacturer. If we have passed manufacturer and year as well it will set model list. It also set image tag for particular selection.

function getAllManufacturers{

}

Above function will return all manufacturers from attribute options to create manufacturer select box for first selection.

Function getAction(){

}

Above function returns action of form in phtml file.

Function getRedirectUrl(){

}

It returns redirect url for each and every selection depending on type. For each combination it returns url to be redirected on selecting select boxes.

**Class Ebulb\_Automativesearch\_Block\_Searchresult**

Above class is the block for listing products depending on automotive search criteria.

Function \_getProductCollection(){

}

Above function is used to generate product list using manufacturer, year, model, type (optional) and bulb location. It gets ids of each selection from query string parameters and generates product collection and create product list for users to buy.

**Controllers**

**Class Ebulb\_Automativesearch\_ManufacturerController**

function indexAction{

}

Default action to be called when called manufacturer controller.

**Class Ebulb\_Automativesearch\_ProductController**

Above controller is extended from core controller Mage\_Catalog\_ProductController to generate product list. It extends all necessary functions to generate product list.

**Class Ebulb\_Automativesearch\_SearchController**

Above class contains default action for automotive search.

**Helpers**

**Class Ebulb\_Automativesearch\_Helper\_Data**

function getAutomativeSearchUrl{

}

Above function returns search url.

function changeNameUrl(){

}

Above function is used to change name of passed parameter to pass it into url. This is used mainly to change urls for manufacturer, model name etc.

**Models**

**Class Ebulb\_Automativesearch\_Model\_Mysql4\_Automativesearch**

Above is main model for automotive search which contains all functions related to database operations.

Function \_construct(){

}

Above is a constructor to initialize the mysql model.

function getManufacturerCollection(){

}

Above function select data from database table for attribute car\_manufacturer and return into ascending order to display it on front end.

Function getYearCollection(){

}

Above function selects data from database table for attribute car\_year and returns into descending order, also when passed manufacturer id into this function, it returns year list for that particular manufacturer.

Function getModelCollection(){

}

Function getTypeCollection(){

}

Function getlocationCollection(){

}

Function getProductCollection (){

}

Above functions works same way as getYearCollection and getManufacturerCollection. Return list of attribute values models ,types , locations and products.

Function getManufacturerIDFromName(){

}

Above function returns ids of manufacturer using its name.

Function getManufacturersByLetter(){

}

Above function returns list of manufacturers started with some letter.

Function getAttrbiuteId(){

}

Above function returns attribute id for attribute code passed to it.

Function getAllOptions(){

}

Above function returns all options (values of an attribute) of attribute code passed to it.

## 1.4. Back end files and details

**Controllers**

**Class Ebulb\_Automativesearch\_Adminhtml\_AutomotivebulbsController**

function indexAction(){

}

Above function is default action called when automotive search is called from back – end.

function editAction(){

}

Above is action being called to edit the automotive search from back end.

function saveAction(){

}

Above is action being called when automotive search save.

function addmoreAction(){

}

Above is action being called from an Ajax function when admin is trying to add more locations into given combination of manufacturer, year, model and type. It added new location div tag into given combination.

function addproductAction(){

}

Above is an action being called from Ajax function when trying to add product in manufacturer, year, model, type and location combination. It adds new input box to enter sku of a product.

function deleteAction(){

}

Delete action to delete some combination of manufacturer, year, model and type.

function massDeleteAction(){

}

Mass delete action is for mass deletion of data from automotive\_product table.

function checkproductskuAction(){

}

Above is a function being called from an ajax function to check whether the sku entered by a user is valid or not and also checks for product’s status and visibility.

## 

## 1.5 Detailed Non-Functional Requirements

### Logical Structure of the Data

**automotive\_product**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| car\_manufacturer\_product\_id | INT(11) | Primary key and auto increment |  |
| car\_manufacturer\_id | INT(10) | Manufacturer id |  |
| car\_manufacturer\_year\_id | INT(10) | Year id |  |
| car\_manufacturer\_model\_id | INT(10) | Model id |  |
| car\_manufacturer\_type\_id | INT(10) | Type id |  |
| location\_id | INT(10) | Location id |  |
| product\_id | INT(10) | Product entity id |  |