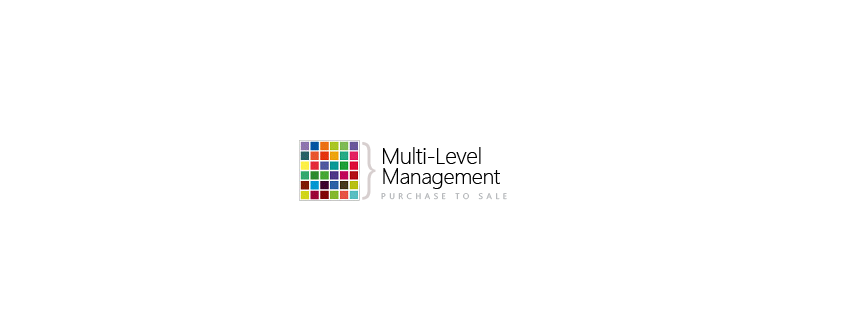
|  |
| --- |
| DA-IICT |
| Multi  Level  Marketing  Management |
| Mansuri GulamNadim :- 201212008  Savalia Hiren :- 201212024  Patel Mayur :- 201212030  Kakadiya Dipesh :- 201212046  Patel Saurabh :- 201212048 |



# Scope Statement: -

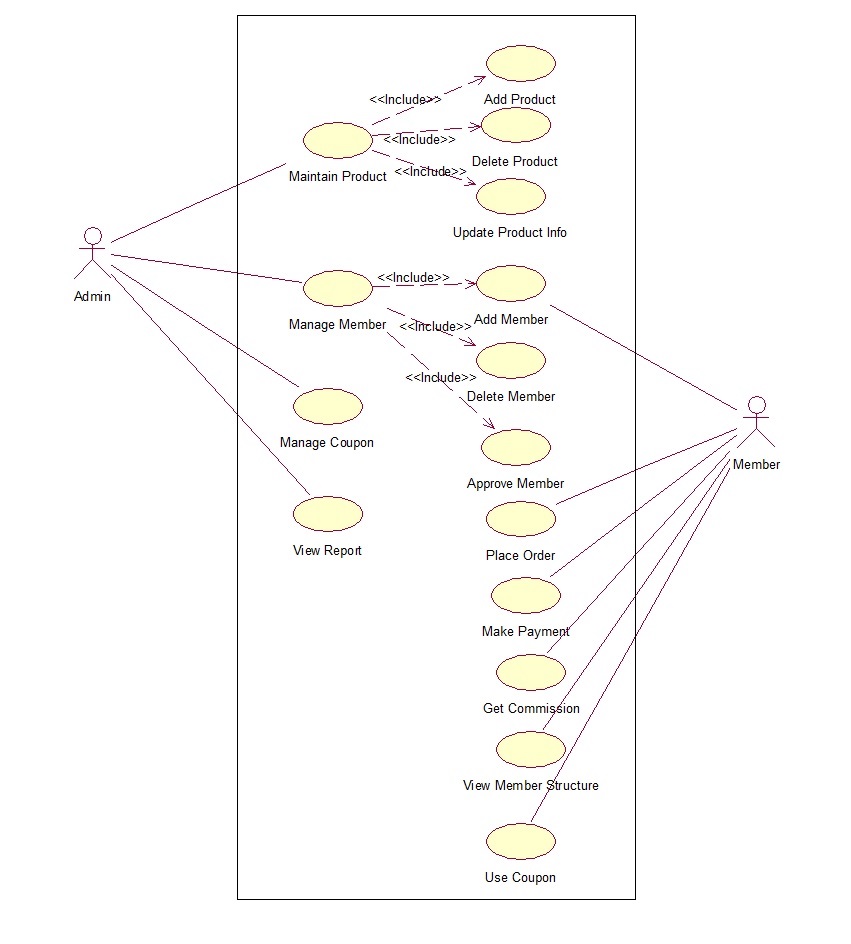
## Multi-Level Management System

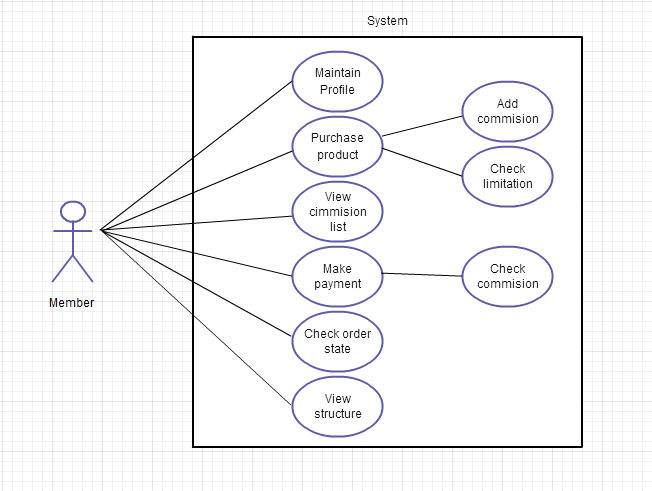
## **Multi-level marketing** (**MLM**) is a marketing strategy in which the sales force is compensated not only for sales they personally generate, but also for the sales of the other salespeople that they recruit. It can provide multiple levels of compensation.

1. **Administrative Features**  
       Create / Edit Profiles   
       Record / View Transactions   
       View User Detail  
   Discounts / coupons     
       Deactivate People From Tree   
       Graphical View of Tree
2. **User Features**   
       Secure Login

Purchase product   
 Category wise searching product   
   View Current Commission Balance

    View Monthly Earnings  
    View Transaction Details   
   Graphical View of Down-line

Use case Diagram:-  
  
  
  




Data Dictionary: -

## City: -

|  |  |  |
| --- | --- | --- |
| Field name | Data type | Description |
| CI\_ID | NUMBER | Stores city id |
| ST\_ID | NUMBER | Stores state id |
| CI\_NAME | VARCHAR2(50) | Stores city name |

## Commission: -

|  |  |  |
| --- | --- | --- |
| Field name | Data type | Description |
| COM\_ID | NUMBER | Stores commission id |
| MEM\_ID | NUMBER | Stores member id |
| COM\_DATE | DATE | Store date od commission |
| AMOUNT | NUMBER | Stores commission amount |
| REMARK | VARCHAR2(500) | Description |

## Com\_Withdraw: -

|  |  |  |
| --- | --- | --- |
| Field name | Data type | Description |
| WITH\_ID | NUMBER | Stores withdraw id |
| MEM\_ID | NUMBER | Stores member id |
| WITH\_DATE | DATE | Sores withdraw date |
| AMOUNT | NUMBER | Stores amount that is withdraw |

## Country: -

|  |  |  |
| --- | --- | --- |
| Field name | Data type | Description |
| CO\_ID | NUMBER | Stores country id |
| CO\_NAME | VARCHAR2(50) | Stores country name |

## Coupon: -

|  |  |  |
| --- | --- | --- |
| Field name | Data type | Description |
| COUPON\_ID | NUMBER | Stores coupon id |
| COUPON\_CODE | VARCHAR2(4000) | Stores coupon code |
| RATE | NUMBER | Stores rate |

## Ctg\_tax: -

|  |  |  |
| --- | --- | --- |
| Field name | Data type | Description |
| CTG\_ID | NUMBER | Stores category id |
| TAX\_ID | NUMBER | Stores tax id |
| TAX | NUMBER | Stores tax |

## Member\_chain: -

|  |  |  |
| --- | --- | --- |
| Field name | Data type | Description |
| PARENT | NUMBER | Stores parent id |
| CHILD | NUMBER | Stores child id |
| STEP | NUMBER | Store id (Distance from root) |

## Member\_details: -

|  |  |  |
| --- | --- | --- |
| Field name | Data type | Description |
| MEM\_ID | NUMBER | Stores member id |
| FNAME | VARCHAR2(50) | Stores first name |
| LNAME | VARCHAR2(50) | Store last name |
| GENDER | VARCHAR2(6) | Stores gender |
| DOB | DATE | Stores date of birth |
| R\_ADDRESS | VARCHAR2(200) | Stores address of member |
| CITY\_ID | NUMBER(5,0) | Stores city id |
| STATE\_ID | NUMBER(5,0) | Store state id |
| COUNTRY\_ID | NUMBER(5,0) | Store country id |
| PINCODE | NUMBER(8,0) | Stored pincode |
| MOBILENO | NUMBER(10,0) | Store mobile of member |
| EMAIL | VARCHAR2(50) | Store emailed of member |
| IMAGE | VARCHAR2(100) | Stores image of member |

## Notification: -

|  |  |  |
| --- | --- | --- |
| Field name | Data type | Description |
| N\_ID | NUMBER | Stores notification id |
| MEM\_ID | NUMBER | Stores member id |
| N\_DATE | DATE | Stores notification date |
| MESSAGE | VARCHAR2(500) | Stores message |
| RED\_FLAG | NUMBER(1,0) | Stores flag (read/unread) |

## Order\_details: -

|  |  |  |
| --- | --- | --- |
| Field name | Data type | Description |
| ORDER\_ID | NUMBER | Stores order id |
| ITEM\_ID | NUMBER | Stores item id |
| QTY | NUMBER | Stores no od quantity |
| RATE | NUMBER | Stores rate |
| T\_GROSS | NUMBER | Stores tax gross |
| TAX | NUMBER | Stores tax |
| TOTAL | NUMBER | Stores total amount |

## Payment: -

|  |  |  |
| --- | --- | --- |
| Field name | Data type | Description |
| PAY\_ID | NUMBER | Stores payment id |
| MEM\_ID | NUMBER | Stores member id |
| ORDER\_ID | NUMBER | Stores order id |
| PAY\_DATE | DATE | Stores payment date |
| AMOUNT | NUMBER | Stores amount |
| REMARK | VARCHAR2(200) | Stores description of payment |

## State: -

|  |  |  |
| --- | --- | --- |
| Field name | Data type | Description |
| ST\_ID | NUMBER | Stores state id |
| CO\_ID | NUMBER | Stores country id |
| ST\_NAME | VARCHAR2(50) | Stores state name |

## Tax\_master: -

|  |  |  |
| --- | --- | --- |
| Field name | Data type | Description |
| TAX\_ID | NUMBER | Stores tax id |
| TAX\_NAME | VARCHAR2(50) | Stores tax name |

## Tbl\_ctg: -

|  |  |  |
| --- | --- | --- |
| Field name | Data type | Description |
| CTG\_ID | NUMBER | Stores category id |
| CTG\_NAME | VARCHAR(50) | Stores category name |

## Tbl\_item: -

|  |  |  |
| --- | --- | --- |
| Field name | Data type | Description |
| ITEM\_ID | NUMBER | Stores item id |
| ITEM\_NAME | VARCHAR2(50) | Store item name |
| CGT\_ID | NUMBER | Store category id |
| STOCK | NUMBER | Store number of item available |
| RATE | NUMBER | Stores rate of item |
| PREORDER\_LEVEL | NUMBER |  |
| IMAGE | VARCHAR2(100) | Stores image od item |

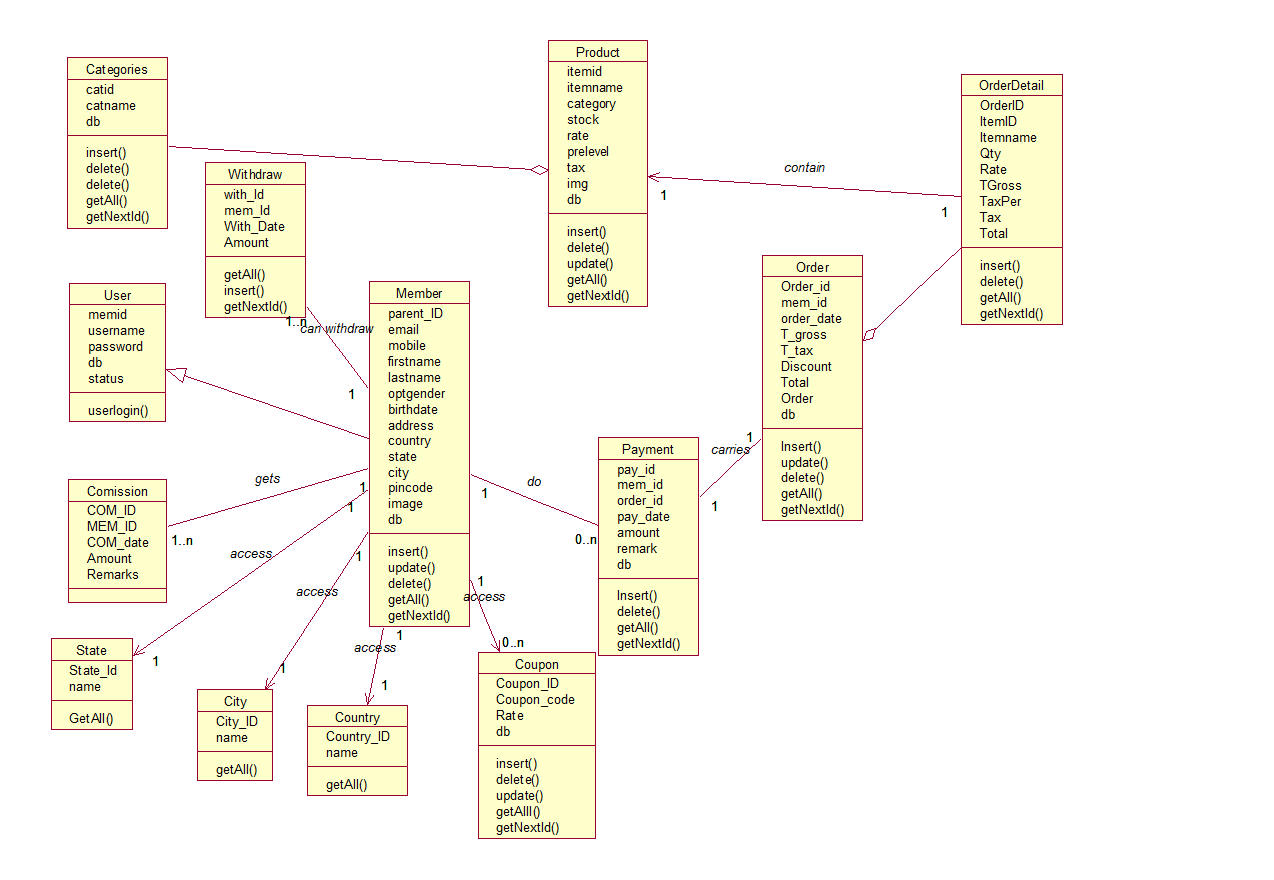
## Tbl\_member: -

|  |  |  |
| --- | --- | --- |
| Field name | Data type | Description |
| MEM\_ID | NUMBER | Stores member id |
| USERNAME | VARCHAR2(30) | Stores username |
| PASSWORD | VARCHAR2(15) | Stores password |
| TYPE | VARCHAR2(15) | Stores memmebr type |
| ACC\_FLAG | NUMBER(1,0) | Flag |

## Tl\_order: -

|  |  |  |
| --- | --- | --- |
| Field name | Data type | Description |
| ORDER\_ID | NUMBER | Stores order id |
| MEM\_ID | NUMBER | Stores member id |
| ORDER-DATE | DATE | Stores order date |
| T\_GROSS | NUMBER | Stores tax gross |
| T\_TAX | NUMBER | Stores tax |
| DISCOUNT | NUMBER | Stores discount |
| TOTAL | NUMBER | Stores total amount |

## Class diagram: -



## Controller: -

A **controller** can send commands to its associated view to change the view's presentation of the model. It can also send commands to the model to update the model's state (e.g., editing a document).

## **/\***

## **\* To change this template, choose Tools | Templates**

## **\* and open the template in the editor.**

## **\*/**

## **packagecom.mlm.controller;**

## **importcom.mlm.action.\*;**

## **importcom.mlm.dbutility.ObjectCreator;**

## **importjava.io.IOException;**

## **importjava.io.PrintWriter;**

## **importjava.util.Properties;**

## **importjavax.servlet.RequestDispatcher;**

## **importjavax.servlet.ServletException;**

## **importjavax.servlet.http.HttpServlet;**

## **importjavax.servlet.http.HttpServletRequest;**

## **importjavax.servlet.http.HttpServletResponse;**

## **importjavax.servlet.http.HttpSession;**

## **importjavax.swing.text.View;**

## **/\*\***

## **\***

## **\* @author sai**

## **\*/**

## **public class Controller extends HttpServlet {**

## **private final static String ACTION\_MAPPING = "com/mlm/controller/ActionMapping.properties";**

## **/\*\***

## **\* Processes requests for both HTTP**

## **\* <code>GET</code> and**

## **\* <code>POST</code> methods.**

## **\***

## **\* @param request servlet request**

## **\* @param response servlet response**

## **\* @throws ServletException if a servlet-specific error occurs**

## **\* @throws IOException if an I/O error occurs**

## **\*/**

## **protected void processRequest(HttpServletRequest request, HttpServletResponse response)**

## **throwsServletException, IOException {**

## **response.setContentType("text/html;charset=UTF-8");**

## **PrintWriter out = response.getWriter();**

## **HttpSession session = request.getSession();**

## **try {**

## **String view=null;**

## **/\* TODO output your page here. You may use following sample code. \*/**

## **if (session.getAttribute("sessionUserName") == null ||session.getAttribute("sessionUserName").equals("") ) {**

## **view="index.jsp";**

## **} else {**

## **String theAction = request.getParameter("Action");**

## **Properties map = new Properties();**

## **map.load(this.getClass().getClassLoader().getResourceAsStream(ACTION\_MAPPING));**

## **String action\_class = map.getProperty(theAction);**

## **Action action = (Action) ObjectCreator.createObject(action\_class);**

## **//session.getAttribute("sessionMemid")**

## **request.setAttribute("cur\_user", session.getAttribute("sessionMemid"));**

## **view = action.execute(request, response);**

## **}**

## **if (view != null) {**

## **RequestDispatcherrd = request.getRequestDispatcher(view);**

## **rd.forward(request, response);**

## **}**

## **} finally {**

## **out.close();**

## **}**

## **}**

## **// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">**

## **/\*\***

## **\* Handles the HTTP**

## **\* <code>GET</code> method.**

## **\***

## **\* @param request servlet request**

## **\* @param response servlet response**

## **\* @throws ServletException if a servlet-specific error occurs**

## **\* @throws IOException if an I/O error occurs**

## **\*/**

## **@Override**

## **protected void doGet(HttpServletRequest request, HttpServletResponse response)**

## **throwsServletException, IOException {**

## **processRequest(request, response);**

## **}**

## **/\*\***

## **\* Handles the HTTP**

## **\* <code>POST</code> method.**

## **\***

## **\* @param request servlet request**

## **\* @param response servlet response**

## **\* @throws ServletException if a servlet-specific error occurs**

## **\* @throws IOException if an I/O error occurs**

## **\*/**

## **@Override**

## **protected void doPost(HttpServletRequest request, HttpServletResponse response)**

## **throwsServletException, IOException {**

## **processRequest(request, response);**

## **}**

## **/\*\***

## **\* Returns a short description of the servlet.**

## **\***

## **\* @return a String containing servlet description**

## **\*/**

## **@Override**

## **public String getServletInfo() {**

## **return "Short description";**

## **}// </editor-fold>**

## **}**

## View: -

A **view** requests from the model the information that it needs to generate an output representation to the user.

Views in MLM:-

**Approvemember.jsp: -**This page is belongs to admin. When ant new user register in our system, admin can approve his/her request from this page.

**Category.jsp :-**This page is used to display different category of products

**Commission.jsp:-**This page is belongs to member of the system. User can check his commission from this page.

**Coupon.jsp :-**   
This page display available coupon offer.

**Member.jsp :-**   
This page belongs to admin mainly. If admin is logged into system then admin can add and remove member. If member is logged into system then member can only add other member.

**Memberchain.jsp:-**This page will show member in chart form.

**Order.jsp:-**From this page member can order various product and add it to cart.

**Order\_all.jsp :-**This page show all product that user has ordered

**Payment.jsp:-**User can make payment using paypal.

**Product.jsp:-**This page shows all details of products.

**Profile.jsp:-**User can view/edit his/her profile from this page.

**Withdraw.jsp:-**This page will show information of amount after withdrawing from commission.

**Index.jsp:-**Index page is main home page of the system. You can

API USED:-

JSTL:-

The Java Server Pages Standard Tag Library (JSTL) is a collection of useful JSP tags which encapsulates core functionality common to many JSP applications.

JSTL has support for common, structural tasks such as iteration and conditionals, tags for manipulating XML documents, internationalization tags, and SQL tags. It also provides a framework for integrating existing custom tags with JSTL tags.

The JSTL tags can be classified, according to their functions, into following JSTL tag library groups that can be used when creating a JSP page:

* **Core Tags**
* **Formatting tags**
* **SQL tags**
* **XML tags**
* **JSTL Functions**
* **FusionCharts**

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%><c:forEach items="${Categories}" var="s">  
<tr>

<td><div class="checksquared"><input type="checkbox" name="checkbox[]" /><label></label></div>

</td>  
 <td>${s.getCatid()}</td> <td>${s.getCatname()}</td>

<td> <span class="tip" >

<a href="Controller?Action=ShowCategory&uid=${s.getCatid()}" title="Edit" ><img src="images/icon/icon\_edit.png" ></a>

</span>

<span class="tip" >

<a href="Controller?Action=DeleteCategory&did=${s.getCatid()}" class="Delete" name="delete name" title="Delete" ><img src="images/icon/icon\_delete.png" ></a>

</span>  
 </td>

</tr>

</c:forEach>