**A**

**Project Report**

**On**

**“Document Management System”**

Towards fulfillment of the requirement in

Masters in Computer Applications (Semester - VI)

Submitted By:

Dharmendra K Jadav

Internal Project Guide:External Project Guide:

Prof.Vinita Nair Mr.AshishPrajapati

(Assistant Professor) (Sr.Software Professional)

**ShriJairambhai Patel Institute of Business Management & Computer Applications(SJPI)**

**ACKNOWLEDGEMENT**

We would like to express our sincere thanks to Prof.Vinita Nair and Mr.AshishPrajapatifor their Guidance throughout the project.

We are very thankful to Prof.VinitaNair and Mr.AshishPrajapatifor their constant support and Encouragement during this phase and forgive us valuable suggestions time to time.

Last but not least we are thankful to our family members and Friends for their silent co-operation and passion, and to all others known and unknown person who helped us in this duration.

**COMPANY PROFILE**

|  |  |
| --- | --- |
| **Name** | Sarjen System Pvt Ltd |
| **Address** | 10th Floor, Akshat Tower ,  Opp. Rajputh Club S.G Highway  Ahmedabad-380054 |
| **Phone Number** | +91-79-66311984 |
| **Fax Number** | +91-79-26870777 |
| **Email ID** | [sarjen@sarjen.com](mailto:sarjen@sarjen.com) |
| **Website** | www.sarjen.com |
| **Contact Person** | Mr.AshishPrajapati |
| **Contact Person Phone Number** | +91-79-66311984/85 |

**PROJECT TITLE**

* Document Management System

**PROJECT DEFINITION**

* **Document Management System,** in which we can manage the intellectual knowledge of an organization.
* This knowledge can be in the form of the documents or it can be in the form of the data.
* Admin use will manage the projects. And normal users can Create, Add, Review and approve the documents, with proper versioning, as per the rights given to them on the various projects
* For data retrieval there are some special reports and search options so that user can easily get the details as per his/her requirement.
* The Super Admin/Admin role can be as follow:
* **Master Menu**
* **Manage Attribute**
  + Add, update, Delete Attribute
* **Manage Region**
  + Add, update, Delete Region
* **Manage Department**
  + Add, update, Delete Department
* **Manage Client**
  + Add, update, Delete Client
* **Manage Notice**
  + Add, update, Delete Notice
* **Project Menu**
* **Create Project**
  + Create Project according requirement of the client.
* **Manage Project**
  + Manage Project Details like upload files, change node status, send comments (users), approved files, Read Comments(Super Admin/Admin)
* **Submission Menu**
* **PDF Publish**
  + Generate PDF Publish of the created project which is merge PDF file
* **HTML Publish**
  + Generate HTML Publish of the created project which is HTML link file
* **User Menu**
* **Manage User**
  + Add , update, delete user on project
* **User On Project**
  + Assign user role on Project
* **Search Menu**
* **By Content**
  + Content search like project name, node name, comments, all attributes, pdf content
* **Report Menu**
* **My Pending Work**
  + Bulk Approval of the Files which you are going to submit for PDF publish
* **Doc Summary**
  + Show Project Report in Table or Tree View.

**SYSTEM REQUIREMENTS**

* Hardware – Intel Core2 Duo, 2GB RAM, 50 GB Hardisk
* Software – Front End : J2EE(Struts 2.0 Framework)

Back End : SQL Server 2008 Express Edition

Tools: Eclipse, Tomcat, JDK, Apache Open Office

Other : Java Script, JQuery, Ajax, CSS

**ANALYSIS AND DESIGN**

**About Existing System:**

* All documents managed manually.
* Reviewing, approving process was lengthy and difficult to manage.

**Feasibility Study:**

**Technical:**

* Open Source Tools and Technology :
  + Our proposed web application Development for Document Management System will use Open Source Technology (java) and Tomcat server
* Web Server :
* User can easily access this web application on internet or in LAN. Easily deploy it on Apache Tomcat Server.
* Browser :
* User can use any browser to access our web application.

**Economical:**

* Web development tool (eclipse), web server (tomcat), and java technology which are free to download and no need to pay any extra charges.
* Thus it will be economically well feasible expect cost of Sql Server 2008 license.

**Operational:**

* Project is user friendly where anyone can operate it easily with a bit of knowledge of operating.
* There is no need of expertness in operating the system.

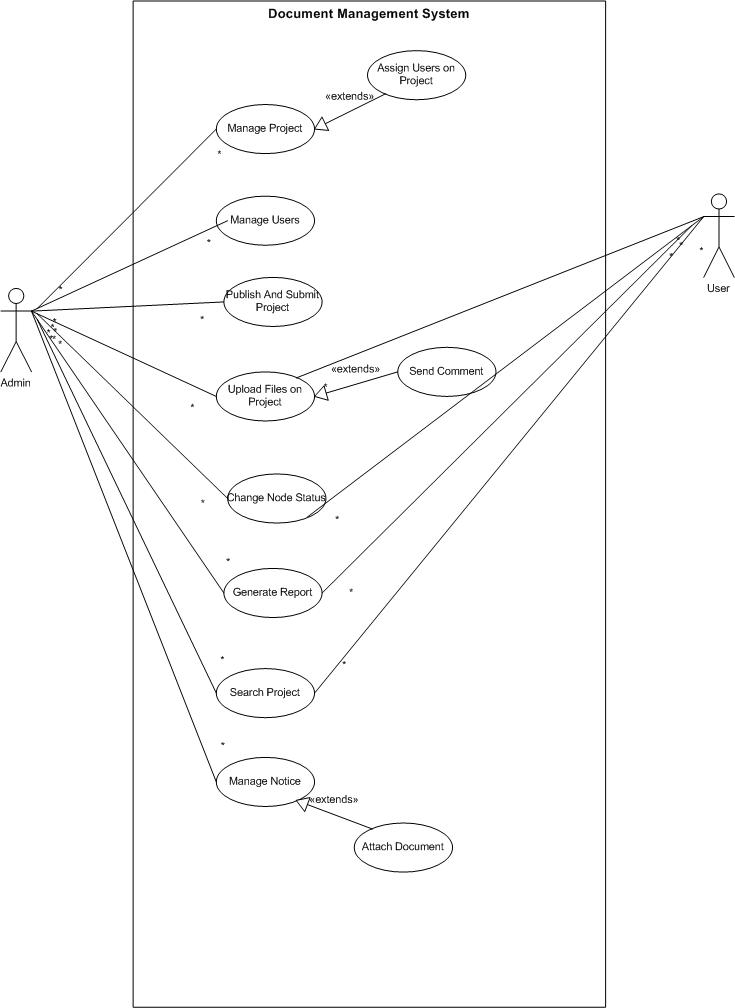
**Limitation of Existing System**

* Difficult to managed dossier for different countries.
* Extra man power and knowledge required.
* Most chances of mistake in preparation of dossier.
* Difficult to search for particular dossier file.
* Difficult to handle reports.
* Difficult to calculate work completed for particular dossier.
* Difficult to track status of each file.
* Reviewing, approving process was lengthy and difficult to manage.

**Scope of the proposed system**

* Document Management System is a powerful, feature-packed document management solution that does virtually everything from storing to organizing to managing the documents quickly, efficiently, reliably and above all, securely.
* It has easy-to-use user interface which supports multiple regional templates
* Document Management System serves multiple users connected parallel in a user-friendly environment providing an interface between the industry and agency for transfer of regulatory information
* Scalable high performance system with advanced technology
* Easily Managing, Registering, Reviewing, Archiving and Publishing documents.
* Most chances of mistake in preparation of dossier.
* Publishes the projects created inside it.
* Provide facility to manage users and user groups and its role.
* Easily maintain dossier sequence.
* Easy way to maintain regulatory files in proper structure
* Provide search facility by different search options.
* Different types of report generation facility as per requirements

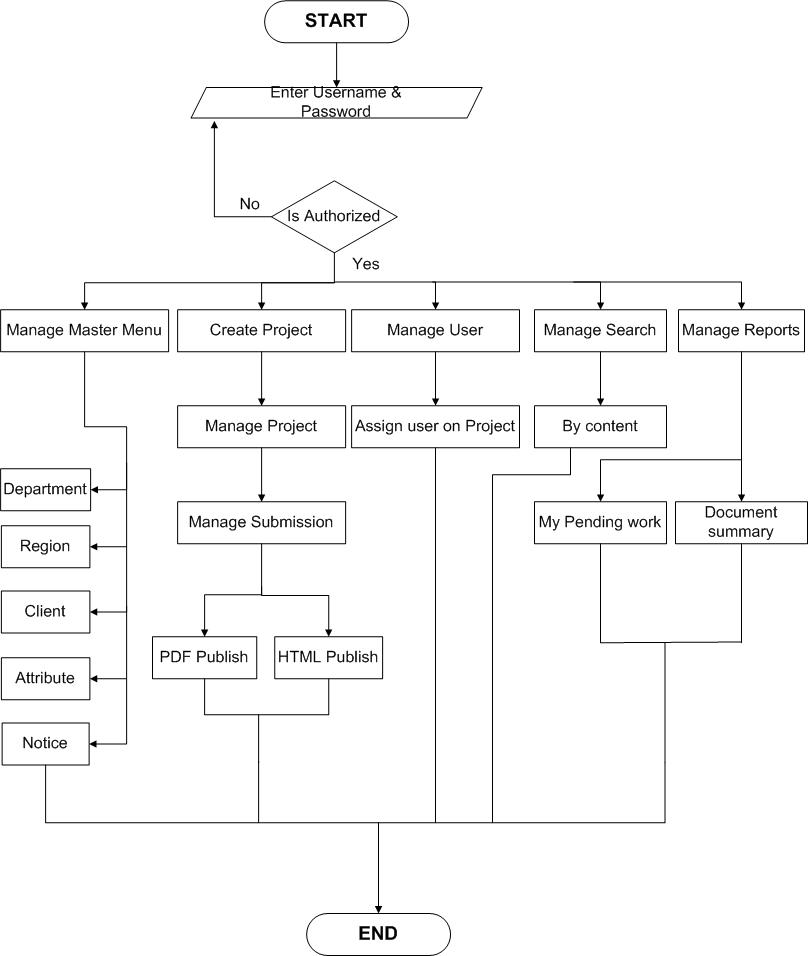
**Use Case Diagram**



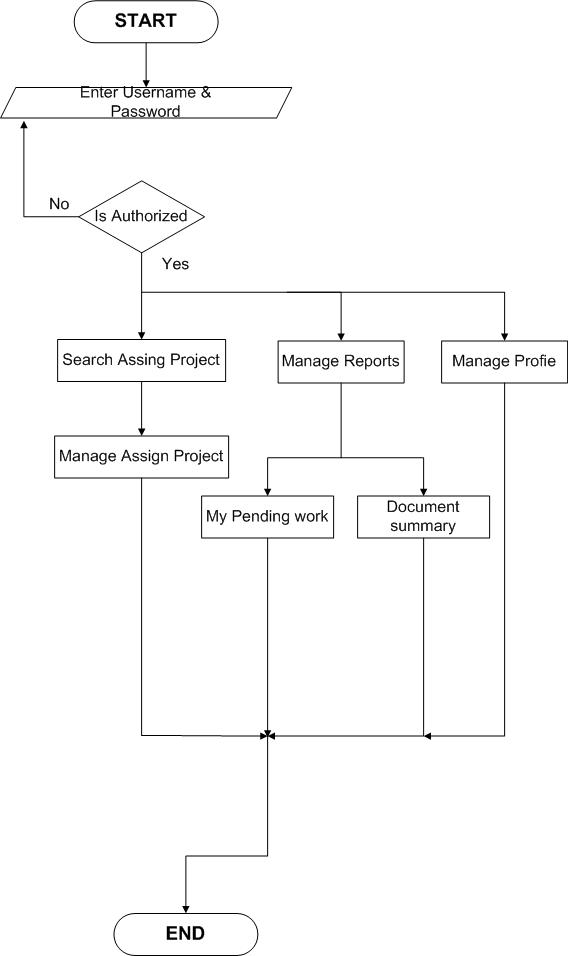
**Activity Diagram**



**System Flow Diagram(Admin)**



**System Flow Diagram(User)**



**Data Dictionary**

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName:** locationmst  **Description:** Contain the location details. This table also contains remark added by user on particular location and status modified by user. | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| vLocationCode | varchar(5) | Primary Key | Uniquely identifies the location Code. |
| vLocationName | varchar(50) |  | Name of The Location |
| vRemark | varchar(250) |  | Given Remark |
| iModifyBy | int |  | Modified by which admin or user. Depend on userMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Status of the location  e.g., E-edit, N=new, D=delete. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName**: deptmst  **Description:** Contain the Department details. This table also contain remark added by user on particular department and status modified by user | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| vDeptCode | varchar(5) | Primary Key | Uniquely identifies the Department Code. |
| vDeptName | varchar(50) |  | Name of The Department |
| vRemark | varchar(250) |  | Given Remark |
| iModifyBy | int |  | Modified by which admin or user. Depend on userMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Status of the department  e.g., E-edit, N=new, D=delete. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName**:clientmst  **Description:** Contain client’s details. This table also contains remark added by user on particular client and status modified by user. | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| vClientCode | varchar(5) | Primary Key | Uniquely identifies the Client Code. |
| vClientName | varchar(50) |  | Name of The Client. |
| vRemark | varchar(250) |  | Given Remark |
| iModifyBy | int |  | Modified by which admin or user. Depend on userMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Status of the client.  e.g., E-edit, N=new, D=delete. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName**:usertypemst  **Description**: Contain the User Type details. This table also contains remark added by user and status modified by user | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| vUserTypeCode | varchar(4) | Primary Key | Uniquely identifies the User Type Code. |
| vUserTypeName | varchar(50) |  | Name of The User Type Like, WA,WU |
| iModifyBy | int |  | Modified by which admin or user. Depend on userMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Status of the User Type.  e.g., E-edit, N=new, D=delete. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName**:attributemst  **Description**: Contain the Attribute details. | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| iAttrId | int | Primary Key | Uniquely identifies the Attribute. |
| vAttrName | varchar(50) |  | Name Of Attribute |
| vAttrValue | varchar(50) |  | Contain Attribute Value. Like combo, textarea. |
| vAttrType | varchar(15) |  |  |
| vAttrForIndiId | varchar(4) | Foreign Key from AttrForIndiMst | Attribute Indcation. |
| cUserInput | char(1) |  | Show user input status Y=Yes,N=No |
| vUserTypeCode | varchar(4) | Foreign Key from  usertypemst | Specify the UserTypeCode |
| vRemark | varchar(250) |  | Contain remark. |
| iModifyBy | int |  | Modified by which admin or user. Depend on userMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Status of the User Group.  e.g., E-edit, N=new, D=delete. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName**:attributevaluematrix  **Description**: Contain the AttributeValueMatrix details. | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| iAttrValueId | int | Primary Key | Uniquely identifies the AttrValueId |
| iAttrId | int | Foreign Key from attributemst | Specify the AttrId from attributemst |
| vAttrValue | varchar(50) |  | Contain Attribute Value. |
| iModifyBy | int |  | Modified by which admin or user. Depend on userMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Status of the Attribute Value.  e.g., E-edit, N=new, D=delete. |
| vAttrDisplayValue | varchar(250) |  | Specify the User side Attribute Value |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName**:NoticeMst  **Description**: Contain Notice Details. | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| nNoticeNo | int | Primary Key | Uniquely identifies the Notice |
| vSubject | varchar(50) |  | Subject Detail. |
| vDescription | text |  | Contain Description. |
| vAttachment | varchar(50) |  | Contain Attachment path of document. |
| dStartdate | datetime |  | Notice Start Date |
| dEndDate | datetime |  | Notice End Date |
| vUserTypeCode | varchar(4) | Foreign Key from usertypemst | Ref UseraTypeMst.vUserTypeCode |
| cIsActive | char(1) |  | Active or inactive flag. |
| iCreatedBy | int |  | Ref: usermst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Contain Status.  e.g., E-edit, N=new, D=delete. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName**:projectmst  **Description**: Contain Project type details. This table also contains remark added by user on particular project type and status modified by user. | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| vProjectCode | varchar(5) | Primary Key | Uniquely identifies the Project. |
| vProjectName | varchar(50) |  | Name of The Project. ex. DMS |
| vRemark | varchar(250) |  | Given Remark |
| iModifyBy | int |  | Modified by which admin or user. Depend on userMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time. |
| cStatusIndi | char(1) |  | Status of the project.  e.g., E-edit, N=new, D=delete. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName**:usergroupmst  **Description**: Contain the User Group details. This table also contains remark added by user and status modified by user. | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| iUserGroupCode | int | Primary Key | Uniquely identifies the User Group. |
| vUserGroupName | varchar(50) |  | Contain Group Name |
| vLocationCode | varchar(5) | Foreign Key from ocationMst | Contain Location Code.  Ref.- locationMst.vLocationCode |
| vDeptCode | varchar(5) | Foreign Key from deptmst | Contain Department Code.  Ref.- departmentmst.vDeptCode |
| vClientCode | varchar(5) | Foreign Key from clientMst | Contain client Code.  Ref.- clientMst.vClientCode |
| vProjectCode | varchar(5) | Foreign Key from projectmst | Ref.-projectmst.vProjectCode |
| vUserTypeCode | varchar(4) | Foreign Key from usertypemst | Ref,-usertypemst.vUserTypeCode |
| vRemark | varchar(250) |  | Contain Remark |
| iModifyBy | int |  | Modified by which admin or user. Depend on userMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Status of the User Group.  e.g., E-edit, N=new, D=delete. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName**:usermst  **Description**: Contain the User details. This table also contains remark added by user and status modified by user. One unique user added to particular UserGroup | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| iUserCode | int | Primary Key | Uniquely identifies the User.  UserCode and UserGroupCode is a composite key |
| iUserGroupCode | int | Primary Key, Foreign Key | Foreign Key. Primary key of usergroupmst  UserCode and UserGroupCode is a composite key |
| vUserName | varchar(50) |  | Contain user name |
| vLoginName | varchar(20) |  | Contain login name |
| vLoginPass | varchar(100) |  | Contain password |
| vUserTypeCode | varchar(4) | Foreign Key | Contain User Type. E.g. WU,WA Ref:UserTypeMst.vUserTypeCode |
| vRemark | varchar(250) |  | Contain remark. |
| iModifyBy | int |  | Modified by which admin or user. Depend on userMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Status of the User Group.  e.g., E-edit, N=new, D=delete. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName**:CountryMst  **Description**: Contain the Country details. This table also contains remark added by user and status modified by user. One unique Countryadded to particular Countrymst | | | |
| **Column Name** | **DataType** | **Constraint** | **Descritpion** |
| vCountryId | varchar(10) | Primary Key | Uniquely identifies the Country |
| vCountryName | varchar(250) |  | Specify the CountryName |
| vRegionId | varchar(10) | Foreign key from locationmst | Ref: CountryMst.locationmst |
| vCountryCode | varchar(10) |  | Specify the code of country like eu,us |
| iModifyBy | int |  | Modified by which admin or user. Depend on CountryMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Status of the User Group.  e.g., E-edit, N=new, D=delete. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName**:AttrForIndiMst  **Description**: Describe the level of attribute for Project | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| vAttrForIndiId | varchar(4) | Primary Key | Uniquely identifies the AttrForIndiId |
| vAttrForIndiName | varchar(50) |  | Specify the Attribute Indication like project level, leaf, ,user defined |
| vRemark | varchar(100) |  | Contain remark. |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| iModifyBy | int |  | Modified by which admin or user. Depend on AttrForIndiMst.iUserCode |
| cStatusIndi | char(1) |  | Status of the User Group.  e.g., E-edit, N=new, D=delete. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName**:workspacemst  **Description:** Contain Workspace Details. | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| vWorkSpaceId | varchar(50) | Primary Key | Uniquely identify each workspace. |
| vWorkSpaceDesc | varchar(255) |  | Contain Workspace Description |
| vLocationCode | varchar(5) | Foreign Key | Ref. Locatiomst.vLocationCode |
| vDeptCode | varchar(5) | Foreign Key | Ref. DeptMst.vDeptCode |
| vClientCode | varchar(5) | Foreign Key | Ref. ClientMst.vClientCode |
| vProjectCode | varchar(5) | Foreign Key | Ref. ProjectMst.vProjectCode |
| vDocTypeCode | varchar(5) |  | Type of document like DMS |
| vTemplateId | varchar(50) | Foreign Key | Ref. TemplateMst.vTemplateId |
| vTemplateDesc | varchar(250) |  | Specify the template description |
| vBaseWorkFolder | varchar(250) |  | Current working folder of workspace |
| vBasePublishFolder | varchar(250) |  | Publish folder of the workdpace |
| iLastTranNo | int |  | Last Transection number of workspace |
| vLastPublishedVersion | varchar(20) |  | Last publish version of workspace |
| iCreatedBy | int |  | Modified by which admin or user. Depend on workspacemst.iUserCode |
| dCreatedOn | datetime |  | Date and time when workspace is created |
| iLastAccessedBy | int |  | Last access of the workspace |
| dLastAccessedOn | datetime |  | Last access of date and time of workspace |
| cProjectType | char(1) |  | Status of project |
| vRemark | varchar(250) |  | Conatain remark |
| iModifyBy | int |  | Modified by which admin or user. Depend on workspacemst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Status of the User Group.  e.g., E-edit, N=new, D=delete. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName**:workspacenodedetail  **Description**: Contain workspace Node Details. | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| vWorkSpaceId | varchar(50) | Composite key  (vWorkSpaceId, iNodeId) | Ref: WorkSpaceMst. vWorkSpaceId |
| iNodeId | int | Composite key  (vWorkSpaceId, iNodeId) | Ref:TemplateNodeDetail.  iNodeId |
| iNodeNo | int |  | Node Number |
| vNodeName | varchar(100) |  | Node Name |
| vNodeDisplayName | Varchar(250) |  | Node Display Description |
| cNodeTypeIndi | varchar(1) |  | Node Indication |
| iParentNodeId | int |  | Parent Node Id Ref. iNodeId |
| vFolderName | varchar(250) |  | Folder Name |
| cRequiredFlag | varchar(1) |  | Required Flag |
| dCheckOutOn | datetime |  | Which time checkout |
| iCheckOutBy | int |  | Which user checkout by |
| cPublishFlag | varchar(1) |  | Publish or not |
| vRemark | varchar(2000) |  | Remark indication |
| iModifyBy | int |  | Modified by which admin or user. Depend on userMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | varchar(1) |  | Contain Status.  e.g., E-edit, N=new, D=delete. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName**:workspacenodeattrdetail  **Description**: Workspace Node Attributes Details. | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| vWorkspaceId | varchar(50) | Composite Key  (vWorkspaceId, iNodeId, iAttrId) | Ref: WorkSpaceMst. vWorkSpaceId |
| iNodeId | int | Composite Key  (vWorkspaceId, iNodeId, iAttrId) | Ref:TemplateNodeDetail.  iNodeId |
| iAttrId | int | Composite Key  (vWorkspaceId, iNodeId, iAttrId) | Ref: AttributeMst. iAttrId |
| vRemark | varchar(250) |  | Contain Remark. |
| iModifyBy | int |  | Modified by which admin or user. Depend on userMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Contain Status.  e.g., E-edit, N=new, D=delete. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName**:workspacenodehistory  **Description**: workspace nodehistory Details. | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| vWorkSpaceId | varchar(50) | Composite Key  (vWorkSpaceId, iNodeId, iTranNo) | Ref: WorkSpaceMst. vWorkSpaceId |
| iNodeId | int | Composite Key  (vWorkSpaceId, iNodeId, iTranNo) | Ref:TemplateNodeDetail.  iNodeId |
| iTranNo | int | Composite Key  (vWorkSpaceId, iNodeId, iTranNo) | Refer to last transection of particular node history |
| vFileName | varchar(250) |  | File name of node |
| vFileType | varchar(10) |  | Type of file in node |
| vFolderName | varchar(250) |  | Folder path of each node |
| iStageId | int | Foreign Key | Ref: StageMst. iStageId |
| vRemark | varchar(250) |  | Contain Remark. |
| iModifyBy | int |  | Modified by which admin or user. Depend on userMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Contain Status.  e.g., E-edit, N=new, D=delete. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName**:workspacenodeattrhistory  **Description**: workspace node attribute history Details. | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| vWorkSpaceId | varchar(50) | Composite Key(  vWorkSpaceId, iNodeId, iTranNo, iAttrId) | Ref: WorkSpaceMst. vWorkSpaceId |
| iNodeId | int | Composite Key(  vWorkSpaceId, iNodeId, iTranNo, iAttrId) | Ref:TemplateNodeDetail.  iNodeId |
| iTranNo | int | Composite Key(  vWorkSpaceId, iNodeId, iTranNo, iAttrId) | Refer to last transection of particular node history |
| iAttrId | int | Composite Key(  vWorkSpaceId, iNodeId, iTranNo, iAttrId) | Ref: AttributeMst. iAttrId |
| vRemark | varchar(250) |  | Contain Remark. |
| iModifyBy | int |  | Modified by which admin or user. Depend on userMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Contain Status.  e.g., E-edit, N=new, D=delete. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName**:workspaceusermst  **Description**: Contain details of users assigned on particular Workspace. | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| vWorkSpaceId | varchar(50) | Composite Key  (vWorkSpaceId, iUserGroupCode, iUserCode) | Ref: WorkSpaceMst. vWorkSpaceId |
| iUserGroupCode | int | Composite Key  (vWorkSpaceId, iUserGroupCode, iUserCode) | Ref: UserGroupMst. iUserGroupCode |
| iUserCode | int | Composite Key  (vWorkSpaceId, iUserGroupCode, iUserCode) | Ref:UserMst.iUserCode |
| cAdminFlag | char(1) |  | Contain Admin Flag Y or N |
| dLastAccessedOn | datetime |  | Date and time of last access |
| vRemark | varchar(250) |  | Contain Remark. |
| iModifyBy | int | f | Modified by which admin or user. Depend on userMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Contain Status.  e.g., E-edit, N=new, D=delete. |
| dFromDt | datetime |  | Contain starting date for assigned workspace. |
| dToDt | datetime |  | Contain last date for rights on workspace. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName:**Forumdtl  **Description:**Contain comment details | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| vSubjectId | varchar(40) | Primary Key | Uniquely identifies the vSubjectId |
| vSubjectDesc | varchar(50) |  | Short Description of node file |
| iModifyBy | int |  | Modified by which admin or user. Depend on CountryMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Contain Status.  e.g., E-edit, N=new, D=delete. |
| vFileName | varchar(500) |  | Name of file in which comment is written |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName:**Forumhdr  **Description:**Contain comment details sent on each node of project of particular workspace | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| vWorkSpaceId | varchar(50) | Composite Key  (vWorkSpaceId, iNodeId,vSubjectI) | Ref: WorkSpaceMst. vWorkSpaceId |
| iNodeId | int | Composite Key  (vWorkSpaceId, iNodeId,vSubjectI) | Ref:TemplateNodeDetail.  iNodeId |
| vSubjectId | varchar(50) | Composite Key  (vWorkSpaceId, iNodeId,vSubjectI) | Ref: Forumdtl.vSubjectId |
| iUserGroupCode | int | Foreign Key | Ref: UserGroupMst. iUserGroupCode |
| iUserCode | int | Foreign Key | Ref:UserMst.iUserCode |
| cStatusIndi | char(1) |  | Contain Status.  e.g., E-edit, N=new, D=delete |
| cReadFlag | char(1) |  | Comment read or not Y/N |
| dCreatedOn | datetime |  | Date and time when comment send |
| vRemark | varchar(250) |  | Contain Remark |
| iModifyBy | int |  | Modified by which admin or user. Depend on CountryMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName:**workspaceuserrightsmst  **Description**: Contain Node Rights given to the users for particular workspace. | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| vWorkSpaceId | varchar(50) | Foreign Key | Ref: WorkSpaceMst. vWorkSpaceId |
| iUserGroupCode | Int | Foreign Key | Ref: UserGroupMst. iUserGroupCode |
| iUserCode | Int | Foreign Key | Ref:UserMst.iUserCode |
| iNodeId | int | Foreign Key | Ref:TemplateNodeDetail.  iNodeId |
| cCanReadFlag | char(1) |  | Contain Read flag |
| cCanAddFlag | char(1) |  | Contain Add Flag |
| cCanEditFlag | char(1) |  | Contain Edit Flag |
| cCanDeleteFlag | char(1) |  | Contain Delete Flag |
| iStageId | int | Foreign Key | Ref. StageMst.iStageId |
| vRemark | varchar(250) |  | Contain Remark. |
| iModifyBy | int |  | Modified by which admin or user. Depend on userMst.iUserCode |
| dModifyOn | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Contain Status.  e.g., E-edit, N=new, D=delete. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TableName:**SubmissionInfoHTML  **Descirption:**Contains SubmissionInformation in HTML content | | | |
| **Column Name** | **DataType** | **Constraint** | **Description** |
| iSubmissionInfoHtmlID | numeric(18, 0) | Primary Key | Unique Identity of iSubmissionInfoHtmlID |
| vWorkSpaceId | varchar(50) | Foreign Key | Ref: WorkSpaceMst. vWorkSpaceId |
| iTranNo | int |  | Last transection number of iSubmissionInfoHtmlID |
| vDescription | varchar(100) |  | Description of HTML Publish submission |
| vPublishPath | varchar(250) |  | HTML Publish path of workspace |
| iCreatedBy | int |  | Ref: usermst.iUserCode |
| iModifyBy | int |  | Modified by which admin or user. Depend on userMst.iUserCode |
| dmodifyon | datetime |  | Added, edited, deleted at what time |
| cStatusIndi | char(1) |  | Contain Status.  e.g., E-edit, N=new, D=delete. |

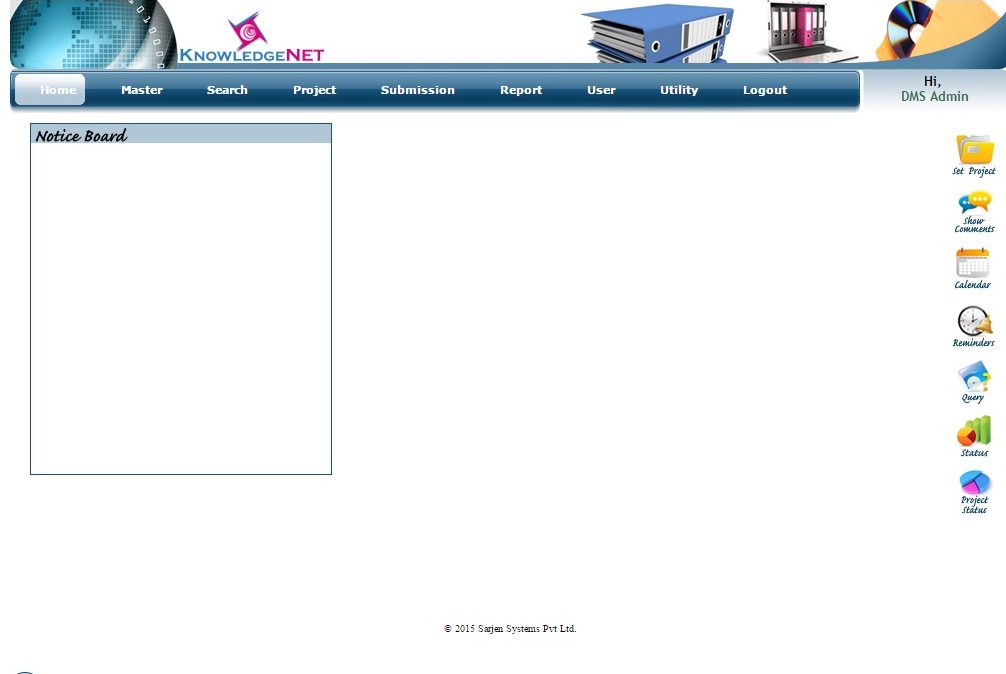
**Screen Layouts**

**Login Page:**

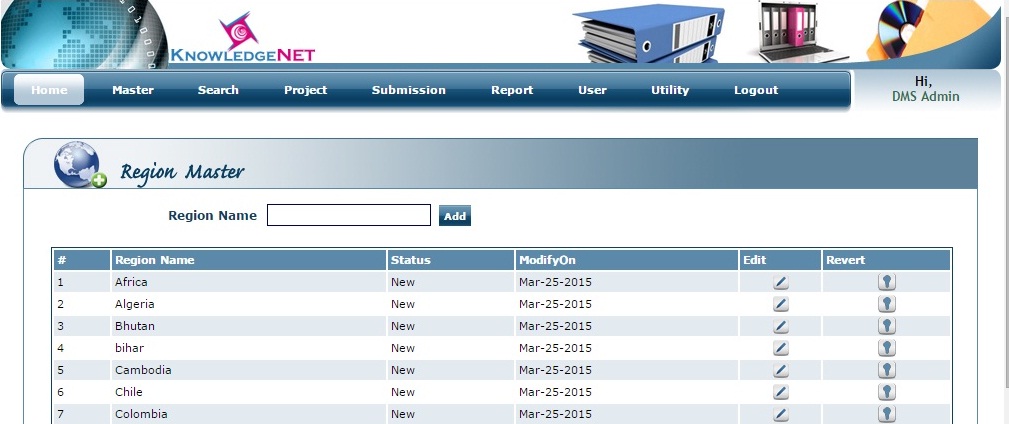
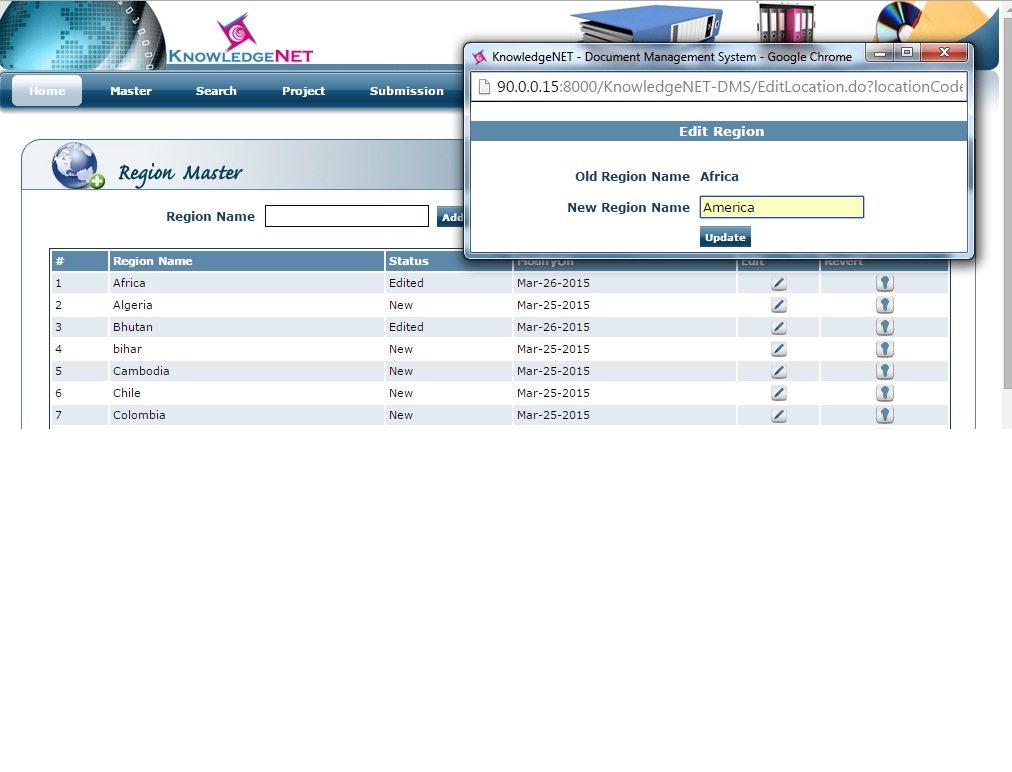
* Login page for to login application
* Username and password is required validation
* Enter valid username and password

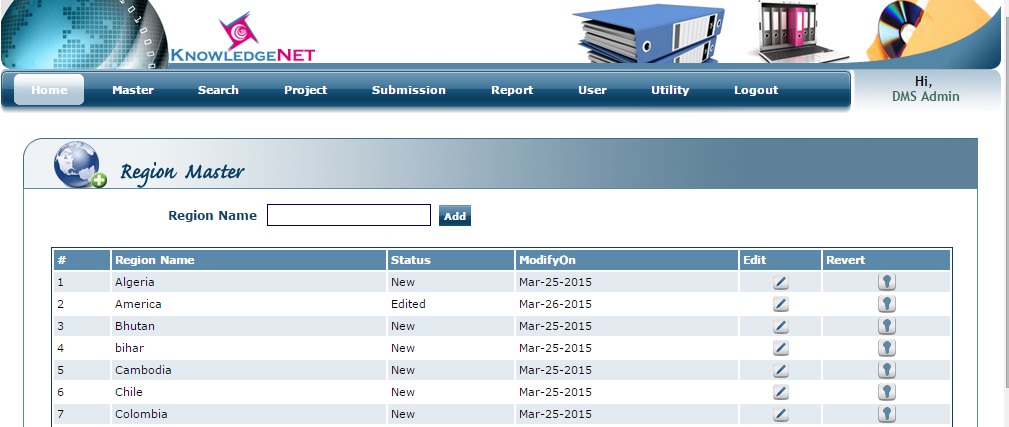


* Home Page

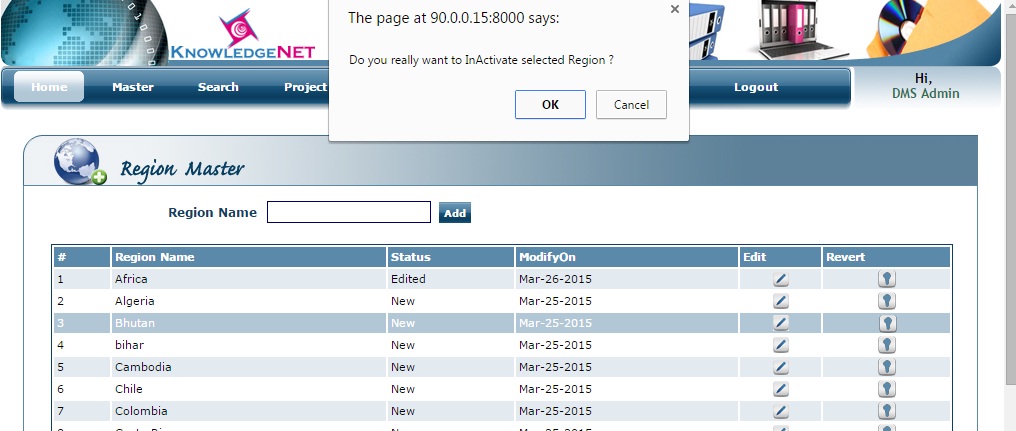


**Master Menu:** Region, Department, Location, Notice

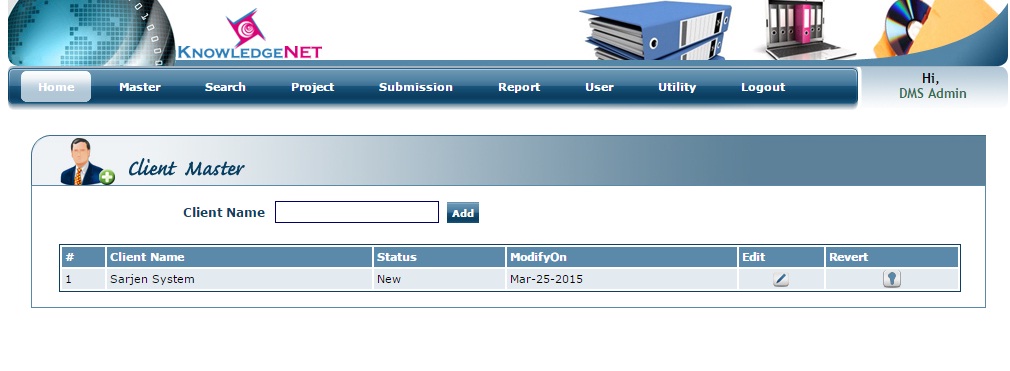
* Add Region
* Update Region to click on edit button in which you need to enter new region
* Update Region Successfully



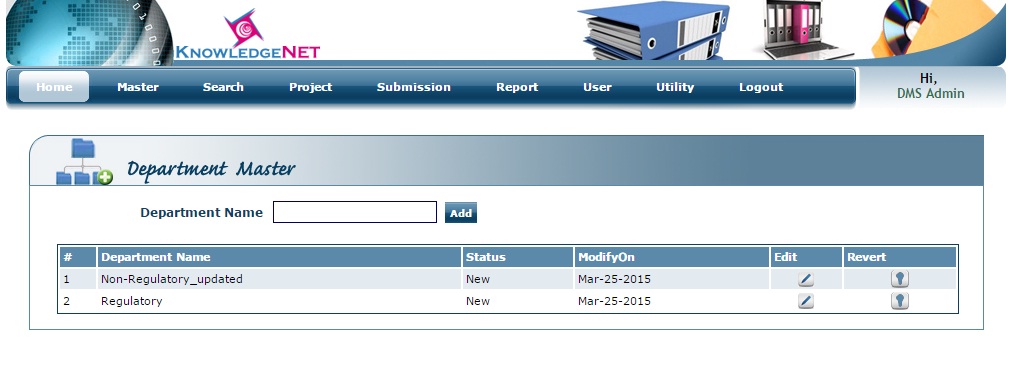
* Click on Revert button to confirm inactive selected region

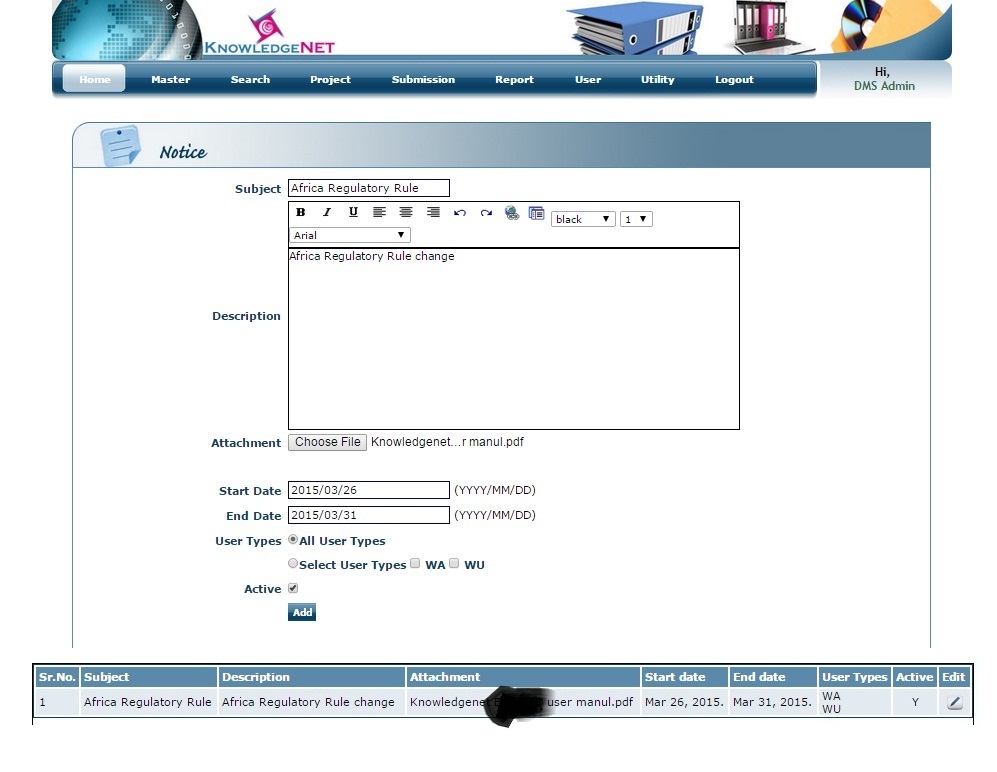


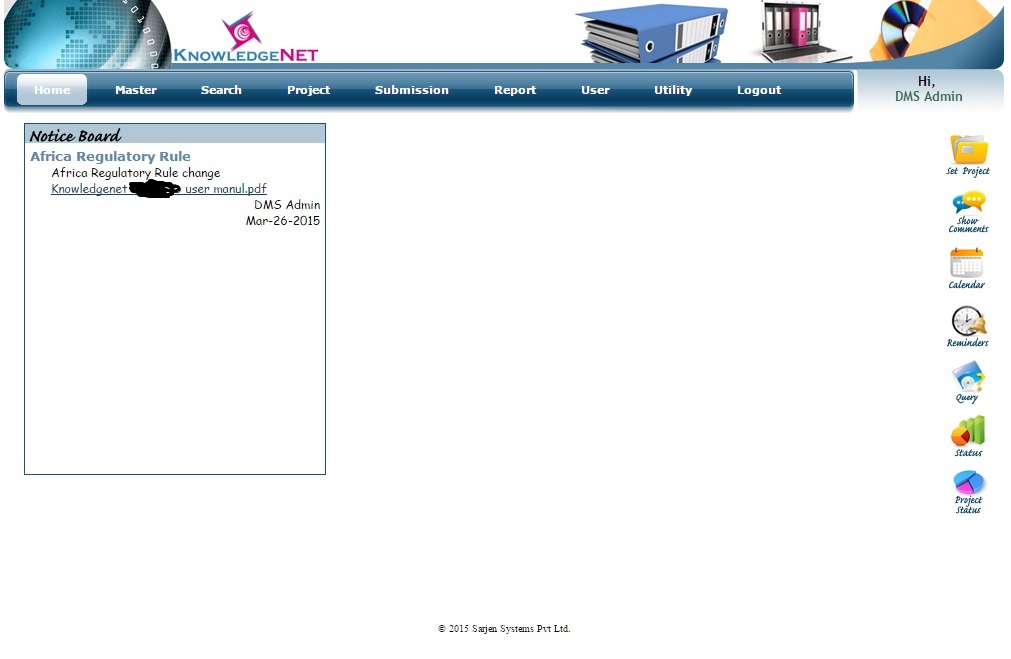
* Region is successfully inactivated
* Client master to add record



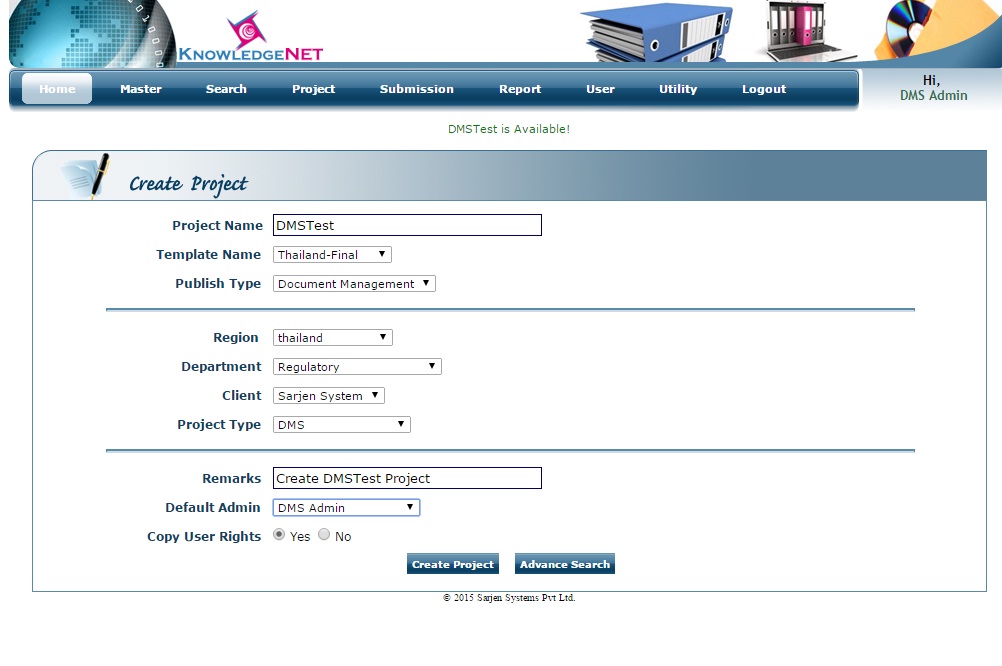
* Department master to add record

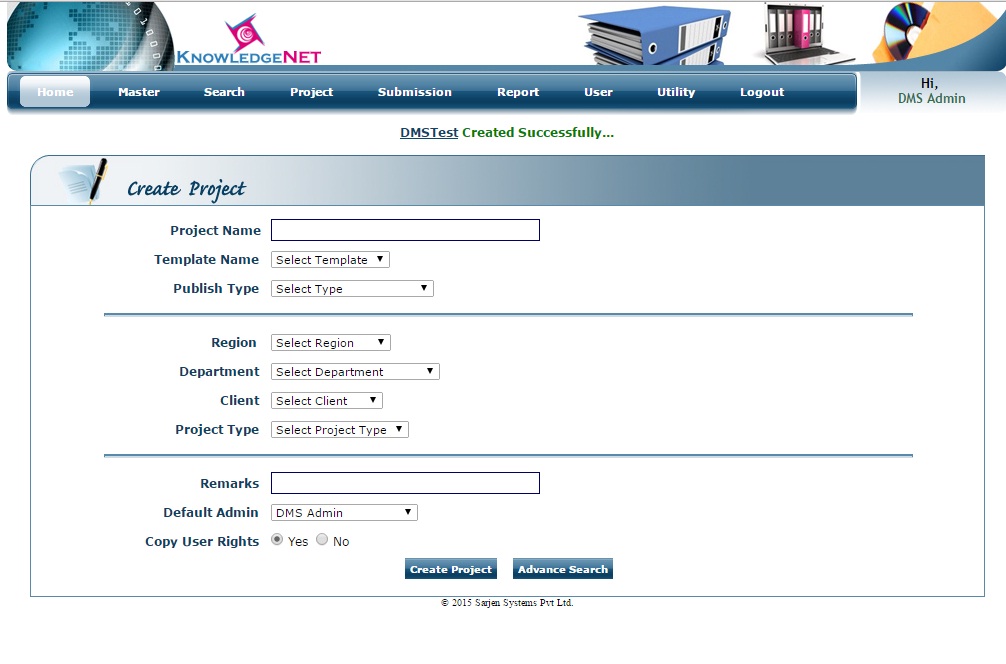


* Notice in which user can attach important notice for all users or selected users only

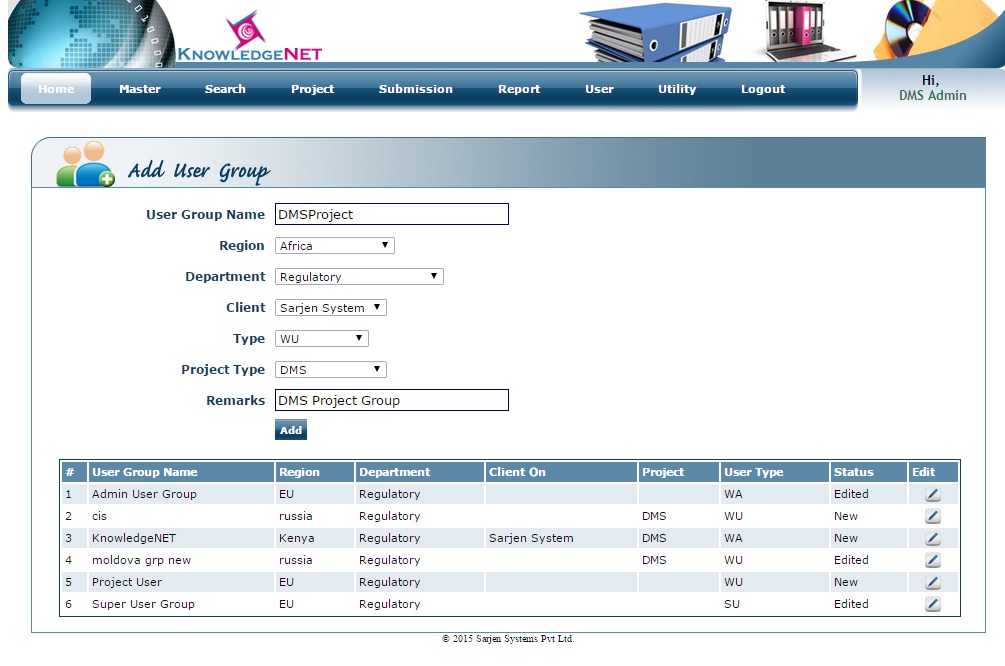
* Added notice is show on home page when particular user is login in

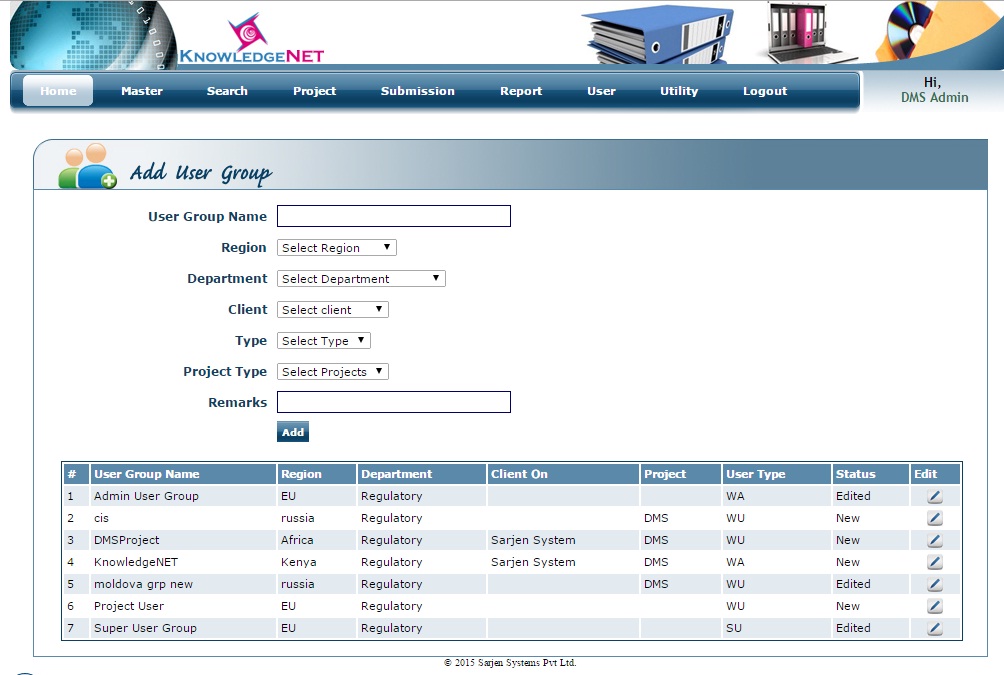
**Project Menu:**

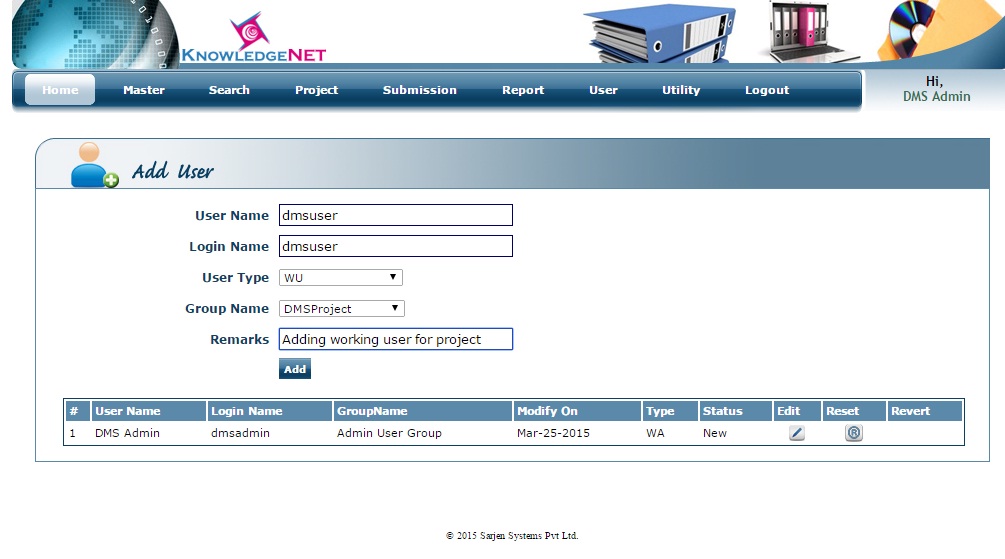
* Create Project: in which DMS admin will create project
* Project is successfully created when all required fields are filling-up.

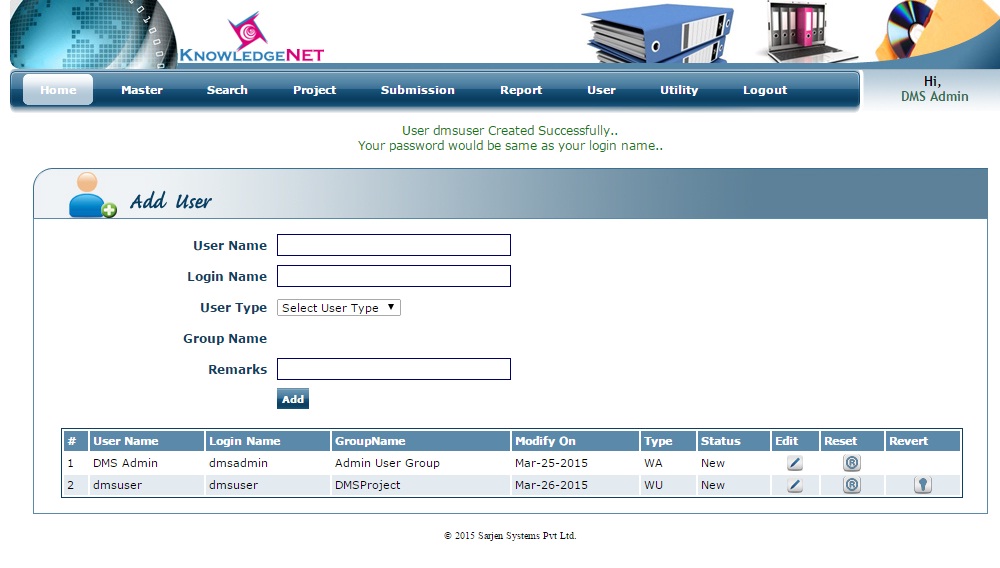


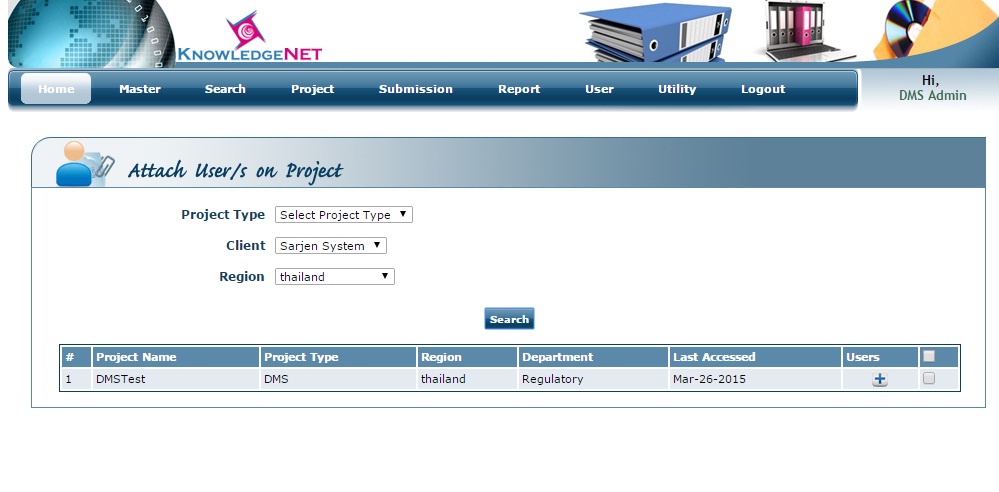
**User Menu:**Add User-Group, Add-User, Attach Users on Project

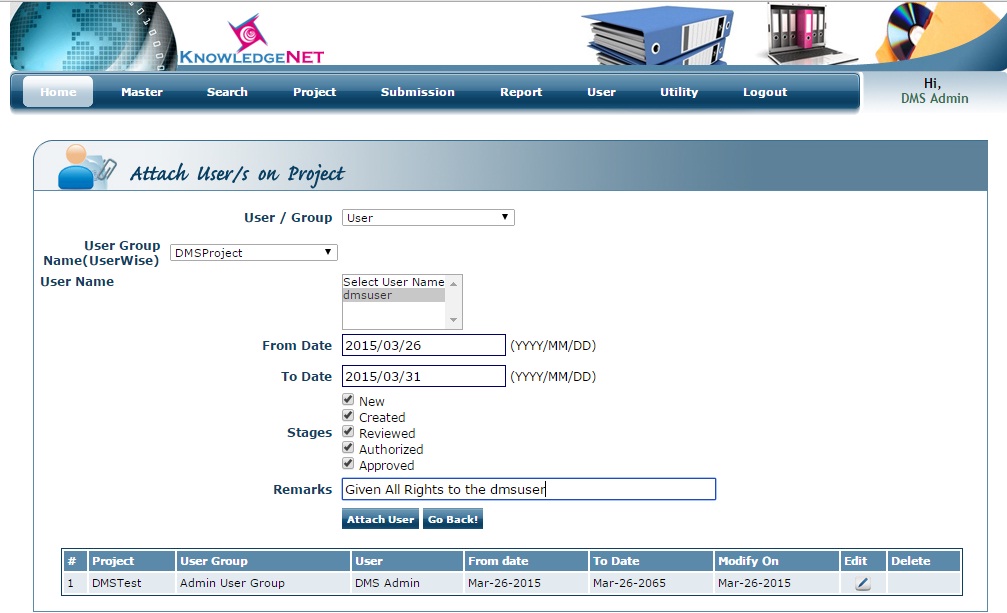
* Add user group: add DMSProject group for DMS project type
* Show added DMSProject group in the table



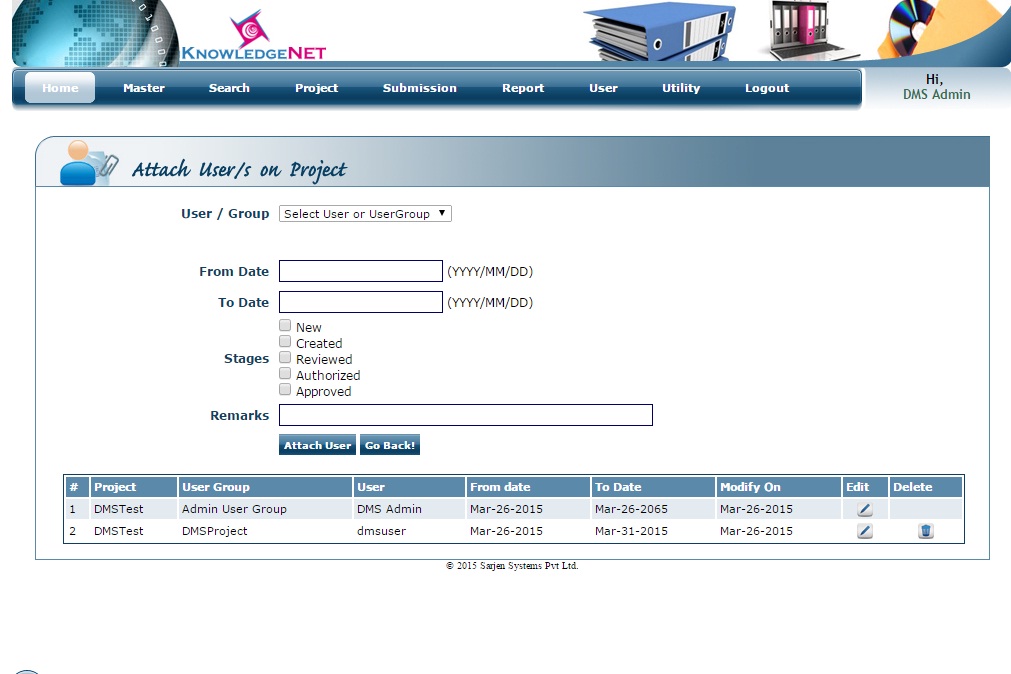
* Add user on DMSProject group, you can add multiple users on DMSProject Group
* dmsuser is successfully added in DMSProject group



* Attach users on project in which you can add user or group on selected project
* Attach users or group on selected project then assign right for project

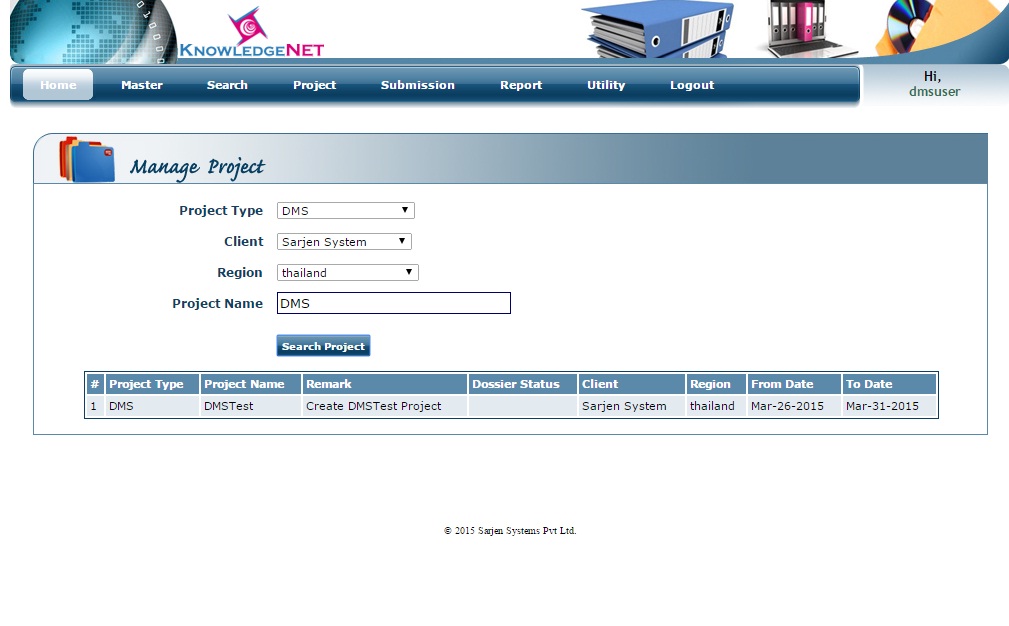


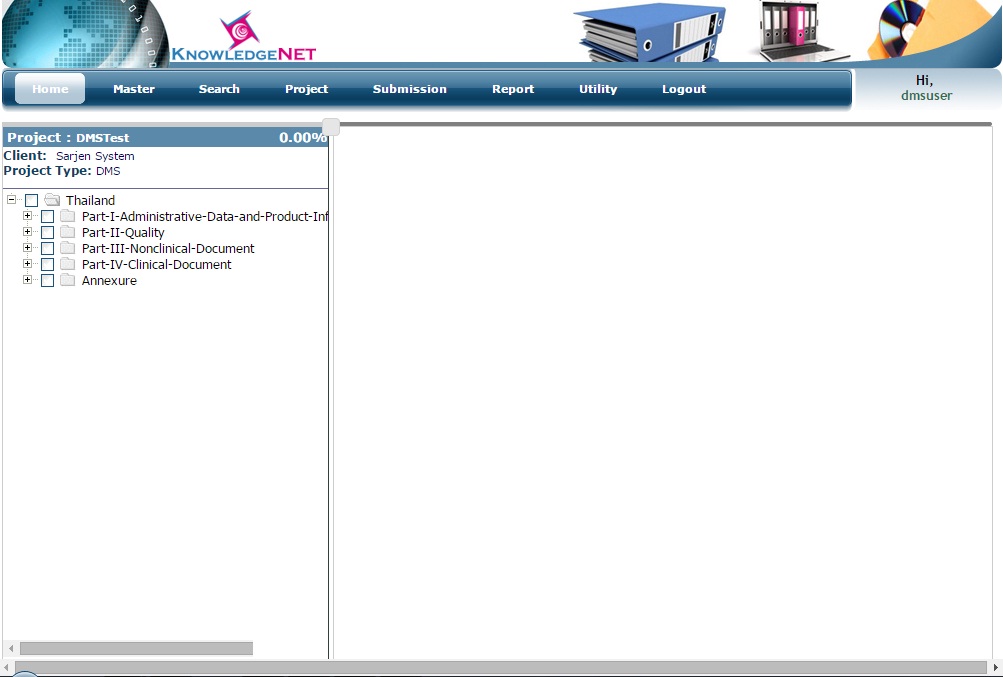
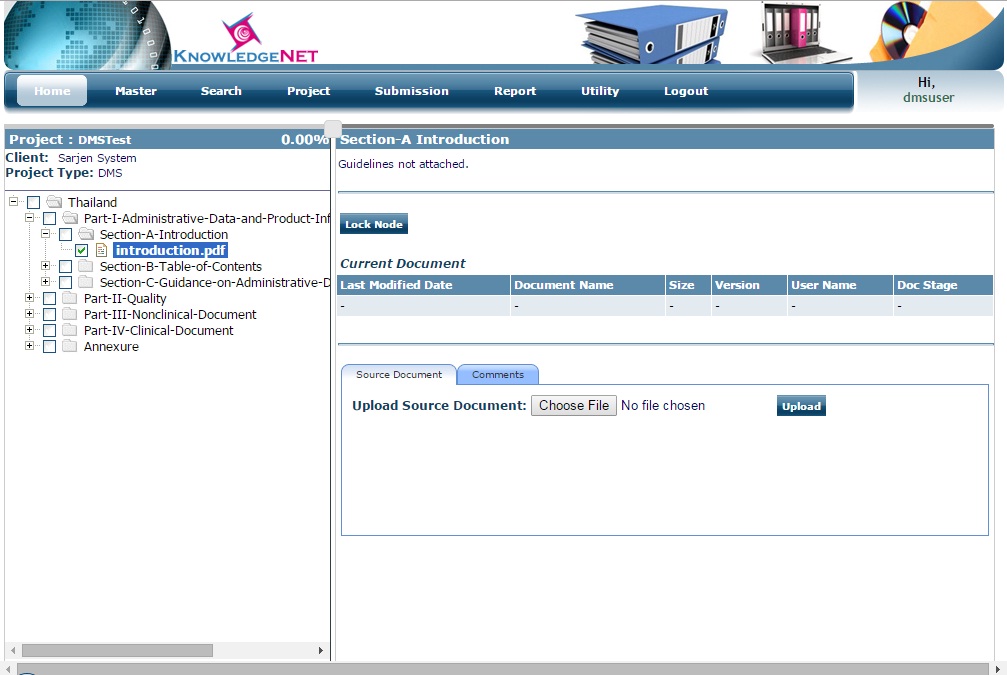
* dmsuser is successfully attached on DMSTest project

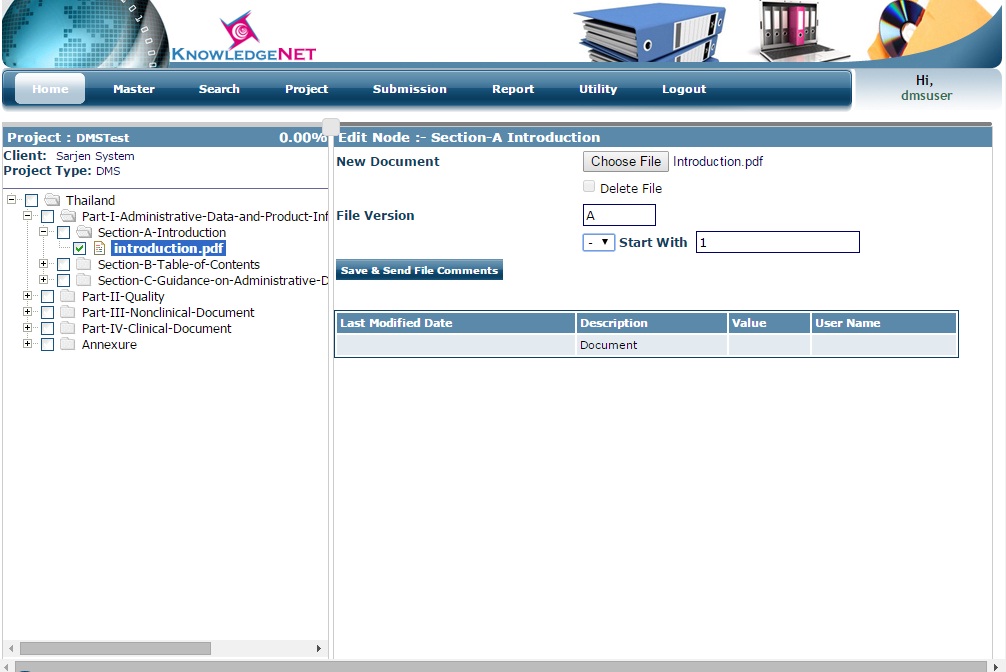


**Manage Project:** Working user will be mange project

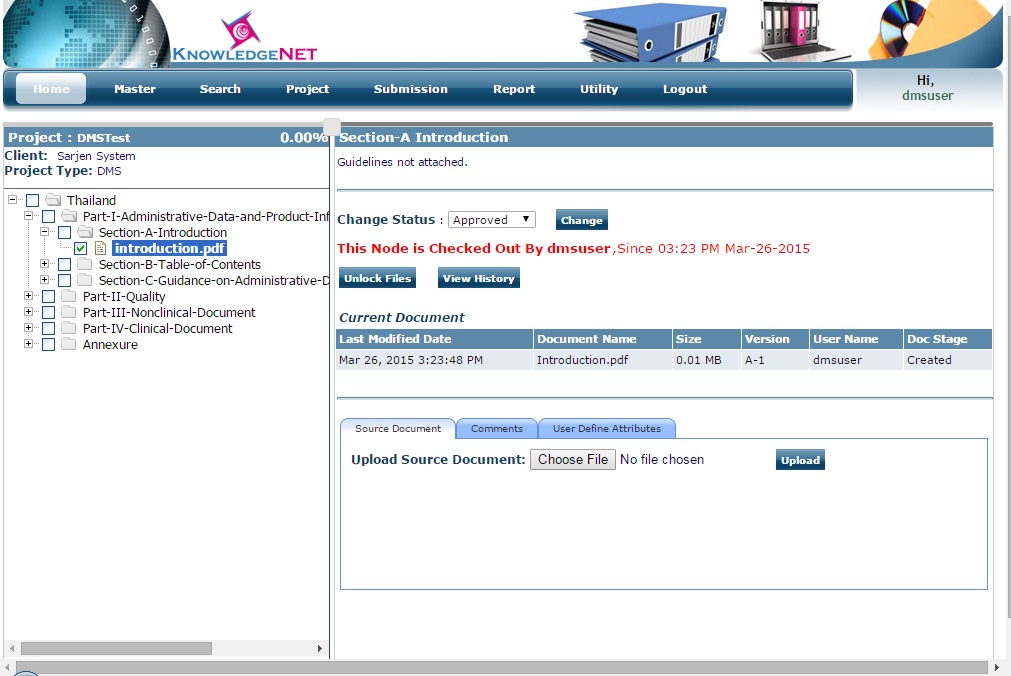
* Manage project in which user need to enter project name, client name, region name so assign project will be open in workspace.



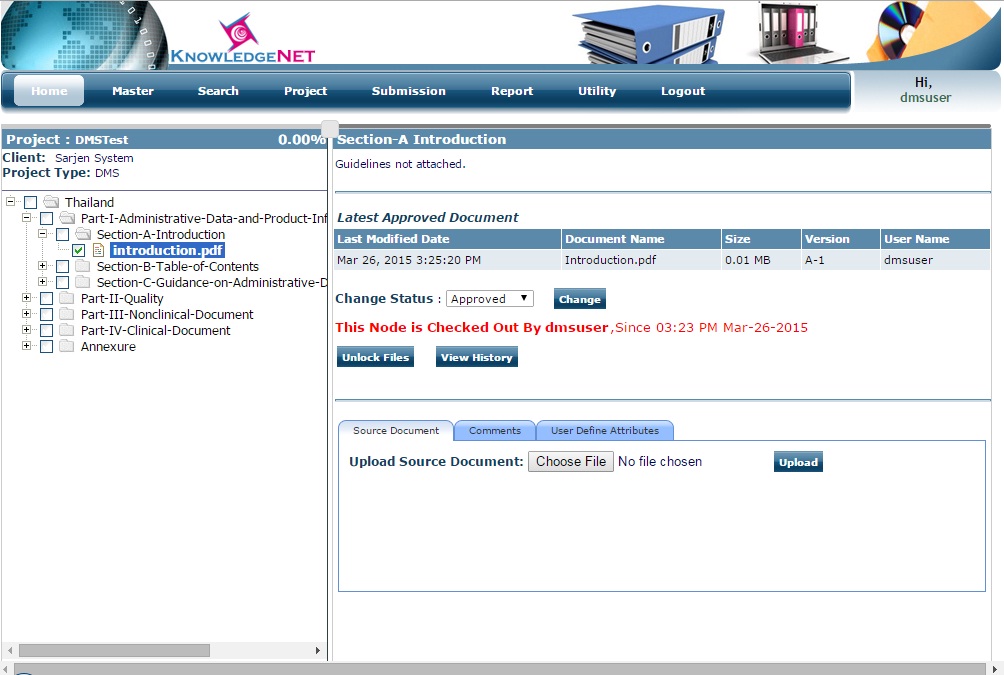
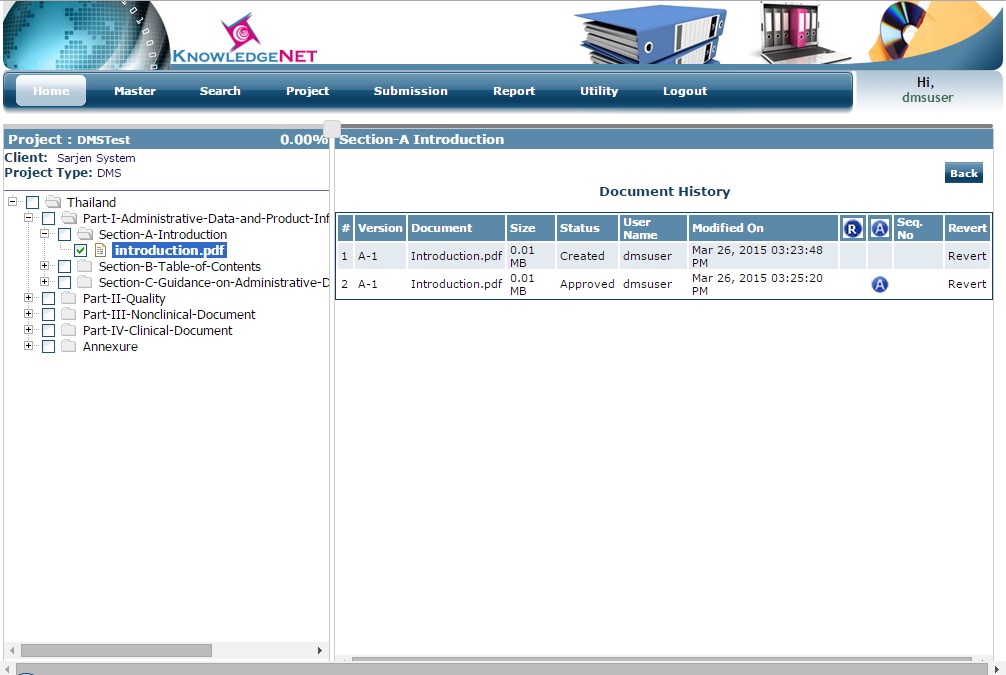
* Selected project will be open in the workspace
* You need to select leaf node to attach file on particular node here introduction.pdf node is selected and click on lock node to attach file
* After selection of introduction.pdf form required path click on save & send file comments so that file is uploaded to the introduction.pdf node

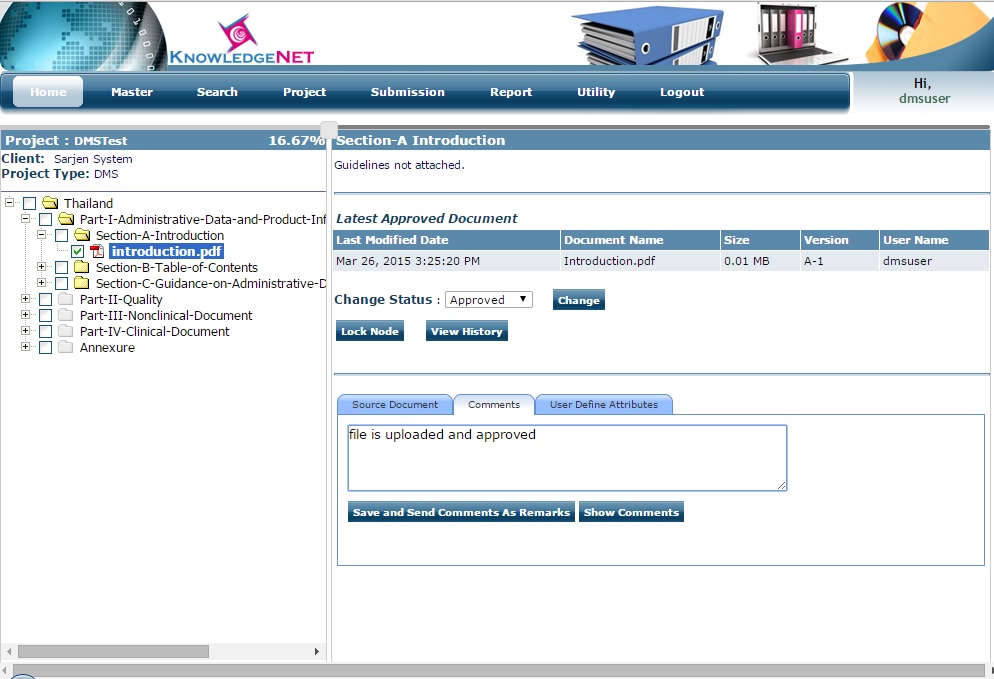


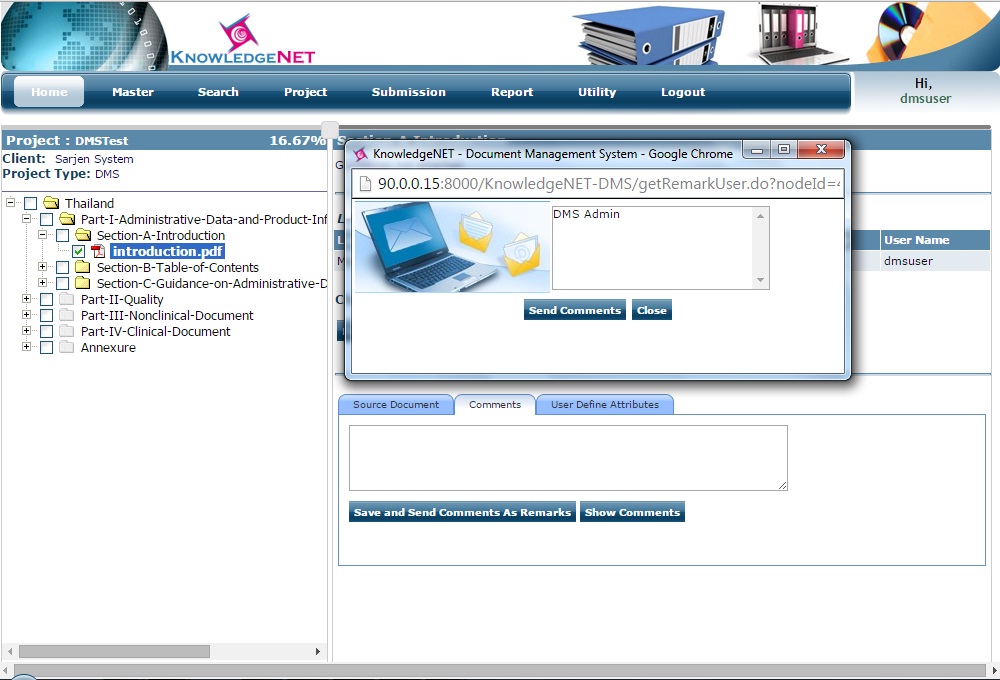
* File is successfully uploaded in the indroductoin.pdf node and change status of upload file to approved click on change button so file is approved by dmsuser.



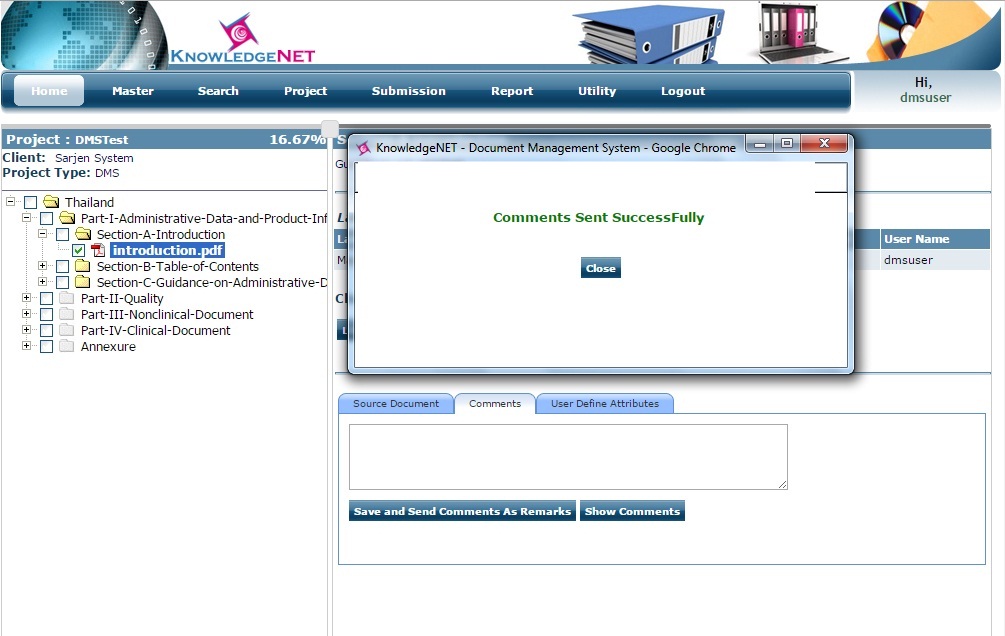
* File is approved by dmsuser successfully and click on unlock file button so that selected node is unlock.



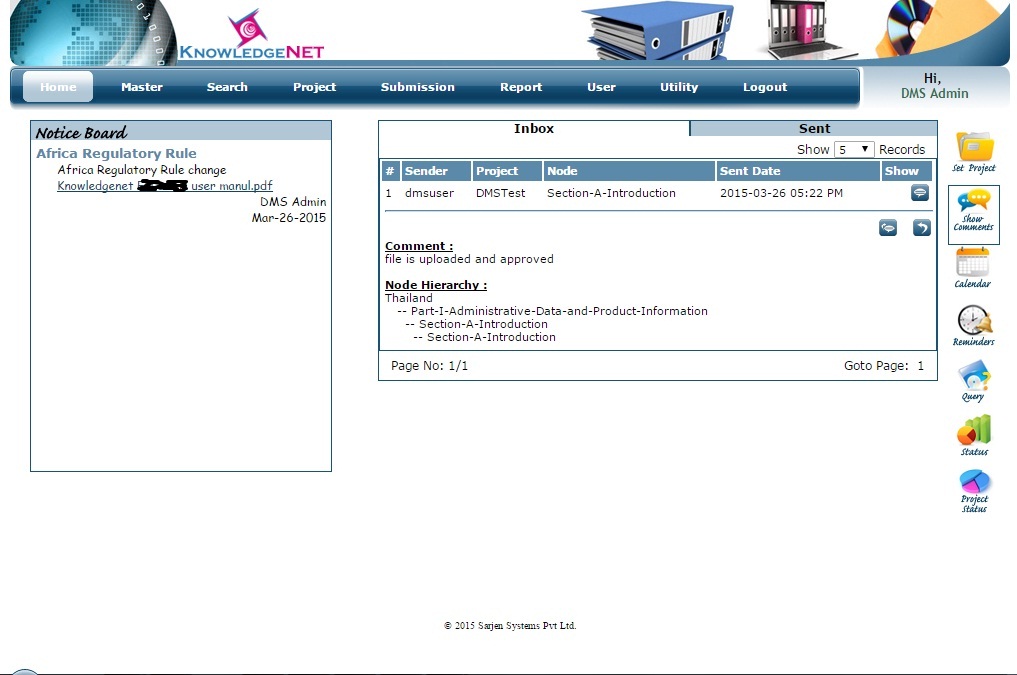
* Selected node is unlocked and to show the history of selected node click on view history button which show you all attached file and status of file
* User can send comments of uploaded file click on save and send comments as remark below screen display
* Then user need to select which users or admin to send comments

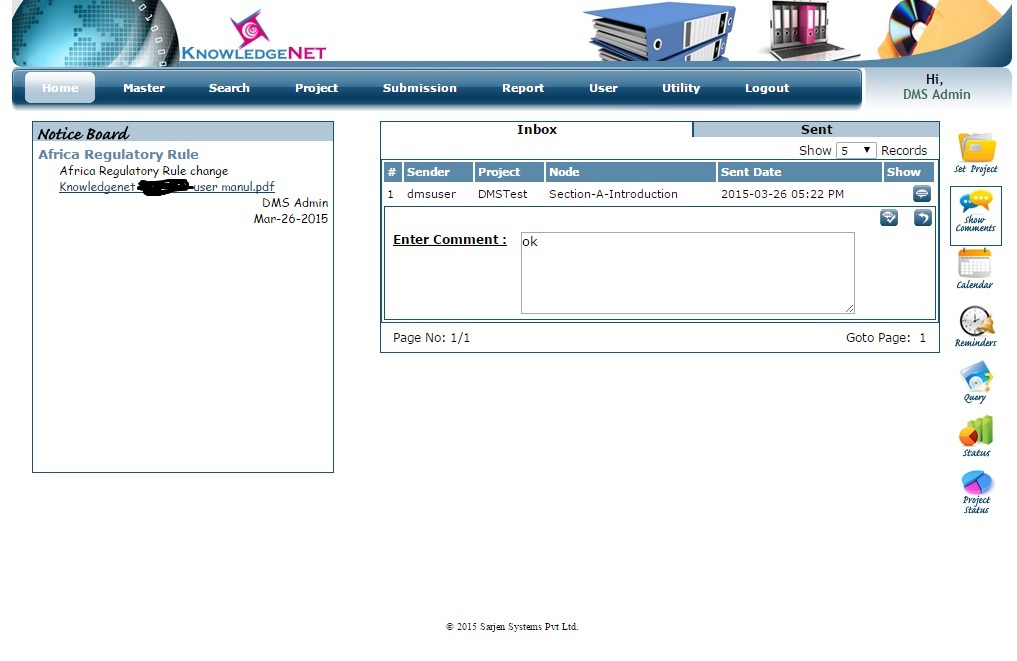
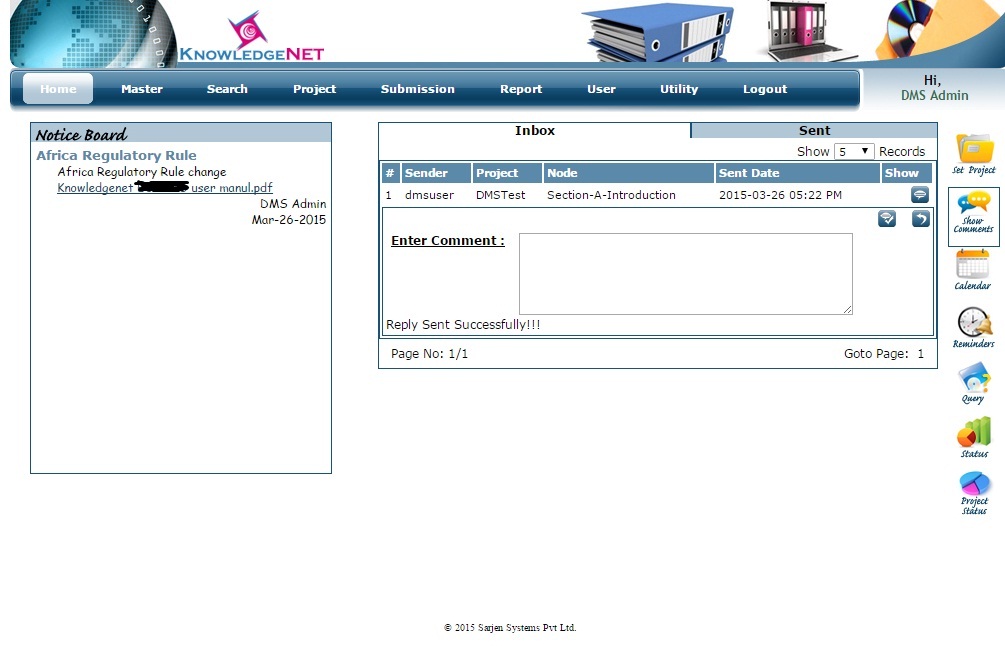


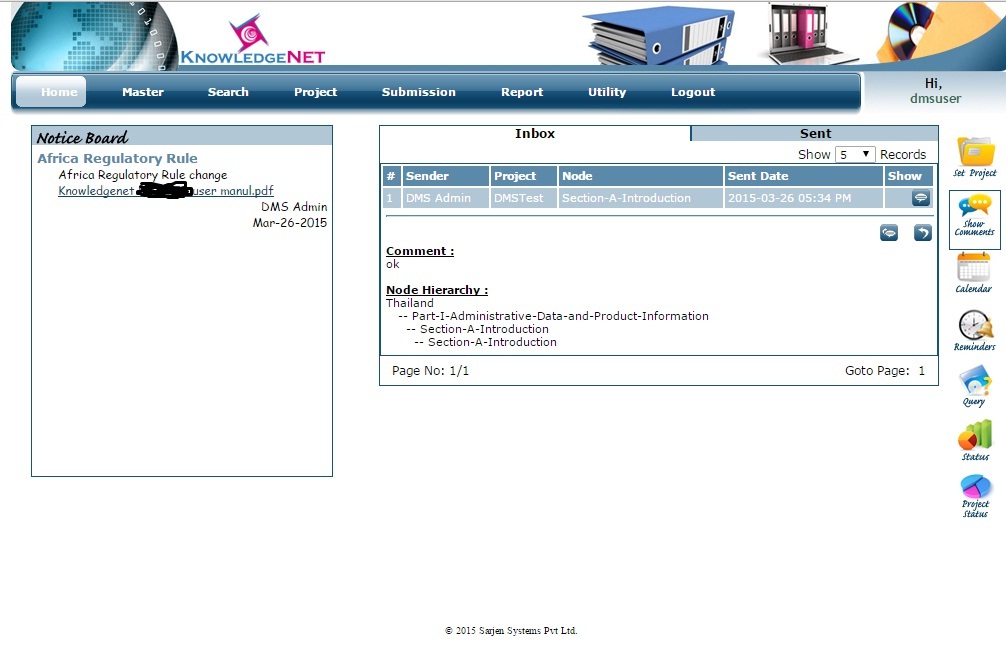
* After successfully comment send below screen display



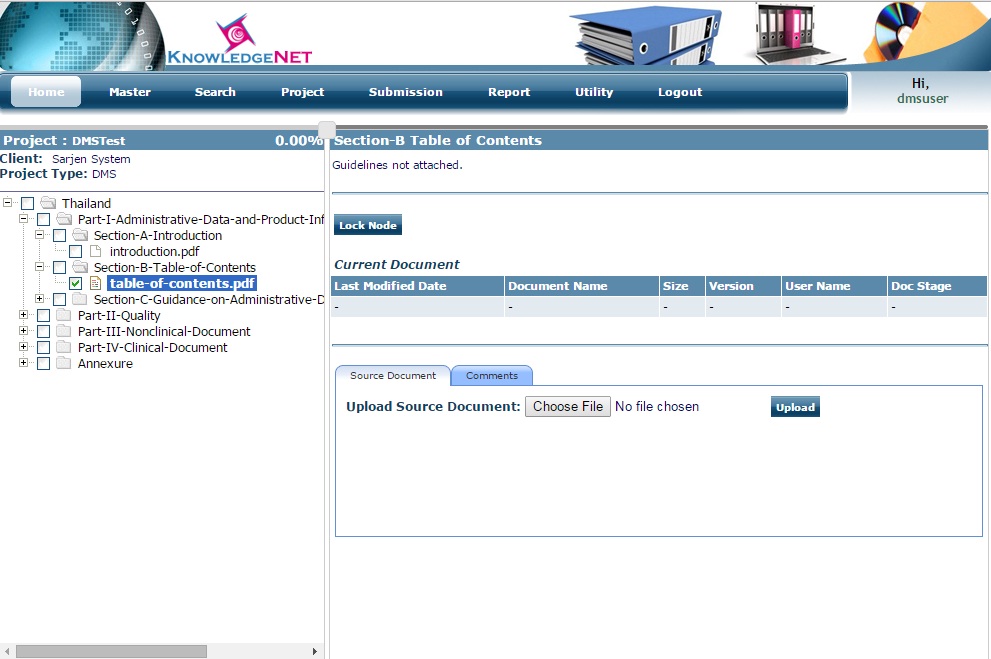
* Comment send by user will be display on home page in which show comments tab is shown on the right hand side click that tab to show all comments

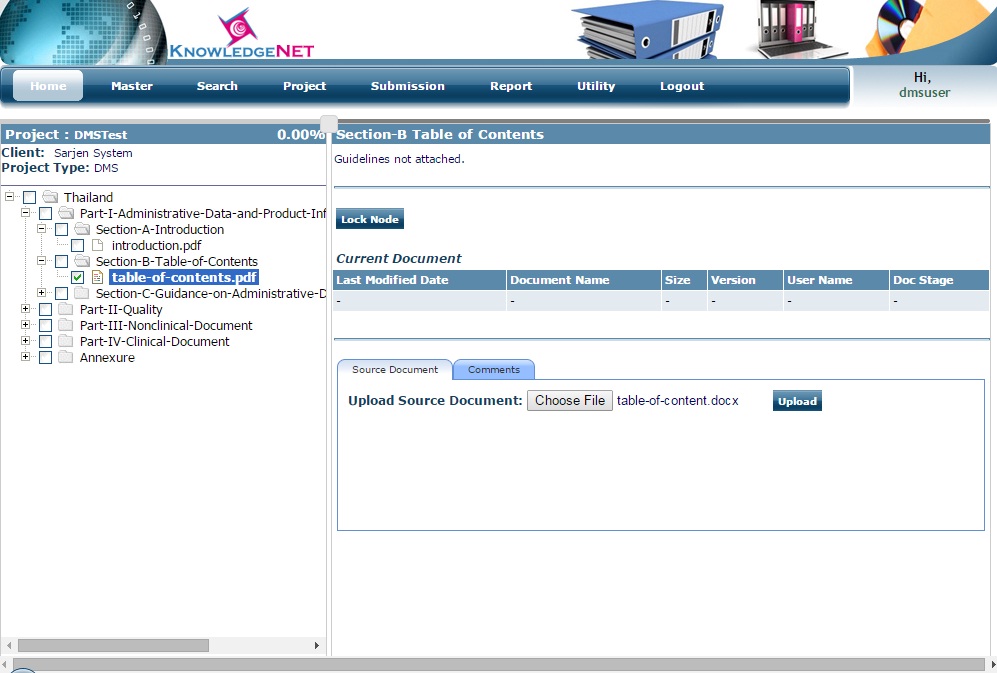


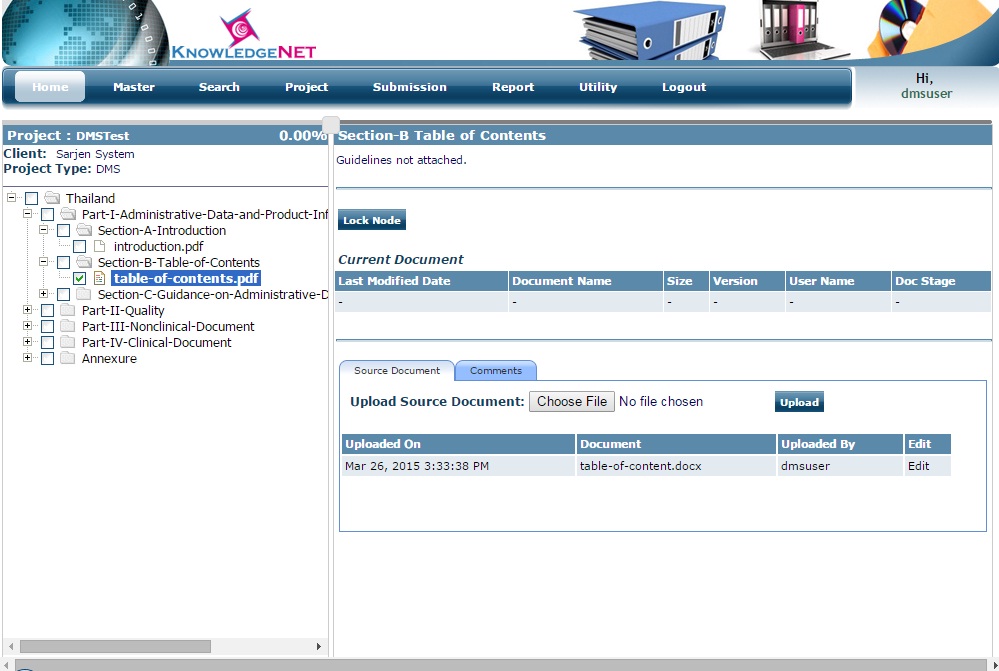
* Dms admin will replay received comment to by ok message to the dms user
* replay send successfully to the dms user
* dms user get replay message send by dms admin



* If you need to attach document file on selected node then first you need to upload source document and click on upload

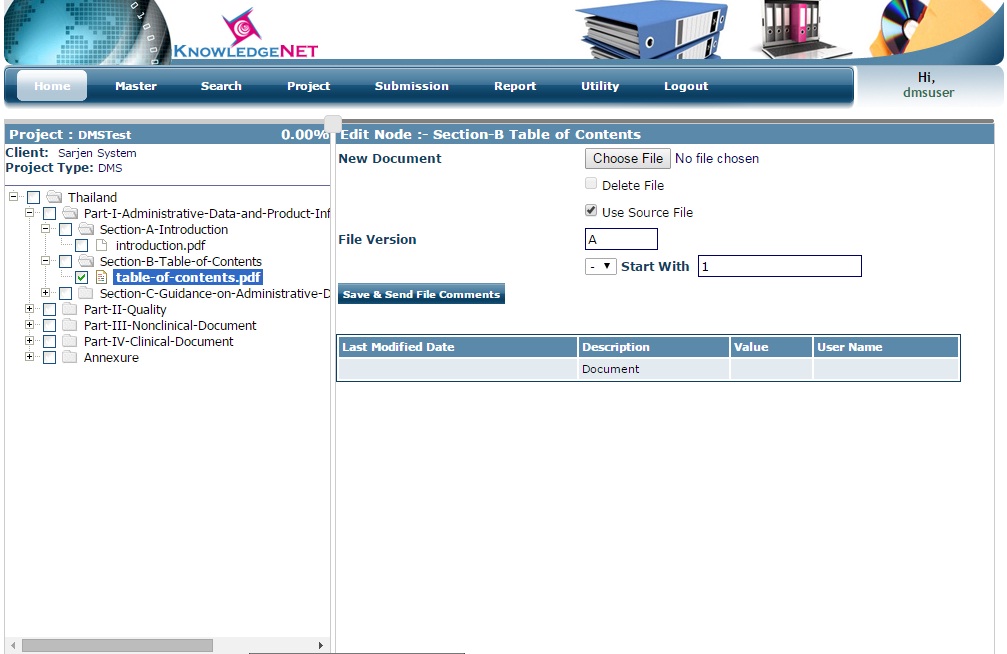


* Choose document file path and upload file
* Click on upload button file is successfully uploaded

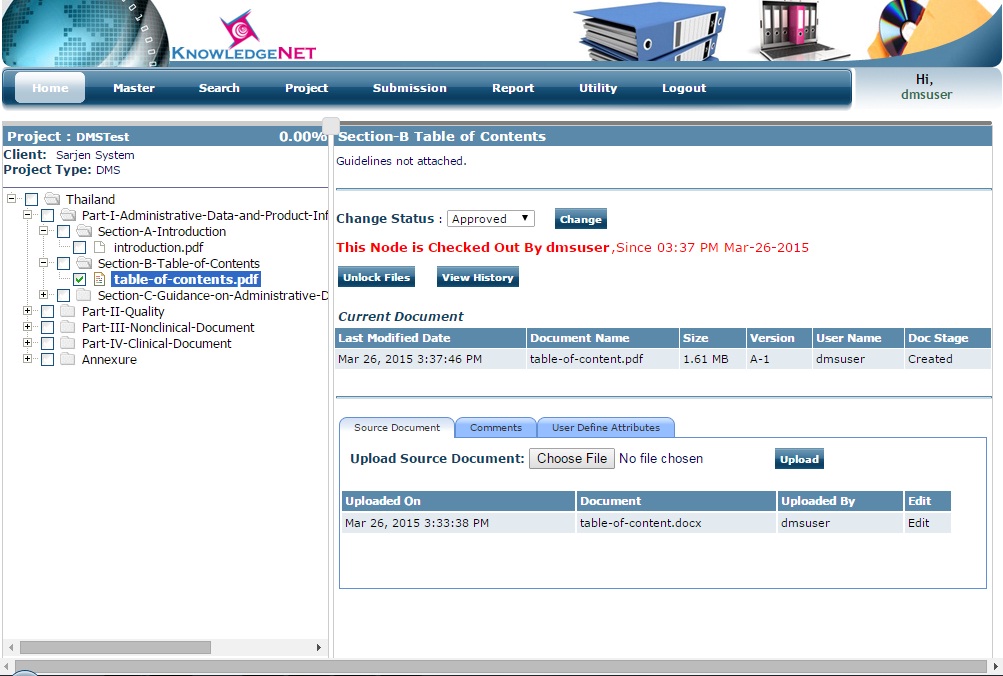


* Select table-of-contents.pdf node and click on lock node to attach document file

Checkmark use source file and click on save & send file comment to attach file

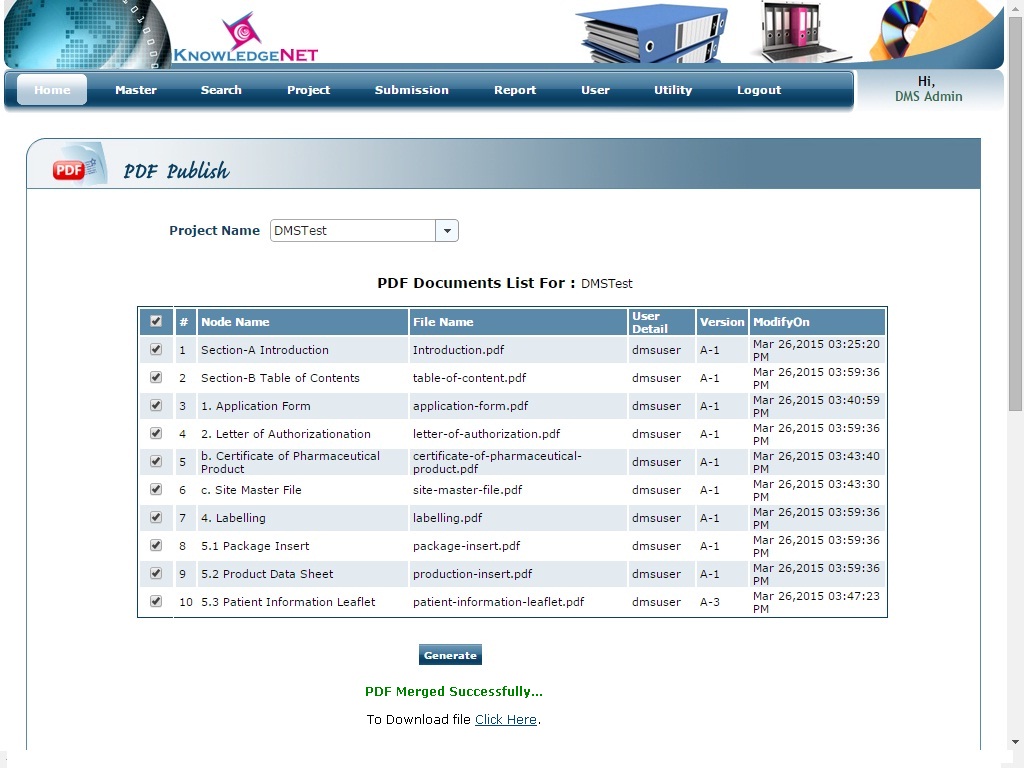


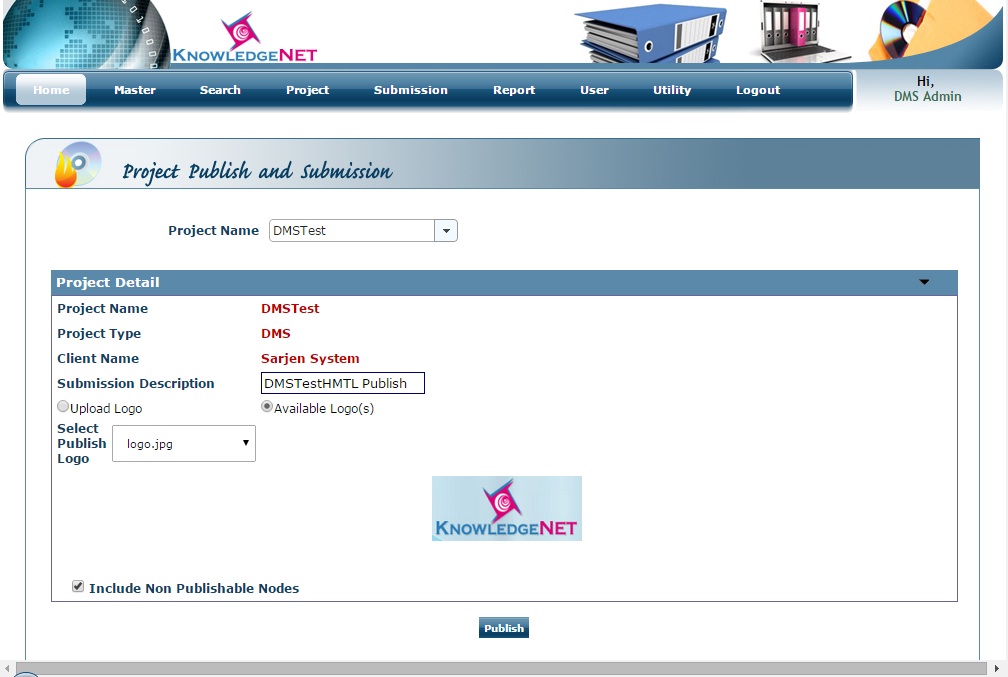
* Table-of-content file is uploaded successfully on table-of-content node.

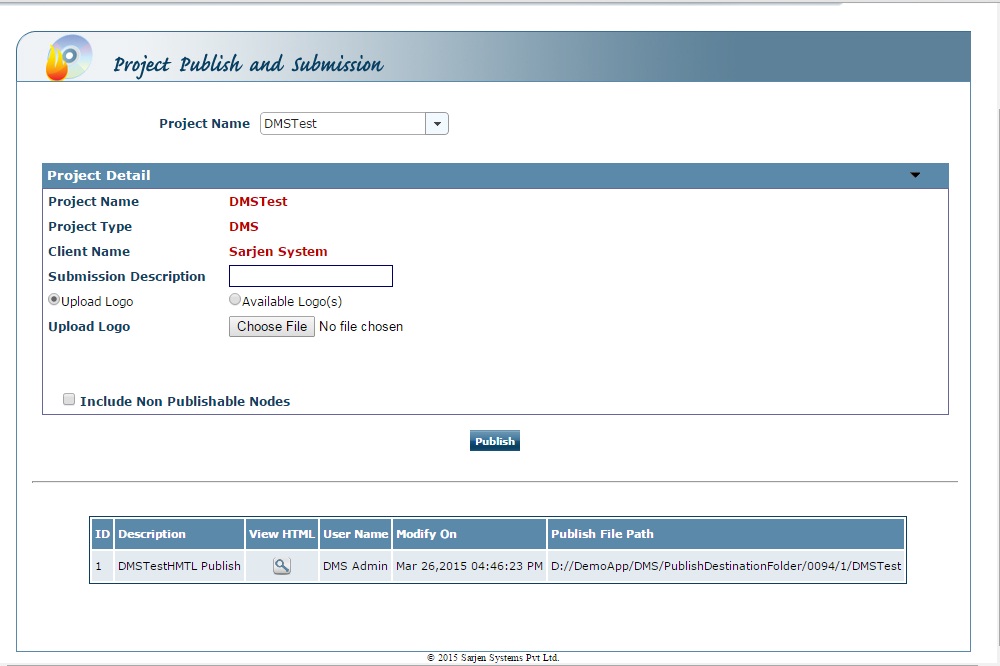


**Submission Menu:** PDF Publish,HTML Publish

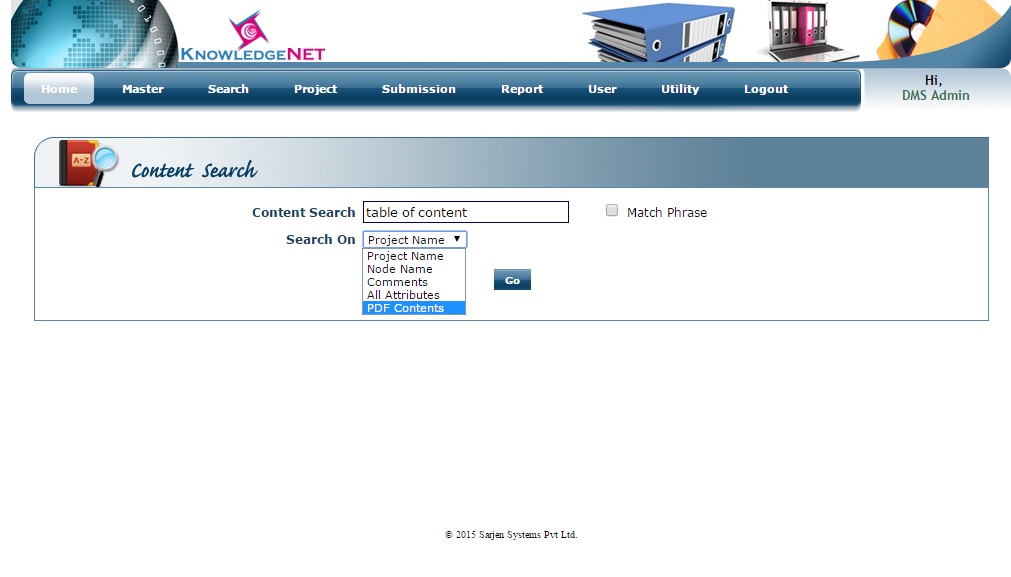
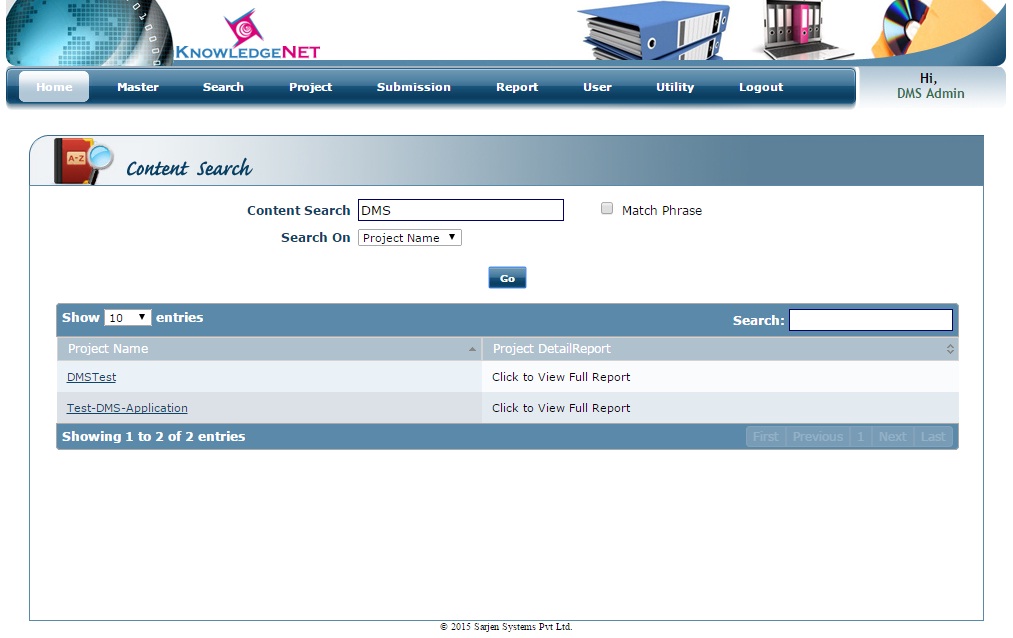
* PDF Publish: Select Project you need to PDF Publish, all uploaded files are merged and generate single pdf file.

****

* HTML Publish: select project name and fill-up required information and click on publish button to generate HTML Publish
* HTML Publish generated on publish file path director

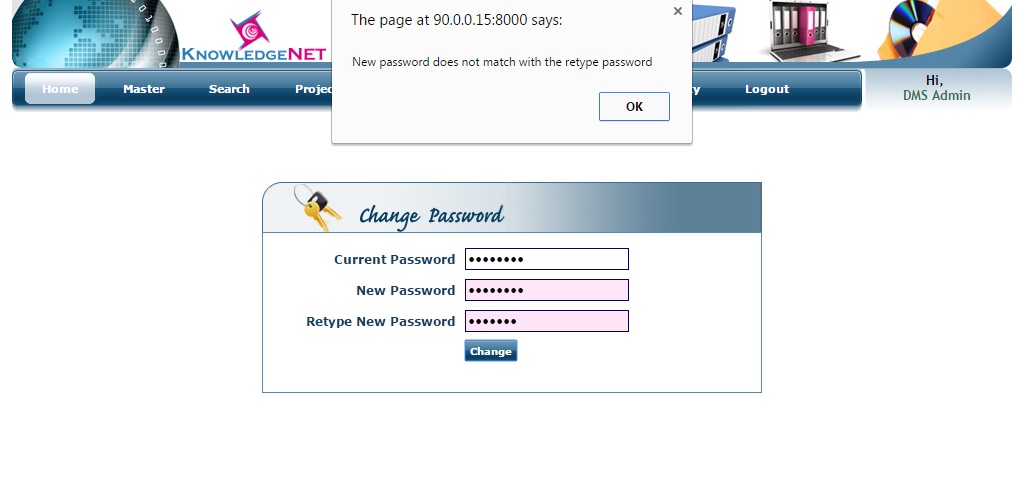


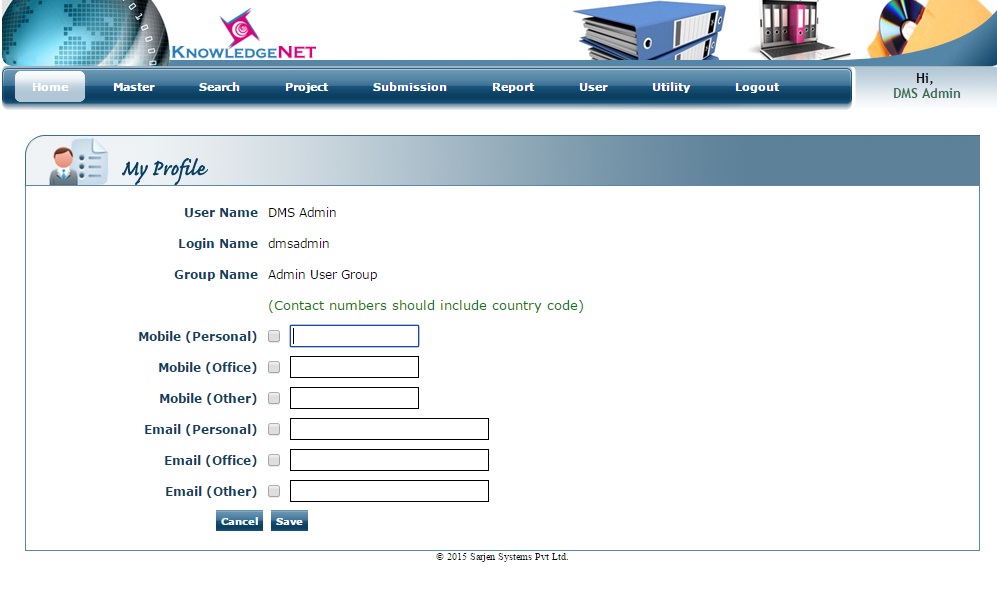
**Search Menu:**By content

* You can search by project name, node name, comments, all attributes, and pdf contents
* Here Content search by project name and display all match content project name

**Utility Menu:**Change Password, My Profile

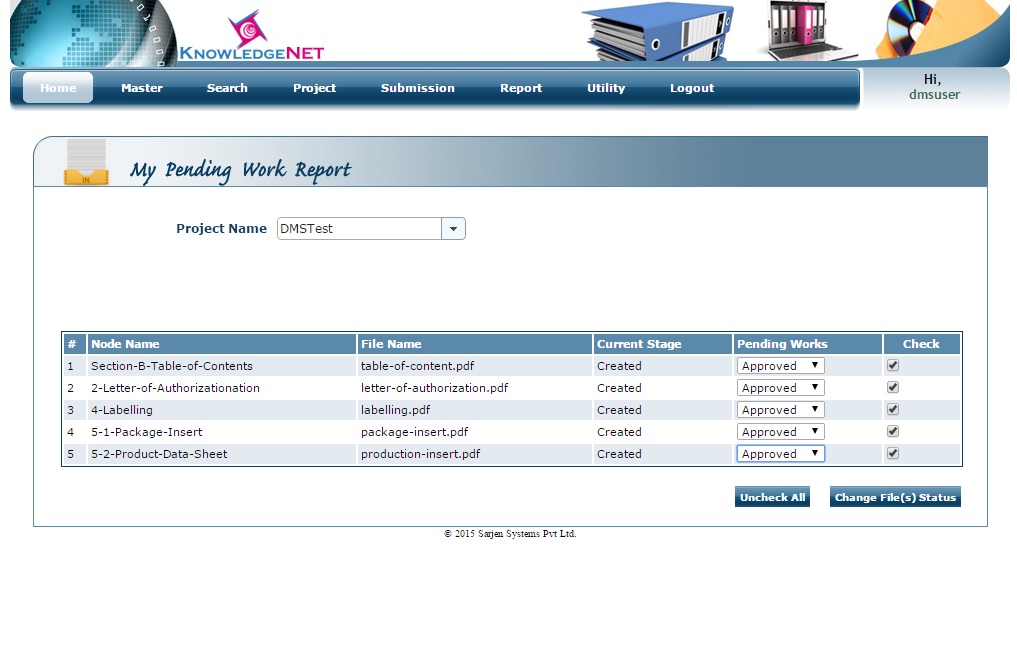
* Change password: in which you can change password



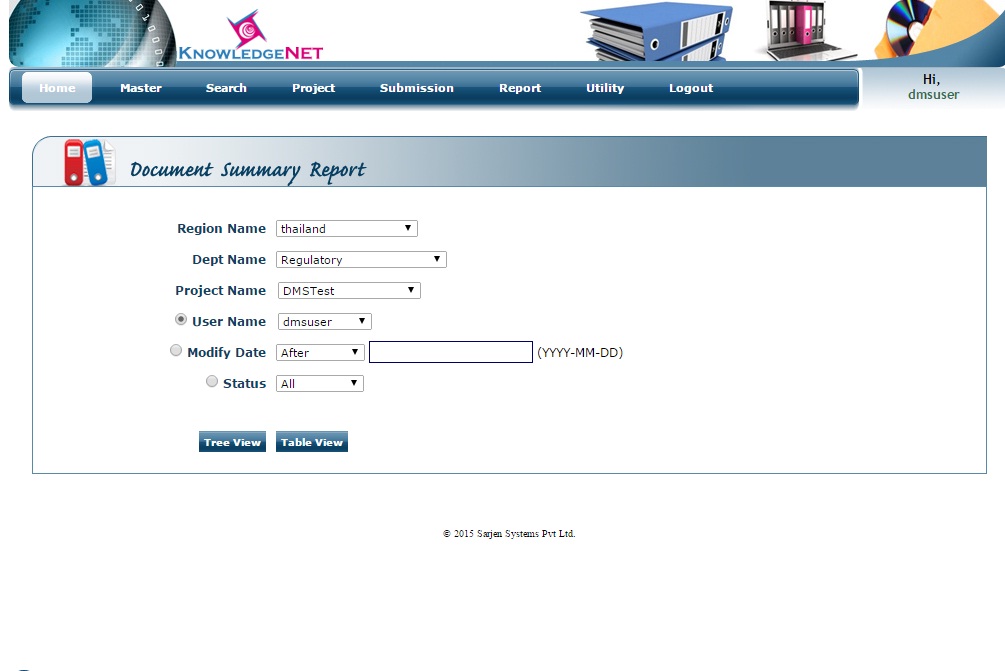
* My Profile: in which you can see your profile details and click on edit button to edit profile details
* Enter required information to edit profile details and click on save button to change details.

**Reports:**

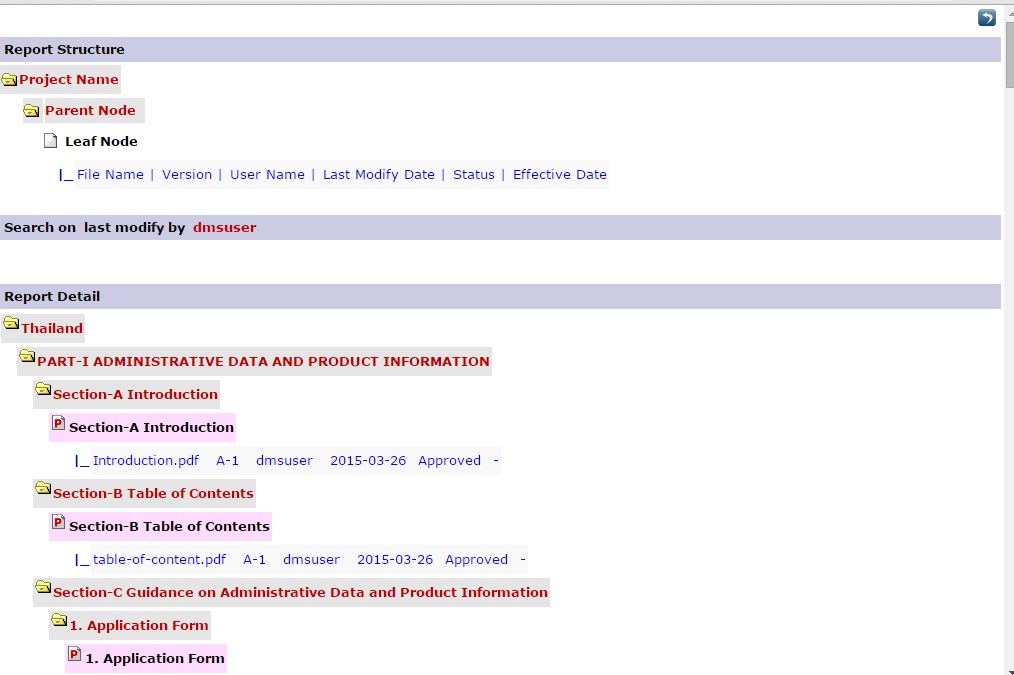
* My Pending Work:
* Document Summary:
* My pending work: in which user need to select project it will display all pending attached file which is not approved so that you can approved all files using checkbox one by one or all the files and click on change file status



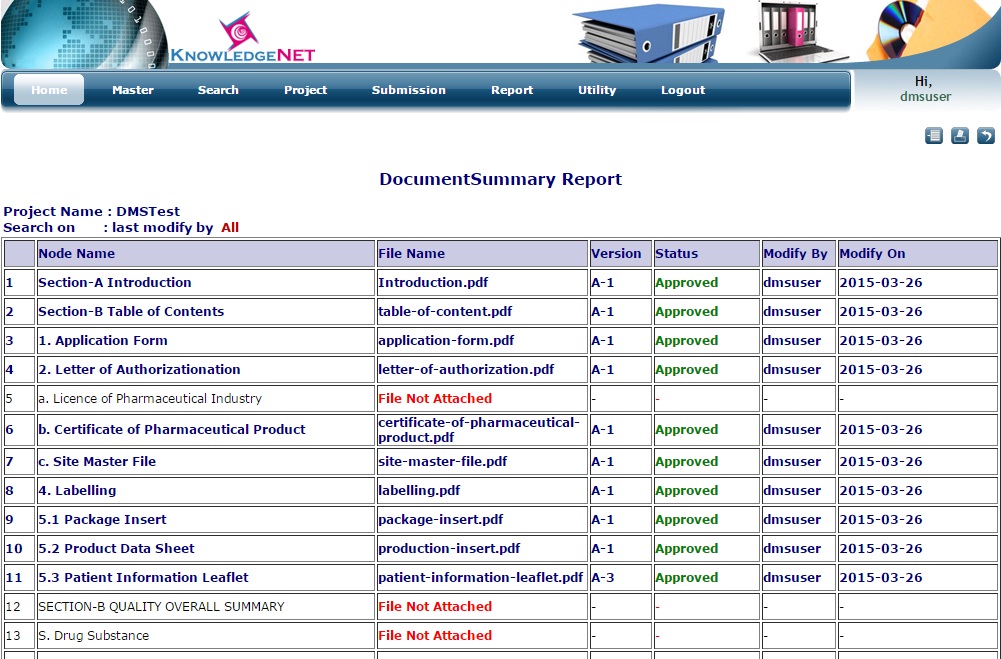
* Document Summary: in which you can see the selected project details, which file is approved, which file is not approved, which node contain which file and which node is empty in tree view and table view report



* Tree view report: display all node details which node contains which file or empty node, status of the file on selected node



* Table view: in which selected project details shown in tabular form which contains node name, file name, version of file, status of uploaded file, modify by, and modify date.



**Test Cases:**

1. **Login validation**

I. Test Purpose

This test script will make ensure that all login validations are available.

II. Test Prerequisites

User has access to all required functionality.

III. Acceptance Criteria

All steps of the test script must produce the specified outcome.

IV. Test Execution

|  |  |  |
| --- | --- | --- |
| **Test Step** | **Expected Results** | **Does Observed Result match Expected Result?**  **Y/N( if N give details)** |
| **1.Open KnowledgeNet**  - Open a web browser  - Enter KnowledgeNet application link  - Enter username as ‘admin’  - Enter ‘password’ in the Password field.  - Button icon ***Login*** | (KnowledgeNet login window opens)  (Home Page Appears) | Y |
| **2.Enter Wrong User Name or Password**  -Enter wrong user name or password  -Click on Clear button  -Enter User Name as “admin1” and password as “a1234” | Message appears “**Invalid user name or password! Please try again!**”  The text box in front of label username and password gets cleared  Home Page appear | Y |

V.Test Status

PASS: FAIL: Other (gives details)

VI. Test Comments and Personnel

Comments:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Executed by: Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Witnessed by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2)Add Region Test Case**

I. Test Purpose

This test script will ensure that a new region can be added, edited, inactivated and activated.

II. Test Prerequisites

User has access to all required functionality.

III. Acceptance Criteria:

All the steps in the test script must produce the specified outcome.

IV. Test Execution

|  |  |  |
| --- | --- | --- |
| **Test Step** | **Expected Results** | **Does Observed Result match Expected Result?**  **Y/N( if N give details)** |
| **1.Open KnowledgeNet**  - Open a web browser  - Enter the KnowledgeNet link  - Enter username as ‘**admin**’  - Enter ‘password’ in the Password field.  - Click on ***Login*** button.  - Under Master menu, click on the **‘Region’** link.  -Enter Region Name as ‘**ABCD**’ and click on the **Add** button. | (KnowledgeNet login window opens)  (Home Page Appears)  A page named **Region Master** appears.  The region name as ABCD will be added to the grid displayed below. | Y |
| **2.ADD another Region**  - Enter a new region name as ‘**EFGH**’.  - Press **Add** button | System should allow to save the data and add a row with the new data as ‘EFGH’ in the grid below. | Y |
| **3.Edit Region Details**  - From grid displayed below, click on the **Edit Icon** against the desired region which needs to be edited.  -Enter the updated value and Click on **Update** button. | A pop up message window will appear with old region name ,new region name and there will be **update** button.  Updated value will be displayed in the grid below. The status is changed to Edited. | Y |
| **-Inactivate Region**  -In the grid displayed click on the **Inactivate Icon** link under the revert column in the grid. | An alert box appears stating **“Do you really want to inactivate selected region?”**  Click on the OK button then the particular record would be displayed in red colour and the Inactivate icon link would be converted to Activate icon link. Also the status is changed to Deleted.  **NOTE: If the Cancel button is pressed in the alert box then no changes takes place in the grid.** | Y |
| **-Activate Region**  In the grid displayed click on the Activate icon link aginst the desired region that needs to be activated again. | An alert box appears stating **“Do you really want to InActivate selected Region?”**  Click on the OK button then the particular record would be again displayed in its original colour and the Activate link would be converted to inactivate link. Also the status is changed to Edited.  **NOTE: If the Cancel button is pressed in the alert box then no changes takes place in the grid.** | Y |

**3) Create Project Test Case:**

I. Test Purpose

This test script will make ensure that Project can be created.

II. Test Prerequisites

User has access to all required functionality.

III. Acceptance Criteria :

All the steps in the test script must produce the specified outcome.

IV. Test Execution

|  |  |  |
| --- | --- | --- |
| **Test Step** | **Expected Results** | **Does Observed Result match Expected Result?**  **Y/N( if N give details)** |
| **1.Open KnowledgeNet**  - Open a web browser  - Enter KnowledgeNet application link  - Enter username as ‘admin’  - Enter ‘password’ in the Password field.  - Button icon ***Login***  -Under Project Detail menu, click on ‘Create Project’ link  -Enter ProjectName as “Abc”  Select any Region from the list, Select any Client from the list.  Select ProjectType “DMS”,  Select Default Admin as “Admin”,  Select TemplateName as “Eu-National-V1.3”,  Select any Department,  Select “Default” as Document Type,  And Click on Create Project | (KnowledgeNet login window opens)  (Home Page Appears)  A page named Create Project appears  A message will be displayed “Abc Created Successfully” | Y |
| **2. ADD same ProjectName again with same location and client**   1. - Enter   ProjectName as “Xyz\*”  Select any Region from the list, Select any Client from the list.  Select ProjectType “DMS”,  Select Type as “dms”,  Select Default Admin as “Admin”,  Select TemplateName as “Eu-National-V1.3”,  Select any Department,  Select “Default” as Document Type,  And Click on Create Project  - Try to add same Project “Abc” with all other details same | A message shall displayed as “Invalid project name. \* is not allowed”  Should not allow the user to add and it should give alert “Project Name already exist” | Y |
| **3.Click on Advance Search**  - click on search project | A page search Admin Project will appear  A list of Project/s created will get displayed in the Grid below | Y |
| **4.Edit Project Details**  Click on the Edit link against the project “Abc”  Change project type to “Default”, likewise change few other details and click on button “Update Project” | A new small window will open up as “Edit Project Details” | Y |
| **5.**5.Click on Logout to logout | Login page will appear | Y |