DETAILED DESIGN - FRONTEND

a. Detailed design for html pages

1. File Name: Login.html

File Description: This page displays a portal for user to login.
Functions present: ValidateLogin()
Mapped to element: Login service (component)
Functional Description:

Function name	ValidateLogin
Function trigger Point	User presses Submit button - onSubmit event gets triggered
Function parameters	void
Function return type	void
Functions description	The function takes the entered values of username and password from the user, It invokes a service LoginService and displays an alert box with a message of “Login Success” or “Login Failure”

Function pseudo code	ValidateLogin()
	{
	//Display fields for entering username and password
	String result = call "LoginService"(username, password)
	if (result.equals("Success")
	Load "home.html"
	else
	alert("Login unsuccessful")
	}

Browser storage variables (Local Storage).	empid - This variable stores the value of the selected user.

2. File Name: Home.html

File description: This file provides the functionality for selecting the projects pertaining to the \user. It also provides facility for the user to select his role in that project. This file also contains a sidebar which will enable the user to switch to any operation that he want to perform.
Functions present: onLoad, onSelectProject, OnSelectRole

Mapped to element: This file runs in the browser process
Function Descriptions:

Function name	onLoad
Function parameters	void
Function return type	void
Functions description	The function would be triggered once the login is successful. This function displays a list of projects corresponding to the user logged in to the system. In order to fetch the list of projects, a service call to the “FetchProjects” servlet would be made.Get the details of the user logged in from the local data storage and pass it as a parameter to the service call
Function trigger Point	On successful login of the user (Page on load)

Function pseudo code	OnLoad()
	{
	// Make a service call to fetch the list of projects. This should be displayed as a drop down to the user.
	
	String result = call "FetchProjects"(username)
	Object selectProject=JSON.parse(result);
	Create a drop down selection box and append the values from the JSON object.
	//
	}

Function name	onSelectProject
Function parameters	void
Function return type	void
Functions description	This function displays a list of roles corresponding to the user and the project he has selected. In order to fetch the list of roles, a service call to the “FetchRoles” servlet would be made. Get the details of the user logged in and the project selected from the local data storage and pass it as a parameter to the service call
Function trigger Point	The function would be triggered once submit project button is selected.
Function pseudo	onSelectProject()

code	{
	// Make a service call to fetch the list of roles.This should be displayed as a drop down to the user.
	
	String result = call "FetchRoles"(project,username)
	Object selectRoles=JSON.parse(result);
	Create a drop down selection box and append the values from the JSON object.
	
	}

Browser storage variables	selectedProject - This variable stores the value of the selected project.

Function name	onSelectRole
Function parameters	void
Function return type	void
Functions description	This function should be used to store the selected role in the local storage such that it can be used in later pages.
Function trigger Point	The function would be triggered once submit role button is selected.

Function pseudo code	onSelectRole()
	{
	//Store the selected value in local storage
	}

Browser storage variables	selectedRole - This variable stores the value of the selected role.

3. File name-CreateInstances.html

File description-This file would present the list of elements to the user and prompt him to choose an entity for creating instances for it.
Functions present: getEntities() ,entityOnClick().
Mapped to element: Browser(component)
Functional Description:

Function name	getEntities()
Function Trigger point	When createInstance.html page's body gets loaded in the browser.

Function input	Projectname, Role
Function return type	Void
Functions description	This function will pass Projectname and role name as input and invoke the “fetchEntity” service. This function will store the output received from the “fetch entity service in a string. It will then parse the string and store it into a java script object named “selectEntity”. It will display the list of entities to the user in a drop down box.

Function pseudo code	function getEntities()
	{
	Var result=xmlhttprequest(“url&?projectname&?role);
	Object selectEntity=JSON.parse(result);
	//create a drop down box and populate the values of entities in it//
	}

Function name	entityOnClick()
Function Trigger point	When the user clicks on the submit button the selected element will be saved in a javascript object.
Function input	-
Function return type	Void
Functions description	This function will invoke the file “**ShowInstances**.html”

Function pseudo code	function entityOnClick()
	{
	//invoke the **ShowInstances**.html“ page//
	}

Browser storage variables	selectedEnity - This variable stores the value of the selected entity.

4. File Name : ShowInstances.html

File Description: This page shows the instances created by the user for the selected entity.
Functions present: getInstances(), enableEdit(), onEditInstance()
Constraints: The first column should be Instance Name or ID such that it is unique. Since this is the attribute that will be saved as unique id in the browser variable. In further pages will be using this variable to access the instance.

Functional Description:

Function name	getInstances
Function trigger Point	User has selected his/her preferred entity-onLoad event is triggered
Function parameters	void
Function return type	void
Functions description	The function takes the entered values of project name, user name and entity name and invokes the FetchInstances service. This service fetches the list of instances created by the user for that chosen entity from the instances collection and displays the instances created by that user along with radio button for each instance.

Function pseudo code	getInstances()
	{
	String result = call "FetchInstances" (entity_name, project_name, user_name)
	\\ Display the list of instances created by the user
	Object Instances=JSON.parse(result);
	
	\\ User clicks on add button instance
	addInstance.onClick()
	Load "addInstance.html"
	
	\\ User selects a particular instance by clicking on a radio button and selects edit instances
	If instance.onChange()
	editInstance.onClick()
	Load "modifyInstance.html"
	}

Function name	enableEdit
Function trigger Point	On selecting one of the instances radio button
Function parameters	void
Function return type	void
Functions description	This function will enable the Edit Instance Json button.

Function pseudo code	void ()
	{
	//Get the button element
	//set the disabled property to false

Function name	onEditInstance
Function trigger Point	User has clicked the EditInstance button
Function parameters	void
Function return type	void
Functions description	This function stores the selected instance in the browser storage

Function pseudo code	onEditInstance(){
	//Get the selected instance value
	//store it in the selectedInstance browser storage
	//move to EditInstance.html
	}

Browser storage variables	selectedInstance - This variable stores the value of the selected instance.

5. File name- AddInstances.html

File description-This file would allow the user to add the instances for a particular entity that he has chosen.
Functions present: onEntry(), saveInstance()
Mapped to element: Browser(component)
Functional Description:

Function name	onEntry()

Function Trigger point	This function will be invoked on load of the page.
Function input	-
Function return type	Void
Functions description	The function will pass the project name and entity name as an input to the fetchInstance service. The function fetches the properties json for the particular entity as a string. The string is parsed as a json and a form is created dynamically by way of a jquery dform plugin.

Function pseudo code	function onEntry()
	{
	Var result=xmlhttprequest("url&?entityname?projectname);
	Object getEntityProperties=JSON.parse(result);
	//create a form dynamically based on the attributes of the getEntityProperties
	}

Function name	saveInstance()
Function Trigger point	When the user clicks on the Add instance button.
Function input	void
Function return type	void
Functions description	The function gets all the form values by using the serialize method and invokes the InsertInstance service. The service then updates the getEntityProperties JSON to add the values for the respective fields and performs an insert into the database. It will display in a alert box stating that the" save operation has been done successfully" if it received positive boolean value from the "insertInstance" service or else it would display that "Instance not saved" message in the dialog box.
Function pseudo code	function saveInstance() { Var result = $(this).serialize();

	Boolean flag=xmlhttprequest(“url&?result);
	if(flag)
	{
	alert(“save operation has been done successfully”);
	}
	Else
	{
	alert(“Instance not saved”);
	}
	}

6. File name- EditInstances.html

File description-This file would allow the user to edit the instances for a particular entity that he has chosen.
Functions present: editInstance(), saveInstance()
Mapped to element: Browser(component)
Functional Description:

Function name	editInstance()
Function Trigger point	This function will be invoked when the page is loaded
Function input	nil
Function return type	Void
Functions description	This function will pass instance_name as input and invoke the “fetchInstance” service. This function will store the output received from the “fetchInstance” service in a string. It will then parse the string and store it into a javascript object named “selectInstance”. It will then dynamically create a form with pre populated values from the attribute present in the selectInstance object.

Function pseudo code	function editInstance()
	{
	Var result=xmlhttprequest(“url&?instancename);
	Object selectInstance=JSON.parse(result);
	//create a form dynamically based on the attribute values of the selectInstance object with pre populated valued//
	}

Function name	saveInstance()
Function Trigger point	When the user clicks the save instances button
Function input	void
Function return type	void
Functions description	This function will convert the form values into a string using the serialize() method and the resultant string is passed to the “UpsertInstance” service. The UpsertInstance service performs the upsert of the instance object JSON to the database. It will display in a dialog box stating that the” save operation has been done successfully” if it received positive boolean value from the “UpsertInstance” service or else it would display that “Instance not saved” message in the dialog box.

Function pseudo code	function saveInstance()
	{
	Var result=$(this).serialize();;
	Var flag=xmlhttprequest(“url&?result);
	if(flag==”success”)
	{
	alert(“save operation has been done successfully”);
	}
	Else
	{
	alert(“Instance not saved”);
	}
	}

7. File name- LinkInstances.html

File description-This file would allow the user to link instances for a particular instance that has been created by him.
Functions present: onLoad(), onEntityClick(), onSubmit()
Mapped to element: Browser(component)
Functional Description:

Function name	onLoad()
Function Trigger point	This function will be invoked when the page is loaded
Function input	nil
Function return type	Void
Functions description	The function passes the role of the person as an input to the FetchEntities() service, which correspondingly loads the entities for which that role is authorized. The list of entities are loaded in a tree format.

Function pseudo code	function onLoad()
	{
	Var result=xmlhttprequest("url&?role);
	//Display the result in the form of a tree format with checkboxes aside each entity.
	}

Function name	onEntityClick()
Function Trigger point	This function will be invoked when the user clicks on a particular entity.
Function input	nil
Function return type	Void
Functions description	The function passes the entity name to a service called FetchInstance() and FetchMappedEntities(). The FetchInstance() service fetches the instances pertaining to the clicked entity created by that particular user. The FetchMappedEntities() populates the list of entities to which the selected entity can be linked. When the entity name is clicked, the corresponding instances can be viewed, again by a call to FetchInstance().

Function pseudo code	function onLoad()
	{
	Var result=xmlhttprequest("url&?entity);
	//Display the list of instances that are created by

	the particular user for that entity in a list format.
	Var result2=xmlhttprequest("url&?entity);
	//Display the list of mapped entities pertaining to an entity in a list format.
	}

Function name	onSaveClick()
Function Trigger point	This function will be invoked when the user clicks on the Save button.
Function input	nil
Function return type	Void
Functions description	The function invokes the UpsertInstance() service, which takes an instance name as input and updates the links of that instance.

Function pseudo code	function onSaveClick()
	{
	Var result=xmlhttprequest("url&?instance);
	//Updates the links of instances
	}

8. ViewModel.html

File name- ViewModel.html
File description-This page would display the model for the selected project
Functions present: loadModel(), entitySelect()
Mapped to element: Browser(component)
Functional Description:

Function name	loadModel()
Function Trigger point	This function will be invoked when the page is loaded
Function input	nil
Function return type	Void

Functions description	This function will pass project name (browser variable) as input and invoke the “fetchModel” service.

Function pseudo code	function loadModel()
	
	{
	Var result=xmlhttprequest(“url&?projectname);
	Object model=JSON.parse(result);
	//create tree structure as depicted in the UI prototype and display it//
	}

Function name	entitySelect()
Function Trigger point	This function will be invoked when the user clicks an entity from the model (onclick entity).
Function input	nil
Function return type	Void
Functions description	This function will store the selected entity in the browser variable (selectedentity) and redirect to ViewTraceablity.html.

Function pseudo code	function entitySelect()
	{
	//retrieve the selected entity
	//Store it in the browser variable
	// Move to new page ViewTraceability.html
	}

9. ViewTraceability.html

File name- ViewTraceablity.html
File description-This page would allow the user to view the traceablity for the selected entity from the model.
Functions present: loadParChild(), loadInstances(), instanceOnLeftClick(),

instanceOnRightClick()
Mapped to element: Browser(component)
Functional Description:

Function name	loadParChild()
Function Trigger point	This function will be invoked when the page is loaded
Function input	nil
Function return type	Void
Functions description	This function will fetch the values of parent and child model for the selected project name using FetchParentChild service. Then it populates in the corresponding fields.

Function pseudo code	function loadparChild()
	{
	//get the selected project (from browser variable)
	//invoke FetchParentChild service to fetch results
	// Display them in the appropriate fields
	}

Function name	loadInstances()
Function Trigger point	This function will be invoked when the page is loaded
Function input	nil
Function return type	Void
Functions description	This function will fetch the instance values for the selected entitity using the FetchInstances Service

Function pseudo code	function loadInstances()
	{
	//get the selected entity(from browser variable)
	//invoke FetchInstances service to fetch results
	// Display them in the appropriate tree as given in prototype
	}

Browser storage variables	selectedAllInstances

Function name	instanceOnLeftClick()
Function Trigger point	This function will be invoked when the user left clicks an instance (onclick entity).
Function input	nil
Function return type	Void
Functions description	This function will display the linked instances for the corresponding selected instance in tree structure

Function pseudo code	function instanceOnLeftClick()
	{
	//get selectedAllInstances and traverse to get the selected instance
	//display the entity and instance names as given in the prototype tree structure.
	}

Function name	instanceOnRightClick()
Function Trigger point	This function will be invoked when the user right clicks over an instance .

Function input	nil
Function return type	Void
Functions description	This function will store the selected instance in browser variable and redirect to the next page ExportPDF.html

Function pseudo code	function instanceOnRightClick()
	{
	//store selected instance in the selectedInstance(browser variable)
	//redirect to ExportPDF.html
	}

10. ExportPDF.html

File name- ExportPDF.html
File description-This page would help to download the traceability linkages in PDF format.
Functions present:
Mapped to element: Browser(component)
Functional Description:

Function name	loadProperties()
Function Trigger point	This function will be invoked by when page is loaded
Function input	nil
Function return type	void
Functions description	This function will display the properties corresponding to each entity

Function pseudo code	function loadProperties()
	{
	//retrieve the selected Instance name and its linked instances (Browser storage variable selected Instance)
	//for each instance invoke the getProperties function

	//display properties in table format as shown in prototype }

Function name	getProperties(enititiyName)
Function Trigger point	This function will be invoked by loadProperties()
Function input	enititiyName
Function return type	Object
Functions description	This function will fetch the properties of the corresponding entity using FetchProperties service.

Function pseudo code	function getProperties(enititiyName)
	{
	//invoke FetchProperties service
	return result
	}

Function name	exportPDF()
Function Trigger point	This function will invoked on clicking the ExportPDF button
Function input	nil
Function return type	void
Functions description	This function will download a PDF version of what is being displayed on the screen.

Function pseudo code	function exportPDF()
	{
	// use existing jquery to convert to do this
	}

Function name	goBack()
Function Trigger point	This function will be invoked on clicking on the back button
Function input	nil
Function return type	void
Functions description	This function will goto previous dashboard page

Function pseudo code	function goBack()
	{
	//redirect to the ViewTraceability.html page (previous state)
	}

b. Detailed design for servlets (web services)

1.Service name:LoginService

Service description: This service will validate user login credentials.
Functions present: getJSON()
Mapped to element: loginService(component)

Functional description:

Function name	getJSON()
Function Trigger point	When the url “loginservice&?username&?password” is invoked.
Function input	Username, password
Function return type	String

Functions description	This function will read the username and password from the URL and it will access the user collection and check if the username and password are correct. If the username and password match a string stating “Success” will be returned. Else a string stating “not successful” will be returned.

Function pseudo code	function getJSON()
	{
	//get the user collection from the storage medium//
	//check if the username and password matches//
	If so return “success”
	Else return “not successful”
	}

2.Service name:FetchProjects

Service description: This service will fetch the list of projects associated with a user and return it.
Functions present: getJSON()
Mapped to element: FetchProject(component)

Functional description:

Function name	getJSON()
Function Trigger point	When the url “fetchproject&?username” is invoked.
Function input	UserId
Function return type	String
Functions description	This function will read the username and it will access the “project collection” and find the list of projects that are associated with the specific user and returns it.
Function pseudo code	function getJSON() {

	//get the list of projects that are associated with a specific user from the project collection present in the storage medium//
	//return back the list of projects fetched”
	}

Mongo Query:

```
db.ProjectCollection.aggregate([
{$match:
  {
        projectName:"Simulated Distillaries"
  }
 },
 { "$lookup": {
        "from": "UserCollection",
        "localField": "_id",
        "foreignField": "projects.projectIndexId",
        "as": "productObjects"
        }},
        { "$unwind": "$productObjects" },
        {$match:
  {
        "productObjects._id":"SSK001"
  }
 },

{ $project : {
        "productObjects.projects.projectRole":1,"_id":0
  } },
]
);
```

3.Service name:FetchRoles

Service description: This service will fetch the list of roles associated with a user for a specific user and return it.
Functions present: getJSON()
Mapped to element: FetchRoles(component)

Functional description:

Function name	getJSON()
Function Trigger point	When the url “fetchroles&?username&?projectname” is invoked.
Function input	Username, projectname
Function return type	String
Functions description	This function will read the username and project name from the URL and it willl access the “Roles collection” and find the list of roles that are associated with the specific user for that specific project and returns it.

Function pseudo code	function getJSON()
	{
	//Read the username and projectname from the URL//
	//get the list of roles that are associated with a specific user for a specific project from the roles collection present in the storage medium//
	//return back the list of roles fetched”
	}

Mongo Query:

```
db.ProjectCollection.aggregate([
{$match:
  {
        projectName:"Simulated Distillaries"
  }
 },
 { "$lookup": {
        "from": "UserCollection",
        "localField": "_id",
        "foreignField": "projects.projectIndexId",
        "as": "productObjects"
        }},
        { "$unwind": "$productObjects" },
        {$match:
```

```
  {
        "productObjects._id":"SSK001"
  }
 },
  { "$unwind": "$productObjects.projects" },
  {$match:
  {
        "productObjects.projects.projectName":"Simulated Distillaries"
  }
 },
 { $project : {
        "productObjects.projects.projectRole":1,"_id":0
  } },

  {$group :
        {_id : "$productObjects.projects.projectRole"}
        },
]
);
```

4.Service name:FetchEntities

Service description: This service will fetch the list of entities associated with a specific user roles and return it.
Functions present: getJSON()
Mapped to element: FetchEntities(component)

Functional description:

Function name	getJSON()
Function Trigger point	When the url “fetchentities&?rolename&?projectname” is invoked.
Function input	rolename, projectname
Function return type	String
Functions description	This function will read the rolename and project name from the URL and it will access the “Model collection” and fetch the list of entities that are associated with the specific role of the user.

Function pseudo code	function getJSON()
	{
	//Read the rolename and project name from the URL//
	//get the list of entities that are associated with a specific user's role for a specific project from the model collection present in the storage medium//
	//return back the list of entities fetched"
	}

Mongo Query:

```
db.project.aggregate([
  {$unwind: { path: "$model"} },
   {$unwind: { path: "$model.entities"} },
                              {$match:        {        $and:        [{"projectName":"Simulated
Distillaries"},{"model.entities.rolesWithWriteAcess":"BusinessAnalyst"} ]}},
   {$project:{"model.entities._id":1}},
  ])
```

5.Service name:FetchInstances

Service description: This service will fetch the list of instances associated with a specific entity
Functions present: getJSON()
Mapped to element: FetchInstances(component)

Functional description:

Function name	getJSON()
Function Trigger point	When the url "fetchentities&?entityname&?username&?project name" is invoked.
Function input	entityname, username,projectname.
Function return type	String
Functions description	This function will read the username,entity name and project name from the URL and it will access the "Instance collection" and fetch the list of

	instances that are associated with a specific user for a specific project.

Function pseudo code	function getJSON()
	{
	//Read the entityname, username and project name from the URL//
	//get the list of instances that are associated with a specific role for a specific project from the instance collection present in the storage medium//
	//return back the list of instances fetched"//
	}

6.Service name:UpsertInstance

Service description: This service will read a string and parse it as JSON object and insert/update it into the "instance collection".
Functions present: getJSON()
Mapped to element: UpsertInstance (component)

Functional description:

Function name	getJSON()
Function Trigger point	When the url "insertinstance&?result" is invoked.
Function input	result
Function return type	String
Functions description	This function will read the result string from the URL and parse it into a JSON object. If the record is already present it will update or it will insert into the "instance collection". This function will finally return a string stating "Success" if the instance has been inserted/updated into the instance collection. Or else it would return a string stating "Instance not inserted/updated".
Function pseudo code	function getJSON() {

	//Read ther result string value from the URL//
	//parse the result string into a JSON object//
	//Upsert the parsed JSON object into the "instance collection" present in the storage medium.
	//if(upsert is successful)
	{
	Return "success";
	}
	Else
	{
	Return "upsert failed";
	}

7.Service name:FetchMappedEntities

Service description: This service takes the entity name as input and fetches the mapped entities for that entity in that project.
Functions present: getMappedEntities()
Mapped to element: FetchMappedEntities (component)

Functional description:

Function name	getMappedEntities()
Function Trigger point	When the url "FetchMappedEntities&?entity" is invoked.
Function input	entity
Function return type	String
Functions description	The function reads the entity string containing the entity name as input and fetches the list of entities that can be linked to that entity.

Function pseudo code	function getMappedEntities()
	{
	//Read the entity string value from the URL//
	if(entity exists)
	{
	//Fetch the linked entities for that particular entity from the Project collection into result.
	return result

	}
	
	else
	{
	return null
	}

8.Service name:FetchProperties

Service description: This service takes the entity name as input and fetches the list of properties defined for that entity.
Functions present: getProperties()
Mapped to element: FetchProperties (component)

Functional description:

Function name	getProperties()
Function Trigger point	When the url "fetchproperties&?entity" is invoked.
Function input	entity
Function return type	String
Functions description	The function reads the entity string containing the entity name as input and fetches the list of properties that have been defined for that entity and populates a drop down list for further selection.

Function pseudo code	function getProperties()
	{
	//Read the entity string value from the URL//
	if(entity exists)
	{
	//Fetch the properties defined for that particular entity from the Properties collection into result.
	return result
	}
	else
	{
	return null
	}

9.Service name:FetchModels

Service description: This service takes the project name as the input and fetches the model for that particular project.
Functions present: getModel(String projectname)
Mapped to element: FetchModels (component)

Functional description:

Function name	getModel()
Function Trigger point	When the url "FetchModels&?project" is invoked.
Function input	String (project name)
Function return type	String
Functions description	The function reads the project name and fetches the entire model for that project

Function pseudo code	function getModel(String projectname)
	{
	//Read the project name from the URL
	
	//Fetch the models for that particular project and copy it in the result and return it
	}

10.Service name: FetchParentChild

Service description: This service takes the project name as input and fetches the parent and child projects for that project.
Functions present: getParentAndChild()
Mapped to element: FetchParentChild (component)

Functional description:

Function name	getParentAndChild()

Function Trigger point	When the url “getParentAndChild&?project_name” is invoked.
Function input	project_name
Function return type	String
Functions description	The function takes the project name as input and fetches the parent and child projects for that project from the project collection.

Function pseudo code	function getParentAndChild()
	{
	//Read the project_name string value from the URL//
	if(project_name exists)
	{
	//Fetch the parent and child versions for the project
	return result
	}
	else
	{
	return null
	}

c. Detailed design for collections

User Collection

```
{
        {
          “username”:”Arul”,
          “password”:”Arul”
        },

        {
          “username”:”Aparrna”,
          “password”:”Aparrna”
        }

}
```

Project collection

Roles collection: