**Service Enabled Workflow Requirement Specifications**

* **Overall Flow Diagram:**
  + Run Nova Workflow Engine or simulate the engine part to get a list of tasks for a workflow.
  + Setup and configure service enabled status of each of the tasks.
  + Provide input, put and QoS properties for each of the service enabled tasks.
  + Run the workflow. For each of the configure tasks, the app will do the followings:
    - Go over the published services of the repository. Match input, output and QoS properties of each of the services. If Input, Output and QoS properties matches, add this to the matched service list.
    - Repeat the same steps for rest of the services. If multiple matched services are found, do the followings:
      * Do the input matching. Calculate scores for this.
      * Do the output matching. Calculates score for this.
      * If input and output matching score is same for the matched services, calculate QoS score for each of the services:
      * Return the service having maximum score.
    - The client app will use OWL-S VM like execution engine to execute the service and move the next steps.
  + Continue to repeat the steps for each of the configured tasks of a workflow to finish up the overall execution.
* **Requirements on Service Registry Manager:**
  + **Service Provider:**
    - Service providers will do signup request from the web console
    - Super admin will approve/deny provider’s request.
    - Notification email will be sent to specific provider’s email address
    - Service provider will login to the console. In home page, she will see a list of uploaded services uploaded by him. He will get option to add new service too.
    - To add new service to the repository, she will have to go through the following steps:
      * Provide Service Specific Information:
        + Service Name
        + Service Description
        + OWL-S Service Files:

Service.owl

Profile.owl

Processs.owl

Grounding.owl

Grounding.wsdl

* + - * Provide Contact Information:
        + Description: Provides a mechanism of referring to humans or individuals responsible for the service (or some aspect of the service). The range of this property is unspecified within OWL-S, but can be restricted to some other ontology, such as FOAF, VCard, or our OWL-S Actor ontology. Our web interface used the OWL-S Actor ontology for Contact Information.
        + Name: Name of service provider
        + Title: Title of service provider
        + Phone: Telephone number
        + Email: Email address
        + Fax : Fax Number
        + Physical Address: Postal address in free format
        + WebURL: URL to the provider's web site
      * OWL-S verifier will be used to check provides owl file’s details. Then upload the service files in the cloud.
    - For each service, a folder will be created on the cloud. Each folder will contain that service specific owl files. These services will be accessible through web URLs.
    - Service and Contact details will be saved to DB
  + **Service Consumer:**
    - They will do signup request from web console.
    - SMS will be sent to her mobile.
    - User will use the activation code to activate her account
    - She will login to the app to search and discover web services.
    - An user friendly UI will be displayed which will contain the following things:
      * List of input/output concepts
      * Text boxes for instances of the concepts
      * Search button
      * List view to show the list of discovered services
      * Select a service from the list to execute.
      * Execute the service using OWL-S API call
      * Show result to user.
* **UI Menus in Service Registry Manager:**
  + **Super Admin:**
    - Manage Signup Requests
    - Display services published by all providers
  + **Service Provider:**
    - Manage Services:
      * Display list of published services by the logged in provider
      * Upload designed SWSs
  + **Service Consumer:**
    - Discover Services:
      * Provide search text (input, output and QoS concepts) to get list of matched services
      * Display list of discovered services
* Technologies:
  + Bootstrap, Html 5.0, CSS 3.0 & Angular JS
  + MVC Framework 5.0 and Entity Framework 6.0
  + Visual Studio 2013 or 2015