**Design & Implementation Document**

* **Resources & Representations:**

I have used RestfulDoctor.java class as the resource in REST API.

Resource Data is respresented in plain, xml and json format.

In RestfulDoctor.java, I have added DoctorRS.java class to implement CRUD operations on resource. The methods like getPlain(), getXml() and getJson() are used to represent the resource data through POJO classes.

* **Available Operations & WS API used:**

The operations like create, update, delete and read data in different formats are implemented with following syntax:

//create

**public** Response create(@FormParam("name") String name, @FormParam("patients") String patientNames)

//update

**public** Response update(@FormParam("id") **int** id, @FormParam("name") String name)

//delete

**public** Response delete(@PathParam("id") **int** id)

//read

**private** Response toRequestedType(**int** id, String type)

**return** Response.*ok*(toJson(doctor), type).build();

Following are the apis used in the implementation:

javax.ws.rs.GET;

javax.ws.rs.POST;

javax.ws.rs.PUT;

javax.ws.rs.DELETE;

javax.ws.rs.Path;

javax.ws.rs.PathParam;

javax.ws.rs.FormParam;

javax.ws.rs.Produces;

javax.ws.rs.QueryParam;

javax.ws.rs.core.MediaType;

javax.ws.rs.core.Context;

javax.ws.rs.core.Response;

In GET api, the code represents the data according to the requested parameter ID as well as the data as a whole.

In PUT api, the code updates the particular record using ID as the FormParam.

In DELETE api, the code deletes the particular record using ID as the FormParam.

In POST api, the code creates a particular record with Dr. name and its patients as the FormParams.

* **Approach Used:**

The system is designed as a list of doctors i.e. the Object of DoctorsList.java class. The doctor has its id and a list of patients. i.e. the PatientsList.java class. The DoctorsList.java has a list of Doctor.java objects and some utility operations on Doctor,java class. The PatientsList.java has a list of Patient.java objects and some utility operations on Patient,java class.

The system first reads the db files of patients and doctors to initialize doctors and its corresponding patients and represents the list in all formats.