Hierarchy Generation Tools

A part of Government Enhanced Administrative Resource (GEAR) Systems.

GEAR

Source Code: https://github.com/gearsystems

Mentors: Sudheesh Singanamalla, Kranthi Kiran Guduru, Rajat Ujawane
ssudheesh@student.nitw.ac.in , gkranthi@student.nitw.ac.in , rujawane@student.nitw.ac.in

Abstract

We complain & we all complain a lot. About the bad roads with potholes, garbage, flowing sewage, improper drains; about broken benches in public areas, improper and broken sign boards. Along with these regular problems there are also problems like improperly connected wires at transformers, dangerously low lying cables. Broken taps, benches in government schools and areas which are sometimes poorly maintained.

These problems are sometimes taken up by the community members in the locality but many a times, people don't want to take these problems to higher authorities because of constraints of one's own personal time and the hassle of dealing with the officials whenever they call and answering to them.

This problem is a major reason why most of the maintenance incidents across the country go unresolved thus causing inconvenience to the residents in that area and those stakeholders affected due to the particular problem.

Problem To be Tackled

The main problem with government systems is the amount of bureaucracy that's involved in getting work done. Each person redirects to another, as a part of the information right, we are allowed to receive the complete rights and details of the people who are responsible for a particular task in our locality.

The main challenge here is to build a generic GUI tool integrated with a smart database table which can handle the different organizations and their hierarchies present inside the GEAR Systems application

eg. Director > Deans > (Each dean further breaks down to others) > so on …

creating a tree structure which may or maynot have cyclic dependencies making a partial element of it a graph

Constraints

1. Multiple people can be present under multiple higher order authorities.
2. Multiple people can be reporting to a higher order authority
3. There may be flat authority order that could be possible. (Dean 1 - Dean 2 - Dean 3)
4. The hierarchy mapping tool should also have the role of the person and their responsibility

As a sample dataset we'd be taking the NIT Warangal college hierarchy as a sample to test and visualize.

The Order of the hierarchy must be served in logical order as a

parent { child : { parent : child } …. } …

Or any other logical order with a REST API which also has the basic functions broadly for

getHigherAuthority()
sendToAnotherAuthority()
assignRoleToAuthority()

with other obvious functions to be

addNewMember()
linkMemberTo()

All these have to be done through the web interface and the required data structure to replicate it onto the database to store in an efficient manner has to be integrated.

Technology Requirements

Preferably Python, but Java, JavaScript, Ruby etc.., are also fine

Implement as an XMLRPC or a JSONRPC format so that the application calls can also trigger the required databases when accessed. If implemented as REST, It should be able to handle POST and GET requests for the corresponding calls accordingly.

Preferred Frameworks:
If you choose -

1) Python - Django or flask
2) Java - Use Spring with Jersey integration.
3) Ruby - Use RoR
4) NodeJs - Express

Database platforms

MySQL / MariaDB, PostGREsql