CECS 571 Fundamentals of Semantic Web Technologies

COMPUTER ENGINEERING AND COMPUTER SCIENCE

Project 3: Semantic Data Retrieval

By:-
Sai Vardhan Kondapaka
Ankit Ramrakhyani
Sai Mallesh Goud Dabbeti
Sri Ram Prasad
Pavan Sai Reddy Atla

Under guidance of:-

Dr. Bo Fu

GIT link- https://github.com/Ankitr0908/WebSemantics_Project3

Content:-

Dataset-

The following dataset has been used in this project-

Ecology and Endangered Species-

The ontology includes classes for different types of animals, such as mammals, birds, and reptiles, and provides information about their characteristics, habitats, conservation status, and more. For example, each animal class has properties that describe its size, diet, and geographic range. The ontology also includes classes for different types of habitats, such as forests, deserts, and freshwater environments, and provides information about the species that live in each habitat.

The ontology is designed to be extensible, allowing new animal classes and properties to be added as needed. It can be used to represent information about animals in a variety of contexts, such as biodiversity research, conservation efforts, and education. The ontology also includes information about various plant species, and as their taxonomy, growth habits, and habitats.

Link- https://drive.google.com/file/d/1SZ-vrET7AEfIrqpgfowOLrZ44KHe4kCB/view

Owl File

Screenshot1

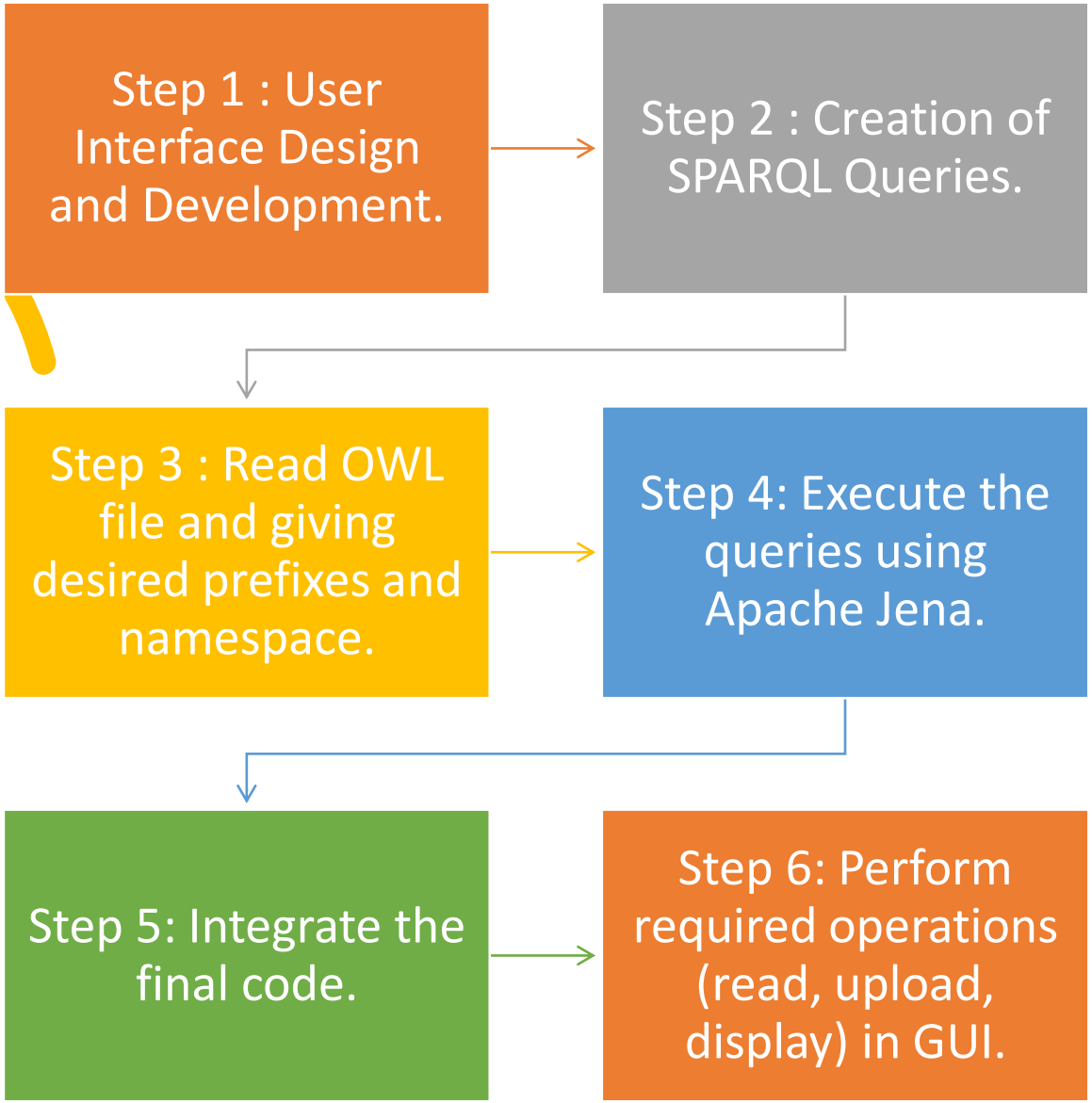
```
<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.ecology-and-endangered-species.com#"
     xml:base="http://www.ecology-and-endangered-species.com"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
    <owl:Ontology rdf:about="http://www.ecology-and-endangered-species.com"/>


    <owl:ObjectProperty rdf:about="http://www.ecology-and-endangered-species.com#eats">
        <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
        <owl:inverseOf rdf:resource="http://www.ecology-and-endangered-species.com#isEatenBy"/>
        <rdfs:domain rdf:resource="http://www.ecology-and-endangered-species.com#Consumer"/>
        <rdfs:domain rdf:resource="http://www.ecology-and-endangered-species.com#LivingOrganism"/>
        <rdfs:range rdf:resource="http://www.ecology-and-endangered-species.com#LivingOrganism"/>
        <rdfs:range>
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                    <rdf:Description rdf:about="http://www.ecology-and-endangered-species.com#Consumer"/>
                    <rdf:Description rdf:about="http://www.ecology-and-endangered-species.com#Producer"/>
                </owl:unionOf>
            </owl:Class>
        </rdfs:range>
    </owl:ObjectProperty>

    <owl:ObjectProperty rdf:about="http://www.ecology-and-endangered-species.com#hasAnimalCharacteristic">
        <rdfs:domain rdf:resource="https://en.wikipedia.org/wiki/Animalia"/>
        <rdfs:range rdf:resource="http://www.ecology-and-endangered-species.com#AnimalCharacteristic"/>
    </owl:ObjectProperty>


    <owl:ObjectProperty rdf:about="http://www.ecology-and-endangered-species.com#hasClimate">
        <rdfs:domain rdf:resource="http://www.ecology-and-endangered-species.com#TerrestrialBiome"/>
        <rdfs:range rdf:resource="http://www.ecology-and-endangered-species.com#Climate"/>
    </owl:ObjectProperty>


    <owl:ObjectProperty rdf:about="http://www.ecology-and-endangered-species.com#hasConservationStatus">
        <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
        <rdfs:domain rdf:resource="http://www.ecology-and-endangered-species.com#LivingOrganism"/>
        <rdfs:range rdf:resource="http://www.ecology-and-endangered-species.com#ConservationStatus"/>
    </owl:ObjectProperty>
```

Owl File

Screenshot2

```
<owl:Class rdf:about="http://www.ecology-and-endangered-species.com#ConservationStatus">
    <owl:disjointUnionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="http://www.ecology-and-endangered-species.com#ExtinctSta
        <rdf:Description rdf:about="http://www.ecology-and-endangered-species.com#LowerRiskS
        <rdf:Description rdf:about="http://www.ecology-and-endangered-species.com#Threatened
    </owl:disjointUnionOf>
    <rdfs:comment>Regroups different conservation status used to discribe the conservation s
</owl:Class>


<owl:Class rdf:about="http://www.ecology-and-endangered-species.com#Consumer">
    <rdfs:subClassOf rdf:resource="http://www.ecology-and-endangered-species.com#Diet"/>
    <owl:disjointWith rdf:resource="http://www.ecology-and-endangered-species.com#Producer"/
    <owl:disjointUnionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="http://www.ecology-and-endangered-species.com#Carnivore"
        <rdf:Description rdf:about="http://www.ecology-and-endangered-species.com#Herbivore"
        <rdf:Description rdf:about="http://www.ecology-and-endangered-species.com#Omnivore"/
    </owl:disjointUnionOf>
</owl:Class>

<owl:Class rdf:about="http://www.ecology-and-endangered-species.com#Continental">
    <rdfs:subClassOf rdf:resource="http://www.ecology-and-endangered-species.com#Climate"/>
</owl:Class>


<owl:Class rdf:about="http://www.ecology-and-endangered-species.com#Coral_reefs">
    <rdfs:subClassOf rdf:resource="http://www.ecology-and-endangered-species.com#MarineBiome
</owl:Class>

<owl:Class rdf:about="http://www.ecology-and-endangered-species.com#CriticallyEndangered">
    <rdfs:subClassOf rdf:resource="http://www.ecology-and-endangered-species.com#ThreatenedS
</owl:Class>


<owl:Class rdf:about="http://www.ecology-and-endangered-species.com#Dams">
    <rdfs:subClassOf rdf:resource="http://www.ecology-and-endangered-species.com#Exploitatio
</owl:Class>


<owl:Class rdf:about="http://www.ecology-and-endangered-species.com#Diet">
    <owl:disjointUnionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="http://www.ecology-and-endangered-species.com#Consumer"/
        <rdf:Description rdf:about="http://www.ecology-and-endangered-species.com#Producer"/
    </owl:disjointUnionOf>
```

Technical Approach: Overview



Code Snippets

SPARQLQueryUI.java

src > SPARQLQueryUI.java > SPARQLQueryUI

```java
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;

import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;
import javax.swing.filechooser.FileFilter;
import javax.swing.filechooser.FileNameExtensionFilter;

import org.apache.jena.query.Query;
import org.apache.jena.query.QueryExecution;
import org.apache.jena.query.QueryExecutionFactory;
import org.apache.jena.query.QueryFactory;
import org.apache.jena.query.ResultSetFormatter;
import org.apache.jena.rdf.model.Model;
import org.apache.jena.rdf.model.ModelFactory;
```

```java
// Create the label and text area for the SPARQL query
JLabel lblQuery = new JLabel(text:"SPARQL query:");
lblQuery.setBounds(x:10, y:40, width:100, height:20);
mainPanel.add(lblQuery);

txtQuery = new JTextArea();
txtQuery.setLineWrap(wrap:true);
txtQuery.setWrapStyleWord(word:true);
JScrollPane scrollQuery = new JScrollPane(txtQuery);
scrollQuery.setBounds(x:120, y:40, width:430, height:100);
mainPanel.add(scrollQuery);

// Create the label and text area for the query results
JLabel lblResult = new JLabel(text:"Results:");
lblResult.setBounds(x:10, y:150, width:100, height:20);
mainPanel.add(lblResult);

txtResult = new JTextArea();
txtResult.setLineWrap(wrap:true);
txtResult.setWrapStyleWord(word:true);
txtResult.setEditable(b:false);
JScrollPane scrollResult = new JScrollPane(txtResult);
scrollResult.setBounds(x:120, y:150, width:430, height:200);
mainPanel.add(scrollResult);

// Create the button for executing the query
btnExecute = new JButton(text:"Execute Query");
btnExecute.setBounds(x:10, y:360, width:120, height:20);
btnExecute.addActionListener(this);
mainPanel.add(btnExecute);
```

Code Snippets

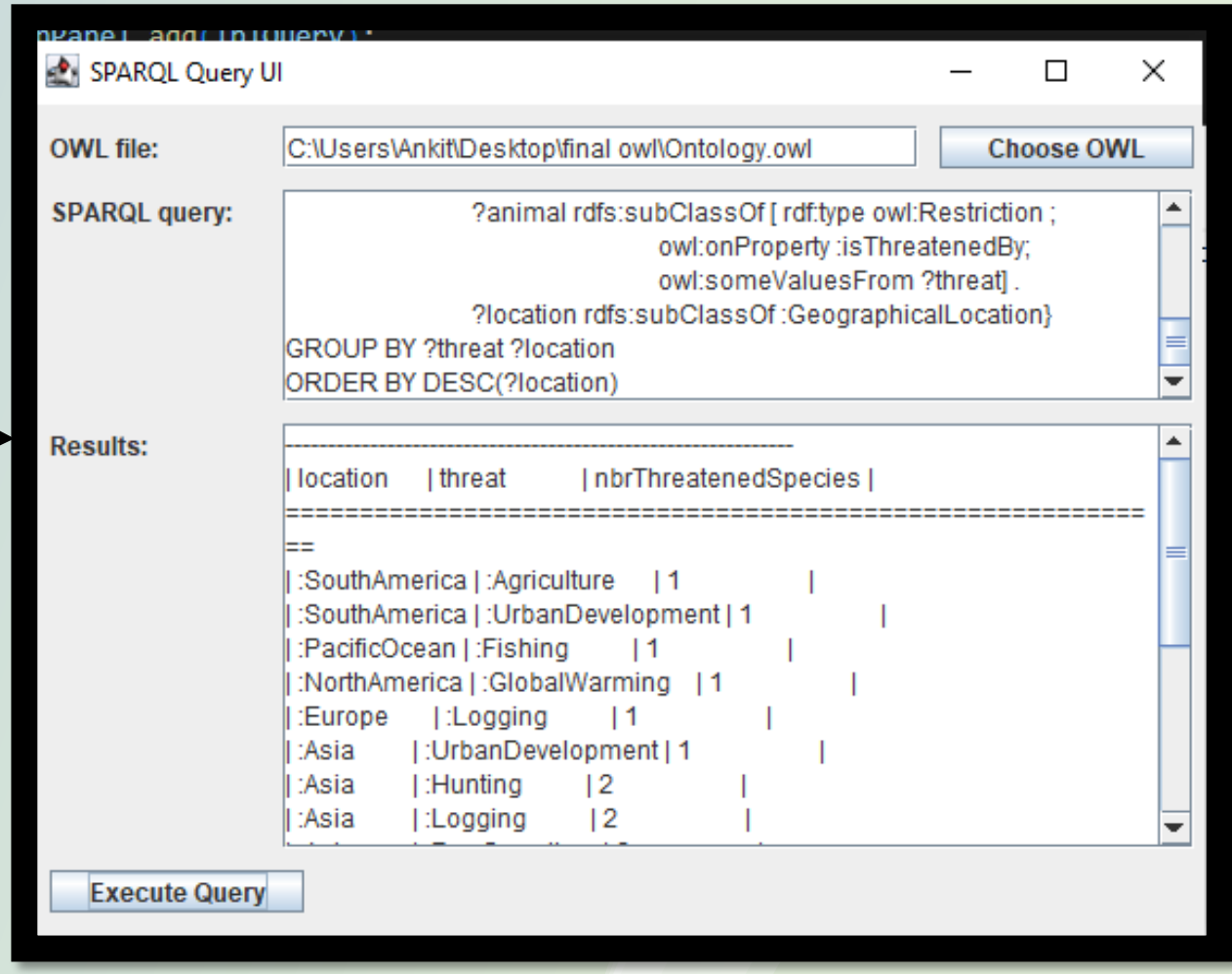
```java
public class SPARQLQueryUI extends JFrame implements ActionListener {

    private static final long serialVersionUID = 1L;

    private JTextField txtOWLFile;
    private JTextArea txtQuery;
    private JTextArea txtResult;
    private JButton btnChooseOWL;
    private JButton btnExecute;

    public SPARQLQueryUI() {
        super(title:"SPARQL Query UI");

        // Create the main panel
        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(mgr:null);

        // Create the label and text field for the OWL file path
        JLabel lblOWLFile = new JLabel(text:"OWL file:");
        lblOWLFile.setBounds(x:10, y:10, width:100, height:20);
        mainPanel.add(lblOWLFile);

        txtOWLFile = new JTextField();
        txtOWLFile.setBounds(x:120, y:10, width:300, height:20);
        mainPanel.add(txtOWLFile);

        btnChooseOWL = new JButton(text:"Choose OWL");
        btnChooseOWL.setBounds(x:430, y:10, width:120, height:20);
        btnChooseOWL.addActionListener(this);
        mainPanel.add(btnChooseOWL);
```

```java
        // Add the main panel to the frame
        getContentPane().add(mainPanel);

        // Set the size and center the window
        setSize(width:570, height:430);
        setLocationRelativeTo(c:null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == btnChooseOWL) {
            // Open a file chooser dialog for selecting the OWL file
            JFileChooser fileChooser = new JFileChooser();
            FileFilter filter = new FileNameExtensionFilter(description:"OWL files", ...extensions:"owl");
            fileChooser.setFileFilter(filter);
            int result = fileChooser.showOpenDialog(this);
            if (result == JFileChooser.APPROVE_OPTION) {
                File selectedFile = fileChooser.getSelectedFile();
                txtOWLFile.setText(selectedFile.getAbsolutePath());
            }
        } else if (e.getSource() == btnExecute) {
            // Get the OWL file path and the SPARQL query text
            String owlFilePath = txtOWLFile.getText();
            String sparqlQuery = txtQuery.getText();

            // Load the OWL file into a Jena model
            Model model = ModelFactory.createDefaultModel();
            model.read(owlFilePath);

            // Create the query and execute it against the model
            Query query = QueryFactory.create(sparqlQuery);
```

Query 1 (Complex):-

What are the locations and the types of threats faced by the endangered animal species, and how many species are threatened in each location?

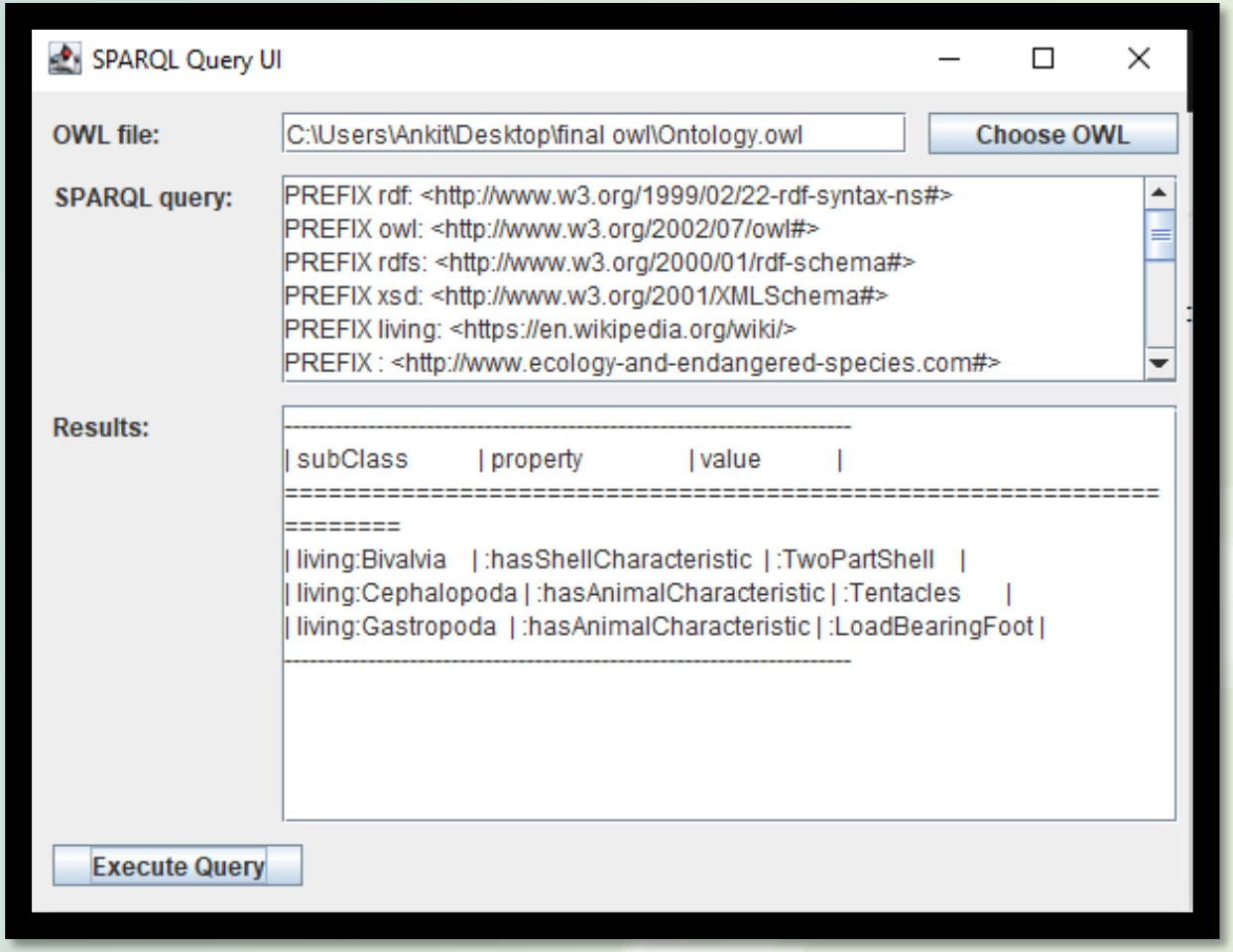
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.ecology-and-endangered-species.com#>
PREFIX living: <https://en.wikipedia.org/wiki/>
SELECT  ?location ?threat (count(?threat) as ?nbrThreatenedSpecies)
WHERE {?animal rdfs:subClassOf* living:Animalia .
        ?animal rdfs:subClassOf [ rdf:type owl:Restriction ;
                owl:onProperty :livesIn;
                owl:someValuesFrom ?location] .
        ?animal rdfs:subClassOf [ rdf:type owl:Restriction ;
                owl:onProperty :isThreatenedBy;
                owl:someValuesFrom ?threat] .
        ?location rdfs:subClassOf :GeographicalLocation}
GROUP BY ?threat ?location
ORDER BY DESC(?location)
```



Query 2:-

What are the properties and their corresponding values that define the subclasses of the class equivalent to living:Mollusca?

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX living: <https://en.wikipedia.org/wiki/>
PREFIX : <http://www.ecology-and-endangered-species.com#>
SELECT ?subClass ?property ?value
        WHERE { ?subClass owl:equivalentClass [
                owl:intersectionOf ( living:Mollusca
                        [rdf:type owl:Restriction ;
                        owl:onProperty ?property ;
                        owl:someValuesFrom ?value]) ;
                rdf:type owl:Class]
}
ORDER BY ASC(?subClass)
```



Query 3 (Complex):-

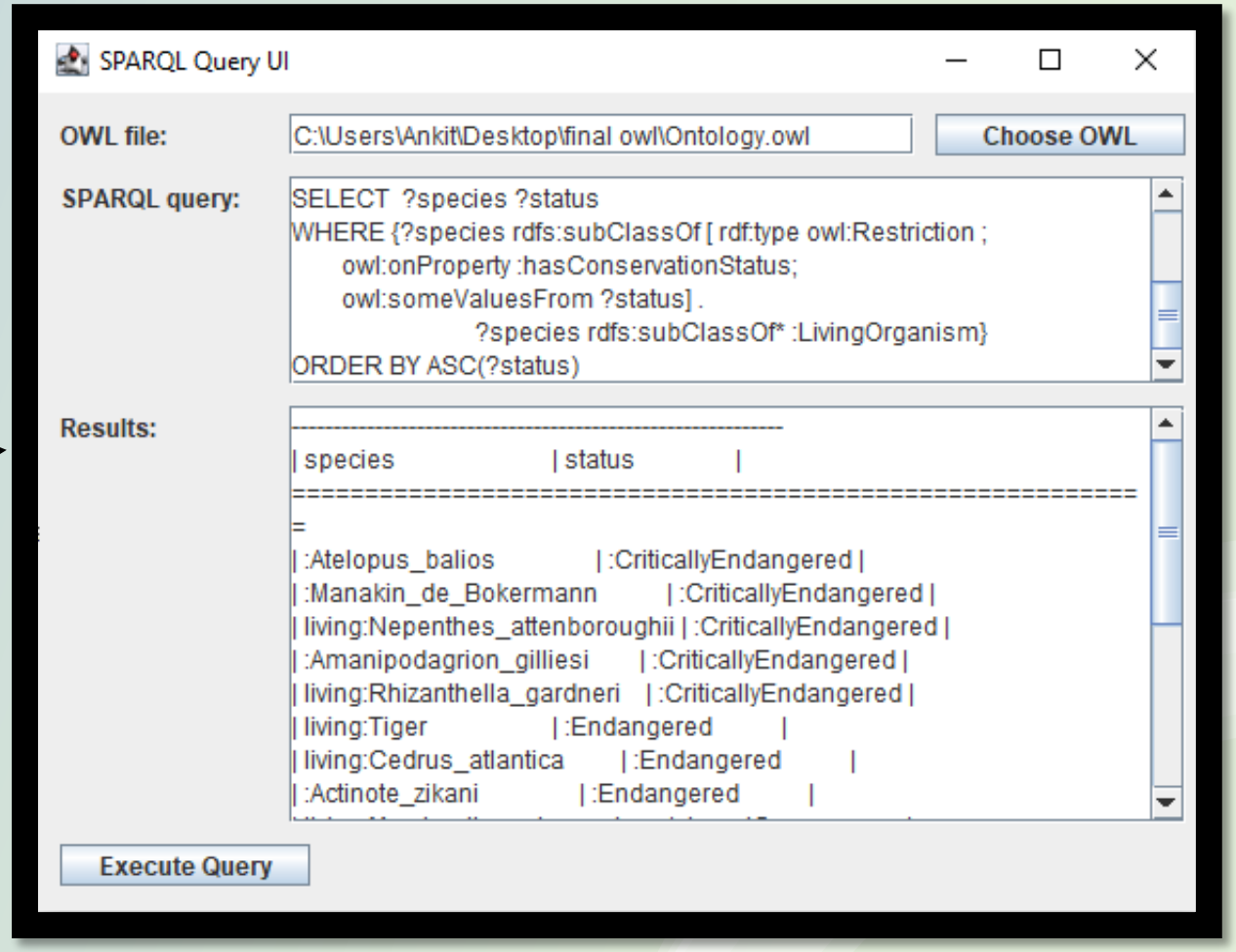
What are the locations and types of threats faced by the plant species that have both conductor vessels and produce seeds, and how are these species further classified based on their plantae characteristics?

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX living: <https://en.wikipedia.org/wiki/>
PREFIX : <http://www.ecology-and-endangered-species.com#>
SELECT ?plante_conductor_seed ?threat ?location
        WHERE { ?plante rdfs:subClassOf* living:Plantae .
         ?plante owl:equivalentClass [ owl:intersectionOf ( ?any_plante
                                                            [ rdf:type owl:Restriction ;
                                                        owl:onProperty :hasPlantaeCharacteristic ;
                                                        owl:someValuesFrom :Conductor_Vessels
                                                        ]
                                                             ) ; rdf:type owl:Class].
                          ?plante_conductor rdfs:subClassOf* ?plante .
                          ?plante_conductor owl:equivalentClass [ owl:intersectionOf ( ?any_plant
                                                            [ rdf:type owl:Restriction ;
                                                        owl:onProperty :hasPlantaeCharacteristic ;
                                                        owl:someValuesFrom :Produce_Seed
                                                        ]
                                                             ) ; rdf:type owl:Class].
        ?plante_conductor_seed rdfs:subClassOf* ?plante_conductor.      ?plante_conductor_seed rdfs:subClassOf [ rdf:type owl:Restriction ;
                owl:onProperty :isThreatenedBy;
                owl:someValuesFrom ?threat] .
        ?threat rdfs:subClassOf :Threat.        ?plante_conductor_seed rdfs:subClassOf [ rdf:type owl:Restriction ;
                owl:onProperty :livesIn;
                owl:someValuesFrom ?location] .
        ?location rdfs:subClassOf :GeographicalLocation
}
```

Query 4:-

What are the living organisms and their corresponding conservation statuses?

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.ecology-and-endangered-species.com#>
PREFIX living: <https://en.wikipedia.org/wiki/>
SELECT  ?species ?status
WHERE {?species rdfs:subClassOf [ rdf:type owl:Restriction ;
        owl:onProperty :hasConservationStatus;
        owl:someValuesFrom ?status] .
        ?species rdfs:subClassOf* :LivingOrganism}
ORDER BY ASC(?status)
```



Query 5:-

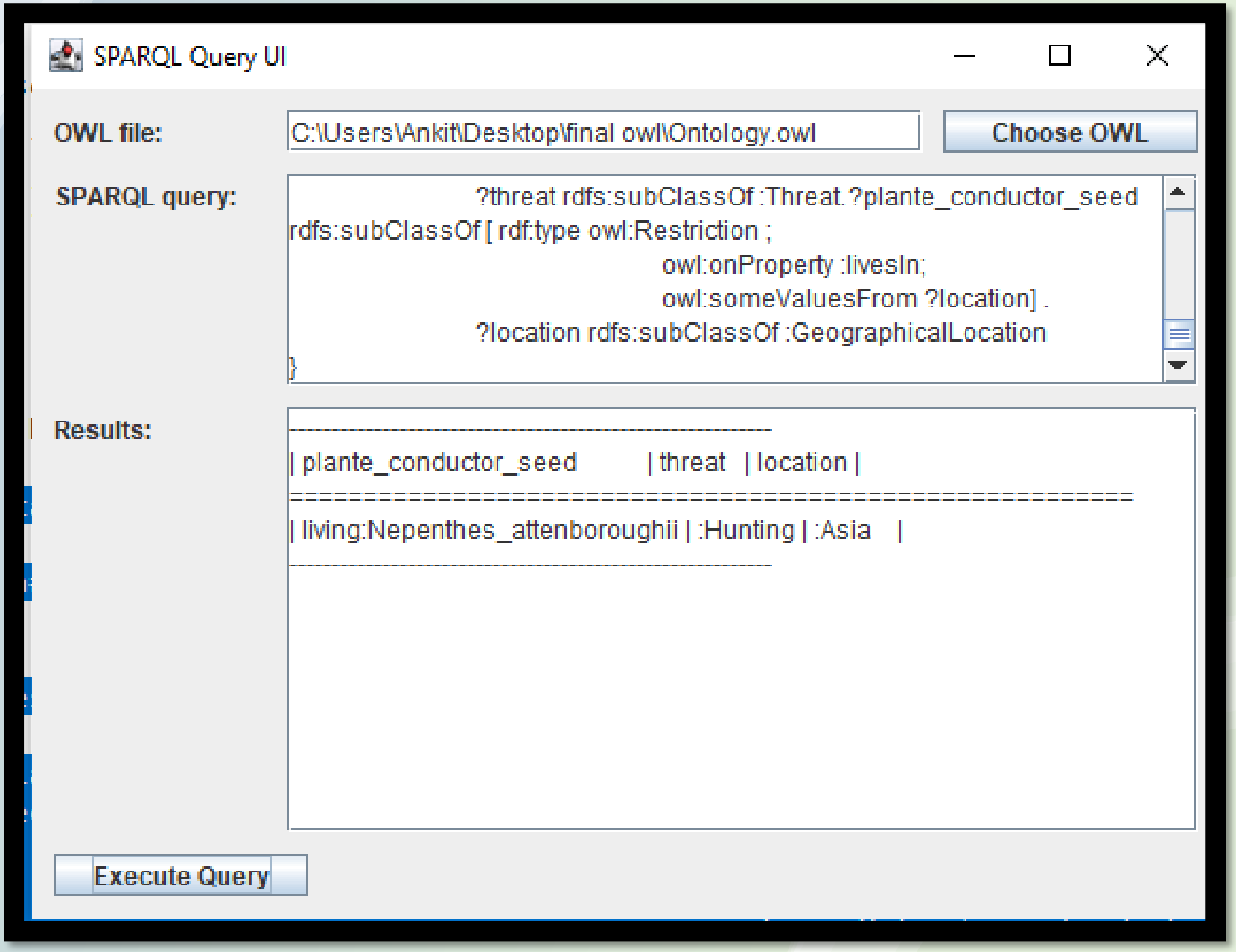
Query to retrieve all the subclasses and equivalent classes of the Eurasian otter class and returns their properties.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.ecology-and-endangered-species.com#>
PREFIX living: <https://en.wikipedia.org/wiki/>
SELECT ?Eurasian_otter_properties
WHERE {
{
        :Eurasian_otter rdfs:subClassOf ?Eurasian_otter_properties
}UNION
{
        :Eurasian_otter rdfs:subClassOf* ?sub.
        ?sub owl:equivalentClass [
                owl:intersectionOf (?intersection ?Eurasian_otter_properties) ; rdf:type owl:Class]
}}
```

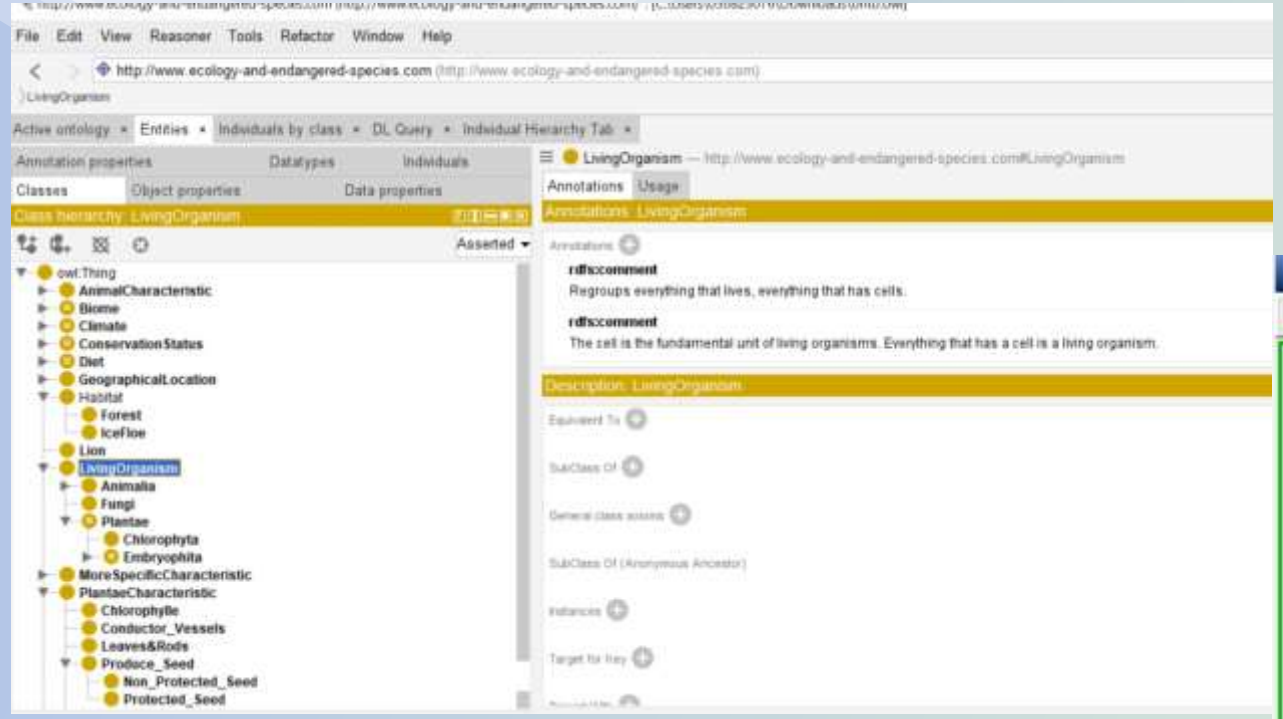
UI Screenshot

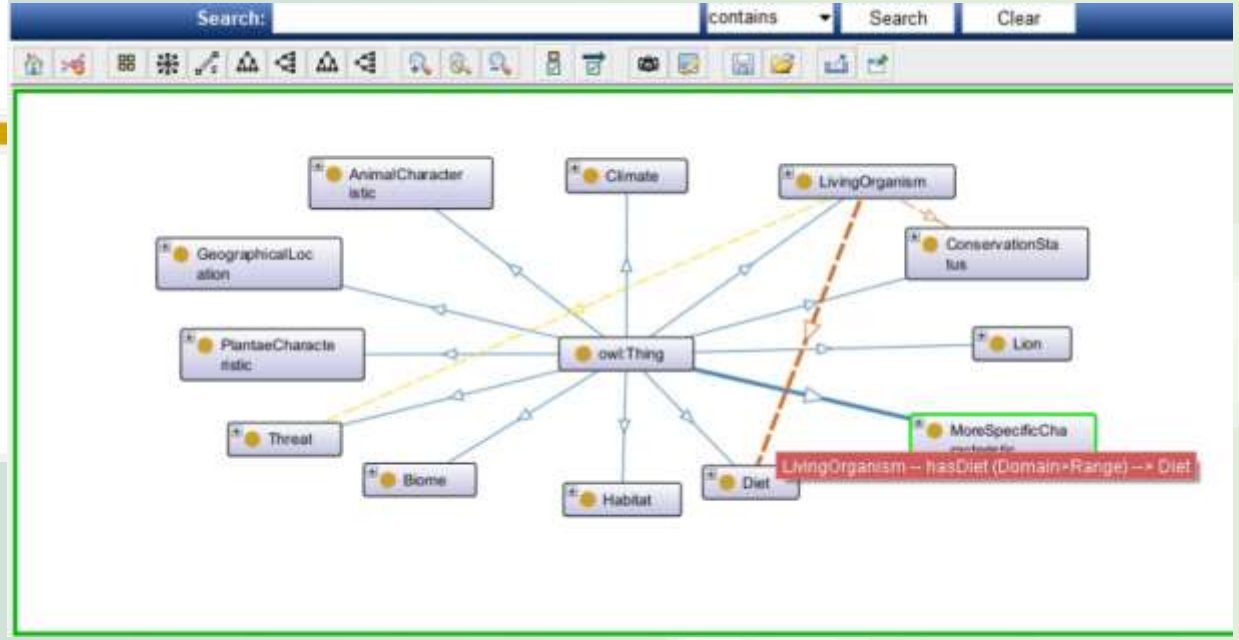
The UI has three components-

- **OWL File-** It is to choose the OWL file input from user locally.
- **SPARQL query-** Input Area where user can write queries, for execution on the selected dataset.
- **Results** – Output box of the desired query result.



OWL Visualization





Technical Difficulties:-

Choosing The Dataset-

Initially Choosing of Project 1 Dataset

Changed the dataset to get Complex Queries.

Auto Refresh of The Application-

Use Timer class to schedule a task that updates the text field periodically.

Or, using DefaultTableModel, which provide automatic refresh functionality when data is added, removed, or updated.

SPARQL and OWL Syntax-

Understanding and working on SPARQL queries syntax and handling it to output the desired results.

Integrating Apache Jena-

Working with and using Apache Jena in the project which is a Semantic Web framework for Java, for executing the queries and interpreting the owl input files.

Appendix –

-> Making sure system runs without error.

-> Making the code self-contained. (Apache Jena exception)

-> Adding informative comments such that people not involved can get a vague understanding about the project and its objective.

References-

Jena - https://jena.apache.org/tutorials/rdf_api.html

Sparql -https://medium.com/wallscope/constructing-sparql-queries-ca63b8b9ac02

OWL -https://www.w3.org/TR/owl-guide/

GIT Link- https://github.com/Ankitr0908/WebSemantics_Project3


THANK
YOU!