****Semantic Web Capstone Project Ontology**  
**Software Requirements Specification (SRS)**  
 Group Members:**

* **Sobia Zafar(BSCS51F21R031)**
* Abeer Zafar(**BSCS51F21R038)**
* **Isbah Imran(BSCS51F21R027)**

### ****1. Introduction****

#### 1.1 Purpose

This document outlines the software requirements for developing an ontology that formally represents the key elements of a capstone project using Semantic Web technologies. It supports structured knowledge representation and semantic querying.

#### 1.2 Scope

The ontology will cover major components of a capstone project including:

* Project Proposals
* Stakeholders (students, supervisors, evaluators)
* Evaluation Criteria
* Documentation (e.g., SRS, design documents)  
  It will provide a machine-readable and reusable framework suitable for academic environments.

#### 1.4 References

* W3C Semantic Web Standards: RDF, OWL, SPARQL
* Capstone Project Guidelines (Instructor: Sir Fahad)

#### 1.5 Overview

This SRS describes both functional and non-functional requirements, design constraints, and interface specifications for the ontology.

### ****2. Overall Description****

The ontology is intended as a flexible and extensible model that can represent any capstone project structure. It is built using Semantic Web standards and enables intelligent data access through semantic technologies.

### ****2.1 Product Functions****

* Define core classes such as Proposal, Stakeholder, Evaluation Criteria, and Documentation
* Establish relationships among these entities
* Enable querying using SPARQL for extracting structured information
* Ensure semantic compatibility for integration with external systems

### ****2.2 User Classes and Characteristics****

**Students:** Input and manage their capstone project details

**Supervisors and Evaluators:** Review project elements through queries.

**Developers:** Extend or integrate the ontology into other applications.

### ****2.3 Operating Environment****

* Developed using Protégé (Ontology Editor)
* Stored in OWL/RDF format
* Executed and queried via SPARQL interfaces and compatible tools

### ****3. Specific Requirements****

**3.1 External Interface Requirements**

#### **3.1.1 User Interfaces**

No direct graphical interface; ontology accessed through tools like Protégé or SPARQL endpoints.

#### **3.1.2 Hardware Interfaces**

No specialized hardware required.

#### **3.1.3 Software Interfaces**

Compatible with standard Semantic Web tools (RDF, OWL, SPARQL engines).

### ****3.2 Functional Requirements****

* **Main Class:** CapstoneProject
* **Subclasses:** Proposal, Stakeholder, EvaluationCriteria, Documentation
* **Relationships:**
* Capstone Project → has Proposal → Proposal
* Capstone Project → has-stakeholder → Stakeholder
* Capstone Project → has Evaluation Criteria → Evaluation Criteria
* Capstone Project → has Documentation → Documentation

#### 3.2.2 SPARQL Query Examples:

* Retrieve all stakeholders for a project
* List evaluation criteria
* Get project-related documentation

### ****3.3 Non-Functional Requirements****

#### 3.3.1 Performance

SPARQL queries should return within 2 seconds for datasets of 10+ projects.

#### 3.3.2 Scalability

Ontology should support multiple capstone projects without affecting performance.

#### 3.3.3 Reusability

Ontology must be reusable for various academic projects across semesters.

#### 3.3.4 Standards Compliance

Compliant with W3C standards for OWL, RDF, and SPARQL.

### ****3.4 Design Constraints****

* Ontology must be modeled using OWL
* Must follow best practices for extensibility and clarity
* Should be platform-independent and easily exportable

### ****4. Other Requirements****

#### 4.1 Documentation

* User Manual for using the ontology and executing SPARQL queries
* Technical Documentation describing all classes, properties, and relationships

#### 4.2 Deliverables

* OWL Ontology File
* SPARQL Query Samples
* Final SRS Document

### ****4.3 Timeline****

| **Week** | **Task** |
| --- | --- |
| 1 | Requirement gathering and SRS preparation |
| 2-3 | Ontology modeling using Protégé |
| 4-5 | SPARQL queries and ontology testing |
| 6 | Final documentation and submission |