SE 4485: Software Engineering Projects

Spring 2025

Requirement Documentation

|  |  |
| --- | --- |
| Group Number | 11 |
| Project Title | Build a Design-of-experiments Interactive Oracle LLM App |
| Sponsoring Company | The Fellows Consulting Group (FCG) |
| Sponsor(s) | Tom Hill |
| Students | 1. Alberto Escobar  2. Dara Moheimani  3. Lee Rafael Filomeno  4. Brandon Bailey  5. Blanca Berrios Henriquez  6. Saishrey Bhandare |

ABSTRACT

The requirements documentation presents the complete software requirements specifications for the Ontology Generation LLM being developed for the Fellow Consulting Group. The document defines both functional and non-functional requirements in accordance with several IEEE/ISO standards, including IEEE Std 830-1998 and ISO/IEC/IEEE Std 29148-2018. The document is structured in a way to provide comprehensive use case models, rationale for those use cases, and detailed technical specifications. Each use case is thoroughly documented with participating actors, entry/exit conditions, flow of events, and special requirements. This specification serves as a complete requirements baseline, and it will be maintained under configuration management throughout the development lifecycle, ensuring that all parties involved work with current requirements and maintain a clear history of requirement evolution.

TABLE OF CONTENTS

1. Introduction………………………………………………………………………………………….
2. Use Case Model For Functional Requirements……………………………………………………..
3. Rationale For Use Case Model……………………………………………………………………...
4. Non-Functional Requirements………………………………………………………………………
5. Configuration Management Evidence………………………………………………………………
6. Engineering Standards and Multiple Constraints…………………………………………………...
7. Additional References……………………………………………………………………………….

LIST OF FIGURES

Figure 1: Use Case Diagram………………………………………………………………………………….

LIST OF TABLES

Table 1: CM tool log………………………………………………………………………………………….

INTRODUCTION

This document specifies the requirements for an Ontology-Generation Large Language Model (LLM) application that leverages Llama 3.2 LLM to automatically generate domain ontologies. The system will be developed through two phases: an initial local Windows implementation for development and testing, followed by deployment to a production web server environment. During development, the system will utilize Ollama for local LLM implementation, and eventually transitioning to server-side LLM processing. The core components include a domain keyword processing system, an integrated ontology generation engine, and a web-based user interface accessible through Chrome.

The document is organized to provide a comprehensive understanding of the system requirements through several key sections. The Use Case Model for Functional Requirements section details system behaviors through use cases, followed by Non-Functional Requirements which specify key performance metrics. The document also includes sections on Configuration Management, Engineering standards that were followed, and any additional references that were used. were followed, and any additional references that were used.

USE CASE MODEL FOR FUNCTIONAL REQUIREMENTS

A diagram of a computer program

AI-generated content may be incorrect.

**Figure 1: Use Case Diagram**

**Display Homepage**

Participating actors: End User

Entry Condition(s): User accesses the WordPress site URL, and system is operational

Normal Flow of Events:

* System loads homepage template
* System displays navigation menu with LLM interface access
* System displays project description

Exit Condition: Homepage is fully loaded and displayed to user

Exceptions: Server unavailable, template loading failure

Special Requirements: Page interface follows Stanford NLP site patterns, load within 3 seconds

**Input Domain Keyword**

Participating actors: End User

Entry Condition(s): User accesses the WordPress application interface, and system is operational  
 Normal Flow of Events:

* User accesses the ontology generation interface.
* User enters domain keyword and submits it.

Exit Condition: The domain keyword is submitted for processing.

Exceptions: System unavailable, interface not accessible.

Special Requirements: Page interface follows Stanford NLP site patterns.

**Validate Input**

Participating actors: End User

Entry Condition(s): User accesses the WordPress application interface, and system is operational  
 Normal Flow of Events:

* System checks input length (1-50 characters)
* System verifies character types
* System confirms input meets requirements

Exit Condition: Input is validated and ready for processing

Exceptions: Input contain invalid characters or length is outside range

Special Requirements: Validation must be completed within 500 ms, and must provide validation status to user

**Generate Ontology**

Participating actors: End User

Entry Condition(s): Valid domain keyword is received, and Local Llama LLM is operational  
Normal Flow of Events:

* System initializes LLM Processing
* System processes domain context
* System Creates node relationships
* System then constructs the graph structure

Exit Condition: Ontology graph structure generated

Exceptions: LLM processing failure

Special Requirements: Must generate ontology within 10 seconds

**Display Ontology Graph**

Participating actors: End User

Entry Condition(s): Ontology graph structure generated

Normal Flow of Events:

* System prepares visualization layout
* System displays completed graph

Exit Condition: Ontology graph visible to end user

Exceptions: Display formatting or rendering issues

Special Requirements: Page interface follows Stanford NLP site patterns

**Display Error Message**

Participating actors: End User

Entry Condition(s): Error condition detected in any use case

Normal Flow of Events:

* System detects error condition
* System determines appropriate error message
* System displays message to user
* System logs error details

Exit Condition: Error message displayed, and error logged

Exceptions: Failure to log error

Special Requirements: Error messages must be clear, and errors should be logged to local file.

Must also support the ability to retry (3 times)

RATIONALE FOR YOUR USE CASE MODEL

The use case model for the Ontology Generation LLM was designed with simplicity and user interaction flow in mind. The rationale for the model’s structure is based on several key considerations. The model is centered around the “Input Domain Keyword” use case, since this represents the fundamental user interaction with the system. This design choice was made intentionally to reflect the system’s core purpose of allowing users to generate domain ontologies through a simple input process.

The model incorporates three include relationships that represent mandatory steps in the process. Input validation is included because every domain keyword submission must be verified against specific constraints. The ontology generation process is included since it represents the system functionality that occurs every time a valid input is processed. Similarly, the display of the ontology graph is included because visualization is a required outcome for a successful ontology generation.

Error message display was designed as an extend relationship because error handling is conditional and only occurs when certain conditions are not met. Additionally, WordPress was incorporated within the system boundary rather than an external actor because it serves as an integral part of the system’s implementation rather than an independent entity interacting with the system. The design approach aligns with the system's functional and non-functional requirements.

NON-FUNCTIONAL REQUIREMENTS

1. Performance

- System shall validate user inputs within 500ms

- System shall complete LLM processing within 15 seconds

- All UI interactions shall complete within 100ms

- Graph visualization shall render within 2 seconds

- Homepage shall complete initial load time within 3 seconds

2. Reliability

- System shall handle inputs without crashing 99.9% of the time

3. Usability

- Users shall be able to learn basic operations within 5 minutes

EVIDENCE THE DOCUMENT HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT  
 GitHub will be the chosen CM tool for our group. The repository has already been set up along with a /docs folder that will contain all documentation related to the project.

|  |  |  |
| --- | --- | --- |
| Version | Author | Change |
| 1 | Alberto Escobar | Made initial version of the document |

Table 1: CM tool change log

ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS

* IEEE Std 830-1998: Software Requirements
* IEEE Std 29148: Requirements Engineering
* ISO/IEC/IEEE Std 29148-2018: Systems and Software Engineering
  + Life Cycle Processes
* Requirements Engineering

ADDITIONAL REFERENCES

* + Lamsweerde, A.V., 2009. *Requirements Engineering: From System Goals to UML Models to Software Specifications.* John Wiley