SE 4485: Software Engineering Projects

Fall 2024

Requirement Documentation

|  |  |
| --- | --- |
| Group Number | 3 |
| Project Title | Knowledge Management Assistant (Team B) |
| Sponsoring Company | The Fellows Consulting Group (FCG) |
| Sponsor(s) | Jeff Buchmiller |
| Students | 1. Blythe Williams  2. Nidhi Prakuzhy  3. Roj Pawig  4. Ashley Primrose  5. Humayl Sheryar  6. Dalton Brua |

### ABSTRACT

* brief summary of the entire document

Table of Contents

[ABSTRACT 2](#_Toc177572688)

[LIST OF FIGURES 2](#_Toc177572689)

[LIST OF TABLES 2](#_Toc177572690)

[INTRODUCTION 2](#_Toc177572691)

[USE CASE MODEL FOR FUNCTIONAL REQUIREMENTS 2](#_Toc177572692)

[RATIONALE FOR YOUR USE CASE MODEL 3](#_Toc177572693)

[NON-FUNCTIONAL REQUIREMENTS 3](#_Toc177572694)

[EVIDENCE THE DOCUMENT HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT 4](#_Toc177572695)

[ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS ~ Nidhi 4](#_Toc177572696)

[ADDITIONAL REFERENCES 4](#_Toc177572697)

### LIST OF FIGURES

### LIST OF TABLES

### INTRODUCTION

* introduction to the entire document
  + In this document, we will provide an overview of our use case model, function/non-functional requirements, as well as standards and additional references. This will provide the sponsor with a full understanding of how we intended the project to operate based off user interactions and other scenarios.
* purpose and scope of the document
* description of the structure of the document
  + As per the structure, within the use case model section there will be mentions of the actors, conditions, flow of events, exceptions along with special requirements. Then the use case model will be followed by the rationale. The next section will cover non-functional requirements. Then the rest of the document will show you where our information is being stored along with our standards and any additional references that were attained during the process of creating this document.

### USE CASE MODEL FOR FUNCTIONAL REQUIREMENTS

* GRAPHIC USE CASE MODEL
* TEXTUAL DESCRIPTION

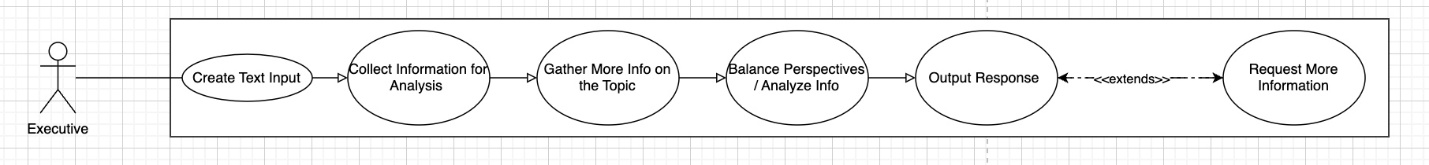
FOR EACH USE CASE

* + Use Case Name
  + Participating Actors
  + Entry Condition(s)
  + Normal Flow of Events
  + Exit Condition(s)
  + Exceptions (Alternate Flow of Events)
  + Special Requirements

Use Case 1: Executive enters input

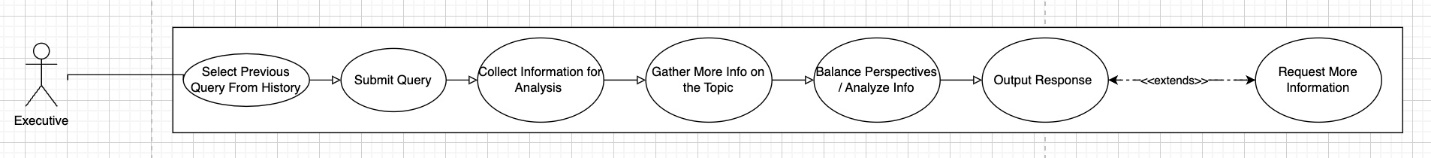
Which programming environment should we utilize for a particular new software system?

* Use Case Name: A knowledge manager asks about programming environment for a new software system
* Participating Actors: Knowledge Manager
* Entry Condition(s): User creates text input
* Normal Flow of Events: text input -> collect info for analysis/gather more info on the topic (multiple iterations) -> balance perspectives/analyze info -> output response
* Exit Condition(s): User is satisfied with the response
* Exceptions (Alternate Flow of Events): text input -> collect info for analysis/gather more info on the topic (multiple iterations) -> balance perspectives/analyze info -> Executive wants to see more information on a specific perspective -> collect info for analysis/gather more info on the topic (multiple iterations) -> output response
* Special Requirements: friendly easy to use design for the user



Use Case 2: User goes through history to find a previous query

* Use Case Name – User wants to re-submit a previous query
* Participating Actors – Knowledge Manager
* Entry Condition(s) - User selects a previous query from history
* Normal Flow of Events
* Exit Condition(s) - User is satisfied with the response
* Exceptions (Alternate Flow of Events)
* Special Requirements



### RATIONALE FOR YOUR USE CASE MODEL

### NON-FUNCTIONAL REQUIREMENTS

* Performance (response time)
  + Response time in 30 seconds or less.
* Usability (User-friendly)
  + Multiple users can act as a knowledge manager.
  + Number of concurrent users up to 10.
  + Only one password for the system and it can be shared among the users outside the system.
* Availability (Always able to access)
  + Utilize servers and software to be available to the team via the university (publicly available for free).
* Scalability (With more information, still produces accurate responses)

### EVIDENCE THE DOCUMENT HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT

### ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS ~ Nidhi

* IEEE Std 1058-1998: Software Project Management Plans
* PMBOK Guide: Project Management Body of Knowledge
* IEEE Std 12207: Software Life Cycle Processes
* IEEE Std 15939: Measurement Process
* IEEE Std 830-1998: Software Requirements
* IEEE Std 29148: Requirements Engineering
* IEEE Std 1471-2000: Software Architecture
* IEEE Std 1016-1998-(Revision-2009): Software Design
* IEEE Std 829-1983: Software Testing
* ISO/IEC/IEEE Std 29119-1-(Revision-2022): Part 1 – Software Testing General Concepts
* ISO/IEC/IEEE Std 29119-2-(Revision-2021): Part 2 – Test Process
* ISO/IEC/IEEE Std 29119-3-(Revision-2021): Part 3 – Test Documentation
* ISO/IEC/IEEE Std 29119-4-(Revision-2021): Part 4 – Test Techniques
* ISO/IEC/IEEE Std 29148-2018: Systems and Software Engineering – Life Cycle Processes – Requirements Engineering
* ISO/IEC/IEEE Std 42030:2019: Software, Systems and Enterprise – Architecture Evaluation Framework

### ADDITIONAL REFERENCES

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