SE 4485: Software Engineering Projects

Fall 2024

Test Plan

|  |  |
| --- | --- |
| 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

* This document provides a comprehensive overview of the testing plan and methodologies for the Knowledge Management Assistant Application. It includes the purpose, scope, structure, and outlines the requirements as well as specifications-based system level test cases. Additionally, it discusses the techniques used for test generation, quality measurement of tests, and the traceability of test cases to use cases. The document also highlights the application of engineering standards and multiple constraints to ensure the reliability and effectiveness of the testing process.

Table of Contents

[ABSTRACT 2](#_Toc254748079)

[LIST OF TABLES 2](#_Toc996938914)

[INTRODUCTION 2](#_Toc1613557776)

[REQUIREMENTS/SPECIFICATIONS-BASED SYSTEM LEVEL TEST CASES 2](#_Toc780435516)

[TECHNIQUES FOR TEST GENERATION 3](#_Toc472263006)

[TRACEABILITY OF TEST CASES TO USE CASES 4](#_Toc1297669146)

[EVIDENCE THE TEST PLAN HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT 5](#_Toc1218387461)

[ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS 5](#_Toc489593575)

[ADDITIONAL REFERENCES 5](#_Toc1473753199)

## LIST OF TABLES

Table 1.1 Requirements/Specifications System Level Test Cases

Table 1.2 Traceability of Test to Use cases

## INTRODUCTION

* This document serves as the test plan for the knowledge management assistant application. It aims to provide a detailed framework and methodologies for testing the system to ensure that it meets the defined requirements and performs as expected. The test plan outlines various testing strategies, and tools that will be employed throughout the testing process.
* The primary purpose of this document is to define the testing approach and processes for the application. It covers all aspects of testing, including unit testing, integration testing, system testing, and acceptance testing. The scope of the documents extends to include the identification of test cases based on system requirements, the techniques used for generating tests, and the criteria for evaluating test quality.
* The document is structured to provide a logical flow of information, starting with the abstract and followed by the table of contents. It includes detailed sections on requirements and specifications-based system-level test cases, techniques for test generation, and the traceability of test cases to use cases. Additionally, it discusses the configuration management of the test plan and the application of relevant engineering standards and constraints. The document also concludes with a list of additional references that provide further insights into software testing methodologies.

## REQUIREMENTS/SPECIFICATIONS-BASED SYSTEM LEVEL TEST CASES

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE ID | TEST NAME | TEST STEPS | EXPECTED RESULT |
| TC01 | Input valid text | 1. Navigate to text input field/form.  2. Type valid text. | Text is displayed in the input field. |
| TC02 | Process valid text | 1. Enter “Sample Question" into the input field.  2. Submit the input. | Text is processed successfully and returns an appropriate response. |
| TC03 | Process invalid text | 1. Enter “Invalid text" e.g. whitespace/invalid characters into the input field.  2. Submit the input. | Application flags invalid input gracefully and returns an appropriate response. |
| TC04 | Gather information from web | 1. Enter question that can answered from the Web into input field.  2. Submit.  3. Observe gathered information. | Application connects to search engine API and returns relevant information from the Web. |
| TC05 | Analyze gathered information | 1. Submit a query.  2. Observe gathered information  3. Observe analysis of gathered information. | Application connects to search engine API and AI services , returning summarized and analyzed information. |
| TC06 | Display response | 1. Submit a query.  2. Check the response displayed. | Response is displayed appropriately and accurately. |
| TC07 | Save queries and responses | 1. Submit a query.  2. Check the database for the stored entry.  3. Check history page for stored entry. | Data is stored successfully into database and is visible in history page. |
| TC08 | View query history | 1. Navigate to the "History" tab.  2. View the list of past queries. | Past queries are displayed. |
| TC09 | Modify and re-submit query | 1. Select a query.  2. Modify the text.  3. Re-submit. | Query is re-submitted successfully. |
| TC10 | Search history | 1. View history.  2. Select search input.  3. Input search ID, text, etc. | History returns relevant queries related to search. |
| TC11 | Login with valid credentials | 1. Enter valid username and password.  2. Click "Login". | User logs in successfully and is sent to dashboard. |
| TC12 | Change password | 1. Login.  2. Go to "Settings".  3. Change password.  4. Save changes. | Password is updated successfully and is reflected in database. |
| TC13 | Save query | 1. Submit a query.  2. Save it.  3. Check database. | Query is saved successfully and is reflected in database. |
| TC14 | Discard query | 1. Submit a query.  2. Click "Discard". | Query is discarded successfully and cannot be found in database. |
| TC15 | Delete stored result | 1. View history  2. Select stored result  3. Select “Delete.” | Stored result is deleted successfully and is reflected in history and database. |
| TC16 | Authorize valid users | 1. Login with valid credentials.  2. Access restricted area. | Access is granted to authorized users. |
| TC17 | Gather data from AI services | 1. Submit a query requiring AI analysis.  2. Check for AI-enhanced response. | Response includes AI-processed insights. |

Table 1.1 Requirements/Specifications System Level Test Cases

## TECHNIQUES FOR TEST GENERATION

* **Techniques used for test generation**: Due to our application, we believe that black box testing methods will be primarily used. Complemented by white box testing for critical components.
* **Black-box vs White-box testing:** Black box testing will validate user interaction and the applications external behaviors, especially for GUI and database components. White-box testing will verify the correct processing of the queries.
* **Quality measurement of tests:** Test coverage will be assessed based on the number of requirements successfully addressed. Criteria includes code coverage percentage and the validation of functional correctness per requirements.

## TRACEABILITY OF TEST CASES TO USE CASES

* provide a mapping between test cases and use cases
* clearly describe how each requirement in the *Requirements Documentation* is captured in testing

|  |  |  |
| --- | --- | --- |
| REQUIREMENT | TEST CASE(S) | DESCRIPTION |
| Users must be able to input text. (UC01) | TC01 | Test case verifies that users can input valid text and handle empty or invalid input gracefully. |
| K.M.A. must be able to process text. (UC01) | TC02, TC03 | Test cases validate that the text entered by the user is processed correctly, including handling invalid inputs. |
| K.M.A. must be able to gather information from the Web. (UC01) | TC04 | Test case ensures that the system can fetch relevant information from web sources based on valid queries. |
| K.M.A. must be able to analyze information gathered. (UC01) | TC05 | Test case validates that the system analyzes the information it gathers from the web or other sources. |
| K.M.A. must be able to output its response to users. (UC01) | TC06 | Test case ensures that the system displays the output of its analysis and gathered information to the user. |
| K.M.A. must be able to save user queries and responses to a database. (UC01) | TC07 | Test case verifies that the system successfully stores user queries and responses in the database for future reference. |
| Users must be able to access and view a history of queries and responses. (UC02) | TC08 | Test case confirms that users can view a history of their previous queries and responses from the application. |
| Users must be able to select previous queries and modify or re-submit to the application. (UC02) | TC09 | Test case ensures that users can edit and resubmit previously submitted queries. |
| Users must be able to search through previous queries. (UC02) | TC10 | Test case validates that users can search their query history effectively. |
| Users must be able to login. (UC03) | TC11 | Test case verifies that users can log in to the application using valid credentials. |
| Users must be able to change passwords. (UC04) | TC12 | Test case ensures users can change their account password through the application settings. |
| Users must be able to save queries. (UC05) | TC13 | Test case validates that users can save queries for future reference. |
| Users must be able to discard queries. (UC05) | TC14 | Test case ensures that users can discard unwanted or incorrect queries. |
| Users must be able to delete stored results. (UC06) | TC15 | Test case confirms users can delete results saved in the database. |
| K.M.A. must be able to authorize users. (UC06) | TC16 | Test case ensures only authorized users can access restricted sections or functionality of the application. |
| K.M.A. must be able to gather information from AI services. (UC07) | TC17 | Test case verifies that the system can connect with AI services to gather insights or analysis based on user queries. |

Table 1.2 Traceability of Test to Use cases

## EVIDENCE THE TEST PLAN HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT

## ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS

* 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

## ADDITIONAL REFERENCES

* Jorgensen, P.C., 2013. Software Testing: A Craftsman's Approach. Auerbach Publications
* Mathur, A.P., 2013. Foundations of Software Testing, 2/e. Pearson Education