Complexity Analyzer

Sprint 3
10/26/2025

Name	**Email Address**
Ryerson Brower	ryerson.brower178@topper.wku .edu
Name2	WKU email address2

CS 396
Fall 2024
Project Technical Documentation

Contents

List of Figures

1 Introduction

1.1 Project Overview

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Aliquet lectus proin nibh nisl condimentum id. Lorem dolor sed viverra ipsum nunc aliquet. Magna fringilla urna porttitor rhoncus dolor. Bibendum at varius vel pharetra vel turpis nunc eget. Fermentum posuere urna nec tincidunt praesent semper.

Nunc congue nisi vitae suscipit tellus mauris a. Tellus at urna condimentum mattis pellentesque id nibh. Massa tincidunt dui ut ornare lectus. Quisque id diam vel quam elementum. Nunc lobortis mattis aliquam faucibus. Tellus elementum sagittis vitae et. Eget felis eget nunc lobortis mattis aliquam faucibus purus.

1.2 Project Scope

Text goes here.

1.3 Technical Requirements

1.3.1 Functional Requirements

Mandatory Functional Requirements
The system must support the creation and execution of automated unit tests for individual components of the disaster response coordination application to ensure that each unit functions correctly in isolation.
The framework must facilitate automated integration testing to verify that different components of the application work together as intended, ensuring data flows correctly between modules.
The system must provide a mechanism for managing test cases, allowing developers to create, modify, and organize test cases within the automation framework for easy access and execution.
The automation framework must seamlessly integrate with existing CI/CD pipelines to enable automatic triggering of test execution upon code commits or pull requests, ensuring immediate feedback on code quality.
The system must generate detailed reports on test results, including pass/fail rates, code coverage metrics, and historical trends, allowing teams to assess code quality and identify areas for improvement.
Extended Functional Requirements
Ext. Req 1
Ext. Req 2
Ext. Req 3

The functional requirements ensure that the test automation framework effectively supports quality assurance throughout all stages of development. Automated testing validates the functionality of individual modules, allowing early detection of errors before the final integration. Integration testing ensures that interconnected components communicate and function correctly before deployment. A centralized test case management system enables developers to easily create, modify, and organize tests. This improves accessibility and collaboration efforts. Integration with CI/CD pipelines provides continuous code validation, preventing bugs from progressing into larger problems. Finally, detailed reporting on test performance along with tracing the code through historical reports helps teams track quality trends in order to make data-driven improvement decisions. Together, these functions streamline the development process, enhancing reliability, and ensures all systems components perform as intended under real-world conditions.

1.3.2 Non-Functional Requirements

The non-functional requirements focus on making the test automation framework reliable, fast, and easy to use. Strong error handling and logging are needed so developers can quickly see what went wrong during testing and fix problems without wasting time or causing delays. The tests should run in under five minutes to give

Mandatory Non-Functional Requirements
The framework must include robust error handling and logging mechanisms to capture and report any failures or exceptions during test execution, providing developers with actionable insights for debugging and resolution.
The test automation framework must execute all automated tests within an average time of under 5 minutes to ensure timely feedback during the development process, facilitating rapid iteration.
The system must be designed to accommodate an increasing number of test cases and applications without degradation in performance, allowing the addition of new tests as the project evolves.
The system must implement security best practices to protect test data and configurations, ensuring that sensitive information is not exposed during test execution or reporting processes.
The test automation framework must provide an intuitive and streamlined interface that allows developers to easily create, manage, and run test cases, ensuring a smooth onboarding process with helpful documentation and interactive tutorials to enhance user engagement and satisfaction.
Extended Non-Functional Requirements
Ext. Req 1
Ext. Req 2
Ext. Req 3

quick feedback, helping teams improve efficiency and make improvements right away. The framework also needs to handle test cases and applications without slowing down, so it can grow as the project expands. Security is important to make sure that any test data, results, or settings are kept safe and private at all times. Finally, the system should be simple to use with clear instructions, examples, and tutorials that help new developers easily use the application. Altogether, these requirements ensure the network is dependable, secure, and efficient for all development teams.

2 Project Modeling and Design

2.1 System Architecture

2.2 Defined Microservices

2.3 Development Considerations

3 Implementation Approach

3.1 Software Process Model

3.2 Key Algorithms and Techniques Used

3.3 Microservices and Docker Usage

3.4 Container Network Communication Setup and Testing

3.5 Infrastructure as Code approach

4 Software Testing and Results

4.1 Software Testing with Visualized Results

Test Plan Identifier:
Introduction:
Test item:
Features to test/not to test:
Approach:

Test deliverables:
Item pass/fail criteria:
Environmental needs:
Responsibilities:
Staffing and training needs:
Schedule:
Risks and Mitigation:
Approvals:

5 Conclusion

Text goes here.

6 Appendix

6.1 Software Product Build Instructions

Text goes here.

6.2 Software Product User Guide

Text goes here.

6.3 Source Code with Comments

Text goes here.