Software Requirements

Specification

for

ReqEx

Version 1.0 approved

Prepared by:
Zachary Bruggen -------- zbruggen2016@my.fit.edu
Nicholas Epler ---------- nepler2018@my.fit.edu
Ivan Hernandez -------- ihernandez2018@my.fit.edu
Thomas Morrison ------- tmorrison2017@my.fit.edu

Faculty Advisor: Dr. Khaled Slhoub ------- kslhoub@fit.edu

Created 9/30/2021

Table of Contents

Version History

Version	Author(s)	Reason for Change	Date Completed
1.0	Z. Bruggen, N. Epler, I. Hernandez, T. Morrison	Initial Writeup	10/4/2021

1. Introduction

1.1 Purpose

The purpose of this project is to extract specific requirements that are being used in a specific code file. This product will allow the user to have a greater understanding of how the components of the code are interacting with each other by generating specifications. The report generated will contain a detailed breakdown of the program.

1.2 Document Conventions

Abbreviations

IEEE	Institute of Electrical and Electronic Engineers
GUI	Graphical User Interface
SQL	Structured Query Language
API	Application Programming Interface
HTTP	Hypertext Transfer Protocol

1.3 Intended Audience and Reading Suggestions

The intended audience of this document are software development industry professionals and researchers who need a better understanding of the requirements of a program written by other developers, without having to manually test the program.

1.4 Product Scope

The product will produce a working list of the new requirements that are extracted from the source files of the code. The product will be initially developed in C++ with file interaction integrating ‘CMake’ in order to parse the requirements from the code. The product will eventually be developed to read code in other object-oriented languages. The code will not generate a list of fixes to the code itself but will instead be a standalone application that tests the functionality of the code to the requirements that are being extracted. The code will be generating the newer list of requirements as it is based on the original requirements but it is what is being parsed from the data.

1.5 References

More tools on requirement extraction referenced for this document:
[Links]

2. Overall Description

2.1 Product Perspective

This product is a new, self-contained product that is used to generate functional requirements from source code files. The product is designed to help software developers understand the high level functionality of their programs. Currently, code analysis is done through a manual review process where the user needs to analyze each part of the source code by hand to determine what the functions are doing, and how the system is interacting as a whole. This product is designed to speed up that process, by analyzing the different aspects of source code, in order to generate a list of functional requirements that the file is trying to meet. With this output, developers can then see if they are meeting the specifications of their own personal projects, and determine what they need to do differently to meet time.

2.2 Product Functions

2.2.1 Upload Input Files

2.2.1.1 The system shall be able to accept source code for any high level language as valid input.

2.2.1.2 The system shall be able to accept header files as valid input.

2.2.1.3 The system shall be able to accept text files as valid input.

2.2.1.4 The system shall allow the user to upload multiple files at once.

2.2.1.5 The system shall allow the user to upload a folder.

2.2.2 Display Uploaded Files

2.2.2.1 The system shall be able to display all previously uploaded individual files.

2.2.2.2 The system shall be able to display all previously uploaded folders.

2.2.3 Select and Navigate Through Files

2.2.3.1 The system shall allow the user to navigate through the listed files.

2.2.3.2 The system shall allow the user to select and switch which file is being displayed.

2.2.4 Generate Requirements Report

2.2.4.1 The system shall be able to generate requirements that encompass the program as a whole.

2.2.4.2 The system shall be able to generate requirements for individual files.

2.2.4.3 The system shall be able to generate requirements for individual methods/functions.

2.2.4.4 The system shall be able to generate requirements from source code.

2.2.4.5 The system shall be able to generate requirements from a text file.

2.2.4.6 The system shall be able to generate requirements from a header file.

2.2.4.7 The system shall be able to allow the user to select between generating a detailed report or a simplified report

2.2.5 Print Generated Report

2.2.5.1 The system shall be able to print a detailed report to the screen.

2.2.5.2 The system shall be able to print a simplified report to the screen.

2.2.6 Save Generated Report

2.2.6.1 The system shall display multiple save formats.

2.2.6.2 The system shall allow the user to select the desired save format for the generated report.

2.2.6.3 The system shall allow the user to select the location to save the generated report.

2.2.7 Modify Generated Report

2.2.7.1 The system shall allow the user to highlight text from the generated report.

2.2.7.2 The system shall allow the user to add text to the generated report.

2.2.7.3 The system shall allow the user to delete text to the generated report.

2.3 User Classes and Characteristics

This product is intended to be used by software developers. In particular, this system is designed for developers or developer teams who understand the basics of the software design and development, where system requirements are extracted from the customers wishes and then translated into code. The intention of this system is to perform an automatic code analysis, extract a list of functional requirements, and offer that to the developer. From there, the developer can determine if their project is on the right track, or if they need to rework something.

Alternatively, software developers may use this product when they have a sample of undocumented source code. In this case, undocumented source code is any sort of code that either does not contain the relevant Software Requirements Specification, or any other relevant design documents. Rather than perform a manual analysis to extract the functional requirements from the code, this software will be able to perform that process automatically.

Though this product is intended for software developers, it could also be extended to software project managers. In this case, the manager needs to ensure that their project is meeting the requirements as outlined by the customer. The system will generate requirements in a way that it can be understood by either developers who are working on a project, or managers who are leading a project.

2.4 Operating Environment

The system will be a standalone software that will be able to run in current Windows, Mac, and Linux operating systems. The system must be able to interact with the SQL database, and the API to interact with that database.

2.5 Design and Implementation Constraints

This system is currently designed to work with high level programming languages. With the current scope of this project, any programming language that is not a high level language will produce an inaccurate output. Along with this, this system is constrained by the style that the user's input file follows. Within the input file, there needs to be some form of English standards to follow in the input. The input cannot be in a different language, and it cannot be composed of gibberish.

2.6 User Documentation

Not applicable at this time

2.7 Assumptions and Dependencies

Due to the imposed constraints on the system, we are assuming that IEEE standards are going to be used in source code file inputs. Along with this, the system is dependent on using source code written in a high level language. Finally, this system is assuming that the input is going to contain English text, rather than any other language.

3. External Interface Requirements

3.1 User Interfaces

The system will utilize a GUI as an interface between the user and the system. The user shall be able to click an upload button to upload the source code file. The user will then select the specific object-oriented language that the source code is written in. The user then clicks a Generate button, this will cause the system to process and display general requirements from the source code. There is an additional section of data that breaks up the individual requirements from basic requirements output, to more individualized requirements that are being identified. The user shall then be able to save the requirements as an output file. From the initial display, the user can highlight from the report, write comments, and provide feedback to the software maintenance team. As well, the user has the ability to see an abridged version of the report, as well as a more detailed version of the report indicating individual requirements generated.

3.2 Hardware Interfaces

Since the system will be a standalone program, it shall be able to interact with the user computer file system. The system shall be able to accept a file to analyze, and create a file of the generated output. To print the report to the screen, the system shall be able to access the host computer's screen. As well the system will be able to interact with the user's individual storage system on the computer to save the data that is provided from the text.

3.3 Software Interfaces

The system shall be able to interact with a SQL database that will break down the parsed data from the program, and process the information in order to allow the system to generate requirements. As well the website interface will be developed using ASP.NET, Angular 11, this integrates a C# backend coding that will allow the user to communicate with the initial application and the web based application to run the code and extract the requirements.

3.4 Communication Interface

The system shall be able to call an API from the application to interface with the SQL database in order to return the data that is being parsed. The API call itself will originate from 'GitHub Pages', generating an API key from their hosted server, this key will be hard coded into the source code from the user, this API will be used transversely between the interface and the SQL database in order to access the vernacular that is being scanned for in the source code. This data transfer will occur over the internet using the HTTP protocol.

4. System Features

4.1 Report Generation

4.1.1 Requirement Extraction from Program

4.1.2 Requirement Extraction from a File

4.1.3 Requirement Extraction from a Method

4.2 File Navigation

Upon clicking “upload” the user can access their file directory. They can upload a single file, multiple files, or a folder containing files.

4.3 Generated Report Options

4.3.1 Print the generated report to the screen

4.3.2 Save the generated report to the file system in multiple formats

4.3.3 Modify the report by either adding or removing text

4.4 Analytical Tools

4.2.1 Keyword Tracker

This feature will allow the user to keep track of key words used throughout the document. Key Words such as variable names and method names will be stored and displayed on a separate panel. The user will also be able to use the panel to locate where in the document the keywords are being used and have them highlighted. This will aid the user while they are performing their static analysis of the code.

4.2.2 Report Complexity

This feature will allow the user to select the complexity of the generated requirements. If the user desires a more simplified report, it will generate a simple list of requirements. Or, the user can select that they want a more complicated report, so the generated report will be more complex, and will indicate keywords that led to that generation of each requirement.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Not applicable at this time

5.2 Safety Requirements

Not applicable at this time

5.3 Security Requirements

5.3.1 Detect SQL Injections

The system must be able to identify and ignore common forms of SQL Injections which may serve as a big vulnerability. An SQL injection is when the input passed into the database is a SQL command allowing malicious users to extract information that otherwise would not be possible.

5.4 Software Quality Attributes

Not applicable at this time

5.5 Business Rules

Not applicable at this time

6. Other Requirements

Not applicable at this time

Appendix A: Glossary

Abbreviations

IEEE	Institute of Electrical and Electronic Engineers
GUI	Graphical User Interface

SQL	Structured Query Language
API	Application Programming Interface
HTTP	Hypertext Transfer Protocol

Appendix B: Analysis Models

Not applicable at this time

Appendix C: To Be Determined List

Not applicable at this time