Software requirements and Specifications for

AI-Powered Code Review Assistant:

Prepared by :

AP21110010926

AP21110010927

AP21110010930

AP21110010931

03-02-2024

Table Of Contents:

1. Introduction:

1.1 Purpose:

This document outlines the software requirements and specifications for the development of an AI-Powered Code Review Assistant. It serves as a comprehensive guide for customers, developers (including designers, testers, and maintainers), providing essential information for the project's understanding and implementation,

-Documents Convention:

1. **Headings and Subheadings**: Use consistent heading styles for sections and subsections. Follow a hierarchical structure (e.g., Heading 1 for main sections, Heading 2 for subsections).
2. **Font Styles**: Differentiate between regular text, headings, and emphasis using font styles. For example, use bold or italic formatting for emphasis.
3. **Code and Commands**: Use a monospaced font (e.g., Courier New) for displaying code snippets or commands. Differentiate code within the text using backticks (`).
4. **Lists**: Clearly distinguish between ordered (numbered) and unordered (bulleted) lists. Use indentation consistently for nested lists.
5. **Annotations**: Use callout symbols (e.g., asterisks, numbers) for annotations or side notes. Link callouts to relevant parts of the text.
6. **Tables**: Clearly label and number tables for easy reference. Use consistent formatting for table headers, content, and footnotes.
7. **Graphics and Diagrams**: Label all figures and diagrams with a clear title and number. Include captions for each visual element to provide context.

1.2 Intended Audience and Reading Suggestions:

-Intended Audience:

. Software Developers
. Software Testers
. Software Designers
. Maintainers
. Customers

- Reading Suggestions :

. Sequential Reading
. Reference Crosschecking
. Visual Elements
. Collaborative Review

1.3 Product Scope:

The product scope for the AI-Powered Code Review Assistant encompasses the features, functionalities, and objectives that define its boundaries and purpose within the software development landscape.

2. Overall Description:

2.1 Product Perspective:

The product perspective for the AI-Powered Code Review Assistant outlines its position within the broader software development ecosystem and how it interacts with other components. This section defines the relationship between the code review assistant and its surrounding environment.

Like: 1. System Architecture, 2. Interaction with Users
3.External Interfaces, 4.Constraints, 5.Scalability

2.2 Product Functions:

The product functions of the AI-Powered Code Review assistant encompass the core capabilities and features that contribute to its role in enhancing code quality, improving developer efficiency, and facilitating collaborative code reviews. These functions are designed to provide an intelligent and context-aware code review experience.

. Automated Code Review

. Context-Aware Suggestions

. Collaborative Code Reviews

. Customizable Rule Sets

. Learning and Adaptation

. Integration with IDEs and VCS

. Performance Monitoring

. Security and Compliance Checks

. Usage Analytics

2.3 User Classes and Characteristics:

-User Classes:

a) System Administrators
b) Team Managers
c) QA (Quality Assurance) Teams
d) System Administrators
e) Project Stakeholders
f) Security Analysis
g) Compliance Officers

-Characteristics:

a) Developers:

. Proficient in programming languages.

. Actively involved in the software development process.

b) Team Leads/Managers:

. Oversee development projects.

. Responsible for code quality and team productivity.

c) Quality Assurance (QA) Teams:

. Focus on software quality and testing.

. Ensure compliance with coding standards.

d) System Administrators:

.Manage software development infrastructure.

. Responsible for tool integrations and configurations.

e) Project Stakeholders:

. Have an interest in project outcomes.

. Concerned with overall project health.

f) Security Analysts:

.Specialized in identifying security vulnerabilities.

.Knowledgeable about secure coding practices.

g) Compliance Officers:

. Familiar with coding regulations and standards.

. Responsible for legal and regulatory compliance.

2.4 Operating Environment:

This operational environment outlines the necessary hardware, software, network, security, and usability considerations for the effective deployment and utilization of the AI-Powered Code Review Assistant in diverse development scenarios.

2.5 Design and integrative constraints:

These design and integration constraints are critical considerations to ensure the effective deployment, usability, and seamless integration of the AI-Powered Code Review Assistant within diverse development environment.

2.6 User Documentation:

The user documentation for the AI-Powered Code Review Assistant has been meticulously crafted to empower users with the essential knowledge required for efficient utilization of the tool. With a comprehensive "Getting Started Guide," users are guided through the setup and configuration process, offering step-by-step instructions for installation on various operating systems. The documentation provides an insightful overview of the user interface, elucidating the main features and navigation pathways. User roles and permissions are clearly defined, allowing users to manage roles effectively within the code review assistant. Detailed workflows for initiating and participating in code reviews, integration guides for popular IDEs, and best practices for version control system integration contribute to a robust understanding of the tool's capabilities. Users are further equipped with customization options, enabling them to tailor the code review environment to their preferences. The documentation delves into real-time collaboration tools, communication channels, and the interpretation of code metrics, offering users a holistic understanding of the assistant's capabilities. A comprehensive troubleshooting guide and a compilation of frequently asked questions (FAQs) ensure users can address common issues independently. Lastly, users are directed to available support channels, including helpdesk and community forums, fostering a supportive environment for continued learning and problem resolution. This user documentation establishes a foundation for a smooth onboarding process, empowering users to leverage the AI-Powered Code Review Assistant effectively throughout their development workflows.

2.7 Assumptions and Dependencies:

. Stable Internet Connection
. Availability of Supported Browsers (e.g., Chrome, Firefox)
. Adequate System Resources (RAM, CPU) for Browser Operation
. Compliance with Standard Security Protocols
. Developer Familiarity with Code Review Processes
. Availability of Version Control Systems (e.g., Git)
. Accessibility to Code Repositories for Integration
. Compatibility with Common Operating Systems (Windows, macOS, Linux)
. Timely Updates and Maintenance of the Code Review Assistant
. Adequate User Training and Familiarity with Software Development Practices.

3. External Interface Requirements:

3.1 User Interfaces:

.Graphical User Interface (GUI) for User Interaction.

. Intuitive Navigation and Input Mechanisms.

3.2 Hardware Interfaces:

.Compatibility with Standard Computer Hardware.
.Input/Output Support for Peripheral Devices.

3.3 Software Interfaces:

.Integration with Supported Integrated Development Environments (IDEs)
.Compatibility with Version Control Systems (e.g., Git)

3.4 Communications Interfaces:

.Standard Protocols for Communication.
.Secure Data Transmission over the Network.

4. System Features:

- **a.** Code Review Initiation
- **b.** Collaborative Review Process
- **c.** Automated Code Analysis
- **d.** Version Control Integration
- **e.** Customizable Workflows
- **f.** Commenting and Annotation
- **g.** Code Metrics and Insights
- **h.** Notification System
- **i.** Dashboard and Reporting
- **j.** Role-Based Access Control
- **k.** Integration with Issue Tracking
- **l.** Continuous Integration Support
- **m.** Code Snippet Sharing
- **n.** Archiving and Documentation
- **o.** Search and Filtering Capabilities
- **p.** Offline Mode
- **q.** Multilingual Support
- **r.** Code Review Metrics Visualization
- **s.** Security Measures
- **t.** API Accessibility.

5. Other non-functional Requirements:

1. Performance Requirements
2. Safety Requirements
3. Security Requirements
4. Business Rules.

6. Other Requirements:

The AI-Powered Code Review Assistant should be implemented on machines with a minimum of 100GB free space on the hard disk drive (HDD) for database storage, with a recommended storage capacity of 500GB for server environments. The system should operate smoothly with a minimum of 64MB RAM for database operations, while the recommended RAM for server environments is 512MB. Additionally, the Code Review Assistant must seamlessly integrate with MS Access applications and MS SQL Server for optimal functionality and data management.

Appendix-A: Glossary:

The glossary provides a concise reference for key terms used in the context of the AI-Powered Code Review Assistant. It includes definitions for terms such as "Code Review," "Pull Request (PR)," "Static Code Analysis," and "Continuous Integration (CI)," among others. This section serves as a quick guide for users, developers, and stakeholders, ensuring a shared understanding of essential concepts related to code review, collaboration, and software development practices.

Appendix-B: Analysis Models:

Appendix B provides a set of analysis models that offer a comprehensive understanding of the AI-Powered Code Review Assistant's structure and functionality. These models include visual representations such as use case diagrams, activity diagrams, sequence diagrams, collaboration diagrams, class diagrams, component diagrams, and deployment diagrams. These models collectively contribute to a holistic view of the system's behavior, interactions, and architecture. They serve as valuable tools for both developers and stakeholders to comprehend the intricacies of the code review process and the underlying components facilitating seamless collaboration in software development.