**一、Test plan**

**1、Testing Objectives**

The purpose of this testing is to ensure that the functionality, performance, and user experience of the software system meet the requirements specified in the requirements document. Specific objectives include:

(1)Verifying that all system functions meet user and business requirements.

(2)Checking that the system’s performance reaches expected standards.

(3)Ensuring the system’s usability, security, reliability, and maintainability.

(4)Identifying and reporting any defects in the system to ensure the quality of the final delivered product.

**2、Testing Strategy**

(1)Functional Testing: Use unit tests to verify that each module’s functionality meets the requirements, and perform integration and system testing to ensure the system’s functions comply with relevant specifications.

(2)Security Testing: Conduct security testing focused on user data safety and data transmission encryption.

(3)Compatibility Testing: Run the system in different environments to assess its performance across platforms.

(4)User Experience Testing: Simulate user scenarios to ensure the system’s usability and intuitiveness.

**3、Testing Environment**

(1)Operating System: Windows, macOS

(2)Browser: Chrome, Firefox, Edge

(3)Database: Mysql

**4、 Types of Testing**

(1) Unit Testing

Objective: Verify the correctness of each module, function, and class.

Tester: JiaTeng Yang,YanWei Liu.

Tools: pytest

(2) Integration Testing

Objective: Verify that the integration interfaces between modules work correctly.

Tester: All group members.

Tools: Postman

(3) System Testing

Objective: Examine the overall functionality and performance of the system.

Tester: All group members.

Tools: Selenium.

(4)Regression Testing

Objective: Verify system stability after bug fixes or new feature additions.

Tester: YanWei Liu.

Tools: Selenium.

(5)User Acceptance Testing

Objective: Ensure the system meets users’ business requirements.

Tester: JiaTeng Yang,YanWei Liu.

Tools: No specific tools; primarily based on user feedback.

**5、Testing Exit Criteria**

(1)All high-priority defects have been resolved and verified.

(2)Primary functionality modules have passed testing and meet the requirements in the requirements document.

(3)Non-functional requirements such as performance and compatibility meet the established standards.

(4)Acceptance testing has been completed with sign-off from the team lead.

**二、Functional testing**

**1、Unit Testing**

Verify that every part of the software operates as expected. By testing smaller code units in an isolated environment, we can locate the cause of the problem and reduce the workload of subsequent debugging.The unit test cases and test results are as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use case number | Function point | Test steps | Expected results | Actual results |
| UiModule01 | Upload code | User uploads python code | Uploaded successfully | Correct |
| UiModule02 | View CWE issues | Click to view CWE issues and observe the displayed results. | Jump to analysis results and its visual report interface | Correct |
| UiModule03 | View the code score. | Click the View Code Quality Score button to observe the results. | Correctly display the score results and their reports. | Correct |
| UiModule04 | View historical record | Click to view history and observe the displayed results. | Jump to the history interface to display trend charts and distribution charts of CWE issues and code quality scores. | Correct |
| UiModule05 | Analyze CWE issues | Upload custom configuration files to classify and detect CWE issues based on the specific needs of your project. | The system will identify CWE issues based on the configuration file | Correct |
| UiModule06 | Export historical data | Click Export Historical Data and watch the system download the data. | When you click the "Export" button and select CSV or PDF format, the system should download the historical data file. | Correct |
| Login1 | User login | Enter the correct username and password | Jump to home page | Correct |
| Login2 | User login | Enter the wrong username and password | Prompt error | Correct |
| Login3 | View information | Click on personal information | Success | Correct |
| Login4 | User registration | Register new user | Success | Correct |
| Login5 | Retrieve password | Enter username and mobile phone number to retrieve password | Success | Correct |
| Login6 | Modify user information | Modify user personal information | Success | Correct |
| Login7 | View all users | View user form | Success | Correct |
| Login8 | Delete user | Delete user information | Success | Correct |
| File1 | Analyze code issues | View the CWE problem analysis results and compare them with the actual problems in the code | The analysis results are consistent with the actual problem | Correct |
| File2 | Code problem classification | View the classification result of the code issue, compare it with the standard classification result, and observe whether it is correct | The system classification results are consistent with the actual classification results | Correct |
| File3 | Code quality score | Observe the system scoring results and compare them with expert scores. | The deviation between the system scoring results and the expert scoring results is within 0.1 | Correct |
| File4 | Override default rules | Upload the customized configuration file and compare the problem analysis results before and after modification | After uploading the file, the analysis results meet the specific needs of the project | Correct |
| Chatgpt1 | Intelligent interaction | Enter relevant CWE questions, scoring results, and view their responses | Basically the same as the standard chatgpt reply result | Correct |
| Database01 | Store data | Save the analysis results and view them in the database management system | Data consistent with the user interface can be viewed in the management system | Correct |
| Database02 | Operate data | Perform data addition, deletion, modification and query operations on the user interface, and enter the database management system to view the results. | The management system can correctly display the relevant operation results of the database | Correct |
| Visualization01 | Visualization of the number of CWE questions | View the CWE question count line chart and observe the results. | The displayed results are consistent with the system analysis results | Correct |
| Visualization02 | Visualization of the distribution of CWE questions | View the bar chart and pie chart of the CWE question and observe the results | The displayed results are consistent with the system analysis results | Correct |
| Visualization03 | Visualization of code quality score changes | Line chart of code quality score changes, observation results | The displayed results are consistent with the system analysis results | Correct |
| Visualization04 | Visualization of the relationship between CWE issues and code quality scores | Heat map or scatter plot of CWE issues and code quality scores, observations | The displayed results are consistent with the system analysis results | Correct |
| Report01 | Report generation | Click the report generation button to download the report and observe its results | Successfully downloaded PDF report containing visual charts and data summary | Correct |

**2、Integration testing**

In order to detect interface defects between integrated components, discover problems in inter-module communication, and verify whether the data flow in each part of the system is as expected, we perform integration testing on each module. This test was conducted from bottom to top using the Postman tool, and interface defects were discovered and modified.

**3、System testing**

In order to verify the functionality, reliability and performance of the entire system and ensure that the software can work as expected after all components are integrated, our team conducted an overall test of the system. First, end-to-end testing was performed, and all test cases of unit testing and integration testing were retested. Then the compatibility was tested on Windows system and Mac system. Finally, Jputer was used for load stress testing.

**三、Other testing**

**1、Security testing**

To verify that the system’s data and security measures can effectively counter potential threats, ensuring user privacy and system stability. The test scope includes data encryption, data transmission security, database backup functionality, system load, and recovery capabilities. Data security testing involves validating user information encryption, HTTPS data transmission security, and database backup functionality. System security testing includes evaluating responsiveness under high-load conditions, effectiveness of caching strategies, recovery speed in abnormal situations, and database indexing optimization. Wireshark is used for encryption validation, JMeter for simulating high concurrent users, database management tools for verifying backup and recovery functions, and Redis for checking cache effectiveness. The exit criteria require sensitive data encryption, stable system operation under high load and abnormal conditions, intact backup functionality, and expected cache hit rates. If issues such as data leakage, cache failure, or load insufficiency are found, immediate correction is necessary to ensure that the system protects user data and maintains reliability against potential attacks.

**2、Visualization Testing**

The testing of the data visualization module verified that the charts accurately present the system analysis results. The CWE issue trend chart displayed the changes in issues over time, with smooth data point connections and clear trends. The CWE issue distribution chart, using bar or pie charts, clearly showed the distribution of issues across different types and levels, with reasonable proportions and distinct color separations. The code quality score change chart accurately displayed the fluctuations in project scores over time, with easily identifiable trends. The CWE issues vs. code quality score relationship chart effectively displayed the correlation between the two, with clear data distribution, helping users understand the relationship between the number of issues and the code quality score. All charts met expectations, being accurate, intuitive, and interactive, aiding users in quickly understanding the analysis results.