System Testing Document

1 System Testing Objectives

The objective of this system testing is to verify whether the features of the "Poetry Enjoyment Life" web application perform as expected. The test cases cover various functionalities, including login and registration, model switching, question recommendations, chat functionality, and history record saving. By performing comprehensive testing, we ensure that each functionality operates correctly, providing users with a secure, efficient, and user-friendly system. These tests are designed to ensure the reliability of the system and offer a good user experience. Through these testing methods, we can effectively identify and fix system defects, ultimately improving software quality.

2 Testing Environment Requirements:

Hardware Requirements: Standard computer or mobile device with internet connectivity.
Software Requirements:
Operating System: Windows 10 or later, macOS, Android, or iOS.
Browsers: Chrome, Firefox, Safari, or Edge, with a version that supports JavaScript.
Required Applications: The chatbot system must be logged in and running.

3 System Testing Methodology

1. **Functional Testing**: The purpose of functional testing is to validate whether all features of the software are implemented correctly as specified in the requirements documentation. This includes checking user interfaces, feature functionality, database access, and external interfaces to ensure they operate correctly. Common techniques used include equivalence class partitioning, boundary value analysis, and decision tables. For example, equivalence class partitioning divides input data into valid and invalid classes, and corresponding test cases are designed to verify the functionality under each condition.

2. **Black-box Testing**: Black-box testing treats the software as an opaque box, focusing solely on inputs and outputs, without considering the internal implementation. The main focus of this test is to verify whether the software meets user requirements,

without considering the internal logic or code structure. Test cases are designed based on user needs and functional specifications to cover various functional scenarios.

4 System Test Cases

1 Login Functionality Test

Id: jc01
Project: Login Functionality Test
Content: User login functionality test
Preconditions: The system has existing users and relevant information.
Test Steps:
1. Open the "Poetry Enjoyment Life" webpage and go to the login page.
2. Enter the correct username and password, then click the login button.
Expected Result: Successful login and redirection to the homepage.
Result and Conclusion: Success

2 Registration Functionality Test

Id: jc02
Project: Registration Functionality Test
Content: New user registration for a new account
Preconditions: The system has no prior knowledge of the user.
Test Steps:
1. Enter the "Poetry Enjoyment Life" website.
2.Click the "Register" button.
3.Enter the required account and password.
Expected Result: Registration successful.
Result and Conclusion: Success

3 Model Switching Functionality Test

Id: jc03
Project: Model Switching Functionality Test
Content: User switches between models in the interaction page.
Preconditions: System is functioning properly.
Test Steps:

1. Click the "Model Switch" button.
2.Select the desired model.
3.Click the "Confirm" button.
Expected Result: Successful model switch with feedback message.
Result and Conclusion: Success

4 Question Recommendation Functionality Test

Id: jc04
Project: Question Recommendation Functionality Test
Content: Successful model switch with feedback message.
Preconditions: User has successfully logged in.
Test Steps:
1. Enter the robot interaction page.
2.Select a recommended question
3.Send the question and wait for the robot’s response.
Expected Result: The robot provides an answer.
Result and Conclusion: Success

5 Chat Functionality Test

Id: jc05
Project: Chat Functionality Test
Content: User asks the robot questions and receives answers.
Preconditions: User has successfully logged in.
Test Steps:
1. Enter the robot interaction page.
2.Ask a question and click "Send".
3.Wait for the robot’s response.
Expected Result: The robot answers successfully.
Result and Conclusion: Success

6 History Record Saving Functionality Test

Id: jc06
Project: History Record Saving Functionality Test
Content: Save previous chat content with the robot for future reference.
Preconditions: There is already a chat history.
Test Steps:

1. Click "Save Record".
2.Go to the history record page.
3.Check if the previous chat content is available.
Expected Result: The chat record should be found and displayed.
Result and Conclusion: Success

5 测试结果

Test Case ID	Test Case Name	Test Result	Remarks
JC01	Login Functionality Test	Pass	No issues encountered
JC02	Registration Functionality Test	Pass	No issues encountered
JC03	Model Switching Functionality Test	Pass	No issues encountered
JC04	Question Recommendation Functionality Test	Pass	Robot successfully recommends and answers questions
JC05	Chat Functionality Test	Pass	Questions and answers function properly
JC06	History Record Saving Functionality Test	Pass	History records are saved and displayed correctly

6 Test Summary

Test Coverage: This testing covered core functionalities such as login, registration, chat, model switching, question recommendation, and history record saving.
Pass Rate: 100% of test cases passed.
Optimization and Improvement Suggestions:
Additional boundary condition testing could be conducted, particularly for model switching and history record saving under exceptional cases (e.g., network issues, no saved records).
The registration process could include email verification to enhance account security.
The history record page could be optimized for faster loading, especially when dealing with large amounts of records.
Add error handling and help documentation to improve the user experience in case of failures.

Overall, the system is stable and meets the expected user experience.