Test part:

1. unit test

Unit testing is the testing of program modules to check correctness. A program unit is the smallest testable part of an application. In procedural programming, a unit is a single program, function, procedure, etc. We chose to conduct our unit tests later in the programming process, and as most of the problems were already solved during the programming process, our tests generally found no obvious logical anomalies and only some redundant code. We tested mainly modules in the database and back-end code, with the main testers being back-end programmers

Here are some test codes in framework:

文本

描述已自动生成文本

描述已自动生成

文本

描述已自动生成文本

描述已自动生成

And here is test result about some example:

图形用户界面, 应用程序

描述已自动生成

1. system test:

System testing is the testing of the entire system, looking at the hardware, software and operators as a whole and checking that it does not conform to the system specification in any way. This type of testing can identify errors in the analysis and design of the system. Security testing, for example, tests whether security measures are sound and whether the system can be secured against illegal intrusion. We use two main types of testing, white box testing and black box testing

* 1. White box test:

White box test case requirements:

1. Ensure that all independent paths in a module are used at least once

2. All logic values need to be tested

3. Run all loops within the upper and lower boundaries and the operable range

4. Check the internal data structure to ensure its validity

The purpose of white-box testing is to test the coverage of logical paths in software by checking the internal logical structure of software. Checkpoints are set up at different places in the program to check the status of the program to determine whether the actual running status is consistent with the expected status. A control structure flow diagram has been chosen to present the test results:

图示

描述已自动生成图示

描述已自动生成图示

描述已自动生成

* 1. Black box test

Black box testing, which is testing to check that every function works properly. In testing, the program is seen as a black box that cannot be opened and is tested at the program interface without any consideration of the program's internal structure and internal characteristics. It only checks that the program functions work properly as specified in the requirements specification and that the program receives the input data appropriately to produce the correct output information.

Login function (Login page)

Equivalence Class form:

|  |  |  |
| --- | --- | --- |
|  | Valid | Invalid |
| Username | 1.Username contains only letters  2.Username contains only numbers.  3.Username contains letters and characters | 4.Username contains spaces  5.No input username  6.Enter a non-existent username |
| Password | 7.Password contains only letters  8.Password contains only numbers  9.Password contains letters and characters | 10.Password contains spaces  11.No input password  12.Enter the wrong password  13.Enter a non-existent password |

Test Cases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case number | Test Scenario Description | Number of the covered equivalence class | Expected Output | Software output results |
| 1 | Username: Eden  Password: abcdef | 1, 7 | Login successful | Consistent |
| 2 | Username: 33051  Password:123456 | 2, 8 | Login successful | Consistent |
| 3 | Username: Eden1117  Password: abc123 | 3, 9 | Login successful | Consistent |
| 4 | Username: E den  Password: 123456 | 4, 8 | Login failed | Consistent |
| 5 | Username: (empty)  Password:123456 | 5, 8 | Login failed | Consistent |
| 6 | Username: Vincent  Password: abcdef | 6, 7 | Login failed | Consistent |
| 7 | Username: Eden1117  Password: a bc1 23 | 3, 10 | Login failed | Consistent |
| 8 | Username: Eden  Password: (empty) | 1, 11 | Login failed | Consistent |
| 9 | Username: 33051  Password: jfgorit | 2, 12 | Login failed | Consistent |
| 10 | Username: Eden1117  Password: wedajds | 3, 13 | Login failed | Consistent |

Check username available function(Register page)

Equivalence Class form:

|  |  |  |
| --- | --- | --- |
|  | Valid | Invalid |
| Username | 1.Enter a new, unused username  2.Enter a username that is already in use | 3. No username entered |

Test Cases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case number | Test Scenario Description | Number of the covered equivalence class | Expected Output | Software output results |
| 1 | Username: Eden1118 | 1 | Username available | Consistent |
| 2 | Username: Eden1117 | 2 | Username not available | Consistent |
| 3 | Username: (empty) | 3 | Username not available | Inconsistency |

Register function(Register page)

Equivalence Class form:

|  |  |  |
| --- | --- | --- |
|  | Valid | Invalid |
| Username | 1.Username contains only letters  2.Username contains only numbers.  3.Username contains letters and characters  4.Username contains spaces | 5.No input username  29.Input a exist username |
| First name | 6.First name contains only letters  7.First name contains only numbers  8.First name contains letters and characters  9.First name contains spaces | 10.No input first name |
| Last name | 11.Last name contains only letters  12.Last name contains only numbers.  13.Last name contains letters and characters  14.Last name contains spaces | 15.No input last name |
| Identity choose | 16.Choose client identity  17.Choose team member identity  18.Choose manager identity |  |
| Password | 19.Password contains only letters  20.Password contains only numbers  21.Password contains letters and characters  22.Password contains spaces | 23.No input password |
| Confirm password | 24.Password contains only letters, match the first password  25.Password contains only numbers, match the first password  26.Password contains letters and characters, match the first password  27.Password contains spaces, match the first password | 28.Confirm password does not match the first password |

Test Cases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case number | Test Scenario Description | Number of the covered equivalence class | Expected Output | Software output results |
| 1 | Username: Eden  First name: Eden  Last name: Wang  Identity: Client  Password: abcdef  Confirm password: abcdef | 1, 6, 11, 16, 19, 24 | Register successful | Consistent |
| 2 | Username: 111  First name: 1111  Last name: 1111  Identity: Team member  Password: 123456  Confirm password: 123456 | 2, 7, 12, 17, 20, 25 | Register successful | Consistent |
| 3 | Username: Eden1117  First name: Eden1111  Last name: Wang1111  Identity: Manager  Password: abc123456  Confirm password: abc123456 | 3, 8, 13, 18, 21, 26 | Register successful | Consistent |
| 5 | Username: Yi dan  First name: Yidan  Last name: Wang  Identity: Manager  Password: 123456  Confirm password: 123456 | 4, 6, 11, 18, 20, 25 | Register successful | Consistent |
| 6 | Username: (empty)  First name: Yidan  Last name: Wang  Identity: Manager  Password: 123456  Confirm password: 123456 | 5, 6, 11, 18, 20, 25 | Registration failed and prompted for a username | Consistent |
| 7 | Username: Yidan  First name: (empty)  Last name: Wang  Identity: Manager  Password: 123456  Confirm password: 123456 | 1, 10, 11, 18, 20, 25 | Registration failed and prompted for a first name | Consistent |
| 8 | Username: Yidan  First name: Yidan  Last name: (empty)  Identity: Manager  Password: 123456  Confirm password: 123456 | 1, 6, 15, 18, 20, 25 | Registration failed and prompted for a last name | Consistent |
| 9 | Username: Yidan1  First name: Yidan  Last name: Wang  Identity: Manager  Password: 12345 6  Confirm password: 12345 6 | 3, 6, 11, 18, 22, 27 | Register successful | Consistent |
| 10 | Username: Yidan1  First name: Yidan  Last name: Wang  Identity: Manager  Password: (empty)  Confirm password: 123456 | 3, 6, 11, 18, 23, 28 | Registration failed and prompted for a password | Consistent |
| 11 | Username: Yidan1  First name: Yidan  Last name: Wang  Identity: Manager  Password:123456  Confirm password: (empty) | 6, 11, 18, 20, 28, 29, | Registration failed and prompted for a confirm password | Consistent |