AITA

软件工程专业综合项目测试报告

1652714 孙浩然

16///// 陈泽徽

16///// 梁钧清

16///// 王嵩豪

1652708 周泽林

1652698 康晓博

## 背景与依据

AITA需求规约文档

AITA需求分析文档

AITA概要设计规约文档

## 选择三个模块进行测试

选择的三个模块分别为用户认证模块，专注度模块和用户信息模块。

用户认证模块是一个非常重要的模块，每一位使用本系统的用户都需要使用本模块进行用户认证。模块中的主要功能为登录和注册，两个功能都包括对数据库内容的增删查改等操作。同时在本系统中，用户登录还使用了 md5 的加密方式。我们决定对用户认证模块采用黑盒测试的测试方法。

专注度模块是本系统的核心功能，在这个模块中，系统接受远端 API 返回的专注度数据，并将专注度数据处理并存储入数据库中。同时还对数据进行处理和分析。我们决定对这个模块采用黑盒测试。

用户信息模块白盒测试。

测试技术和过程

4.1 Black box testing for auth module using ECP

Here's the equivalence classes and their representatives.

|  |  |  |
| --- | --- | --- |
| parameter | equivalence classes | representatives |
| username | vEC11: a certain sting inside database username storage | "init" |
|  | iEC11: null | "" |
|  | iEC12: a certain string but not in database username storage | "test" |
| password | vEC11: a certain sting matches password of username in database | "rightpwd" |
|  | iEC11: null | "" |
|  | iEC12: a certain string but does not match the password of username | "wrongpwd" |

Here's the representitive test cases.

|  |  |  |  |
| --- | --- | --- | --- |
|  | parameter | |  |
| test case | username | password | result |
| 1 | "init" | "rightpwd" | success login |
| 2 | "init" | "" | invalid input |
| 3 | "init" | "wrongpwd" | wrong password |
| 4 | "" | "rightpwd" | invalid input |
| 5 | "" | "" | invalid input |
| 6 | "" | "wrongpwd" | invalid input |
| 7 | "test" | "rightpwd" | wrong username |
| 8 | "test" | "" | invalid input |
| 9 | "test" | "wrongpwd" | wrong username |
| ... | ... | ... | ... |

4.2 Black box testing for focus module using BVA

Here's the equivalence classes and their representatives.

|  |  |  |
| --- | --- | --- |
| parameter | equivalence classes | representatives |
| emotion | vEC11: [0, \_, 100] | 50 |
| iEC11: [MIN\_DOUBLE, \_, 0[ | -1 |
| iEC12: [100, \_, MAX\_DOUBLE] | 101 |
| iEC13: NaN | "ABC" |
| gaze | vEC21: [-20, \_, 20] | 0 |
| iEC21: [MIN\_DOUBLE, \_, -20[ | -22 |
| iEC22: [20, \_, MAX\_DOUBLE] | 22 |
| iEC23: NaN | "ABC" |
| rate | vEC31: [0, \_, 1] | 0.5 |
| iEC31: [MIN\_DOUBLE, \_, 0[ | -22 |
| iEC32: [1, \_, MAX\_DOUBLE] | 22 |
| iEC33: NaN | "ABC" |

Here's the lower boundary value, equivalence class and upper boundary value of each parameter.

|  |  |
| --- | --- |
| parameter | LBV, [EC], UBV |
| emotion | 0-δ, [0, \_, 100], 100+δ |
| gaze | -20-δ, [-20, 20], 20+δ |
| rate | 0-δ, [0, 1], 1+δ |

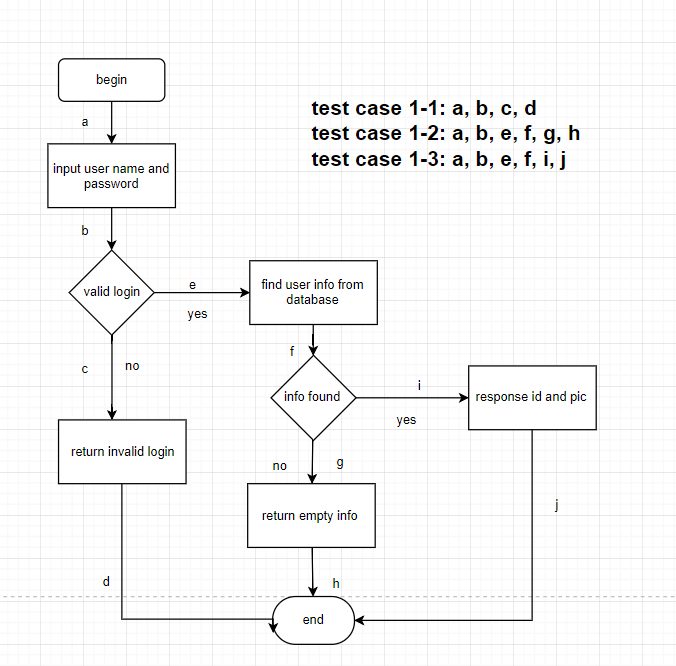
Here's the representitive test cases.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | parameter | | |  |
| test case | emotion | gaze | rate | result |
| 1 | 0 | -20 | 0 | success |
| 2 | 100 | 20 | 1 | success |
| 3 | -0.01 | -20.01 | -0.01 | invalid |
| 4 | 0.01 | -19.99 | 0.01 | success |
| 5 | 100.01 | 20.01 | 1.01 | invalid |
| 6 | 99.99 | 19.99 | 0.99 | success |
| 7 | 0.01 | 20.01 | 0.99 | invalid |
| 8 | 99.99 | -19.99 | 1.01 | invalid |
| ... | ... | ... | ... | ... |

4.3 White box testing for user module

We use white box testing to test the user module. Since in this module we don't have much input conditions, we choose branch testing rather than condition testing or condition determination testing. Here's the control flow graph for branch testing:

Get user info:

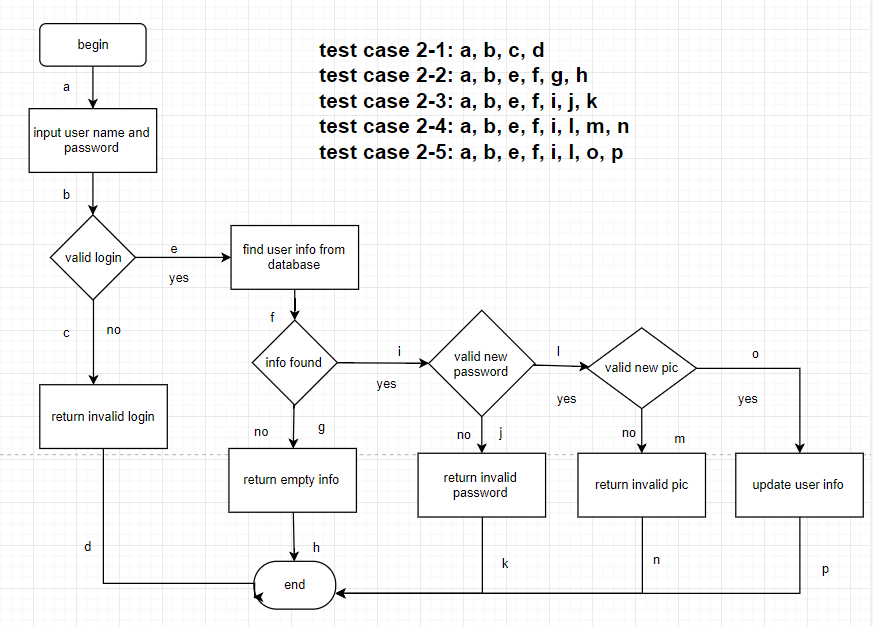


|  |  |
| --- | --- |
| NO. | Test case 1-1 |
| Functional module | User module |
| Case title | Test the input of invalid login |
| Prerequisite | Users input user name and password |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

|  |  |
| --- | --- |
| NO. | Test case 1-2 |
| Functional module | User module |
| Case title | Test user information which is not found in database |
| Prerequisite | Users have logged in successfully |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

|  |  |
| --- | --- |
| NO. | Test case 1-3 |
| Functional module | User module |
| Case title | Test the response id and pictures with valid login and found information |
| Prerequisite | Users have logged in successfully and user information can be found in database |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

Update user info:



|  |  |
| --- | --- |
| NO. | Test case 2-1 |
| Functional module | User module |
| Case title | Test the input of invalid login |
| Prerequisite | Users input user name and password |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

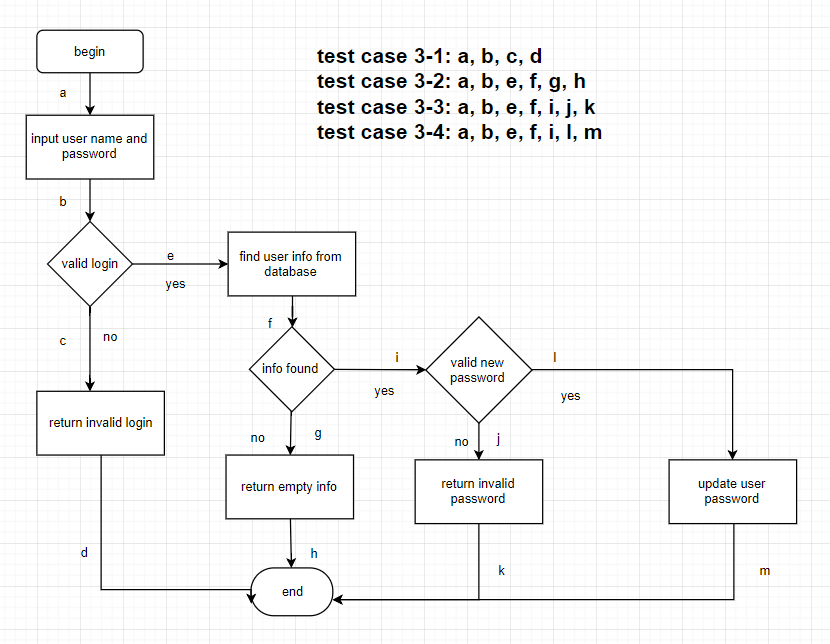
|  |  |
| --- | --- |
| NO. | Test case 2-2 |
| Functional module | User module |
| Case title | Test user information which is not found in database |
| Prerequisite | Users have logged in successfully |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

|  |  |
| --- | --- |
| NO. | Test case 2-3 |
| Functional module | User module |
| Case title | Test the input of invalid password while updating user information |
| Prerequisite | Users have logged in successfully and user information can be found in database |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

|  |  |
| --- | --- |
| NO. | Test case 2-4 |
| Functional module | User module |
| Case title | Test the input of invalid pictures while updating pictures |
| Prerequisite | Users have logged in successfully and user information can be found in database |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

|  |  |
| --- | --- |
| NO. | Test case 2-5 |
| Functional module | User module |
| Case title | Test the successfull update of user information |
| Prerequisite | Users have logged in successfully and user information can be found in database, and users have input valid new password or pictures while updating |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

Update user password:



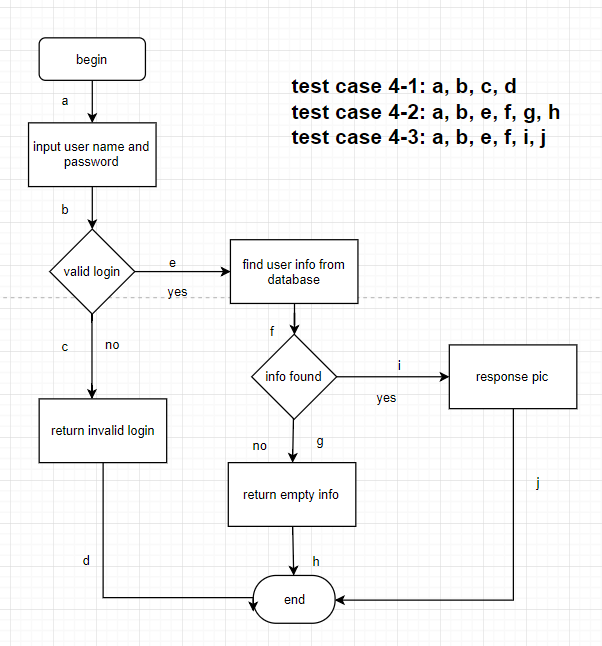
|  |  |
| --- | --- |
| NO. | Test case 3-1 |
| Functional module | User module |
| Case title | Test the input of invalid login |
| Prerequisite | Users input user name and password |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

|  |  |
| --- | --- |
| NO. | Test case 3-2 |
| Functional module | User module |
| Case title | Test user information which is not found in database |
| Prerequisite | Users have logged in successfully |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

|  |  |
| --- | --- |
| NO. | Test case 3-3 |
| Functional module | User module |
| Case title | Test the input of invalid password while updating user password |
| Prerequisite | Users have logged in successfully and user information can be found in database |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

|  |  |
| --- | --- |
| NO. | Test case 3-4 |
| Functional module | User module |
| Case title | Test the input of valid password while updating user password |
| Prerequisite | Users have logged in successfully and user information can be found in database |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

Get user pic:



|  |  |
| --- | --- |
| NO. | Test case 4-1 |
| Functional module | User module |
| Case title | Test the input of invalid login |
| Prerequisite | Users input user name and password |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

|  |  |
| --- | --- |
| NO. | Test case 4-2 |
| Functional module | User module |
| Case title | Test user information which is not found in database |
| Prerequisite | Users have logged in successfully |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

|  |  |
| --- | --- |
| NO. | Test case 4-3 |
| Functional module | User module |
| Case title | Test the response pictures with valid login and found information |
| Prerequisite | Users have logged in successfully and user information can be found in database |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

1. Testing framework introduction and reason

During our testing procedure, we choose pytest as our testing framework. The pytest framework makes it easy to write small tests, yet scales to support complex functional testing for applications and libraries. Due to pytest’s detailed assertion introspection, only plain assert statements are used.

There are many reasons why we choose pytest as our testing framework. First of all, our project is written in python, so we want to find a testing framework that fits python project better, where pytest is apparently an outstanding choice for us. It contains many features that meet our need. For example, it contains detailed info on failing assert  statements and there's no need for us to remember self.assert\* names, which largely reduce our workload and facilitate our efficiency. Besides, it has auto-discovery of test modules and functions. What's more, it has modular fixtures for managing small or parametrized long-lived test resources. Last but not least, it has rich plugin architecture, with a large scale of external plugins and thriving community, which help us design the test cases a lot.

Based on the discussion above, we choose pytest as our testing framework in this project and according to the result we have, it certainly simplify the process and complexity.