AITA

软件工程专业综合项目编程规范

1652714 孙浩然

1652763 陈泽徽

1652751 梁钧清

1553545 王嵩豪

1652708 周泽林

1652698 康晓博

# 目录

[AITA 1](#_Toc534129803)

[目录 2](#_Toc534129804)

[引言 3](#_Toc534129805)

[背景与依据 3](#_Toc534129806)

[选择模块 3](#_Toc534129807)

[测试技术和过程 4](#_Toc534129809)

[使用ECP技术对auth模块进行黑盒测试 4](#_Toc534129810)

[使用BVA技术对foucus模块进行黑盒测试 5](#_Toc534129811)

[对usr模块进行白盒测试 6](#_Toc534129811)

[测试框架 1](#_Toc534129809)5

# 引言

## 背景与依据

AITA 系统设计说明书

AITA需求分析文档

AITA 项目方案书

## 选择模块

本项目共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。该三个模块的使用频率在本项目中最高，所以我们选择对这三个模块进行单元测试。

用户认证模块是系统中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。我们决定对用户认证模块采用黑盒测试的 ECP 测试方法。

专注度模块是本系统的核心功能，这个模块在系统主要负责管理用户的专注度数据，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的 BVA 测试方法。

用户信息模块是本系统中使用频率最高的一个模块，这个模块主要负责用户信息的管理，我们决定对这个模块采用白盒测试。

# 测试技术和过程

4.1 使用 ECP 技术对 auth 模块进行黑盒测试

以下是ECP技术的等价类及其代表：

|  |  |  |
| --- | --- | --- |
| 参数 | 等价类 | 代表 |
| 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" |

以下是全部的测试情景：

|  |  |  |  |
| --- | --- | --- | --- |
|  | p参数 | |  |
| 测试情景 | username | password | 结果 |
| 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 使用 BVA 技术对 focus 模块进行黑盒测试

以下是测试类及其边界：

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

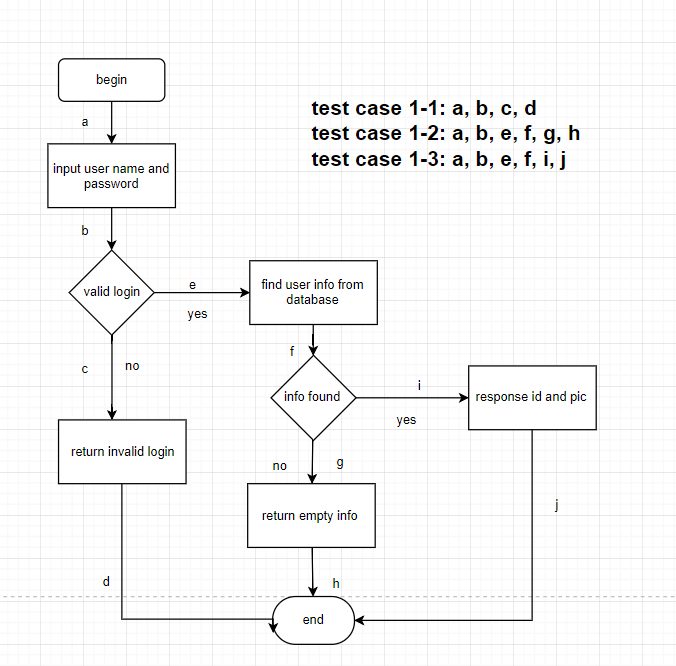
以下是全部的测试情景：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 参数 | | |  |
| 测试情景 | emotion | gaze | rate | 结果 |
| 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 对usr 模块进行白盒测试

我们队usr模块进行白盒测试，由于这个模块没有太多输入数据，我们没有很多测试情景。以下是测试的流程图：

**Get user info:**

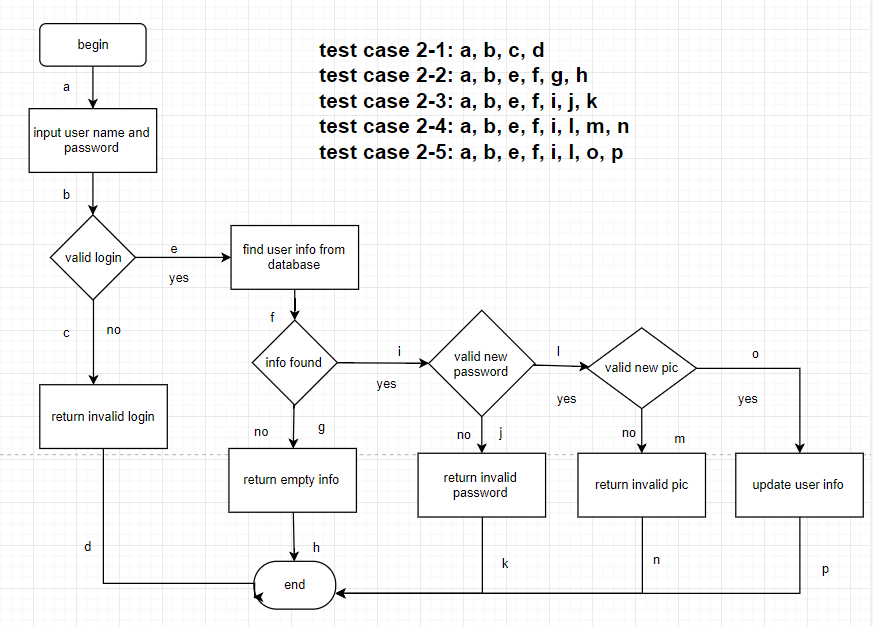


|  |  |
| --- | --- |
| NO. | Test case 1-1 |
| Functional module | User module |
| Case title | Test the input of invalid login |
| Prerequisite | Users input user name and password |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

|  |  |
| --- | --- |
| 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 |  |

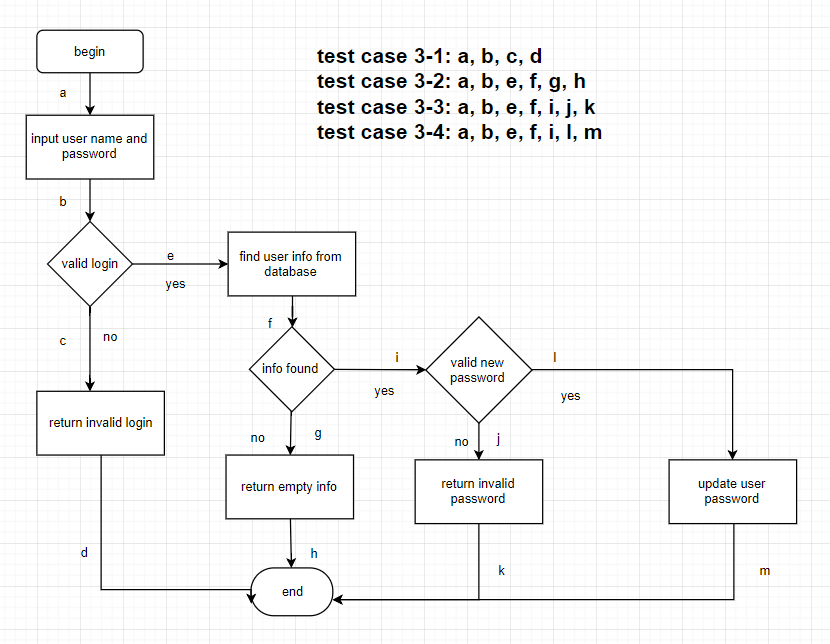
|  |  |
| --- | --- |
| NO. | Test case 2-2 |
| Functional module | User module |
| Case title | Test user information which is not found in database |
| Prerequisite | Users have logged in successfully |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

|  |  |
| --- | --- |
| 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:**



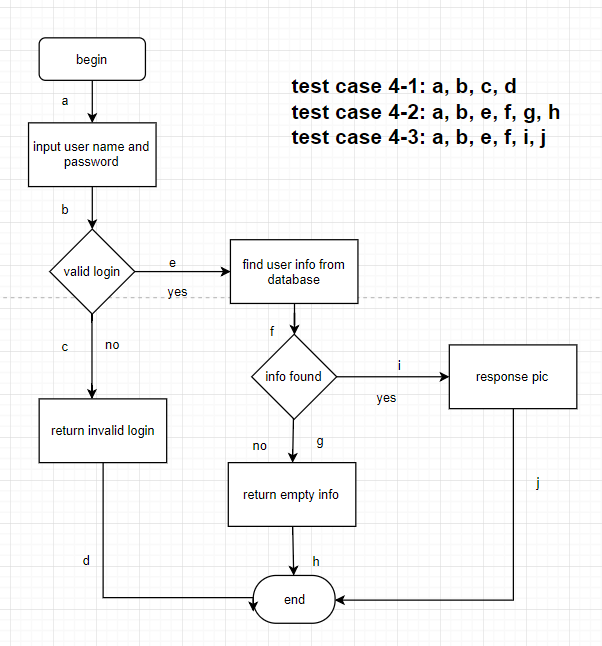
|  |  |
| --- | --- |
| NO. | Test case 3-1 |
| Functional module | User module |
| Case title | Test the input of invalid login |
| Prerequisite | Users input user name and password |
| Testing procedure |  |
| Expected results |  |
| Actually results |  |
| Remarks |  |

|  |  |
| --- | --- |
| 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 |  |

测试框架

在本系统的测试过程中，我们将选用 pytest 作为我们的测试框架，。

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.