Testing Reports

Revision History:

|  |  |  |
| --- | --- | --- |
| Date | Author | Description |
| 8-10-20 | Shizhe LIU | The first version of the test report was released |
| 15-10-20 | Kedong XIU | Some contents of the test report are modified |
| 22-20-20 | Shizhe LIU & Weikang DU | According to the requirements, the automatic test part is added |
| 23-20-20 | Kedong XIU | The test report is improved again |
| 6-11-20 | Shizhe LIU & Weikang DU | Update TR document |
| 13-11-20 | Shizhe LIU & Weikang DU | Update TR document |
|  |  |  |
|  |  |  |
|  |  |  |

Contents

[1. Introduction 3](#_Toc56202568)

[1.1. Intended Audience and Purpose 3](#_Toc56202569)

[1.2    How to use the document 3](#_Toc56202570)

[2. Principles of Software testing 3](#_Toc56202571)

[2.1. Principles of Testing as early as possible 3](#_Toc56202572)

[2.2. Principles of IPO 3](#_Toc56202573)

[2.3. Principles of Independent Testing 3](#_Toc56202574)

[3. Manual Testing 3](#_Toc56202575)

[3.1. Integration Testing 4](#_Toc56202576)

[4. Automatic Testing 4](#_Toc56202577)

[4.1. Robot Framework 5](#_Toc56202578)

[5. Testing Procedure 5](#_Toc56202579)

[6. Current Testing Results 6](#_Toc56202580)

[7. Future test plan and test goal 7](#_Toc56202581)

[8. Real integration testing 7](#_Toc56202582)

## Introduction

## Intended Audience and Purpose

This document provides the testing method and procedure, corresponding to the requirement from the customer.

## 1.2    How to use the document

You may refer to the content section for the structure of the document, in which Sec. Principles of Software testing contains the principles when you testing you need to obey; Sec. Manual Testing shows the steps and expected results of the integration test; Sec. Automatic Testing describes the aim of automatic testing and the categories of automatic testing framework. Sec. Test Procedure shows the test flow chart.

## Principles of Software testing

## Principles of Testing as early as possible

Software testing should be carried out as soon as possible. According to statistics, about 60% of the errors come from before design, and the cost of correcting a software error will increase with the development of software life cycle. The earlier a mistake is found, the less it will cost to fix it.

## Principles of IPO

The test case consists of two parts: the test input data and the corresponding expected output results.

## Principles of Independent Testing

Software testing is conducted by organizations that are economically and administratively independent of the development organization. Programmers should avoid checking their own programs, and programming agencies should not test their own programs.

## Manual Testing

In this section, there are four kinds of tests: Unit test, System test, Integration test and Acceptance test. In the process of testing, it is necessary to strictly check whether the relevant code has completed the specified function according to the requirements analysis (RA) document.

Generally, the test will first detect whether the use cases in the RA document have been completed. For example, the server needs to be able to respond to the client's related requests. Then, we need to test whether the server can actually handle these requests.

## Integration Testing

In the integration test, we need to check whether the docking with other modules is good. In fact, integration testing is a big part of the workload, because the server is responsible for interacting with the three parts. We need to check whether the interface between the server and other modules is good one by one. If the interface cannot be connected, we need to negotiate with other modules and redesign the interface for docking.

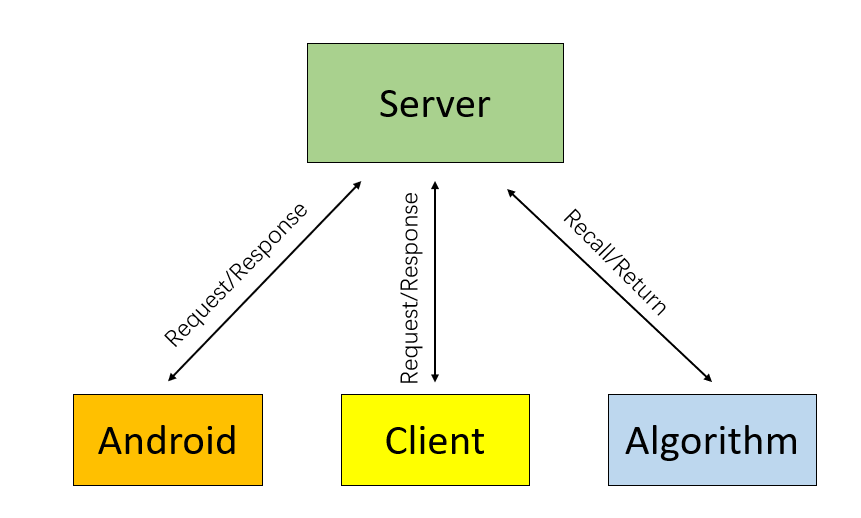


Fig 1: Module interaction

## Automatic Testing

Software automatic testing is to change the old traditional manual testing method into the new work of using machine software testing. Automatic testing runs the tested program under the present conditions, and analyzes the running results.

Automatic testing framework can be divided into five categories:

i) Test Script Modularity

ii) Test Library Architecture

iii) Data-driven Testing

iv) Keyword Driven Testing

v) Hybrid Test Automation

We plan to use Keyword Driven Testing as our Automatic testing framework, and choose robot framework software as our automated testing tool.

## Robot Framework

Robot Framework is a generic open source automation framework. It can be used for test automation and robotic process automation (RPA). Robot Framework has a modular architecture that can be extended with bundled and self-made [libraries](https://robotframework.org/#libraries).

Data is defined in files using the syntax shown in the [examples](https://robotframework.org/#examples) below. A file containing tests or tasks creates a suite, and placing these files into directories creates a nested structure of suites.

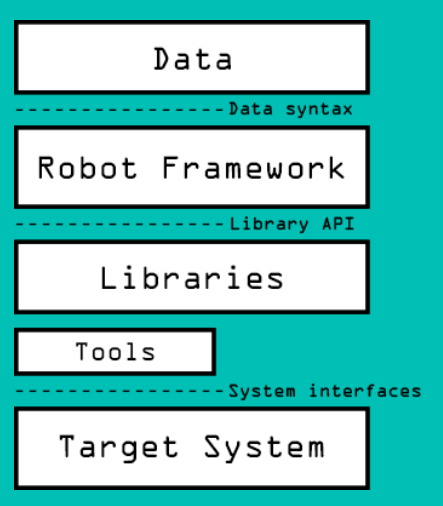


Fig 2: Architecture of Robot framework

If you want to learn more about the robot framework, visit the website: <https://robotframework.org/>.

## Testing Procedure

We plan to conduct manual testing first, and then use the existing automated testing framework and tools for automated testing.

When we finish the server-side code, what we need to do is unit test. Then, we need to do integration testing. However, due to the change of requirements, each part of the code needs to be modified again. Therefore, we have not done a complete integration test for the time being, only a part of the integration testing work has been done. At present (2020/11/6), we are discussing a time to do the integration test in a strict sense. The time is tentatively scheduled for November 9 in Ding Xin library.

The test flow chart is shown below.

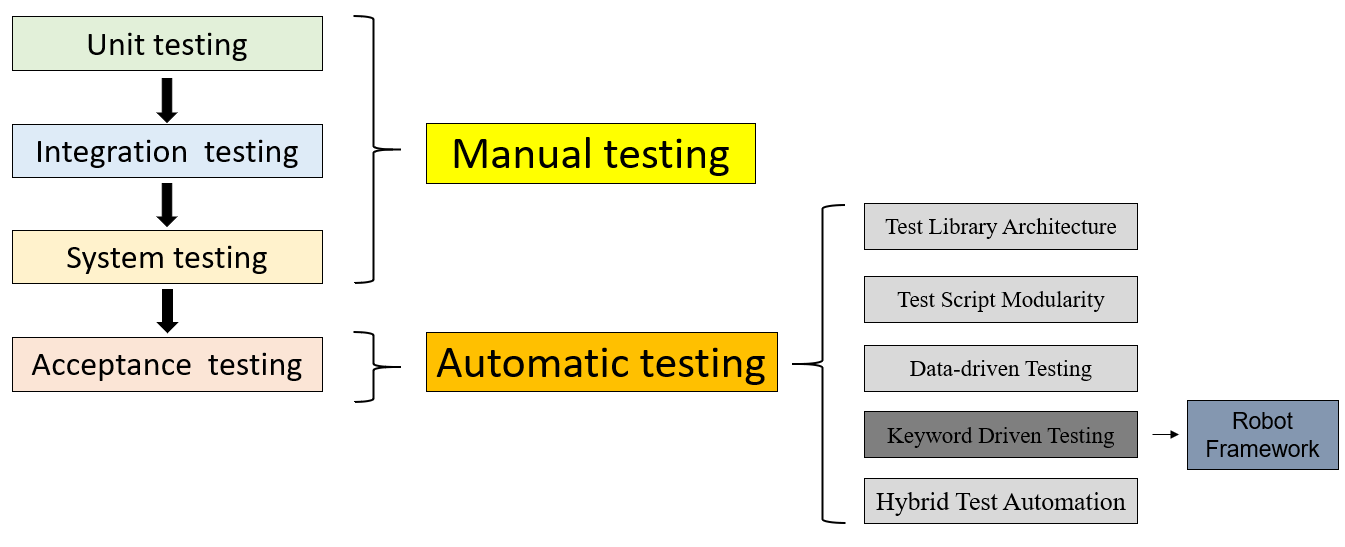


Fig 3: Testing procedure

## Current Testing Results

Because we didn't complete the whole software, we didn't do the acceptance test. At present, we have completed part of the work of integration test, and found that the server can connect with the client and algorithm side successfully.

It is worth mentioning that the coder and tester of our group are jointly held by two people. Considering the principle of software testing, they should test each other's code to ensure the effectiveness of the test.

The results of the current testing(unit test and partial integration test ) are listed here and you may find the correspondence to requirements in the RA document.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case NO.** | **Module** | **Result** | **Corresponding Requirement** |
| 1 | User Log in | Finished | Users can log in and enter the corresponding webpage |
| 2 | User Log out | Finished | Users log out, and go back to the log in page |
| 3 | User Sign up | Finished | The user can create a new account |
| 4 | Call Algorithm | Finished | It can successfully call the algorithm and get the running results |
| 5 | Store Results | Finished | Users can store relevant results |
| 6 | Query Results | Finished | Users can query relevant results |
| 7 | Handle Webpage request | Finished | Server can handle Webpage request |

## Future test plan and test goal

As mentioned earlier, we will conduct integration testing on November 9. In this integration test, we require testers from all groups to participate. We need to fully discuss whether the connection of each part works normally. If there is any problem in the integration, who should be responsible for it?

What we want to achieve is that all the parts are well integrated. In addition, we also need to do some performance tests, such as: how long does it take to process an image? How many visitors can the website accommodate at one time? What is the accuracy of the algorithm results?

During the integration test, I will complete the following table. Therefore, the entire document will be updated after November 9.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case NO.** | **Module** | **Result** | **Corresponding Requirement** |
| 1 | Server-Android | … | … |
| 2 | Server-Algorithm | … | … |
| 3 | Server-Web | … | … |
| 4 | Performance Test | … | … |
| … | … | … | … |

## Real integration testing

On November 9, we completed the integration test on schedule. The following is the result table of our integration test.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case NO.** | **Module** | **Result** | **Corresponding Requirement** |
| 1 | Server-Android | Finished | They can communicate with each other |
| 2 | Server-Algorithm | Some test results deviate from the reality seriously | The algorithm has high accuracy |
| 3 | Server-Web | Finished | They can communicate with each other |
| 4 | Performance Test | Finished | It can support no more than 10 concurrent accesses (Because the algorithm is time-consuming and the server performance is not good) |
| 5 | Upload the image | After the picture is uploaded to the server, a prompt box will pop up | Special test case (Abnormal test) |
| 6 | User upload the wrong image | Webpage can’t get the results in 30 seconds | Special test case (Abnormal test) |
| 7 | User logout the go back homepage | Webpage will display the last login name of the user | Special test case (Abnormal test) |
| 8 | Multiple users upload images and want to see the results | It takes longer to get results | Special test case (Abnormal test) |
| 9 | Database crashes | Algorithm won’t execute | Special test case (Abnormal test) |
| 10 | Algorithm crashes | Algorithm does not return results within the specified time; the algorithm will be forced to stop running | Special test case (Abnormal test) |

In this test table, multiple related test cases are replaced by related modules to reduce the number of test cases. For example, test case 1 is the overall test of the server and Android module. We have tried it many times to ensure that there is no problem with the interaction between them.

In addition, we conducted special test cases (many unexpected operations or events occurred). The whole integration test shows that our system can work normally. The only drawback is that the accuracy of the algorithm is not high enough.