Sprint 2 Testing plan

During Sprint 2, the testing team would firstly conduct the function test for all the left requirements. Then, performance tests would be done to ensure that our system could meet requirements of performance made by the clients. Lastly, the system test that contains all the test cases would be operated, followed by the user acceptance test executed by our client representatives.

Steps the team would take to test functional and non-functional requirements of Sprint 2

The flow chart below shows a general overview of testing processes we would operate during sprint 2:

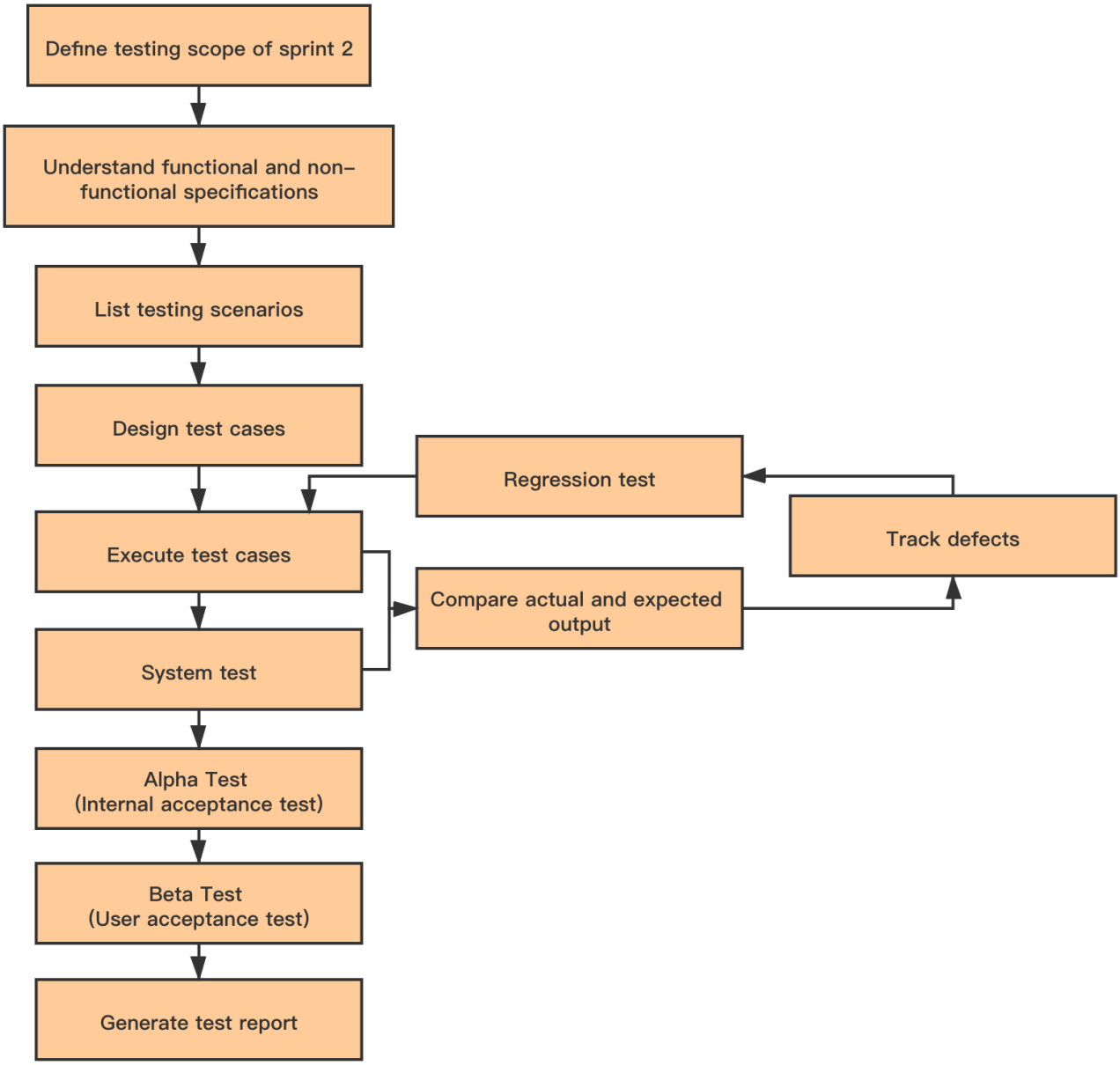

Figure 1: Sprint 2 testing processes

It is important to mention that, when the team starts to design or execute test cases, the team should consider not only functional requirements, but also other non-functional expectations put forward by the clients. For example, performance of the server, UI layout of the website, compatibility and usability should also be taken into consideration. In the following paragraphs, detailed description of each step of the sprint 2 testing plan would be shown.

1.Define the testing scope

In this step, the testing team makes a list of user stories included in sprint 2 that are going to be examined. Below is the list of all the requirements we would test in sprint 2:

- **Functional requirements:**

User story ID	User story Name
1	Forget password
5	Delete files / pictures
6	Copy and paste URL
7	Send email to another user
8	Download a user's profile in PDF format

- **Non-functional requirements:**

Requirement type	Requirement name
Performance requirement	1000 threads concurrently simulate U1 - 'forget password'
Performance requirement	1000 threads concurrently simulate U2 - 'Google login' and 'Facebook login
Performance requirement	1000 threads concurrently simulate U3 - 'hide and show six sections'
Performance requirement	1000 threads concurrently simulate U4 - 'resetting the user account'
Performance requirement	1000 threads concurrently simulate U5 - 'delete texts, files or pictures'
Performance requirement	1000 threads concurrently simulate U6 - 'copy and paste URL'
Performance requirement	1000 threads concurrently simulate U7 - 'send email to another user'
Performance requirement	1000 threads concurrently simulate U8 - 'download user's profile in PDF format'
UI requirement	layout of the home page
UI requirement	layout of the 'register' page
UI requirement	layout of the 'login' page
UI requirement	layout of the 'reset' page
UI requirement	layout of the 'profile' page
UI requirement	layout of the 'public view' page
Compatibility requirement	Mac OS X
Compatibility requirement	Windows 10
Compatibility requirement	Google Chrome
Compatibility requirement	Safari
Compatibility requirement	Firefox
Usability requirement	usability of basic functions: register, login; upload, delete and download files.
Usability requirement	usability of advanced functions: view others' profile, contact another user, copy and paste URL.

2.List testing scenarios

In this section, we list down possible test scenarios for all required functions. Test scenarios are helpful when designing test cases, since they offer a basic idea of what would be examined when testing a specific requirement.

3.Design test cases

This time, the team would begin to design use cases. Use cases should be created strictly follow the table below:

Code	Module	Name	Designer	Maintainer	Create time	Type	Status	Pre-condition	Steps	Expected outcome	Importance level

ID of the use case.	Functional module that the use case belongs to.	Name of the use case.	Name of the designer.	Name of the maintainer.	Time when the use case is created.	Type of the test case.	Current status of the use case.	Pre-condition needed to fulfill before executing the test case.	Steps need to take when executing the use case.	Expected output after running the test case.	Level of importance of the test case.

4.Execute test cases

In this step, the team would start running these test cases above. These test cases would firstly be executed after completion of each function. After the full achievements of all requirements in sprint 1, system test and regression test would be conducted, during which the team would go through all these test cases again to see whether functions the team completed in sprint 1 would meet requirements of clients.

The following table indicates different status of test cases:

Status of a test case	**description**
Passed (P)	Actual results exactly match the expectations.
Failed (F)	This may contain some sub-conditions. On one hand, actual results do not match initial expectations; On the other hand, actual results match the expected outputs but cause other problems. Defects need to be logged and tracked.
Unprocessed (UP)	Test cases that have not been executed yet. This is the default status.
In Progress (IP)	Test cases have been put into process, but not all steps have been finished.
Investigating (I)	Test cases have been executed and corresponding outputs are generated. However, it is hard to decide at the first glance whether these cases are passed or not. Needs additional discussion.
Blocked (B)	Test cases cannot be executed because of blocking issues.

It is the comparison of expected results to actual results that will determine whether the test has a ‘Pass’ or ‘Fail’ status.

5.System test and regression test

This time, all the functional modules have been integrated and the whole system has been built successfully. In this step, the team would start with a smoke test to see if the software is runnable. Afterwards, functional testing cases would be executed one by one again. However, test cases conducted during the system test are different to those operated in function tests, not only because in the system test we focus more on the overall system, but also due to that we are meant to find 'problems' existing in our product. That is to say, during function tests, the team wants to guarantee that all the basic functional modules are working well; While during system test, the major goal is to find any existing defects, logging them into requirement traceability matrix (RTM) and tracking them. Then, a regression test would be done with the help of RTM. In regression tests, test cases are re-executed so as to check whether the previous defects is fixed and the new changes have not brought any new bugs. All the team members would try their best fixing the defects and optimize the product. If no severe defects exist in the system, the testing team would move to the next step.

6.Internal acceptance test (Alpha test)

If the system test is passed without finding any severe defects, alpha test would be put into process. In the alpha test, the team would go through the sprint 1 test cases again to see if all functions are met at a high level of quality. If there are no severe functional defects left the team would move to the next step, beta tests operated by client representatives.

7.User acceptance test (Beta test)

Beta test is an important step in our testing plan. Once passed the user acceptance test, our product could be regarded as quality guaranteed and could be released and put into actual use. In this step, the team would invite client representatives to a user acceptance test meeting to see if they have any suggestions on the functional and non-functional modules of our project.