DocNo: 001.H.1.1

**Grape**

**Test Plan**

**Version 2.0**

**By**:

Group Undefined

2015-05

**Group Member**:

Hunter Lin

Birdy

Listen

Morning

Syachi

**Document Language**:

English

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 2015.5.24 | 1.0 | Initialization of the report | Hunter Lin |
| Final Date | 2.0 | Integrating all of the works | Hunter Lin |

**Key Word**

**Abstract**

Content

[1. Introduction 6](#_Toc420232730)

[1.1. Purpose 6](#_Toc420232731)

[1.2. Background 6](#_Toc420232732)

[1.3. Definition 6](#_Toc420232733)

[1.4. Reference 6](#_Toc420232734)

[2. Test Plan 6](#_Toc420232735)

[2.1. Project Review 6](#_Toc420232736)

[2.2. Test Cases 6](#_Toc420232737)

[2.3. Unit Test 6](#_Toc420232738)

[2.3.1. Test Schedule 6](#_Toc420232739)

[2.3.2. Conditions 6](#_Toc420232740)

[2.3.3. Test References 6](#_Toc420232741)

[2.3.4. Test Training 6](#_Toc420232742)

[2.4. Integration Test 6](#_Toc420232743)

[2.4.1. Test Schedule 6](#_Toc420232744)

[2.4.2. Conditions 6](#_Toc420232745)

[2.4.3. Test References 6](#_Toc420232746)

[2.4.4. Test Training 7](#_Toc420232747)

[2.5. System Functional Test 7](#_Toc420232748)

[2.5.1. Test Schedule 7](#_Toc420232749)

[2.5.2. Conditions 7](#_Toc420232750)

[2.5.3. Test References 7](#_Toc420232751)

[2.5.4. Test Training 7](#_Toc420232752)

[2.6. Runtime Test 7](#_Toc420232753)

[2.6.1. Test Schedule 7](#_Toc420232754)

[2.6.2. Conditions 7](#_Toc420232755)

[2.6.3. Test References 7](#_Toc420232756)

[2.6.4. Test Training 7](#_Toc420232757)

[2.7. Stress Test 7](#_Toc420232758)

[2.7.1. Test Schedule 7](#_Toc420232759)

[2.7.2. Conditions 7](#_Toc420232760)

[2.7.3. Test References 7](#_Toc420232761)

[2.7.4. Test Training 7](#_Toc420232762)

[3. Test Design Specification 8](#_Toc420232763)

[3.1. Unit Test 8](#_Toc420232764)

[3.1.1. Control Method 8](#_Toc420232765)

[3.1.2. Test Case 8](#_Toc420232766)

[3.1.3. Process 8](#_Toc420232767)

[3.2. Integration Test 8](#_Toc420232768)

[3.2.1. Control Method 8](#_Toc420232769)

[3.2.2. Test Case 8](#_Toc420232770)

[3.2.3. Process 8](#_Toc420232771)

[3.3. System Functional Test 8](#_Toc420232772)

[3.3.1. Control Method 8](#_Toc420232773)

[3.3.2. Test Case 8](#_Toc420232774)

[3.3.3. Process 8](#_Toc420232775)

[3.4.Runtime Test 8](#_Toc420232776)

[3.4.1. Control Method 9](#_Toc420232777)

[3.4.2. Test Case 9](#_Toc420232778)

[3.4.3. Process 9](#_Toc420232779)

[3.5. Stress Test 9](#_Toc420232780)

[3.5.1. Control Method 9](#_Toc420232781)

[3.5.2. Test Case 9](#_Toc420232782)

[3.5.3. Process 9](#_Toc420232783)

[4. Criteria 9](#_Toc420232784)

[4.1. Scope 9](#_Toc420232785)

[4.1.1. Deflect verified rate criteria 9](#_Toc420232786)

[4.1.2. Coverage Rate Criteria 9](#_Toc420232787)

[4.2. Data Catalog 9](#_Toc420232788)

[4.3. Scale 9](#_Toc420232789)

[4.3.1. Test Ceasing Criteria 9](#_Toc420232790)

[4.3.2. Unit Test Ceasing Criteria 9](#_Toc420232791)

[4.3.3. Integration Test Ceasing Criteria 9](#_Toc420232792)

[4.3.4. System Test Ceasing Criteria 10](#_Toc420232793)

[5. Conclusion 10](#_Toc420232794)

Note:

黑色字部分为大家都需要写的。（大部分为功能测试的内容）

根据大家之前画use case时的分工，来写相应的system functional test.

黄色部分: Hunter Lin

蓝色部分: Morning

绿色部分: Birdy

红色部分: Listen

紫色部分: Syachi

**1. Introduction**

## 1.1. Purpose

## 1.2. Background

## 1.3. Definition

## 1.4. Reference

**2. Test Plan**

## 2.1. Project Review

|  |  |  |
| --- | --- | --- |
| Function | Input | Output |
| Create group | groupName,topic,confirmMessage | a corresponding group in the database |
| delete group | group\_id | a group deleted in the database |
| search group | group\_id | the information of the group |
| join group | group\_id | an association between the group and the current user is created in the database; |
| quit group | group\_id | an association between the group and the current user is deleted in the database; |
| Create a voting | group\_id, voting topic, voting content, deadline | a corresponding voting in the database |
| Attend a voting | group\_id, user\_id, voting\_id | get data |
| Watch a voting result | group\_id, user\_id, voting\_id | the result of the voting |
| Create a discussion | group\_id, user\_id, discussion content | a corresponding discussion in the database |
| Reply to a discussion | group\_id, user\_id, discussion\_id, reply content | a corresponding reply in discussion part in the database |
| Delete a voting | group\_id, user\_id, discussion\_id | a voting deleted in the database |
| Delete a discussion | group\_id, user\_id, discussion\_id | a discussion deleted in the database |

## 2.2. Test Cases

## 2.3. Unit Test

All members of team Grape will participate in this test. Because of functional test, almost all parts of the software will be tested.

### 2.3.1. Test Schedule

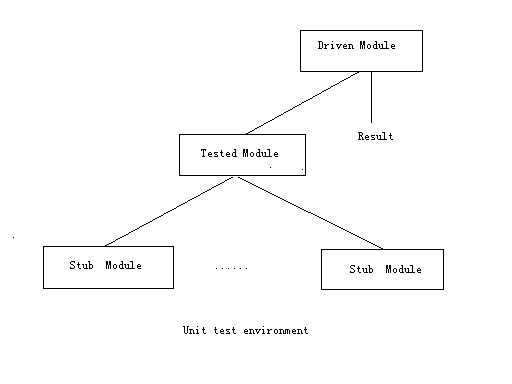
This test dates from 7.24 to 8.3.The work is to write stub module, driven module, test script, design test cases, and have tests.

### 2.3.2. Conditions

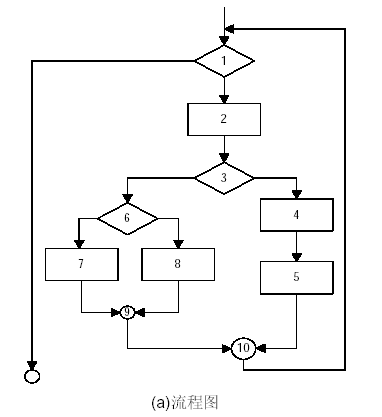
1. 5 computers, 5 days to be used.
2. All members of the twister team will participate in this test. We must know how to do black box testing and white box testing

### 2.3.3. Test References

1. Grape requirement document
2. All program units such as front controller, group management delegate, voting operations, discussion operations and so on.
3. Unit test environment

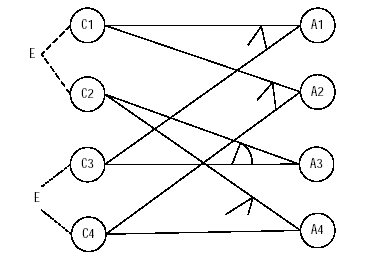


1. When using white box testing, testers must translate program procedure into workflow graph:



**Work Flow Graph**

1. When testing on query module and statistics module , causation graph can be used:



**Causation Graph**

### 2.3.4. Test Training

NULL

## 2.4. Integration Test

All members in our team will participate in this test. Several units will be integrated as a part and be tested respectively.

### 2.4.1. Test Schedule

This test will be executed from 5.29 to 6.3.The work is to write stub module, driven module, test script, design test cases, and do the tests.

### 2.4.2. Conditions

1. 5 computers,5 days to be used.
2. All members of the our team will participate in this test. We must know how to do black box test.

### 2.4.3. Test References

1. Grape requirement document.
2. The whole code including (XXXX要填充).
3. Integration test environment and the causation graph are the same with the unit test.

### 2.4.4. Test Training

Null

## 2.5. System Functional Test

### 2.5.1. Test Schedule

### 2.5.2. Conditions

### 2.5.3. Test References

### 2.5.4. Test Training

## 2.6. Runtime Test

### 2.6.1. Test Schedule

### 2.6.2. Conditions

### 2.6.3. Test References

### 2.6.4. Test Training

## 2.7. Stress Test

### 2.7.1. Test Schedule

### 2.7.2. Conditions

### 2.7.3. Test References

### 2.7.4. Test Training

**3. Test Design Specification**

## 3.1. Unit Test

The smallest test unit is class. Use white box testing to test inside instruction of unit, and black box testing to test function and action of test object. Unit testing goes with developing. Testers must write proper stub module, driven module, and test script.

### 3.1.1. Control Method

Each unit is tested manually by tester. Because in our developing process the developer is just the tester, they can fix bugs right now once they find a bug.

### 3.1.2. Test Case

|  |  |  |
| --- | --- | --- |
| Test case number | Input | Output |
| 1 | Sign up with proper mail, username and password. | Sign up successfully. |
| 2 | Sign up with wrong mail address. | Can’t sign up and remind that the mail address is wrong. |
| 3 | Sign up with wrong username. | Can’t sign up and remind that the username has been used. |
| 4 | Sign up with different password and confirming password. | Can’t sign up and remind that the password and confirming password are different. |
| 5 | Log in with proper username and pass word. | Log in successfully. |
| 6 | Log in with wrong username. | Can’t log in and remind that the username is wrong. |
| 7 | Log in with wrong password. | Can’t log in and remind that the password is wrong |
| 8 | Create new group with proper group name, topic, description and confirm message. | Create a new group successfully and other users can have access to the information of the group. The creator is appointed as the leader. |
| 9 | Search a group with the group id. | If the group id exists, you will find the group information. Otherwise, you will get nothing. |
| 10 | Attend the group with confirm message. | If the confirm message is right, you will attend the group successfully. Otherwise, you will fail to attend. |
| 11 | Leader creates a vote. | Voting will be published onto the Voting Board. The members in the group have access to the vote. |
| 12 | Members attend the vote. | The system will receive the members’ votes and make a statistic after the voting ends. |
| 13 | Member in the group generates a question. | Question will be published onto the Discussion Board. The members in the group have the access to the question and can reply to it. |
| 14 | Member in the group replies to the question. | The reply will be published onto the Question sub interface in Discussion Board. The questioner will receive message and members in the group have access to the reply. |
| 15 | Admin delete user or group. | The user account and group id will be invalid. |

### 3.1.3. Process

1. Design test cases.
2. Write stub module, driven module and test script.
3. Execute code, and compare result with expected.
4. Fix bugs found, and continue testing till there are no bugs.
5. When no bug is found, the test is over.

## 3.2. Integration Test

According to system business tier , present tier ,and subsystem , integrate related units to test the integration version. Use black box testing to check the function and action of integration version. The whole process employs bottom – top integration. Testers must write proper stub module , driven module, and test script.

### 3.2.1. Control Method

Every integration component is tested manually by testers. Since in our developing process the developers are just the testers, they can fix bugs right now once they find a bug.

### 3.2.2. Test Case

|  |  |  |
| --- | --- | --- |
| Test case number | Input | Output |
| 1 | Operations about group in the webpage | Corresponding respond in the front-end and the database |
| 2 | Operations about discussion in the webpage | Corresponding respond in the front-end and the database |
| 3 | Operations about vote in the webpage | Corresponding respond in the front-end and the database |
| 4 | Operations about user himself in the webpage | Corresponding respond in the front-end and the database |

### 3.2.3. Process

1. Design test cases.
2. Write stub module, driven module and test script. Create a database for test.
3. Run server, Execute code, and compare result with expected.
4. Fix bugs found, and continue testing till there are no bugs.
5. When no bug is found, the test is over.

## 3.3. System Functional Test(Group Function)

This part is tested by morning. The goal is to test the functions concerning group including join,quit,create,delete and so on.

### 3.3.1. Control Method

Manual operations on the web page. The testing result will be recorded in Excel.

### 3.3.2. Test Case

|  |  |  |
| --- | --- | --- |
| Function | Input | Output |
| Create group | groupName,topic,confirmMessage | a corresponding group in the database |
| Create group | groupName,topic,confirmMessage same as previous one | report that the group already exists |
| delete group | correct group\_id | a group deleted in the database |
| delete group | wrong group\_id | report fail to delete group due to authority or other errors |
| search group | correct group\_id | the information of the group |
| search group | wrong group\_id | return no information found |
| join group | correct group\_id | an association between the group and the current user is created in the database; |
| join group | wrong group\_id | report fail to join group |
| quit group | group\_id | an association between the group and the current user is deleted in the database; |

### 3.3.3. Process

1. Generate some pre-defined information about users and groups in the database.
2. Design test cases.
3. Run web server and database server.
4. Manually execute the operations about group at the front-end, and record bugs found.
5. After all the test cases reach system testing ceasing criteria, this test is over.

**3.4. Runtime Test**

### 3.4.1. Control Method

### 3.4.2. Test Case

### 3.4.3. Process

## 3.5. Stress Test

### 3.5.1. Control Method

### 3.5.2. Test Case

### 3.5.3. Process

**4. Criteria**

## 4.1. Scope

### 4.1.1. Deflect verified rate criteria

### 4.1.2. Coverage Rate Criteria

## 4.2. Data Catalog

## 4.3. Scale

### 4.3.1. Test Ceasing Criteria

### 4.3.2. Unit Test Ceasing Criteria

1. Unit test cases have accessed.
2. According to unit test cases, testers have finished all the tests of units.
3. Reach the coverage rate criteria of unit testing.
4. Make sure that more than 3 errors should be found every KLOC of units.
5. Unit function must be consistent with design model.
6. All the defects have been verified, and the verified rate has reached the criteria.

### 4.3.3. Integration Test Ceasing Criteria

1. Integration test cases have accessed.
2. According to integration test cases, testers have finished all the tests of integration.
3. Reach the coverage rate criteria of integration testing.
4. Make sure that more than 2 errors should be found every KLOC of integration versions.
5. Integration version function and capability must be consistent with definition.
6. All the defects have been verified, and the verified rate has reached the criteria.

### 4.3.4. System Test Ceasing Criteria

**5. Conclusion**