DocNo: 001.I.1.1

Grape

Test Analysis Report

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
2015.5.24	1.1	Finish test unit on integration test and test cases predefined by Morning	Morning
2015.5.24	1.2	Finish the document on unit test and the test cases predefined by Birdy	Birdy
2015.5.24	1.3	Finish the document on stress test and the test cases predefined by Listen	Listen
2015.5.24	1.4	Finish the document on system functional test and runtime test and the test cases predefined by Syachi	Syachi Cui
Final Date	2.0	Integrating all of the works	Hunter Lin

Key Word

Grape, Defect
Black box testing, White box testing
Stub module, Driven module
Boundary testing
Unit test, Integration test, System test, Run time test, Stress test

Abstract

This document describes in detail the testing methodology and different test cases. It is of great important because of the high cost to pay if the hidden bug is found in the released version. So we need to take much attention on the designation and test cases of our software. The main contents include unit test, integration test, system test and stress test.

Content

1. Introduction

1.1. Purpose

This document is our test analysis plan for the Grape System, which illustrates the details for the test context, test scope, test standard, and so on. This document will be the display some main results of our testing. This document can also be our reference for bug tracking and testing logs which can facilitate our further debugging.

1.2. Background

The system tested is named as "Grape", which is developed by the Undefined Group (members are: Hunter Lin, Morning, Syachi, Listen, Birdy).

The Grape can be widely used in classroom and discussion room, the users can share their opinions and resources using this software as the communication platform. Also, the leader of the group can promulgate or share some important messages in the certain group.

The whole project began at April 5th. After requirement analysis, system designing, and coding, the next step is testing. The testing goes along in the computer center of Shanghai Jiaotong University. After coding out the system and our testers master the testing knowledge and skills, we can do our test.

1.3. Definition

Grape: A interactive software for resource sharing created by the Undefined group.

Defect: Software bug

Black box testing: A test method, which testers only pay attention to input and output.

White box testing: A test method, which testers must know the inside instruction of test object. Including branch testing, statement testing, path testing and so on.

Stub module: When taking unit testing and integration testing, the test object needs to call other unit, and then stub module can take instead of the called unit. It can be viewed as a **Proxy** pattern in the design pattern

Driven module: When taking unit testing and integration testing, the test object needs to make active by others, then driven module can take instead of the caller. It can also be viewed as a Proxy design pattern.

Test script: A small teat program for testing to call unit or be called by unit.

Equivalence partition: A test method in black box testing. It uses a set of values selected, instead of many input value, which are dealt with in the same way.

Boundary designing: It is the extension of the equivalence partition; usually it is the boundary of equivalent class.

Causation graph: When considering the relationship of each input, causation graph can show the combinations of all inputs and outputs.

Unit testing: Test on the smallest unit such as class in the software.

Integration testing: Test on the combination of several units to check if they can work together.

Regression testing: In integration testing, some integration test cases must be test again to check if they can work with other integrations.

System testing: Compared with requirement definition, look for some parts which are not coincident with the requirement.

Run time testing: Test if the request-response time reaches criteria.

Stress testing: Test if the system can afford heavy using stress.

WAS: Web Application Stress Tool, a testing tool for stress testing.

1.4. Reference

"Software Testing"

by Ron Patton

"Object-Oriented Software Engineering – Using UML, Patterns and Java"

by Allen H. Dutoit

2. Test Overview

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 bulletin	user_id, group_id, bulletin content	an association between the leader and the bulletin is created. an association between the group and the bulletin is created. Also note that, if the user is not the leader in the group, the creation should be denied.
Delete bulletin	user_id, group_id, bulletin_id	Associations between the leader and the bulletin, group

		and bulletin should both be deleted.
Create message	Type, time, content, receiver, generator	A message is generated for certain receivers, which can be seen once they log into our system.
Mark message as read	Message_id	Indicating that the user has already known the message. And the message will be removed from the interface.
Create news	Type, time content, receiver, generator	A news is generated for certain receivers in a certain group, which can be seen once they get into the group interface.
Create vote	vote_content,vote_options, vote_timelimit	A corresponding vote in the database
delete vote	vote_id	Delete all the corresponding information in the database.
Operate vote	Option,vote_id	Update the votes of the corresponding option by add 1 in the database and insert the record what the option the user votes.
View vote voted	vote_id	Show the vote and the option voted
display vote result	vote_id	One bar graph displaying the distribution of the votes over different options in the database.
Finish vote	vote_timelimit	The database automatically set the vote status to 0 which means the end by using the event of MySQL.
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 discussion	group_id,user_id,discussion_id	a discussion deleted in the database

3. Test Result & Findings

3.1. Unit Test

3.1.1. Test Case

Test case number	Input	Output
1	Sign up with proper mail, username and password.	Sign up successfully. And a message will be sent to the user.
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 confirmed 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. And a message will be sent to the group members.
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. And a message will be sent to the group members.
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. And a message will be sent to the group members.
15	Admin delete user or group.	The user account and group id will be invalid.

3.1.2. Process

a) Design test cases.
b) Write stub module, driven module and test script.
c) Execute code, and compare result with expected.
d) Fix bugs found, and continue testing till there are no bugs.
e) When no bug is found, the test is over.

3.1.3. Compare with Test Plan

In general, everything in the test performs as expected and our system works well. But still there are some little bugs like invalid input and some boundary conditions. These kinds of bugs are relatively difficult to find since we can't test every case of input .We will concentrate on fixing that in the next step.

3.2. Integration Test

3.2.1. 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

3.2.2. Process

a) Design test cases.
b) Write stub module, driven module and test script. Create a database for test.
c) Run server, Execute code, and compare result with expected.
d) Fix bugs found, and continue testing till there are no bugs.
e) When no bug is found, the test is over.

3.2.3. Compare with Test Plan

In general, everything in the test performs as expected and our system works well. But still there are some little bugs like invalid input and some boundary conditions. These kinds of bugs are relatively difficult to find since we can't test every case of input .We will concentrate on fixing that in the next step.

3.3. System Functional Test

Please refer to part 4 (Function Test Report) for details.

3.4. Runtime Test

3.4.1. Test Case

Every model must choose the test case whose run time is longest,and if this is less than 20 second, the model passes the criteria.

3.4.2. Process

a) Set up a complete database.
b) Design test cases.
c) Have tests, and record bugs.
d) After the test reaches run time criteria, test is over.

3.4.3. Compare with Test Plan

In general, everything in the test performs as expected and our system works well. But still there are some little bugs like invalid input and some boundary conditions. These kinds of bugs are relatively difficult to find since we can't test every case of input .We will concentrate on fixing that in the next step.

3.5. Stress Test

Use LOCUST which is deemed to be awesome by the author of Flask, Jinja2 to test.

3.5.1. Test Case

Test case number	Input	Output
1	Simulate this case that at a time 1000 users visit the server, and distribute the flux in different page groups.	LOCUST provide us with the ReponseContextManager class to see the request result and whether it was successful.
2	Throttle bandwidth to test the capability when user takes dial-up or other connection to surf on Internet.	

3.5.2. Process

a) Set up a complete database
b) Design test cases
c) Run server
d) Have tests by LOCUST, and record defects
e) After the test reaches stress criteria, test is over.

3.5.3. Compare with Test Plan

4. Function Test Report

4.1. Login

4.1.1. Test Case

Test Case	Input	Output
Login	wrong user ID	Show that your user id is wrong.
Login	wrong password	Show that your pw is wrong
Login	NULL user ID	Show that user ID cannot be

		NULL
Login	NULL password	Show that password cannot be NULL
Login	correct user ID and pw	Show user index
Login	correct admin ID and pw	Show admin index

4.1.2. Limitation

4.2. Group Operation

4.2.1. 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;

4.2.2. Limitation

4.3. Bulletin Operation

4.3.1. Test Case

Test Case Number	Input	Output
Create a bulletin	User_id, Group_id (the user is the group leader)	A new item in bulletin table is inserted; An association between the bulletin and the group is created.
Create a bulletin	User_id, Group_id (the user is only a group member)	No change in databases. Report "No authority" in the front end.
Create a bulletin	User_id, Group_id (the user is not a member in the group.)	No change in databases. Report "No authority" in the front end.
Delete a bulletin	User_id, Group_id, Bulletin_id (the user is the group leader and the bulletin is created by the leader himself)	The corresponding item is deleted from the bulletin table. The corresponding association between the group and the bulletin is deleted.
Delete a bulletin	User_id, Group_id, Bulletin_id (the user is the group leader but the bulletin is not created by the leader himself)	No change in databases. Report "No authority" in the front end.
Delete a bulletin	User_id, Group_id, Bulletin_id (the user is the group leader but the bulletin is not created by the leader himself)	No change in databases. Report "No authority" in the front end.
Delete a bulletin	User_id, Group_id, Bulletin_id (the user is the group leader but the bulletin is not created by the leader himself)	No change in databases. Report "No authority" in the front end.

4.3.2. Limitation

4.4. Vote Operation

4.4.1. Test Case

Test Case Number	Input	Expected Output	Real Output
create vote	vote_content,vote_options,vote_timelimit	A corresponding vote in the database and the webpage	A corresponding vote in the database and the webpage
create vote	timelimit not set or empty vote_options and vote_content	Ban user to submit	user can still create a vote but the web crashes then.
delete vote	vote_id	Delete all the corresponding information in the database.	Delete all the corresponding information in the database.
delete vote	wrong vote_id	report fail to delete group due to authority or other errors	report fail to delete group due to authority or other errors
operate vote	option,vote_id	Update the votes of the corresponding option by adding 1 in the database and insert the record which option the user voted.	Update the votes of the corresponding option by adding 1 in the database and insert the record which option the user voted.
operate vote	option,wrongvote_id	report fail to vote due to authority or other errors	report fail to vote due to authority or other errors
View vote voted	vote_id	Show the vote and the option voted	Show the vote and the option voted
view vote voted	wrong vote_id	Report fail to view the option voted due to authority or other errors	Report fail to view the option voted dueto authority or other errors
display vote result	vote_id	One bar graph displaying the distribution of the votes over different options in the database.	One bar graph displaying the distribution of the votes over different options in the database.
finish vote	vote_timelimit	The database automatically set the vote status to 0 which means the end by using the event of MySQL.	The database automatically set the vote status to 0 which means the end by using the event of MySQL.

4.4.2. Limitation

We detect the input to make sure that it is not null and if so, we will alert the user to input again. We use a timer rather than let user input the time himself. On the one hand, it makes it easier to input. On the other hand, it avoids some invalid input. Once a user cast a vote, we will remember this state to avert multiple votes.

4.5. Message Operation

4.5.1. Test Case

Function	Input	Output
Sign up (Successfully)	username, email, password	A welcome message will be sent to our user.
Create new bulletin in group	Title, content	A new bulletin message will be sent to the group members except the genereator.
Create new discussion in group	Title, content	A new discussion message will be sent to the group members except the genereator.
Create new vote in group	Title, content	A new vote message will be sent to the group members except the genereator.
A user is removed from a group	User_id	A informative message will be sent to the rest of the group member. (A short message)
The leader deleted a group	Group_id	A informative message will be sent to the formal members of the gruop
The leader deleted a discussion	Discuss_id	A informative message will be sent to the creator of the discussion
The leader deleted a reply	reply_id	A informative message will be sent to the creator of the reply.
A group member quitted the group	Group_id, user_id	A informative message will be sent to the leader of the group.

4.5.2. Limitation

The limitation is that, if the system is large enough, a great number of messages will be generated, thus causing the speed of retrieving items to be slower.

4.6. News Operation

4.6.1. Test Case

Function	Input	Output
Create new bulletin in group	Title, content	A bulletin news will be sent to the group members except the genereator.
Create new discussion in group	Title, content	A discussion news will be sent to the group members except the genereator.
Create new vote in group	Title, content	A vote news will be sent to the group members except the genereator.
A user is removed from a group	User_id	A informative news will be sent to the user being removed.
A group member quitted the group	Group_id, user_id	A informative news will be sent to the leader of the group.

4.6.2. Limitation

The limitation is that, if the system is large enough, a great number of messages will be generated, thus causing the speed of retrieving items to be slower.

5. Analysis Abstracts

5.1. Capacity

The Tested Grape system has the follow functions:

a) Functions about the group

The user can create or join some groups in which they can have some notifications and discussions and votes.

b) Casting a vote

The leader of a group can raise a vote and let the members to vote,which is convenient for determination. And the result will be displayed in a figure. Note that there are two types of vote for convenience of user: instant vote and long-lasting vote.

5.2. Flaws & Limitations

5.2.1. Some Existing Problems

a) There is no feedback when some operation is done,which may be not user-friendly.
b) There is no check when creating a vote,which may lead to crash when the user enters nothing.

5.2.2. Some Unrealized Functions

a) The leader of a group stops the vote or modifies it halfway
b) The countdown on the screen when voting.

5.3. Suggestions

Here are several suggestions we come up with after the testing activity:
a) Add some notes or comments in the source code in order to make other programmers understood more quickly.
b) Unify the coding style in our team as early as possible. Avoid aliasing especially in the naming activity.
c) Clarify the project architecture as early as possible in order to make other programmers know the whole framework better. Thus creating and maintaining a reliable system.

5.4. Evaluation

The Grape system can run in a correct and efficient way and its basic function do work according to its user's guide and manual.

6. Test Cost

No special cost is spent but the students' free time during the spring semaster.
Here list the details:

Item	Name	Price
Developing software	Python(flask.py as back end) html, css, js as basic front end. Mysql as database. Window 2007 server.	Free software.
Developing hardware	Lenovo Y400 as developing machine.	Our own computer.
Developing Time cost	/	Priceless