# *Term Project: Instant Messaging App*

Test Plan Document

Table of Contents

1 Introduction 2

1.1 Purpose 2

1.2 Target Audience 2

1.3 Terms and Definitions 2

2 Test Plan Description 3

2.1 Scope of Testing 3

2.2 Testing Schedule 3

2.3 Release Criteria 3

3 Unit Testing 4

3.1 <Authentication> 4

3.1.1 <Test Case 1> 4

3.2 <Create Channel> 4

3.2.1 <Test Case 1> 4

3.2.2 <Test Case 2> 4

3.3 <Channel Container> 4

3.3.1 <Test Case 1> 4

3.3.2 <Test Case 2> 4

3.4 <Edit Channel> 4

3.4.1 <Test Case 1> 4

3.4.2 <Test Case 2> 4

4 Feature Testing 5

4.1 <Authentication Test> 5

4.1.1 <Test Case 1> 5

4.1.2 <Test Case 2> 5

4.1.3 <Test Case 3> 5

4.1.4 <Test Case 4> 5

4.1.5 <Test Case 5> 5

4.2 < Outside Channel Test> 5

4.2.1 <Test Case 1> 5

4.2.2 <Test Case 2> 5

4.2.3 <Test Case 3> 5

4.2.4 <Test Case 4> 5

4.2.5 <Test Case 5> 5

4.2.6 <Test Case 6> 5

4.2.7 <Test Case 7> 5

4.2.8 <Test Case 8> 5

4.2.9 <Test Case 9> 5

4.2.10 <Test Case 10> 5

4.3 < Inside Channel Test> 5

4.1.1 <Test Case 1> 5

4.3.2 <Test Case 2> 5

4.3.3 <Test Case 3> 5

4.3.4 <Test Case 4> 5

4.3.5 <Test Case 5> 5

4.3.6 <Test Case 6> 5

4.3.7 <Test Case 7> 5

4.3.8 <Test Case 8> 5

4.3.9 <Test Case 9> 5

5 System Testing 6

# Introduction

**Provide an overview of this test plan.**

This is a test plan documentation for instant messaging app. Even after the implementation is complete, the testing process is essential because it is necessary to check whether this application is implemented properly.

## Purpose

**Describe the purpose of this document. How will this document help you or others when performing testing?**

This test plan documentation will present the necessary steps and approaches to what parts should be tested in the instant messaging app and how the test process should be.

## Target Audience

**Describe the target audience for this document.**

The target audience of this test plan documentation will be the development team members who implemented the instant messaging app and the project manager in charge of this application project.

## Terms and Definitions

**Define any terms or acronyms you will be using in the remainder of this document.**

# Test Plan Description

**Describe the purpose of this section and outline its contents. Only a few sentences are expected here.**

This section lists the full list of key tests required for an instant messaging app and explains the criteria for passing the tests. The overall test is divided into three main categories: authentication test, outside channel test, and inside channel test.

## Scope of Testing

**Describe the scope of the test plan. What areas of the project will you be testing, to what degree? Are there any aspects you will not be testing? What kind of tests will you perform, and what kind of tests will you omit?**

*Authentication Test*

- Whether account creation works normally

- Whether the login works normally

- Whether logout works normally

*Outside Channel Test*

- Whether channel creation works normally

- Whether channel deletion works normally

- Whether direct channel creation works normally

- Whether direct channel deletion works normally

- Whether channel search works normally (channel / direct channel)

*Inside Channel Test*

- Whether the channel modification works normally (change channel name / add users)

- Whether the thread is working normally (channel / direct channel)

- Whether sending / receiving message works normally

- Whether the loading history conversation works normally

*Tests that are difficult to proceed*

- Maximum account creation

- Maximum channel creation

- Maximum direct channel creation

- Channel search speed when user has too many channels

- Maximum number of concurrent users

## Testing Schedule

**Outline your testing schedule here. You are encouraged to design your own testing schedule template.**

Day 1: Authentication Test

Day 2: Outside Channel Test

Day 3: Inside Channel Test

## Release Criteria

**Describe your maximum fault tolerance for all your tests and the criteria your system must meet before being deployment.**

- Whether account creation works normally

If the correct input value is entered, an account should be created normally, and duplicate account creation should be prevented.

- Whether the login works normally

If the user enters the correct ID and password, the login should be successful and go to the main page. If the user enters an ID that does not exist or a password that does not match, the login should not be successful and remain in the log in page.

- Whether logout works normally

When the user presses the logout button, the user should be logged out and has to move from the main page to the authentication page.

- Whether channel creation works normally

When the user presses the channel creation button on the main page, a list of users to be added to the channel should be popped up with an input box to set the channel name. After setting the channel name, selecting users to add, and clicking the Create Channel button, a channel should be created with the corresponding channel name and user list, and the page should be automatically switched from the main page to the corresponding channel.

- Whether channel deletion works normally

When the user presses the delete channel button, the corresponding channel must be hard deleted, and even if the channel is re-created with the same name, the history conversation cannot be loaded.

- Whether direct channel creation works normally

When a user presses the Create Direct Channel button, a user list pops up, selects a desired user and presses Create Channel, then a one-on-one chat room with the user should be opened.

- Whether direct channel deletion works normally

When the user presses the delete channel button, the corresponding channel must be hard deleted, and even if the channel is re-created with the same user, the history conversation cannot be loaded.

- Whether channel search works normally (channel / direct channel)

When a user inputs a string in the search input box, the matching channel name or user ID should pop up, and when the user presses the desired channel, the page should move to the corresponding channel.

- Whether the channel modification works normally (change channel name / add users)

When the user presses the edit channel button, the channel name edit input box and user list should pop up, and when the user edits the channel name or adds a user and press save changes, the changes should be applied.

- Whether the thread is working normally (channel / direct channel)

When a user creates a thread based on a message in a channel conversation, an individual chat room should be opened on the right, and the conversation contents of the thread should not be visible to other users.

- Whether sending / receiving message works normally

When a user sends a message to a channel or direct channel, it should be displayed on that channel and then remain as a history conversation.

- Whether the loading history conversation works normally

After the application is closed, when the user logs in again and enters the channel where the previous conversation was going, the user should be able to see the previous conversation.

# Unit Testing

**Briefly explain the significance of performing unit test here. (Chapter9 p8-13)**

The aim of unit testing is to test program units in isolation. Tests should be designed to execute all of the code in a unit at least once. Individual code units are tested by the programmer as they are developed.

**Please also provide the overview of all the unit tests you have performed. What testing framework did you use? What is the outcome?**

*Authentication*

- Check whether the input information required for account creation is entered into the appropriate data socket

*Create Channel*

- When creating a channel, distinguish whether the type is "team" or "message", and check whether it is moved to the channel creation process if it is "team" and to direct message if it is "message"

*Channel Container*

- Check whether the channel container displays on the team channel list if the channel type is “team” and on the direct message list if the channel type is “message”

*Edit Channel*

- When editing a channel, if the channel type is "team", check whether the channel name edit input box, user list, save changes button, and delete channel button are displayed and if the channel type is "message", check whether only the delete channel button is displayed

*Outcome*

- All these unit tests were conducted frequently during application implementation, and the expected outcome was confirmed and proceeded to the next step.

**Describe this unit and its functionality in more detail here. Tell us the scope of your testing on this unit and the minimum criteria for release.**

## <Authentication>

Check whether the input information required for account creation is entered into the appropriate data socket

**Describe this test case and any sub-cases it may have in detail. Sequence diagrams or other graphics may help. List inputs you will give and outputs you expect to be returned. Explain how you will know if your project has passed this test case.**

### <Test Case 1>

Input: Full Name, Username, Phone Number, Avatar URL, Password, Confirm Password

Expected Output: All goes to appropriate data socket in user database and create account

Actual Output: All went to appropriate data socket and account created

## <Create Channel>

When creating a channel, distinguish whether the type is "team" or "message", and check whether it is moved to the channel creation process if it is "team" and to direct message if it is "message"

### <Test Case 1>

Input: Pressed create channel button

Expected Output: Moved to channel creation process

Actual Output: Moved to channel creation process

### <Test Case 2>

Input: Pressed direct message button

Expected Output: Moved to direct message creation process

Actual Output: Moved to direct message creation process

## <Channel Container>

Check whether the channel container displays on the team channel list if the channel type is “team” and on the direct message list if the channel type is “message”

### <Test Case 1>

Input: Created channel

Expected Output: Team channel list displays created channel

Actual Output: Team channel list displays created channel

### <Test Case 2>

Input: Created direct message

Expected Output: Direct message list displays created channel

Actual Output: Direct message list displays created channel

## <Edit Channel>

When editing a channel, if the channel type is "team", check whether the channel name edit input box, user list, save changes button, and delete channel button are displayed and if the channel type is "message", check whether only the delete channel button is displayed

### <Test Case 1>

Input: Pressed modification button within channel

Expected Output: Channel name edit input box, user list, save changes button, and delete channel button are displayed

Actual Output: Channel name edit input box, user list, save changes button, and delete channel button are displayed

### <Test Case 2>

Input: Pressed modification button within direct channel

Expected Output: Only the delete channel button is displayed

Actual Output: Only the delete channel button is displayed

# Feature Testing

**Briefly explain the significance of performing feature test here. (Chapter9 p14-17)**

Code units are integrated to create features. Feature tests should test all aspects of a feature. All of the programmers who contribute code units to a feature should be involved in its testing.

**Please also provide the overview of all the feature tests you have performed. What testing framework did you use? Are you satisfied with the outcome?**

*Authentication Test*

- Whether account creation works normally

- Whether the login works normally

- Whether logout works normally

*Outside Channel Test*

- Whether channel creation works normally

- Whether channel deletion works normally

- Whether direct channel creation works normally

- Whether direct channel deletion works normally

- Whether channel search works normally (channel / direct channel)

*Inside Channel Test*

- Whether the channel modification works normally (change channel name / add users)

- Whether the thread is working normally (channel / direct channel)

- Whether sending / receiving message works normally

- Whether the loading history conversation works normally

*Outcome*

After completing the implementation for the instant messaging app, feature tests were conducted for all the above-mentioned major features, and the test results were consistent with the expected outcomes.

**Describe this feature and its functionality in more detail here. Describe the scope of your testing on this feature and the minimum criteria for release. Specify if you were performing interaction test or usefulness test. (Chapter9 15)**

## <Authentication Test>

Whether account creation works normally

Whether the login works normally

Whether logout works normally

**Describe this test case and any sub-cases it may have in detail. Sequence diagrams or other graphics may help. List inputs you will give and outputs you expect to be returned. Explain how you will know if your project has passed this test case.**

### <Test Case 1>

Input: Entered all the required information and pressed create account

Expected Output: Account created

Actual Output: Account created

### <Test Case 2>

Input: Entered duplicate username when creating account

Expected Output: Account not created

Actual Output: Account not created

### <Test Case 3>

Input: Entered correct ID and appropriate matching password and pressed sign in

Expected Output: Successfully logged in and moved to main page

Actual Output: Successfully logged in and moved to main page

### <Test Case 4>

Input: Entered incorrect information when logging in

Expected Output: Log in failure and remain in sign in page

Actual Output: Log in failure and remain in sign in page

### <Test Case 5>

Input: Pressed log out button

Expected Output: Successfully logged out and moved to sign up page

Actual Output: Successfully logged out and moved to sign up page

## < Outside Channel Test>

Whether channel creation works normally

Whether channel deletion works normally

Whether direct channel creation works normally

Whether direct channel deletion works normally

Whether channel search works normally (channel / direct channel)

### <Test Case 1>

Input: Pressed create channel button, entered channel name, selected users

Expected Output: Channel created with correct channel name and only selected users are invited

Actual Output: Channel created with correct channel name and only selected users are invited

### <Test Case 2>

Input: Pressed create channel button, empty channel name, not selecting any user

Expected Output: Nothing happens

Actual Output: Nothing happens

### <Test Case 3>

Input: Pressed create channel button, entered channel name, not selecting any user

Expected Output: Channel created with correct channel name, and no one is invited

Actual Output: Channel created with correct channel name, and no one is invited

### <Test Case 4>

Input: Pressed delete channel within channel

Expected Output: Channel deleted and moved back to main page

Actual Output: Channel deleted and moved back to main page

### <Test Case 5>

Input: Pressed create direct message, selected user

Expected Output: Direct channel created with the selected user

Actual Output: Direct channel created with the selected user

### <Test Case 6>

Input: Pressed create direct message, not selecting any user

Expected Output: Nothing happens

Actual Output: Nothing happens

### <Test Case 7>

Input: Pressed delete channel within direct message

Expected Output: Direct channel deleted

Actual Output: Direct channel deleted

### <Test Case 8>

Input: Entered some common letters in the search input box

Expected Output: Displays channel names and usernames that contains the input

Actual Output: Displays channel names and usernames that contains the input

### <Test Case 9>

Input: Pressed channel name in the search result

Expected Output: Opens the channel

Actual Output: Opens the channel

### <Test Case 10>

Input: Pressed username in the search result

Expected Output: Opens the direct message

Actual Output: Opens the direct message

## < Inside Channel Test>

Whether the channel modification works normally (change channel name / add users)

Whether the thread is working normally (channel / direct channel)

Whether sending / receiving message works normally

Whether the loading history conversation works normally

### <Test Case 1>

Input: Pressed modification button within channel, changed channel name and pressed save changes

Expected Output: Channel name changed

Actual Output: Channel name changed

### <Test Case 2>

Input: Pressed modification button within channel, changed channel name, selected users, pressed save changes

Expected Output: Channel name changed, new users added

Actual Output: Channel name changed, new users added

### <Test Case 3>

Input: Pressed modification button within channel, did not change channel name, selected users, pressed save changes

Expected Output: New users added

Actual Output: New users added

### <Test Case 4>

Input: Pressed modification button within channel, did not change channel name, did not select any user, pressed save changes

Expected Output: Nothing changed and back to the channel

Actual Output: Nothing changed and back to the channel

### <Test Case 5>

Input: Pressed thread button on the message within channel

Expected Output: Thread created

Actual Output: Thread created

### <Test Case 6>

Input: Pressed thread button on the message within direct message

Expected Output: Thread created

Actual Output: Thread created

### <Test Case 7>

Input: Entered text and pressed enter in the channel

Expected Output: Message is displayed on the channel

Actual Output: Message is displayed on the channel

### <Test Case 8>

Input: Entered text and pressed enter in the direct message

Expected Output: Message is displayed on the direct message

Actual Output: Message is displayed on the direct message

### <Test Case 9>

Input: Entered channel where the conversation was going

Expected Output: Opens the channel and displays all the history conversation

Actual Output: Opens the channel and displays all the history conversation

# System Testing

**Briefly explain the significance of performing system testing here. (Chapter9 p33-34)**

System testing verifies that an application performs tasks as designed. This step, a kind of black box testing, focuses on the functionality of an application. System testing, for example, might check that every kind of user input produces the intended output across the application.

**Provide detailed process of system testing and its result. Please also remember to add security testing (Chapter9 p41-44). Make sure your system is safe from injection attacks, cross-site scripting attacks, session hijacking attacks, brute force attacks (Chapter7) and so on.**

*System Test*

Sending bulk messages (sending 50 messages less than 30 seconds)

* Expected Output: The system should be reliable and cause no system error
* Actual Output: The system was reliable and there was no system error

Message sending latency

* Expected Output: It should not take more than 1 second to send a message
* Actual Output: No latency in sending a message

Sending and receiving multimedia (images / documents)

* Expected Output: Successful file transfer with no damage to file
* Actual Output: Successful file transfer with no damage to file

*Security Test*

Reliable authentication process

* Expected Output: All user’s passwords are hashed and cannot be cracked
* Actual Output: All user’s passwords are hashed and cannot be cracked

**If all your tests results have met with the release criteria. Congratulations! Your system is now finally ready for deployment.**