

# Software Test Plan

Copyright © 2015 Luke Kazmierowicz

Version 1.0

December 7, 2015

TauNet System

Submitted in partial fulfillment of the requirements of  
CS 300 Elements of Software Engineering

## Table of Contents

Table of Contents	i
1.0. Introduction	1
1.1. Reference Material	1
2.0. Independent Module Testing	1
2.1 Contact	1
2.2 Message	1
2.3 CipherSaber2	1
3.0. Sending and Receiving Messages	2
3.1 EchoServer Test	2
3.2 Test Sending Message to Classmate	2
3.3 Test Receiving Message from Classmate	3
3.4 Test Sending Message to Unavailable User	3
3.5 Test Receiving a Message While Composing One	3
3.6 Test Replying to the Last Received Message	4
3.7 Test Receiving a Message from an Unknown Sender	4
4.0. Menu Selection Tests	4
4.1 Quitting the Program ( 'Q' )	5
4.2 Composing a Message ( 'C' )	5
4.2 Replying to a Message ( 'R' )	5

## **1.0. Introduction**

The purpose of this document is to describe how the TauNet software system has been tested. Our testing plan is to first break the system up into its individual components and test them separately from the larger system. Then we will test the system as a whole to verify that it meets the requirements and behaves as expected. This document is intended as a reference for the developer and customer of the TauNet software system.

### **1.1. Reference Material**

TauNet Documentation:

- *Software Requirements Specification Document for TauNet system, by Luke Kazmierowicz*
- *TauNet Communication Protocol Document*
- *Software Design Document for TauNet system, by Luke Kazmierowicz*

## **2.0. Independent Module Testing**

This section will go through the tests for each independent module in the software system. Because the software was build around an OO design the modules will be the classes in the system. Here we will look at each secondary class that can be tested independently of the primary classes.

### **2.1. Contact Class**

*Testing for the Contact class can be found in the main function of the Contact class.*

### **2.2. Message Class**

*Testing for the Message class can be found in the main function of the Message class.*

### **2.3. CipherSaber2 Class**

*Testing for the CipherSaber2 class can be found in the main function of the CipherSaber2 class.*

### 3.0 Sending and Receiving Messages

Because the Sender, Receiver and DisplayMessages class all work together we will test them as a unit. To test them I will manually send and receive messages to and from the TauNet EchoServer. I will also send and receive messages from other classmate's TauNet systems and verify the outcome of each test with the expected output. The expected output is decided from the TauNet SDD. User input is marked in bold.

#### 3.1 EchoServer Test

Expected outcome:

```
Enter the username you wish to send a message to: bart
Enter your message: Test
Your message was successfully transmitted.
Waiting for incoming messages...
```

```
Message from bart: DATE TIME IP:PORT
version: 0.2
from: luke
to: bart
```

```
Test
```

Actual outcome:

```
Enter the username you wish to send a message to: bart
Enter your message: test
Your message was successfully transmitted.
Waiting for incoming messages...
```

```
Message from bart: 2015-12-03 14:16:27-0800
198.29.41.107:40517
version: 0.2
from: luke
to: bart
```

```
test
```

#### 3.2 Test Sending Message to Classmate

Expected outcome:

```
Enter the username you wish to send a message to: natreed
Enter your message: Reply if you get this message!
```

Your message was successfully transmitted.  
Waiting for incoming messages...

Actual outcome:

Enter the username you wish to send a message to: **natreed**  
Enter your message: **Reply if you get this message!**  
Your message was successfully transmitted.  
Waiting for incoming messages...

### **3.3 Test Receiving Message from Classmate**

Expected outcome:

Message from natreed: *some message*

Actual outcome:

Message from natreed: I got it.

### **3.4 Test Sending Message to Unavailable User**

Expected outcome:

Enter the username you wish to send a message to: lukekaz9  
Checking connection...  
The user "lukekaz9" is currently unavailable. Try again later.

Actual outcome:

Enter the username you wish to send a message to: lukekaz9  
Checking connection...  
The user "lukekaz9" is currently unavailable. Try again later.

### **3.5 Test Receiving a Message While Composing One**

The expected outcome is for the received message to be displayed as soon as the message being composed is sent. It would look like this:

Enter the username you wish to send a message to: **natreed**  
Checking connection...  
Enter your message: **I was sent a message while writing this.**  
Your message was successfully transmitted.  
Waiting for incoming messages...

Message from natreed: Did I interrupt something?

When manually running this test I received the same outcome:

```
Enter the username you wish to send a message to: natreed
Checking connection...
Enter your message: This is a real test.
Your message was successfully transmitted.
Waiting for incoming messages...

Message from natreed: Ok, did it work?
```

### **3.6 Test Replying to the Last Received Message**

If the user enters 'R' after a message is received they should be presented a way to reply to the sender of that message. We tested this and the output was as expected:

```
Message from natreed: Hey luke. Let me know if you get this
r
Enter your reply to user "natreed": I did!
Your message was successfully transmitted.
Waiting for incoming messages...
```

### **3.7 Test Receiving a Message from an Unknown Sender**

If a message is received from a user who is not in our contacts it should still be displayed. However, if we try to reply to it we should get an error like this:

```
Message from SomeUnknownUser: test
r
Can't reply to the message because the user is not in your
contacts.
```

We tested this and the outcome was as follows:

```
Message from lukekaz9: test
r
Can't reply to the message because the user is not in your
contacts.
```

## **4.0 Menu Selection Tests**

In this section we will test each menu option and verify that it performs the expected action. The menu options are as follows:

- Enter 'Q' to quit.
- Enter 'C' anytime to compose a message.
- Enter 'R' to reply to the last received message.

#### **4.1 Quitting the Program ('Q')**

When the user enters 'Q' the program should immediately terminate. We ran this test and the program gave the following output:

```
q
Program closing...
Process finished with exit code 0
```

#### **4.2 Composing a Message ('C')**

When the user enters 'C' they should be presented with the following menu:

```
Authorized TauNet users:
  Username: someUser1
  Username: someUser2
  Username: someUser3
Enter the username you wish to send a message to:
```

The list of authorized TauNet users should be loaded from the data.txt file and presented. We created a test data.txt file and tested this command. The output was as follows:

```
Authorized TauNet users:
  Username: bart
  Username: brodie
  Username: dom
  Username: LukePi
  Username: mancat
  Username: natreed
  Username: patter5
  Username: pyrrh
  Username: relsqui
  Username: tdulcet
Enter the username you wish to send a message to:
```

The contacts were successfully loaded and displayed as expected.

#### **4.3 Replying to a Message ('R')**

This option was tested in section 3.6 *Test Replying to the Last Received Message* and passed the test.