Software Requirements Specification

Version 1.0

October 23, 2015

TauNet v1

Rachael Johnson

Table of Contents

1.0 Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description and specification of the requirements for TauNet provided by the customer. It will outline customer expectations and the expected outcome of TauNet.

1.2 Scope

TauNet will be published online as an open source software alternative for users wanting a secure, private network with which to communicate. As such, the scope is limited to those who have access to the internet and who will also have access to a Raspberry Pi, as outlined in the Requirement Specification section.

1.3 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

Joan, Paul Adams, Bobbie Baker, and Charles Charlie. "Software Requirements Specification." (n.d.): n. pag. 15 Apr. 2004. Web. 23 Oct. 2015.

2.0 General Description

2.1 Overview

The idea behind TauNet is to allow two or more users, each with a RaspberryPi, to communicate with each other securely and privately over the internet. TauNet will be able to be setup by the customer, to allow the customer to interact with other people who he determines should be added to his TauNet network.

3.0 Requirements Specification

3.1 Functional Requirements

The Main Functional Requirement for TauNet is a piece of software that allows the user to communicate with his friends across a private network.

3.1.1 Address Book

The address book is needed in TauNet to allow the user to choose from a list of users to send a message.

3.1.2 Log of Unsent Messages

Another requirement is the ability to access a log of messages that were not successfully sent to a user (i.e. the user was offline). This log should allow the user to resend messages without having to retype the whole message to a user.

3.2 Non-Functional Requirements

3.2.1 Hard Requirements:

- Short text messages (arbitrary length, 300 char is fine)
- Over internet, from anywhere to anywhere
- Taking down a node won't take out the entire network
- Username is necessary
- 1 to 1 messaging
- Minimum of 12 nodes supported, max of 300
- Each instance of the network is separate
- No gui, must be able to use terminal
- Text Message communication must be encrypted

- The network gets formed when it gets created
- Username/ ip address and port / encryption keys per node (hard-wired)
- One RaspberryPi per User

3.2.2 Soft (Optional) Requirements:

- Message to more than 1 person at a time
- Ability to add users in the future
- Ability to delete users
- Unicode Rendering / Input
- Resistance to traffic analysis (anonymise traffic)
- Each network is invite only (only a member of the network can invite new members)

3.2.3 Notes (General):

- Manual enrollment to the network is fine
- Don't have to automatically join the network after physically moving pi
- One user per pi, each user can be hardcoded to a particular pi (another user can't use the same pi)
- Use whatever language to program that makes you happy

3.2.4 Constraints:

- Due Monday of finals week
- Raspberry Pi, Monitor, and Keyboard

3.2.5 Notes on Security:

- Advised to use TLS / SSL, or RC4 (Google CipherSaber)

3.2.6 Notes on Protocol:

- The complete documentation for protocol is located at http://moodle.svcs.cs.pdx.edu/mod/page/view.php?id=646. Note, a moodle username and password is needed to login and obtain the document.

3.3 Use Cases

3.3.1 Selecting a User

Preconditions	User accesses the Pi and is at the main screen of TauNet
Main Flow	User Types in “Address Book” or /a and presses enter
Sub-flows	“Address Book” screen displays, and users are numbered. The User types in the number of the user he wants to send a message to, and then presses enter.
Alternative Flows	Address Book is empty, in which case an error message displays explaining that the address book contains no addresses. The ability to add addresses will be included in a later version of the software.

3.3.2 Sending a Message

Preconditions	3.3.1 or 3.3.5
Main Flow	The User then types a message, followed by the “enter Key” to send the message.
Sub-flows	When the message is sent, the program will display a message status. If the TauNet program on the receiving end is online and has received the message successfully, then a “Message Received” message will be displayed to the user.

Alternative Flows	If the message is not received successfully, a message will display after a period of time saying “Message Not Received”. The text for the message is reloaded and the user can press enter again to send it.

3.3.3 Receiving a Message

Preconditions	User is at the Main TauNet screen
Main Flow	From the main screen, any messages that are sent instantly appear
Sub-flows	User can enter /r to reply to the most recent message
Alternative Flows	

3.3.4 Sending a Message to an Offline User

Preconditions	3.3.1 or 3.3.5
Main Flow	The user sends a message to an offline user
Sub-flows	A message will appear explaining the message was not received. The user can then type /log-add to add the message to a log to send later
Alternative Flows	

3.3.5 Alternative ways to send messages

Preconditions	User is at Main TauNet screen
Main Flow	User types /u followed by the person the user would like to send a message to. I.E. /u Nick Hai!
Sub-flows	The user then presses enter, and the “Hai!” is sent to Nick.
Alternative Flows	

3.3.6 Accessing Not-Received Message Log

Preconditions	User is at Main TauNet screen
Main Flow	User types /log. The TauNet then displays a numbered list of messages, and the user they were to be sent to.
Sub-flows	The user can enter a number and press enter to resend, or they can enter a number and /d to delete the logged message
Alternative Flows	

3.3.7 User returns to Main Screen

Preconditions	User is not at Main TauNet screen
Main Flow	The User can type /main and press enter at any point to return to the main TauNet screen.
Sub-flows	
Alternative Flows	The user can not be typing a message or anything else, if he is, the /main command will not work and instead will be treated as an extension of the previous command.