Use Case Description

Use Case Name	Form DATA	
Brief Description	Client reads up to 512 bytes form the file and stores in new TFTP DATA packet	
Precondition	Client sent WRQ request to Server and received an ACK packet	
Primary Actor	Client	
Secondary Actor	N/A	
Dependencies	N/A	
Generalization	N/A	
Basic Flow	**RFS** N/A	
	Steps	1. Client reads up to 512 bytes from file 2. Client stores it into a new TFTP DATA packet
	Postcondition	New TFTP DATA packet has been created.
Specific Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A
Global Alternative Flows	N/A	
	Steps	N/A
	Postcondition	N/A
Bounded Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A

Use Case Name	Form Datagram	
Brief Description	Client forms a UDP datagram packet containing a TFTP DATA packet OR Server forms a UDP datagram packet containing a TFTP ACK or ERROR packet	
Precondition	Client has read up to 512 bytes from file and created a DATA packet OR Server has verified the DATA packet	
Primary Actor	Client, Server	
Secondary Actor	Server, Client	
Dependencies	N/A	
Generalization	N/A	
Basic Flow	**RFS** N/A	
	Steps	1. Client creates UDP datagram containing the TFTP DATA packet to be sent to the Server OR Server creates UDP datagram packet containing the TFTP ACK or ERROR packet to be sent to the Client
	Postcondition	New UDP datagram packet containing the DATA, ACK or ERROR packet has been created.
Specific Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A
Global Alternative Flows	N/A	
	Steps	N/A
	Postcondition	N/A
Bounded Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A

Use Case Name	Send Datagram	
Brief Description	Client sends the UDP datagram packet containing the TFTP DATA packet to the ephemeral port OR Server sends the UDP datagram packet containing the TFTP ACK packet to Port 69	
Precondition	Client OR Server has made a UDP datagram packet.	
Primary Actor	Client, Server	
Secondary Actor	Datagram Socket	
Dependencies	N/A	
Generalization	Ephemeral Port, Port 69	
Basic Flow	**RFS** N/A	
	Steps	1. Client sends UDP datagram containing the TFTP DATA packet to the ephemeral port (created by the Server during connection establishment OR Server sends UDP datagram packet containing the TFTP ACK packet Port 69 (Client's Datagram Socket)
	Postcondition	UDP datagram packet containing the DATA or ACK has been sent.
Specific Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A
Global Alternative Flows	N/A	
	Steps	N/A
	Postcondition	N/A
Bounded Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A

Use Case Name	Receive Datagram	
Brief Description	Client receives the UDP datagram packet containing the TFTP ACK packet from the ephemeral port OR Server receives the UDP datagram packet containing the TFTP DATA packet from Port 69	
Precondition	Client has read up to 512 bytes from file and created a DATA packet OR Server has verified DATA packet	
Primary Actor	Client, Server	
Secondary Actor	Datagram Socket	
Dependencies	N/A	
Generalization	N/A	
Basic Flow	**RFS** N/A	
	Steps	1. Client receives UDP datagram containing the TFTP ACK packet from the ephemeral port (created by the Server during connection establishment OR Server receives UDP datagram packet containing the TFTP DATA packet Port 69 (Client's Datagram Socket)
	Postcondition	UDP datagram packet containing the DATA or ACK has been created received.
Specific Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A
Global Alternative Flows	N/A	
	Steps	N/A
	Postcondition	N/A
Bounded Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A

Use Case Name	Extract Message	
Brief Description	Server extracts message from the received UDP datagram packet OR Client extracts message from the received UDP datagram packet	
Precondition	Server has received UDP datagram packet from Port 69 OR Client has received UDP datagram packet from Ephemeral Port	
Primary Actor	Server, Client	
Secondary Actor	Client, Server	
Dependencies	N/A	
Generalization	N/A	
Basic Flow	**RFS** N/A	
	Steps	1. Server extracts message from UDP datagram packet that should contain the TFTP DATA packet from Port 69 (Client's Datagram Socket) OR Client extracts message from UDP datagram that should contain the TFTP ACK packet from Ephemeral Port
	Postcondition	Server or Client has extracted the message from the UDP datagram packet.
Specific Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A
Global Alternative Flows	N/A	
	Steps	N/A
	Postcondition	N/A
Bounded Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A

Use Case Name	Verify DATA	
Brief Description	Server verifies that the received message is a valid TFTP DATA packet	
Precondition	Server has extracted the message from the UDP datagram packet	
Primary Actor	Server	
Secondary Actor	N/A	
Dependencies	N/A	
Generalization	N/A	
Basic Flow	**RFS** N/A	
	Steps	1. Server verifies message from UDP datagram is a valid TFTP DATA packet
	Postcondition	Server has verified the message from the datagram.
Specific Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A
Global Alternative Flows	N/A	
	Steps	N/A
	Postcondition	N/A
Bounded Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A

Use Case Name	Write File	
Brief Description	Server extracts data from the TFTP DATA packet and writes it to the file.	
Precondition	Server has extracted the message from the UDP datagram packet	
Primary Actor	Server	
Secondary Actor	N/A	
Dependencies	N/A	
Generalization	N/A	
Basic Flow	**RFS** N/A	
	Steps	1. Server extracts data from the TFTP DATA packet and writes to file
	Postcondition	Server has written data to the file.
Specific Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A
Global Alternative Flows	N/A	
	Steps	N/A
	Postcondition	N/A
Bounded Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A

Use Case Name	Form ACK	
Brief Description	Server creates TFTP ACK packet	
Precondition	Server has written into the file.	
Primary Actor	Server	
Secondary Actor	N/A	
Dependencies	N/A	
Generalization	N/A	
Basic Flow	**RFS** N/A	
	Steps	1. Server creates new TFTP ACK packet after writing into the file
	Postcondition	New TFTP ACK packet has been created.
Specific Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A
Global Alternative Flows	N/A	
	Steps	N/A
	Postcondition	N/A
Bounded Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A

Use Case Name	Verify ACK	
Brief Description	Client verifies that the received message is a valid TFTP ACK packet	
Precondition	Client has extracted the message from the UDP datagram packet	
Primary Actor	Client	
Secondary Actor	N/A	
Dependencies	N/A	
Generalization	N/A	
Basic Flow	**RFS** N/A	
	Steps	1. Client verifies message from UDP datagram is a valid TFTP ACK packet
	Postcondition	Client has verified the message from the datagram.
Specific Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A
Global Alternative Flows	N/A	
	Steps	N/A
	Postcondition	N/A
Bounded Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A

Use Case Name	Form ERROR	
Brief Description	Server forms a new TFTP ERROR packet when received packet is not as unexpected.	
Precondition	Server has verified received UDP datagram packet from Client	
Primary Actor	Server	
Secondary Actor	N/A	
Dependencies	N/A	
Generalization	N/A	
Basic Flow	**RFS** N/A	
	Steps	1. Server creates TFTP ERROR packet based on proper error code
	Postcondition	New TFTP ERROR packet has been created.
Specific Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A
Global Alternative Flows	N/A	
	Steps	N/A
	Postcondition	N/A
Bounded Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A

Use Case Name	Verify ERROR	
Brief Description	Client verifies that the received message is a TFTP ERROR packet	
Precondition	Client has extracted the message from the UDP datagram packet	
Primary Actor	Client	
Secondary Actor	N/A	
Dependencies	N/A	
Generalization	N/A	
Basic Flow	**RFS** N/A	
	Steps	1. Client verifies message from UDP datagram is a TFTP ERROR packet
	Postcondition	Client has verified the message from the datagram.
Specific Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A
Global Alternative Flows	N/A	
	Steps	N/A
	Postcondition	N/A
Bounded Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A

Use Case Name	Handle ERROR	
Brief Description	Client handles TFTP ERROR based on the error code	
Precondition	Client has extracted the message from the UDP datagram packet	
Primary Actor	Client	
Secondary Actor	N/A	
Dependencies	N/A	
Generalization	N/A	
Basic Flow	**RFS** N/A	
	Steps	1. Client retransmits, or times-out based on error code
	Postcondition	Client attempts to fix the error either by retransmitting or by timing out.
Specific Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A
Global Alternative Flows	N/A	
	Steps	N/A
	Postcondition	N/A
Bounded Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A

Use Case Name	Select Operation	
Brief Description	Error Simulator creates an intentional error within the program	
Precondition	Client, Server, and Host are running	
Primary Actor	Error Simulator	
Secondary Actor	N/A	
Dependencies	N/A	
Generalization	Damage Packet, Normal, Lose Packet, Delay Packet, and Duplicate Packet	
Basic Flow	**RFS** N/A	
	Steps	1. Error Simulator creates an error based on User Input: 0. Normal Mode 1. Losing a Packet 2. Delaying a Packet 3. Duplicating a Packet 4. Damaging a Packet
	Postcondition	Error Simulator creates the corresponding error based on user input
Specific Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A
Global Alternative Flows	N/A	
	Steps	N/A
	Postcondition	N/A
Bounded Alternative Flows	**RFS** N/A	
	Steps	N/A
	Postcondition	N/A