Explanation of System Responsibilities

For: Socket Library

Created by: Josh Lee on 11/30/15

Last Modified by: Josh Lee on 11/30/15

# Overview:

The Socket Library is meant to provide a collection of classes and interfaces for asynchronous communication over sockets. The classes within this library will handle the creation of sockets, threads, the sending and receiving of data, and other functions that may be needed to achieve this goal. The classes within this library will not handle formatting, interrupting, or any form of data manipulation outside of what is needed for sending data.

# Responsibilities of Current Classes and Interfaces:

There are currently 2 classes and 1 interface in the Socket Library. Only the UDP\_Sock is currently scheduled for implementation, but the TCP\_Sock is there for completeness and will be implemented if there are available resources.

## NetworkInterface:

The network interface is a pure abstract class that contains all the necessary methods for a socket to function. All sockets, both present and future, must implement this interface to be included in this library. This interface should never be changed unless absolutely necessary and only if express permission is given from a system architect.

## UDP\_Sock:

The UDP\_Sock is a concreate class that handles sending and receiving data raw both synchronously and asynchronously over UDP based sockets.

## TCP\_Sock:

The TCP\_Sock is a concreate class that handles sending and receiving data raw both synchronously and asynchronously over TCP based sockets.