Kyle Parry

Peter Brady

Samuel Jenkins

CSE 687 Object Oriented Design Project

Test Harness – Phase #3

**Architecture**

The architecture of the test harness will consist of three components: Executer, Harness and Logger.

**Executer:** The Executer will be the controller that operates the program and calls and runs the Harness with the tested program. This determines log level and the file location. It also determines the test to be performed. This sends messages to the harness via Message and Comm class. Also, creates multiple threads to execute tests at a given time.

**Harness:** The Harness is where the functionality will be that will take the callable object and run the tests to be performed and then return a true or false if the function is operating appropriately. The Harness will have a Logger that will log the results of the tests against the tested program. It passes along the log level and file string to the logger. It also sends the message and level 2 log for that individual message. Receives messages from the Executer via Message and Comm class

**Logger:** The logger is the functionality of the Test Harness that will log the results back to the Harness and the Executor to log to the user what the results of the tested program are. The logger then sends all the information to the output file and the location.

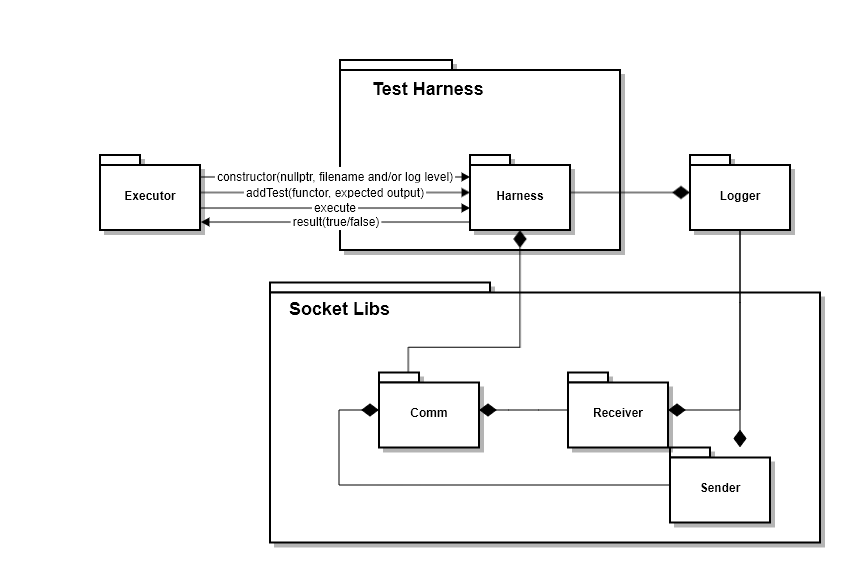
**Socket:** parent class to the SocketConnector and SocketListener.

**SockerListener:** Used to listen for messages and receive the message.

**SocketConnector:** Used to connect to and send messages to the listener.

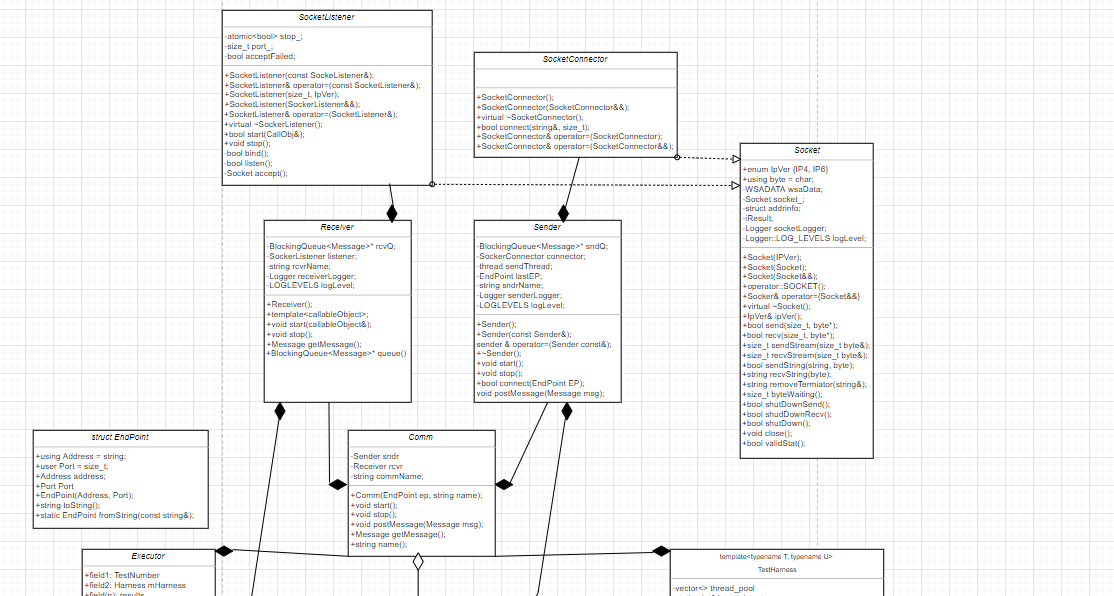
**BlockingQueue:** blocks from executing until the whole message is sent.

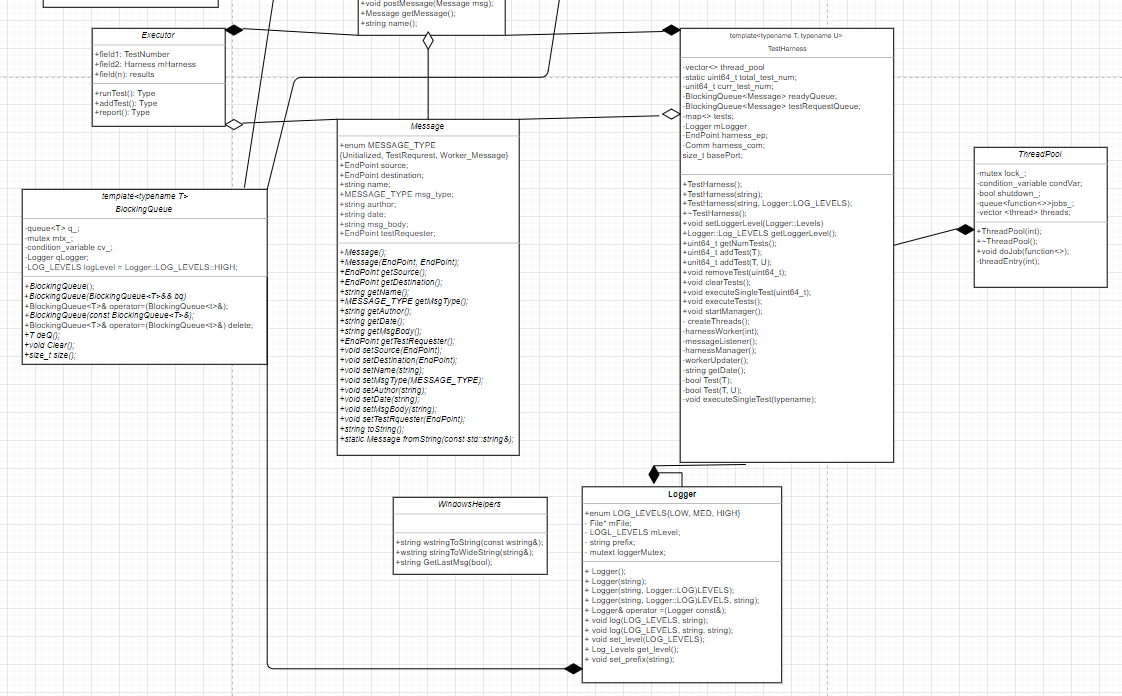
**ThreadPool:** Creates multiple threads and reuses the threads to execute multiple tests passes to the harness and executed at the same time.



**Design**

The design of the program is that it will consist of three classes: Executer, Harness, and Logger. Each of these classes will have the given member variables and methods as shown in the UML diagram below.



0

Peter Brady worked on the socket communication

Kyle Perry worked on the message passing from client to test harness and running the tests

Samuel Jenkins worked on the multithreading and the ThreadPool for thread reuse.