Load Balancer for an application server farm

|  |  |
| --- | --- |
| Document Version | 0.01 |
| Document Status | Initial Draft |
| Issue Date | 2013-03-11 |
| Project | Final Project |
| Term | Winter 2013 |
| Course | Computer Networks : coen 233-01 |

**Abstract:**

This Test Plan specifies the design of the code and the test results for a reverse proxy system that provided round-robin load balancing of network connections and fault-tolerance features to an application server farm.

**Table of Contents**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. Description of Code 3
2. Storage Location
3. Software Used
4. Environment setup
5. How to execute code?
6. What to expect?
7. Output

# **Description of Code**

## **Storage Location**

1. The following files are stored under the folder,

* LoadBalancer.java
* Server.java
* Server1.java
* Client.java
* Client1.java
* Client2.java
* Client3.java
* ReadMe.doc
* Project Report.docx
* Load balancer – A single instance of the load balancer is implemented in LoadBalancer.java.
* Server – Two separate (and mostly identical) instances of servers are implemented in Server.java and Server1.java.
* Client –Four separate instances of identical client implementations Client.java, Client1.java, Client2.java and Client4.java.

These files were used for a simulation 4 clients, 2 servers and 1 load balancer for a classroom demo. With very minimal or no extensions to client, server and load balancer java files, the simulation can be extended to support a large no. of servers and clients.

1. **Software Used**
2. Language Used: Java
3. Application: Eclipse and mysql workbench
4. **Environment setup**

i. Steps to create Database

1) Goto <http://www.mysql.com/downloads/>

2) For details on how to install MySQL, please refer to the following link:

<https://www.youtube.com/watch?v=iP1wOSsKjW>8

3) Download the JDBK connector and include the jar file in your project.

<http://www.mysql.com/products/connector/>

4) Install eclipse with java1.5.\* or higher

1. **How to execute code?**

1) Run the Load Balancer process.

2) Start up the Database.

2) Run the Server processes.

3) Run the Client processes.

**Note:** Input text files are read from a path specified in the clients program, the path can be changed to the location of the files, the text files contain the Clients id and First Name. For example: 3 PatientName

1. **What to expect?**

1) Initially when all the processes are running:

a) The servers’ status is reported on the Load Balancer every 5 seconds.

b) The server status port and listener port is included in connection table of load balancer.

c) The clients request will be processed i.e, when the patients ID and First Name are passed; the server gives the patients details from the database.

2) If one of the servers go down:

a) The Load Balancer reports if the server is down.

b) All of the incoming connections from clients are sent to the next available server. If server is not available, client request is held till a a server is available.

1. **Output**

Please find the output screen shots on the project report on section7 [Data analysis and Discussion]