**Load Testing** in which the performance of a software application is tested under a specific expected load.

**Example**: Let’s assume that our client requirement for a Login page is 2-5 sec and this 2-5 sec should be consistent all throughout until the load is 5000 users. So what should we observe hear? Is it just the load handling capability of the system or is it just the response time requirement?

The answer is both. We want the system which can handle a load of 5000 users with the response time of 2-5 seconds for all the concurrent users.

So what is meant by a concurrent user and a virtual user?

Concurrent users are those who log in to the application and at the same time, perform a set of activities together and log off the application at the same time. On the other hand, virtual users just hop in and hop out of the system irrespective of the other user activities.

**With a proper Load test, we can have an exact understanding of the following**:

* The number of users the system is able to handle or is capable of scaling to.
* The response time of each transaction.
* How does each component of the entire system behave under Load i.e Application server components, web server components, Database components etc.
* What server configuration is best to handle the load?
* Whether the existing hardware is enough or is there any need for additional hardware.
* Bottlenecks like CPU utilization, Memory Usage, Network delays, etc., are identified.

**Approach**

* we need information on how much is the current application handling capability
* what is the targeted load
* what is the expected response time
* The response time of the Login page shouldn’t be more than 5 sec even during the max load conditions.
* CPU utilization should not be more than 80%.
* The throughput of the system should be 100 transactions per sec.

**Example**

Overview of the application – Let’s assume an online shopping, where the users will log into the application and have a wide variety of dresses to shop, and they can navigate across each product.

To view the details about each product, they need to click on the product. If they like the cost and make of the product, then they can add to the cart and buy the product by checking out and making the payment.

Given below are a list of scenarios:

1. Browse – Here, the user launches the application, Logs into the application, Browses through different categories and Logs out of the application.
2. Browse, Product View, Add to Cart – Here, the user logs into the application, Browses through different categories, views product details, adds the product to cart and Logs out.
3. Browse, Product View, Add to Cart and Check out – In this scenario, the user logs into the application, Browses through different categories, views product details, adds the product to the cart, does check out and Logs out.
4. Browse, Product view, Add to cart Check out and Makes Payment – Here, the user logs into the application, Browses through different categories, views product details, adds the product to the cart, does check out, makes Payment and Logs out.

| S.No | Business Flow | Number of Transactions | Virtual User Load | Response Time (sec) | % Failure rate allowed | Transactions per hour |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Browse | 17 | 1600 | 3 | Less than 2% | 96000 |
| 2 | Browse, Product View, Add to Cart | 17 | 200 | 3 | Less than 2% | 12000 |
| 3 | Browse, Product View, Add to Cart and Check out | 18 | 120 | 3 | Less than 2% | 7200 |
| 4 | Browse, Product view, Add to cart Check out and Makes Payment | 20 | 80 | 3 | Less than 2% | 4800 |

The above values were derived based on the following calculations:

* Transactions per hour = Number of users\*Transactions made by a single user in one hour.
* The number of users = 1600.
* The total number of transaction in the Browse scenario = 17.
* Response Time for each transaction = 3.
* Total time for a single user to complete 17 transactions = 17\*3 = 51 rounded to 60 sec (1 min).
* Transactions per hour = 1600\*60 = 96000 Transactions.

More-:<https://www.softwaretestinghelp.com/load-testing/>