# 

TEST DOCUMENT TITLE

Quality Assurance of the solution

[Document information](#h.qwydk7x1nscw)

[Document version](#h.qgvwgi1jjpq)

[Approval List](#h.6r7j8dh65x32)

[Confidentiality Rating](#h.hes98nyepwis)

[General](#h.1agiq4l899wt)

[Stress Performance Test](#h.hk3acqe9xl2n)

[User Acceptance Test](#h.mz9eoe1a5djk)

[Other Tests?](#h.e4pd5f1cycku)

# Document information

## Document version

|  |  |  |
| --- | --- | --- |
| Version | Author e-mail | Description |
| 1.0 | trlb@kea.dk | Initial draft |
|  |  |  |

## Approval List

|  |  |  |
| --- | --- | --- |
| Who | Function | E-mail |
| Nikolaj B. Hemmeshøj | Head of Enterprise Architecture | nibh@kea.dk |
| Jarl Tuxen | Chief Information Security Officer | jart@kea.dk |

## Confidentiality Rating

|  |  |
| --- | --- |
| Rating |  |
| Company Confidential | X |
| Non Confidential |  |

# **General**

This document contains information about our testing process.

We need our performance testing to ensure that our program is fast and error free under any user load. And to ensure that the application will work as intended when all software units and methods are executed together.

# **Stress Performance Test**

**Load test:**

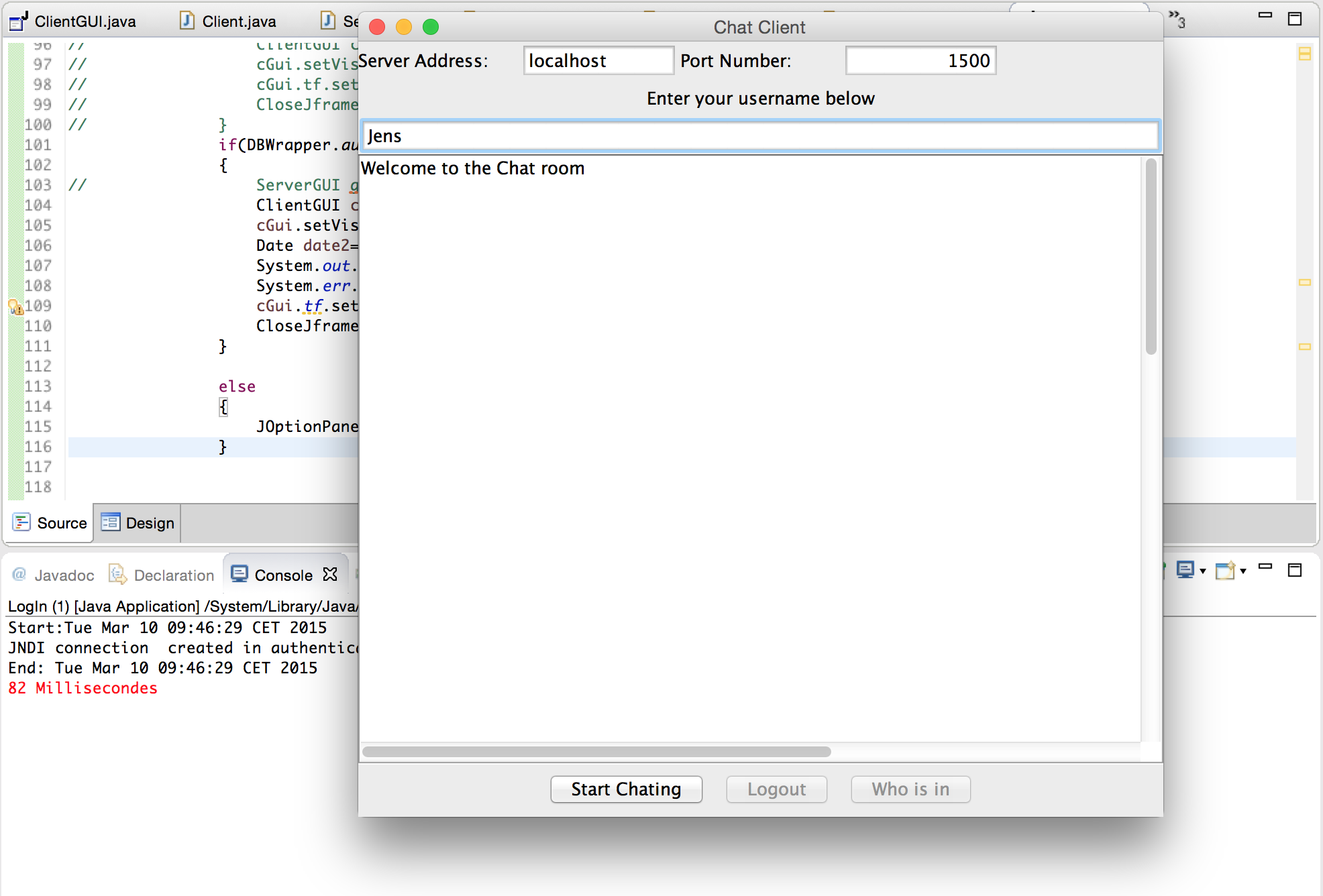
This test is for the process of putting demand on a system and measuring its response. In JMeter testing tool we tried to test our system by specifying the number of transactions and number of threads.

But since we couldn’t measure our system response in JMeter, AppPerfect or any other testing tools, we used (System.***out***.println()).

**Load test case: User Log in:**

We have tested how long it does takes after the user entered his/her username and password and clicked Log on until he/she entered the Chatting.

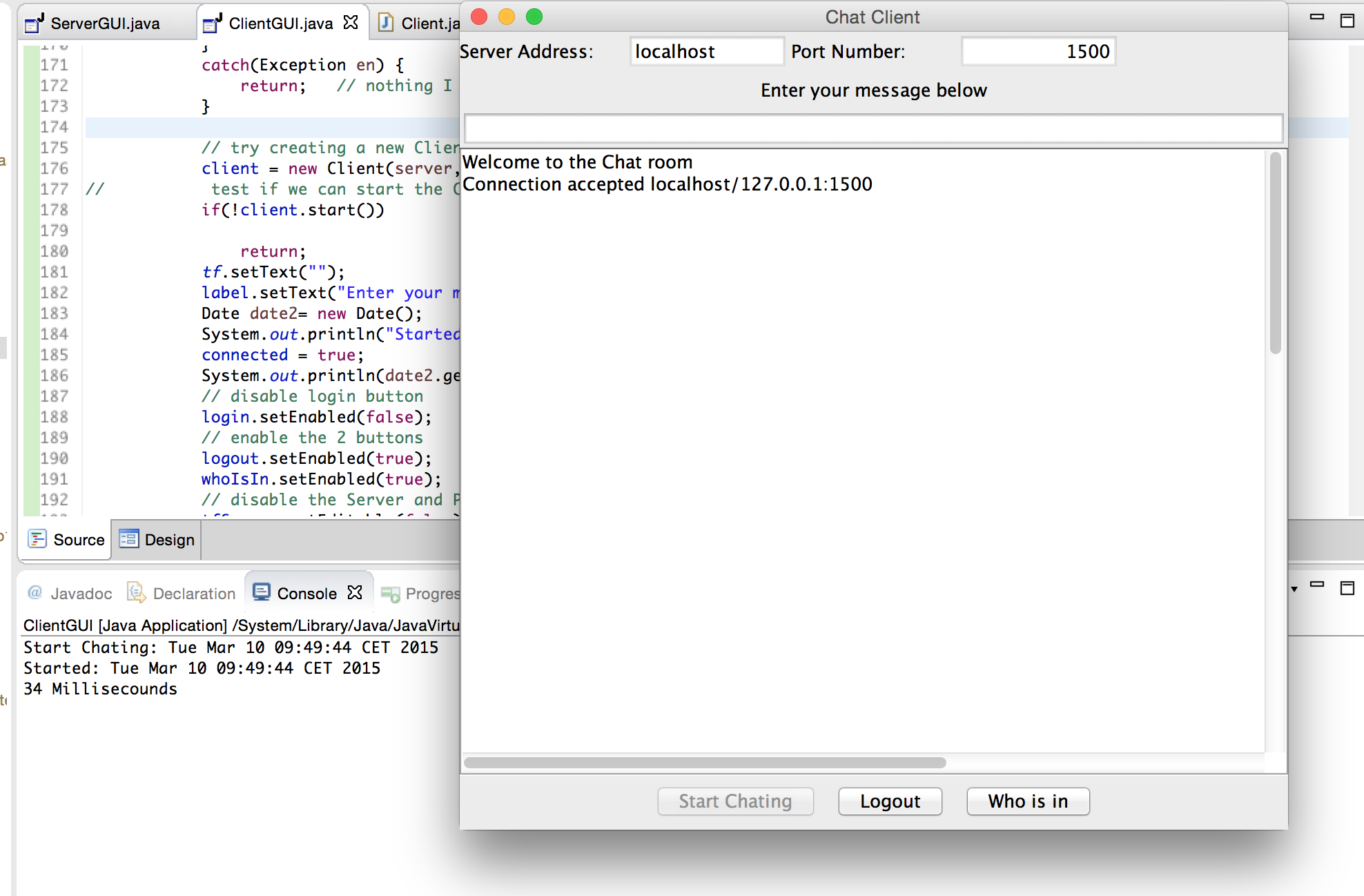
We have used (System.***out***.println()) at the very start when the user click log on, and (System.***out***.println()) when the chatting started, as shown in console in the picture below.



**Load test case Start chatting:**

Showing the time it takes between the user clicking start chatting until the chat starts.

It takes 34 milliseconds.

****

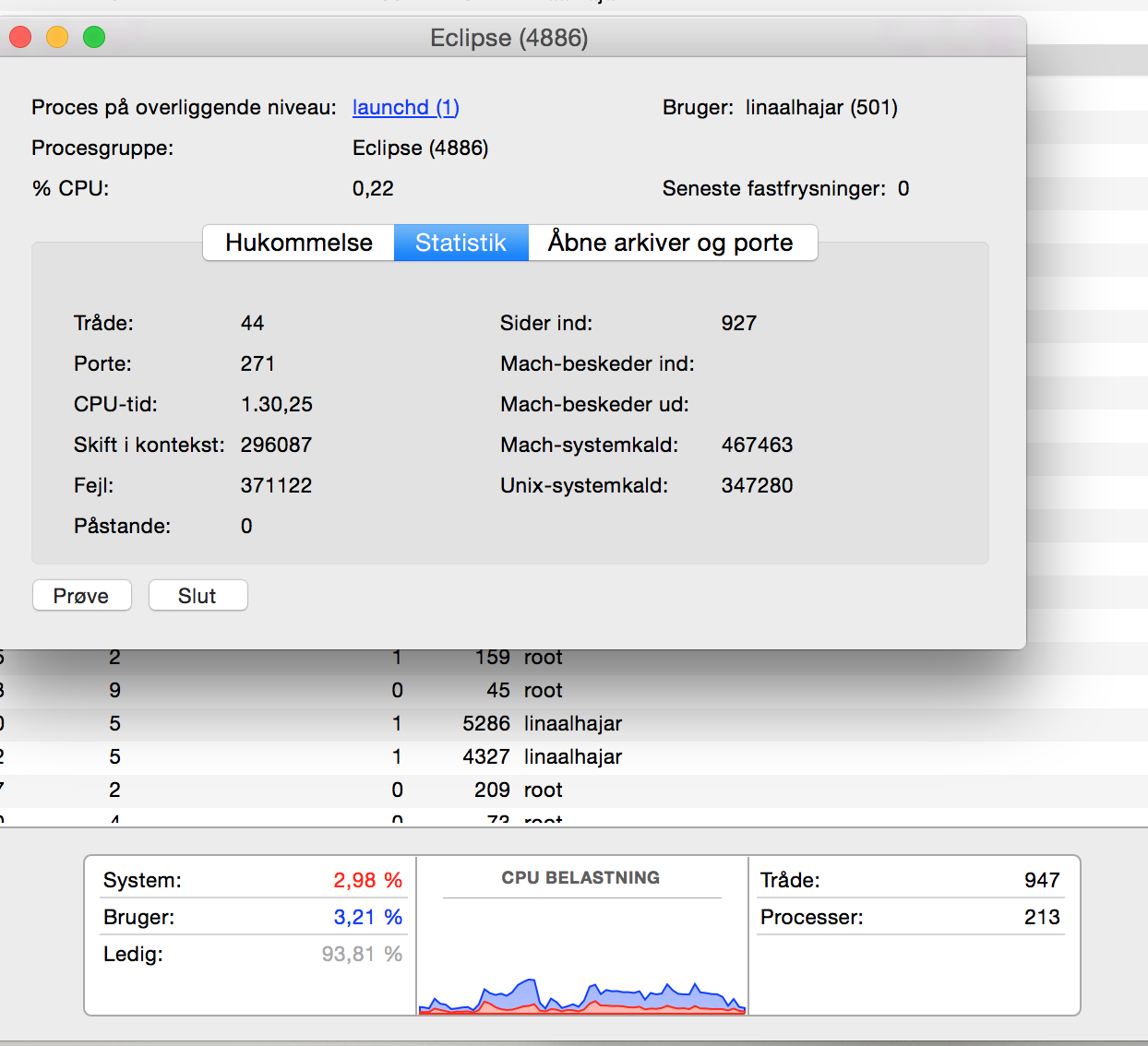
**Stress Test**

Stress testing is the testing to evaluate how system behaves under unfavorable conditions.

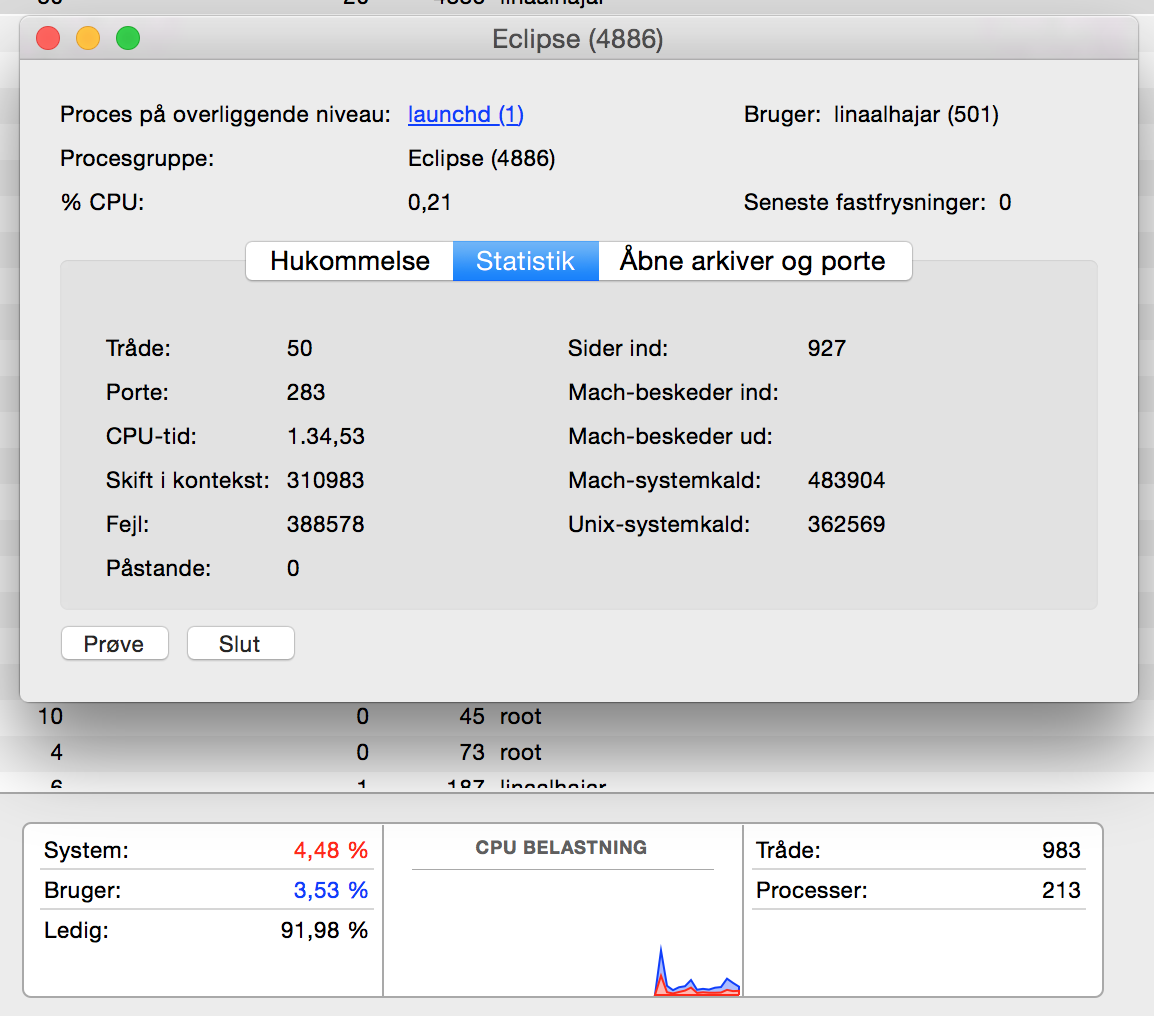
In JMeter testing tool we tried to test our system by specifying the number of transactions and number of threads and the number of users. JMeter is for the web biased systems, and the version that could test our application is not free and would cost us about 1000$.

This is why we have test our program by written System.***out***.println()in the code.

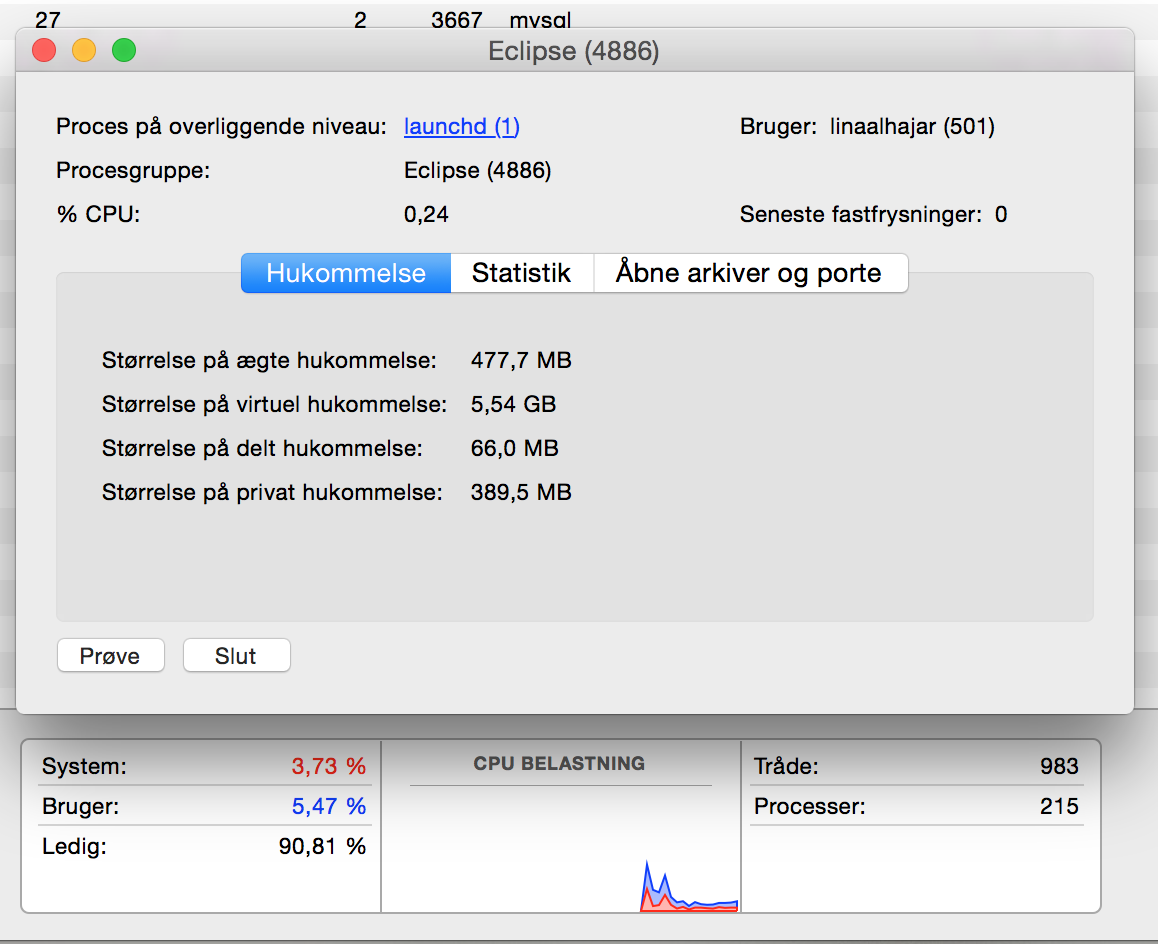
**Screenshots of memory usage:**

1.Lunching the Application

2. Starting the Server



3. Starting the Client



**Spike Test** (how does the system recover from a load burst)

to verify a system's stability during bursts of concurrent user and or system activity to varying degrees of load over varying time periods.

Spike testing should also verify that an application recovers between periods of spike activity.

**Soak Test** (no system degradation over time - e.g. memory utilization)

This test is to test how the system with a significant load extended over a significant amount of time

# 

# 

# **User Acceptance Test**

The Client or the user can sign out of the application at anytime.

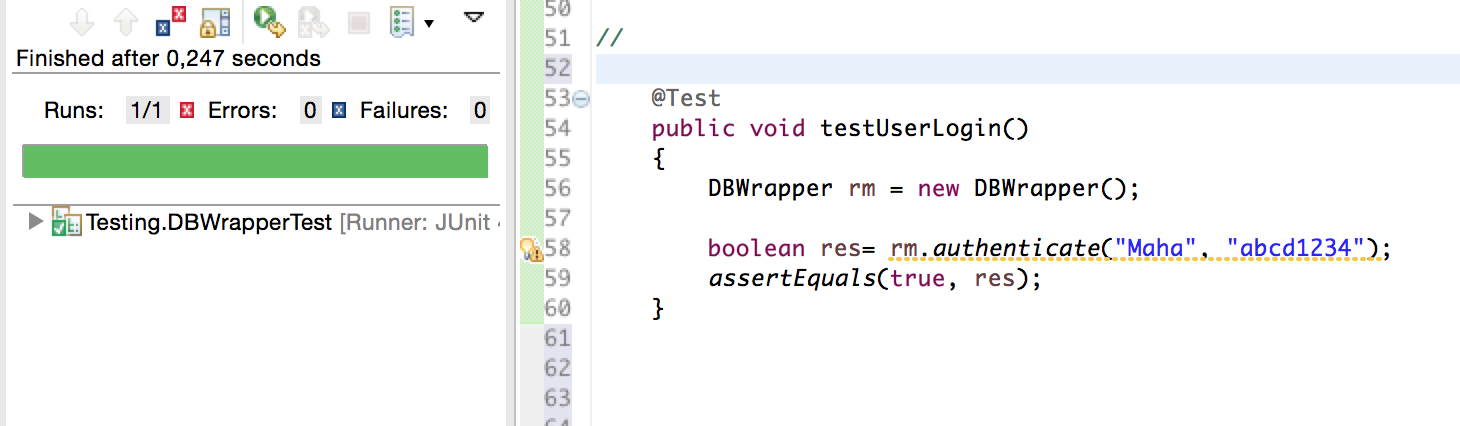
**JUnit Test**

We have used JUnit test to test the functionality required for certain methods of our project to ensure that the code is implemented, and that there are no breaking changes that have not been taken into consideration.

We have tested the following cases:

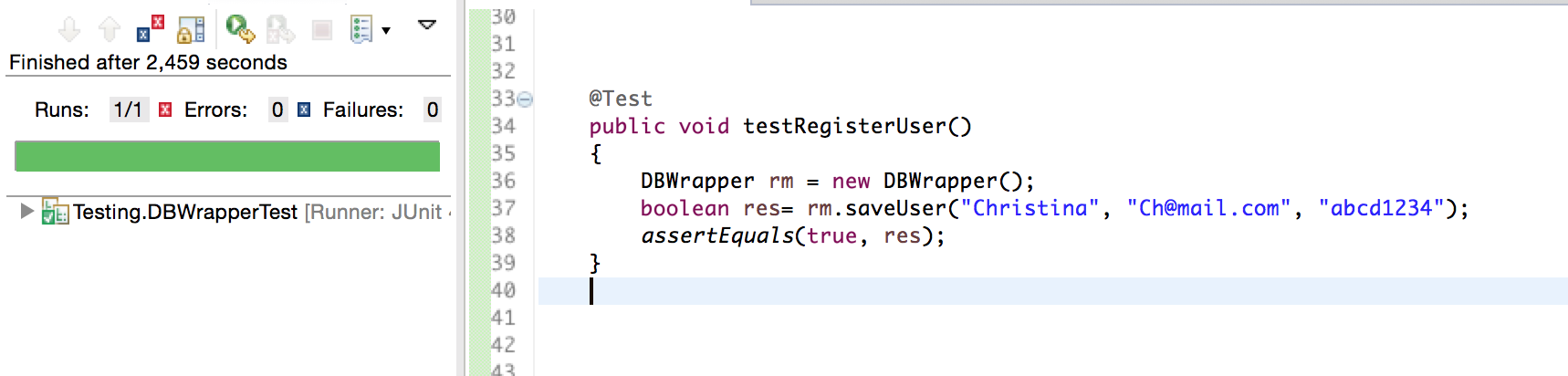
* User log in.

This test is testing if the user name and password is existing in the Mysql Database.



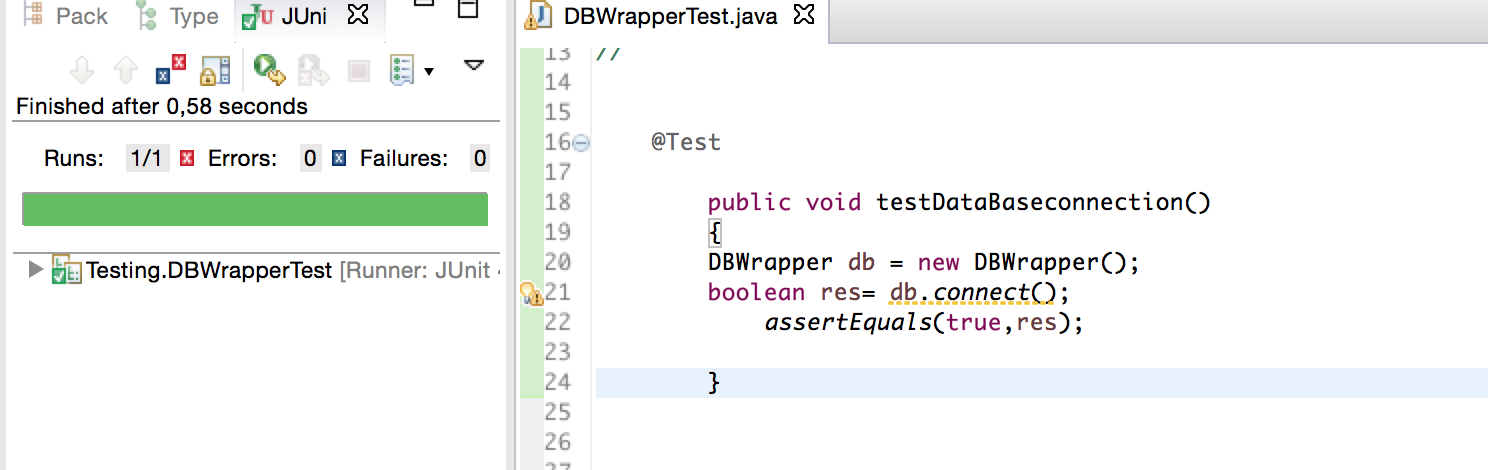
* Save User (Register User )

Testing if the saveUser method is saving the user with the given data.



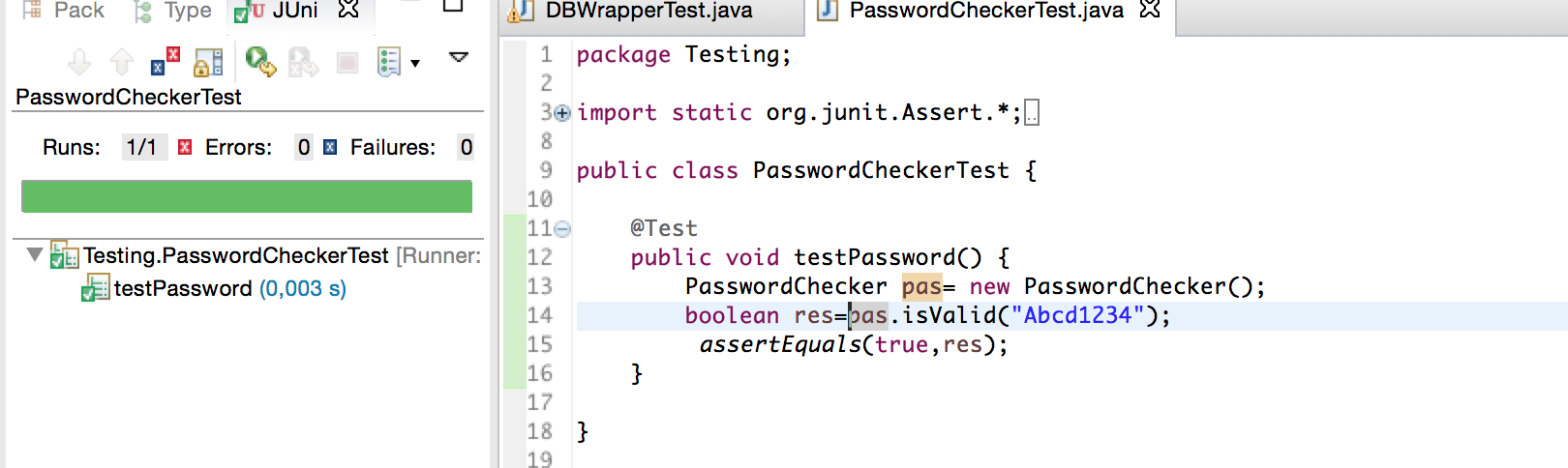
* Database connection.

Testing if the Database connection is established.



* Password validation.

Testing if the password is valid according on the requirements



* Email Validation

We have tested to cases the first one testEmailValidation() where the email is written right and valid, and the second one TestEmailValidation2() is where the email is not valid.