|  |  |
| --- | --- |
| Preliminary Report | [Friday 29th]2009 |
| [Preliminary report and overview of the first three month in ECI] | [Stage ECI] |



Contents

[Contents 2](#_Toc228936411)

[Overview 3](#_Toc228936412)

[First Subject (Tests) 4](#_Toc228936413)

[Overview 4](#_Toc228936414)

[Purpose 4](#_Toc228936415)

[Context 5](#_Toc228936416)

[Environment 5](#_Toc228936417)

[Second Subject (GUI) 6](#_Toc228936418)

[Purpose 7](#_Toc228936419)

[Context 7](#_Toc228936420)

[Environment 8](#_Toc228936421)

[Functionalities 8](#_Toc228936422)

[Subject 1 (Tests) 8](#_Toc228936423)

[Subject 2 (GUI) 9](#_Toc228936424)

[Project Management 11](#_Toc228936425)

[Subject 1 (Tests) 11](#_Toc228936426)

[Priorities 11](#_Toc228936427)

[Limits and Interfaces 12](#_Toc228936428)

[Mean of Control 13](#_Toc228936429)

[Methods an Tools 15](#_Toc228936430)

[Subject 2 (GUI) 16](#_Toc228936431)

[Priorities 17](#_Toc228936432)

[Limits and Interfaces 17](#_Toc228936433)

[Mean of Control 18](#_Toc228936434)

[Methods an Tools 18](#_Toc228936435)

[Planning 19](#_Toc228936436)

[Forecast planning 19](#_Toc228936437)

[Effective planning 20](#_Toc228936438)

[Differences: 20](#_Toc228936439)

[Projection 21](#_Toc228936440)

# Overview

 ECI Telecoms is a leading supplier of networking infrastructure for carrier and service provider networks worldwide. From tier-one national network operators to city carriers, ECI's equipment serves as the platform for key applications including business services, voice, video and wireless back haul.

 They provide flexible networks according to the specific needs of each network operator. This customization and agility are the cornerstones of ECI's relationships with its customers. ECI customers value the company's innovative spirit, efficient solutions and agile attitude in meeting their needs.

 ECI has been at the forefront of telecommunications innovations for decades, from the telephone line doublers (1977), which enabled for the first time two conversations over one phone line, to more recent achievements, including HDSL (1993), optical ring-based networks (2000), toll-quality Voice-over-IP (2002) Coarse Wavelength Division Multiplexing (2003) and 10-Degree Reconfigurable Optical Add-Drop Multiplexer (2007).

 Headquartered in Israel, with a strong global presence around the world, ECI has development centers in India, China, USA and Israel.

 As a member of the research and development department of ECI, I help to prepare the future of the company, by participating in the development of the new versions of the products and by ensuring the good operating of the system.

 More specifically, my scope concerns the main application, that round up the services proposed by ECI. Amongst those services Octopus, Fault management, Hi-Care runs on shelves called MSAN (Multi-Service Access Node). Precisely, a new version of the shelves is expected. In order to exploit the best way the capacities of the servers and their good functioning, two missions were given to me.

 In a first time in order to have an overview the product, I had to prepare tests on the Shelves in a stressed environment.

 In a second time I'm in charge of developing a graphical user interface that allows generating and decoding key license of ECI clients.

 A third part is to be in the near future, but it wasn't planed because of the unexpected progress of my tasks.

## First Subject (Tests)

### Overview

 ECI’s Next-Generation Multi-Service Access Node (MSAN) provides access, aggregation, edge services and capabilities.

 The telecommunications industry is currently experiencing multi-dimensional and multi-layer convergence at all network levels. This includes the application layer, the control layer and the transport layer.

 The ECI MSAN solution introduces an innovative approach for multi-dimensional convergence in the access space by converging access technologies, current and future multi play broadband service support, and residential and business services. ECI Telecom's MSAN supports industry-standard transport-layer convergence, and enables convergence of protocols (ATM, Ethernet/ IP, etc.) and functionality (aggregation, edge, etc.) in a single access node.

 MSAN works in concert with the entire ECI portfolio to provide a complete solution tied together by ECI's 1Net architectural framework



### Purpose

In this internship I have to complete several objectives related to the application and the development of the modules. The main tasks I am to do is:

To ensure that ECI NMS 'N'-tier application meets required performance goals. To manage such task:

 **Project resources:**

 I was part of performance team responsible for:

* 1. Weed out bottlenecks in application;
	2. Identify server and database needs;
	3. Learn if application recovers gracefully.

**Project scope:** I’m responsible for exploring and implementation JMETER Test Plans.

First Part Studying

The first part of my mission was to study **JMeter**. As nobody knew very well, how this product worked, I had to explore the possibilities and find a way to manage with the challenges encountered, such as:

* + Prepare a test that stick to a client behavior.
	+ Be able to verify test results with assertions.
	+ Store data in excel format in order to make statistics.
	+ Launch tests in disturbed way (Several type of client all together.).
	+ Fill log file in a correct way.

 Concurrently, I had to edit a documentation on how using **JMeter** as a tutorial, because, the detailed documentation already existed, but was hard to use for beginners.

Second Part Formation and Tests

 I also had to prepare slides in order to present how using **JMeter** to my team, and to make an introduction course to those I will have to form. First where made course of introduction, and then we worked all together to prepare tests.

### Context

 ECI boasts a comprehensive element management system for economically managing the mass deployment of broadband networks, real-time performance monitoring, analysis and fault management.

 ECI's OPS is a field-proven management system, currently supporting millions of end-customers on mass-deployed triple-play enabled networks providing broadband voice, video and data services.

Before trying JMeter ECI used IRobot, another application for stressed test. Unfortunately the application didn't allow managing all tests needed. That's why ECI decided to move from IRobot to JMeter.

### Environment

**JMeter** is an Apache Jakarta project that can be used as a loading testing for analyzing and measuring the performance of a variety of services, with a focus on web applications. **JMeter** allows recording the behavior of a client by using a HTTP proxy server, to organize your tests using thread groups, loops… and finally verify your hypothesizes by inserting assertions. One of the advantages of **JMeter** is that you can run your tests from a several machines but monitored by one. This single thing, gives you the opportunity to test your application in a stressed environment, which was one of our purposes.

## Second Subject (GUI)

 "A carrier-class broadband access element management system with a wide array of tools, for network topology, fault, configuration and security management and performance monitoring - all through a single, unified GUI; designed to efficiently support the mass deployment of current as well as future broadband services by addressing the varied needs of network technicians."

 OPS works in concert with the entire ECI portfolio to provide a complete solution tied together by ECI's 1Net architectural framework



### Purpose

An additional part of functionality migration from C++ to Java is “Control Center”. The Control Center will be the main window of the ECI NMS and should be a convenient work area for NMS user, i.e., network administrators.

In order to provide to connection we need to give a key License to the customers, as a guarantee for the customers have the connection they paid for.

This key depends on how are filled different fields. That’s why it needs to be encrypted, so clients won’t change their own key to extend the expiry date for instance.

The application is a part of three different modules.

### Context

The management system managing ECI access products is moving from C++ and MFC technology to JAVA and Web service technology.  This system allows configuration, fault management, performance monitoring and maintenance of various shelves, cards and communication protocols. Switching technologies needs to be progressive since new cards and new features are continuously required in order to meet customers’ expectations.

* **Project Resources:** I am part of a team developing “Control Center” application using Swing.
* **Project Scope:** I am responsible for developing different components of “Control Center”.

### Environment

One of the goals of moving to JAVA and WEB technologies is to be able to have a system that will support thousands of managed elements. The system has already been enhanced with JAVA modules:

* + **Fault Management** that reports all current and historical alarms in a user’s friendly way.
	+ **Hi-Care**, that provides user’s focus management.
	+ **Octopus** that allows software downloads simultaneously to several network elements.

# Functionalities

## Subject 1 (Tests)

The tests, were supposed to simulate the following scenarios:

**In the “Fault Management” module:**

1. **Scenario 1:**
2. Pressing the refresh button every thirty second,
3. Then verify the loading of the alarms problems.
4. **Scenario 2:**
5. Pressing “make search” button,
6. Fill the NEs name (shelves name),
7. And verify that correct cards are selected.
8. **Scenario 3:**
9. Pressing “make search” button,
10. Fill the NEs name (shelves name),
11. Specify a card number.
12. And verify that correct cards are selected.
13. **Scenario 4:**
14. Pressing “make search” button,
15. Fill the Path name,
16. Specify a card number.
17. And verify that correct cards are selected.
* **In the “Hi-care” Module:**
1. **Scenario 5 :**
2. Filling the port, shelf and card fields,
3. Then press “load” button,
4. And eventually verify the data device.
5. **Scenario 6:**
6. Filling the port, shelf and card fields,
7. Then press “load” button,
8. And restart the process with new port, shelf and card fields.

All this scenarios had to be run concurrently a huge number of times.

## Subject 2 (GUI)

The second subject is to develop a GUI and program functionalities in order to encrypt and to decrypt License key. More specifically the functionalities are the following ones:

* **Generation of key** from the parameters entered by the user in the Graphical interface using appropriated fields.
* **Printing** **the encrypted key** into a file, so the profile of a user can be remembered.
* **Printing the key and all the parameters** in accordance with a specific format (to be read by another module).
* **Generation of the parameters** of a user, by two ways:
* By getting the encrypted key from a file.
* By writing it in “License Key” text field area.



Here is the GUI in its final version

# Project Management

## Subject 1 (Tests)

### Priorities

**Level 1:** The first of the priorities was to verify that JMeter was able to do what IRobot couldn't. This step was needed to understand how basically work the project in order to avoid any waste of time. (If JMeter was unfortunately not adapted for our needs.)

**Level 1B:** Is it possible to overcome JMeter weaknesses in some way to reach the needed capabilities? (Only if the first step is not passed).

**Level 2:** Once we do know that can manage the tests. Find rapidly how to do it, in order to form the technicians, who will be in charge of the tests.

**Level 3:** Edit the tests, and make the documentation, for ECI.

### Limits and Interfaces

**Limits:**

This task doesn't include the following of the test. I can however be called at any time if my help is needed, for other formations course or specific cases.

**Interfaces:**

The computers are linked to the shelves by the Intranet network of the company. As the amount of threads run by JMeter is too high to be manage by only one machine, we had to use a distributed testing plan.

The Picture below represents the Architecture of the Tests.



Where:

* *Master* –Is the system running JMeter, which controls the tests.
* *Slave* – Is the system running JMeter-server, which takes commands from the GUI and sends requests to the target system(s).
* *Target* – Are the shelves we plan to stress test.

### Mean of Control

The main mean of control are assertions provided by JMeter itself.

Once you have well organized your tests, you need for them to be helpful, to know if the responses expected occurred well. To do so, JMeter provided for Assertion. Assertion are hypothesizes that you make about the result. If the result doesn’t match with what you expected, a failure will pop up (nothing in the other case). In your result tree



 Here are the main Assertions needed. There are some more, but you will need it quite rarely. The most important ones are:

* + **Response assertions:** Which check the content of the file returned by the server using pattern matching. For example you can check if the response code is 200 as expected.



* + **Size Assertions:** Which check the size of the returned file from the server, to recognize the file.



* + **Xpath Assertions:** To verify the presence of an element in the file returned by the server using the DOM. To check if a table of content was correctly filled after the php requests for instance.



### Methods an Tools

Two problems occurred with JMeter. Indeed, it by itself didn't cover all our expectancies.

**BadBoy:**

First JMeter can't record HTTPS request using the HTTP proxy which is used to record the behaviors of users, facing the application. To do so, I downloaded and install badboy 1.0.6 on the machine in order to record correctly all the request. I didn't need to get the last version of this product. Actually a former version was enough and above all free.



After recording the tests badboy allows to export the file to be edited by JMeter, as shown below.



**Excel:**

In order to make statistics treatments on the data harvested, we use the software Excel 2009.

## Subject 2 (GUI)

In order to provide to connection we need to give a key License to the customers, as a guarantee for the customers have the connection they paid for.

This key depends on how are filled different fields. That’s why it needs to be encrypted, so clients won’t change their key to extend the expiry date for instance.

The application is a part of three different modules.

### Priorities

**Level 1:** Make the key Class attributes following functionalities run:

* + Encryption/Decryption of the key
	+ Printing the encrypted key into a file, so the profile of a user, can be remembered.
	+ Printing the key and all the parameters in accordance with a specific format (to be read by another module).

Indeed those functionalities allow the other modules to be tested even if this one is not finished.

**Level 2:** Terminate the programming of the GUI with all the functionalities, and tests.

**Level 3:** Make the code review and the documentation.

### Limits and Interfaces

1. Even if I am supposed to make the interface and all its functionalities, I am also due to provide for some other ones in order to make the connection between the other modules. Such as make functionalities to write in a file all the parameters by reading a key, read such a file.
2. Concerning the encryption and decryption I don't have to find by myself the algorithm. I only have to use an API. Still I have to make the conversion from the format of the key structure to the encryption functions.
3. I also have to make the structure of the key and imagine by myself.

### Mean of Control

JUnit will be the tool used to verify if the objectives where reached correctly

### Methods an Tools

**Tools:**

I used Eclipse 3.2 to develop the application.



In order to create the graphical user interface, I used Eclipse Plugin “visual Editor” to elaborate graphically the user interface.



The look and feel of the interface was created by the company, that’s the reason why it looks different from the standards

JUnit will be used to test the application.

Microsoft Word 2009 to make documentation together with Gant project to realize the planning diagrams.

# Planning

The best way to understand the evolution of the task in ECI is by starting comparing the forecast and effective planning.

## Forecast planning



Here in Israel, I don't exactly work in terms of project, but in missions. This explains the way my work plan is scheduled. We can split in three missions, the internship. The first part is about JMeter, The second part developing the key License manager in the ECI product application, and the last part still unknown will be a module of the application. The module will depend on the time that is going to left.

## Effective planning



### Differences:

As we consider the two planning, we can notice some differences between them. At the beginning I thought that writing documentation for ECI would be enough, but after more consideration we took the opportunity of editing the test to form the technicians of the team so they could be able to overcome the main problems they could encounter. To do so the lecture turned into a course, and I, was to supervise the way the test would be done.

This explains to some extent, the change in the schedule but we have also to consider that things went more quickly as we believed. Therefore the programming of GUI of the SRS EOPS was deported to May the 6th. As things are heading, the development of the first module progresses fast. So a new mission is to be added in the last 2 months of the internship. Even if the subject it is not fixed yet, a new module of the application is expected.

# Projection

As the calendar had to be revised, we can expect a new mission concerning the ECI application. But for now, it is important to consider the last and the current missions and have a look about what it implies for me, in a near future and at long term.

Concerning the first mission, the experiencing collected gave me a JMeter expert status, which means that I could be in charge of a group, if we had to go over again on the subject in another mission, or that I could be sent as a consultant in case of need.

 Concerning the current mission, the code review shown various point to revise in order to make the code more easy to read and maintainable. I am so asked to fulfill the check list we made during the last week. I'm also due to prepare the documentation of the work done this is the Java documentation, but also prepare UML diagrams such as class diagrams and state diagrams. Eclipse Plugins will be useful to generate them however.

Then we will pass to the integration of the module step, indeed the other modules are already made.

After all that, as previously said, I will pass to another mission concerning the ECI application.