**SERVICE ENCRYPTION AND SCHEDULING**

**USING CLOUD COMPUTING**

A MINI PROJECT REPORT

***Submitted by***

**ROSHAN ZAMEER ALI S 1605100**

**SHAENMUGAPRIYAN S 1605104**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**



**COIMBATORE INSTITUTE OF TECHNOLOGY**

***(Government Aided Autonomous Institution Affiliated to Anna University)***

**COIMBATORE-641 014**

**ANNA UNIVERSITY** - **CHENNAI 600 025**

**APRIL 2017**

**COIMBATORE INSTITUTE OF TECHNOLOGY**

***(A Govt. Aided Autonomous Institution Affiliated to Anna University)***

**COIMBATORE – 641 014**

**BONAFIDE CERTIFICATE**

Certified that this mini project **“SERVICE ENCRYPTION AND SCHEDULING”** is the bonafide work of  **ROSHAN ZAMEER ALI .S (1605100), SHAENMUGAPRIYAN .S (1605104)** under my supervision during the academic year 2018-2019.

|  |  |  |
| --- | --- | --- |
| **Dr.G.Kousalya M.E., Ph.D .,** |  | **Mr.G. Karthikeyan M.Tech.,** |
| **HEAD OF THE DEPARTMENT,** |  | **SUPERVISIOR,** |
| Department of CSE, |  | Department of CSE & IT, |
| Coimbatore Institute of Technology, |  | Coimbatore Institute of Technology, |
| Coimbatore – 641 014. |  | Coimbatore – 641 014. |

Certified that the candidates were examined by us in the project work viva-voce

examination held on …………………

**Internal Examiner External Examiner**

Place:

Date:

**ACKNOWLEDGEMENT**

**ACKNOWLEDGEMENT**

We express our sincere thanks to our Secretary **Dr.R.Prabhakar** and our Principal **Dr.V.Selladurai** for providing us a great opportunity to carry out our work. The following words are a small part to express our gratitude to them. This work is the outcome of their inspiration and product of plethora of their knowledge and rich experience.

We record the deep sense of gratefulness to **Dr.G.Kousalya**, Head of the Department of Information Technology, for his encouragement during this tenure.

We equally tender our sincere thankfulness to our project guide **Dr.G.Kousalya**,

Assistant Professor, Department of Computer Science and Engineering & Information Technology, for his valuable suggestions and guidance during this course.

During the entire period of study, the entire staff members of the Department of Computer Science and Engineering & Information Technology have offered ungrudging help. It is also a great pleasure to acknowledge the unfailing help we have received from our friends.

It is matter of great pleasure to thank our parents and family members for their constant support and co-operation in the pursuit of this Endeavour.

**ABSTRACT**

**ABSTRACT**

 By using the Web as an application platform, an enterprise can quickly and easily deploy new systems in response to market changes as well as reap the savings of using a single, standard platform. Recently, there has been a tremendous growth in the number of ASPs.Organizations of the twenty‐first century are going to be Web‐centric and focus their objectives on agility and cost savings. Pressure to reduce overhead in information technology (IT) pushes these organizations to consider outsourcing. This growth in ASPs has raised several issues and challenges that need to be resolved to realize the potential benefits of ASPs.

Due to the various types of users, the size of jobs among which the server is providing service may vary. One user may carry out a job which would take more time and other user may take up a job which may take lesser time. We developed an ASP which effectively does service encryption which provides security as a primary parameter and schedules the services according to the number of requests and time consumption. The scheduling of content delivery helps in dynamic creation of cloudlets to serve the users.

**TABLE OF CONTENTS**

# **CHAPTER NO. TITLE PAGE NO**

# **ACKNOWLEDGEMENT i**

# **ABSTRACT ii**

### LIST OF FIGURES iv

**LIST OF ABBREVIATION v**

1. **INTRODUCTION 1**

1.1 CLOUDSERVICE 1

1.2 APPLICATION SERVICE PROVIDER 2

1.3 SERVICE ENCRYPTION

1.4 SCHEDULING

1. **LITERATURE SURVEY 4**
2. **SYSTEM SPECIFICATION 6**

3.1 SOFTWARE SPECIFICATION 6

3.2 HARDWARE SPECIFICATION 6

1. **SYSTEM ANALYSIS 7**

4.1 EXISTING SYSTEM 7

4.2 PROPOSED SYSTEM

4.3 SYSTEM ARCHITECTURE 8

1. **SYSTEM IMPLEMENTATION** 9

6 **CONCLUSION AND FUTURE WORK**

**7 REFERENCES 18**

iii

**LIST OF FIGURES**

**FIGURE NO. TITLE PAGE NO.**

**1** DATA SET 3

**2** REMOVING OUTLIERS 11

**3** DATA TRANSFORMATION 14

**4** SPLITTING DATASET 15

**5** TRAINING MODEL 15 **6** PREDICTION 16

iv

##### LIST OF ABBREVIATIONS

ASP Application Service Provider

AES Advanced Encryption Standard

v

**INTRODUCTION**

**CHAPTER 1**

**INTRODUCTION**

**1.1 CLOUDSERVICE:**

A cloud service is any service made available to users on demand via the Internet from a [cloud computing](https://www.webopedia.com/TERM/C/cloud_computing.html) provider's servers as opposed to being provided from a company's own [on-premises](https://www.webopedia.com/TERM/O/on-premises.html)servers. Cloud services are designed to provide easy, scalable access to applications, resources and services, and are fully managed by a [cloud](https://www.webopedia.com/TERM/C/cloud.html) services provider.

**Cloud Services are Dynamic and Scale:**

A cloud service can dynamically scale to meet the needs of its users, and because the service provider supplies the hardware and software necessary for the service, there’s no need for a company to provision or deploy its own resources or allocate IT staff to manage the service.  Examples of cloud services include online data storage and backup solutions, Web-based e-mail services, hosted office suites and document collaboration services, database processing, managed technical support services and more.

**1.2 APPLICATION SERVICE PROVIDER:**

Computer hardware produces computing resources in forms including processor time, data, and data access bandwidth. Application software consumes these resources and delivers services to users. To users, computers and applications are only means to get computing services.

The separation of resources and application from service can find analogies in public utilities. For electricity, the electrical power is the resource, and light bulbs and

electrical heaters, which are examples of electrical application devices, transform the electrical resource into light and heat. Comparably, for telecommunications, the

virtual channels are the resource, and telephones and fax machines, which are examples of telecommunications application devices, use this resource to deliver phone-call and faxing services. Similar analogies can also be made for computing based on the Internet. For instance, Internet bandwidth as well ashardware and data on web and application servers are resources; server software and server-based software

embedded in web pages are applications. Users can get services by accessing server data or running the server applications, and using the Internet as an “extension cord”

between the server machines and the user I/O devices. Since processor time cannot be easily delivered on a network, the computing applications are usually hosted on

servers. Due to technological limitations, the computing industry has so far mainly adopted the commercial model of selling computers, which are computing resource generators, and licenses of applications. The users are responsible for the installation and maintenance of computing infrastructure and applications, which may constitute significant overhead on the users. For example, such overhead may exclude

small and medium-sized enterprises from the benefits of advanced enterprise management services. This model also leads to very low utilization of office and personal computing resources and applications. On the contrast, the utility industries have long adopted the commercial model of selling resources or services. The resources or services are delivered by comprehensive delivering infrastructures to the users. Users pay by a fixed subscription fee, or by actual usage of the resources or

services. For electricity, water, and gas, the corresponding industries sell resources only. For telecommunications industry, in addition to the basic bandwidth resource, value added services, such as directory service, caller-ID service, and call-waiting service, are becoming important components of the services sold to the users. This

commercial model enjoys sharing of resources and amortization of infrastructure cost, thus achieving economies of scale; expertise pooling; value-added services; rich experience in service quality level control; rich cost models; and standardization of the resources and main services. For the computing industry to benefit from the model of selling services, some basic technological and infrastructure hurdles must be overcome. First, the computing resources must be divisible, thus the processor time from a single computer can be shared by large number of users. Second, there should be a comprehensive delivering system for services. Third, the applications must be designed to support server hosting.

**1.3 AES Encryption:**

The Advanced Encryption Standard, or AES, is a symmetric [block cipher](https://searchsecurity.techtarget.com/definition/block-cipher) chosen by the U.S. government to protect classified information and is implemented in software and hardware throughout the world to encrypt sensitive data.

The [National Institute of Standards and Technology (NIST)](https://searchsoftwarequality.techtarget.com/definition/NIST) started development of AES in 1997 when it announced the need for a successor algorithm for the [Data Encryption Standard (DES)](https://searchsecurity.techtarget.com/definition/Data-Encryption-Standard), which was starting to become vulnerable to [brute-force attacks](https://searchsecurity.techtarget.com/definition/brute-force-cracking).

This new, advanced encryption [algorithm](https://whatis.techtarget.com/definition/algorithm) would be unclassified and had to be "capable of protecting sensitive government information well into the next century," according to the NIST announcement of the process for development of an advanced [encryption](https://searchsecurity.techtarget.com/definition/encryption) standard algorithm. It was intended to be easy to implement in hardware and software, as well as in restricted environments (for example, in a [smart card](https://searchsecurity.techtarget.com/definition/smart-card)) and offer good defenses against various attack techniques.

**AES features:**

The selection process for this new [symmetric key algorithm](https://searchsecurity.techtarget.com/definition/secret-key-algorithm) was fully open to public scrutiny and comment; this ensured a thorough, transparent analysis of the designs submitted.

NIST specified the new advanced encryption standard algorithm must be a block cipher capable of handling 128 bit blocks, using keys sized at 128, 192, and 256 bits; other criteria for being chosen as the next advanced encryption standard algorithm included:

**Security:**

Competing algorithms were to be judged on their ability to resist attack, as compared to other submitted [ciphers](https://searchsecurity.techtarget.com/definition/cipher), though security strength was to be considered the most important factor in the competition.

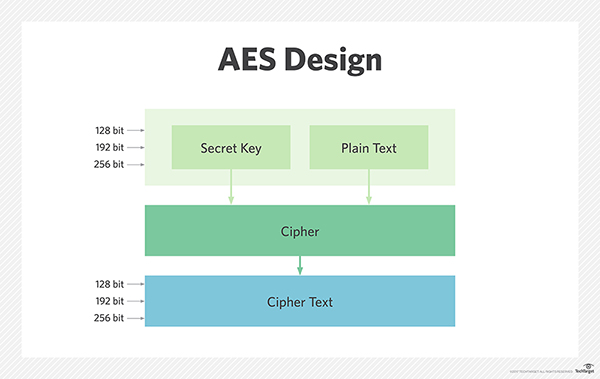
**Cost:**:

Intended to be released under a global, nonexclusive and royalty-free basis, the candidate algorithms were to be evaluated on computational and memory efficiency.

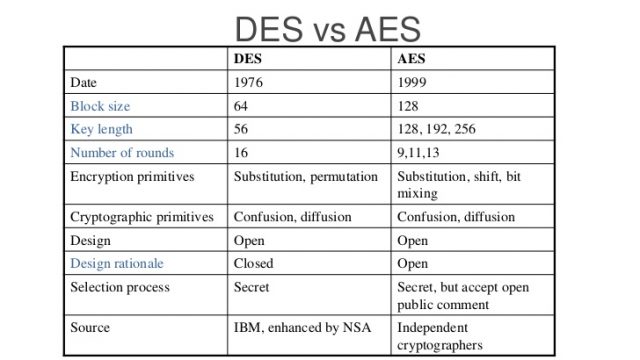
**Implementation:**

Algorithm and implementation characteristics to be evaluated included the flexibility of the algorithm; suitability of the algorithm to be implemented in hardware or software; and overall, relative simplicity of implementation.

**AES DESIGN:**

****

**AES Vs DES:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_LITERATURE SURVEY**

**LITERATURE SURVEY**

**2.1 PacketCloud: A Cloudlet-Based Open Platform for In-Network Services**

**Yang Chen, Senior Member, IEEE, Yu Chen, Qiang Cao, and Xiaowei Yang:**

Described the challenges of the traffic occuring during service provision. Gave us an idea about the concept of creating cloudlet ata n instance to service various requests. THE Internet was designed with the end-to-end principle where the network layer offered merely the best-effort forwarding function. Many important communication functions such as reliability and state management are placed at the end points [1]. This design provides low complexity and high robustness at network routers, and is necessary to meet the technology constraints of the 1970s when the Internet was launched. However, nowadays the Internet connectivity has become a commodity, users and applications increasingly demand new in-network services. Deploying in-network services can either enhance the user experience while surfing the Internet, or optimize the network traffic. Prospective in-network services include host mobility support [2], efficient delivery for frequently accessed content [3], on-path storage [4], or malicious traffic filtering [5]. To host in-network services, middleboxes [6] are widely used. In today’s Internet, ISPs have deployed a number of middleboxes such as content caches, firewalls, and load balancers. We aim to make the Internet infrastructure as an open platform to host in-network services, and avoid the inflexibility of using middleboxes. We propose an architectural solution, called PacketCloud. A cloudlet is a general-purpose cluster consisting of a set of commodity servers, and it can host in-network services. ISPs can choose suitable Point of Presences (PoPs) to deploy cloudlets. In each cloudlet, commodity servers can be efficiently shared among different services, and accordingly an elastic

resource allocation can be achieved. PacketCloud supports Linux-based general-purpose services, and it is open to third-party service providers, including content providers and application providers. PacketCloud can host a number of viable in-network functions, such as fast content fetching, privacy preserving, and energy saving for mobile devices.

**2.2 Application Service Provider Model: Perspectives and Challenges**

**Lixin Tao , Member IEEE and ACM**

Discusses the challenges faced by the Application Service Provider. Provided clarity about the problem statement and gave us a route to follow-on to meet the solution to the problem. With the advance of Internet technology and network bandwidth, Application Service Provider (ASP) mode of computing is becoming a very attractive target of the next wave of Internet revolution. However, the ASP model and its implementation technologies have not been thoroughly investigated. In this paper we highlight the essence, benefits, and importance of the ASP model as the first form of commercial service-based computing, and the interplay of the ASP model and component technologies. We survey the main supporting technologies for ASP, identify major challenges to the

development of ASP applications, and propose solution approaches. We predict that the ASP model coupled with service standardization will lead to networked economy in which cooperative and specialized computing will be realized at the service level.

Computer hardware produces computing *resources* in forms including processor time, data, and data access bandwidth. *Application* software consumes these resources and delivers *services* to users. To users, computers and applications are only means to get computing services. The separation of resources and application from service

can find analogies in public utilities. For electricity, the electrical power is the resource, and light bulbs and`electrical heaters, which are examples of electrical

application devices, transform the electrical resource into light and heat. Comparably, for telecommunications, the virtual channels are the resource, and telephones and fax

machines, which are examples of telecommunications application devices, use this resource to deliver phone-call and faxing services. Similar analogies can also be made for computing based on the Internet. Users can get services by accessing server data or running the server applications, and using the Internet as an “extension cord”

between the server machines and the user I/O devices. Since processor time cannot be easily delivered on a network, the computing applications are usually hosted on

servers. Due to technological limitations, the computing industry has so far mainly adopted the commercial model of selling computers, which are computing resource generators, and licenses of applications. The users are responsible for the installation and maintenance of computing infrastructure and applications, which may constitute significant overhead on the users. For example, such overhead may exclude small and medium-sized enterprises from the benefits of advanced enterprise management services. This model also leads to very low utilization of office and personal

computing resources and applications. On the contrast, the utility industries have long adopted the commercial model of selling resources or services. The resources or services are delivered by comprehensive delivering infrastructures to the users. Users pay by a fixed subscription fee, or by actual usage of the resources or services. For electricity, water, and gas, the corresponding industries sell resources only. For telecommunications industry, in addition to the basic bandwidth resource, *valueadded*

*services*, such as directory service, caller-ID service, and call-waiting service, are becoming important components of the services sold to the users. This

commercial model enjoys sharing of resources and amortization of infrastructure cost, thus achieving economies of scale; expertise pooling; value-added services; rich experience in service quality level control; rich cost models; and standardization of the resources and main services.

For the computing industry to benefit from the model of selling services, some basic technological and infrastructure hurdles must be overcome. First, the computing resources must be divisible, thus the processor time from a single computer can be shared by large number of users. Second, there should be a comprehensive delivering system for services. Third, the applications must be designed to support server hosting.

**2.3 Modeling and Simulation of Scalable Cloud Computing Environments and**

**the CloudSim Toolkit: Challenges and Opportunities**

**Rajkumar Buyya1, Rajiv Ranjan2 and Rodrigo N. Calheiros1,3**

This paper paved a way for the implementation of cloudlets using java. This gave us an idea of how cloudlets work and their architecture. Cloud computing delivers infrastructure, platform, and software as services, which are made available as

subscription-based services in a pay-as-you-go model to consumers. These services in industry are respectively referred to as Infrastructure as a Service (IaaS), Platform as

a Service (PaaS), and Software as a Service (SaaS). The importance of these services is highlighted in a recent report from Berkeley as: “Cloud computing, the long-held

dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service” [11]. Clouds [10] aim to power the next generation data centers by exposing them as a network of virtual services (hardware, database, user-interface, application logic) so that users are able to access and deploy applications from anywhere in the world on demand at competitive costs depending on users QoS (Quality of Service) requirements [1]. Developers with innovative ideas for new services are no longer required to make large capital outlays in the hardware and software infrastructures to deploy their services or human expense to operate it [11]. It offers significant benefit to IT companies by freeing them from the low level task of setting up basic hardware and software infrastructures and thus enabling more focus on innovation and creation of business values. Some of the traditional and emerging Cloud-based applications include social networking, web hosting, content delivery, and real time instrumented data processing. Each of these application types has different composition, configuration, and deployment requirements.Quantifying the performance scheduling and allocation policies in a real Cloud environment for different application and service models under different conditions is extremely challenging because: (i) Clouds exhibit varying demand, supply patterns, and system size; and (ii) users have heterogenous and competing QoS requirements. The use of real infrastructures such as Amazon EC2, limits the experiments to the scale of the infrastructure, and makes the reproduction of results an extremely difficult undertaking. The main reason for this being the conditions prevailing in the Internet-based environments are beyond

the control of developers of resource allocation and application scheduling algorithms.

## SYSTEM SPECIFICATION

**CHAPTER 3**

**SYSTEM SPECIFICATION**

### 3.1 SOFTWARE SPECIFICATION:

* IDE : NetBeans
* Coding Language : Java

### 3.2 HARDWARE SPECIFICATION:

* Operating system : Windows 10
* Processor : AMD eVision Dual Core Processor
* Hard disk : 320GB
* RAM : 4GB(minimum)

6

**SYSTEM ANALYSIS**

**CHAPTER 4**

**SYSTEM ANALYSIS**

##### EXISTING SYSTEM

**3.1 EXISTING SYSTEM:**

The existing works for this problem include:

1. Ability to Support minimal number of users.
2. Activation of all services at once related with the server.

**DISADVANTAGES:**

1. One user may carry out a Job which would take more time and space while other user could take up a very small job.
2. Cost sharing, economies of scale, etc. were not accessible to ASP providers
3. Security to Services were less efficient.

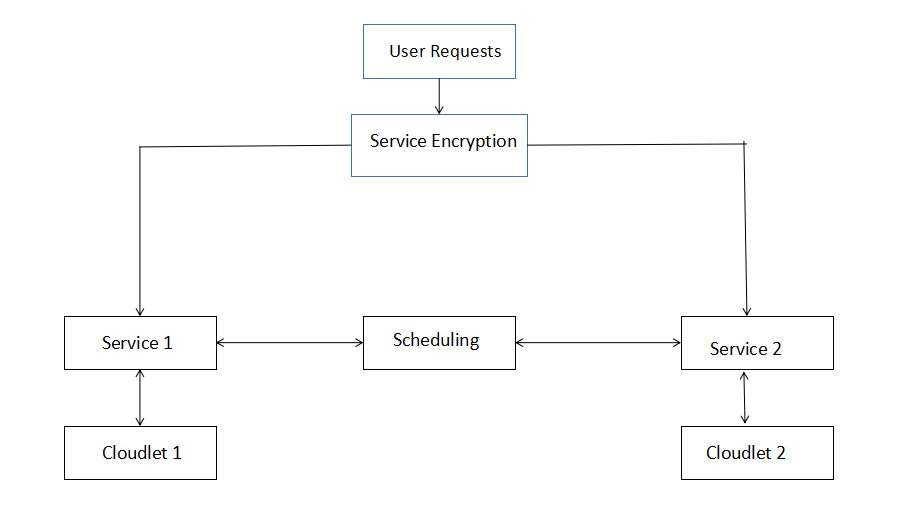
**4.2 PROPOSED SYSTEM**

The proposed system provides efficient Service provision using Scheduling as well as Security to services through AES encryption and also providesDynamic creation of cloudlets for the respective services.

**ADVANTAGES:**

The proposed system considers number of requests for a particular service as well as the location of the request and schedules the provision. It also provides instance-based security to the service everytime it is used so the maintainence team could track and learn the progression.

**SYSTEM ARCHITECTURE:**

****

**SYSTEM IMPLEMENTATION**

**CHAPTER 5**

**SYSTEM IMPLEMENTATION**

**SERVICE 1:**

For demonstrating a service with less space requirement, we took a basic calculator.

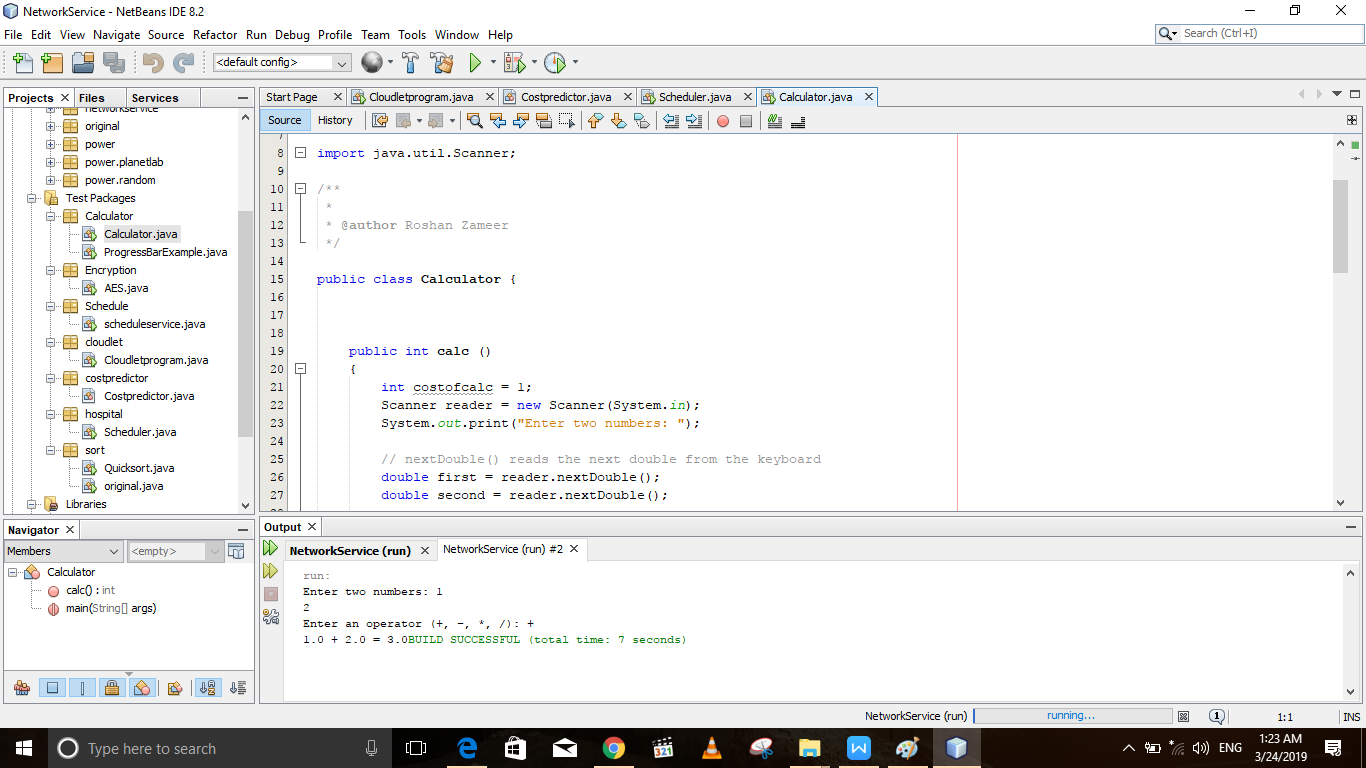
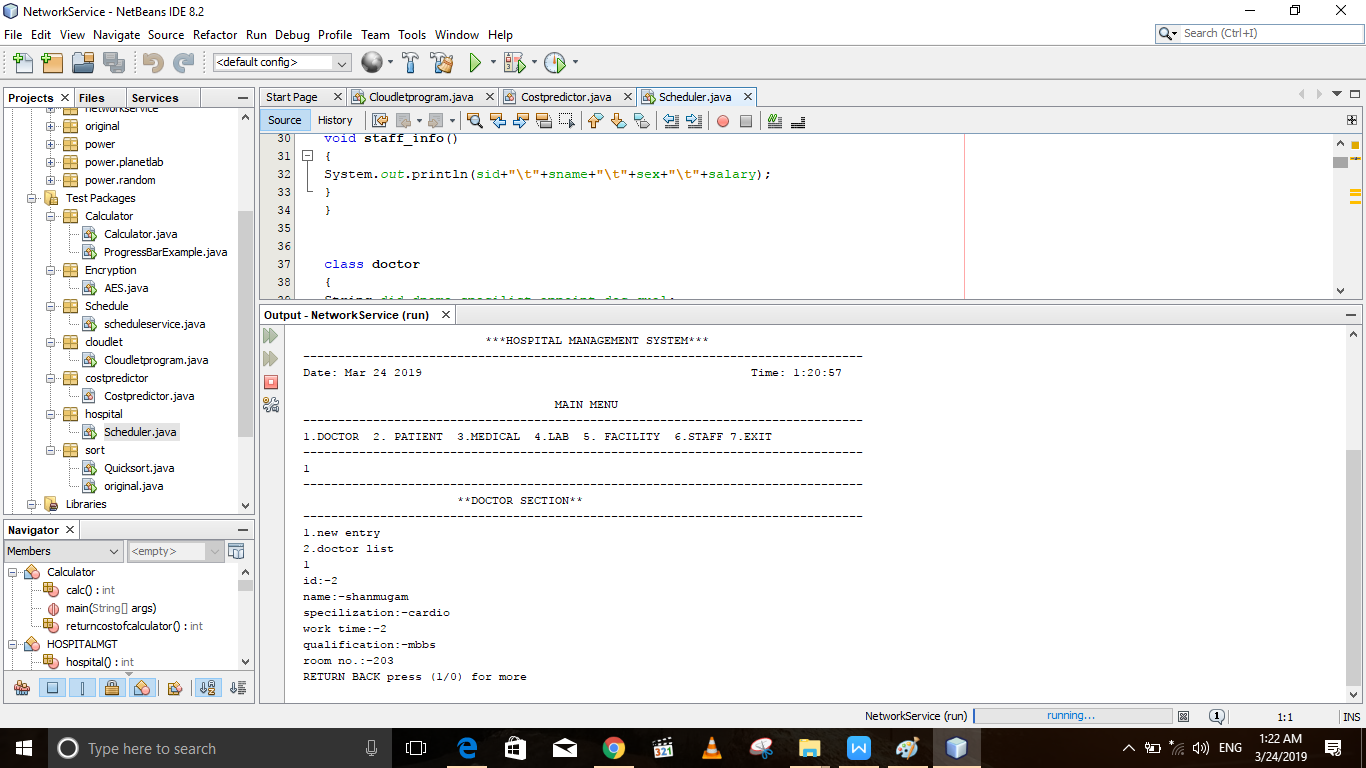


FIGURE 1:CALCULATOR

**SERVICE 2:**

To demonstrate a large cost service, We created a Hospital management system which manages the data of the doctor, patient, and staffs of a particular hospital.

****

**AES ENCRYPTION DEMO:**

We chose AES encryption for service encryption. This encryption provides security for the services before provision.

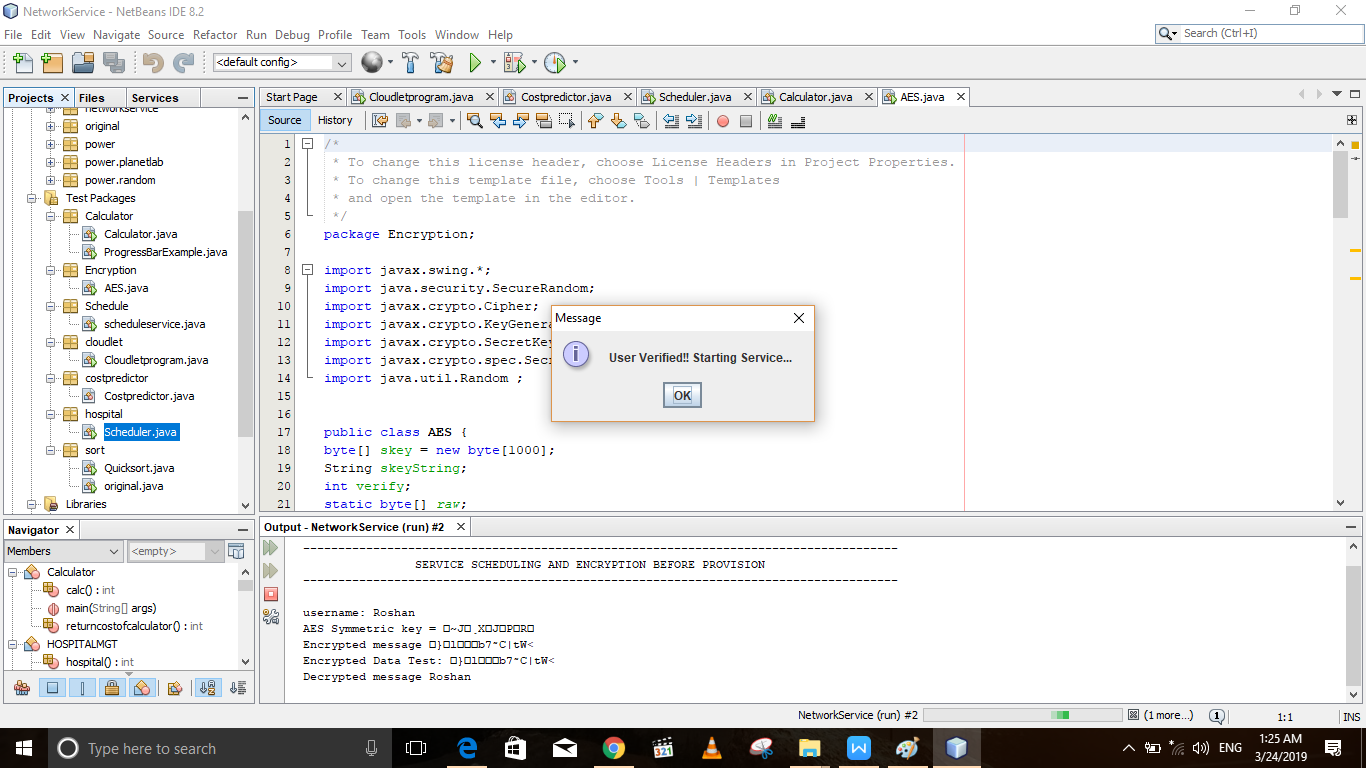
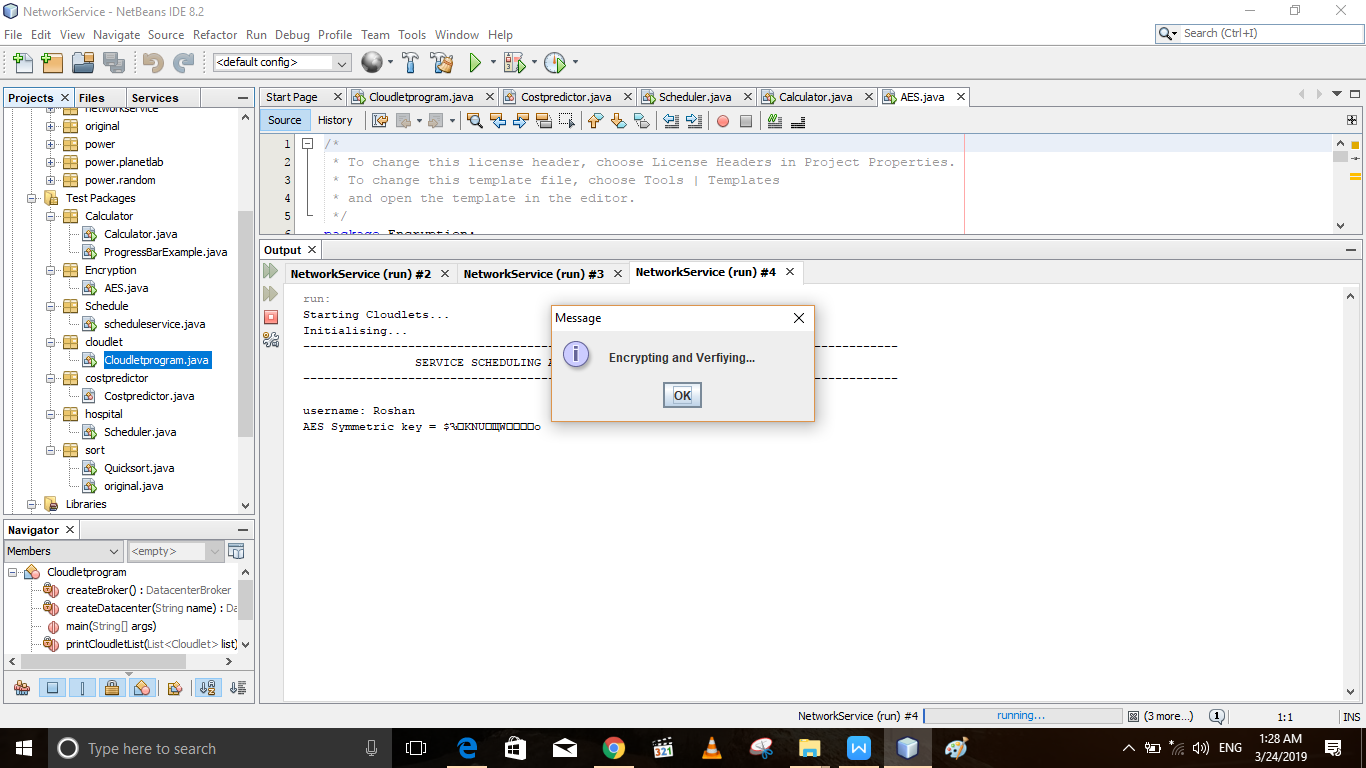
.

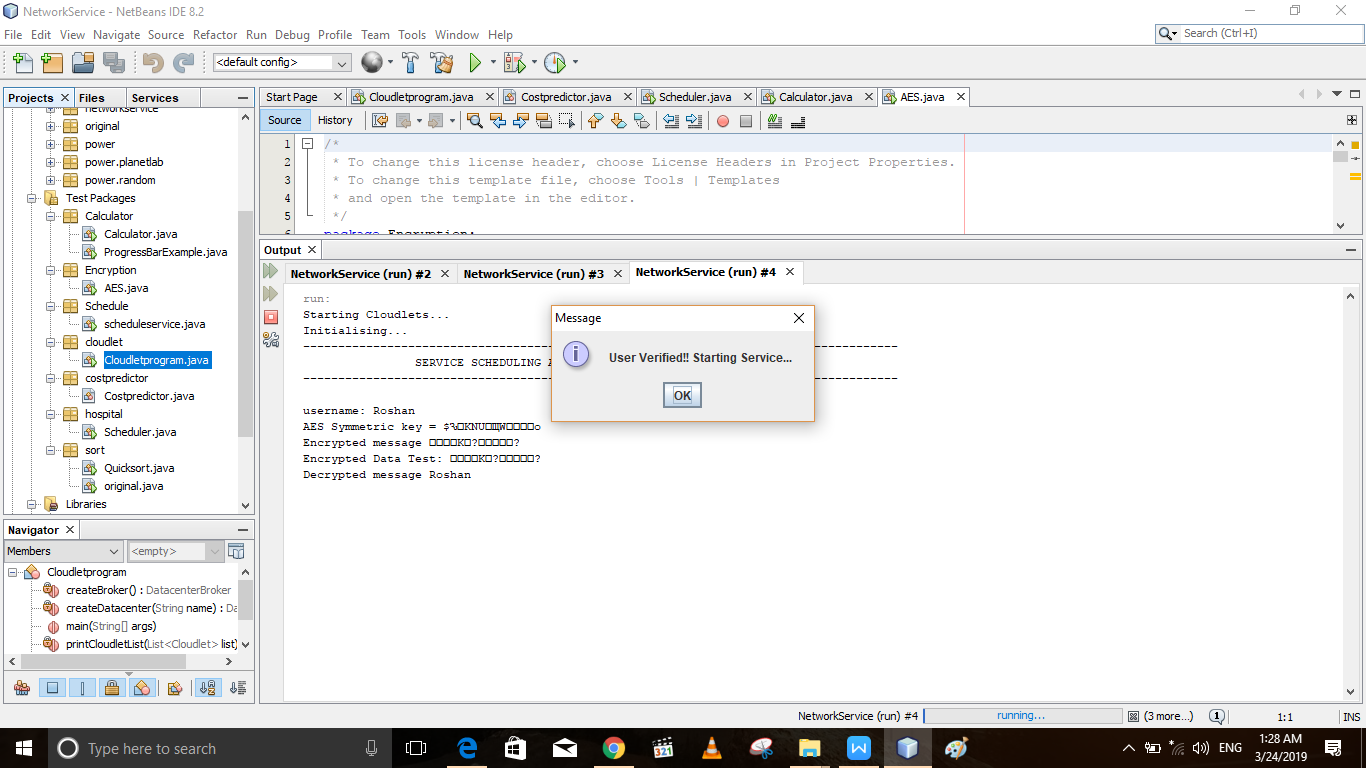
FIGURE 2: AES ENCRYPTION

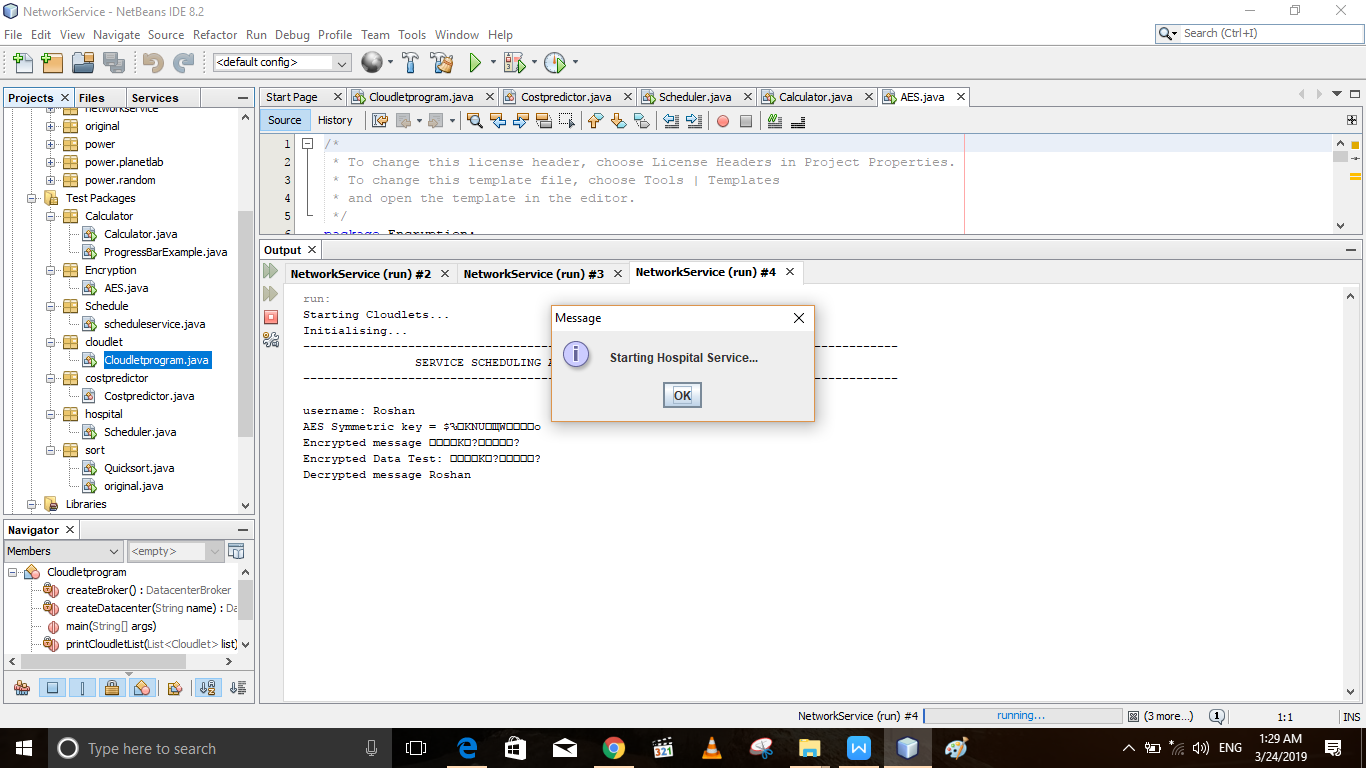
**SCENARIO 1:**

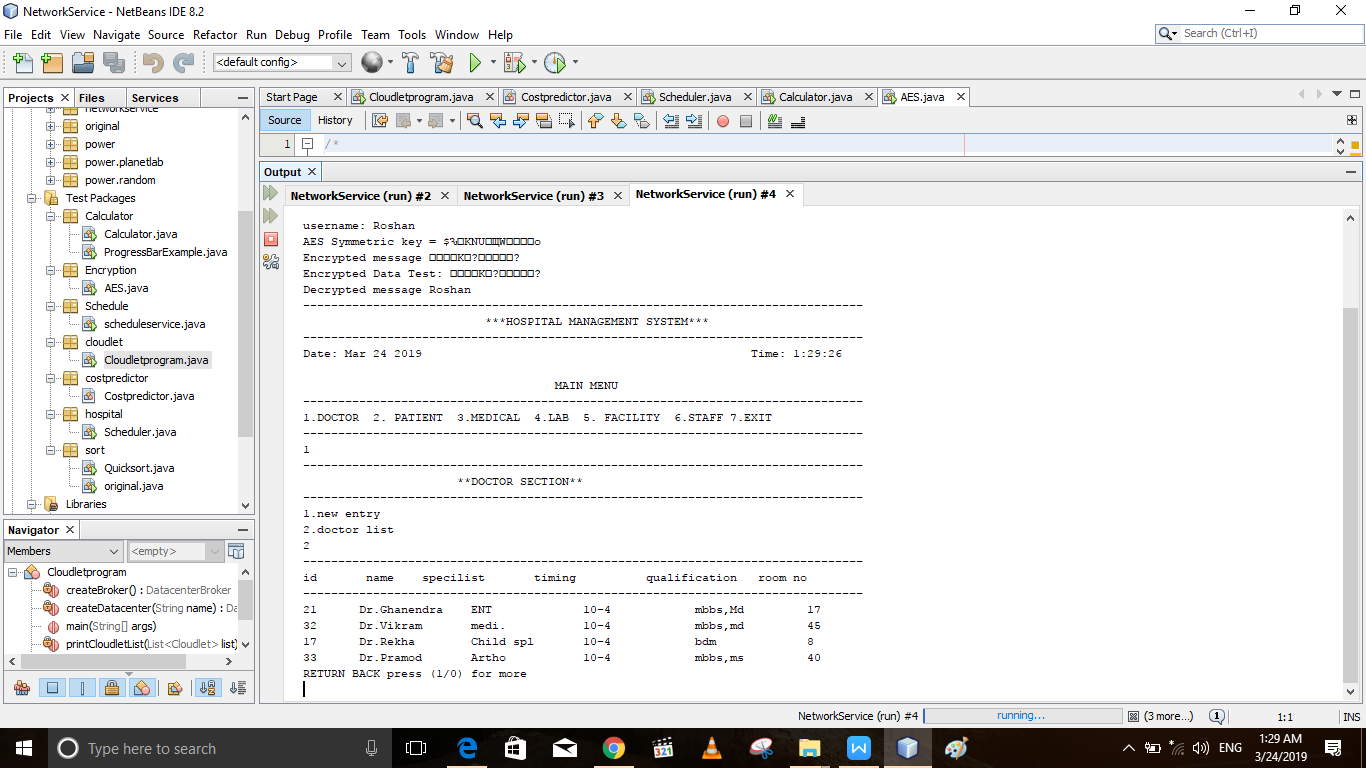
**Number of requests for the Hospital is less than the calculator requests and the cost of the hospital requests is more.**

**STEP 1: VERIFYING USING ENCRYPTION BEFORE HOSPITAL SERVICE PROVISION.**

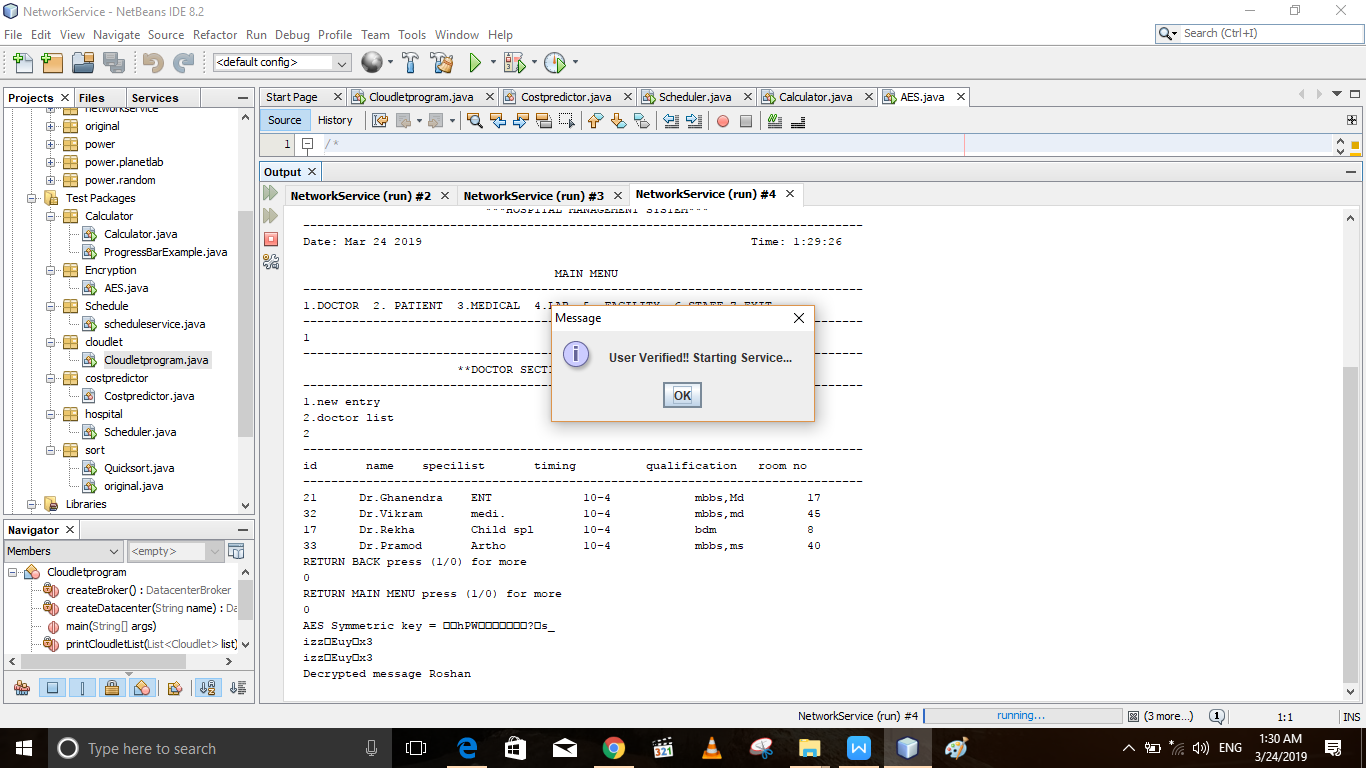


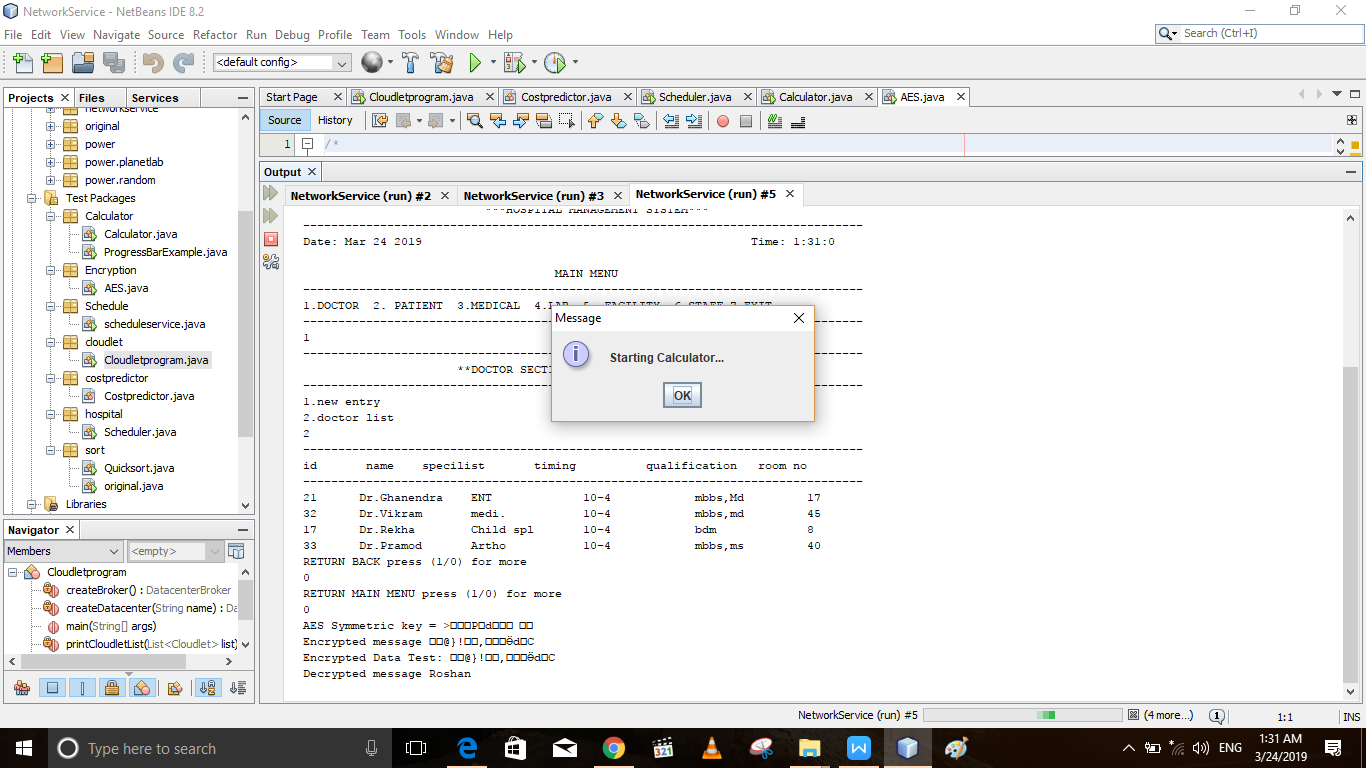




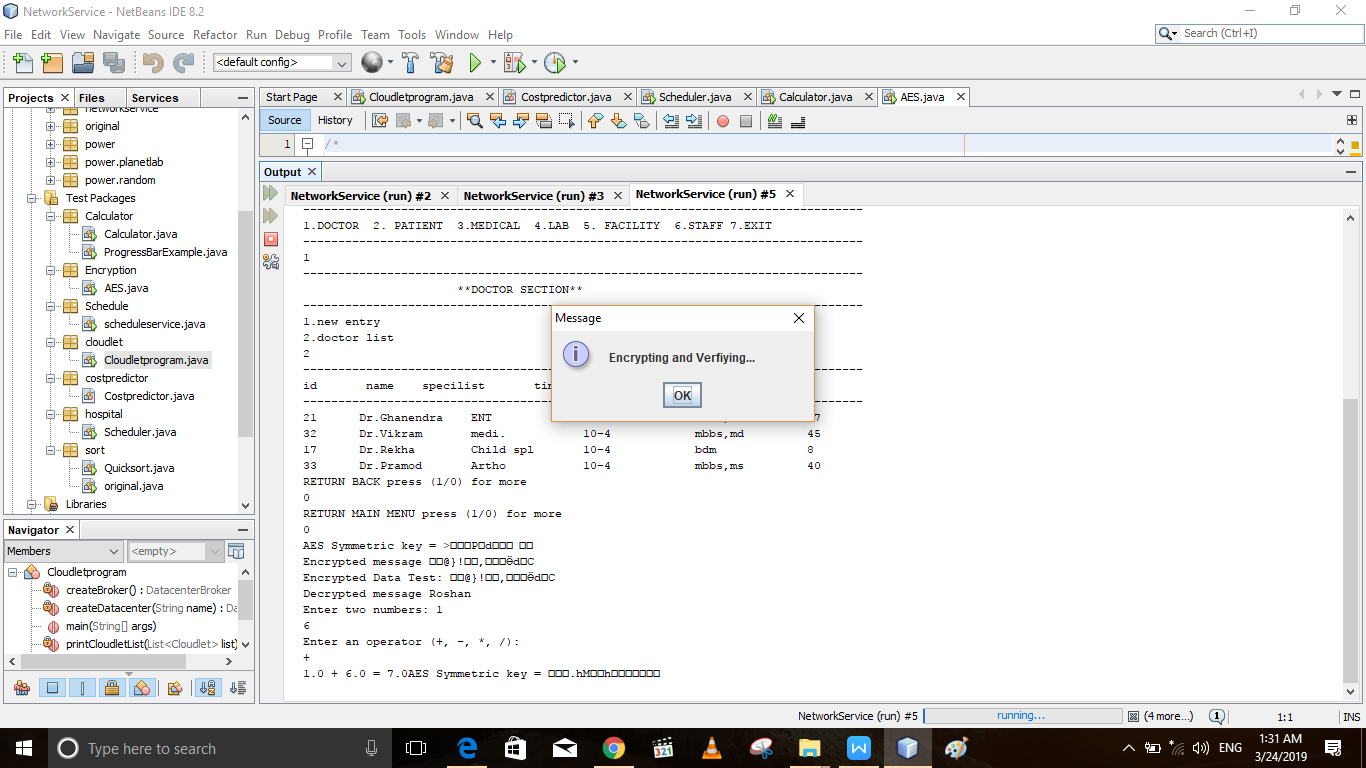


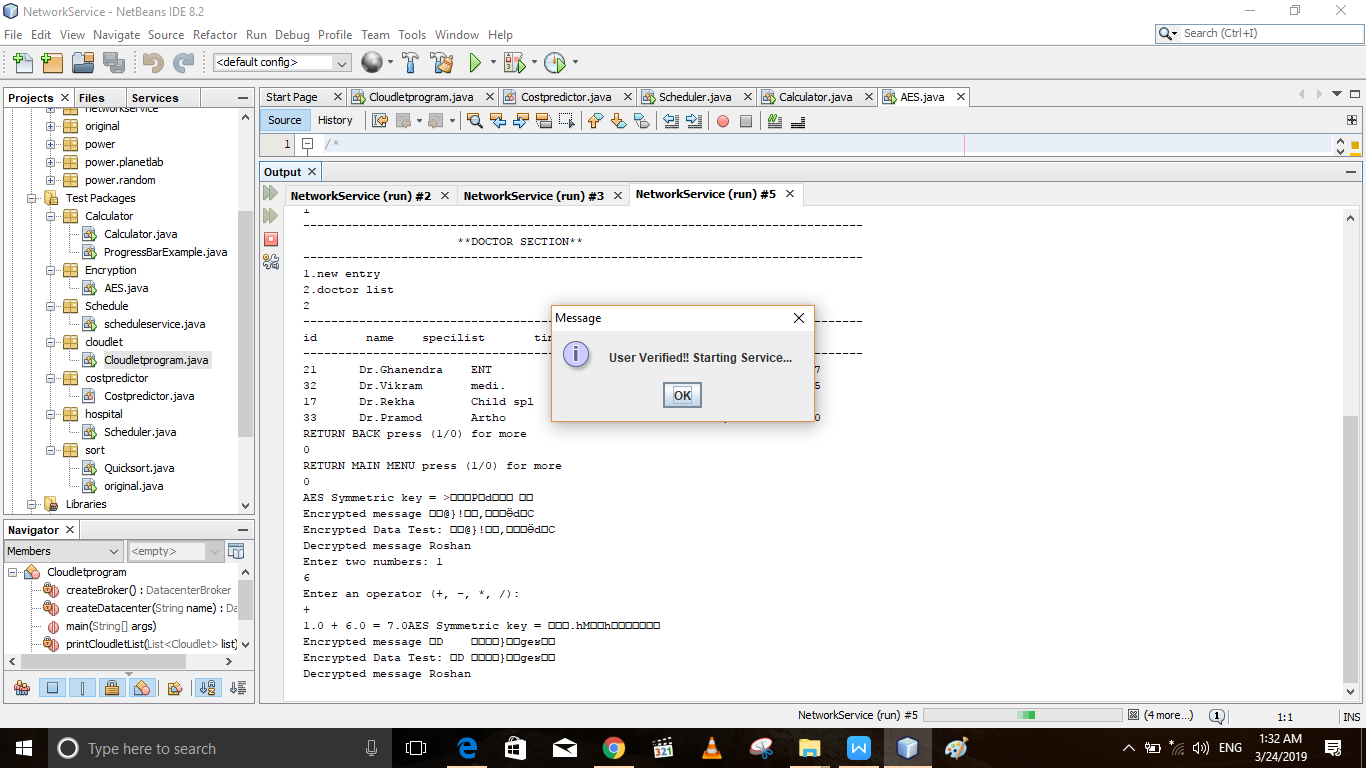
Starting Calculator Service…

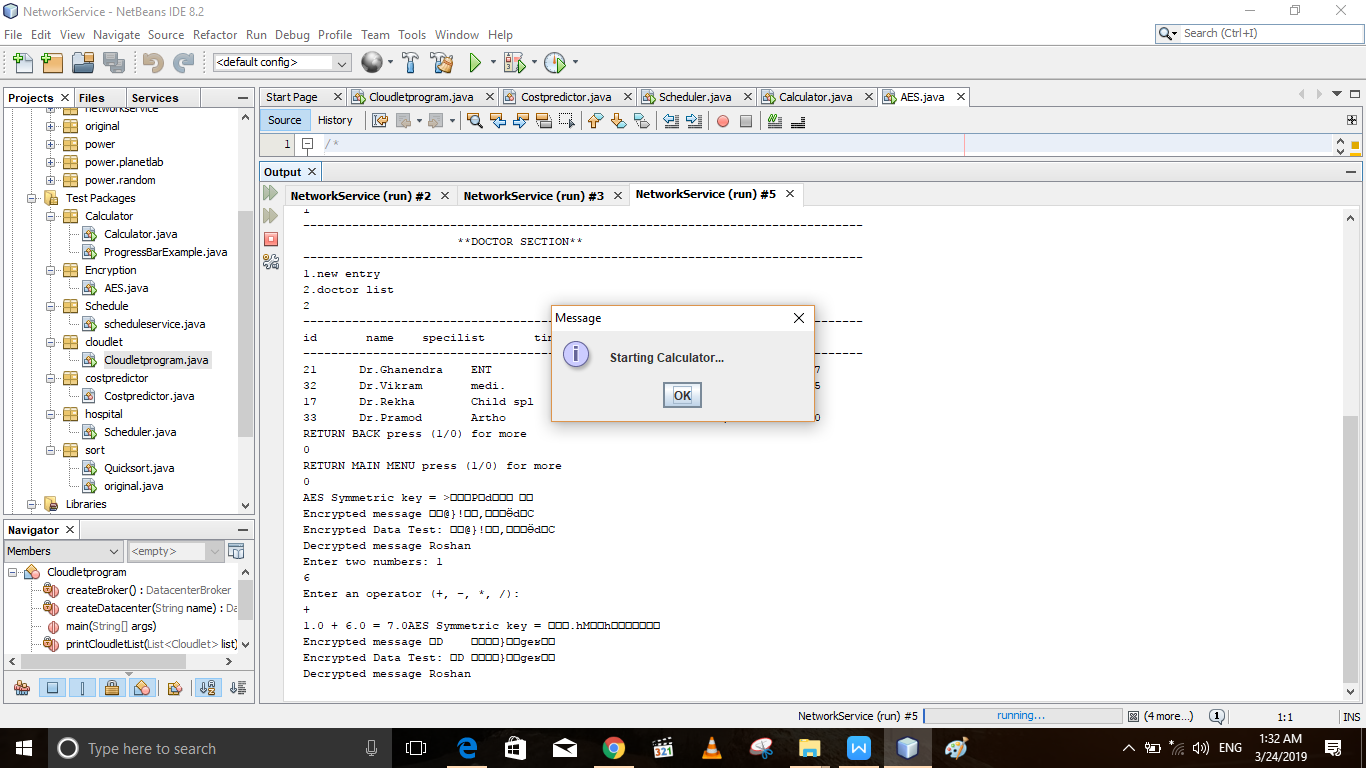


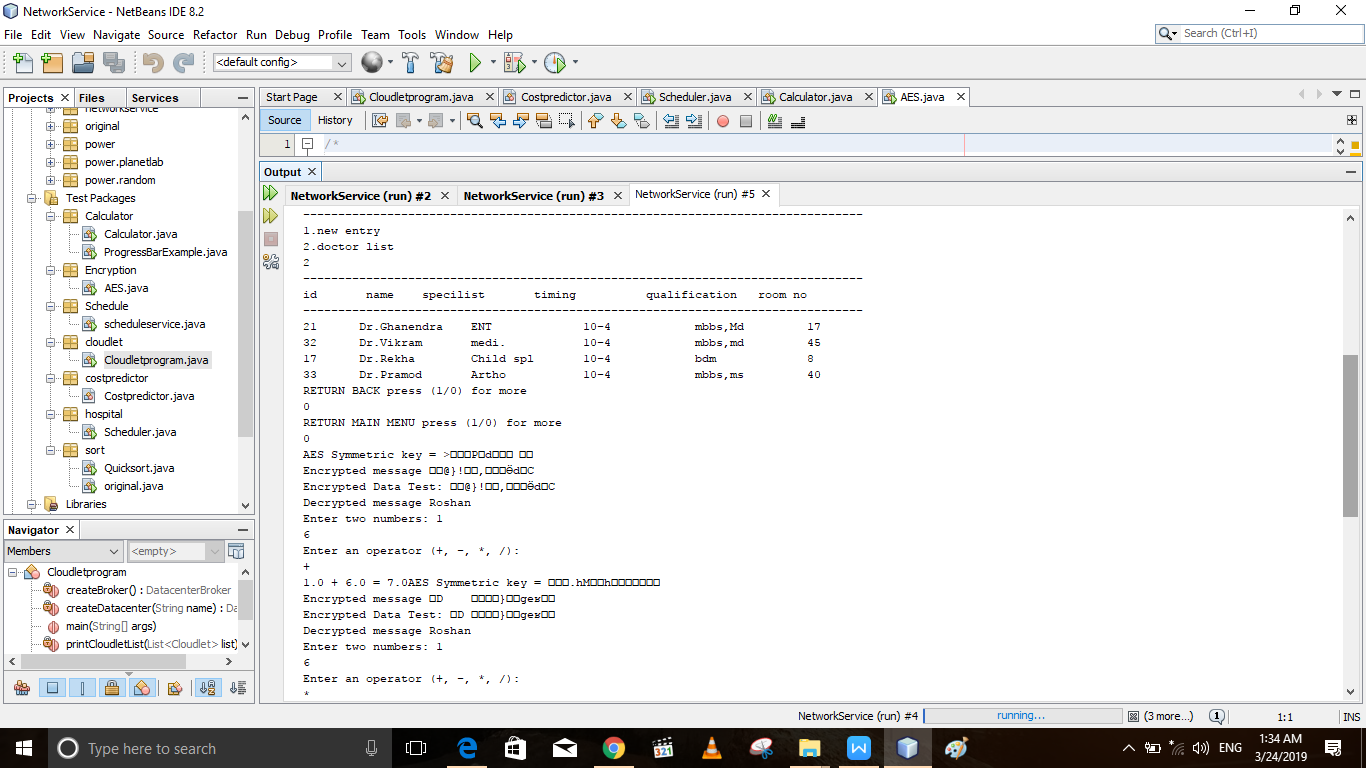


Verifying again and start the service for the request 2:





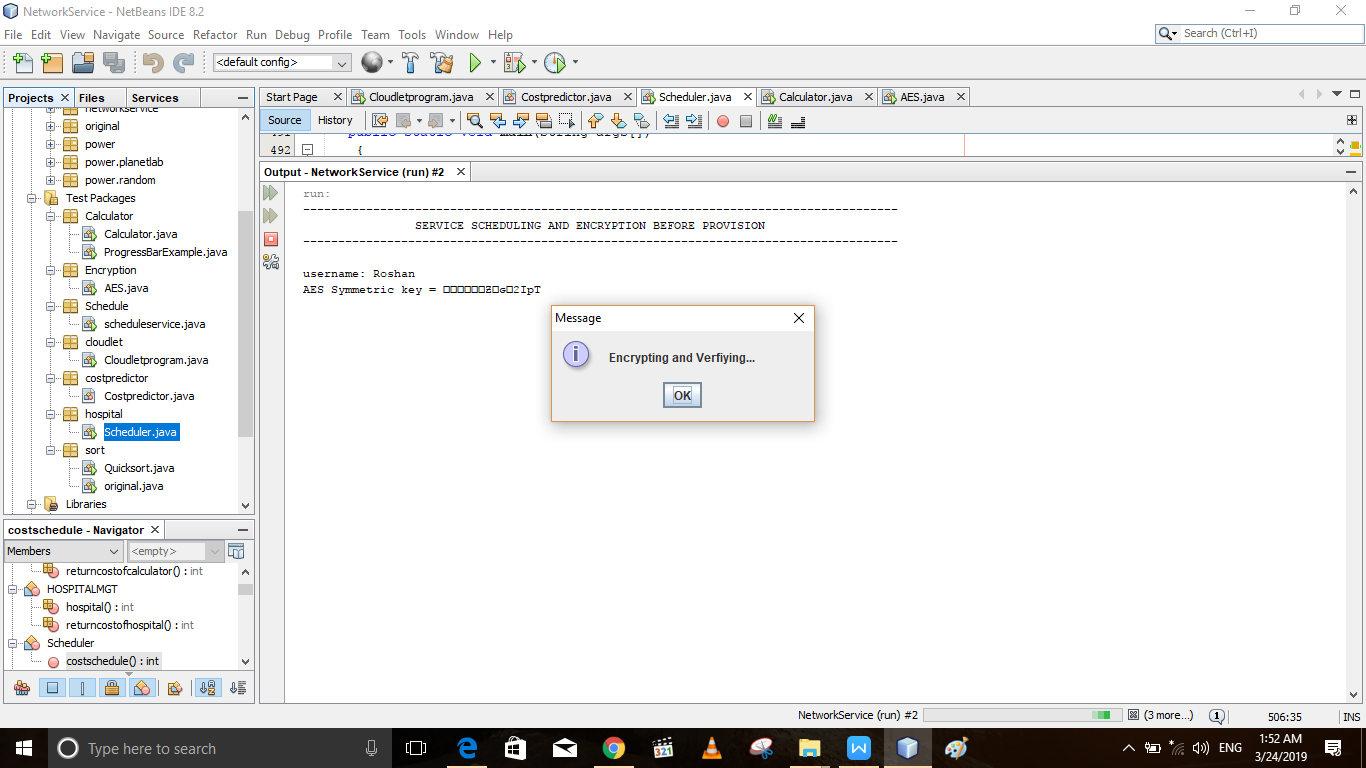




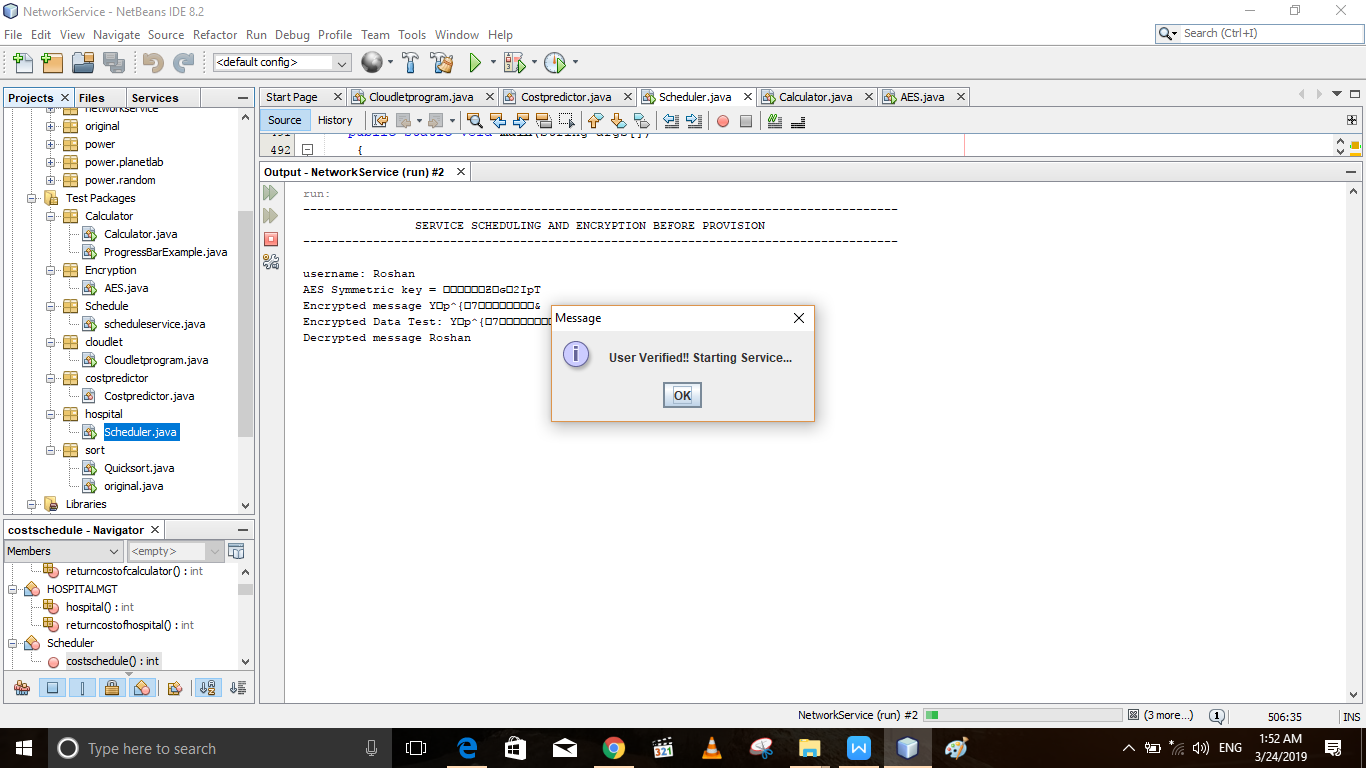
**SCENARIO 2:**

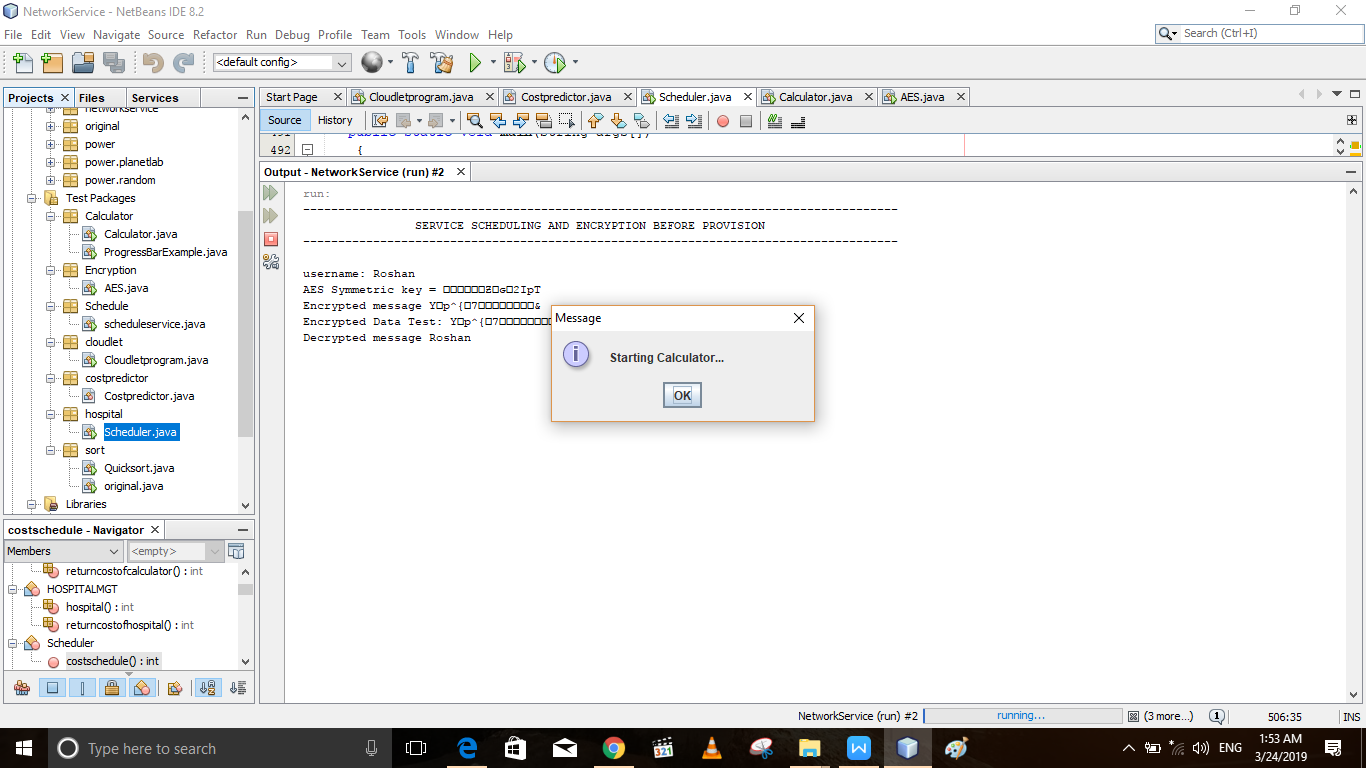
**Where calculator requests is less than the hospital requests as well as the cost is also less.**

**STEP 1: VERIFYING AND STARTING CALCULATOR SERVICE.**

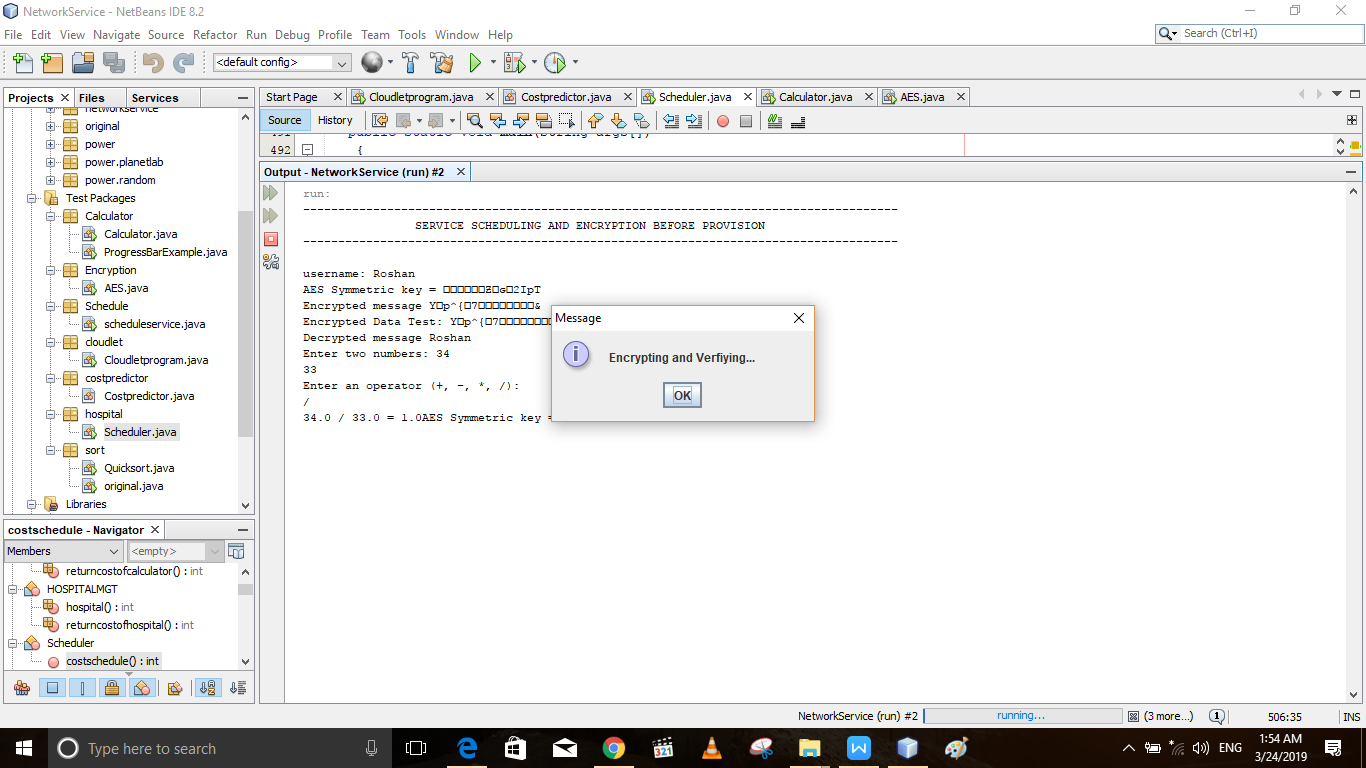
****

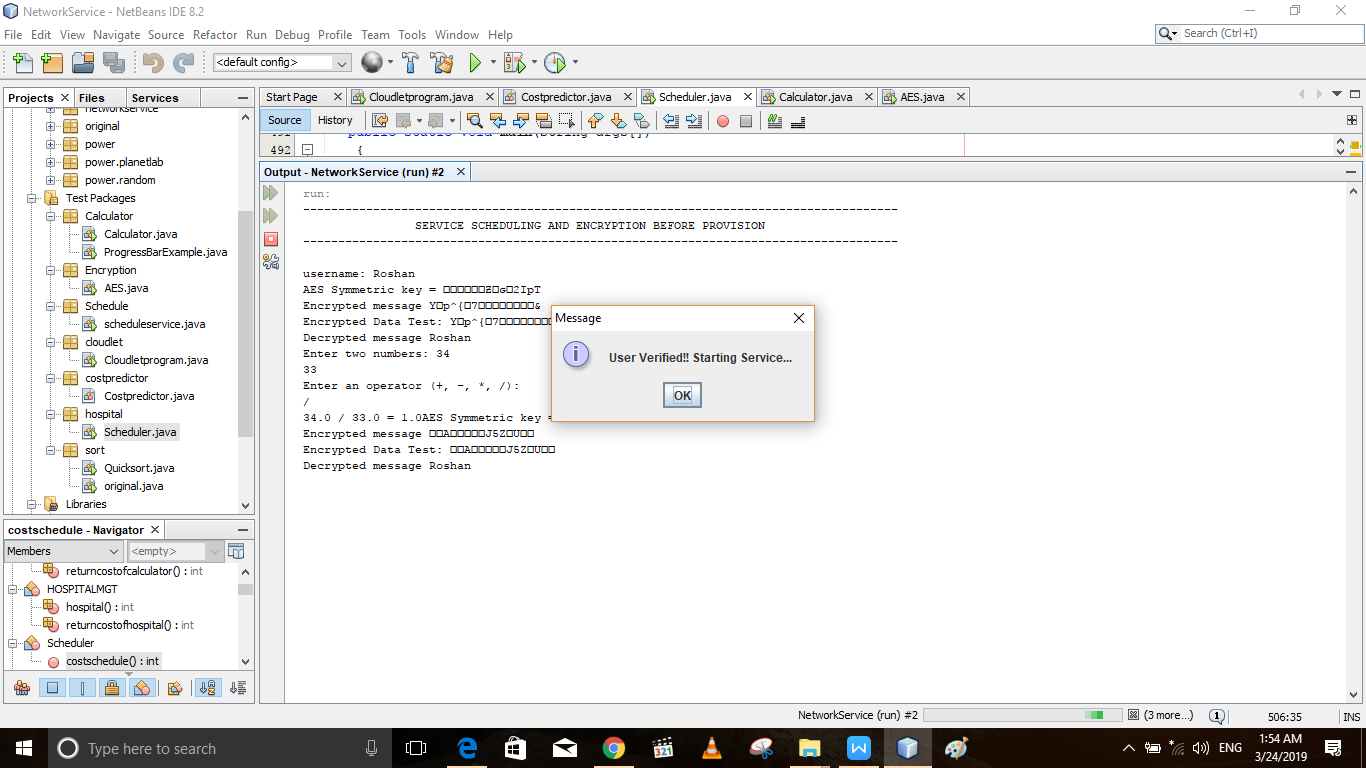
Starting Calculator Service…

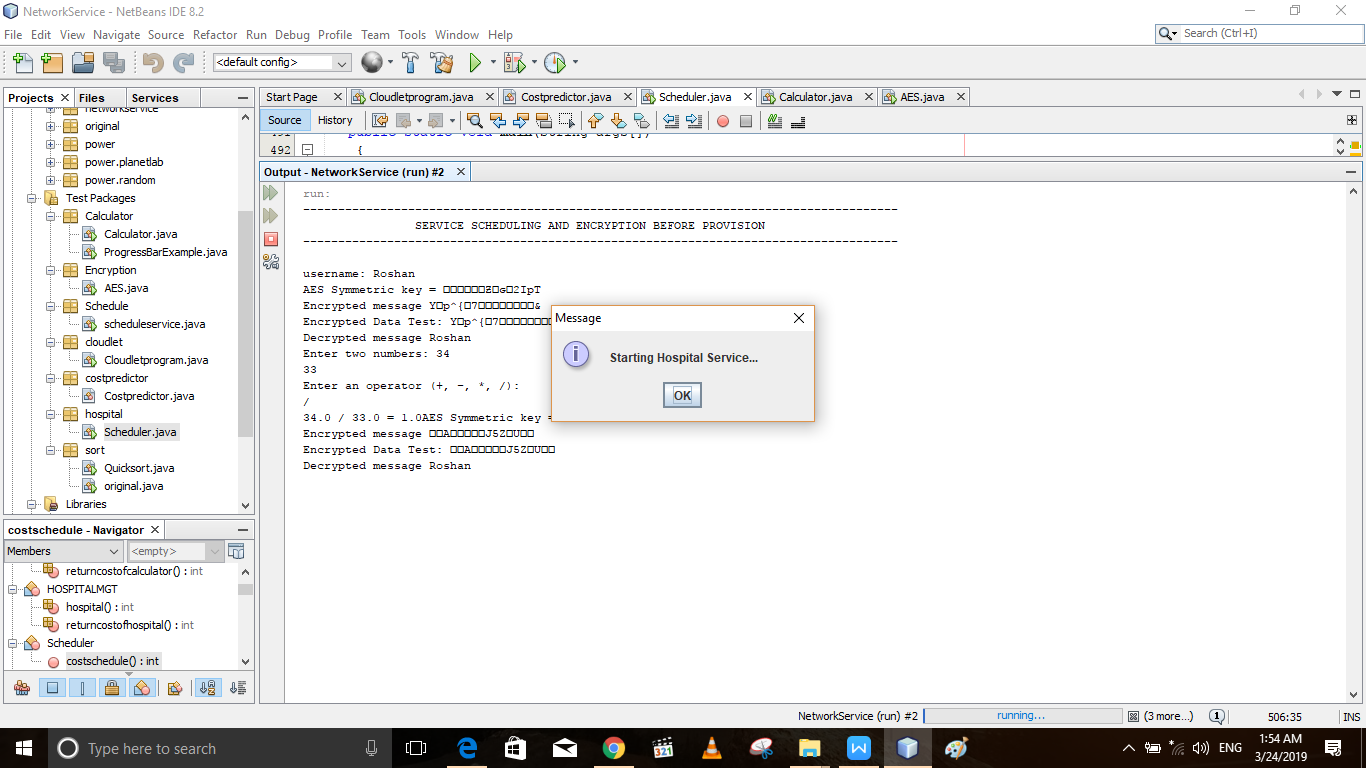


****

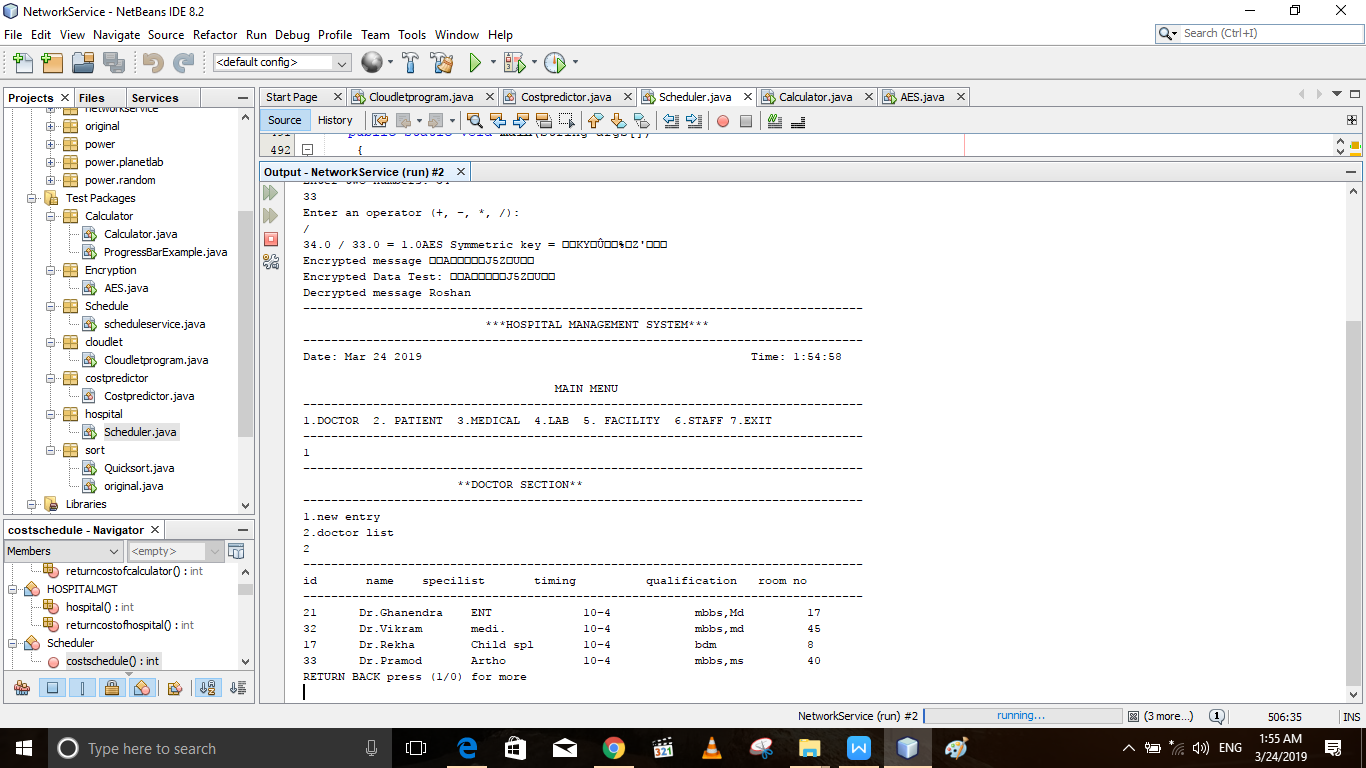
**Verifying again for the next request.**

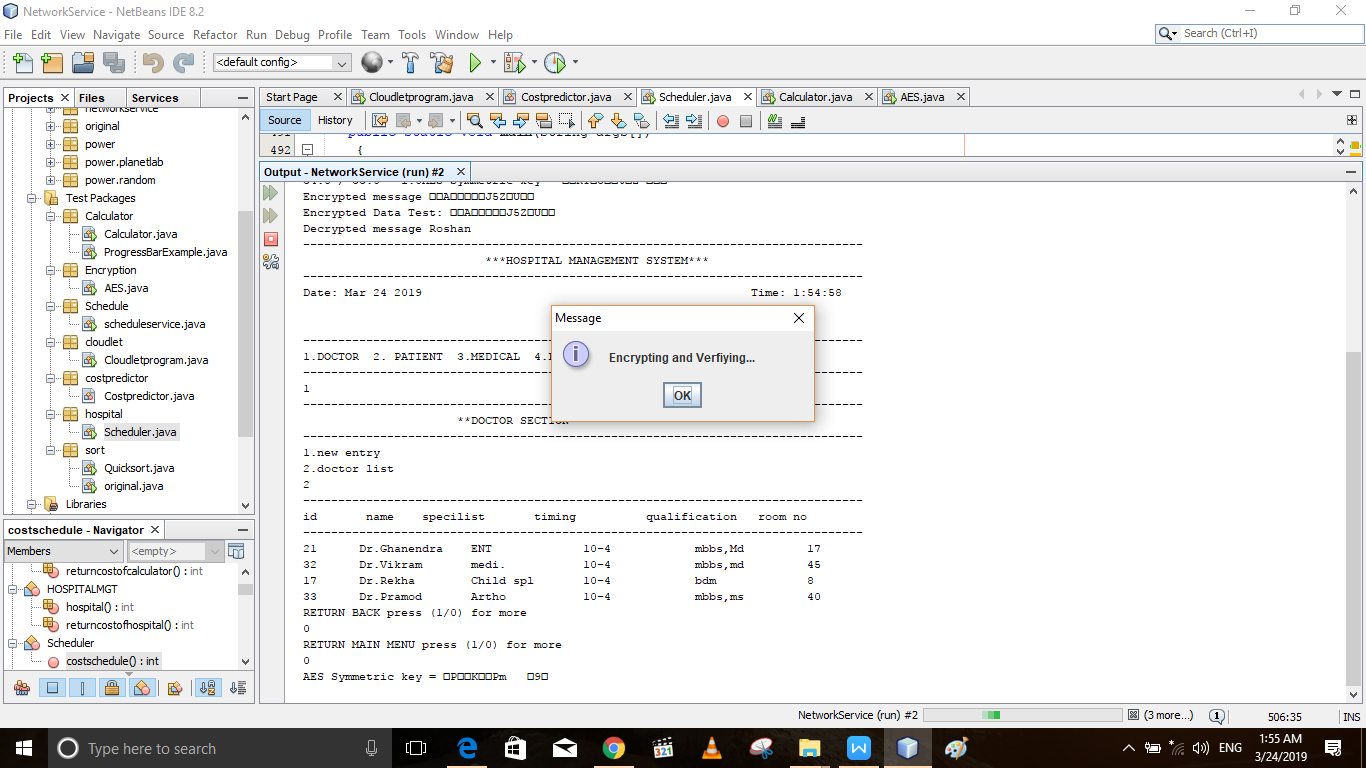
****

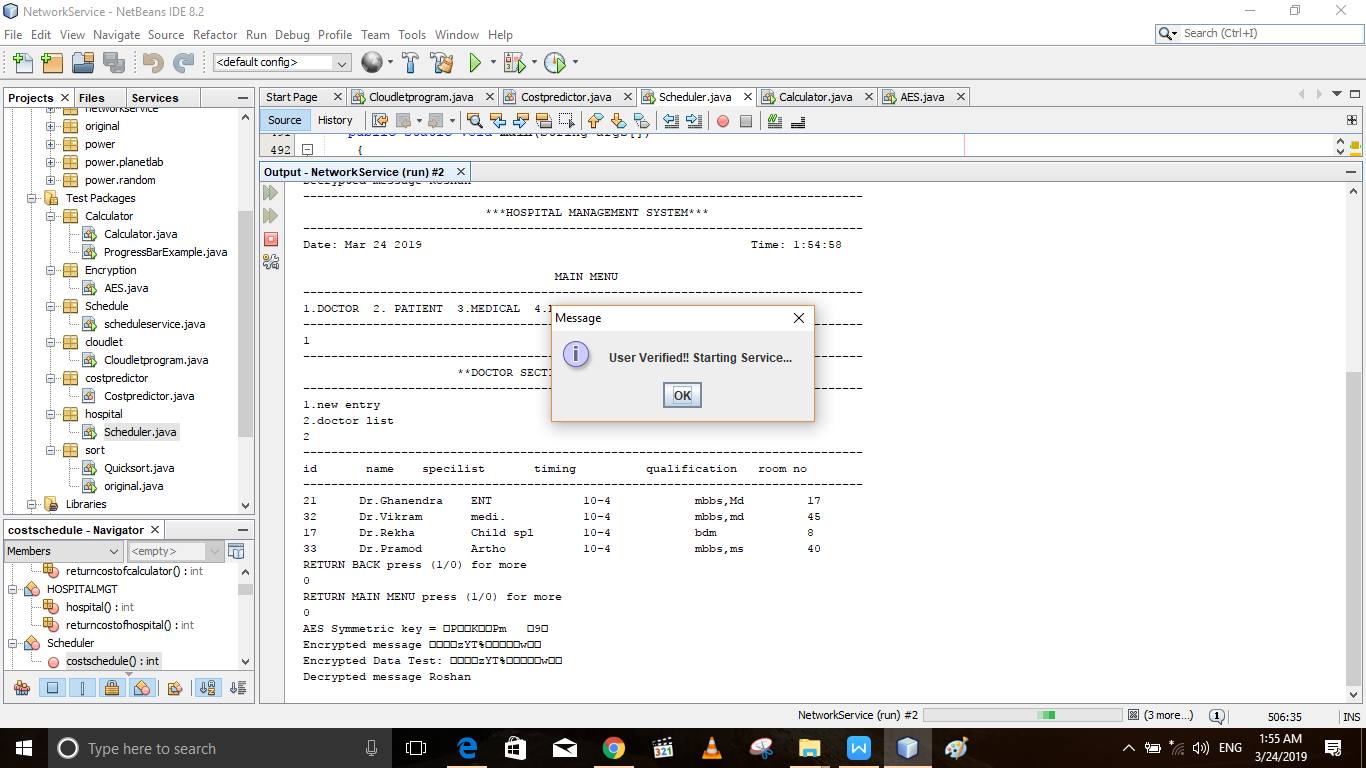


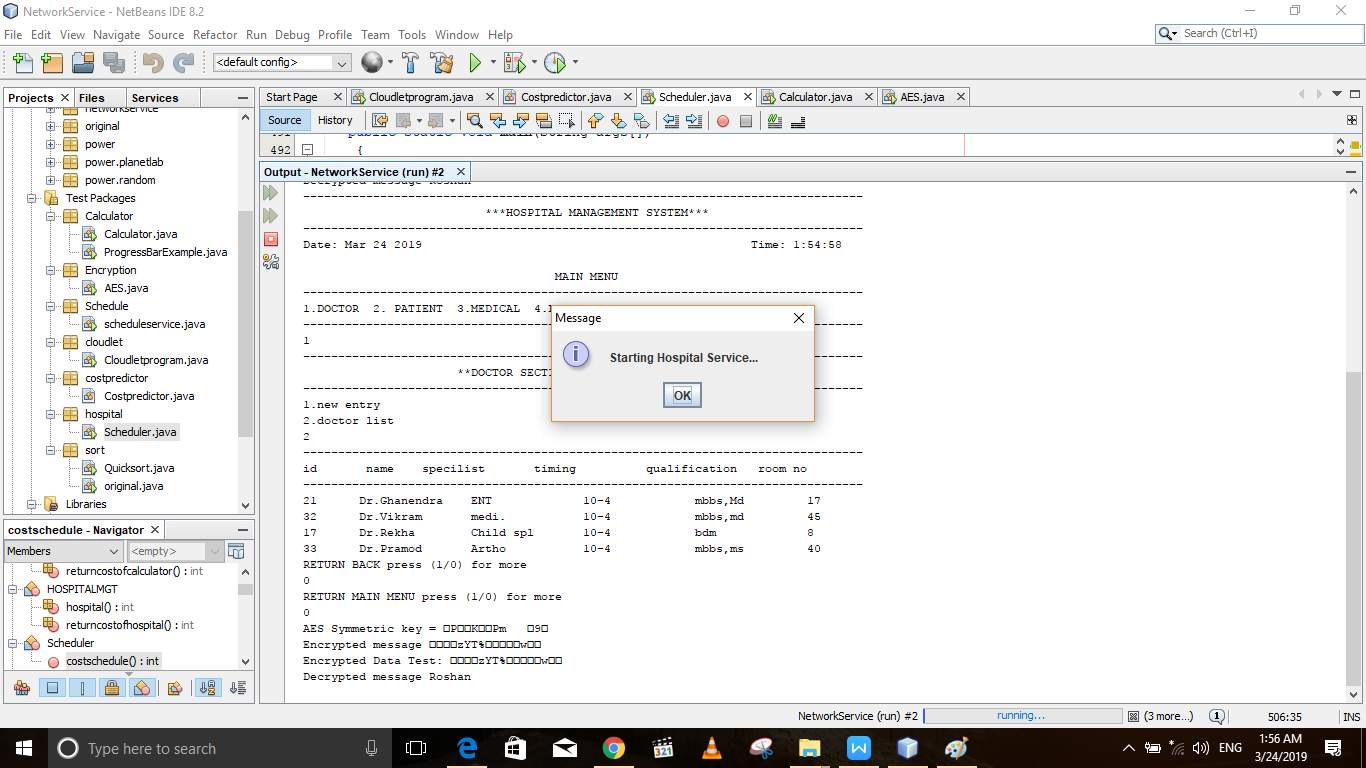
****

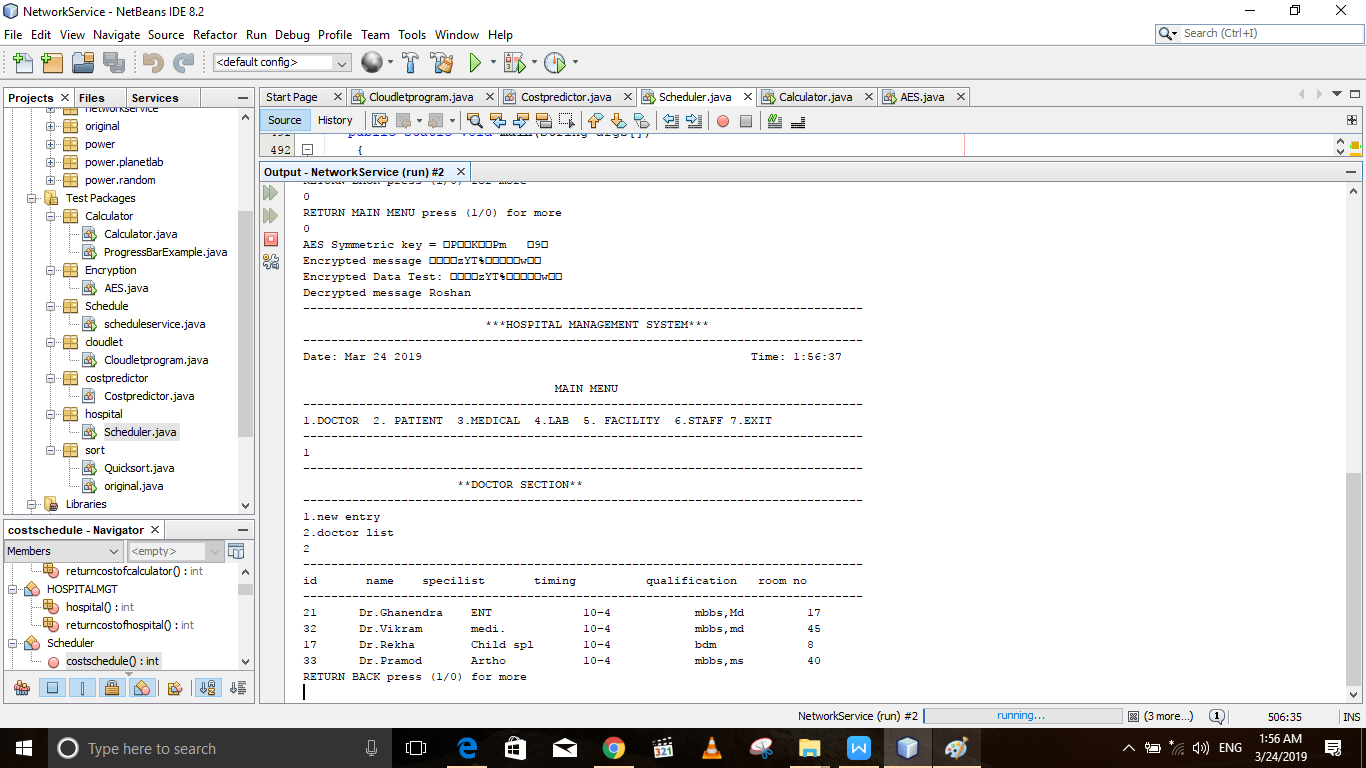
Starting Hospital Service.

****

****

****

****

****

**CONCLUSION AND SCOPE FOR FUTURE WORK:**

This project has huge scope in future for effiecient service provision and adds an additional parameter “Security” to the application service providers which is given huge priority in this cyber-threat-prone world.

**REFERENCES:**

1)PacketCloud: A Cloudlet-Based Open Platform for In-Network Services

Yang Chen, Senior Member, IEEE, Yu Chen, Qiang Cao, and Xiaowei Yang

2)Application Service Provider Model: Perspectives and Challenges

Lixin Tao , *Member IEEE* and *ACM*.

3) Modeling and Simulation of Scalable Cloud Computing Environments and

the CloudSim Toolkit: Challenges and Opportunities

Rajkumar Buyya1, Rajiv Ranjan2 and Rodrigo N. Calheiros1,3