

A Project Report

On

Secured Banking Application using Continuous Authentication

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

by

D.Lakshmi	164G1A0544
M.Archana	164G1A0507
M.Mani Chaitanya Sreenivasa Reddy	164G1A0552
V.Divya	164G1A0524

Under the Guidance of

**Mrs.M.Soumya,M.Tech.
Assistant Professor**



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(B.Tech Program Accredited by NBA)

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY:ANANTAPURAMU

(Accredited by NAAC with 'A' Grade, Affiliated to JNTUA, Approved by AICTE, New Delhi)

2019-2020



Certificate

This is to certify that the project report entitled **Secured Banking Application using Continuous Authentication** is the bonafide work carried out by **D.Lakshmi** bearing Roll Number **164G1A0544**, **M.Archana** bearing Roll Number **164G1A0507**, **M.Mani Chaitanya Sreenivasa Reddy** bearing Roll Number **164G1A0552** and **V.Divya** bearing Roll Number **164G1A0524** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2019-2020.

Guide

Mrs.M.Soumya,M.Tech.
Assistant Professor

Head of the Department

Dr. G.K.V. Narasimha Reddy, Ph.D
Professor & HOD

Date:

EXTERNAL EXAMINER

Ananthapuramu

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express my gratitude for all of them.

It is with immense pleasure that we would like to express my indebted gratitude to my Guide **Mrs.M.Soumya,M.Tech,Computer Science & Engineering**,who has guided me a lot and encouraged me in every step of the project work. We thank her for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deep-felt gratitude to **Mr.R.SandeepKumar, (Ph.D),Assistant Professor**, project coordinator valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We are very much thankful to **Dr. G.K.V.Narasimha Reddy, Ph.D, Professor & Head of the Department, Computer Science & Engineering**, for his kind support and for providing necessary facilities to carry out the work.

We wish to convey my special thanks to **Dr.T.Hitendra Sarma,Ph.D,Principal of Srinivasa Ramanujan Institute of Technology**for giving the required information in doing our project work. Not to forget, we thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities. Finally,we wish to convey our gratitude to our family who fostered all the requirements and facilities that we need.

Project Associates

DECLARATION

We D.Lakshmi bearing reg no : 164G1A0544, M.Archana bearing reg no : 164G1A0507, M.Mani Chaitanya Sreenivasa Reddy bearing reg no : 164G1A0552, V.Divya bearing reg no : 164G1A0524, students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, Rotarypuram , hereby declare that the dissertation entitled “SECURED BANKING APPLICATION USING CONTINUOUS AUTHENTICATION” embodies the report of our project work carried out by us during IV Year Bachelor of Technology under the guidance of Mrs.M.Soumya, M.Tech, Department of CSE and this work has been submitted for the partial fulfillment of the requirements for the award of Bachelor of Technology degree.

The results embodied in this project report have not been submitted to any other Universities or Institute for the award of Degree.

D.Lakshmi	Reg no: 164G1A0544
M.Archana	Reg no: 164G1A0507
M.Mani Chaitanya Sreenivasa Reddy	Reg no: 164G1A0552
V.Divya	Reg no: 164G1A0524

CONTENTS

	Page NO
List of Figures	v
List of Screens	vi
List of Abbreviations	vii
Abstract	viii
1. Introduction	1
1.1 Secure Computing	1
1.2 Benefits of Secure Computing	3
2. Literature Survey	5
2.1 Model-based evaluation of scalability and security tradeoffs	5
2.2 Attacks on Biometric Systems	6
2.3 Automated Generation and Analysis of Attack Graphs	6
3.System Analysis	7
3.1 Existing System	7
3.1.1 Disadvantage of Existing System	7
3.2 Proposed System	7
4.Technology	8
4.1 Language Used	8
4.1.1 History of Java	8
4.1.2 Features of Java	9
4.1.3 Applications of java	11
4.2 Java Installation Procedure	13
4.3 Java IDE	25
4.4 Net Beans IDE	25

4.4.1 Net Beans Installation Procedure	26
5.Design	33
5.1 UML Introduction	33
5.2 Usage of UML in Project	33
5.3 Goals of UML	34
5.4 Diagrams	34
5.4.1 Data Flow Diagram	34
5.4.2 Use Case Diagram	36
5.4.3 Class Diagram	36
5.4.4 Activity Diagram	37
6.Implementation	39
6.1 Modules	39
6.2 Modules Description	39
6.2.1 System Model	39
6.2.2 Authentication Server	40
6.2.3 Image Verification	40
6.2.4 Continuous Authentication	41
6.3 Outputs	41
6.3.1 user	41
6.3.2 Server	46
6.4 AES(Advanced Encryption Standard)	47
6.4.1 Working of AES	48
6.4.2 AES features	49
7. System Testing	50
7.1 Types Of Tests	50
7.1.1 Unit testing	50

7.1.2 Integration testing	50
7.1.3 Functional testing	51
7.1.4 System Test	51
7.1.5 White Box testing	51
7.1.6 Black Box testing	52
Conclusion	53
Bibliography	54

LIST OF FIGURES

Figure Number	Description	PAGE NO
4.2.1	Select java SE6	14
4.2.2	Select window version	15
4.2.3	Installation wizard	16
4.2.4	Change Location	17
4.2.5	Extracting Installer	18
4.2.6	Successfully installed	18
4.2.7	Opening environment variables	19
4.2.8	Click New Environment Variable	20
4.2.9	Enter variable name and value	20
4.2.10	Select path variable	21
4.2.11	Edit environment variables	22
4.2.12	Close environment variable	23
4.2.13	Command prompt	24
4.2.14	Java version	24
4.2.15	Successfully installed	25
4.4.1	install net bean 7.2.1	26
4.4.2	Run installer	26
4.4.3	Select application server to install	27
4.4.4	Accept license agreement	28
4.4.5	Junit License agreement	28
4.4.6	Choose destination folder for Net beans	29
4.4.7	Destination folder for glass fish	30
4.4.8	start installation	30
4.4.9	Wait to install	31
4.4.10	Installation completed	32
5.4.1	Data flow diagram	35
5.4.2.1	Use case diagram	36
5.4.3.1	Class diagram	37
5.4.4.1	Activity diagram	38
6.4.1.1	AES design	48

LIST OF SCREENS

Screen Number	Description	Page NO
6.3.1.1	Registration form	42
6.3.1.2	User login page	43
6.3.1.3	Image verification	44
6.3.1.4	User page	45
6.3.2.1	Server login page	46
6.3.2.2	Server page	47

List Of Abbreviations

PDA	Personal Digital Assistant
SAV	Symantec Antivirus
FSA	Financial Conduct Authority
SOX	Sarbanes-Oxley
ICT	Information and Communications Technology
SAN	Stochastic Activity Networks
WORA	Write Once Run Anywhere
Java SE	Java Standard Edition
Java ME	Java Platform Micro Edition
Java EE	Java platform Enterprise Edition
API's	Application Program Interface
AWT	Abstract Windowing Toolkit
GUI	Graphical User Interface
SDK	Software Development Kit
SIM	Subscriber Identity Module
RWS	Rimfaxe Web Server
JBoss	Java Beans Open Source Software
EAP	Enterprise Application Platform
MAT	Matrix Laboratory
JDK	Java Development Kit
IDE	Integrated Development Environment
HTML	Hyper Text Markup Language
UML	Unified Modeling Language
DFD	Data Flow Diagram
ID	Identification
AES	Advanced Encryption Standard
NIST	National Institute Of Standard and Technology

ABSTRACT

The Internet of today has become an integral part of our everyday life and the proportion of users expecting to be able to manage their bank accounts securely is increasing. As such, Internet banking has come to age as a crucial component of present society, it has become an indispensable part of social life and industrial activity for mankind. In recent years, the demand for online banking has increased and the number of people who rely on online transactions has tremendously increased. Thus, necessity for a reliable security for online transactions is ever than before. Furthermore, security concerns still exist among the general public when using online applications.

Present banking applications are performing single time authentication, in this paper we proposed continuous authentication. Session management in distributed Internet services is traditionally based on username and password, user session expiration using timeouts is performed in this project. Additionally, we use image name verification which helps to increase the security of banking application..we defined authentication through continuous user verification.

CHAPTER 1

INTRODUCTION

Malicious users can take advantage of these security weaknesses to penetrate to the system. In order to provide an effective solution for the security of the data of ensure a completely secure migration of all their data anywhere in the cloud through continuous authentication.

1.1 Secure Computing:

Computer security is information security as applied to computers and networks. The field covers all the processes and mechanisms by which computer-based equipment, information and services are protected from unintended or unauthorized access, change or destruction. Computer security also includes protection from unplanned events and natural disasters. Otherwise, in the computer industry, the term security -- or the phrase computer security -- refers to techniques for ensuring that data stored in a computer cannot be read or compromised by any individuals without authorization. Most computer security measures involve data encryption and passwords. Data encryption is the translation of data into a form that is unintelligible without a deciphering mechanism. A password is a secret word or phrase that gives a user access to a particular program or system.

❖ Working conditions and basic needs in the secure computing:

If you don't take basic steps to protect your work computer, you put it and all the information on it at risk. You can potentially compromise the operation of other computers on your organization's network, or even the functioning of the network as a whole.

- **Physical security:**

Technical measures like login passwords, anti-virus are essential.

However, a secure physical space is the first and more important line of defence.

Is the place you keep your workplace computer secure enough to prevent theft or access to it while you are away? While the Security Department provides coverage across the Medical centre, it only takes seconds to steal a computer, particularly a portable device like a laptop or a PDA. A computer should be secured like any other valuable possession when you are not present.

Human threats are not the only concern. Computers can be compromised by environmental mishaps (e.g., water, coffee) or physical trauma. Make sure the physical location of your computer takes account of those risks as well.

- **Access passwords:**

The University's networks and shared information systems are protected in part by login credentials (user-IDs and passwords). Access passwords are also an essential protection for personal computers in most circumstances. Offices are usually open and shared spaces, so physical access to computers cannot be completely controlled.

To protect your computer, you should consider setting passwords for particularly sensitive applications resident on the computer (e.g., data analysis software), if the software provides that capability.

- **Anti-virus software:**

Up-to-date, properly configured anti-virus software is essential. While we have server-side anti-virus software on our network computers, you still need it on the client side (your computer).

- **Firewalls:**

Anti-virus products inspect files on your computer and in email. Firewall software and hardware monitor communications between your computer and the outside world. That is essential for any networked computer.

- **Software updates:**

It is critical to keep software up to date, especially the operating system, anti-virus and anti-spyware, email and browser software. The newest versions will contain fixes for discovered vulnerabilities.

Almost all anti-virus have automatic update features (including SAV). Keeping the "signatures" (digital patterns) of malicious software detectors up-to-date is essential for these products to be effective.

- **Keep secure backups:**

Even if you take all these security steps, bad things can still happen. Be prepared for the worst by making backup copies of critical data, and keeping those backup copies in a separate, secure location. For example, use supplemental hard drives, CDs/DVDs, or flash drives to store critical, hard-to-replace data.

- **Report problems:**

If you believe that your computer or any data on it has been compromised, you should make a information security incident report. That is required by University policy for all data on our systems, and legally required for health, education, financial and any other kind of record containing identifiable personal information.

1.2 Benefits of secure computing:

Protect your civil liability:

You may be held legally liable to compensate a third party should they experience financial damage or distress as a result of their personal data being stolen from you or leaked by you.

- **Protect your credibility compliance:**

You may require compliancy with the Data Protection Act, the FSA, SOX or other regulatory standards. Each of these bodies stipulates that certain measures be taken to protect the data on your network.

- **Protect your reputation Spam:**

A common use for infected systems is to join them to a collection of infected machines which takes orders from a command server and use them to send out spam. This spam can be traced back to you, your server could be blacklisted and you could be unable to send email.

- **Protect your income competitive advantage:**

There are a number of “hackers-for-hire” advertising their services on the internet selling their skills in breaking into company’s servers to steal client databases, proprietary software, merger and acquisition information, personnel details.

- **Protect your business blackmail:**

A seldom-reported source of income for “hackers” is to break into your server, change all your passwords and lock you out of it. The password is then sold back to you. Note: the “hackers” may implant a backdoor program on your server so that they can repeat the exercise at will.

- **Protect your investment Free storage:**

Your server’s hard drive space is used (or sold on) to house the hacker's video clips, music collections, pirated software or worse. Your server or computer then becomes continuously slow and your internet connection speeds deteriorate due to the number of people connecting to your server in order to download the offered wares.

CHAPTER 2

LITERATURE SURVEY

2.1 Model-based evaluation of scalability and security tradeoffs: A case study on a multi-service platform

AUTHORS: L. Montecchi, N. Nostro, A. Ceccarelli, G. Vella, A. Caruso, and A. Bondavalli

Current ICT infrastructures are characterized by increasing requirements of reliability, security, performance, availability, adaptability. A relevant issue is represented by the scalability of the system with respect to the increasing number of users and applications, thus requiring a careful dimensioning of resources. Furthermore, new security issues to be faced arise from exposing applications and data to the Internet, thus requiring an attentive analysis of potential threats and the identification of stronger security mechanisms to be implemented, which may produce a negative impact on system performance and scalability properties. The paper presents a model-based evaluation of scalability and security tradeoffs of a multi-service web-based platform, by evaluating how the introduction of security mechanisms may lead to a degradation of performance properties. The evaluation focuses on the OPENNESS platform, a web-based platform providing different kind of services, to different categories of users. The evaluation aims at identifying the bottlenecks of the system, under different configurations, and assess the impact of security countermeasures which were identified by a thorough threat analysis activity previously carried out on the target system. The modeling activity has been carried out using the Stochastic Activity Networks (SANs) formalism, making full use of its characteristics of modularity and reusability. The analysis model is realized through the composition of a set of predefined template models, which facilitates the construction of the overall system model, and the evaluation of different configuration by composing them in different ways.

2.2 Attacks on Biometric Systems: A Case Study in Fingerprints

AUTHORS: U. Uludag and A.K. Jain

In spite of numerous advantages of biometrics-based personal authentication systems over traditional security systems based on token or knowledge, they are vulnerable to attacks that can decrease their security considerably. In this paper, we analyze these attacks in the realm of a fingerprint biometric system. We propose an attack system that uses a hill climbing procedure to synthesize the target minutia templates and evaluate its feasibility with extensive experimental results conducted on a large fingerprint database. Several measures that can be utilized to decrease the probability of such attacks and their ramifications are also presented.

2.3 Automated Generation and Analysis of Attack Graphs

AUTHORS: O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J.M. Wing

An integral part of modeling the global view of network security is constructing attack graphs. Manual attack graph construction is tedious, error-prone, and impractical for attack graphs larger than a hundred nodes. In this paper we present an automated technique for generating and analyzing attack graphs. We base our technique on symbolic model checking algorithms, letting us construct attack graphs automatically and efficiently. We also describe two analyses to help decide which attacks would be most cost-effective to guard against. We implemented our technique in a tool suite and tested it on a small network example, which includes models of a firewall and an intrusion detection system.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

- ❖ Once the user's identity has been verified, the system resources are available for a period of time or until explicit logout from the user. This approach assumes that a single verification (at the beginning of the session) is sufficient, and that the identity of the user is constant during the whole session.

3.1.1 DISADVANTAGES OF EXISTING SYSTEM:

- ❖ Existing approaches single time verification.

3.2 PROPOSED SYSTEM:

- ❖ This paper presents a new approach for user verification and session management that is applied in the internet.
- ❖ It is able to operate securely with any kind of web service, including services with high security demands as online banking services, and it is intended to be used from different client devices, e.g., smartphones, Desktop
- ❖ Our continuous authentication approach is grounded on timeout management, we also use image verification as second level security which increases the security.

CHAPTER 4

TECHNOLOGY

4.1 Language Used:

The programming language that was used in this Anomaly detection project is Java. The implementation of source code was done through Java. Java is a general-purpose programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as possible. It is intended to let application developers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.

4.1.1.HISTORY OF JAVA:

James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991. Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time. The language was initially called *Oak* after an oak tree that stood outside Gosling's office. Later the project went by the name *Green* and was finally renamed *Java*, from Java coffee, the coffee from Indonesia. Gosling designed Java with a C/C++-style syntax that system and application programmers would find familiar.

Sun Microsystems released the first public implementation as Java 1.0 in 1996. It promised Write Once, Run Anywhere (WORA) functionality, providing no-cost run-times on popular platforms. Fairly secure and featuring configurable security, it allowed network- and file-access restrictions. Major web browsers soon incorporated the ability to run Java applets within web pages, and Java quickly became popular. The Java 1.0 compiler was re-written in Java by Arthur van Hoff to comply strictly with the Java 1.0 language specification. With the advent of Java 2 (released initially as J2SE 1.2 in December 1998 – 1999), new versions had multiple configurations built for different types of platforms. J2EE included technologies and APIs for enterprise applications typically run in server environments, while J2ME featured APIs optimized for mobile

applications. The desktop version was renamed J2SE. In 2006, for marketing purposes, Sun renamed new J2 versions as *Java EE*, *Java ME*, and *Java SE*, respectively.

4.1.2 Features of Java

Java provides lot of Features that are listed below:

➤ **Simple:**

Java is very easy to learn, and its syntax is simple, clean and easy to understand.

➤ **.Object-Oriented:**

Java is an object-oriented programming language. Everything in Java is an object. Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behaviour.

➤ **Platform Independent:**

Java is platform independent because it is different from other languages like C, C++, etc. which are compiled into platform specific machines while Java is a write once, run anywhere language. A platform is the hardware or software environment in which a program run.

➤ **Secured:**

Java is best known for its security. With Java, we can develop virus-free-systems.

➤ **Robust:**

Robust simply means strong. Java is robust because:

- It uses strong memory management.
- There is a lack of pointers that avoids security problems.

- There is automatic garbage collection in java which runs on the Java Virtual Machine to get rid of objects which are not being used by a Java application anymore.
- There are exception handling and the type checking mechanism in Java. All these points make Java robust.

➤ **Architecture-Neutral:**

Java is architecture neutral because there are no implementation dependent features, for example, the size of primitive types is fixed.

➤ **Portable:**

Java is portable because it facilitates you to carry the Java byte code to any platform. It doesn't require any implementation.

➤ **High Performance:**

Java is faster than other traditional interpreted programming languages because Java byte code is "close" to native code. It is still a little bit slower than a compiled language (e.g., C++). Java is an interpreted language that is why it is slower than compiled languages, e.g., C, C++, etc.

➤ **Distributed:**

Java is faster than other traditional interpreted programming languages because Java bytecode is "close" to native code. It is still a little bit slower than a compiled language (e.g., C++). Java is an interpreted language that is why it is slower than compiled languages, e.g., C, C++, etc.

➤ **Multi-threaded:**

A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area. Threads are important for multi-media, Web applications, etc

➤ **Dynamic:**

Java is a dynamic language. It supports dynamic loading of classes. It means classes are loaded on demand. It also supports functions from its native languages, i.e., C and C++.Java supports dynamic compilation and automatic memory management (garbage collection).

4.1.3 Applications of Java

Various types of applications that runs on Java are as follows:

➤ **Desktop GUI Applications:**

Java provides GUI development through various means like Abstract Windowing Toolkit (AWT), Swing and JavaFX. While AWT contains a number of pre-constructed components such as menu, button, list, and numerous third-party components, Swing, a GUI widget toolkit, additionally provides certain advanced components like trees, tables, scroll panes, tabbed panel and lists. JavaFX, a set of graphics and media packages, provides Swing interoperability, 3D graphic features and self-contained deployment model which facilitates quick scripting of Java applets and applications.

➤ **Mobile Applications:**

Java Platform, Micro Edition (Java ME or J2ME) is a cross-platform framework to build applications that run across all Java supported devices,

including feature phones and smart phones. Further, applications for Android, one of the most popular mobile operating systems, are usually scripted in Java using the Android Software Development Kit (SDK) or other environments.

➤ **Embedded Systems:**

Embedded systems, ranging from tiny chips to specialized computers, are components of larger electromechanical systems performing dedicated tasks. Several devices, such as SIM cards, blue-ray disk players, utility meters and televisions, use embedded Java technologies. According to Oracle, 100% of Blu-ray Disc Players and 125 million TV devices employ Java.

➤ **Web Applications:**

Java provides support for web applications through Servlets, Struts or JSPs. The easy programming and higher security offered by the programming language has allowed a large number of government applications for health, social security, education and insurance to be based on Java. Java also finds application in development of e-commerce web applications using open-source e-commerce platforms, such as Broadleaf.

➤ **Web Servers and Application Servers:**

The Java ecosystem today contains multiple Java web servers and application servers. While Apache Tomcat, Simple, Jo!, Rimfaxe Web Server (RWS) and Project Jigsaw dominate the web server space, WebLogic, WebSphere, and Jboss EAP dominate commercial application server space.

➤ **Enterprise Applications:**

Java Enterprise Edition (Java EE) is a popular platform that provides API and runtime environment for scripting and running enterprise software, including network applications and web-services. Oracle claims Java is running in 97% of

enterprise computers. The higher performance guarantee and faster computing in Java has resulted in high frequency trading systems like Murex to be scripted in the language. It is also the backbone for a variety of banking applications which have Java running from front user end to back server end.

➤ **Scientific Applications:**

Java is the choice of many software developers for writing applications involving scientific calculations and mathematical operations. These programs are generally considered to be fast and secure, have a higher degree of portability and low maintenance. Applications like MATLAB use Java both for interacting user interface and as part of the core system.

4.2 Java Installation Procedure:

Step.1: There are Several Java platforms .We will install the Java standard Edition 6(SE) from official oracle site.

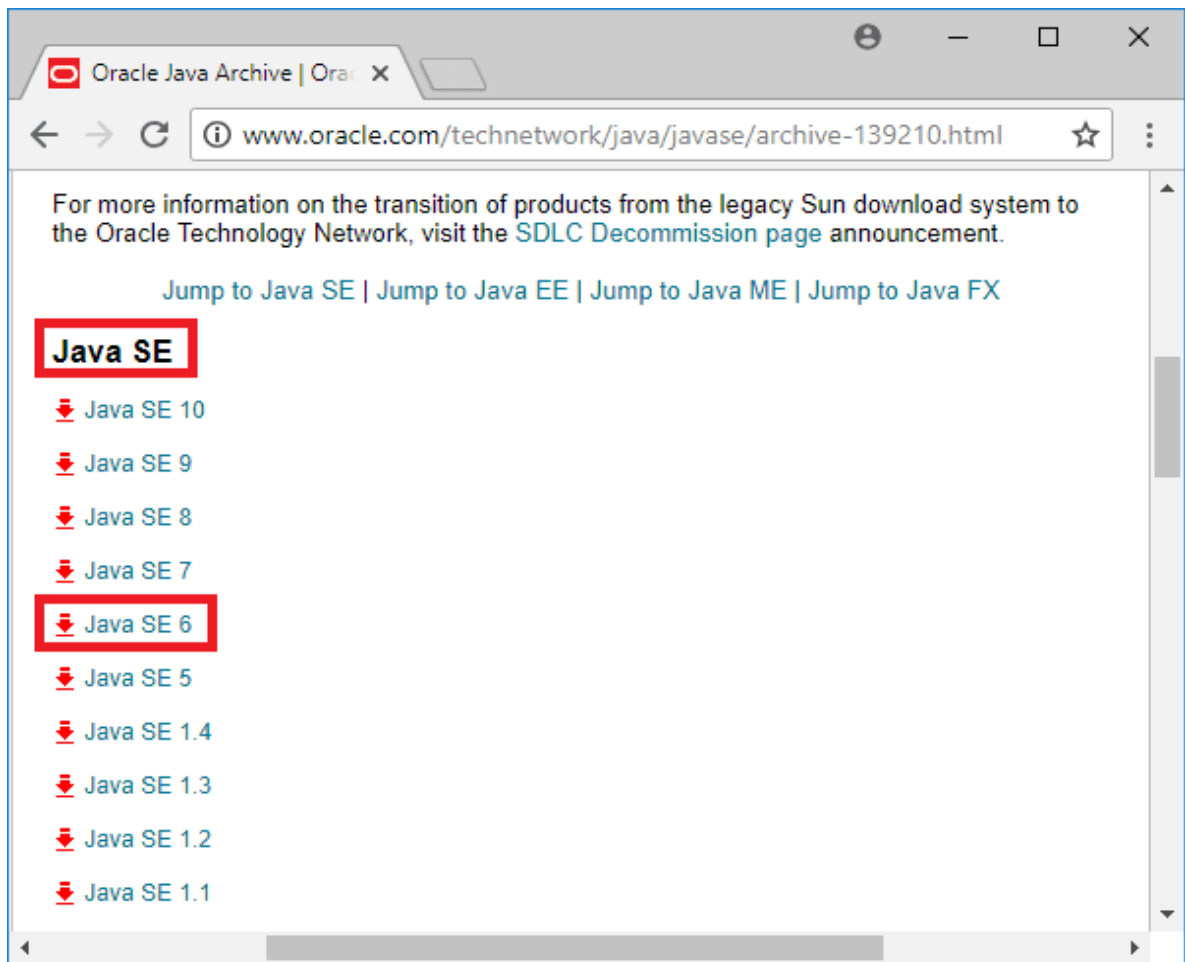


Figure 4.2.1 Select java SE6

Now, select the Java SE 6 to download.

Step-2: Verify your windows bit version and click on the corresponding link:

- For 32-bit = Windows x86 installer
- For 64-bit = Windows x64 installer

In this guide, we will download the 64-bit installer

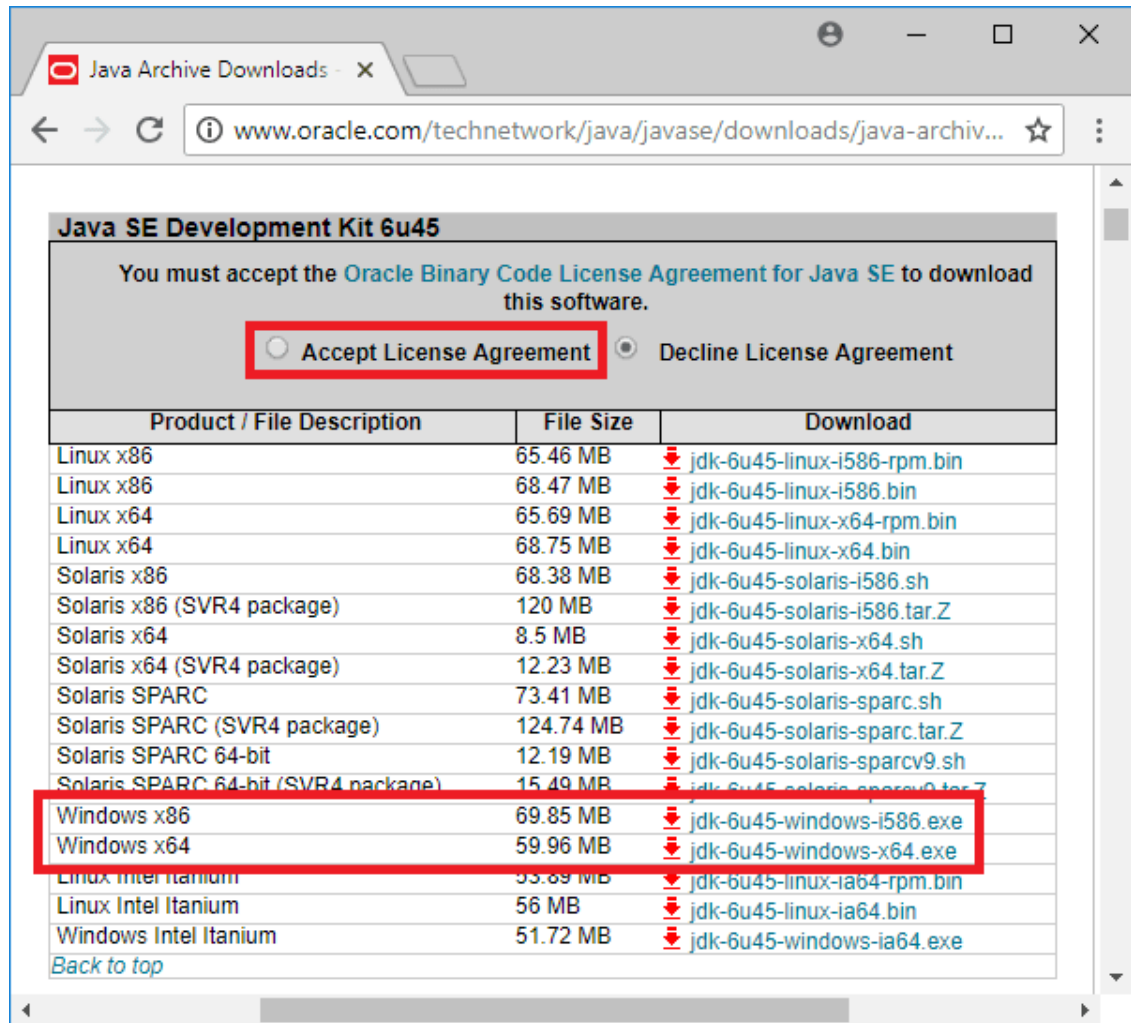


Figure 4.2.2 Selecting Windows Version

- To download archived JDK versions you need an Oracle account. Create a new account or sign in with an existing one.
- Wait for the download to complete.

Step-3:

- Open the location of the downloaded executable
- Double – click it to run the Installer
- The JDK installer will start. Click the Next Button



Figure 4.2.3 Installation Wizard

- You can change the installation location by clicking on the Change... button.
- We keep the default install location of C:\Program Files\Java\jdk1.6.0_45. From now on we will refer to this directory as [JAVA_INSTALL_DIR].
- We will not install the public JRE as the JDK development tools already include a private JRE.
- Select the Public JRE dropdown and click on this feature will not be available. As shown below.

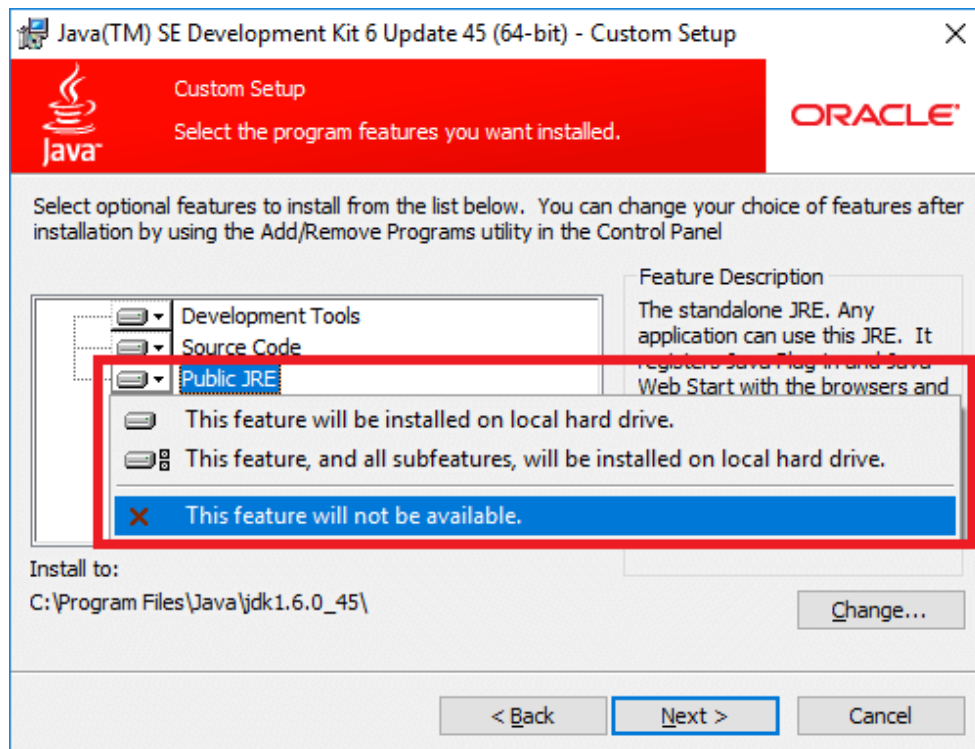


Figure 4.2.4 Change Location

- Click Next to start the installation.
- The JDK installation will now start.
- A progress bar shows the various steps that are executed.

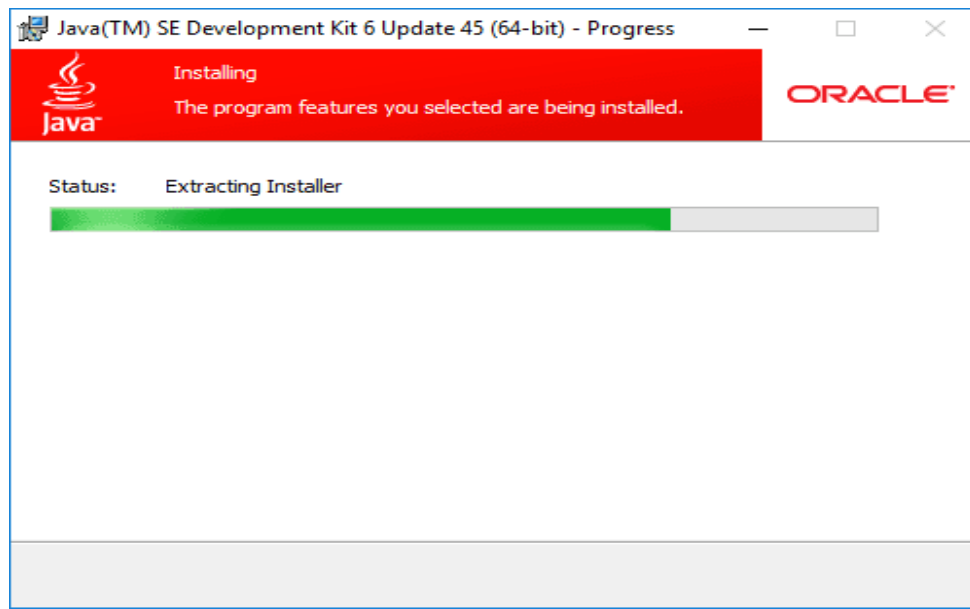


Figure 4.2.5 Extracting Installer

- Once the installation is complete, click Close.

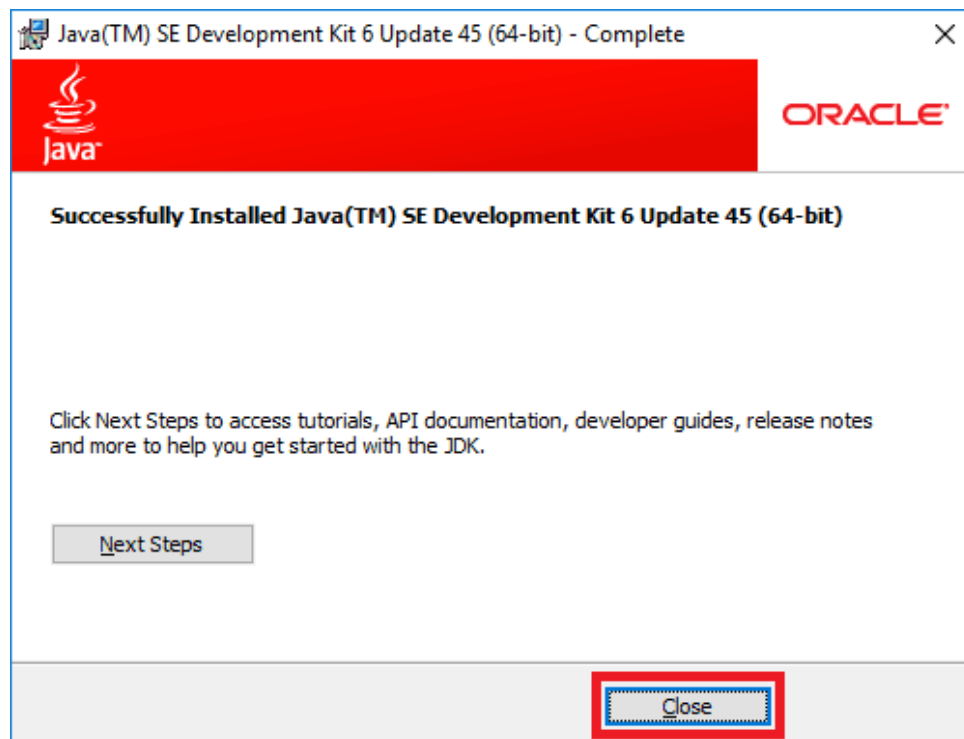


Figure 4.2.6 Successfully Installed

Step-4:

- We need to set up an environment variable that will point to our JDK installation.
- Click on the search button. Then type “env” (without quotes).

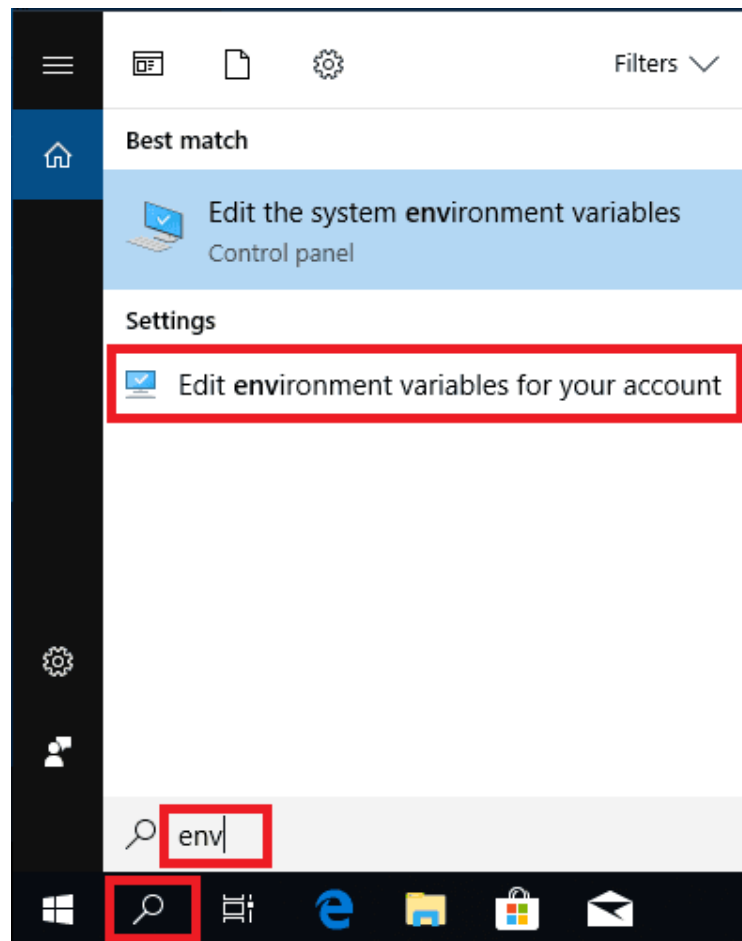


Figure 4.2.7 Opening Environment Variables

- Wait for the environment variables window to open.
- Click on New...

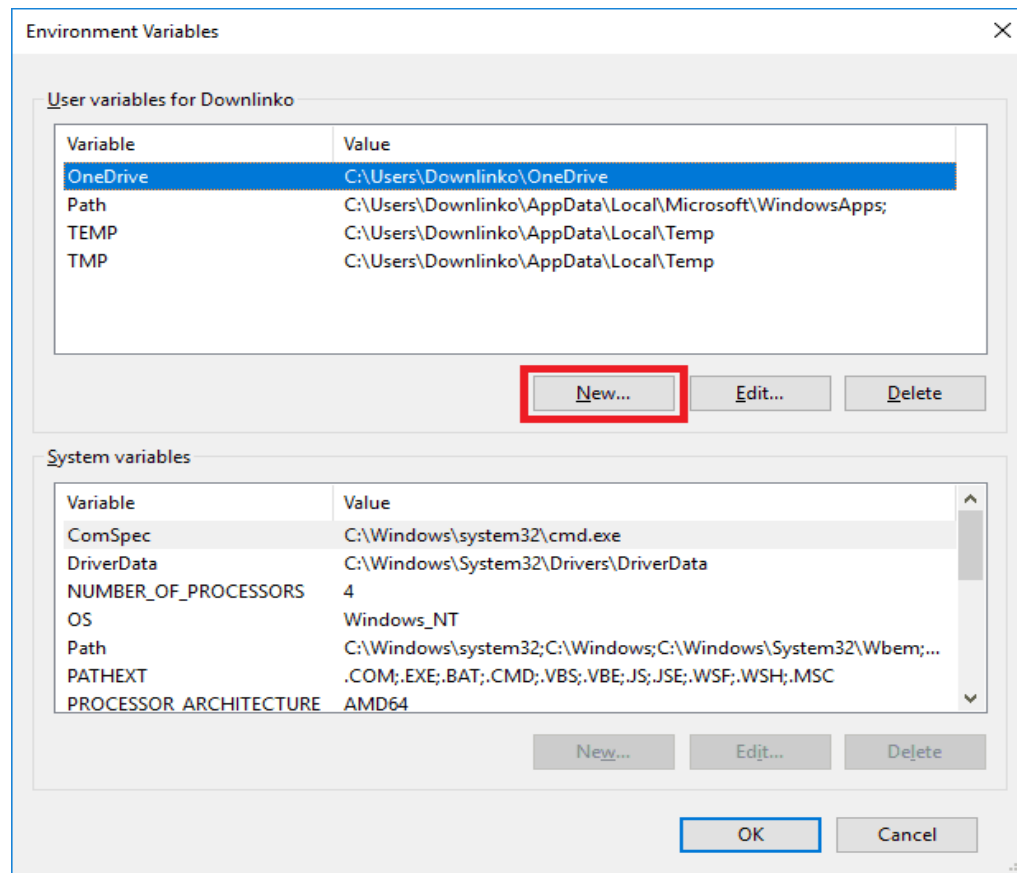


Figure 4.2.8: Click New Environment Variable

- Enter “**JAVA_HOME**” as variable name. Enter the [JAVA_INSTALL_DIR] as variable value.
- In this tutorial, the Java installation directory is C:\Program Files\Java\jdk1.6.0_45.
- Click OK.

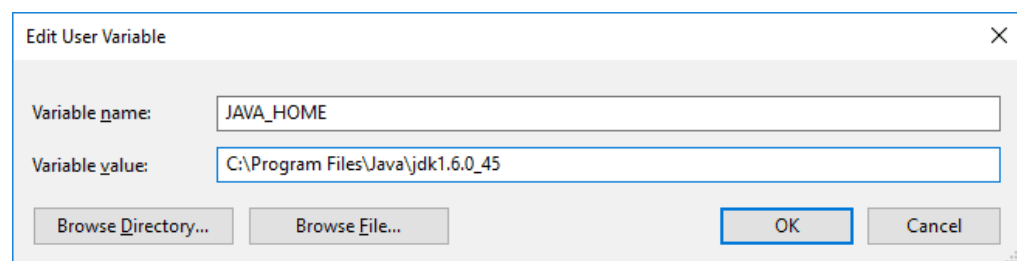


Figure 4.2.9: Enter Variable name and Value

- Next, we need to configure the PATH environment variable so we can run Java from a command prompt.
- Select the Path variable. Click on Edit....

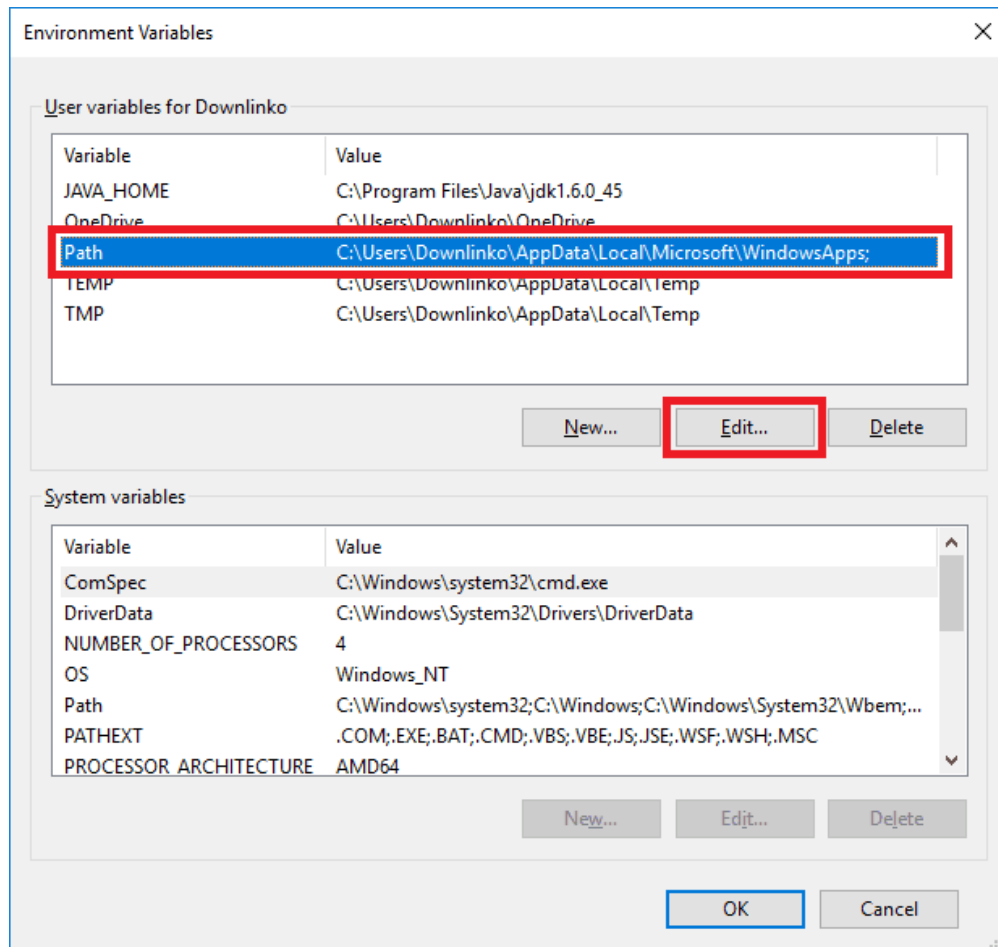


Figure 4.2.10 : Select Path Variable

- Click on New and type “%JAVA_HOME%\bin” as shown below.
- Click OK.

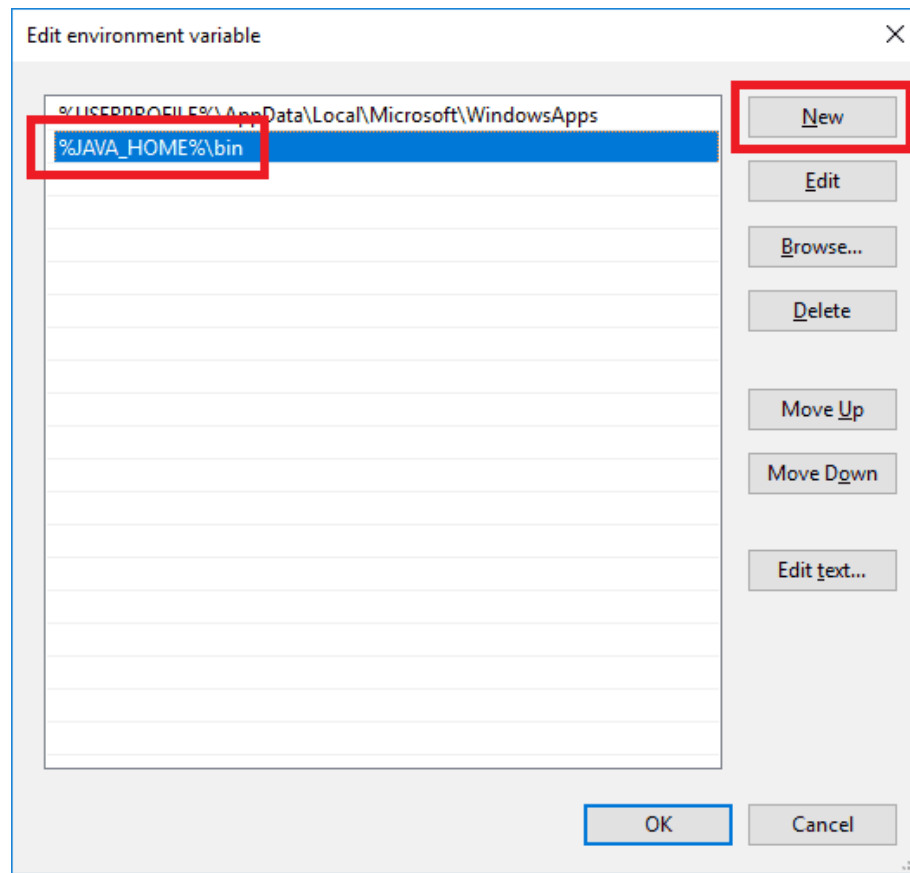


Figure 4.2.11 : Edit Environment Variables

- Click OK once more to close the environment variables window.

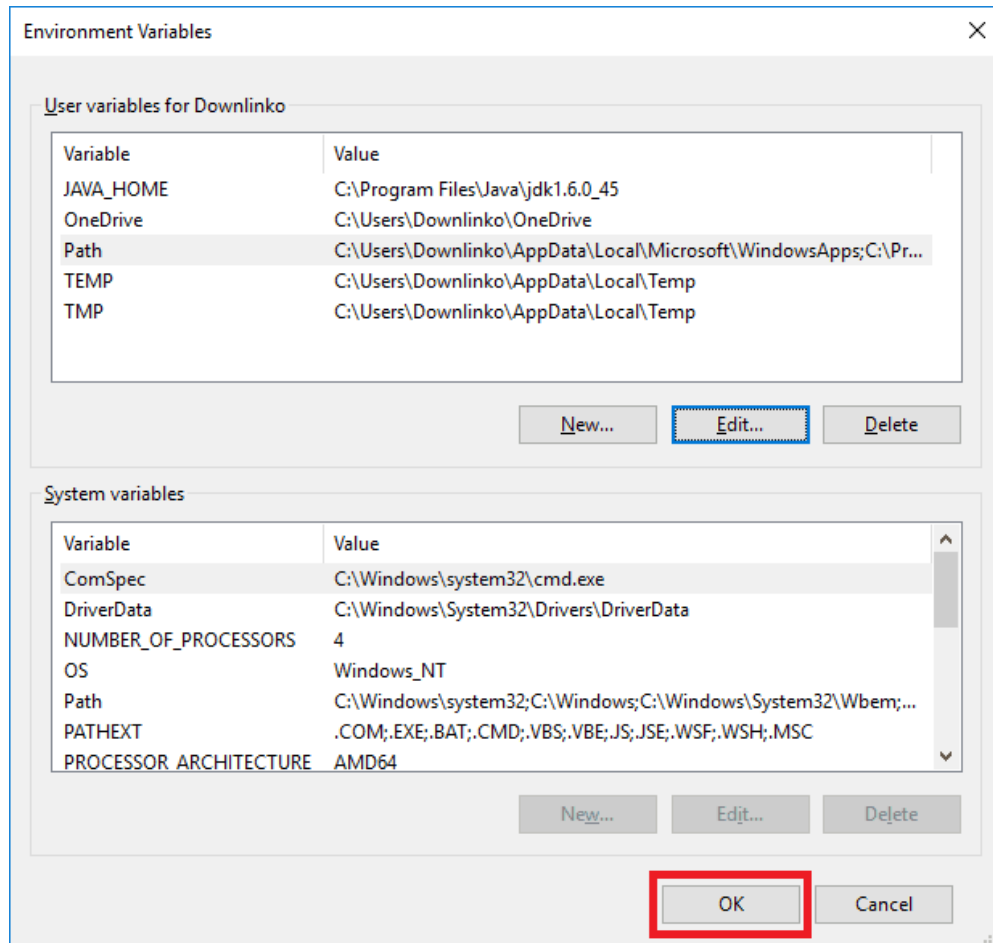


Figure 4.2.12 : Close Environment variable

- If a Path variable does not exist you need to create it. Use “Path” as variable name and “%JAVA_HOME%\bin” as variable value.

Step-5:

- Let's test the setup.
- Click on the search button. Then type “cmd” (without quotes).
- Click on the Command Prompt shortcut.

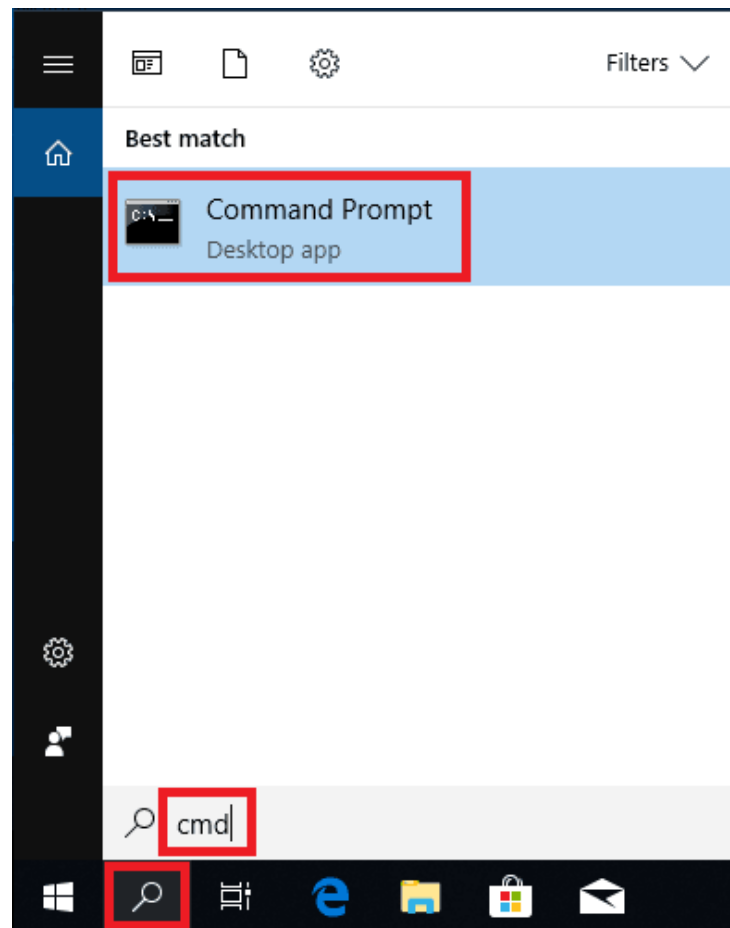


Figure 4.2.13 : Command Prompt

- Wait for the command prompt to open.
- Type “`java -version`” and press ENTER.

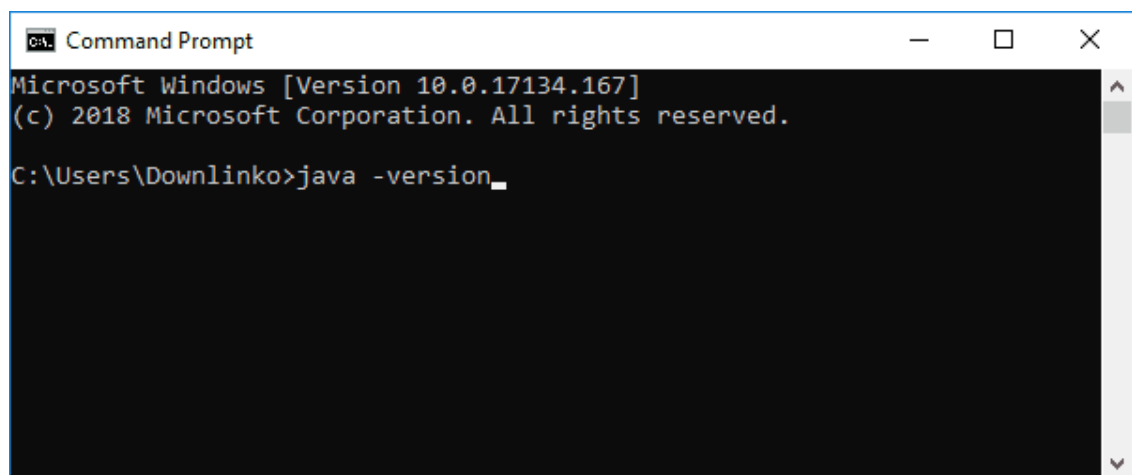
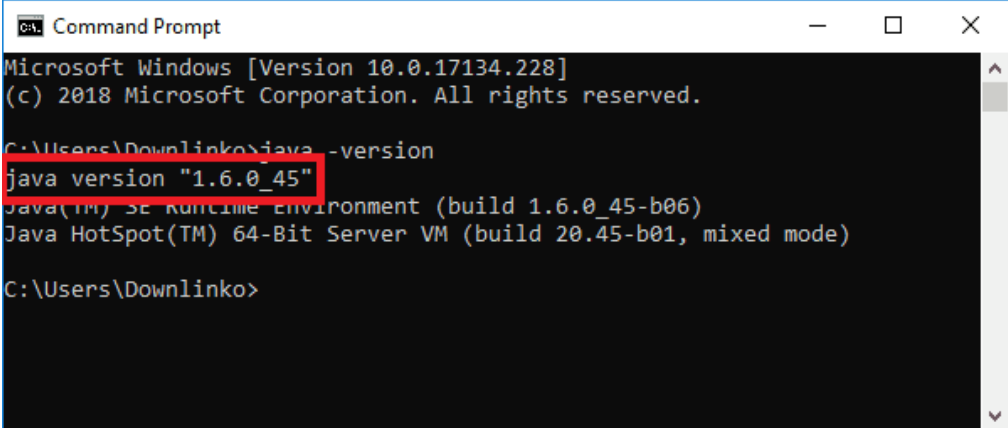


Figure 4.2.14: Java Version

- The above command prints the installed JDK version: 1.6.0_45



```
Microsoft Windows [Version 10.0.17134.228]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Downlinko>java -version
java version "1.6.0_45"
Java(TM) SE Runtime Environment (build 1.6.0_45-b06)
Java HotSpot(TM) 64-Bit Server VM (build 20.45-b01, mixed mode)

C:\Users\Downlinko>
```

Figure 4.2.15: Successfully Installed

- Java is successfully installed in your system.

4.3 Java IDE:

A Java IDE (for Integrated Development Environment) is a software application which enables users to more easily write and debug Java programs. Many IDEs provide features like syntax highlighting and code completion, which help the user to code more easily.

- In our project, we use Net Beans IDE.

4.4 Net Beans IDE:

Net Beans is an integrated development environment (IDE) for Java. Net Beans allows applications to be developed from a set of modular software components called *modules*. Net Beans runs on Windows, macOS, Linux and Solaris. In addition to Java development, it has extensions for other languages like PHP, C, C++, HTML5, and JavaScript. Applications based on Net Beans, including the Net Beans IDE, can be extended by third party developers.

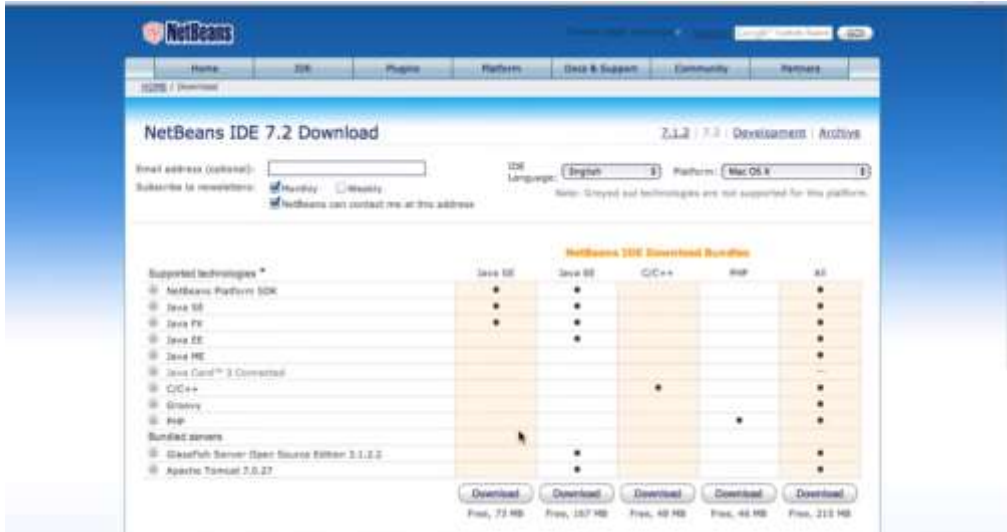


Figure 4.4.1 Install Net beans 7.2.1

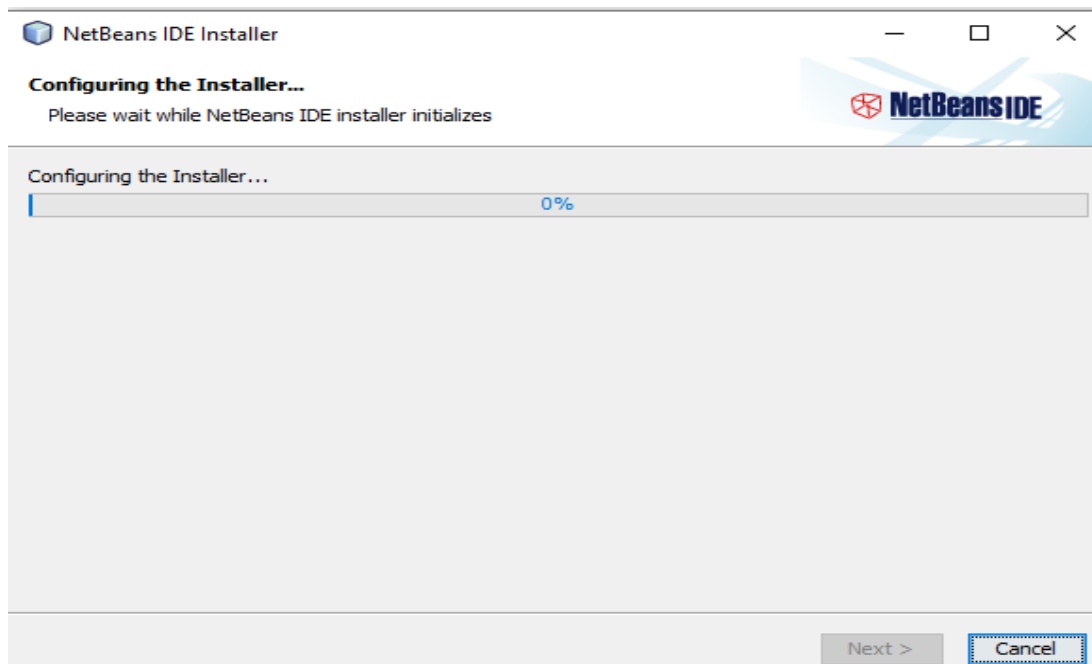


Figure 4.4.2 : Run Installer

- After the configuration, the installation will start.

Step-3: The installer will ask to select the application server to install along with the IDE . Select the first one.



Figure 4.4.3: Select Application Server to Install

- Accept the license agreement and click next.

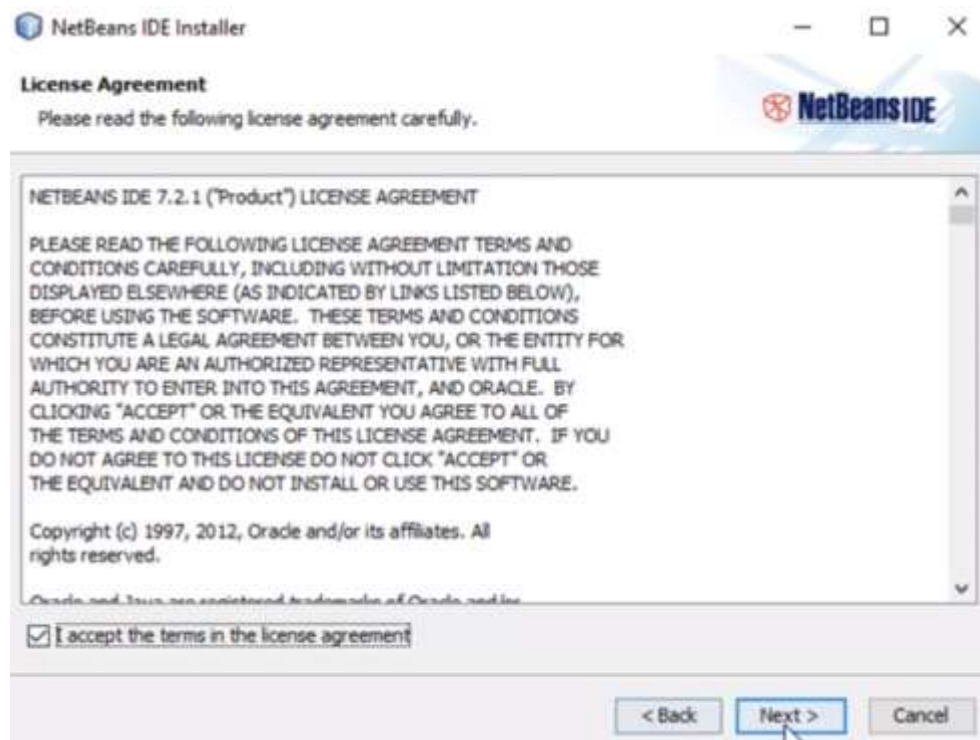


Figure 4.4.4 : Accept License Agreement

- Also accept the license agreement for installing JUnit.

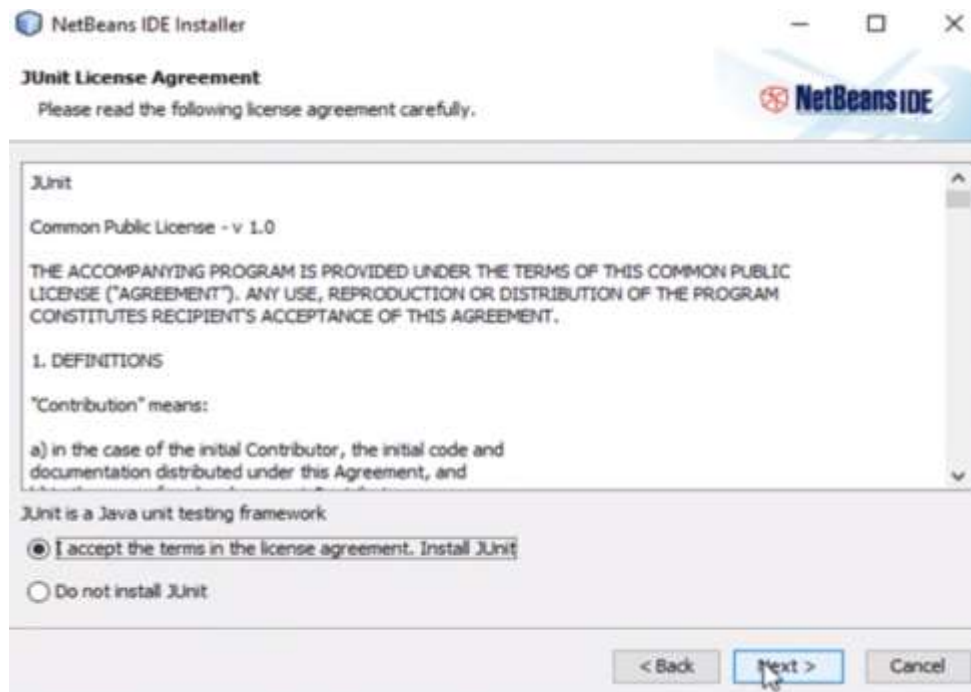


Figure 4.4.5 : Lincense agreement for JUnit

Step-4: Choose the destination folder to install the NetBeans.



Figure 4.4.6 : Choose destination folder for Netbeans

- Choose the destination folder for installing the GlassFish

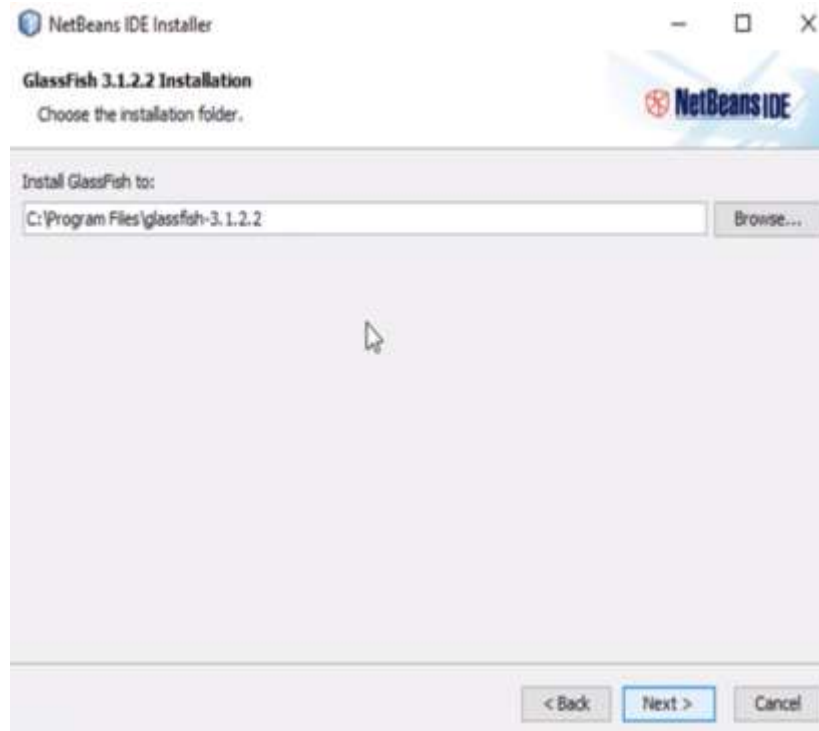


Figure 4.4.7 : Destination folder for glass fish

Step-5: Click the install button to start the installation

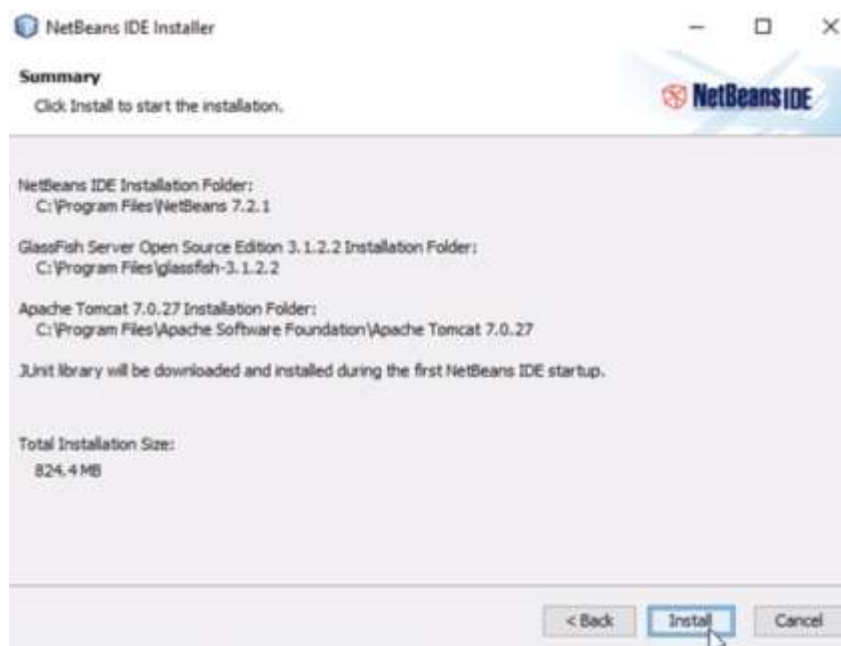


Figure 4.4.8: Start Installation

- The progress of the installation will be displayed. Wait for the completion of the installation.



Figure 4.4.9 : Wait to Install

- Click finish to complete the installation.

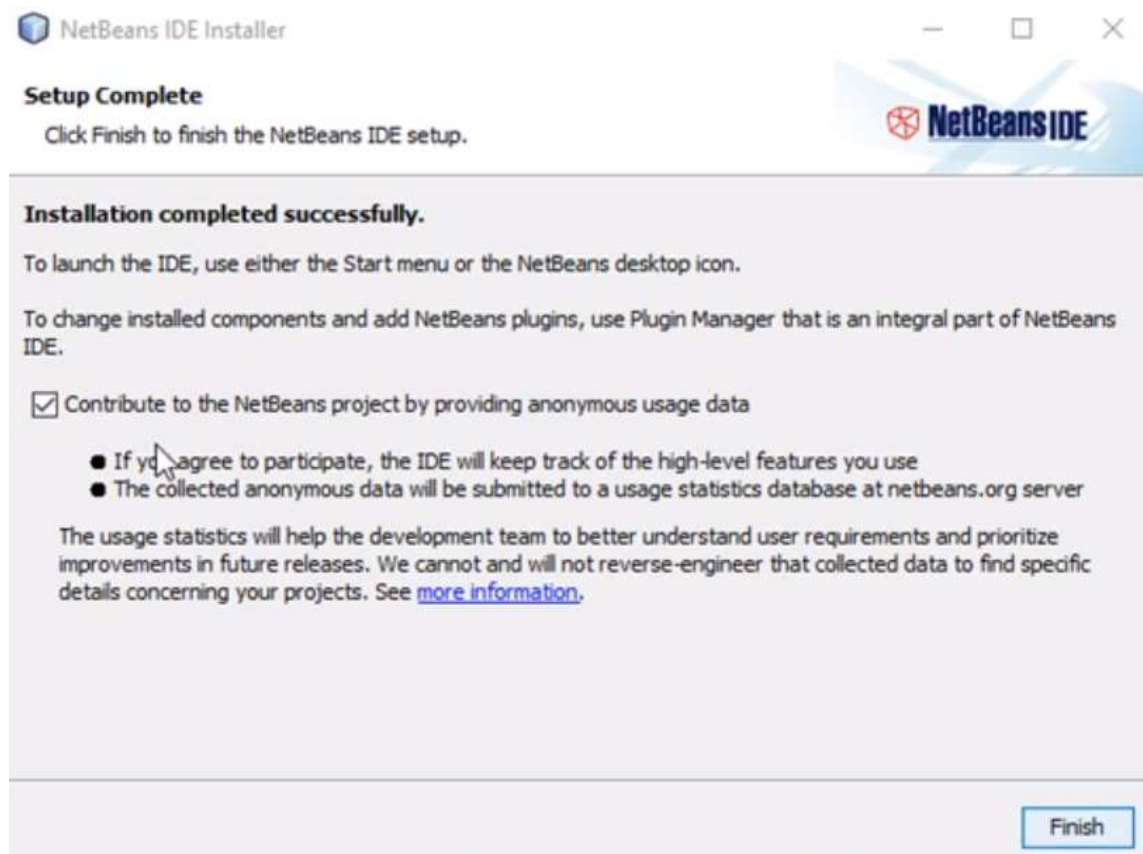


Figure 4.4.10: Installation Completed

- Net Beans is successfully installed in your System.

CHAPTER 5

DESIGN

5.1 UML Introduction:

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective.

UML is specifically constructed through two different domains, they are:

- UML Analysis modelling, this focuses on the user model and structural model views of the systems.
- UML Design modelling, which focuses on the behavioural modelling, implementation modelling and environmental model views.

5.2 Usage of UML in Project :

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time to the market. These techniques include component technology, visual programming, patterns and frameworks. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The UML was designed to respond to these needs. Simply, systems design refers to the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements which can be done easily through UML diagrams.

5.3 Goals of UML:

The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.
- Support higher level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

5.4 Diagrams:

5.4.1 Data Flow Diagram:

- The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
- The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
- DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

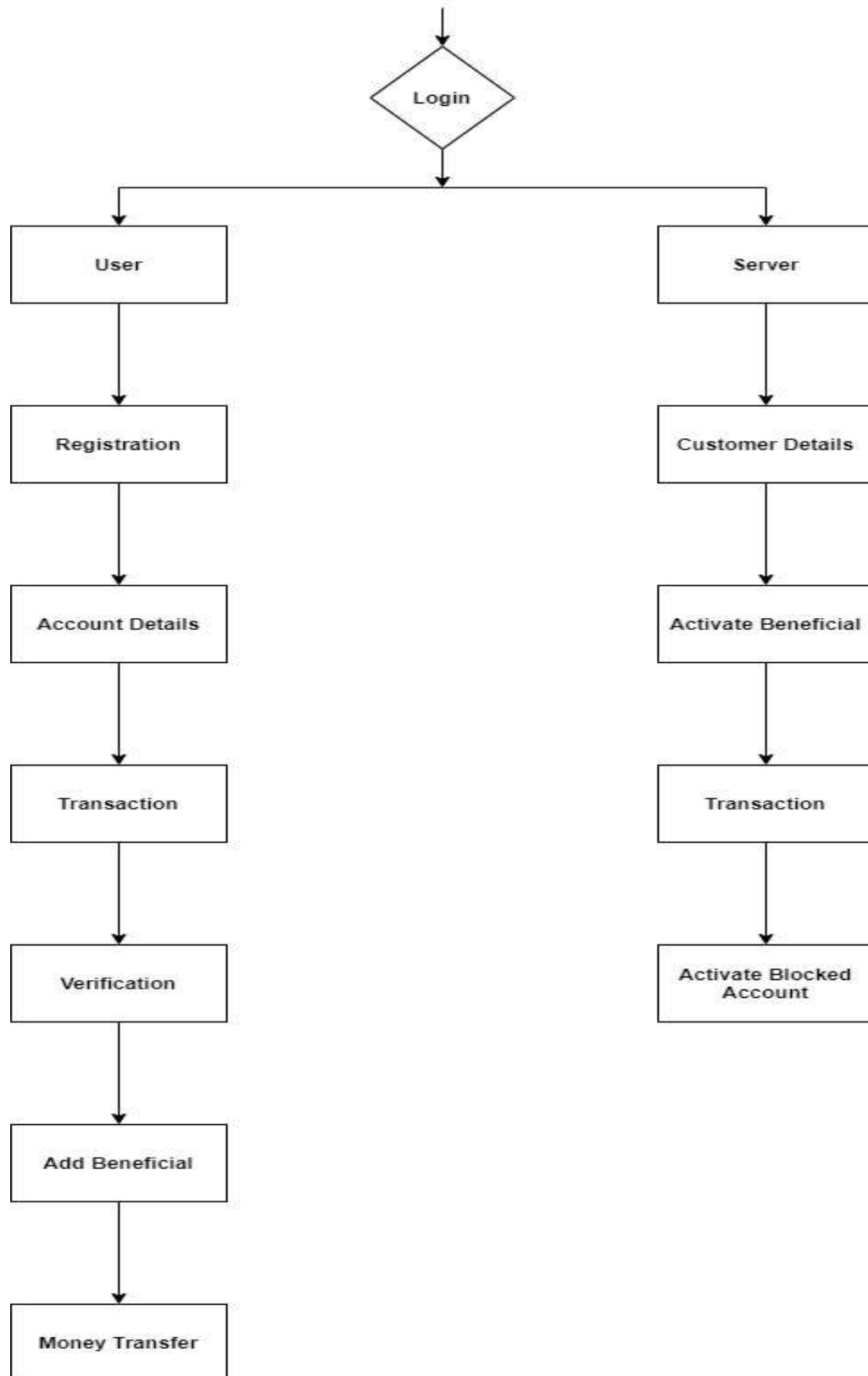


Figure 5.4.1 :Data Flow Diagram

5.4.2 Use Case Diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

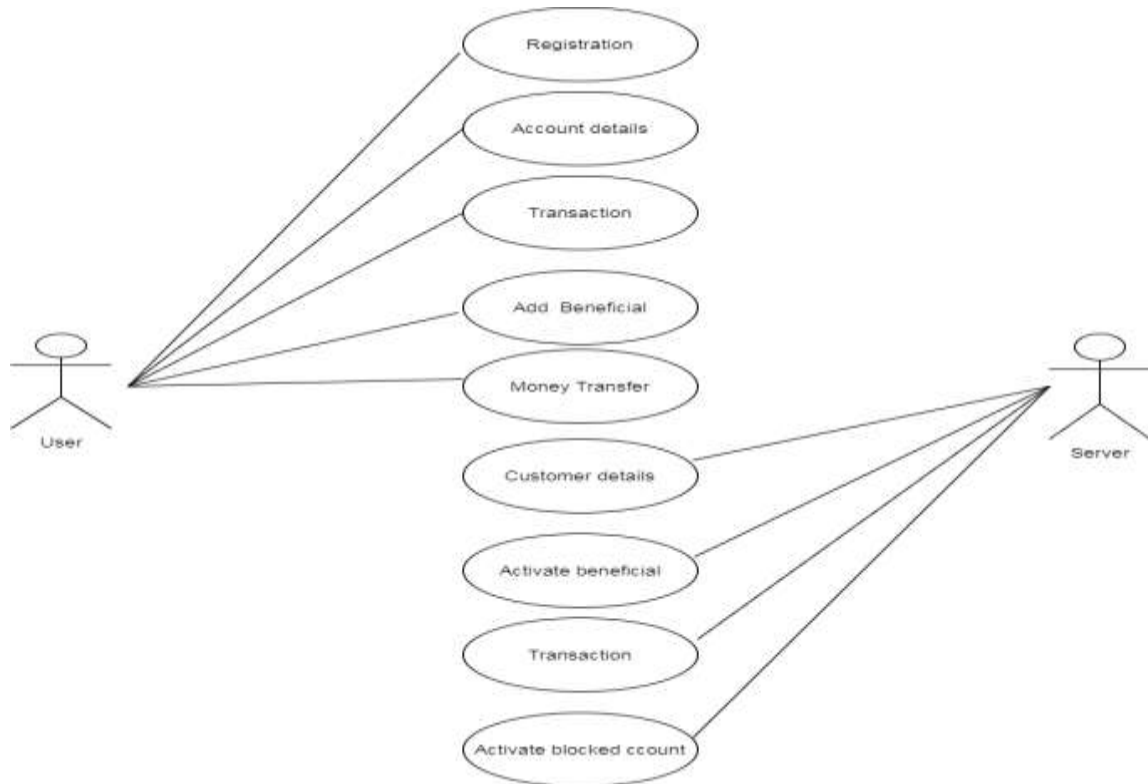


Figure 5.4.2.1 : Use Case Diagram

5.4.3 Class Diagram:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

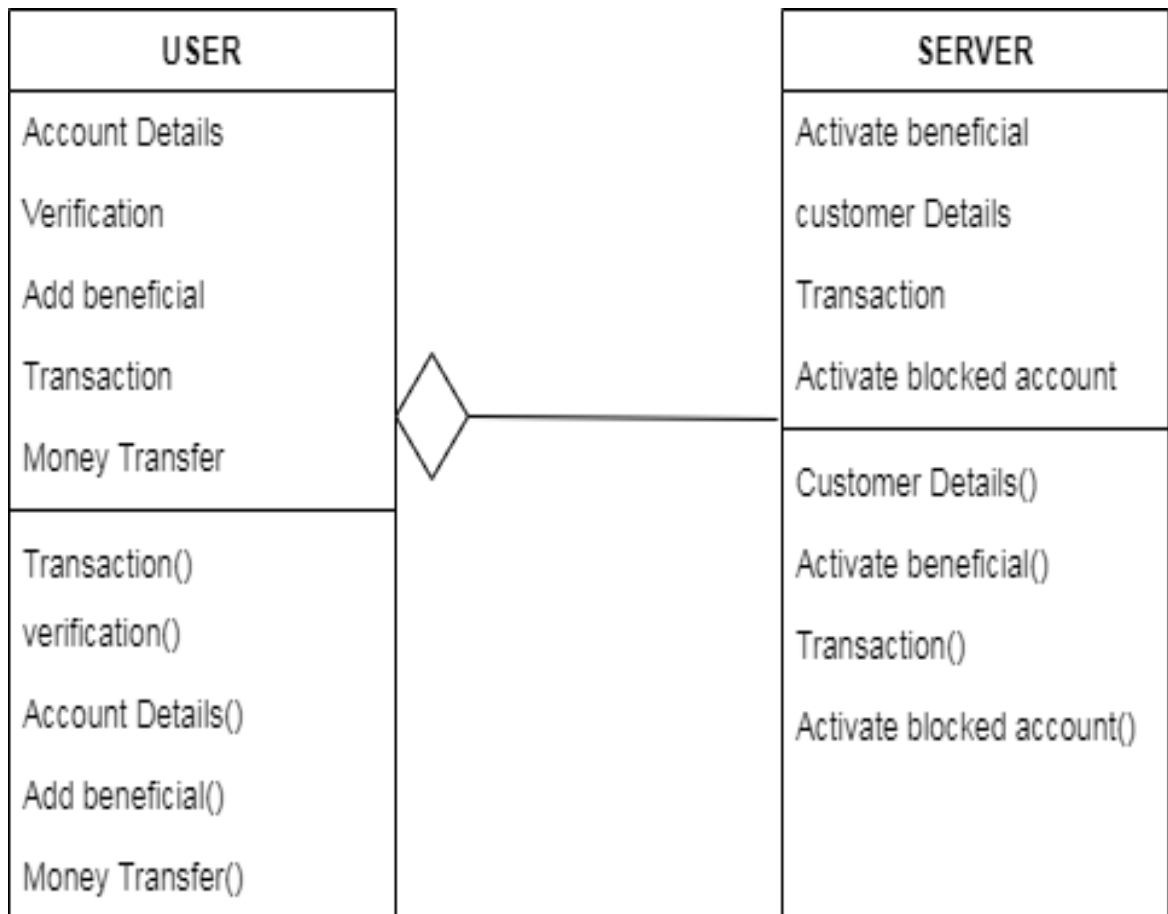


Figure 5.4.3.1: Class Diagram

5.4.4 Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

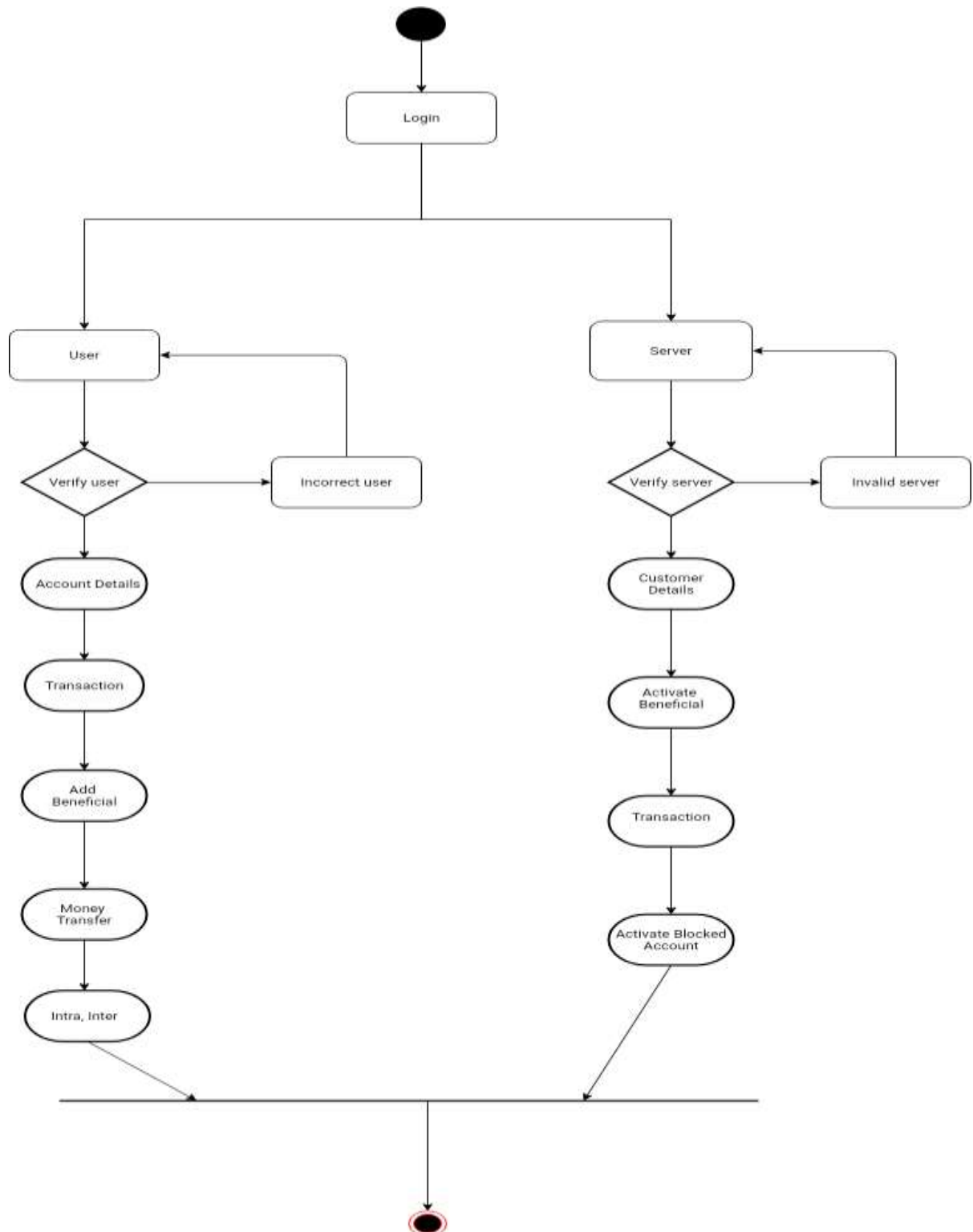


Figure 5.4.4.1 : Activity Diagram

CHAPTER 6

IMPLEMENTATION

6.1 MODULES

The following are the modules :

- ❖ System Model
- ❖ Authentication Server
- ❖ Image Verification
- ❖ Continuous Authentication

6.2 MODULES DESCRIPTION

6.2.1 System Model:

- In this module, we create the System model to evaluate and implement our proposed system. It can authenticate to web services, ranging from services with strict security requirements as online banking services to services with reduced security requirements as forums or social networks. We explain the usage of the authentication service by discussing the sample application scenario, where a user u wants to log into an online banking service.
- "User Id" refers to the identity of the user obtained from the Bank for the purpose of logging into the Internet Banking facility provided by the Bank.
- "Login Password" is a unique and randomly generated password known only to the customer, which can be changed by the user to his/her convenience. This is a means of authenticating the user ID for logging into Internet Banking. It is encrypted and decrypted using AES.
- "Transaction Password" is a unique and randomly generated password known only to the customer, which can be changed to his/her convenience. This is a means of authentication required to be provided by the customer for putting through the transaction in his/her/their/its accounts with Bank through Internet

Banking. While User ID and Password are for valid access into the internet application, giving valid Transaction Password is for authentication of transaction/requests made through internet. It is encrypted and decrypted using AES .

6.2.2 Authentication Server:

- In Internet banking as with traditional banking methods, security is a primary concern. Server will take every precaution necessary to be sure your information is transmitted safely and securely. The latest methods in Internet banking system security are used to increase and monitor the integrity and security of the system.
- The Server maintains the functionality:
 - Customer Details
 - Activation of Beneficiary
 - Transaction Details
 - Activate Blocked Account
- Customer details displays all the customer details, activation of beneficiary includes activating beneficiary , transaction details contain the transactions that has to be transferred, activate blocked account provide the right to activate the blocked account.

6.2.3 Image Verification:

- In this module, we verify the name of image that is given in registration ,if it matches it opens otherwise the user page will not open.
- This act as a second level security which improves the security.
- Every time session time out happens the user has to enter account password and should also verify image.

6.2.4 Continuous Authentication:

- A secure protocol is defined for perpetual authentication through continuous user verification. The protocol determines adaptive timeouts based on the quality, frequency acquired from the user.
- The idea behind the execution of the protocol is that the client continuously and transparently acquires and transmits evidence of the user identity to maintain access to a web service. The main task of the proposed protocol is to create and then maintain the user session adjusting the session timeout on the basis of the confidence that the identity of the user in the system is genuine.

6.3 Outputs:

There are two actors user and server which includes in this application.

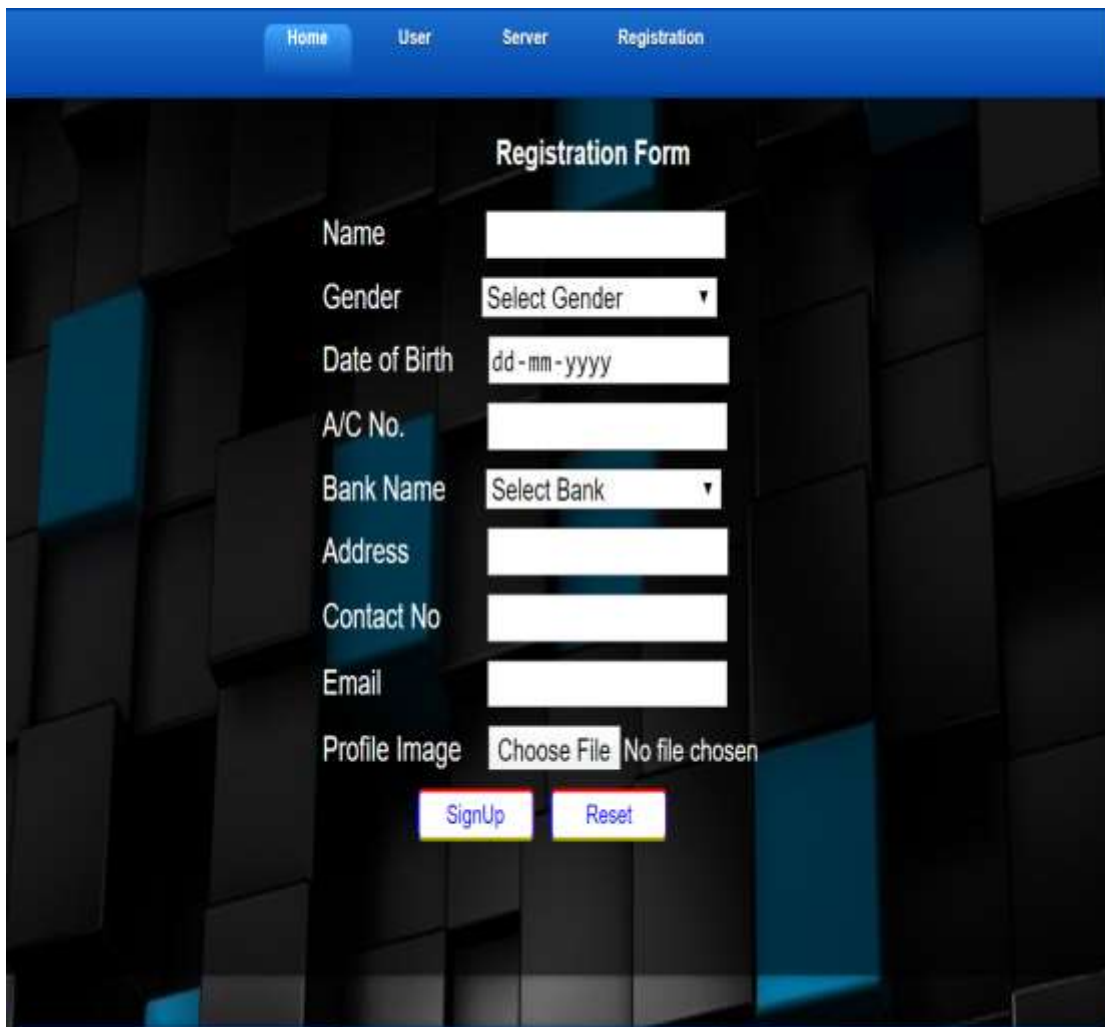
They are:

User

Server

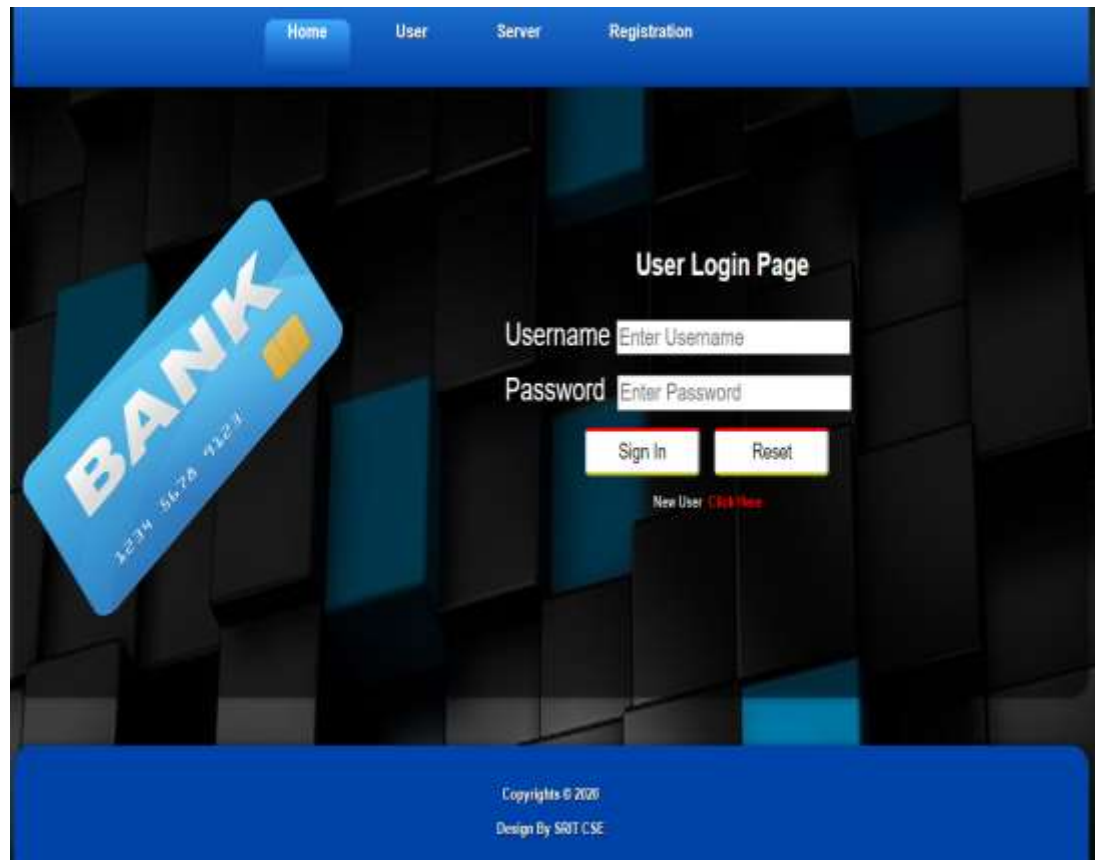
6.3.1 user :

If user is new user has to register to generate account password and transaction password. Figure 6.3.1 shows registration form for user.

A screenshot of a web application's registration form. The form is titled "Registration Form" and is set against a dark background with a 3D cube pattern. At the top, there is a blue navigation bar with four buttons: "Home", "User", "Server", and "Registration". The form fields include: "Name" (text input), "Gender" (dropdown menu with "Select Gender" selected), "Date of Birth" (text input with a date mask "dd-mm-yyyy"), "A/C No." (text input), "Bank Name" (dropdown menu with "Select Bank" selected), "Address" (text input), "Contact No" (text input), "Email" (text input), and "Profile Image" (file upload button labeled "Choose File" with the text "No file chosen" next to it). At the bottom of the form are two buttons: "SignUp" and "Reset".

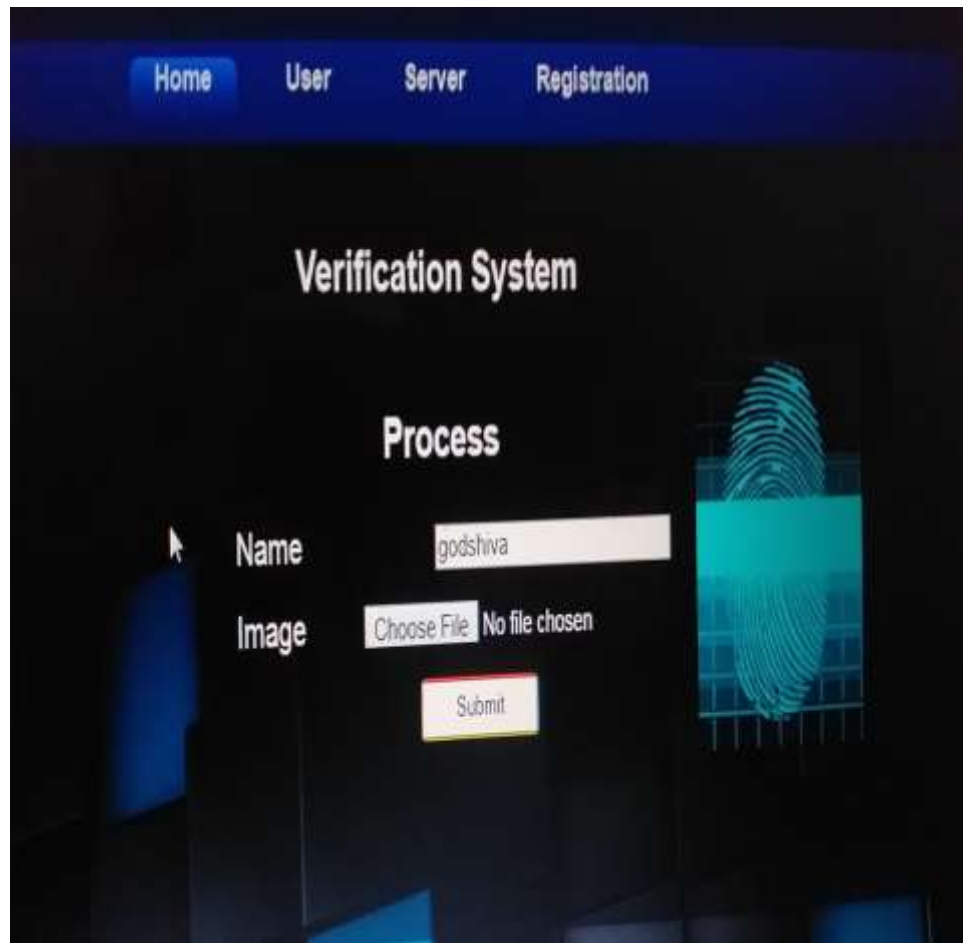
Screen 6.3.1.1: Registration form

- If user has registered, he can login using account password. Figure 6.3.1.2 shows login page for user.



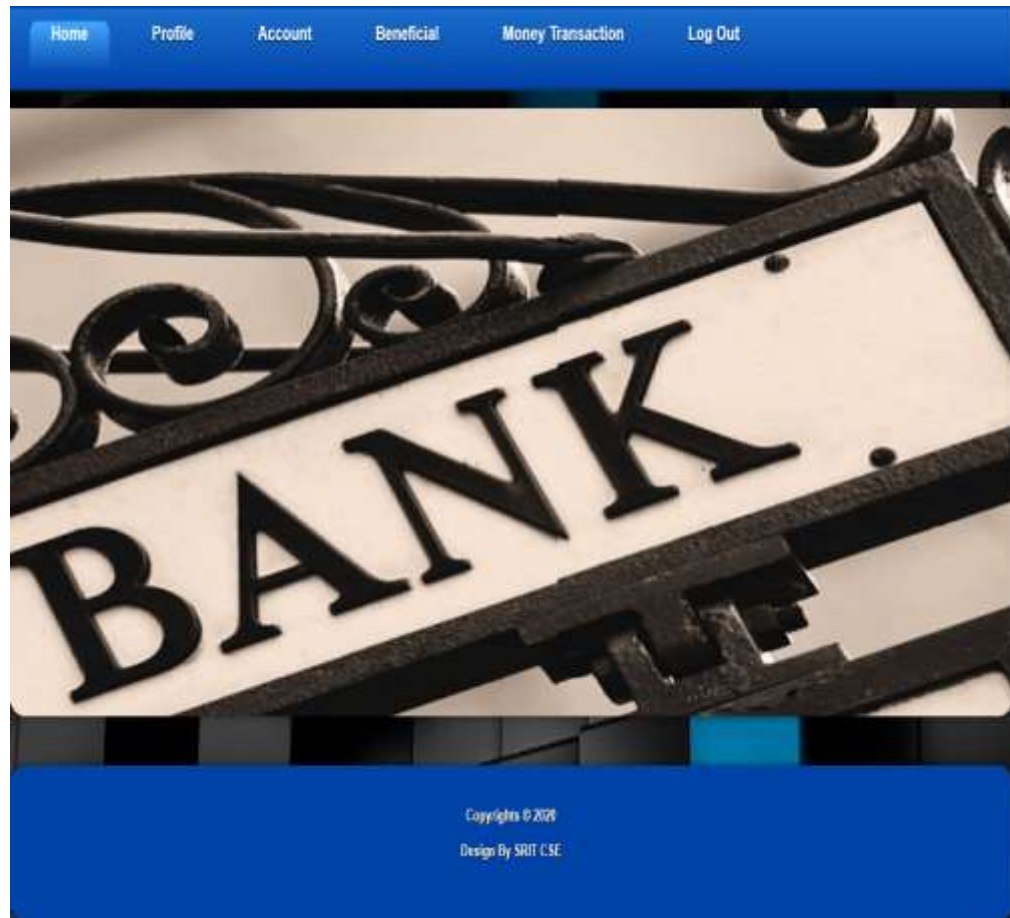
Screen 6.3.1.2 : User Login Page

- If user has entered correct account then u have to verify that is given in the registration process. Figure 6.3.1.3 shows image verification.



Screen 6.3.1.3 Image Verification

- If user enters correct account password and correct image then user page will open. Figure 6.3.1.4 shows the user page.

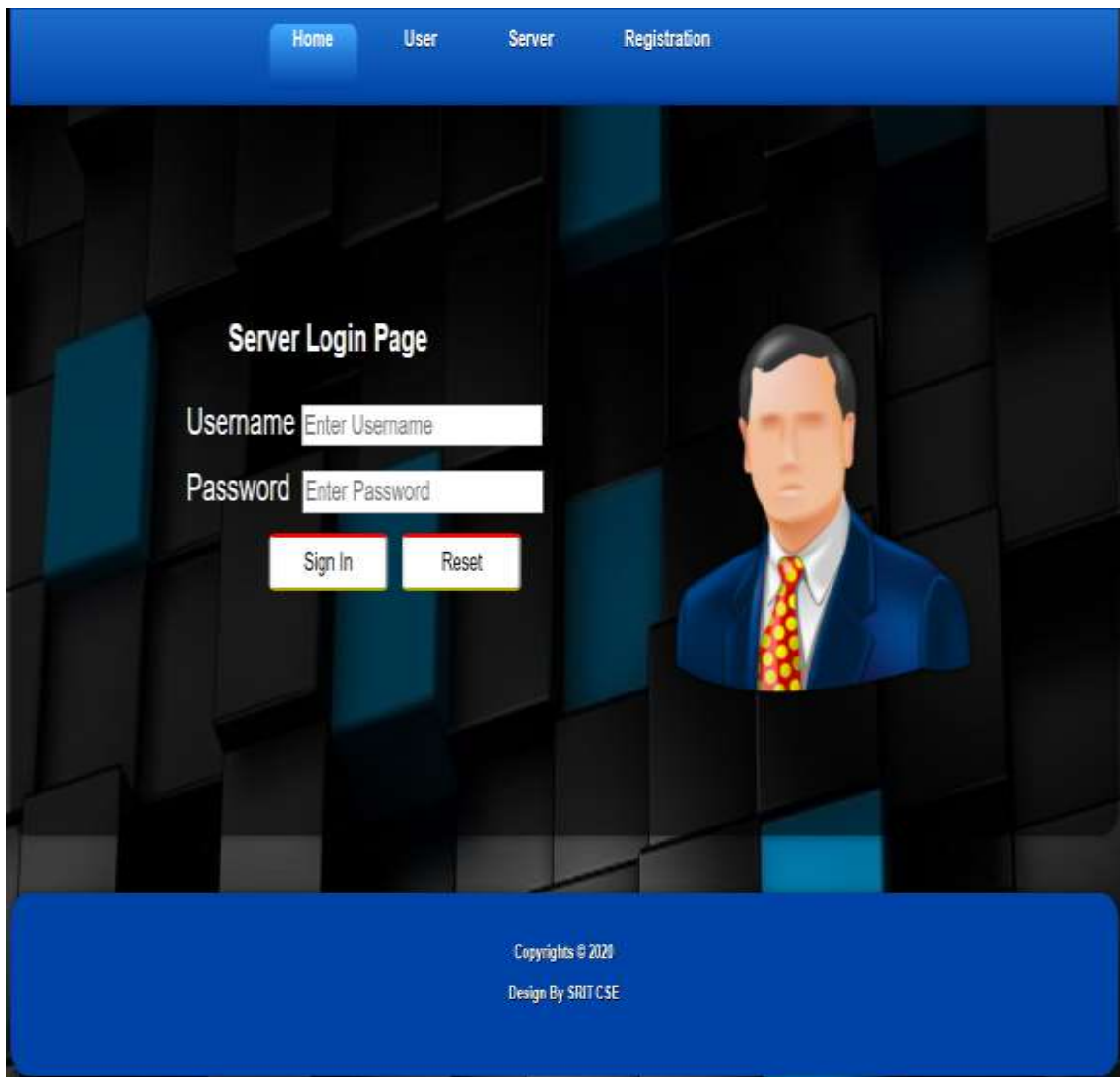


Screen 6.3.1.4 : User Page

- User page contain Profile,Account,Beneficial,Money transaction.
- Profile page displays the details of user like name,gender,date of birth,email,account number ,bank ainsname,location.
- Account contains account details,transaction and verification.
- Account details displays name,account number,bank name and available balance.
- Transaction contain transaction details and money credited details.
- Verification contain account password, transaction password and verification, which helps to change the passwords.
- Beneficial helps to add beneficial account whether it is same bank or different bank.
- Money transaction helps to transfer money by entering transaction password and otp(one time password).

6.3.2 Server:

Admin can login using username and password. Figure 6.3.2.1 shows server login page.

The screenshot displays a web application interface for a server login. At the top, a blue navigation bar contains four buttons: 'Home', 'User', 'Server', and 'Registration'. The 'Server' button is highlighted. The main content area has a dark background with a grid of blue and black squares. On the left, the text 'Server Login Page' is displayed. Below it, there are two input fields: 'Username' with the placeholder text 'Enter Username' and 'Password' with the placeholder text 'Enter Password'. To the right of these fields is a stylized illustration of a man in a blue suit and a red tie with yellow polka dots. Below the input fields are two buttons: 'Sign In' and 'Reset'. At the bottom of the page, a blue footer bar contains the text 'Copyrights © 2020' and 'Design By SRIT CSE'.

Screen 6.3.2.1 : Server Login Page

Once server login server page will display server page which contain customer details, beneficial, transaction and blocked account.



Screen 6.3.2.2 :Server Page

- Server contain customer details,it can activate blocked account,it can perform the action to grant the transaction.

6.4 AES(Advanced Encryption Standard):

The Advanced Encryption Standard (AES) is a symmetric block cipher chosen by the U.S. government to protect classified information. AES is implemented in software and hardware throughout the world to encrypt sensitive data. It is essential for government computer security, cyber security and electronic data protection. Account password and Transaction password are encrypted and decrypted using AES.

6.4.1 Working of AES:

Symmetric, also known as secret key, ciphers use the same key for encrypting and decrypting, so the sender and the receiver must both know -- and use -- the same secret key. The government classifies information in three categories: Confidential, Secret or Top Secret. All key lengths can be used to protect the Confidential and Secret level. A round consists of several processing steps that include substitution, transposition and mixing of the input plaintext to transform it into the final output of cipher text.

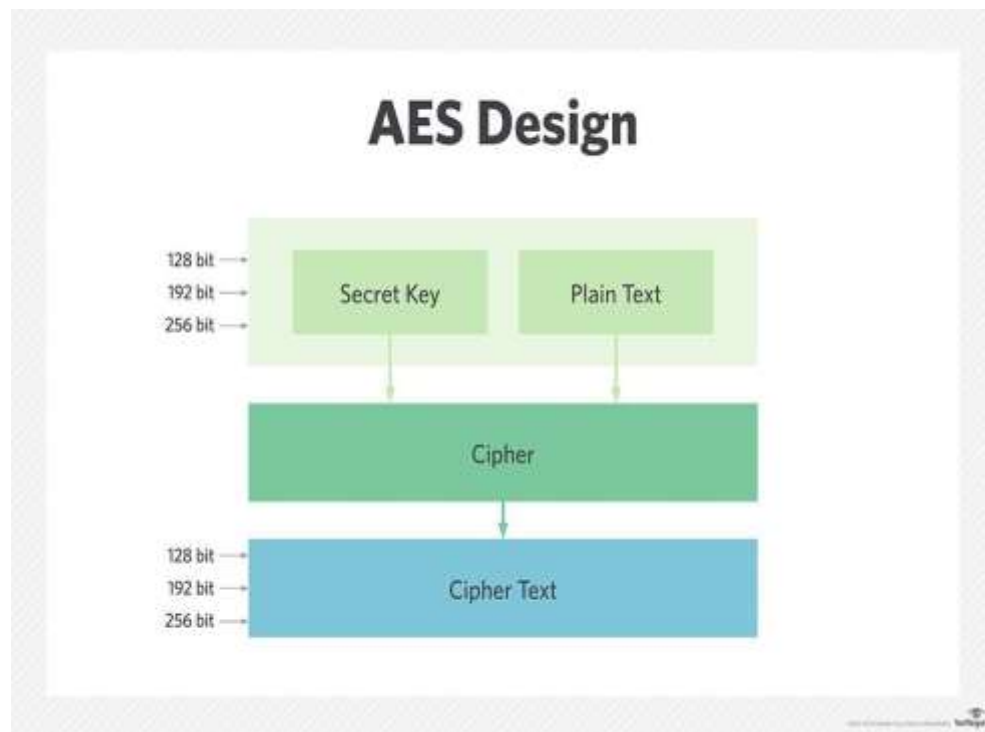


Figure 6.4.1.1 AES Design

The AES encryption algorithm defines numerous transformations that are to be performed on data stored in an array. The first step of the cipher is to put the data into an array -- after which, the cipher transformations are repeated over multiple encryption rounds.

The first transformation in the AES encryption cipher is substitution of data using a substitution table; the second transformation shifts data rows, and the third mixes

columns. The last transformation is performed on each column using a different part of the encryption key. Longer keys need more rounds to complete.

6.4.2 AES features:

NIST specified the new AES 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 AES algorithm included the following:

- **Security.** Competing algorithms were to be judged on their ability to resist attack as compared to other submitted ciphers. Security strength was to be considered the most important factor in the competition.
- **Cost.** Intended to be released on a global, nonexclusive and royalty-free basis, the candidate algorithms were to be evaluated on computational and memory efficiency.
- **Implementation.** Factors to be considered included the algorithm's flexibility, suitability for hardware or software implementation, and overall simplicity.

CHAPTER 7

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.1 TYPES OF TESTS:

7.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.1.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.1.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.1.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.1.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

7.1.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

CONCLUSION

The online service management and distribution need to preserve the sensitive information such as banking, online shopping applications. Inorder to provide an effective solution for the security of the data to ensure a completely secure migration of all their data through continuous authentication. In these services the information security and authentication is a key aim of application design. Session management in internet services is traditionally based on username and password with single time authentication, We defined a protocol that supports continuous authentication that improves security which approves authentication multiple times based on time. we also use image name verification, which is a second level verification that improves security. Finally we developed a banking application with more security.

In this presented work the online banking security is done using continuous authentication, in addition we also perform image verification which increase the security. We exploited a protocol for continuous authentication that improves security and usability of user session. The protocol performs session management based on time outs.

BIBLIOGRAPHY

- [1] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J.M. Wing , “Automated Generation and Analysis of Attack Graphs,” Proc. IEEE Symp. Security and Privacy, pp. 273-284, 2002.
- [2] D.M. Nicol, W.H. Sanders, and K.S. Trivedi, “Model-Based Evaluation: From Dependability to Security,” IEEE Trans. Dependable and Secure Computing, vol. 1, no. 1, pp. 48-65, Jan.-Mar. 2004.
- [3] T. Casey, “Threat Agent Library Helps Identify Information Security Risks,,” White Paper, Intel Corporation, Sept. 2007.
- [4] L. Montecchi, N. Nostro, A. Ceccarelli, G. Vella, A. Caruso, and A. Bondavalli, “Model-based evaluation of scalability and security tradeoffs: A case study on a multi-service platform,” Electronic
- [5] N. Mendes, A.A. Neto, J. Duraes, M. Vieira, and H. Madeira, “Assessing and Comparing Security of Web Servers,” Proc. IEEE Int’l Symp. Dependable Computing (PRDC), pp. 313-322, 2008.
- [6] L. Hong, A. Jain, and S. Pankanti, “Can Multibiometrics Improve Performance?” Proc. Workshop on Automatic Identification Advances Technologies (AutoID ’99) Summit, pp. 59-64, 1999.
- [7] K. Pousttchi and M. Schurig, "Assessment of today's mobile banking applications from the view of customer requirements," in System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on, 2004, p. 10 pp.
- [8] "Mobile Banking – The Future," Infogile Technologies 2007.
- [9] Mahmoud Elkhodr¹, Seyed Shahrestani¹ and Khaled Kourouche² , “A Proposal to Improve the Security of Mobile Banking Applications” 2012 Tenth International Conference on ICT and Knowledge Engineering.