

A Multiplayer TicTacToe Games in SOAP Platform

Submitted By

Rezaul Hasan

ID: 2012-2-60-010

Rakib Chowdhury

ID: 2012-1-60-021

Supervised by

Dr. Shamim Akhter

Assistant Professor

Department of Computer Science & Engineering

East West University

**A project Submitted in partial Fulfillment of the Requirements for the Degree of
Bachelors of Science in Computer Science and Engineering to the**



Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

Abstract

We present a new Simple Object Access Protocol (SOAP) system based games. In this project, we worked to develop a simple TicTacToe games using SOAP where clients and server are present. The clients are connected through the server. In this system, we do not use any socket programming. When one client tries to invoke a service it sends a request to server. In server side, there is a two dimensional array which keeps track of different move from different clients. When server sends the response to the other client it passes response according to saved data in server and the user interface changes according to saved data.

Declaration

We hereby declare that, this project was done under CSE497 and has not been submitted elsewhere for requirement of any degree or diploma or for any purpose except for publication.

Rezaul Hasan
ID: 2012-2-60-010
Department of Computer Science and Engineering
East West University

Rakib Chowdhury
ID: 2012-1-60-021
Department of Computer Science and Engineering
East West University

Letter of Acceptance

We hereby declare that this project is from the student's own work and the best effort of mine, and all other source of information used have been acknowledged. This project has been submitted with our approval.

Board of Examiners

Dr. Md. Shamim Akhter
Assistant Professor
Department of Computer Science and Engineering
East West University

Supervisor

Dr. Md. Mozammel Huq Azad Khan
Professor & Chairperson
Department of Computer Science and Engineering
East West University

Chairperson

Acknowledgement

First of all, I would like to thank almighty God for giving us the strength and proper knowledge to complete my thesis work.

I would like to express our deep sense of gratitude and sincere thanks to our honorable supervisor Dr. Md. Shamim Akhter, Assistant Professor, Department of Computer Science and Engineering, East West University, Aftabnagar, Dhaka, Bangladesh for his kind guidance, sharing knowledge and constant inspiration throughout this work.

Our sincere gratefulness to all the faculty members of Computer Science and Engineering for their friendly attitude and enthusiastic supports that has given me four years.

We are very grateful for the motivation and stimulation from our friends and seniors.

Finally, our most heartfelt gratitude goes to our beloved parents for their endless support, continuous inspiration, great contribution and perfect gratitude from the beginning to the end.

Table of Contents

Abstract	II
Declaration	III
Letter of Acceptance	IV
Acknowledgement	V
List of Figures	VIII-IX

Chapter 1:

Introduction

1.1 Overview	01
1.2 Scope and limitations	01
1.3 Objective	01
1.4 Contribution outlining	01
1.5 Organization of the project	01

Chapter 2:

Background Study

2.1 Overview	02
2.2 Distributed Systems	02
2.3 Middleware	03
2.3.1 Examples of middleware	03

Chapter 3:

Simple Object Access Protocol

3.1 What is SOAP?	04
3.2 What is web service?	04
3.3 How SOAP architecture works	05
3.4 SOAP message format	05
3.5 Security	05

Chapter 4:

SOAP Implementation

4.1 How to Install Compiler	06
4.2 Java SOAP: String Hello World (in 1 PC)	07
4.3 Java SOAP: Adding 2 numbers (in 1 PC)	17
4.4 Java SOAP: Subtracting 2 numbers (in 1 PC)	23
4.5 Java SOAP: Adding 2 numbers (Between 2 PCs)	28

Chapter 5:

Experimental Results

5.1 The steps and Outputs of our project	33
--	----

Chapter 6:

Conclusion and Future Work

6.1 Conclusion	36
6.2 Future Work	36

References	37
-------------------	-----------

List of Figures

Figure 2.2.1: Distributed System Architecture	02
Figure 2.3.1: Middleware	03
Figure 3.1: SOAP request response through http	04
Figure 3.2: The general purpose of engaging web service	04
Figure 3.4: SOAP message formats	05
Figure 4.1: Show the GlassFish server and download link	06
Figure 4.2.1: Show the project name and location of the project	07
Figure 4.2.2: Show the server name and Java version	08
Figure 4.2.3: Show the Web Service	10
Figure 4.2.4: Adding Operation to Web Service	10
Figure 4.2.5: Web Service design view after design operation is added.	11
Figure 4.2.6: Testing WelcomeSOAP welcome method.	12
Figure 4.2.7: TESTER URL	12
Figure 4.2.8: Change Local host in client pc by server pc IP	13
Figure 4.2.9: SOAP Request and Response	14
Figure 4.2.10: Show the WSDL URL	15
Figure 4.3.1: NetBeans Project show the Web Service in server side	17
Figure 4.3.2: Open a Java Client Application	19
Figure 4.3.3: Show the Web Service Client in client	19
Figure 4.3.4: Show the WSDL URL in client side	20
Figure 4.3.5: NetBeans Project tab after adding a web service reference to the project	20
Figure 4.3.6: The desired output from server to client	21
Figure 4.4.1: Testing SubSOAPServer method	25
Figure 4.4.2: The desired output from server to client	26
Figure 4.5.1: Show the server IP address in WSDL file	30
Figure 4.5.2: Show the desired output	31

Figure 5.1: Deploy the Web Server	33
Figure 5.2: Test Web Service	33
Figure 5.3: Show TicTacToe Web Service Tester	34
Figure 5.4: Show the WSDL file in local host	34
Figure 5.5: Show message	35
Figure 5.6: After press the first button instantly block this button	35
Figure 5.7: After second client click it graphically show and enable all buttons	35
Figure 5.8: Game win, lose and tie message will show both clients	36

A Multiplayer TicTacToe Games in SOAP Platform

Submitted By

Rezaul Hasan

ID: 2012-2-60-010

Rakib Chowdhury

ID: 2012-1-60-021

Supervised by

Dr. Shamim Akhter

Assistant Professor

Department of Computer Science & Engineering

East West University

**A project Submitted in partial Fulfillment of the Requirements for the Degree of
Bachelors of Science in Computer Science and Engineering to the**



Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

Abstract

We present a new Simple Object Access Protocol (SOAP) system based games. In this project, we worked to develop a simple TicTacToe games using SOAP where clients and server are present. The clients are connected through the server. In this system, we do not use any socket programming. When one client tries to invoke a service it sends a request to server. In server side, there is a two dimensional array which keeps track of different move from different clients. When server sends the response to the other client it passes response according to saved data in server and the user interface changes according to saved data.

Declaration

We hereby declare that, this project was done under CSE497 and has not been submitted elsewhere for requirement of any degree or diploma or for any purpose except for publication.

Rezaul Hasan
ID: 2012-2-60-010
Department of Computer Science and Engineering
East West University

Rakib Chowdhury
ID: 2012-1-60-021
Department of Computer Science and Engineering
East West University

Letter of Acceptance

We hereby declare that this project is from the student's own work and the best effort of mine, and all other source of information used have been acknowledged. This project has been submitted with our approval.

Board of Examiners

Dr. Md. Shamim Akhter
Assistant Professor
Department of Computer Science and Engineering
East West University

Supervisor

Dr. Md. Mozammel Huq Azad Khan
Professor & Chairperson
Department of Computer Science and Engineering
East West University

Chairperson

Acknowledgement

First of all, I would like to thank almighty God for giving us the strength and proper knowledge to complete my thesis work.

I would like to express our deep sense of gratitude and sincere thanks to our honorable supervisor Dr. Md. Shamim Akhter, Assistant Professor, Department of Computer Science and Engineering, East West University, Aftabnagar, Dhaka, Bangladesh for his kind guidance, sharing knowledge and constant inspiration throughout this work.

Our sincere gratefulness to all the faculty members of Computer Science and Engineering for their friendly attitude and enthusiastic supports that has given me four years.

We are very grateful for the motivation and stimulation from our friends and seniors.

Finally, our most heartfelt gratitude goes to our beloved parents for their endless support, continuous inspiration, great contribution and perfect gratitude from the beginning to the end.

Table of Contents

Abstract	II
Declaration	III
Letter of Acceptance	IV
Acknowledgement	V
List of Figures	VIII-IX
Chapter 1:	
Introduction	
1.1 Overview	01
1.2 Scope and limitations	01
1.3 Objective	01
1.4 Contribution outlining	01
1.5 Organization of the project	01
Chapter 2:	
Background Study	
2.1 Overview	02
2.2 Distributed Systems	02
2.3 Middleware	03
2.3.1 Examples of middleware	03
Chapter 3:	
Simple Object Access Protocol	
3.1 What is SOAP?	04
3.2 What is web service?	04

3.3 How SOAP architecture works	05
3.4 SOAP message format	05
3.5 Security	05

Chapter 4:

SOAP Implementation

4.1 How to Install Compiler	06
4.2 Java SOAP: String Hello World (in 1 PC)	07
4.3 Java SOAP: Adding 2 numbers (in 1 PC)	17
4.4 Java SOAP: Subtracting 2 numbers (in 1 PC)	23
4.5 Java SOAP: Adding 2 numbers (Between 2 PCs)	28

Chapter 5:

Experimental Results

5.1 The steps and Outputs of our project	33
--	----

Chapter 6:

Conclusion and Future Work

6.1 Conclusion	36
6.2 Future Work	36

References	37
-------------------	-----------

List of Figures

Figure 2.2.1: Distributed System Architecture	02
Figure 2.3.1: Middleware	03
Figure 3.1: SOAP request response through http	04
Figure 3.2: The general purpose of engaging web service	04
Figure 3.4: SOAP message formats	05
Figure 4.1: Show the GlassFish server and download link	06
Figure 4.2.1: Show the project name and location of the project	07
Figure 4.2.2: Show the server name and Java version	08
Figure 4.2.3: Show the Web Service	10
Figure 4.2.4: Adding Operation to Web Service	10
Figure 4.2.5: Web Service design view after design operation is added.	11
Figure 4.2.6: Testing WelcomeSOAP welcome method.	12
Figure 4.2.7: TESTER URL	12
Figure 4.2.8: Change Local host in client pc by server pc IP	13
Figure 4.2.9: SOAP Request and Response	14
Figure 4.2.10: Show the WSDL URL	15
Figure 4.3.1: NetBeans Project show the Web Service in server side	17
Figure 4.3.2: Open a Java Client Application	19
Figure 4.3.3: Show the Web Service Client in client	19
Figure 4.3.4: Show the WSDL URL in client side	20
Figure 4.3.5: NetBeans Project tab after adding a web service reference to the project	20
Figure 4.3.6: The desired output from server to client	21

Figure 4.4.1: Testing SubSOAPServer method	25
Figure 4.4.2: The desired output from server to client	26
Figure 4.5.1: Show the server IP address in WSDL file	30
Figure 4.5.2: Show the desired output	31
Figure 5.1: Deploy the Web Server	33
Figure 5.2: Test Web Service	33
Figure 5.3: Show TicTacToe Web Service Tester	34
Figure 5.4: Show the WSDL file in local host	34
Figure 5.5: Show message	35
Figure 5.6: After press the first button instantly block this button	35
Figure 5.7: After second client click it graphically show and enable all buttons	35
Figure 5.8: Game win, lose and tie message will show both clients	36

Chapter 1

Introduction

1.1: Overview

In this project we were asked to work with SOAP (Simple Object Access Protocol) messaging protocol to make connection between client and server computer. The main reason to work with this messaging protocol was that it allows program to access object on different operating system (Windows and Linux) on a network. It is also reliable protocol to send message on remote distributed computer. SOAP specifies exactly how to encode an HTTP header and an XML file so that a program in one computer can call a program in another computer and pass along information.

We used an IP address for making connection between server and client computer. To get a service from server client should be connected with server then it will be capable of to request a method to execute.

The client sends parameters to server and after compiling it sends response to client. In this project we developed a game without database.

1.2: Scope and limitation

- SOAP messages are simple XML messages to understand.
- Can be built with any language and can be used in any platform.
- It is slower than CORBA, RMI.
- Dependency on WSDL file

1.3: Objective

Objectives of this project are:

- Analyzing and studying Simple Object Access Protocol and its most features.
- Implement a simple game and play between multiple clients using common server to understand SOAP protocol.
- Platform Independent Implementation learning in LAN network.

1.4: Contribution Outlining

In next chapter we simply described the concept of Distributed system and Middleware. Chapter 3 contains all about SOAP architecture.(Here something will be added).Chapter 4 contains simple SOAP example between

server and client and 5 contains conclusion and future plan and chapter 6 contains implementations of our project.

1.5: Organization of the project

As we tried to develop a SOAP architecture based game first of all we developed a single simple “Hello World” program that receives parameter from client and returns a message to client from server where client and server exist in same computer. Then we made access that web method server from another computer. We discuss later how those connections were made and according to this simple program how we made our project.

Chapter 2

Background Study

2.1: Overview

In this era distributed system is everywhere. The biggest example of distributed system is internet. Many organizations want to create their own network in their organization. Mobile phone network, factory network, home network, campus network, corporate network etc are some of the examples of distributed systems. Distributed network not only exist only in single organization. Two computer connected in different countries can also be said distributed network.

2.2 Distributed systems

A distributed system consists of a collection of autonomous computers, connected through a network and distribution middleware, which enables computers to coordinate their activities and to share the resources of the system, so that users perceive the system as a single, integrated computing facility [1].In distributed systems user and application can communicate with each other consistently.

Resource sharing, openness, concurrency, scalability, transparency, fault tolerance are some of the major characteristics of distributed system.

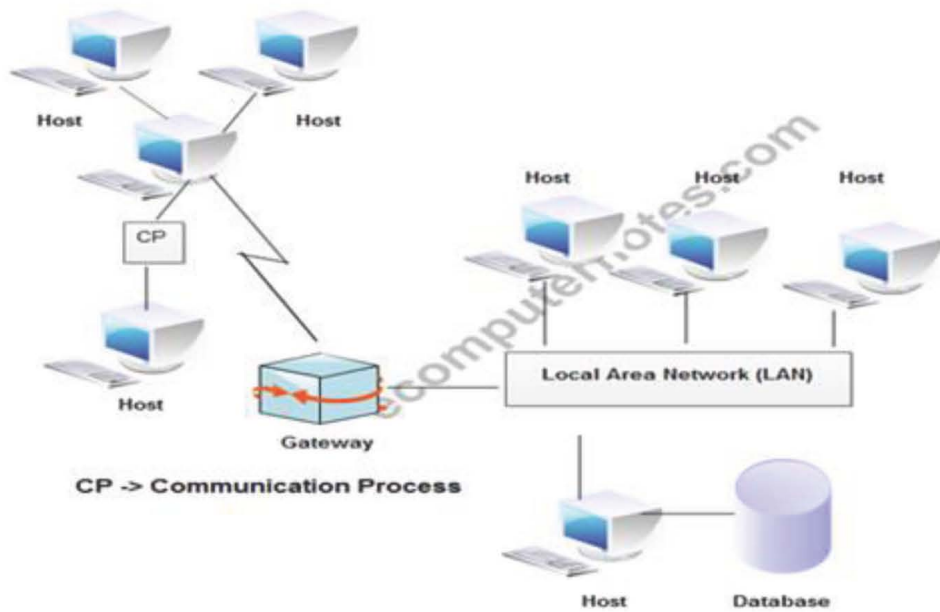


Figure 2.2.1: Distributed System Architecture [2]

2.3 Middleware

Middleware is software that makes connection between operating system and software applications in a network. Sometimes it becomes necessary to work with different application existed on different environment for required result. In those situations middleware is needed. It is also called as “Software Glue”. Typically middleware programs provide messaging services so that different applications can communicate with each other using messaging framework like Simple Object Access Protocol (SOAP), Representational State Transfer (REST) .This picture will help to understand the task of middleware where middleware is broadcasting request to many servers.

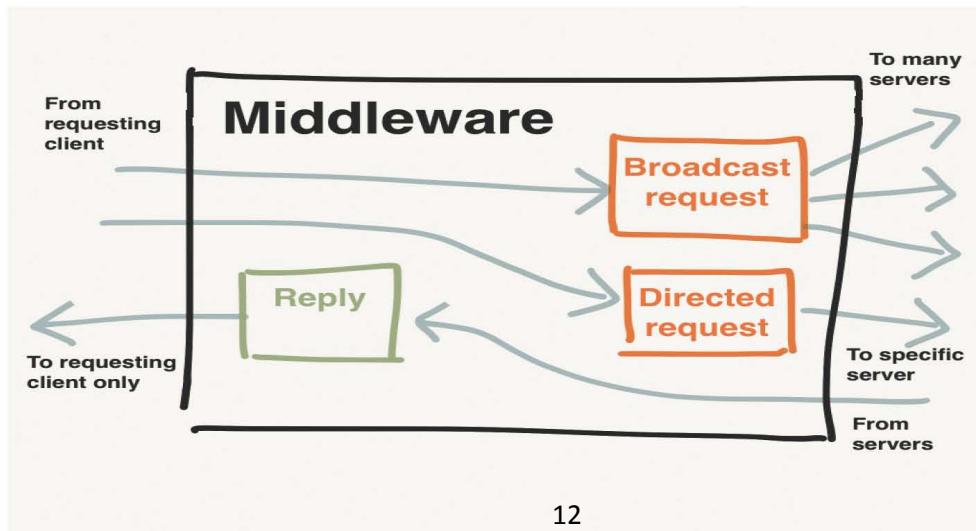


Figure 2.3.1 Middleware

2.3.1 Examples of middleware

Different types of Middleware examples are as follows:

- RPC(Request Procedure Call)
- Java's Database connectivity API :JDBC
- Distributed Component Object Model(DCOM)

Chapter 3

Simple Object Access Protocol

3.1 What is SOAP?

Abbreviation of SOAP is Simple Object Access Protocol. It was invented by Microsoft. This is a stateless, platform independent, XML based protocol that uses HTTP as its transport medium and a protocol for sending and receiving message and can be used for developing distributed complex computing environments. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses.[4]

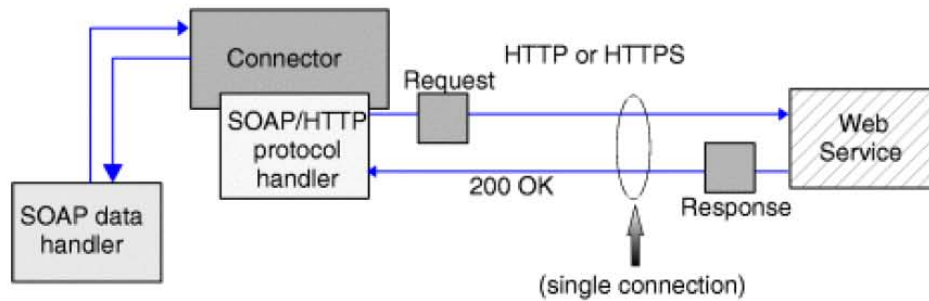


Figure 3.1: SOAP request response through http [5]

3.2 What is Web service?

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. WSDL file is a machine processable format for describing interface used in web service.

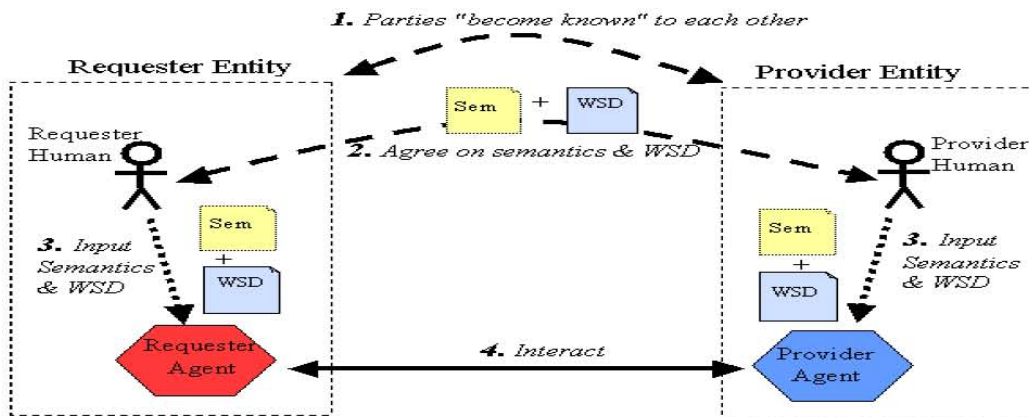


Figure 3.2: The general purpose of engaging web service

3.3 How SOAP architecture works?

Web Service is a software module which is contained by a service provider. An XML-based descriptor about the Web service is published to a service registry. Information about the Web service and how to run it is found in the descriptor. It contacts the service registry when a service requestor wants to run a web service. The requestor binds to the service provider and runs the web service.

All this take place by SOAP. It is used to publish the descriptor into a service registry to send a service request from a requestor to the registry to send information from the registry to the requestor and then to allow the requestor to bind to the service provider and run the web service. There could be no web services without SOAP.

Web service works with request response procedure. There will be an entity which will request for services at its specific service provider entity. According to request server will response with a response message. So there are two XML based message request and response which are basically xml based.

3.4 SOAP message format

The framework consists of the following XML elements: Envelope, Header, Body and Fault all of which are from <http://schemas.xmlsoap.org/soap/envelop>. Here are the details,

- Envelope- Envelope element is the core or root element of a SOAP message. This element defines the XML document as a SOAP message.
- Header- SOAP Header element contains application-specific information (like authentication, payment, etc)
- Body-Contains the actual SOAP message

```
<?xml version="1.0"?>

<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">

  <soap:Header>
  ...
  </soap:Header>

  <soap:Body>
  ...
    <soap:Fault>
    ...
    </soap:Fault>
  </soap:Body>

</soap:Envelope>
```

Figure 3.4: SOAP message formats [6]

3.5 Security

- SOAP is more secure than RESTful and JSON web service.
- Existence of various standards that relate to security.

Chapter 4

4.1 How to Install NetBeans for Java Programming

Firstly, we need to download jdk 1.8.0_71. We can download from this following link:

<http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>

when our download will be finish, we will install it in our computer.

After that, we need to download NetBeans IDE. NetBeans is an open-source Integrated Development Environment. Of course GlassFish Server Open Source Edition 4.1.1 will be internally adds with NetBeans.

Here the following download link:

<https://netbeans.org/downloads/>

our desired download page will be like this:

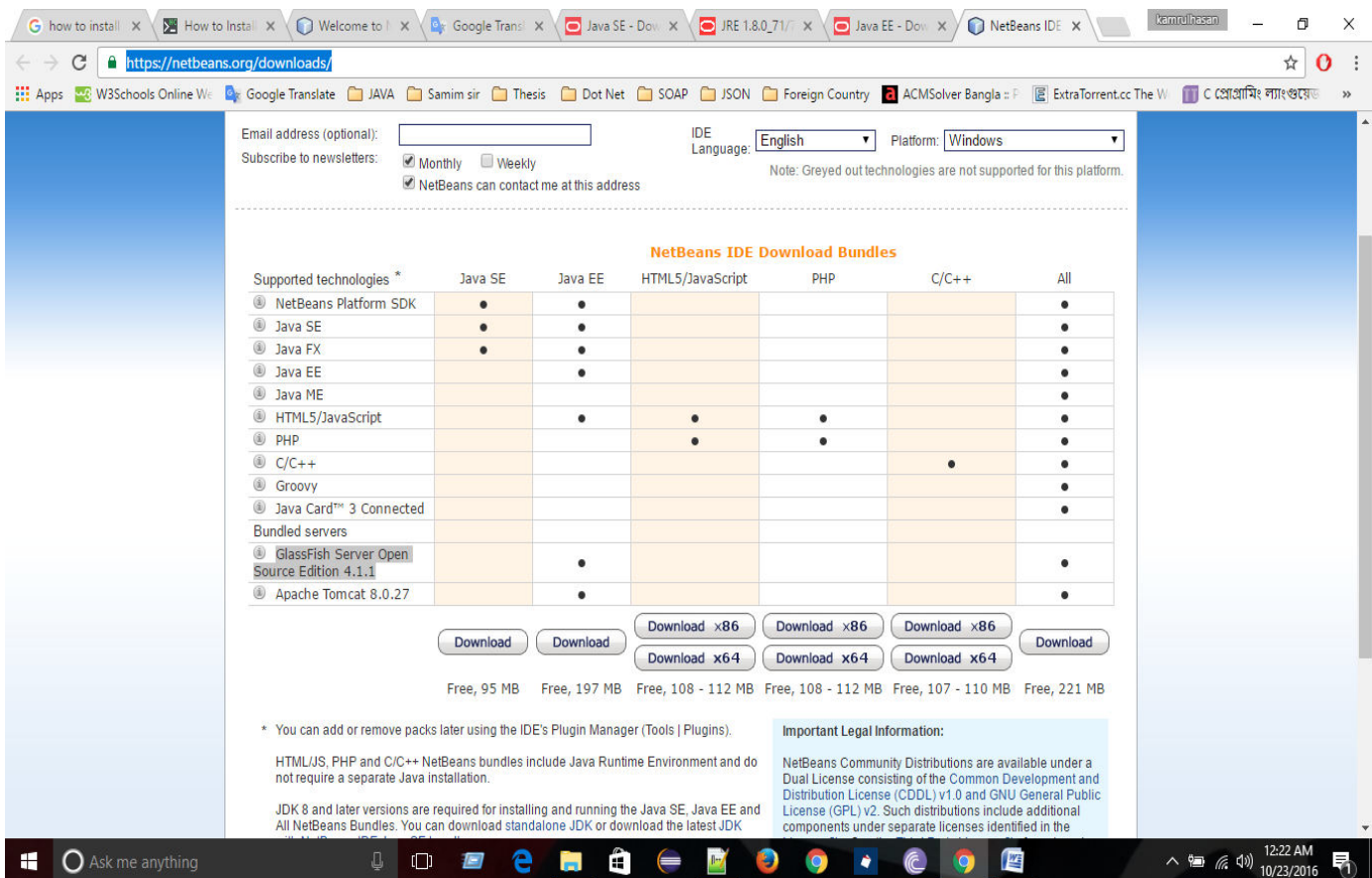


Figure 4.1: Show the GlassFish server and download link

4.2 String Hello World

Publishing and Consuming SOAP-Based web services

When you create a web service mainly focus on its logic, make sure that there is IDE and server handle its infrastructure. Firstly you create a Web application project after that you need to create a Client application project.

There are some steps for Creating web application project and Adding web service class in NetBeans. We begin with a SOAP-based web service.

@@:: Creating a Web Application Project in NetBeans and the following steps are given below:-

1. Select **File -> New Project** to open the new project dialog.
2. Select **Java Web** from the dialog's **Categories** list then select **Web Application** from the **Project** list. Click **Next >**.

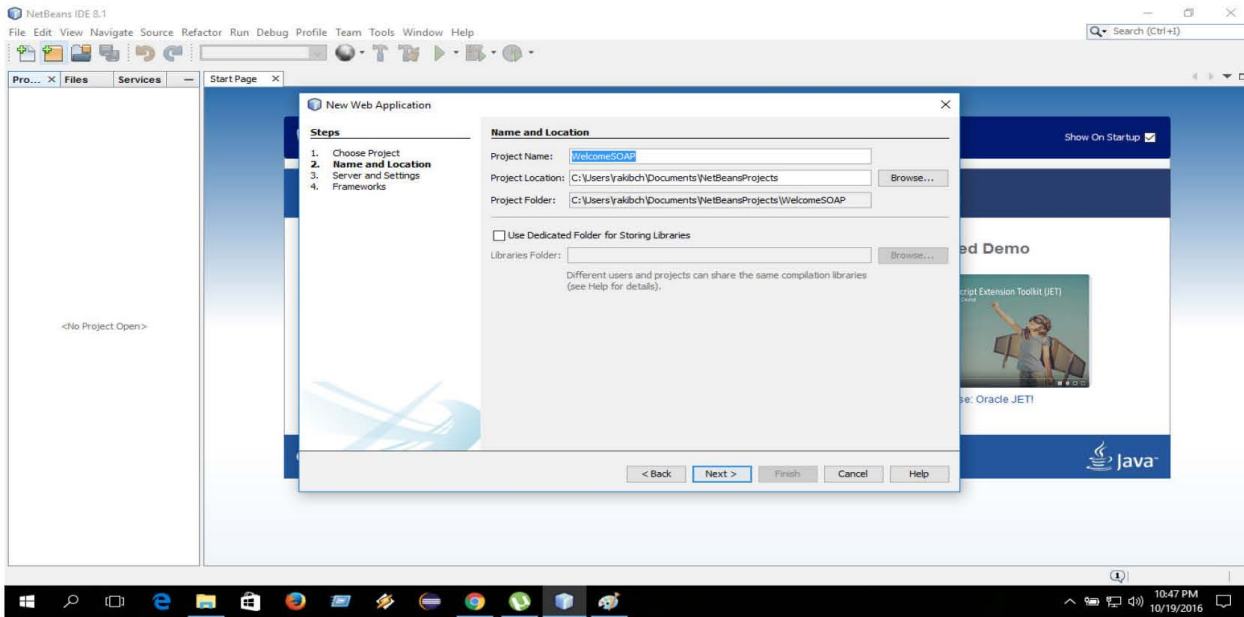


Figure 4.2.1: Show the project name and location of the project

3. Specify the name of your project (**WelcomeSOAP**) in the **Project Name** field and specify where you want to store the project in the **Project Location** field. You can click the **Browse** button to select the location. Click **Next >**.

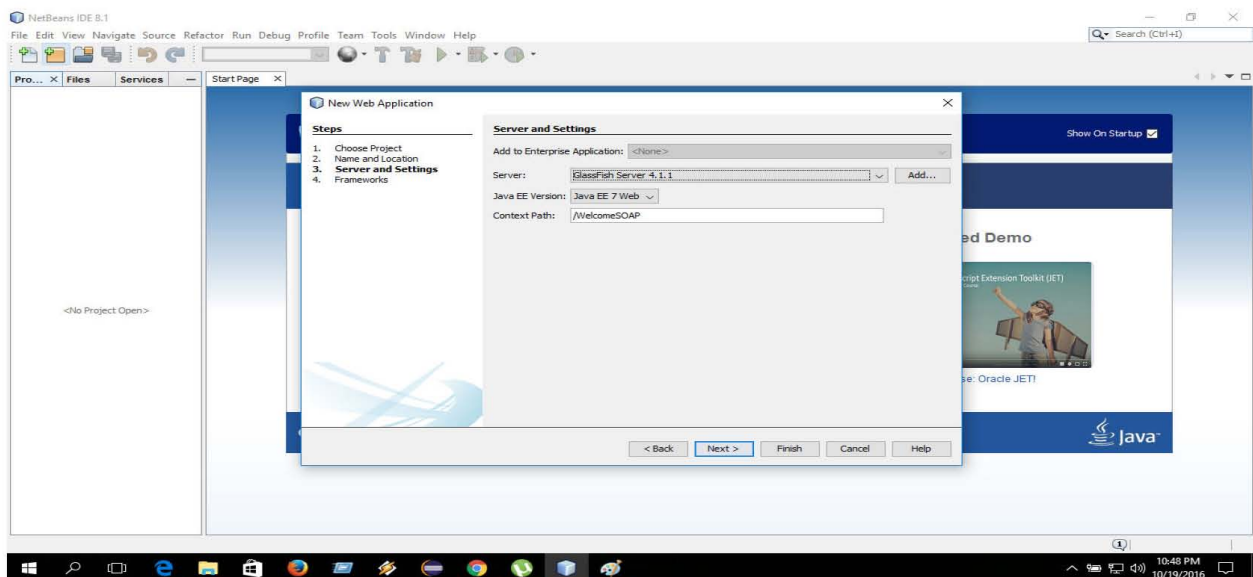


Figure 4.2.2: Show the server name and Java version

4. Select **GlassFish Server3** from the **Server** drop-down list and **Java EE 6 Web** from the **Java EE Version** drop-down list.
5. Click **Finish** to create the web service class.

@@:: Adding a Web Service Class to a Web Application Project in NetBeans and the following steps are given below:-

1. In the Projects tab in NetBeans right click the **WelcomeSOAP** project's node and select **New -> Web Service** to open the New Web Service dialog.

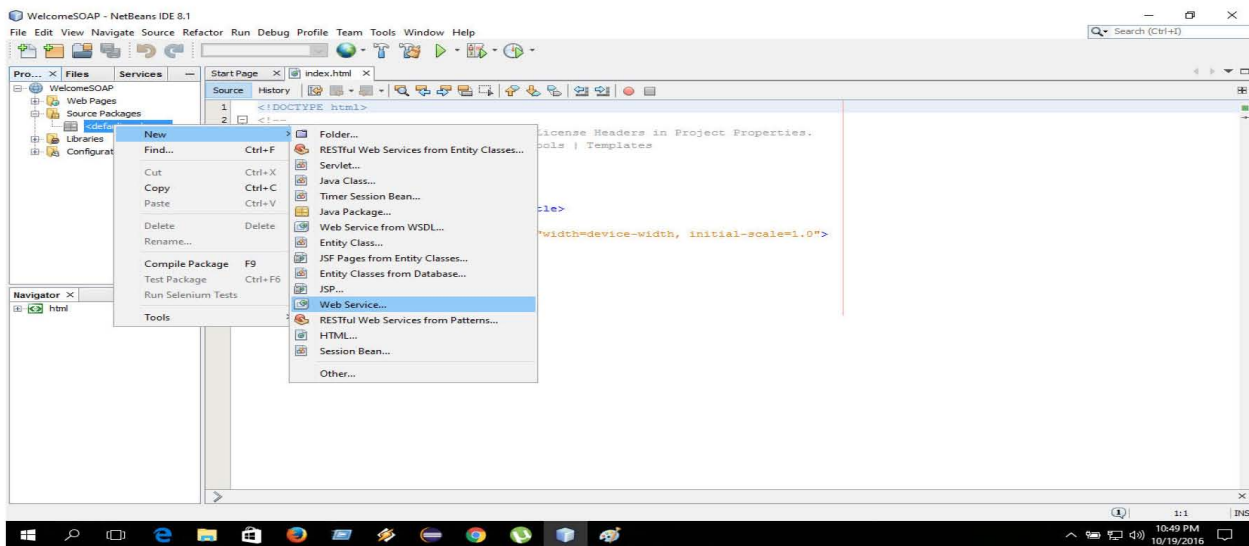


Figure 9: Show the Web Service

2. Specify **WelcomeSOAP** in the **Web Service Name** field.
3. Specify **com.deitel.welcomesoap** in the **Package** field.
4. Click **Finish** to create the web service class.

Now your Web Service is available to client applications. When you build your application the IDE will generate other supporting files for your web services.

@@:: Defining the WelcomeSOAP Web Service in NetBeans

The completed WelcomeSOAP Service code is here:

```
package com.prgguru.example;

import javax.jws.WebService;

import javax.jws.WebMethod;

import javax.jws.WebParam;

@WebService(serviceName = "sayHelloService")

public class sayHelloService {
```



```

@WebMethod(operationName = "hello")

public String hello(@WebParam(name = "name") String name) {

    return "Welcome to JAX-WS web services with SOAP " + name + " !";

}

@WebMethod(operationName = "welcome")

public String welcome(@WebParam(name = "name") String name) {

    return "Your name is " + name + " !";

}

}

```

@@:: Completing the Web Service’s Code

“NetBeans provides a web service design view so you can define the methods and parameters for your web services. To define the WelcomeSOAP class’s welcome method, perform the following steps:”

1. In the project’s **Web Services** node, double click **WelcomeSOAP** to open the file **WelcomeSOAPService.java** in the code editor.
2. Click the **Design** button at the top of the code editor to show the web service design view
3. Click the **Add Operation** button to display the **Add Operation** dialog
4. Specify the method name **welcome** in the **Name** field then the default **Return Type (String)** is correct for this example.
5. Add the method’s **name** parameter by clicking the **Add** button to the right of the **Parameters** tab then entering **name** in the **Name** field. The parameter’s default **Type (String)** is correct for this example.

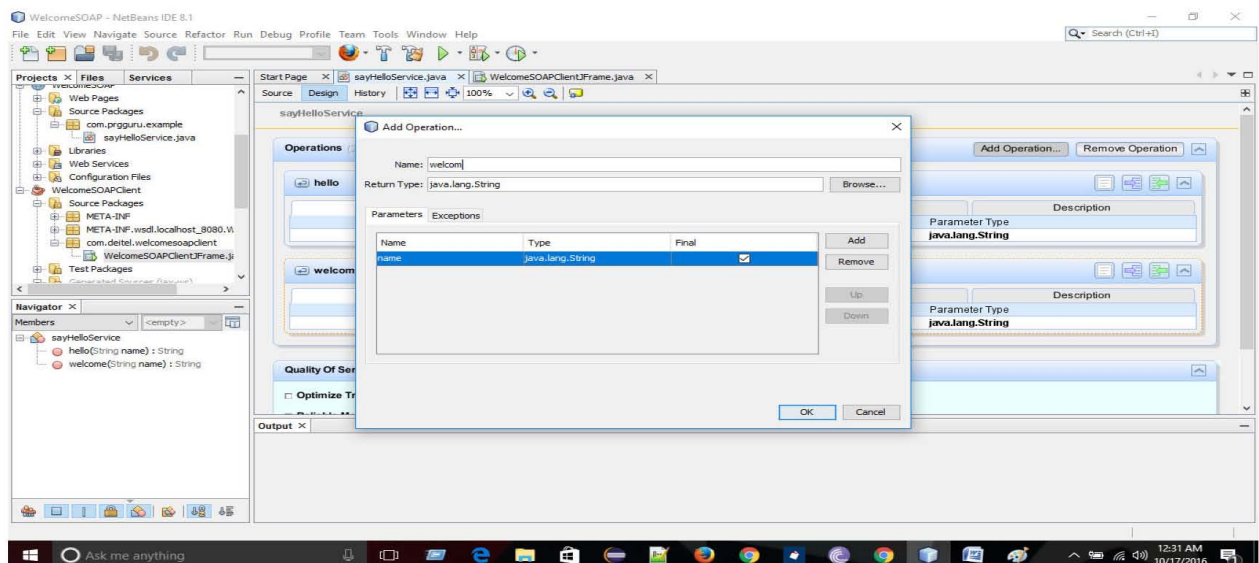


Figure 4.2.3: Adding Operation to Web Service

6. Click OK to create the welcome method. The design view should now appear in the Figure.

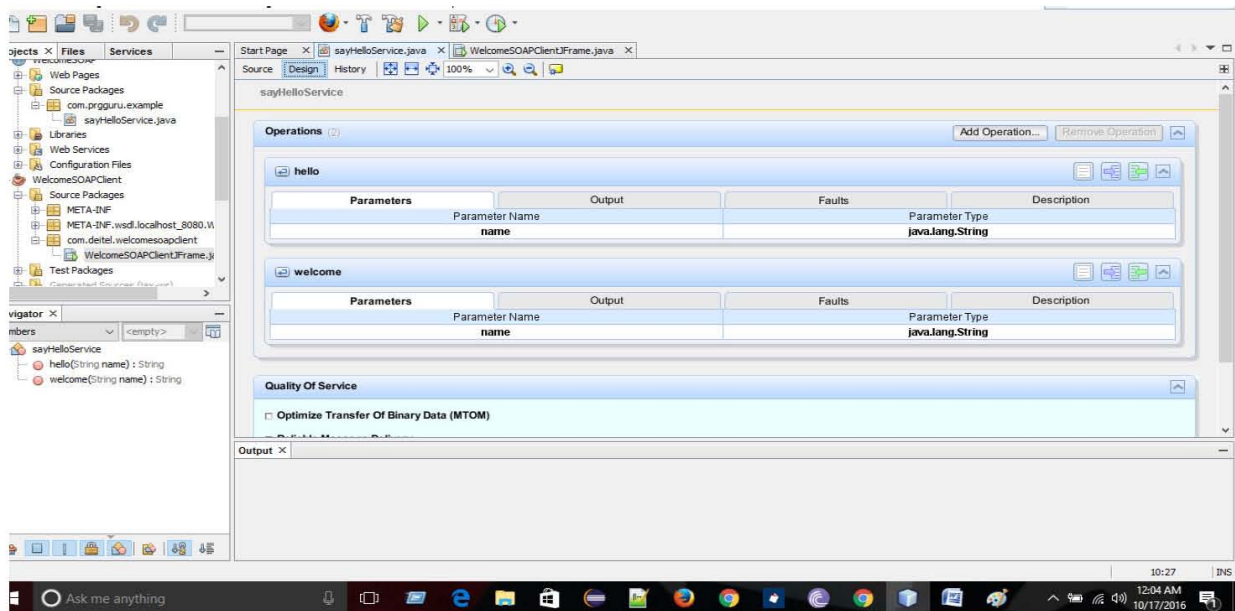


Figure 4.2.4: Web Service design view after design operation is added.

7. At the top of the design view, click the Source button to display the class's source code.

@@:: Publishing the WelcomeSOAP Web Service from NetBeans

NetBeans handles all the details of building and deploying a web service. Right click the project name **WelcomeSOAP** in the Projects tab and select **Deploy** to build and deploy the web application to the GlassFish server.

@@:: Testing the WelcomeSOAP Web Service with GlassFish Application Server's Tester Web Page

You need to test the WelcomeSOAP web service. The GlassFish server is dynamically create a web page for you to test a web service method from a web browser. We need to use like this way:

1. Expand the project's **Web Service** in the NetBeans project tab.
2. Right click the web service class **name (WelcomeSOAP)** and select Test Web Service.

Figure: Tester web page created by Glassfish for the WelcomeSOAP web service.

Once you have deployed the web service, you can also see the URL

<http://localhost:8080/WelcomeSOAP/sayHelloService?Tester>

To test WelcomeSOAP welcome web method, type your name in the text field to the right of the **welcome** button and click the welcome button that will invoke the method.

You can access the server when server is on running mode. If NetBeans launches GlassFish for you, it will automatically shut it down when you close NetBeans. The GlassFish server shows how manually server start and stop, search this link

glassfish.java.net/downloads/quickstart/index.html

The web page will be look like:



Figure 4.2.5: Testing WelcomeSOAP welcome method.

Testing the WelcomeSOAP Web Service from Another Computer:-

If your computer is connected to a network and allows HTTP requests then you can test the web service from another computer on the network by typing the following URL

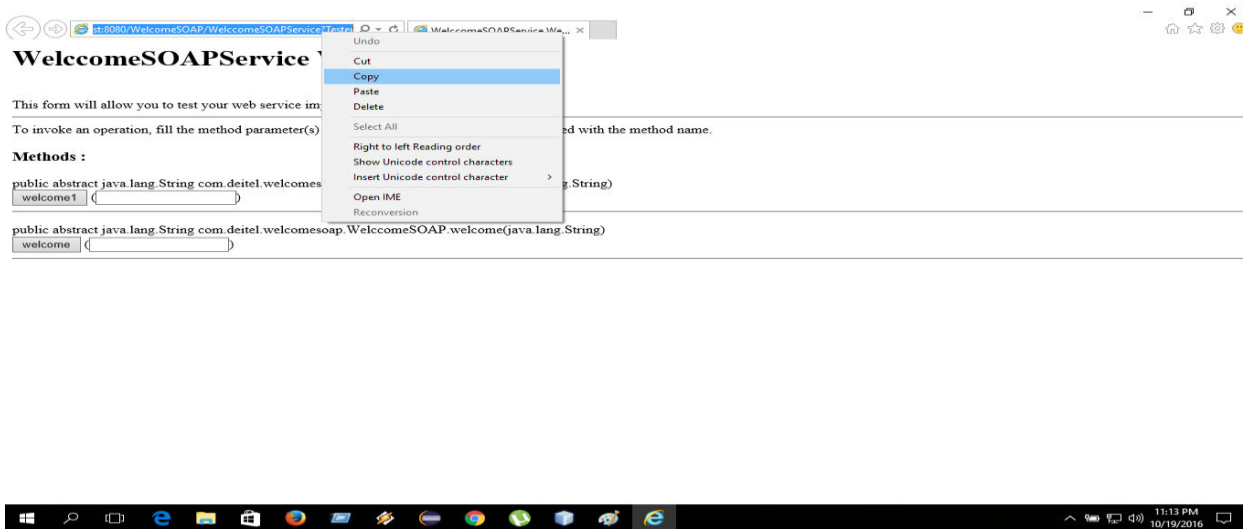


Figure 4.2.6: TESTER URL

You can also copy the URL:

Paste the URL in another computer browser and change the local host by your IP. When you press enters it will look like this:

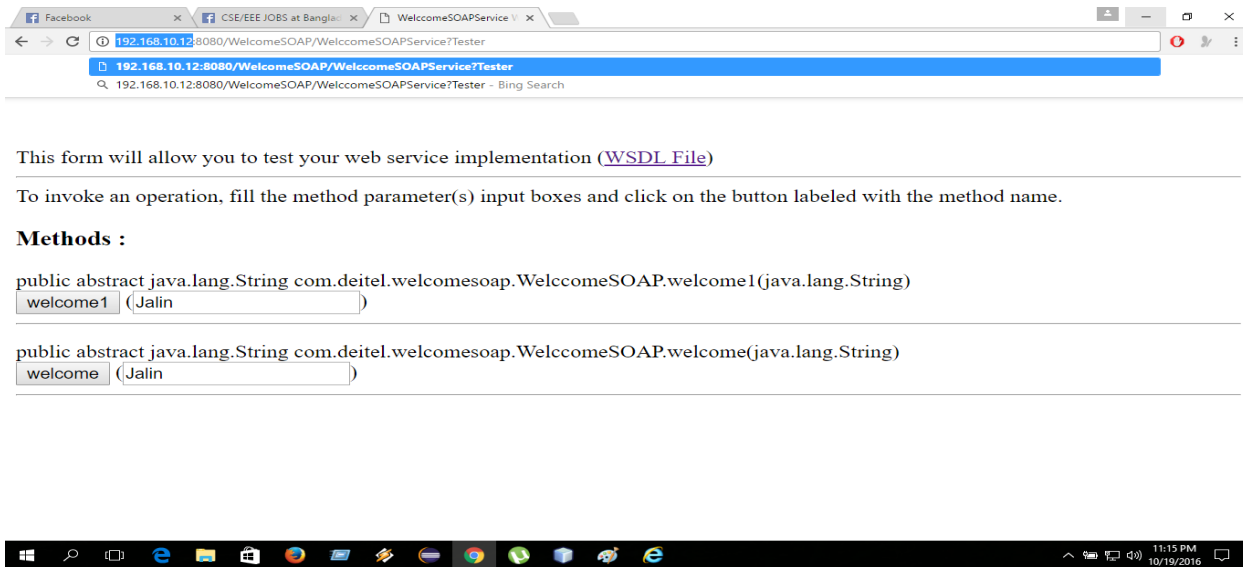


Figure 4.2.7: Change Local host in client pc by server pc IP

Type your name in the text field to the right of the welcome button; click the welcome button that will invoke the method.



Figure 4.2.8: SOAP Request and Response

Creating a Client to Consume the WelcomeSOAP Web Service

Creating a Desktop Application Project in NetBeans:-

Before performing those steps you need to ensure that the WelcomeSOAP web service has been deployed and Glassfish server application is running. Perform the following steps to create a client Java desktop application in NetBeans:

1. Select **File -> New project** to open the New Project dialog.
2. Select **java** from the **Catalogs** list and **Java Application** from the list then click **Next ->**.
3. Specify name **WelcomeSOAPClient** in the **Project Name** field and uncheck the **Create Main Class** checkbox. Later you will add a subclass of JFrame that contains a main method.
4. Click **Finish** to create the Project.

Step 2: Adding a web Service Reference to an Application

You will add a web service reference to your applicant that can interact with the WelcomeSOAP web service.

1. Right click the **project name (WelcomeSOAPClient)** in the NetBeans Projects tab and select **New -> Web Service Client** from the pop-up menu to display the **New Web Service Client** dialog.
2. In the **WSDL URL** field use the URL <http://localhost:8080/WelcomeSOAP/sayHelloService?WSDL> . This URL tells the IDE where to find the web service's WSDL description. The IDE uses this WSDL description to generate the client-side artifacts that compose and support the service endpoint interface.

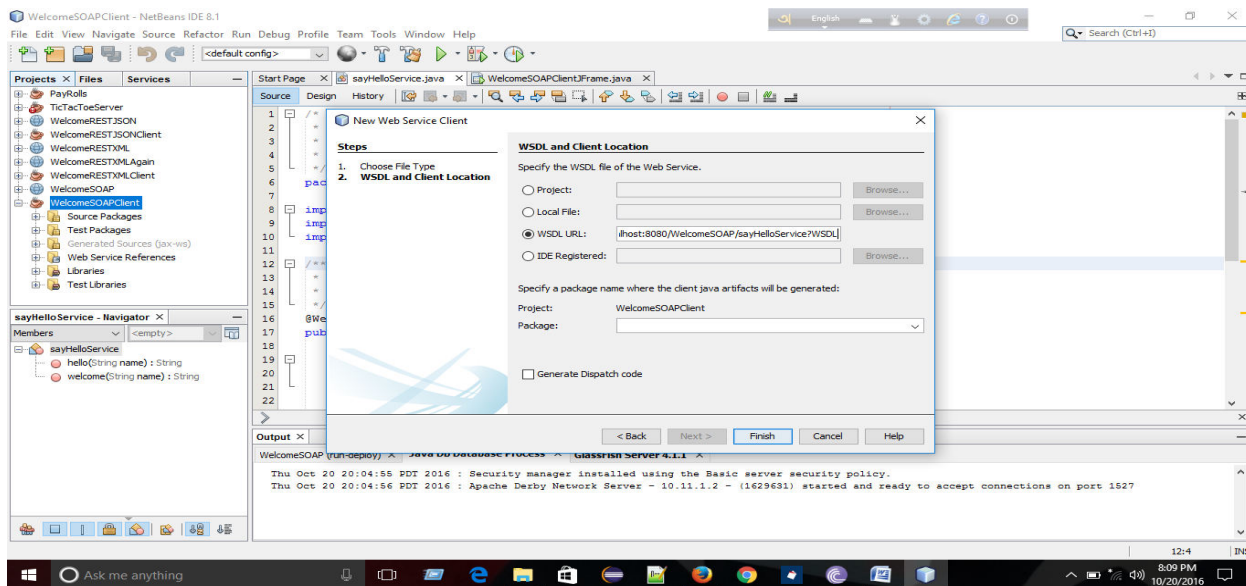


Figure 4.2.9: Show the WSDL URL

3. Click **Finish** to create the web service reference and dismiss the **New Web Service Client** dialog.

Consuming the WelcomeSOAP Web Service:

We will use a GUI application to interact with the WelcomeSOAP web service. To build the client application's GUI add a subclass of JFrame to the project by performing the following steps:

1. Right click the project name (**WelcomeSOAPClient**) in the NetBeans Project tab and select **New -> JFrame Form** to display the **New JFrame Form** dialog.

2. Specify WelcomeSOAPClient JFrame in the Class Name Field.
3. Specify com.deitel.welcomesoapclient in the package field.
4. Click **Finish** to close the New JFrame From dialog.

Next use the NetBeans GUI design tools to build the GUI shown in the sample screen captures at the end. We use WelcomeSOAP web service to display a welcome message to the user.

Firstly, we will create a JFrame and use JLabel, JTextField also JSubmitButton.

In the text field you need to give a name and click the submit button. You will see your desired output.

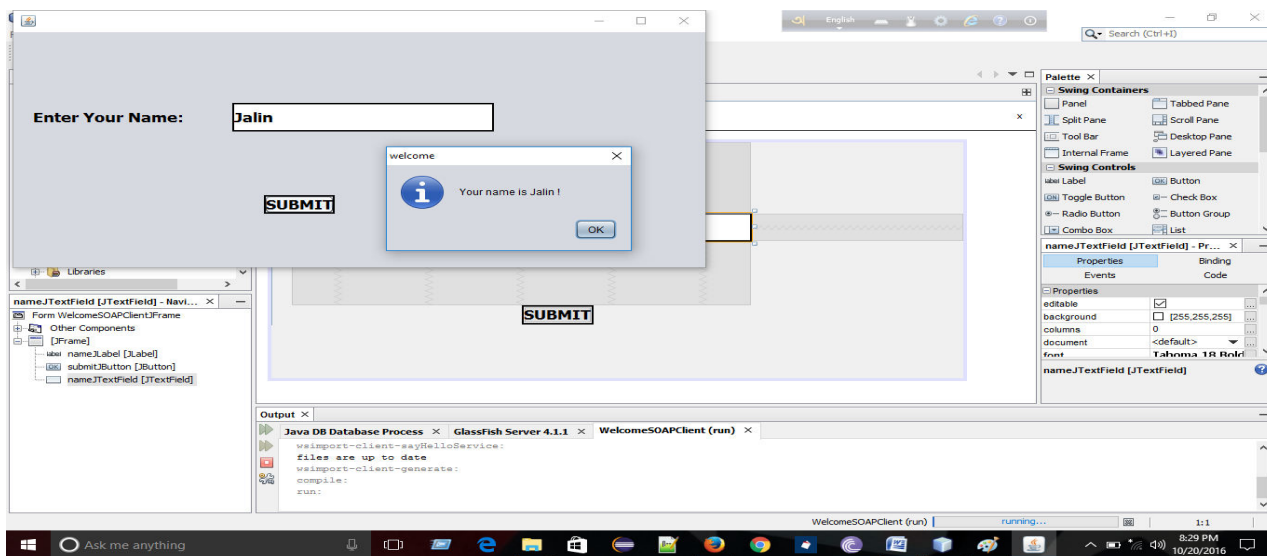


Figure 4.2.10: Final result client through server

Here the Client Code:-

```
package com.deitel.welcomesoapclient;

import com.prgguru.example.SayHelloService_Service;
import com.prgguru.example.SayHelloService;
import javax.swing.JOptionPane;

public class WelcomeSOAPClientJFrame extends javax.swing.JFrame {

    public WelcomeSOAPClientJFrame() {

        initComponents();

        try

        {

            SayHelloService_Service service = new SayHelloService_Service();
```

```

        welcomeSOAPPProxy = service.getSayHelloServicePort();
    }
    catch(Exception exception)
    {
        exception.printStackTrace();

        System.exit( 1 );
    }
}

private void submitJButtonActionPerformed(java.awt.event.ActionEvent evt) {
    String name = nameJTextField.getText();

    String message = welcomeSOAPPProxy.welcome(name);

    JOptionPane.showMessageDialog(this, message, "welcome",
JOptionPane.INFORMATION_MESSAGE);
}

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new WelcomeSOAPClientJFrame().setVisible(true);
        }
    });
}

private javax.swing.JLabel nameJLabel;
private javax.swing.JTextField nameJTextField;
private javax.swing.JButton submitJButton;
}

```

4.3 Publishing and Consuming SOAP-Based Add web services

When you create a web service mainly focus on its logic, make sure that there is IDE and server handle its infrastructure. Firstly you create a Web application project after that you need to create a Client application project.

There are some steps for Creating web application project and Adding web service class in NetBeans. We begin with a SOAP-based web service.

@@:: Creating a Web Application Project in NetBeans and the following steps are given below:-

6. Select **File -> New Project** to open the new project dialog.
7. Select **Java Web** from the dialog's **Categories** list then select **Web Application** from the **Project** list. Click **Next ->**.
8. Specify the name of your project (**AddSOAPServer**) in the **Project Name** field and specify where you want to store the project in the **Project Location** field. You can click the **Browse** button to select the location. Click **Next >**.
9. Select **GlassFish Server3** from the **Server** drop-down list and **Java EE 6 Web** from the **Java EE Version** drop-down list.
10. Click **Finish** to create the web service class.

@@:: Adding a Web Service Class to a Web Application Project in NetBeans and the following steps are given below:-

5. In the Projects tab in NetBeans right click the **AddSOAPServer** project's node and select **New -> Web Service** to open the New Web Service dialog.

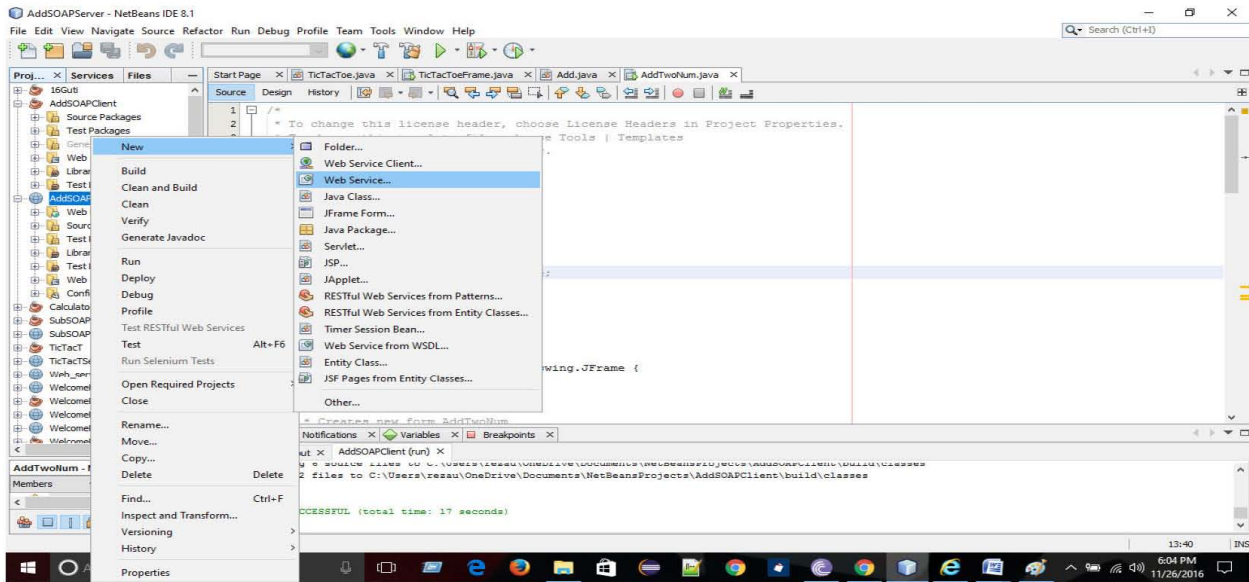


Figure 4.3.1: NetBeans Project show the Web Service in server side

6. Specify **WelcomeSOAP** in the **Web Service Name** field.
7. Specify **com.deitel.welcomesoap** in the **Package** field.
8. Click **Finish** to create the web service class.

Now your Web Service is available to client applications. When you build your application the IDE will generate other supporting files for your web services.

@@:: Defining the AddSOAPServer Web Service in NetBeans

The completed AddSOAPServer Service code is here:

```
package com.addsoapservice;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

@WebService(serviceName = "Add")

public class Add {
    @WebMethod(operationName = "add")

    public int add(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {

        int k = a + b;

        return k;

    }
}
```


}

Creating a Client to Consume the AddSOAPClient Web Service

Creating a Desktop Application Project in NetBeans:-

Before performing those steps you need to ensure that the AddSOAPClient web service has been deployed and Glassfish server application is running. Perform the following steps to create a client Java desktop application in NetBeans:

5. Select **File -> New project** to open the New Project dialog.
6. Select **java** from the **Catalogs** list and **Java Application** from the list then click **Next ->**.

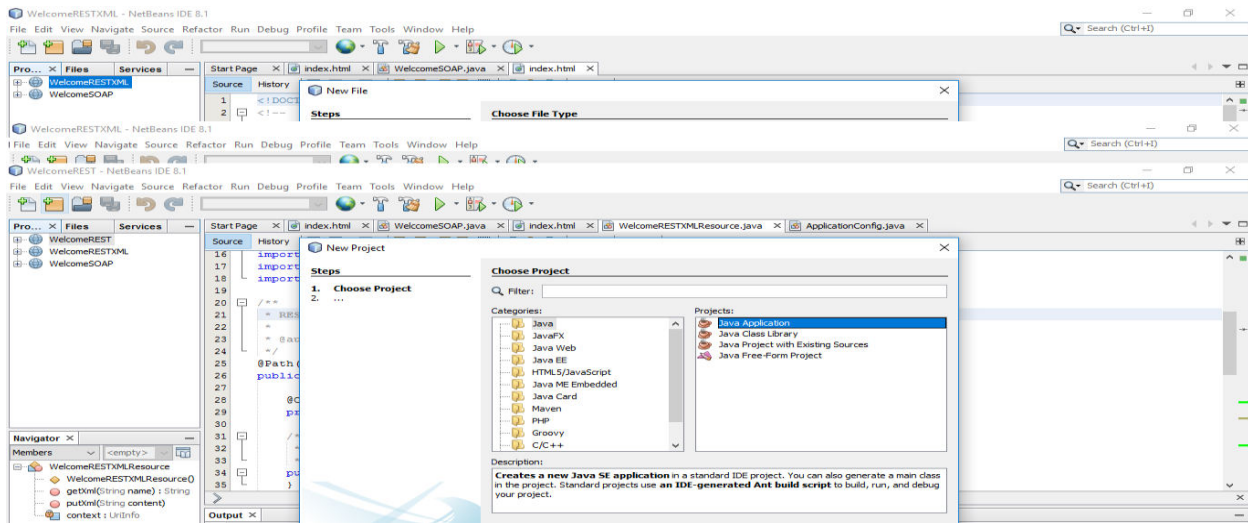


Figure 4.3.2: Open a Java Application

7. Specify name **AddSOAPClient** in the **Project Name** field and uncheck the **Create Main Class** checkbox. Later you will add a subclass of JFrame that contains a main method.
8. Click **Finish** to create the Project.

Step 2: Adding a web Service Reference to an Application

You will add a web service reference to your applicant that can interact with the WelcomeSOAP web service.

4. Right click the **project name (AddSOAPClient)** in the NetBeans Projects tab and select **New -> Web Service Client** from the pop-up menu to display the **New Web Service Client** dialog.

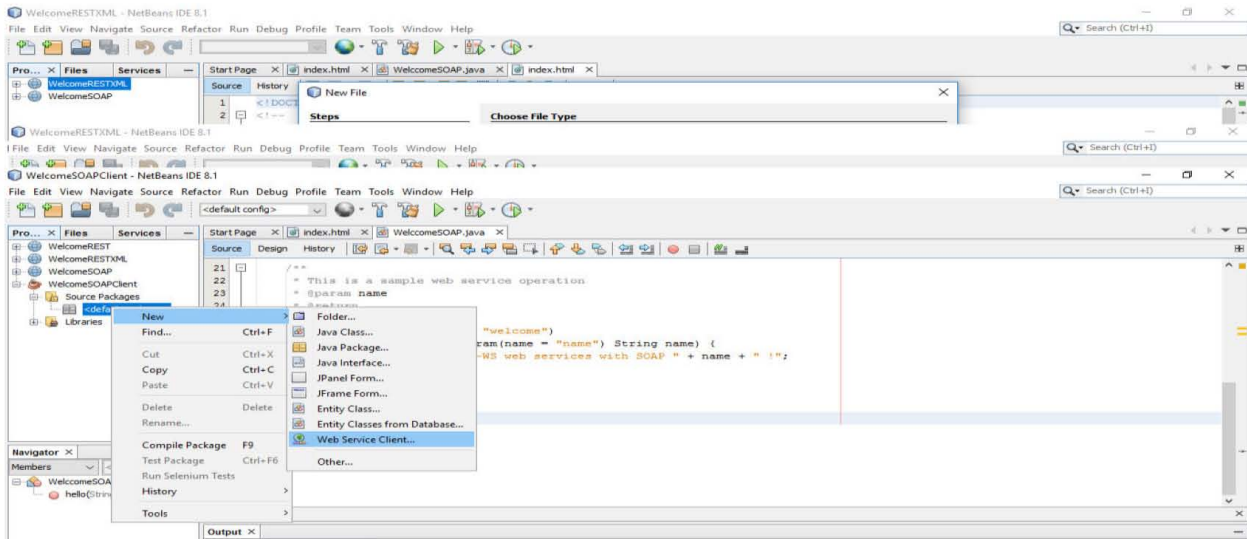


Figure 4.3.3: Show the Web Service Client in client side

- In the WSDL URL field use the URL <http://localhost:8080/AddSOAPServer/Add?WSDL>. This URL tells the IDE where to find the web service's WSDL description. The IDE uses this WSDL description to generate the client-side artifacts that compose and support the service endpoint interface.

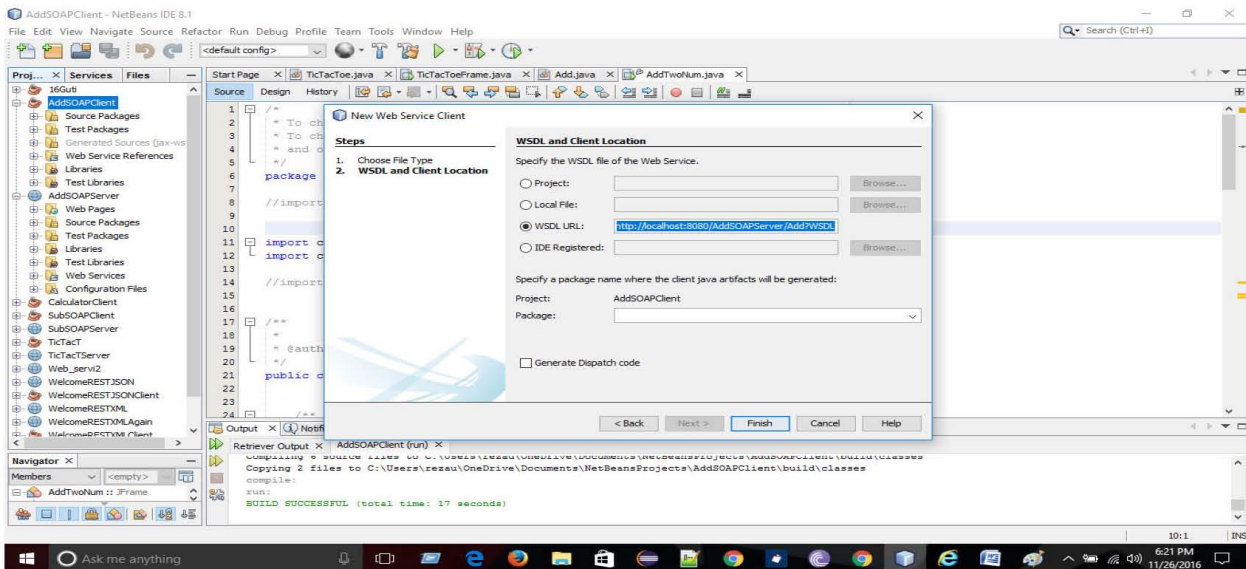


Figure 4.3.4: Show the WSDL URL in client side

- Click Finish to create the web service reference and dismiss the New Web Service Client dialog.

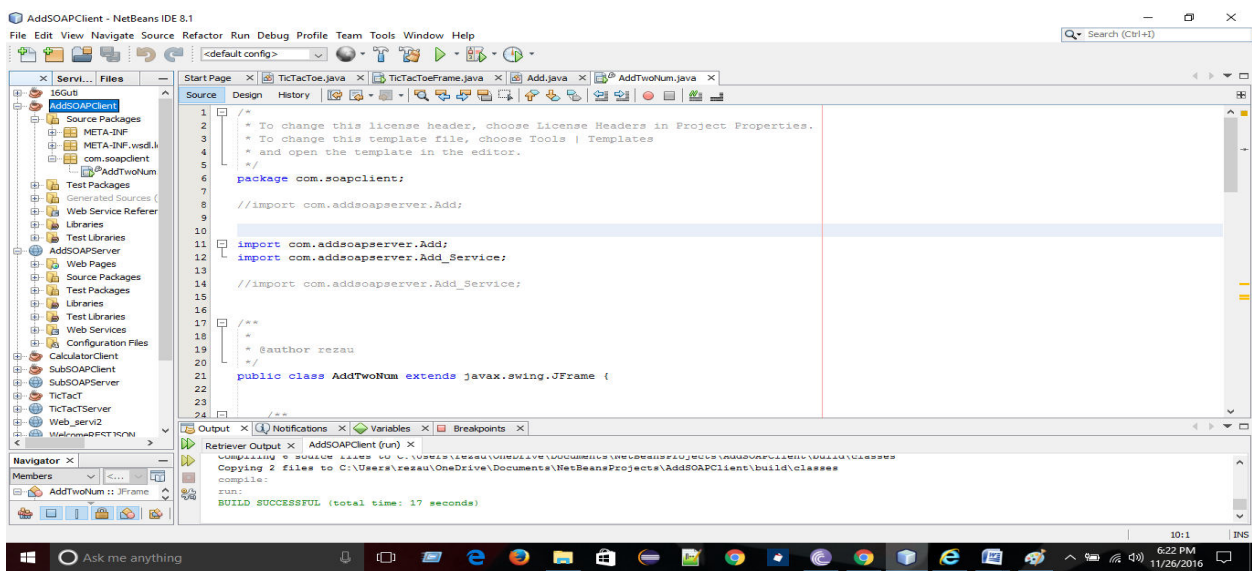


Figure 4.3.5: NetBeans Project tab after adding a web service reference to the project

Consuming the AddSOAPClient Web Service:

We will use a GUI application to interact with the AddSOAPClient web service. To build the client application’s GUI add a subclass of JFrame to the project by performing the following steps:

- 5. Right click the project name (AddSOAPClient) in the NetBeans Project tab and select New -> JFrame Form to display the New JFrame Form dialog.
- 6. Specify AddSOAPClient JFrame in the Class Name Field.
- 7. Specify com.deitel.welcomesoapclient in the package field.
- 8. Click **Finish** to close the New JFrame Form dialog.

Next use the NetBeans GUI design tools to build the GUI shown in the sample screen captures at the end. We use AddSOAPClient web service to display a integer value to the user.

Firstly, we will create a JFrame and use JLabel, JTextField also JSubmitButton. In the text field you need to give a name and click the submit button. You will see your desired output.

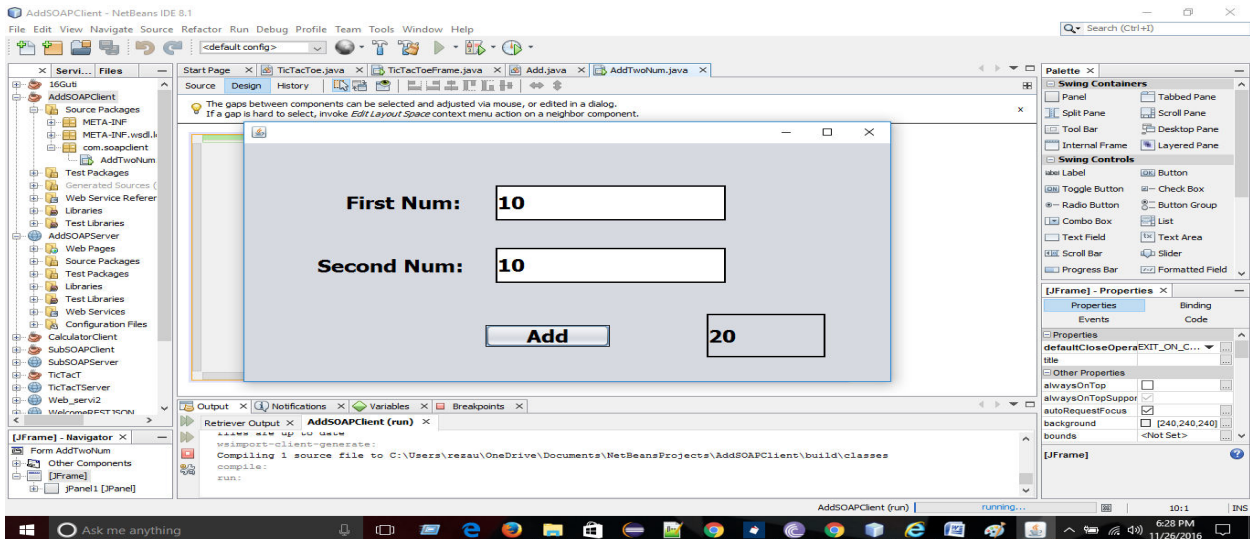


Figure 4.3.6: The desired output from server to client

Here the Client Side code:-

```

package com.soapclient;

import com.addsoapservice.Add;
import com.addsoapservice.Add_Service;

public class AddTwoNum extends javax.swing.JFrame {

    public AddTwoNum() {
        initComponents();
        SOAPproxy = (new Add_Service()).getAddPort();
    }

    private void addActionPerformed(java.awt.event.ActionEvent evt) {
        addTwoNumber();

        int result = SOAPproxy.add(num1, num2);
        resultTextLabel.setText(result + "");
    }

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {

```

```

        new AddTwoNum().setVisible(true);
    }
});
}
private javax.swing.JButton add;
private javax.swing.JTextField firstNumberTextField;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JPanel jPanel1;
private javax.swing.JLabel resultTextLabel;
private javax.swing.JTextField secondNumberTextField;
int num1, num2;
Add SOAPproxy;
public void addTwoNumber()
{
    num1 = Integer.parseInt(firstNumberTextField.getText());
    num2 = Integer.parseInt(secondNumberTextField.getText());
}
}

```

4.4 Publishing and Consuming SOAP-Based Sub web services

When you create a web service mainly focus on its logic, make sure that there is IDE and server handle its infrastructure. Firstly you create a Web application project after that you need to create a Client application project.

There are some steps for Creating web application project and Subtracting web service class in NetBeans. We begin with a SOAP-based web service.

@@:: Creating a Web Application Project in NetBeans and the following steps are given below:-

1. Select **File -> New Project** to open the new project dialog.
2. Select **Java Web** from the dialog's **Categories** list then select **Web Application** from the **Project** list. Click **Next >**.
3. Specify the name of your project (**AddSOAPServer**) in the **Project Name** field and specify where you want to store the project in the **Project Location** field. You can click the **Browse** button to select the location. Click **Next >**.
4. Select **GlassFish Server3** from the **Server** drop-down list and **Java EE 6 Web** from the **Java EE Version** drop-down list.
5. Click **Finish** to create the web service class.

@@:: Adding a Web Service Class to a Web Application Project in NetBeans and the following steps are given below:-

1. In the Projects tab in NetBeans right click the **AddSOAPServer** project's node and select **New -> Web Service** to open the New Web Service dialog.
2. Specify **WelcomeSOAP** in the **Web Service Name** field.
3. Specify **com.deitel.welcomesoap** in the Package field.
4. Click **Finish** to create the web service class.

Now your Web Service is available to client applications. When you build your application the IDE will generate other supporting files for your web services.

@@:: Defining the SubSOAPServer Web Service in NetBeans

The completed SubSOAPServer Service code is here:

```
package com.subtractsoap;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

@WebService(serviceName = "SubtractSOAP")

public class SubtractSOAP {

    @WebMethod(operationName = "sub")
```

```
public int sub(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {  
    int k = a - b;  
    return k;  
}  
}
```

@@:: Publishing the SubSOAPServer Web Service from NetBeans

NetBeans handles all the details of building and deploying a web service. Right click the project name **SubSOAPServer** in the Projects tab and select **Deploy** to build and deploy the web application to the GlassFish server.

@@:: Testing the SubSOAPServer Web Service with GlassFish Application Server's Tester Web Page

You need to test the **SubSOAPServer** web service. The GlassFish server is dynamically create a web page for you to test a web service method from a web browser. We need to use like this way:

1. Expand the project's **Web Service** in the NetBeans project tab.
2. Right click the web service class **name (SubSOAPServer)** and select Test Web Service.

Once you have deployed the web service, you can also see the URL

<http://localhost:8080/SubSOAPServer/SubtractSOAP?Tester>

To test **SubSOAPServer** welcome web method, type your values in the text field to the right of the **names** button and click the welcome button that will invoke the method.

You can access the server when server is on running mode. If NetBeans launches GlassFish for you, it will automatically shut it down when you close NetBeans. The GlassFish server shows how manually server start and stop, search this link

glassfish.java.net/downloads/quickstart/index.html

The web page will be look like:

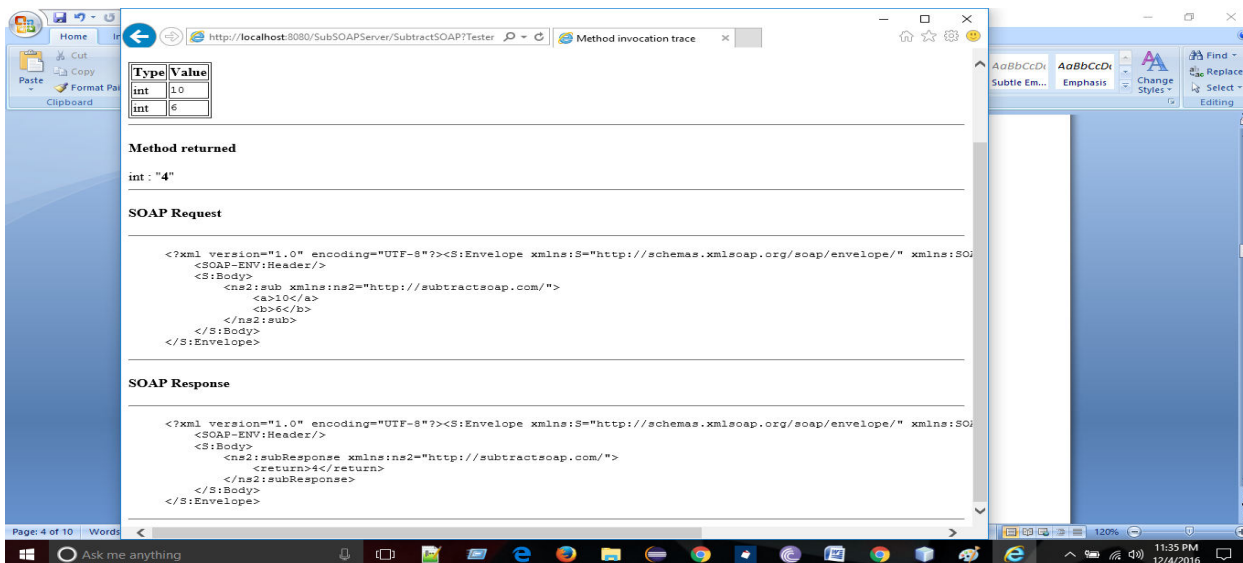


Figure 4.4.1: Testing SubSOAPServer method.

Creating a Client to Consume the SubSOAPClient Web Service

Creating a Desktop Application Project in NetBeans:-

Before performing those steps you need to ensure that the SubSOAPClient web service has been deployed and Glassfish server application is running. Perform the following steps to create a client Java desktop application in NetBeans:

1. Select **File** -> New project to open the New Project dialog.
2. Select **java** from the **Catalogs** list and **Java Application** from the list then click **Next** ->.
3. Specify name **SubSOAPClient** in the **Project Name** field and uncheck the Create Main Class checkbox. Later you will add a subclass of JFrame that contains a main method.
4. Click **Finish** to create the Project.

Step 2: Adding a web Service Reference to an Application

You will add a web service reference to your applicant that can interact with the SubSOAPClient web service.

1. Right click the **project name (SubSOAPClient)** in the NetBeans Projects tab and select **New** -> **Web Service Client** from the pop-up menu to display the **New Web Service Client** dialog.
2. In the **WSDL URL** field use the URL **http://localhost:8080/SubSOAPServer/SubtractSOAP?WSDL**. This URL tells the IDE where to find the web service's WSDL description. The IDE uses this WSDL description to generate the client-side artifacts that compose and support the service endpoint interface.
3. Click **Finish** to create the web service reference and dismiss the **New Web Service Client** dialog.

Consuming the SubSOAPClient Web Service:

We will use a GUI application to interact with the SubSOAPClient web service. To build the client application's GUI add a subclass of JFrame to the project by performing the following steps:

1. Right click the project name (SubSOAPClient) in the NetBeans Project tab and select New -> JFrame Form to display the New JFrame Form dialog.
2. Specify SubSOAPClient JFrame in the Class Name Field.
3. Specify com.deitel.welcomesoapclient in the package field.
4. Click **Finish** to close the New JFrame Form dialog.

Next use the NetBeans GUI design tools to build the GUI shown in the sample screen captures at the end. We use SubSOAPClient web service to display a integer value to the user.

Firstly, we will create a JFrame and use JLabel, JTextField also JSubmitButton.

In the text field you need to give two numbers and click the sub button. You will see your desired output.

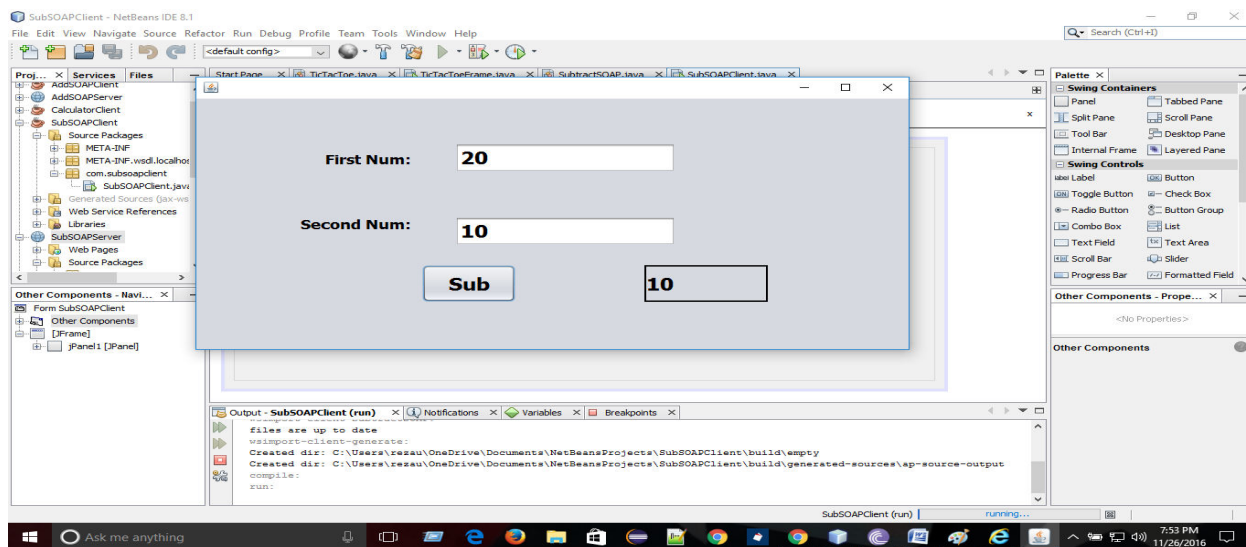


Figure 4.4.2: The desired output from server to client

Here the Client Side code:-

```
package com.subsoapclient;

import com.subtractsoap.SubtractSOAP;
import com.subtractsoap.SubtractSOAP_Service;

public class SubSOAPClient extends javax.swing.JFrame {

    public SubSOAPClient() {
        initComponents();
        SOAPproxy = (new SubtractSOAP_Service()).getSubtractSOAPPort();
    }

    private void subActionPerformed(java.awt.event.ActionEvent evt) {
```

```

    addTwoNumber();

    int result = SOAPproxy.sub(num1, num2);
    resultTextLabel.setText(result + "");
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new SubSOAPClient().setVisible(true);
        }
    });
}

private javax.swing.JTextField firstNumberTextField;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JPanel jPanel1;
private javax.swing.JLabel resultTextLabel;
private javax.swing.JTextField secondNumberTextField;
private javax.swing.JButton sub;
int num1, num2;
SubtractSOAP SOAPproxy;
public void addTwoNumber()
{
    num1 = Integer.parseInt(firstNumberTextField.getText());
    num2 = Integer.parseInt(secondNumberTextField.getText());
}
}

```

4.5 Publishing and Consuming SOAP-Based Add web service

When you create a web service mainly focus on its logic, make sure that there is IDE and server handle its infrastructure. Firstly you create a Web application project after that you need to create a Client application project.

There are some steps for Creating web application project and Adding web service class in NetBeans. We begin with a SOAP-based web service.

@@:: Creating a Web Application Project in NetBeans and the following steps are given below:-

6. Select **File -> New Project** to open the new project dialog.
7. Select **Java Web** from the dialog's **Categories** list then select **Web Application** from the **Project** list. Click **Next ->**.
8. Specify the name of your project (**AddServer**) in the **Project Name** field and specify where you want to store the project in the **Project Location** field. You can click the **Browse** button to select the location. Click **Next >**.
9. Select **GlassFish Server3** from the **Server** drop-down list and **Java EE 6 Web** from the **Java EE Version** drop-down list.
10. Click **Finish** to create the web service class.

@@:: Adding a Web Service Class to a Web Application Project in NetBeans and the following steps are given below:-

5. In the Projects tab in NetBeans right click the **AddServer** project's node and select **New -> Web Service** to open the New Web Service dialog.
6. Specify **AddServer** in the **Web Service Name** field.
7. Specify **com.addition** in the Package field.
8. Click **Finish** to create the web service class.

Now your Web Service is available to client applications. When you build your application the IDE will generate other supporting files for your web services.

@@:: Defining the AddServer Web Service in NetBeans

The completed AddServer Service code is here:

```
package com.addition;

import javax.jws.WebService;

import javax.jws.WebMethod;

import javax.jws.WebParam;

@WebService(serviceName = "Addition")
```

```

public class Addition {
    int k;
    @WebMethod(operationName = "add")
    public int add(@WebParam(name = "a") int a,@WebParam(name = "b") int b) {
        k=a+b;
        return k;
    }
    public int add1()
    {
        return k;
    }
}

```

Creating a Client to Consume the C1 and C2 Web Service

Creating a Desktop Application Project in NetBeans:-

Before performing those steps you need to ensure that the C1 and C2 web service has been deployed and Glassfish server application is running. Perform the following steps to create a client Java desktop application in NetBeans:

5. Select **File** -> New project to open the New Project dialog.
6. Select **java** from the **Catalogs** list and **Java Application** from the list then click **Next** ->.
7. Specify name **C1 and C2** in the **Project Name** field and uncheck the Create Main Class checkbox. Later you will add a subclass of JFrame that contains a main method.
8. Click **Finish** to create the Project.

Step 2: Adding a web Service Reference to an Application

You will add a web service reference to your applicant that can interact with the WelcomeSOAP web service.

4. Right click the **project name (C1 and C2)** in the NetBeans Projects tab and select **New** -> **C1 and C2** from the pop-up menu to display the **New Web Service Client** dialog.
5. In the **WSDL URL** field use the URL **http://localhost:8080/Calculator/Calculator?WSDL**. This URL tells the IDE where to find the web service's WSDL description. The IDE uses this WSDL description to generate the client-side artifacts that compose and support the service endpoint interface.

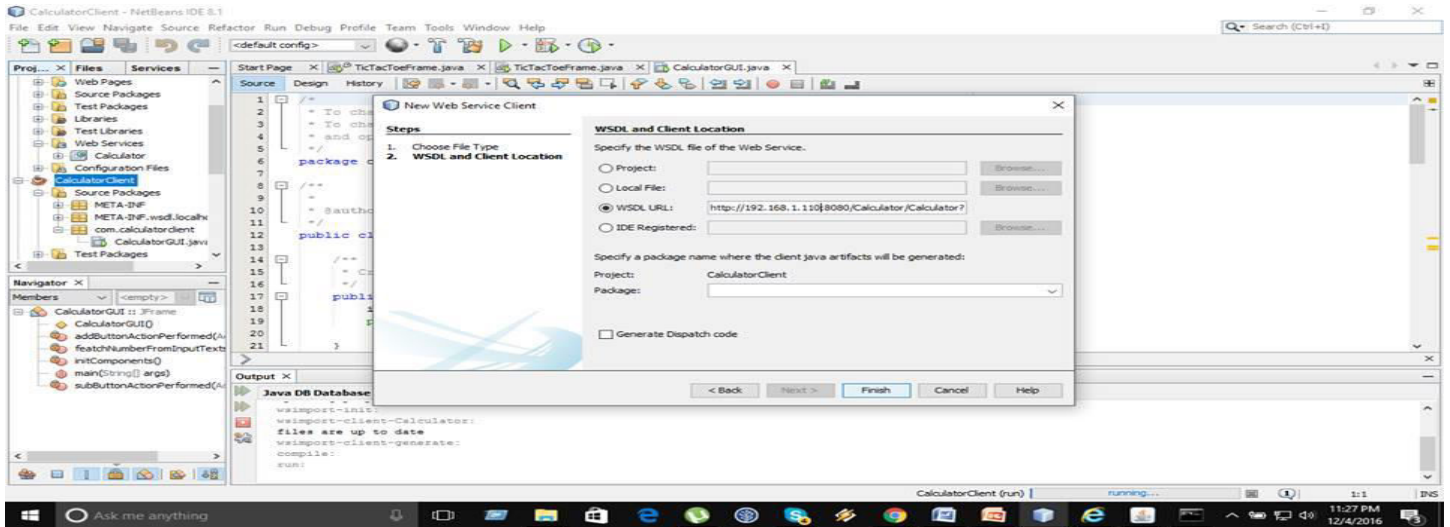


Figure 4.5.1: Show the server IP address in WSDL file

6. Click Finish to create the web service reference and dismiss the **New Web Service Client** dialog.

Consuming the AddSOAPClient Web Service:

We will use a GUI application to interact with the C1 and C2 web service. To build the client application's GUI add a subclass of JFrame to the project by performing the following steps:

5. Right click the project name (C1 and C2) in the NetBeans Project tab and select New -> JFrame Form to display the New JFrame Form dialog.
6. Specify C1 and C2 JFrame in the Class Name Field.
7. Specify com.client1 in the package field.
8. Click **Finish** to close the New JFrame Form dialog.

Next use the NetBeans GUI design tools to build the GUI shown in the sample screen captures at the end. We use AddSOAPClient web service to display a integer value to the user.

Firstly, we will create a JFrame and use JLabel, JTextField also JSubmitButton.

In the text field you need to give a name and click the submit button. You will see your desired output.

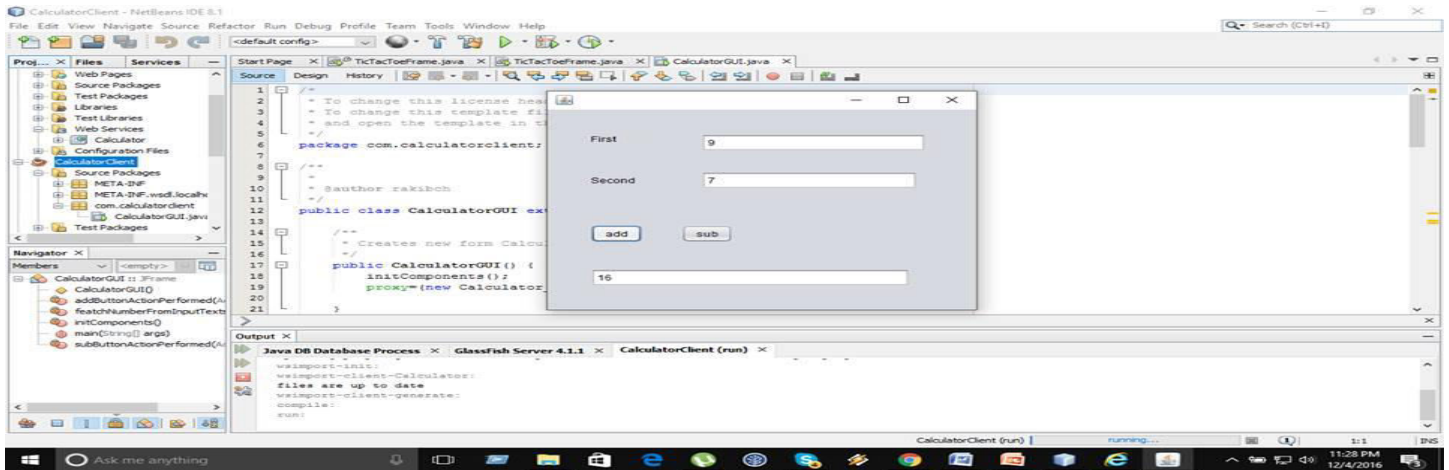


Figure 4.5.2: Show the desired output

Here the Client Side code:-

```

package com.client1;

import com.addition.Addition;

import com.addition.Addition_Service;

public class Client1 extends javax.swing.JFrame {

    public Client1() {

        initComponents();

        proxy=(Addition) (new Addition_Service()).getAdditionPort();

    }

    private void addButtonActionPerformed(java.awt.event.ActionEvent evt) {

        fetchNumberFromTextField();

        int result=proxy.add(num1, num2);

        resultTextField.setText(result+ "");

    }

    static Addition proxy;

    int num1,num2;

    public void fetchNumberFromTextField()

    {

```

```
    num1=Integer.parseInt(firstTextField.getText());  
    num2=Integer.parseInt(secondTextField.getText());  
}  
private javax.swing.JTextField firstTextField;  
private javax.swing.JTextField resultTextField;  
private javax.swing.JTextField secondTextField;  
}
```

Chapter 5:

Experimental Results

5.1 The steps and Outputs of our project

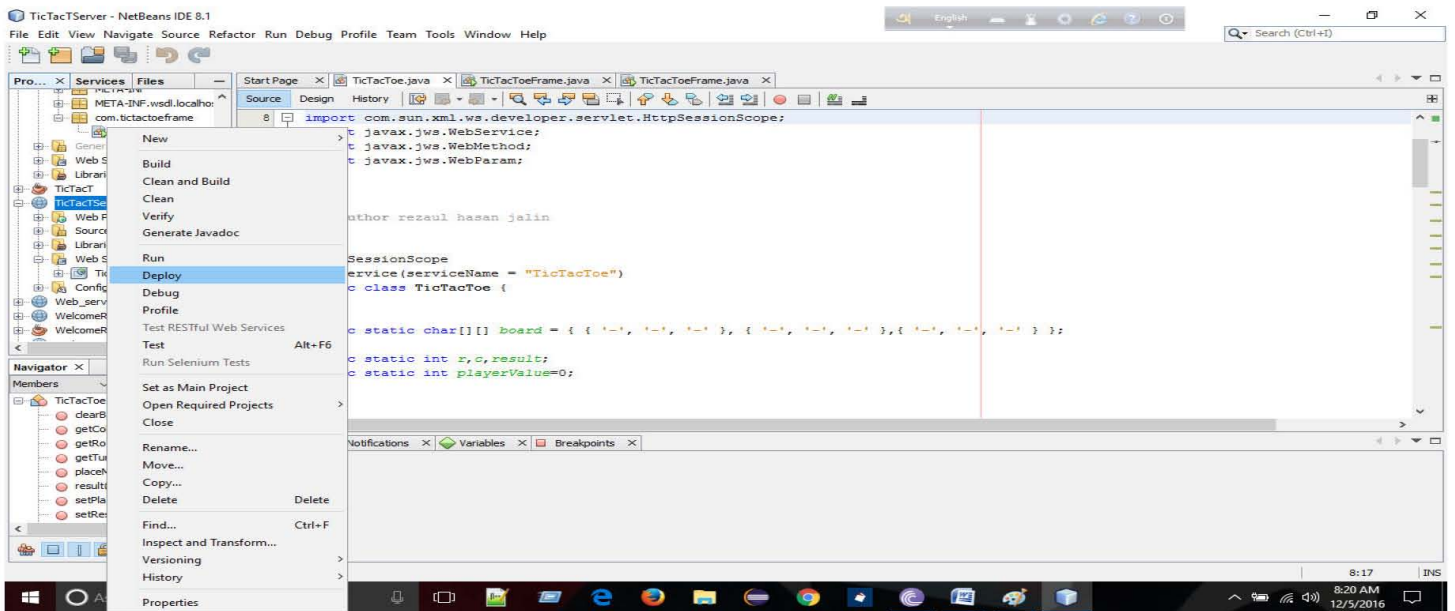


Figure 5.1: Deploy the Web Server

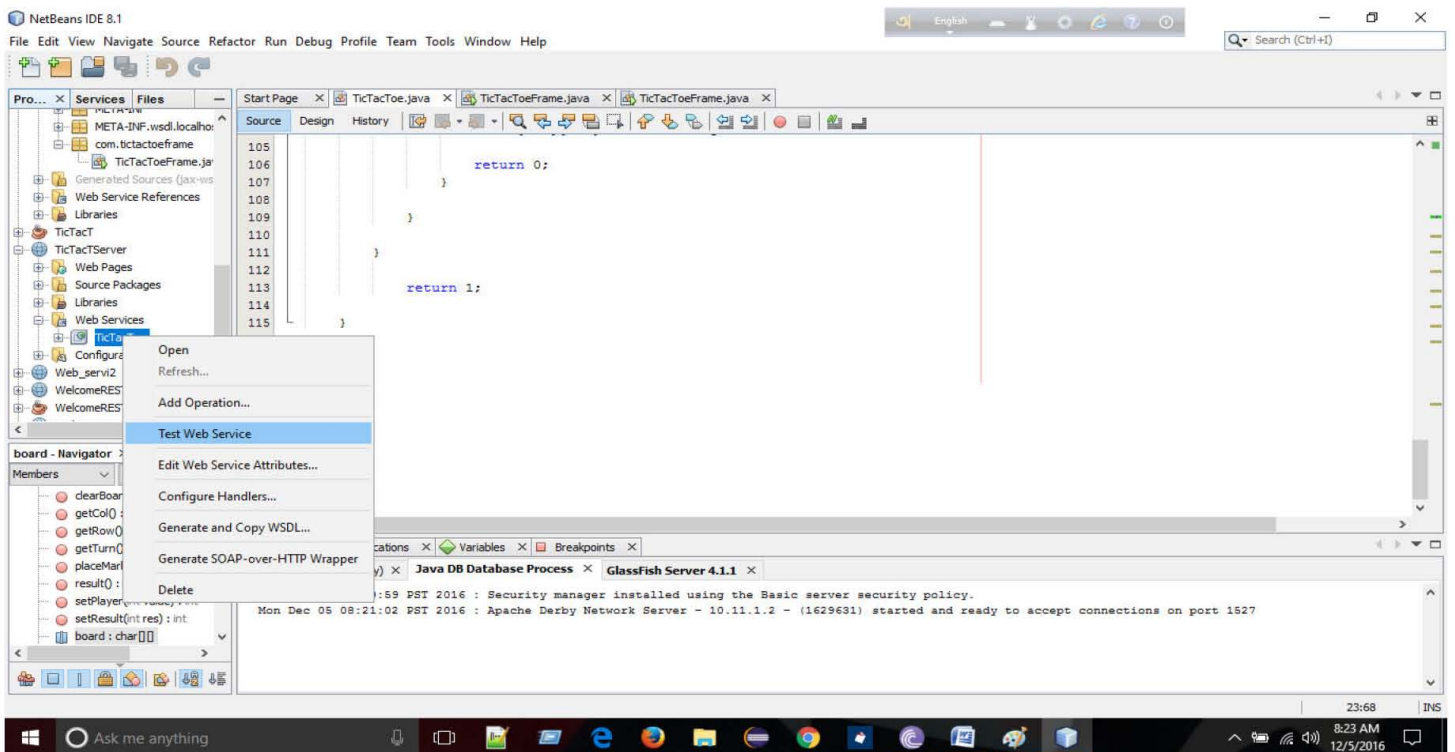


Figure 5.2: Test Web Service

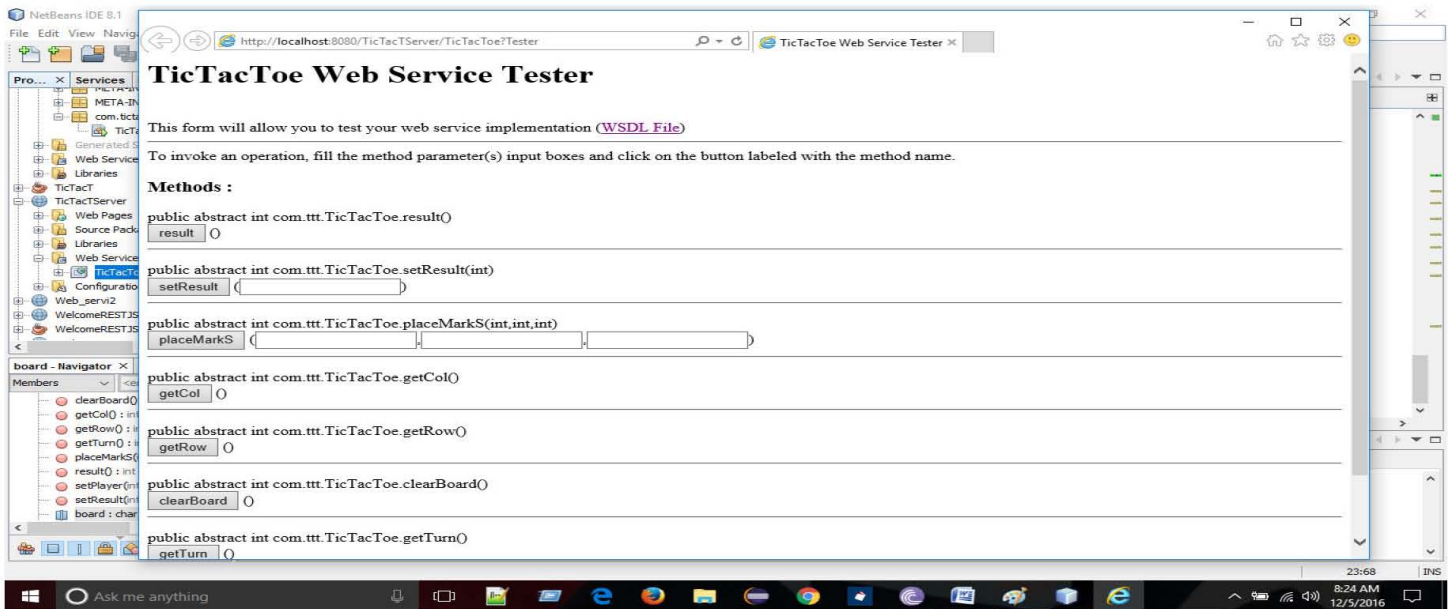


Figure 5.3: Show TicTacToe Web Service Tester

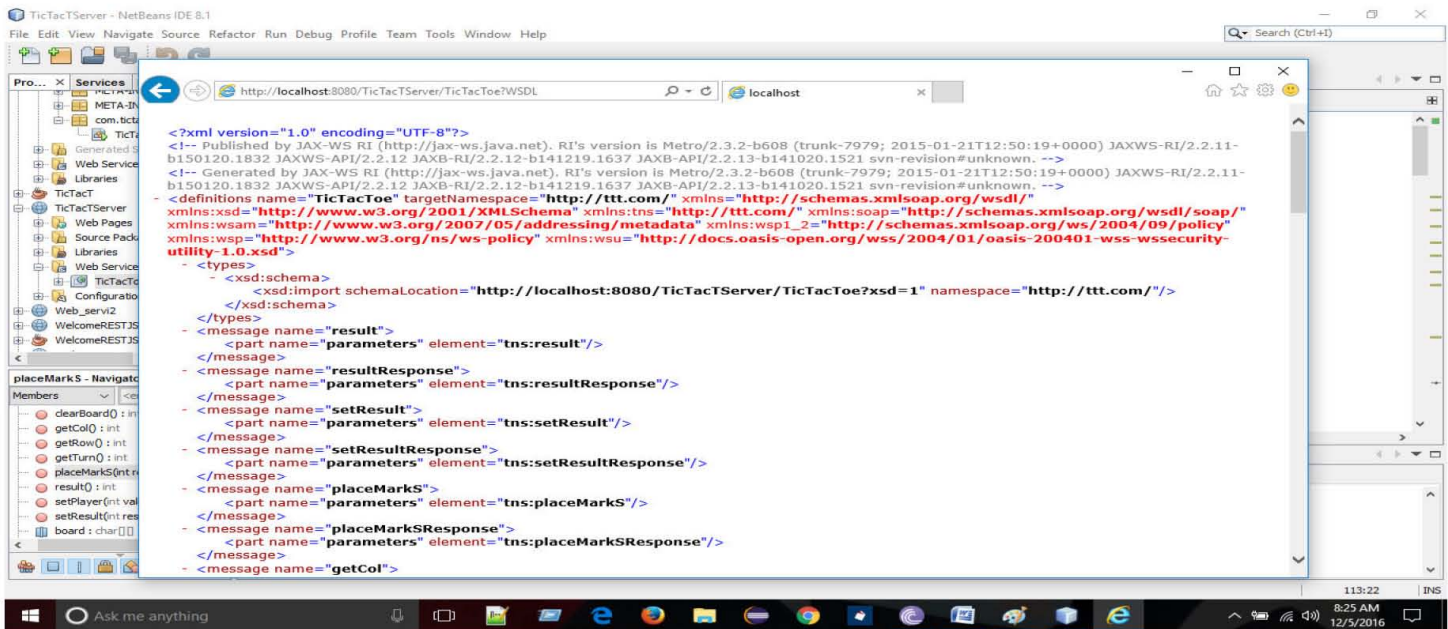


Figure 5.4: Show the WSDL file in local host

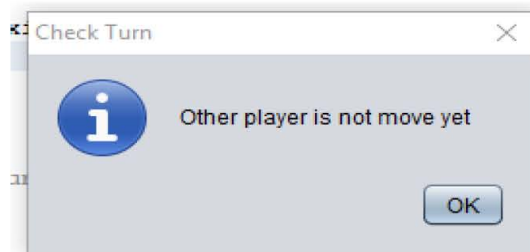


Figure 5.5: Show message

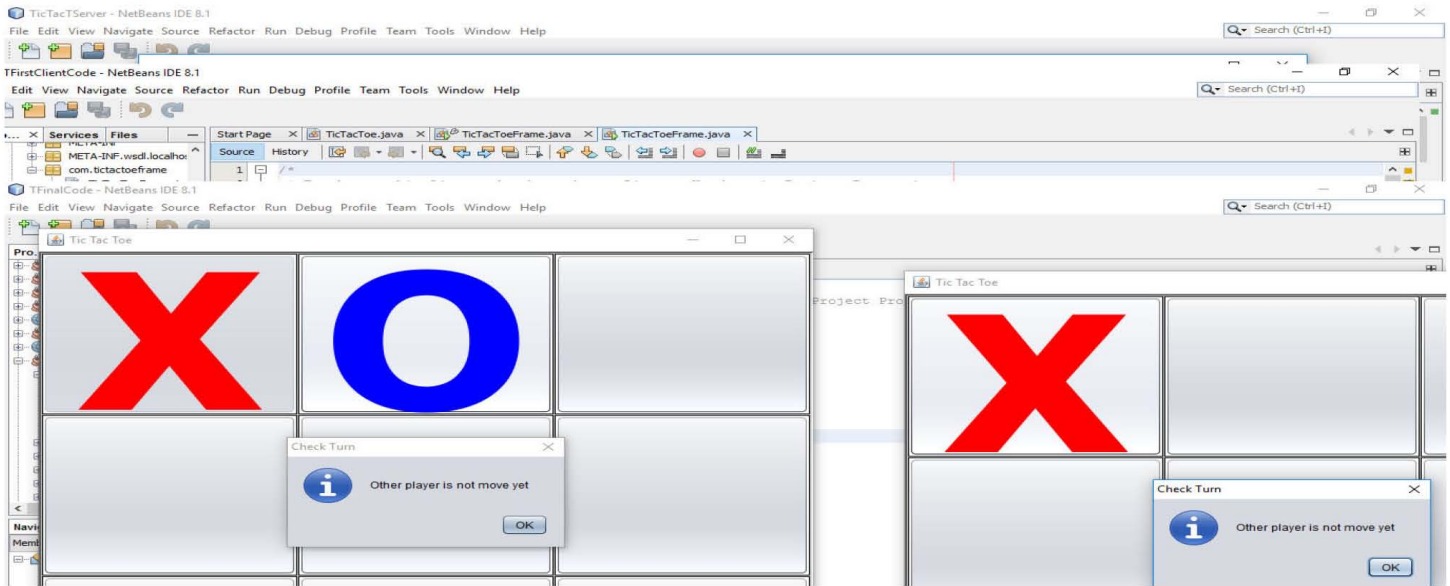


Figure 5.6: After press the first button instantly block this button

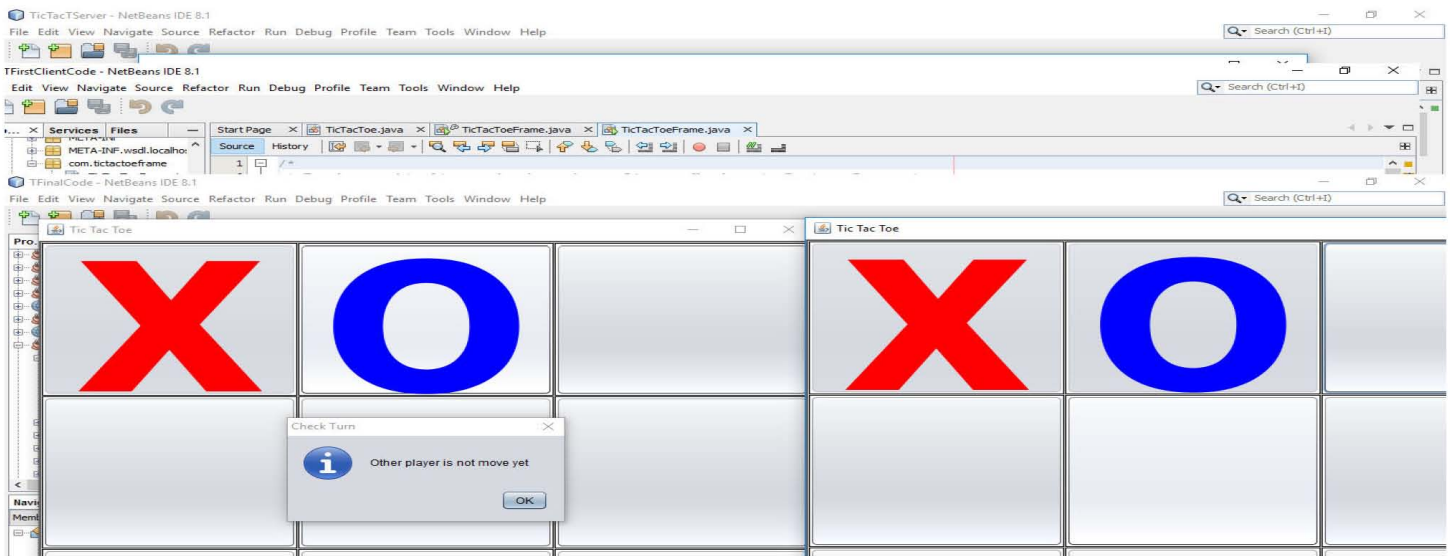


Figure 5.7: After second client click it graphically show and enable all buttons

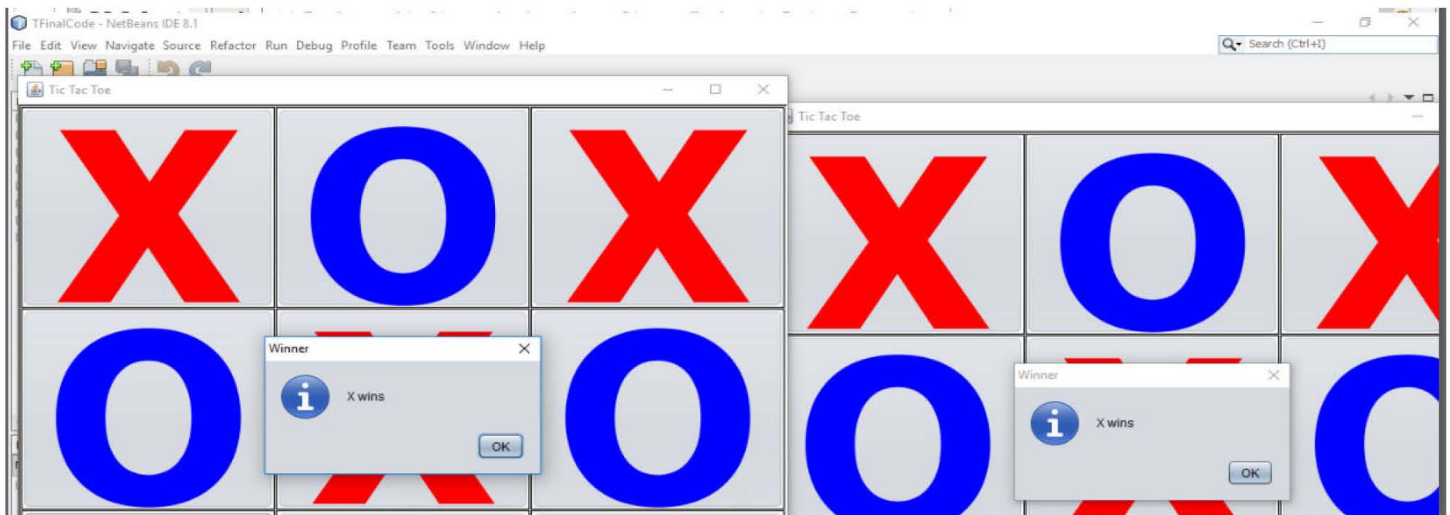


Figure 5.8: Game win, lose and tie message will show both clients

Chapter 6:

Conclusion and Future Work

6.1 Conclusion

In this project we have developed a simple gaming program with server and clients where server keep tracks of every movement by one client in server and according to that moves that server allows another client to move. We used Simple Object Access Protocol (SOAP) to complete our project.

6.2 Future Work

1. RESTful and JSON based system will be developed to create an Application protocol.
2. Traffic Management System (TMS) portal can be implemented.
3. Security will be interchanged with the web portal.
4. Database oriented model need to be interchanged.
5. Multiclients programming.

References

- [1] <http://www0.cs.ucl.ac.uk/staff/ucacwxe/lectures/ds98-99/dsee3.pdf>
- [2] <http://ecomputernotes.com/images/ATypical-View-of-Distributed-System.jpg>
- [3] <https://docs.puppet.com/mcollective/images/middleware.gif>
- [4] <https://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [5] http://www.ibm.com/support/knowledgecenter/SSBTEG_4.3.0/com.ibm.wbia_adapters.doc/doc/webservices/handlrsync.gif
- [6] http://www.w3schools.com/xml/xml_soap.asp

Server Side Code:

```
package com.ttt;

import com.sun.xml.ws.developer.servlet.HttpSessionScope;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

@HttpSessionScope
@WebService(serviceName = "TicTacToe")

public class TicTacToe {

public static char[][] board = { { '-', '-', '-' }, { '-', '-', '-' }, { '-', '-', '-' } };

public static int r,c,result;

public static int playerValue=0;

@WebMethod(operationName = "setResult")

public int setResult(@WebParam(name = "res")int res)

{

    result= res;

    return result;

}

@WebMethod(operationName = "result")

public int result()

{

    return result;

}

@WebMethod(operationName = "getRow")

public int getRow()

{
```

```

    return r;
}
@WebMethod(operationName = "getCol")
public int getCol()
{
    return c;
}
@WebMethod(operationName = "setPlayer")
public int setPlayer(@WebParam(name = "value")int value)
{
    playerValue=value;
    return playerValue;
}
@WebMethod(operationName = "getTurn")
public int getTurn()
{
    return playerValue;
}
@WebMethod(operationName = "clearBoard")
public int clearBoard()
{
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            board[i][j] = '-';
        }
    }
}

```

```

    result=0;

    playerValue=0;

    return 0;
}

@WebMethod(operationName = "placeMarkS")

public int placeMarkS(@WebParam(name = "row")int row,@WebParam(name = "col") int col
,@WebParam(name = "currentPlayer") char currentPlayer ) {

    r=row;

    c=col;

    playerValue=currentPlayer;

    if ((row >= 0) && (row < 3)) {

        if ((col >= 0) && (col < 3)) {

            if (board[row][col] == '-') {

                board[row][col] = currentPlayer;

                return 0;

            }

        }

    }

    return 1;

}
}

```

Client Side Codes:

```
/*
```

* To change this license header, choose License Headers in Project Properties.

* To change this template file, choose Tools | Templates

* and open the template in the editor.

*/

```
package com.tictactoe.frame;
```

```
import com.ttt.TicTacToe;
```

```
import static java.lang.System.exit;
```

```
import com.ttt.TicTacToe_Service;
```

```
import java.awt.Color;
```

```
import javax.swing.JOptionPane;
```

```
/**
```

```
*
```

```
* @author rezau
```

```
*/
```

```
public class TicTacToeFrame extends javax.swing.JFrame {
```

```
    private String whoseTurn = "X";
```

```
    private String PlayerOne = "Player One";
```

```
    private String PlayerTwo = "Player Two";
```

```
    private int playerOneCount = 0;
```

```
    private int playerTwoCount = 0;
```

```
    private String playersTurn = "";
```



```
/**
 * Creates new form TicTacToeFrame
 */
public TicTacToeFrame() {

    proxy=(TicTacToe)(new TicTacToe_Service()).getTicTacToePort();

    initComponents();
    setSize(750, 700);
    setLocationRelativeTo(null);
    getPlayerNames();
    setPlayerName();
    setScore();

}

public static void newGame(){

    int newGame = proxy.clearBoard();

}
```

```

private void tieGame()
{
    String one = jButton1.getText();
    String two = jButton2.getText();
    String three = jButton3.getText();
    String four = jButton4.getText();
    String five = jButton5.getText();
    String six = jButton6.getText();
    String seven = jButton7.getText();
    String eight = jButton8.getText();
    String nine = jButton9.getText();

    if( one != "" && two != "" && three != ""
        && four != "" && five != "" && six != ""
        && seven != "" && eight != "" && nine != "")
    {
        int w = proxy.setResult(33);
        viewResult();
    }
}

private void setScore()
{
    jLabel_Score.setText(PlayerOne + "'s Score is: " + String.valueOf(playerOneCount) + " \t " +
        playersTurn + "'s move " + PlayerTwo
        + "'s Score is: " + String.valueOf(playerTwoCount));
}

```

```

}

private void setPlayerName()
{

    /* if(whoseTurn.equalsIgnoreCase("X"))
    {
        playersTurn = PlayerOne;
    }
    else
    {
        playersTurn = PlayerTwo;
    }

    jLabel_Score.setText(PlayerOne + "'s Score is: " + String.valueOf(playerOneCount) + " \t " +
        playersTurn + "'s move " + PlayerTwo
    + "'s Score is: " + String.valueOf(playerTwoCount));
    */
}

```

```

private void getPlayerNames()
{
    /* PlayerOne = JOptionPane.showInputDialog(this,
        "Player One Name: ",
        "Player Name",
        JOptionPane.INFORMATION_MESSAGE);

```

```
PlayerTwo = JOptionPane.showInputDialog(this,  
    "Player two Name: ",  
    "Player Name",  
    JOptionPane.INFORMATION_MESSAGE);
```

```
if(PlayerOne.equals(""))  
{  
    PlayerOne = "Player One";  
}
```

```
if(PlayerTwo.equals(""))  
{  
    PlayerTwo = "Player Two";  
}  
*/  
}
```

```
private void determineWhoseTurn()  
{  
    if( whoseTurn.equalsIgnoreCase("X"))  
    {  
        whoseTurn = "O";  
    }  
    else  
    {
```

```

        whoseTurn = "X";
    }
}

private void viewResult(){

    int res = proxy.result();

    if(res==11){
        JOptionPane.showMessageDialog(this,
            "X wins",
            "Winner",
            JOptionPane.INFORMATION_MESSAGE);

        playerOneCount++;
        resetGame();
        enableButtons();
        exit(0);
    }
    else if(res==21){
        JOptionPane.showMessageDialog(this,
            "Y wins",
            "Winner",
            JOptionPane.INFORMATION_MESSAGE);

        playerTwoCount++;
    }
}

```

```
resetGame();
enableButtons();
exit(0);
}

else if(res==33){
    JOptionPane.showMessageDialog(this,
        "Game is a tie!",
        "Tie Game",
        JOptionPane.INFORMATION_MESSAGE);

    resetGame();

    enableButtons();
    exit(0);
}
else{

}

}

private void xWins()
{
    int w = proxy.setResult(11);
```

```
viewResult();

}

private void oWins()
{
    int w = proxy.setResult(21);
    viewResult();
}

private void enableButtons()
{
    jButton1.setEnabled(true);
    jButton2.setEnabled(true);
    jButton3.setEnabled(true);
    jButton4.setEnabled(true);
    jButton5.setEnabled(true);
    jButton6.setEnabled(true);
    jButton7.setEnabled(true);
    jButton8.setEnabled(true);
    jButton9.setEnabled(true);

}

private void executeTurn(){
```

```
int playerValue = proxy.getTurn();
System.out.println("Player value : "+playerValue);
if(playerValue==1)
{
    viewResult();
    checkTurn();
}
else if(playerValue==2){

    int row = proxy.getRow();
    int col = proxy.getCol();

    if(row==0 && col==0){
        jButton1.setText("O");

        jButton1.setForeground(Color.blue);

        jButton1.setEnabled(false);
    }
    else if(row==0 && col==1){
        jButton2.setText("O");

        jButton2.setForeground(Color.blue);

        jButton2.setEnabled(false);
```



```
}  
else if(row==0 && col==2){  
    jButton3.setText("O");  
  
jButton3.setForeground(Color.blue);  
  
jButton3.setEnabled(false);  
}  
else if(row==1 && col==0){  
    jButton4.setText("O");  
  
jButton4.setForeground(Color.blue);  
  
jButton4.setEnabled(false);  
}  
else if(row==1 && col==1){  
    jButton5.setText("O");  
  
jButton5.setForeground(Color.blue);  
  
jButton5.setEnabled(false);  
}  
else if(row==1 && col==2){  
    jButton6.setText("O");  
  
jButton6.setForeground(Color.blue);
```

```
jButton6.setEnabled(false);
}
else if(row==2 && col==0){
    jButton7.setText("O");

jButton7.setForeground(Color.blue);

jButton7.setEnabled(false);
}
else if(row==2 && col==1){
    jButton8.setText("O");

jButton8.setForeground(Color.blue);

jButton8.setEnabled(false);
}
else if(row==2 && col==2){
    jButton9.setText("O");

jButton9.setForeground(Color.blue);

jButton9.setEnabled(false);
}
viewResult();
```

```

    }
else{
    checkTurn();
}

}

private void checkTurn(){
    JOptionPane.showMessageDialog(this,
        "Other player is not move yet",
        "Check Turn",
        JOptionPane.INFORMATION_MESSAGE);
    excuteTurn();

}

private void resetGame()
{

    jButton1.setText("");
    jButton2.setText("");
    jButton3.setText("");
    jButton4.setText("");
    jButton5.setText("");
}

```

```
    jButton6.setText("");
    jButton7.setText("");
    jButton8.setText("");
    jButton9.setText("");

    setScore();
    newGame();
    enableButtons();
}
```

```
private void determineIfWin()
```

```
{
    String one = jButton1.getText();
    String two = jButton2.getText();
    String three = jButton3.getText();
    String four = jButton4.getText();
    String five = jButton5.getText();
    String six = jButton6.getText();
    String seven = jButton7.getText();
    String eight = jButton8.getText();
    String nine = jButton9.getText();

    // If x wins

    if( one == "X" && two == "X" && three == "X")
    {
```

```
xWins();  
}  
  
if( four == "X" && five == "X" && six == "X")  
{  
    xWins();  
}  
  
if( seven == "X" && eight == "X" && nine == "X")  
{  
    xWins();  
}  
  
if( one == "X" && four == "X" && seven == "X")  
{  
    xWins();  
}  
  
if( two == "X" && five == "X" && eight == "X")  
{  
    xWins();  
}  
  
if( three == "X" && six == "X" && nine == "X")  
{  
    xWins();  
}
```

```
}  
  
//  
  
if( seven == "X" && five == "X" && three == "X")  
{  
    xWins();  
}
```

```
if( one == "X" && five == "X" && nine == "X")  
{  
    xWins();  
}
```

```
// IF O Wins
```

```
if( one == "O" && two == "O" && three == "O")  
{  
    oWins();  
}
```

```
if( four == "O" && five == "O" && six == "O")  
{  
    oWins();  
}
```

```
}  
  
if( seven == "O" && eight == "O" && nine == "O")  
{  
    oWins();  
}
```

```
if( one == "O" && four == "O" && seven == "O")  
{  
    oWins();  
}
```

```
if( two == "O" && five == "O" && eight == "O")  
{  
    oWins();  
}
```

```
if( three == "O" && six == "O" && nine == "O")  
{  
    oWins();  
}  
//
```

```
if( seven == "O" && five == "O" && three == "O")  
{  
    oWins();  
}
```

```

    }

    if( one == "O" && five == "O" && nine == "O")
    {
        oWins();
    }
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    jLabel_Score = new javax.swing.JLabel();
    jPanel_GridHolder = new javax.swing.JPanel();
    jPanel_1 = new javax.swing.JPanel();
    jButton1 = new javax.swing.JButton();
    jPanel_2 = new javax.swing.JPanel();
    jButton2 = new javax.swing.JButton();
    jPanel_3 = new javax.swing.JPanel();
    jButton3 = new javax.swing.JButton();

```



```
jPanel_4 = new javax.swing.JPanel();
jButton4 = new javax.swing.JButton();
jPanel_5 = new javax.swing.JPanel();
jButton5 = new javax.swing.JButton();
jPanel_6 = new javax.swing.JPanel();
jButton6 = new javax.swing.JButton();
jPanel_7 = new javax.swing.JPanel();
jButton7 = new javax.swing.JButton();
jPanel_8 = new javax.swing.JPanel();
jButton8 = new javax.swing.JButton();
jPanel_9 = new javax.swing.JPanel();
jButton9 = new javax.swing.JButton();
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Tic Tac Toe");
```

```
jPanel1.setBackground(new java.awt.Color(153, 153, 153));
```

```
jPanel1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.LOWERED)
);
```

```
jPanel1.setLayout(new java.awt.BorderLayout());
```

```
jLabel_Score.setBackground(new java.awt.Color(255, 255, 255));
```

```
jLabel_Score.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
jLabel_Score.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
```

```
jLabel_Score.setText("Place Holder for Scores");
```

```
jPanel1.add(jLabel_Score, java.awt.BorderLayout.PAGE_END);
```

```

jPanel_GridHolder.setBackground(new java.awt.Color(204, 204, 204));
jPanel_GridHolder.setLayout(new java.awt.GridLayout(3, 3, 1, 1));

jPanel_1.setBackground(new java.awt.Color(255, 255, 255));
jPanel_1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel_1.setLayout(new java.awt.BorderLayout());

jButton1.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
jPanel_1.add(jButton1, java.awt.BorderLayout.CENTER);

jPanel_GridHolder.add(jPanel_1);

jPanel_2.setBackground(new java.awt.Color(255, 255, 255));
jPanel_2.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel_2.setLayout(new java.awt.BorderLayout());

jButton2.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

```

```

    }
});
jPanel_2.add(jButton2, java.awt.BorderLayout.CENTER);

jPanel_GridHolder.add(jPanel_2);

jPanel_3.setBackground(new java.awt.Color(255, 255, 255));
jPanel_3.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel_3.setLayout(new java.awt.BorderLayout());

jButton3.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});
jPanel_3.add(jButton3, java.awt.BorderLayout.CENTER);

jPanel_GridHolder.add(jPanel_3);

jPanel_4.setBackground(new java.awt.Color(255, 255, 255));
jPanel_4.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel_4.setLayout(new java.awt.BorderLayout());

jButton4.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N
jButton4.addActionListener(new java.awt.event.ActionListener() {

```

```

    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton4ActionPerformed(evt);
    }
});
jPanel_4.add(jButton4, java.awt.BorderLayout.CENTER);

jPanel_GridHolder.add(jPanel_4);

jPanel_5.setBackground(new java.awt.Color(255, 255, 255));
jPanel_5.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel_5.setLayout(new java.awt.BorderLayout());

jButton5.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N
jButton5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton5ActionPerformed(evt);
    }
});
jPanel_5.add(jButton5, java.awt.BorderLayout.CENTER);

jPanel_GridHolder.add(jPanel_5);

jPanel_6.setBackground(new java.awt.Color(255, 255, 255));
jPanel_6.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel_6.setLayout(new java.awt.BorderLayout());

```

```

jButton6.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N
jButton6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton6ActionPerformed(evt);
    }
});
jPanel_6.add(jButton6, java.awt.BorderLayout.CENTER);

jPanel_GridHolder.add(jPanel_6);

jPanel_7.setBackground(new java.awt.Color(255, 255, 255));
jPanel_7.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel_7.setLayout(new java.awt.BorderLayout());

jButton7.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N
jButton7.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton7ActionPerformed(evt);
    }
});
jPanel_7.add(jButton7, java.awt.BorderLayout.CENTER);

jPanel_GridHolder.add(jPanel_7);

jPanel_8.setBackground(new java.awt.Color(255, 255, 255));
jPanel_8.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

```

```

jPanel_8.setLayout(new java.awt.BorderLayout());

jButton8.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N
jButton8.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton8ActionPerformed(evt);
    }
});
jPanel_8.add(jButton8, java.awt.BorderLayout.CENTER);

jPanel_GridHolder.add(jPanel_8);

jPanel_9.setBackground(new java.awt.Color(255, 255, 255));
jPanel_9.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel_9.setLayout(new java.awt.BorderLayout());

jButton9.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N
jButton9.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton9ActionPerformed(evt);
    }
});
jPanel_9.add(jButton9, java.awt.BorderLayout.CENTER);

jPanel_GridHolder.add(jPanel_9);

```

```

jPanel1.add(jPanel_GridHolder, java.awt.BorderLayout.CENTER);

getContentPane().add(jPanel1, java.awt.BorderLayout.CENTER);

pack();
setLocationRelativeTo(null);
} // </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    jButton1.setText("X");

    jButton1.setForeground(Color.red);
    int res = proxy.placeMarkS(0, 0, 'x');
    int t = proxy.setPlayer(1);
    System.out.println("Result = " + res);

    // determineWhoseTurn();
    determineIfWin();
    tieGame();
    // setPlayerName();
    checkTurn();

    jButton1.setEnabled(false);

```

```
}
```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    jButton2.setText("X");
```

```
    jButton2.setForeground(Color.red);
```

```
    int res = proxy.placeMarkS(0, 1, 'x');
```

```
    int t = proxy.setPlayer(1);
```

```
    System.out.println("Result = " + res);
```

```
    // determineWhoseTurn();
```

```
    determineIfWin();
```

```
    tieGame();
```

```
    // setPlayerName();
```

```
    checkTurn();
```

```
    jButton2.setEnabled(false);
```

```
}
```

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    jButton3.setText("X");
```

```
    jButton3.setForeground(Color.red);
```

```
    int res = proxy.placeMarkS(0, 2, 'o');
```



```

        int t = proxy.setPlayer(1);
        System.out.println("Result = " + res);

// determineWhoseTurn();
determineIfWin();
tieGame();
// setPlayerName();
    checkTurn();
    jButton3.setEnabled(false);
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    jButton4.setText("X");

    jButton4.setForeground(Color.red);
    int res = proxy.placeMarkS(1, 0, 'x');
    int t = proxy.setPlayer(1);
    System.out.println("Result = " + res);

// determineWhoseTurn();
determineIfWin();
tieGame();
// setPlayerName();
    checkTurn();

```

```
    jButton4.setEnabled(false);  
}
```

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    jButton5.setText("X");
```

```
    jButton5.setForeground(Color.red);
```

```
    int res = proxy.placeMarkS(1, 1, 'x');
```

```
    int t = proxy.setPlayer(1);
```

```
    System.out.println("Result = " + res);
```

```
    // determineWhoseTurn();
```

```
    determineIfWin();
```

```
    tieGame();
```

```
    // setPlayerName();
```

```
    checkTurn();
```

```
    jButton5.setEnabled(false);
```

```
}
```

```
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    jButton6.setText("X");
```

```
    jButton6.setForeground(Color.red);
```

```

int res = proxy.placeMarkS(1, 2, 'x');

int t = proxy.setPlayer(1);

System.out.println("Result = " + res);

// determineWhoseTurn();
determineIfWin();
tieGame();
// setPlayerName();
checkTurn();
jButton6.setEnabled(false);
}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
jButton7.setText("X");

jButton7.setForeground(Color.red);
int res = proxy.placeMarkS(2, 0, 'x');
int t = proxy.setPlayer(1);
System.out.println("Result = " + res);

// determineWhoseTurn();
determineIfWin();
tieGame();
}

```

```

// setPlayerName();
    checkTurn();
    jButton7.setEnabled(false);
}

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    jButton8.setText("X");

    jButton8.setForeground(Color.red);
    int res = proxy.placeMarkS(2, 1, 'x');
    int t = proxy.setPlayer(1);
    System.out.println("Result = " + res);

    // determineWhoseTurn();
    determineIfWin();
    tieGame();
    // setPlayerName();
    checkTurn();
    jButton8.setEnabled(false);
}

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    jButton9.setText("X");
}

```

```

jButton9.setForeground(Color.red);

int res = proxy.placeMarkS(2, 2, 'x');

int t = proxy.setPlayer(1);

System.out.println("Result = " + res);

// determineWhoseTurn();

determineIfWin();

tieGame();

// setPlayerName();

checkTurn();

jButton9.setEnabled(false);
}

/**
 * @param args the command line arguments
 */

public static void main(String args[]) {

    /* Set the Nimbus look and feel */

    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */

    try {

        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {

            if ("Nimbus".equals(info.getName())) {

```

```

        javax.swing.UIManager.setLookAndFeel(info.getClassName());
        break;
    }
}
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(TicTacToeFrame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(TicTacToeFrame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(TicTacToeFrame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(TicTacToeFrame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    }
//</editor-fold>

/* Create and display the form */

java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {

```

```
        new TicTacToeFrame().setVisible(true);
        newGame();
    }
});
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JButton jButton6;
private javax.swing.JButton jButton7;
private javax.swing.JButton jButton8;
private javax.swing.JButton jButton9;
private javax.swing.JLabel jLabel_Score;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel_1;
private javax.swing.JPanel jPanel_2;
private javax.swing.JPanel jPanel_3;
private javax.swing.JPanel jPanel_4;
private javax.swing.JPanel jPanel_5;
private javax.swing.JPanel jPanel_6;
private javax.swing.JPanel jPanel_7;
```

```
private javax.swing.JPanel jPanel_8;
private javax.swing.JPanel jPanel_9;
private javax.swing.JPanel jPanel_GridHolder;
// End of variables declaration
static TicTacToe proxy;
}
```

Another Client Side Codes:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.tictactoe.frame;

import com.ttt.TicTacToe;
import com.ttt.TicTacToe_Service;
import java.awt.Color;
import static java.lang.System.exit;
import javax.swing.JOptionPane;

/**
 *
 * @author rezau
 */
```



```

public class TicTacToeFrame extends javax.swing.JFrame {

    private String whoseTurn = "X";
    private String PlayerOne = "Player One";
    private String PlayerTwo = "Player Two";
    private int playerOneCount = 0;
    private int playerTwoCount = 0;
    private String playersTurn = "";

    public int v=0;

    /**
     * Creates new form TicTacToeFrame
     */
    public TicTacToeFrame() {

        proxy=(TicTacToe)(new TicTacToe_Service()).getTicTacToePort();

        initComponents();

        setSize(750, 700);
        setLocationRelativeTo(null);
        getPlayerNames();
        setPlayerName();
        setScore();
        turnCheck();
    }
}

```

```
}
```

```
public static void newGame(){
```

```
    int newGame = proxy.clearBoard();
```

```
}
```

```
private void tieGame()
```

```
{
```

```
    String one = jButton1.getText();
```

```
    String two = jButton2.getText();
```

```
    String three = jButton3.getText();
```

```
    String four = jButton4.getText();
```

```
    String five = jButton5.getText();
```

```
    String six = jButton6.getText();
```

```
    String seven = jButton7.getText();
```

```
    String eight = jButton8.getText();
```

```
    String nine = jButton9.getText();
```

```
    if( one != "" && two != "" && three != ""
```

```
        && four != "" && five != "" && six != ""
```

```
        && seven != "" && eight != "" && nine != "")
```

```

{
    int w = proxy.setResult(33);
    viewResult();

}

}

private void setScore()
{
    jLabel_Score.setText(PlayerOne + "'s Score is: " + String.valueOf(playerOneCount) + " \t " +
        playersTurn + "'s move " + PlayerTwo
        + "'s Score is: " + String.valueOf(playerTwoCount));
}

private void setPlayerName()
{

    /* if(whoseTurn.equalsIgnoreCase("X"))
    {
        playersTurn = PlayerOne;
    }
    else
    {
        playersTurn = PlayerTwo;
    }
}

```

```

jLabel_Score.setText(PlayerOne + "s Score is: " + String.valueOf(playerOneCount) + " \t " +
    playersTurn + "'s move " + PlayerTwo
+ "'s Score is: " + String.valueOf(playerTwoCount));
*/
}

```

```

private void getPlayerNames()

```

```

{
    /*PlayerOne = JOptionPane.showInputDialog(this,
        "Player One Name: ",
        "Player Name",
        JOptionPane.INFORMATION_MESSAGE);

```

```

    PlayerTwo = JOptionPane.showInputDialog(this,
        "Player two Name: ",
        "Player Name",
        JOptionPane.INFORMATION_MESSAGE);

```

```

if(PlayerOne.equals(""))
{
    PlayerOne = "Player One";
}

```

```

if(PlayerTwo.equals(""))
{
    PlayerTwo = "Player Two";
}

```

```
    }*/  
}  
  
private void determineWhoseTurn()  
{  
    if( whoseTurn.equalsIgnoreCase("X"))  
    {  
        whoseTurn = "O";  
    }  
    else  
    {  
        whoseTurn = "X";  
    }  
}
```

```
private void viewResult(){  
  
    int res = proxy.result();  
  
    if(res==11){  
        JOptionPane.showMessageDialog(this,  
            "X wins",  
            "Winner",  
            JOptionPane.INFORMATION_MESSAGE);  
  
        playerOneCount++;  
    }  
}
```

```
resetGame();
enableButtons();
exit(0);
}
else if(res==21){
    JOptionPane.showMessageDialog(this,
        "O wins",
        "Winner",
        JOptionPane.INFORMATION_MESSAGE);

playerTwoCount++;
resetGame();
enableButtons();
exit(0);
}

else if(res==33){
    JOptionPane.showMessageDialog(this,
        "Game is a tie!",
        "Tie Game",
        JOptionPane.INFORMATION_MESSAGE);

    resetGame();

    enableButtons();
    exit(0);
```

```
    }  
    else{  
  
    }  
  
}
```

```
private void xWins()  
{  
    int w = proxy.setResult(11);  
    viewResult();  
  
}
```

```
private void oWins()  
{  
    int w = proxy.setResult(21);  
    viewResult();  
  
}
```

```
private void enableButtons()  
{  
    jButton1.setEnabled(true);  
    jButton2.setEnabled(true);  
}
```

```
        jButton3.setEnabled(true);
        jButton4.setEnabled(true);
        jButton5.setEnabled(true);
        jButton6.setEnabled(true);
        jButton7.setEnabled(true);
        jButton8.setEnabled(true);
        jButton9.setEnabled(true);

    }
```

```
private void executeTurn(){
    int playerValue = proxy.getTurn();

    System.out.println("Player value : "+playerValue);
    if(playerValue==2)
    {
        viewResult();
        checkTurn();
    }

    else if(playerValue==1){

        int row = proxy.getRow();
        int col = proxy.getCol();
```



```
if(row==0 && col==0){
jButton1.setText("X");

jButton1.setForeground(Color.red);

jButton1.setEnabled(false);
}
else if(row==0 && col==1){
    jButton2.setText("X");

jButton2.setForeground(Color.red);

jButton2.setEnabled(false);
}
else if(row==0 && col==2){
    jButton3.setText("X");

jButton3.setForeground(Color.red);

jButton3.setEnabled(false);
}
else if(row==1 && col==0){
    jButton4.setText("X");

jButton4.setForeground(Color.red);
```

```
jButton4.setEnabled(false);
}
else if(row==1 && col==1){
    jButton5.setText("X");

jButton5.setForeground(Color.red);

jButton5.setEnabled(false);
}
else if(row==1 && col==2){
    jButton6.setText("X");

jButton6.setForeground(Color.red);

jButton6.setEnabled(false);
}
else if(row==2 && col==0){
    jButton7.setText("X");

jButton7.setForeground(Color.red);

jButton7.setEnabled(false);
}
else if(row==2 && col==1){
    jButton8.setText("X");
```

```
jButton8.setForeground(Color.red);
```

```
jButton8.setEnabled(false);
```

```
}
```

```
else if(row==2 && col==2){
```

```
    jButton9.setText("X");
```

```
jButton9.setForeground(Color.red);
```

```
jButton9.setEnabled(false);
```

```
}
```

```
viewResult();
```

```
}
```

```
else{
```

```
    checkTurn();
```

```
}
```

```
}
```

```
public void turnCheck(){
```

```
    int playerValue = proxy.getTurn();
```

```

if(playerValue==0)
{
    JOptionPane.showMessageDialog(this,
        "Other player is not move yet",
        "Check Turn",
        JOptionPane.INFORMATION_MESSAGE);

    turnCheck();
}
else{
    excuteTurn();
}

}

public void checkTurn(){
    JOptionPane.showMessageDialog(this,
        "Other player is not move yet",
        "Check Turn",
        JOptionPane.INFORMATION_MESSAGE);
    excuteTurn();

}

```

```
private void resetGame()
{

    jButton1.setText("");
    jButton2.setText("");
    jButton3.setText("");
    jButton4.setText("");
    jButton5.setText("");
    jButton6.setText("");
    jButton7.setText("");
    jButton8.setText("");
    jButton9.setText("");

    setScore();
    newGame();
    enableButtons();

}
```

```
private void determineIfWin()
{
    String one = jButton1.getText();
    String two = jButton2.getText();
    String three = jButton3.getText();
    String four = jButton4.getText();
    String five = jButton5.getText();
```

```
String six = jButton6.getText();
```

```
String seven = jButton7.getText();
```

```
String eight = jButton8.getText();
```

```
String nine = jButton9.getText();
```

```
// If x wins
```

```
if( one == "X" && two == "X" && three == "X")
```

```
{
```

```
    xWins();
```

```
}
```

```
if( four == "X" && five == "X" && six == "X")
```

```
{
```

```
    xWins();
```

```
}
```

```
if( seven == "X" && eight == "X" && nine == "X")
```

```
{
```

```
    xWins();
```

```
}
```

```
if( one == "X" && four == "X" && seven == "X")
```

```
{
```

```
    xWins();
```

```
}
```

```
if( two == "X" && five == "X" && eight == "X")
{
    xWins();
}

if( three == "X" && six == "X" && nine == "X")
{
    xWins();
}

//

if( seven == "X" && five == "X" && three == "X")
{
    xWins();
}

if( one == "X" && five == "X" && nine == "X")
{
    xWins();
}

// IF O Wins
```

```
if( one == "O" && two == "O" && three == "O")
{
    oWins();
}
```

```
if( four == "O" && five == "O" && six == "O")
{
    oWins();
}
```

```
if( seven == "O" && eight == "O" && nine == "O")
{
    oWins();
}
```

```
if( one == "O" && four == "O" && seven == "O")
{
    oWins();
}
```

```
if( two == "O" && five == "O" && eight == "O")
{
    oWins();
}
```



```

if( three == "O" && six == "O" && nine == "O")
{
    oWins();
}
//

if( seven == "O" && five == "O" && three == "O")
{
    oWins();
}

if( one == "O" && five == "O" && nine == "O")
{
    oWins();
}
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

```

```
jPanel1 = new javax.swing.JPanel();
jLabel_Score = new javax.swing.JLabel();
jPanel_GridHolder = new javax.swing.JPanel();
jPanel_1 = new javax.swing.JPanel();
jButton1 = new javax.swing.JButton();
jPanel_2 = new javax.swing.JPanel();
jButton2 = new javax.swing.JButton();
jPanel_3 = new javax.swing.JPanel();
jButton3 = new javax.swing.JButton();
jPanel_4 = new javax.swing.JPanel();
jButton4 = new javax.swing.JButton();
jPanel_5 = new javax.swing.JPanel();
jButton5 = new javax.swing.JButton();
jPanel_6 = new javax.swing.JPanel();
jButton6 = new javax.swing.JButton();
jPanel_7 = new javax.swing.JPanel();
jButton7 = new javax.swing.JButton();
jPanel_8 = new javax.swing.JPanel();
jButton8 = new javax.swing.JButton();
jPanel_9 = new javax.swing.JPanel();
jButton9 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Tic Tac Toe");

jPanel1.setBackground(new java.awt.Color(153, 153, 153));
```

```
jPanel1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.LOWERED)
);
```

```
    jPanel1.setLayout(new java.awt.BorderLayout());
```

```
        jLabel_Score.setBackground(new java.awt.Color(255, 255, 255));
```

```
        jLabel_Score.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
        jLabel_Score.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
```

```
        jLabel_Score.setText("Place Holder for Scores");
```

```
        jPanel1.add(jLabel_Score, java.awt.BorderLayout.PAGE_END);
```

```
        jPanel_GridHolder.setBackground(new java.awt.Color(204, 204, 204));
```

```
        jPanel_GridHolder.setLayout(new java.awt.GridLayout(3, 3, 1, 1));
```

```
        jPanel_1.setBackground(new java.awt.Color(255, 255, 255));
```

```
        jPanel_1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
```

```
        jPanel_1.setLayout(new java.awt.BorderLayout());
```

```
        jButton1.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N
```

```
        jButton1.addActionListener(new java.awt.event.ActionListener() {
```

```
            public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
                jButton1ActionPerformed(evt);
```

```
            }
```

```
        });
```

```
        jPanel_1.add(jButton1, java.awt.BorderLayout.CENTER);
```

```
        jPanel_GridHolder.add(jPanel_1);
```

```

jPanel_2.setBackground(new java.awt.Color(255, 255, 255));
jPanel_2.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel_2.setLayout(new java.awt.BorderLayout());

jButton2.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});
jPanel_2.add(jButton2, java.awt.BorderLayout.CENTER);

jPanel_GridHolder.add(jPanel_2);

jPanel_3.setBackground(new java.awt.Color(255, 255, 255));
jPanel_3.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel_3.setLayout(new java.awt.BorderLayout());

jButton3.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});
jPanel_3.add(jButton3, java.awt.BorderLayout.CENTER);

```

```

jPanel_GridHolder.add(jPanel_3);

jPanel_4.setBackground(new java.awt.Color(255, 255, 255));
jPanel_4.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel_4.setLayout(new java.awt.BorderLayout());

jButton4.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N
jButton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton4ActionPerformed(evt);
    }
});
jPanel_4.add(jButton4, java.awt.BorderLayout.CENTER);

jPanel_GridHolder.add(jPanel_4);

jPanel_5.setBackground(new java.awt.Color(255, 255, 255));
jPanel_5.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel_5.setLayout(new java.awt.BorderLayout());

jButton5.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N
jButton5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton5ActionPerformed(evt);
    }
}

```

```

});

jPanel_5.add(jButton5, java.awt.BorderLayout.CENTER);

jPanel_GridHolder.add(jPanel_5);

jPanel_6.setBackground(new java.awt.Color(255, 255, 255));
jPanel_6.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel_6.setLayout(new java.awt.BorderLayout());

jButton6.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N
jButton6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton6ActionPerformed(evt);
    }
});

jPanel_6.add(jButton6, java.awt.BorderLayout.CENTER);

jPanel_GridHolder.add(jPanel_6);

jPanel_7.setBackground(new java.awt.Color(255, 255, 255));
jPanel_7.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel_7.setLayout(new java.awt.BorderLayout());

jButton7.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N
jButton7.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        jButton7ActionPerformed(evt);
    }
});
jPanel_7.add(jButton7, java.awt.BorderLayout.CENTER);

jPanel_GridHolder.add(jPanel_7);

jPanel_8.setBackground(new java.awt.Color(255, 255, 255));
jPanel_8.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel_8.setLayout(new java.awt.BorderLayout());

jButton8.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N
jButton8.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton8ActionPerformed(evt);
    }
});
jPanel_8.add(jButton8, java.awt.BorderLayout.CENTER);

jPanel_GridHolder.add(jPanel_8);

jPanel_9.setBackground(new java.awt.Color(255, 255, 255));
jPanel_9.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel_9.setLayout(new java.awt.BorderLayout());

jButton9.setFont(new java.awt.Font("Tahoma", 1, 250)); // NOI18N

```

```

jButton9.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton9ActionPerformed(evt);
    }
});

jPanel_9.add(jButton9, java.awt.BorderLayout.CENTER);

jPanel_GridHolder.add(jPanel_9);

jPanel1.add(jPanel_GridHolder, java.awt.BorderLayout.CENTER);

getContentPane().add(jPanel1, java.awt.BorderLayout.CENTER);

pack();
setLocationRelativeTo(null);
// turnCheck();
} // </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    jButton1.setText("O");

    jButton1.setForeground(Color.blue);
    int res = proxy.placeMarkS(0, 0, 'o');
    int t = proxy.setPlayer(2);

```



```

        System.out.println("Result = " + res);

// determineWhoseTurn();

    determineIfWin();

    tieGame();

// setPlayerName();

    checkTurn();

    jButton1.setEnabled(false);

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    jButton2.setText("O");

    jButton2.setForeground(Color.blue);

    int res = proxy.placeMarkS(0, 1, 'o');

    int t = proxy.setPlayer(2);

    System.out.println("Result = " + res);

// determineWhoseTurn();

    determineIfWin();

    tieGame();

// setPlayerName();

```

```
checkTurn();  
    jButton2.setEnabled(false);
```

```
}
```

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    jButton3.setText("O");
```

```
        jButton3.setForeground(Color.blue);
```

```
        int res = proxy.placeMarkS(0, 2, 'o');
```

```
int t = proxy.setPlayer(2);
```

```
    System.out.println("Result = " + res);
```

```
    // determineWhoseTurn();
```

```
    determineIfWin();
```

```
    tieGame();
```

```
    // setPlayerName();
```

```
checkTurn();
```

```
jButton3.setEnabled(false);
```

```
}
```

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    jButton4.setText("O");
```

```

        jButton4.setForeground(Color.blue);
        int res = proxy.placeMarkS(1, 0, 'o');
int t = proxy.setPlayer(2);
        System.out.println("Result = " + res);

// determineWhoseTurn();

determineIfWin();
tieGame();
// setPlayerName();

checkTurn();

jButton4.setEnabled(false);
}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    jButton5.setText("O");

    jButton5.setForeground(Color.blue);
        int res = proxy.placeMarkS(1, 1, 'o');
int t = proxy.setPlayer(2);
        System.out.println("Result = " + res);

// determineWhoseTurn();

determineIfWin();
tieGame();

```

```

// setPlayerName();

checkTurn();

jButton5.setEnabled(false);
}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

jButton6.setText("O");

jButton6.setForeground(Color.blue);

int res = proxy.placeMarkS(1, 2, 'o');

int t = proxy.setPlayer(2);

System.out.println("Result = " + res);

// determineWhoseTurn();

determineIfWin();

tieGame();

// setPlayerName();

checkTurn(); jButton6.setEnabled(false);
}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

jButton7.setText("O");

jButton7.setForeground(Color.blue);
}

```

```

        int res = proxy.placeMarkS(2, 0, 'o');
    int t = proxy.setPlayer(2);

        System.out.println("Result = " + res);

// determineWhoseTurn();

    determineIfWin();

    tieGame();

// setPlayerName();

    checkTurn();

    jButton7.setEnabled(false);
}

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    jButton8.setText("O");

        jButton8.setForeground(Color.blue);

        int res = proxy.placeMarkS(2, 1, 'o');

    int t = proxy.setPlayer(2);

        System.out.println("Result = " + res);

// determineWhoseTurn();

    determineIfWin();

    tieGame();

// setPlayerName();

    checkTurn();
}

```

```

    jButton8.setEnabled(false);
}

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    jButton9.setText("O");

    jButton9.setForeground(Color.blue);
    int res = proxy.placeMarkS(2, 2, 'o');
    int t = proxy.setPlayer(2);
    System.out.println("Result = " + res);

    // determineWhoseTurn();
    determineIfWin();
    tieGame();
    // setPlayerName();
    checkTurn();
    jButton9.setEnabled(false);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */

```

```

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
/* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
 * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
 */
try {
    for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(TicTacToeFrame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(TicTacToeFrame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(TicTacToeFrame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(TicTacToeFrame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    }
}
//</editor-fold>

```

```

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {

        new TicTacToeFrame().setVisible(true);
        newGame();

    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JButton jButton6;
private javax.swing.JButton jButton7;
private javax.swing.JButton jButton8;
private javax.swing.JButton jButton9;
private javax.swing.JLabel jLabel_Score;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel_1;
private javax.swing.JPanel jPanel_2;
private javax.swing.JPanel jPanel_3;

```



```
private javax.swing.JPanel jPanel_4;  
private javax.swing.JPanel jPanel_5;  
private javax.swing.JPanel jPanel_6;  
private javax.swing.JPanel jPanel_7;  
private javax.swing.JPanel jPanel_8;  
private javax.swing.JPanel jPanel_9;  
private javax.swing.JPanel jPanel_GridHolder;  
static TicTacToe proxy;  
}
```