

An Internship report on
E-VOTING MANAGEMENT SYSTEM

submitted for the partial fulfillment of the requirements of the award of degree of

Bachelor of Technology

In

Computer Science Engineering

Submitted by

K.RANGA DINAKAR(16341A0597)

Under the esteemed guidance of

A.Anusha



Department of Computer Science Engineering

G M R Institute of Technology

Jawaharlal Nehru Technological University, Kakinada

An Autonomous Institute, Affiliated to JNTUK, Kakinada, NAAC “A” Graded,
NBA Accredited, ISO 9001:2008 Certified Institution

G.M.R. Nagar, Rajam-532 127, A.P JUNE -2018

BONAFIDE CERTIFICATE

Certified that this internship report “**E-VOTING MANAGEMENT SYSTEM**” is the bonafide work of “**KOLLA RANGA DINAKAR (16341A0597)**” who carried out summer internship under our supervision at M/s “**SIGNER TECH**”, **HYDERABAD.**

Mr S.Velliangiri
Assistant Professor
CSE Dept.

Mr. A.V.RAMANA
Professor, HOD
CSE Dept.

ACKNOWLEDGEMENT

I express my profound sense of gratitude to my guide **Mr S.Velliangiri. Assistant Professor**, Department of Computer Science Engineering, for his keen guidance and valuable suggestions. I am deeply indebted to him for providing constant support throughout this work.

I am grateful my profound sense of gratitude to the respected **Head of the Department Dr.A.V.Ramana** has encouragement and providing necessary facilities in complete this project.

I also wish to express my profound sense of gratitude to the respected **Principal Dr.C.L.V.R.S.V.Prasad** for his help in completion of summer internship program

Last but not least we are thankful to all our staff members and friends who helped us a lot directly or indirectly in successfully completing this project.

CONTENT

1. Introduction

2. Basics Of Java

- 2.1. Overview Of Java
- 2.2. Features Of Java
- 2.3. Java ClassPath Setting
- 2.4. Steps For Compiling And Executing Programs
- 2.5. Difference Between JDK,JRE And JVM
- 2.6. Object And Class.
- 2.7. Data Types
- 2.8. Variables
- 2.9. Operators And Its Types

5. Programming Concepts

- 5.1. Structure Of Java Program
- 5.2. Main Method.
- 5.3. Decision Making Statement
- 5.4. Looping Statement
- 5.5. Wrapper Classes
- 5.6. Access Modifiers

6. Java Useful Keywords

- 6.1. Final
- 6.2. Static
- 6.3. This
- 6.4. Super
- 6.5. Synchronized

7. Java Classes, Blocks And Constructors

- 7.1. Abstract Classes.
- 7.2. Constructors
- 7.3. Constructors And Its Types
- 7.4. Static Block In Java
- 7.5. Relationship In Java.

8. Java Object Oriented

- 8.1 . Inheritance
- 8.2. Method Overloading
- 8.3. Method Overriding
- 8.4. Interface
- 8.5. Abstraction
- 8.6. Encapsulation
- 8.7. Polymorphism

9. Java Advanced

- 9.1. Package
- 9.2. Exception Handling
- 9.3. Multithreading
- 9.4. String
- 9.5. Collection

10. Projects Undertaken

INTRODUCTION

Java is one of the programming language or technology used for developing web applications. Java language developed at SUN Micro Systems in the year 1995 under the guidance of James

Gosling and there team. Originally SUN Micro Systems is one of the Academic university (Stanford University Network)

Whatever the software developed in the year 1990, SUN Micro Systems has released on the name of oak, which is original name of java (scientifically oak is one of the tree name). The OAK has taken 18 months to develop. The oak is unable to fulfill all requirements of the industry. So James Gosling again reviews this oak and released with the name of java in the year 1995. Scientifically java is one of the coffee seed name.

Java divided into three categories, they are

J2SE (Java 2 Standard Edition)

J2EE (Java 2 Enterprise Edition)

J2ME (Java 2 Micro or Mobile Edition)

J2SE

J2SE is used for developing client side applications.

J2EE

J2EE is used for developing server side applications.

J2ME

J2ME is used for developing mobile or wireless application by making use of predefined protocol called WAP (wireless Access / Application protocol).

BASICS OF JAVA

Overview Of Java

Java is a platform independent, more powerful, secure, high performance, multithreaded programming language. Here we discuss some points related to java.

Define JRE

The **Java Runtime Environment (JRE)** is part of the Java Development Kit (JDK). It contains set of libraries and tools for developing java application. The Java Runtime Environment provides the minimum requirements for executing a Java application.

Define JVM

JVM is set of programs developed by sun Micro System and supplied as a part of jdk for reading line by line of byte code and it converts into native understanding form of operating system. Java language is one of the compiled and interpreted programming language.

Garbage Collector

Garbage Collector is the system Java program which runs in the background along with regular Java program to collect un-Referenced (unused) memory space for improving the performance of our applications.

Define an API

An **API (Application Programming Interface)** is a collection of packages, a package is the collection of classes, interfaces and sub-packages. A sub-package is a collection of classes interfaces and sub sub packages etc. Java programming is containing user friendly syntax so that we can develop effective application s. in other words if any language is providing user friendly syntax, we can develop error free applications.

Definition of JIT

JIT is the set of programs developed by SUN Micro System and added as a part of JVM, to speed up the interpretation phase

Features Of JAVA

Features of a language are nothing but the set of services or facilities provided by the language vendors to the industry programmers. Some important features are:

Java Classpath Setting

Path Variable

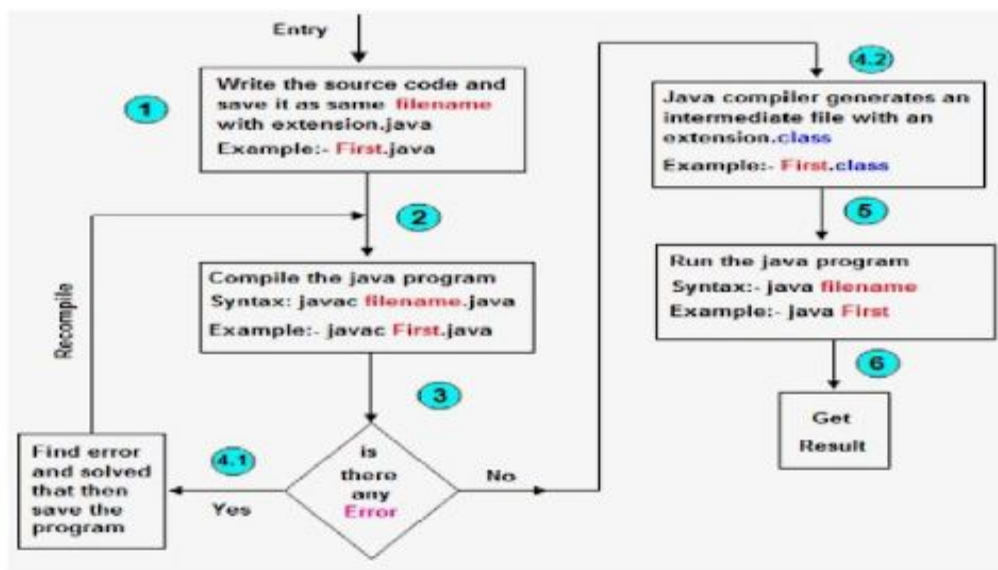
Path variable is set for providing path for all java tools like java, javac, javap, javah, jar, appletviewer which are use in java programming. These all tools are available in **bin** folders so we set path upto bin folders.

Classpath Variable

Classpath variable is set for providing path for predefined java classes which is used in our application. All classes are available in **lib/rt.jar** so we set classpath upto lib/rt.jar.

Steps For Compiling And Executing Programs

The following sequence of steps represented in the diagram use compiling the java program and executing the java programs.



Difference Between JDK,JRE AND JVM

Jvm, Jre, Jdk these all the backbone of java language. Each components have separate works. Jdk and Jre physically exists but Jvm are abstract machine it means it not physically exists.

JVM : JVM (Java Virtual Machine) is a software. It is a specification that provides runtime environment in which java bytecode can be executed. It not physically exists. JVMs are not same for all hardware and software, for example for window os JVM is different and for Linux VJM is different. JVM, JRE and JDK are platform dependent because configuration of each OS differs. But, Java is platform independent.

JRE : The Java Runtime Environment (JRE) is part of the Java Development Kit (JDK). It contains set of libraries and tools for developing java application. The Java Runtime Environment provides the minimum requirements for executing a Java application. It physically exists. It contains set of libraries + other files that JVM uses at runtime.

JDK : The Java Development Kit (JDK) is primary components. It physically exists. It is collection of programming tools and JRE, JVM.

Object: Object is the physical as well as logical entity where as class is the only logical entity.

Class: Class is a blueprint which is containing only list of variables and method and no memory is allocated for them. A class is a group of objects that has common properties.

A class in java contains:

- Data Member
- Method
- Constructor

- Block
- Class and Interface

Object: Object is a instance of class, object has state and behaviors. An Object in java has three characteristics:

- State
- Behavior
- Identity

State: Represents data (value) of an object.

Behavior: Represents the behavior (functionality) of an object such as deposit, withdraw etc. **Identity:** Object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. But, it is used internally by the JVM to identify each object uniquely. Class is also can be used to achieve user defined data types.

Operators And Its Types

Operator is a special symbol that tells the compiler to perform specific mathematical or logical Operation. Java supports following lists of operators.

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Bitwise Operators
- Assignment operator
- Ternary operators

Package is a collection of classes, interfaces and sub-packages. A sub package contains collection of classes, interfaces and sub-sub packages etc. `java.lang.*`; package is imported by default and this package is known as default package.

Class is keyword used for developing user defined data type and every java program must start with a concept of class.

Main() Method

main() method is starting execution block of a java program or any java program start their execution from main method. If any class contain main() method known as main class.

Decision Making Statement

Decision making statement statements is also called selection statement. That is depending on the condition block need to be executed or not which is decided by condition. If the condition is "true" statement block will be executed, if condition is "false" then statement block will not be executed. In java there are three types of decision making statement:

- if
- if-else
- switch
- for loop
- while loop
- do-while

Access modifiers:

Access modifiers are those which are applied before data members or methods of a class. These are used to where to access and where not to access the data members or methods. In java programming we have four access modifiers they are

Modifiers	Within Same Class	Within other class of Same package	Within derived class of other package	Within external Class of other package
Private (Class level A.S)	Yes	No	No	No
Default (Package level A.S)	Yes	Yes	No	No
Protected (Derived level A.S)	Yes	Yes	Yes	No
Public (Universal A.S)	Yes	Yes	Yes	Yes

JAVA USEFUL KEYWORDS

Final keyword

In java language final keyword can be used in following way.

- Final at variable level
- Final at method level
- Final at class level

Static keyword

Static keyword is used in java mainly for memory management. Static keyword are used with variables, methods, blocks and nested class. Static is a keyword that are used for share the same variable or method of a given class. This is used for a constant variable or a method that is the same for every instance of a class. The main method of a class is generally labeled static.

In java language static keyword can be used for following

1. variable (also known as class variable)
2. method (also known as class method)
3. block
4. nested class

This keyword

this is a reference variable that refers to the current object. It is a keyword in java language represents current class object

"**this**" keyword can be use in two ways.

- this . (this dot)
- this() (this off)

Super keyword

Super keyword in java is a reference variable that is used to refer parent

class object. **Super** is an implicit keyword create by JVM and supply each and every java program for performing important role in three places.

- At variable level
- At method level
- At constructor level

Synchronized Keyword

Synchronized Keyword is used for when we want to allow only one thread at a time then use Synchronized modifier. If a method or block declared as a Synchronized then at a time only one thread is allowed to operate on the given object.

JAVA CLASSES, BLOCKS AND CONSTRUCTORS

Abstract Classes

We know that every java program must start with a concept of class that is without classes concept there is no java program perfect. In java programming we have two types of classes they are

1. Concrete class
2. Abstract class

Constructor And Its Types

A **constructor** is a special member method which will be called implicitly (automatically) by the JVM whenever an object is created for placing user or programmer defined values in place of default values. In a single word constructor is a special member method which will be called automatically whenever object is created.

The purpose of constructor is to initialize an object called object initialization. Constructors are mainly create for initializing the object. Initialization is a process of assigning user defined values at the time of allocation of memory space.

Types of constructors

Based on creating objects in Java constructor are classified in two types. They are

- Default or no argument Constructor
- Parameterized constructor.

Static Block in Java

Static block is a set of statements, which will be executed by the JVM before execution of main method. At the time of class loading if we want to perform any activity we have to define that activity inside static block because static block execute at the time of class loading.

In a class we can take any number of static block but all these static block will be execute from top to bottom.

Relationship in Java

Type of relationship always makes to understand how to reuse the feature from one class to another class. In java programming we have two types of relationship they are.

- Is-A Relationship
- Has-A Relationship

JAVA OBJECT ORIENTED

Inheritance

The process of obtaining the data members and methods from one class to another class is known as **inheritance**. It is one of the fundamental features of object-oriented programming. A class that is declared with abstract keyword, is known as **abstract class**. An abstract class is one which is containing some defined method and some undefined method.

Types of Inheritance

- Single inheritance

- Multiple inheritance
- Hierarchical inheritance
- Multilevel inheritance
- Hybrid inheritance

Why use Inheritance ?

For Method Overriding (used for Runtime Polymorphism).

It's main uses are to enable polymorphism and to be able to reuse code for different classes by putting it in a common super class for code Re-usability

Method Overloading

Whenever same method name is existing multiple times in the same class with different number of parameter or different order of parameters or different types of parameters is known as **method overloading**.

Why method Overloading ?

Suppose we have to perform addition of given number but there can be any number of arguments, if we write method such as a(int, int) for two arguments, b(int, int, int) for three arguments then it is very difficult for you and other programmer to understand purpose or behaviors of method they can not identify purpose of method. So we use method overloading to easily figure out the program. For example above two methods we can write sum(int, int) and sum(int, int, int) using method overloading concept.

Different ways to overload the method

There are two ways to overload the method in java

- By changing number of arguments or parameters
- By changing the data type
- By changing the order of arguments.

Method Overriding

Whenever same method name is existing in both base class and derived class with same types of parameters or same order of parameters is known as **method Overriding**.

Advantage of Java Method Overriding

Method Overriding is used to provide specific implementation of a method that is already provided by its super class.

Method Overriding is used for Runtime Polymorphism

Interface

Interface is similar to class which is collection of public static final variables (constants) and abstract methods. The interface is a mechanism to achieve fully abstraction in java. There can be only abstract methods in the interface. It is used to achieve fully abstraction and multiple inheritance in Java.

Why we use Interface ?

It is used to achieve fully abstraction. By using Interface, you can achieve multiple inheritance in java. When we use abstract and when Interface. If we do not know about any things about implementation just we have requirement specification then we should be go for **Interface**

If we are talking about implementation but not completely (partially implemented) then we should be go for **abstract**

Abstraction

Abstraction is the concept of exposing only the required essential characteristics and behavior with respect to a context.

Hiding of data is known as **data abstraction**. In object oriented programming language this is implemented automatically while writing the code in the form of class and object.

Real life example of Abstraction:

Abstraction shows only important things to the user and hides the internal details for example when we ride a bike, we only know about how to ride bike but can not know about how it work ? and also we do not know internal functionality of bike.

Encapsulation

Encapsulation is a process of wrapping of data and methods in a single unit is called encapsulation. Encapsulation is achieved in java language by class concept. Combining of state and behavior in a single container is known as encapsulation. In java language encapsulation can be achieve using **class** keyword, state represents declaration of variables on attributes and behavior represents operations in terms of method.

Benefits of encapsulation

- Provides abstraction between an object and its clients.
- Protects an object from unwanted access by clients.

Example: A bank application forbids (restrict) a client to change an Account's balance

Polymorphism

The process of representing one form in multiple forms is known as **Polymorphism**. Here original form or original method always resides in base class and multiple forms represents overridden method which resides in derived classes.

Polymorphism is not a programming concept but it is one of the principal of OOPs. For many objects oriented programming language polymorphism principle is common but whose implementations are varying from one objects oriented programming language to another object oriented programming language.

Polymorphism principal is divided into two sub principal they are:

- Static or Compile time polymorphism
- Dynamic or Runtime polymorphism

JAVA ADVANCED

Package

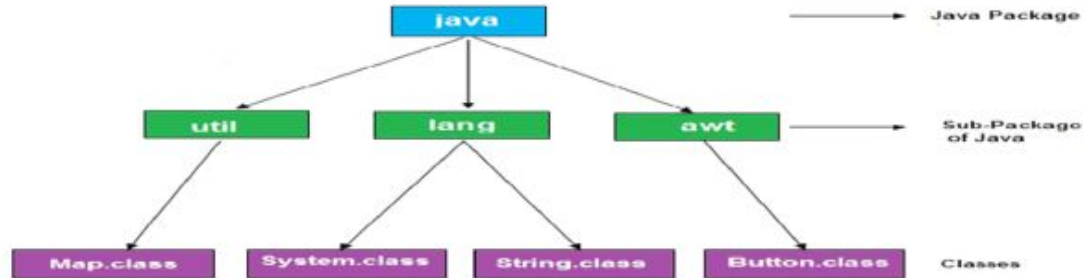
A package is a collection of similar types of classes, interfaces

and sub-packages.

The purpose of package concept is to provide common classes and interfaces for any program separately. In other words if we want to develop any class or interface which is common for most of the java programs than such common classes and interfaces must be place in a package.

Exception Handling:

Hierarchy of Exception classes



The process of converting system error messages into user friendly error message is known as **Exception handling**. This is one of the powerful feature of Java to handle run time error and maintain normal flow of java application.

An **Exception** is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's Instructions.

Type of Exception

- Checked Exception
- Un-Checked Exception

Multithreading

Multithreading in java is a process of executing multiple threads simultaneously.

The aim of multithreading is to achieve the concurrent execution **thread**

Thread is a lightweight components and it is a flow of control. In other words a flow of control is known as thread. State of a thread are classified into five types they are

1. New State
2. Ready State
3. Running State
4. Waiting State
5. Halted or dead State

String

String is a sequence of characters enclosed within double quotes (" ") is known as **String**. Example: "Java Programming".

In java programming to store the character data we have a fundamental datatype called **char**. Similarly to store the string data and to perform various operation on String data, we have three predefined classes they are:

- String
- StringBuffer
- StringBuilder

Collection

Collections in java is a framework that provides an architecture to store and manipulate the group of objects. All the operations that you perform on a data such as searching, sorting, insertion, manipulation, deletion etc. can be performed by Java Collections. Java Collection simply means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque etc.) and classes (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet etc).

ABSTRACT

- Computer and internet are the most brilliant gift of the science. It provides a better way of communication and interaction .As we all know that in present world everything based on internet so why not e-voting system.
- E-voting system project is implemented in java platform. Main aim of online voting system is to develop an online application like online reservation system, for citizen who are above 18 years of age to vote through online. Using these system citizens of India can vote through online without visiting polling booth. A centralized database is maintained where citizens information is maintained when ever citizen is using online voting system his/her information is authenticated with the data present in database if user is not in the list he cannot use e-voting system.

SOURCE CODE

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;
```

```
class Electronicvoting implements ActionListener  
{
```

```
    JFrame frame;
```

```
    JButton button;
```

```
    JButton button2;
```

```
    JRadioButton A;
```

```
    JRadioButton B;
```

```
    JRadioButton C;
```

```
    JRadioButton D;
```

```
    JRadioButton E;
```

```
    JLabel l=new JLabel("ELECTRONIC VOTING  
");
```

```
    int t=0;
```

```
    int votesA=0;
```

```
    int votesB=0;
```

```
    int votesC=0;
```

```
    int votesD=0;
```

```
    int votesE=0;
```

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

```
    {
```

```
        new Electronicvoting();
```

```
    }
```

```
    Electronicvoting()
```

```
    {
```

```
frame = new JFrame("ELECTION VOTING");

JPanel panel = new JPanel(new GridLayout(6, 1));
JPanel buttons = new JPanel(new FlowLayout());
Container contentPane = frame.getContentPane();
contentPane.setLayout(new BorderLayout());
contentPane.add(panel, BorderLayout.CENTER);
contentPane.add(buttons, BorderLayout.SOUTH);


button = new JButton("Submit");
button2 = new JButton("Results");


button.addActionListener(this);
button2.addActionListener(this);


A = new JRadioButton("1.#");
B = new JRadioButton("2.@");
C = new JRadioButton("3.$");
D = new JRadioButton("4.&");
E = new JRadioButton("5.OTHERS");


ButtonGroup bg = new ButtonGroup ();


bg.add(A);
bg.add(B);
bg.add(C);
bg.add(D);
bg.add(E);
panel.add(l);
panel.add(A);
panel.add(B);
panel.add(C);
panel.add(D);
panel.add(E);
buttons.add(button);
buttons.add(button2);
```

```
int fw = 300;
int fh = 400;
Dimension screensize = Toolkit.getDefaultToolkit().getScreenSize();
frame.setBounds((int) screensize.getWidth() - fw, 0, fw, fh);

frame.setVisible(true);
frame.setDefaultCloseOperation(frame.EXIT_ON_CLOSE);
}

public void actionPerformed(ActionEvent e) {

    if(e.getSource()==(button)) {
        if(A.isSelected()) {
            votesA++;
            System.out.println("TDP(#) voted for!" + votesA);
        } else if(B.isSelected()) {
            votesB++;
            System.out.println("BJP(@) voted for!" + votesB);
        } else if(C.isSelected()) {
            votesC++;
            System.out.println("YCP($) voted for!" + votesC);
        } else if(D.isSelected()) {
            votesD++;
            System.out.println("LOKSATHA(&) voted
for!" + votesD);
        } else if(E.isSelected()) {
            votesE++;
            System.out.println("OTHERS voted for!" + votesE);
        }
        }

    t = votesA + votesB + votesC + votesD + votesE;
    System.out.println("total = " + t);

    }

    if (e.getSource()==(button2)) {
        System.out.println("votes scured:" + votesA + " "
+ votesB + " " + votesC + " " + votesD + " " + votesE);
    }
```

```
        int ap = (votesA*
100/(votesA+votesB+votesC+votesD+votesE));
        System.out.println("TDP(#) has " + ap + "% of votes ");
        int bp = (votesB*
100/(votesA+votesB+votesC+votesD+votesE));
        System.out.println("BJP(@) has " + bp + "% of votes ");
        int cp = (votesC*
100/(votesA+votesB+votesC+votesD+votesE));
        System.out.println("YCP($) has " + cp + "% of votes ");
        int dp = (votesD*
100/(votesA+votesB+votesC+votesD+votesE));
        System.out.println("LOKSATHA(&) has " + dp + "% of
votes ");
        int ep = (votesE*
100/(votesA+votesB+votesC+votesD+votesE));
        System.out.println("OTHERS has " + ep + "% of votes
");
        int a[]=new int[5];

        a[0]=votesA;
        a[1]=votesB;
        a[2]=votesC;
        a[3]=votesD;
        a[4]=votesE;

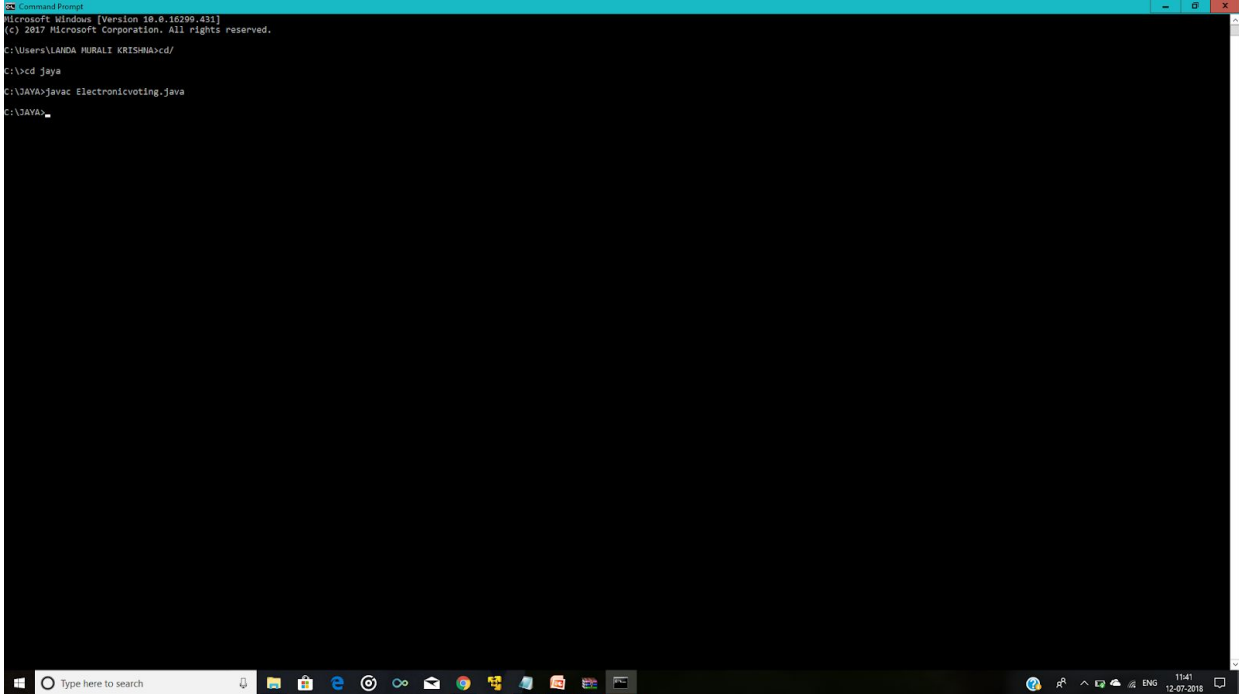
        for(int i=0;i<5;i++)
        {
            for(int j=1;j<5-i;j++)
            {
                if(a[i]<a[j])
                {
                    int temp;
                    temp=a[i];
                    a[i]=a[j];
                    a[j]=temp;
                }
            }
        }
        System.out.println();
        if(a[0]==votesA)
```



```
System.out.println("TDP(#) has been elected");
if(a[0]==votesB)
System.out.println("BJP(@) has been elected");
if(a[0]==votesC)
System.out.println("YCP($) has been elected");
if(a[0]==votesD)
System.out.println("LOKSATHA(&) has been elected");
if(a[0]==votesE)
System.out.println("others has been elected");
System.out.println("percentage of participation is " +((t*100)/20)+"% ");

}
}
}
```

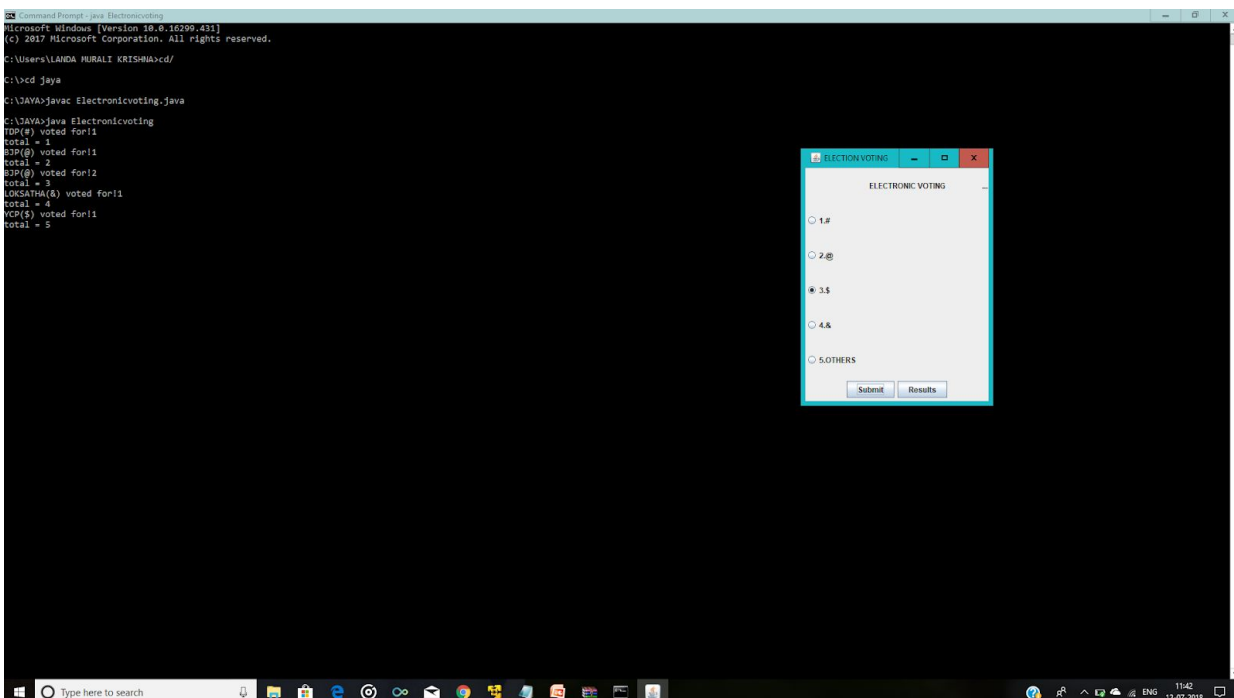
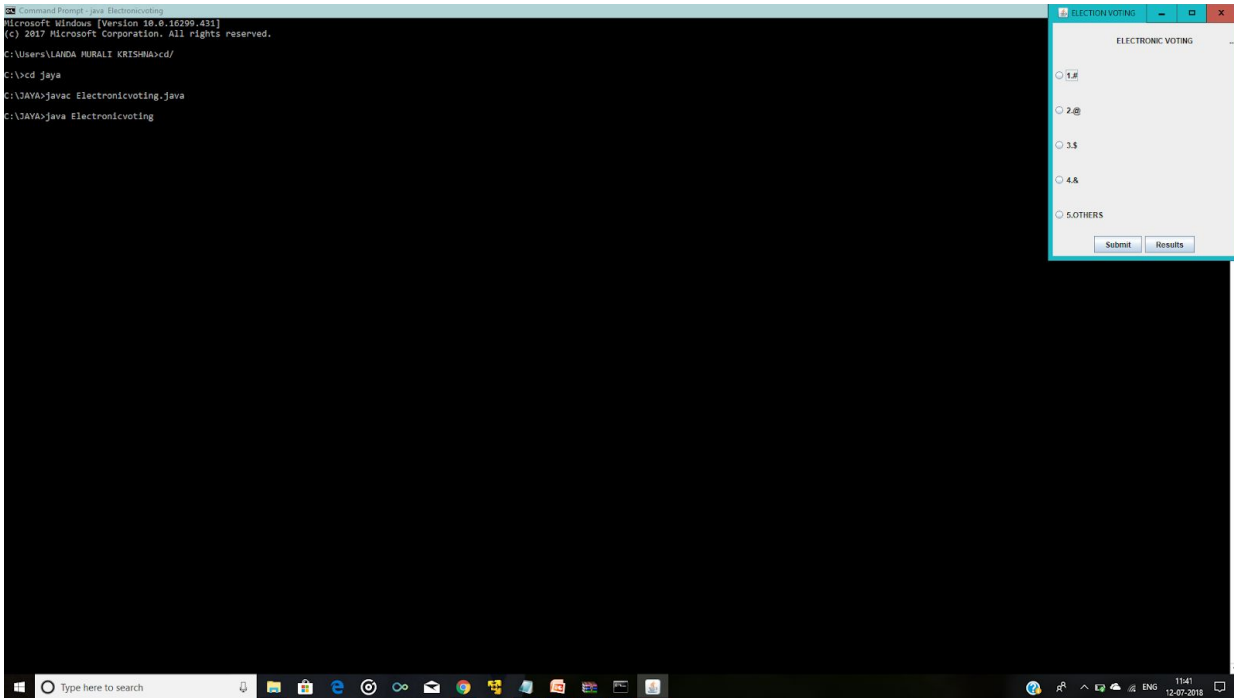
OUTPUT:

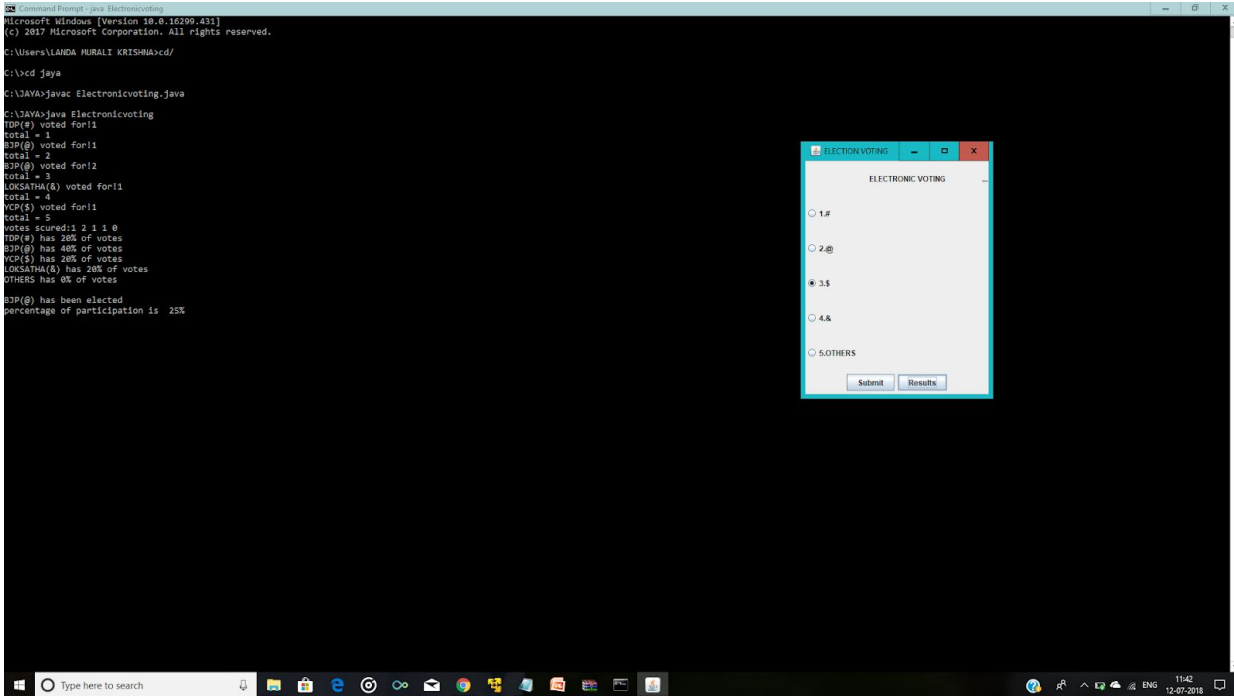


```
Microsoft Windows [Version 10.0.16299.431]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\LANOA MURALI KRESHA>cd /
C:\>cd java
C:\JAVA>javac Electronicvoting.java
C:\JAVA>
```

The screenshot shows a Windows Command Prompt window with a black background and white text. The window title is "Command Prompt". The text inside shows the user navigating to the "java" directory and compiling a file named "Electronicvoting.java" using the "javac" command. The compilation appears to be successful as there are no error messages. The Windows taskbar is visible at the bottom, showing various application icons and the system clock indicating 11:41 on 12-07-2018.





```
Microsoft Windows [Version 10.0.16299.431]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\LANDA MURALI KRISHNA>cd /

C:\>cd java

C:\JAVA>javac ElectronicVoting.java

C:\JAVA>java ElectronicVoting
TDP($) voted for1
total = 1
BJP($) voted for1
total = 2
BJP($) voted for2
total = 3
LOKSATHA($) voted for1
total = 4
YCP($) voted for1
total = 5
votes scored:1 2 1 1 0
TDP($) has 20% of votes
BJP($) has 40% of votes
YCP($) has 20% of votes
LOKSATHA($) has 20% of votes
OTHERS has 0% of votes

BJP($) has been elected
percentage of participation is 25%
```

CONCLUSION

- This system is able to store billing records securely and retrieve the records whenever needed easily. Data entering of customers.
- Customers, restaurant records and employees are interconnected in order to maintain the accuracy of the system. This system can also be further improved adding many other features and including the other systems as well.
- Finally we believe that we able to launch an effective computerized system to the restaurant causing the restaurant to perform well in the future regarding the billing and restaurant records.