Nusratullaev Abrorjon Kholkhujaevich

Project name: CalculateVote

1. Introduction

The "CalculateVote" program is an innovative electronic voting system designed to enhance transparency and accessibility in democratic processes. This report delves into the program's architecture, functionalities, and potential for future development. Built upon Java and leveraging CSV databases, the system offers a user-friendly interface for seamless voting while upholding robust verification mechanisms.

The traditional method of conducting elections can be cumbersome and prone to errors. "CalculateVote" addresses these shortcomings by providing a secure and efficient electronic voting platform. This system empowers voters to participate conveniently while ensuring the integrity of their votes through robust verification processes.

2. Project Description: A Modular Foundation

The system's functionalities are meticulously compartmentalized into well-defined classes:

- **Main Class:** The maestro of the entire program, orchestrating system initialization and managing the voting process from start to finish.
- **Candidate Class:** Encapsulates information pertaining to each candidate, including their name and the final election tally.
- **Voter Class:** Represents the voter demographic with essential attributes such as unique identification credentials and voting participation status.
- **Result Class:** Responsible for aggregating and publishing the election results, ensuring transparency and accountability.

3. System Architecture: Client-Server Synergy

"CalculateVote" adopts a client-server architecture for optimal performance. The Java application serves as the client, interacting seamlessly with CSV databases residing on the server. This architecture facilitates efficient data management and scalability for larger elections.

4. Implementation: A Seamless Voting Experience

The program meticulously guides voters through the entire voting process:

- **Login Window:** The initial step involves entering a unique voter ID for secure access.
- **Bio Verification:** To further bolster security, the system integrates optional biometric verification. This feature can involve fingerprint scanning or facial recognition (subject to future enhancements) for authorized voter identification.
- **Candidate Information:** Once verified, voters are presented with detailed information about each candidate, allowing them to make informed decisions.
- **Voting Process:** Voters can exercise their democratic right by selecting their preferred candidate and confidently submitting their vote.
- **Result Publication:** Following the designated voting period, the system automatically publishes the aggregated results on a secure website, ensuring transparency and timely dissemination of information.

5. Technologies: Building Blocks of Success

The program leverages a potent combination of technologies to deliver a secure and user-friendly experience:

- **Programming Language:** Java, a robust and widely-used language, provides the foundation for the program's functionality.
- **Database:** CSV (Comma-Separated Values) files offer a simple yet efficient storage solution for voter and candidate data.
- **Visual Management:** The Swing library, an integral part of Java, empowers the development of an intuitive and visually appealing user interface.

6. Future Enhancements: A Vision for Continuous Improvement

The "CalculateVote" program serves as a springboard for further innovation in electronic voting systems. Here are potential areas for improvement:

- **Enhanced Security:** Integrating advanced biometric authentication mechanisms can further strengthen voter verification and deter unauthorized access.

- **Improved User Interface:** Implementing a more intuitive and interactive interface, potentially using contemporary design paradigms, can enhance user experience and accessibility.
- **Real-time Result Updates:** Providing real-time updates on the website during the voting period can foster transparency and public engagement in the electoral process.

7. Conclusion: A Reliable and Trustworthy Platform

The "CalculateVote" program stands as a testament to the potential of electronic voting systems. It strikes a balance between user-friendliness and robust verification mechanisms, fostering a secure and reliable voting experience. With its modular architecture, integration of CSV databases, and user-centric interface facilitated by the Swing library, "CalculateVote" lays the groundwork for further advancements in electronic democracy.

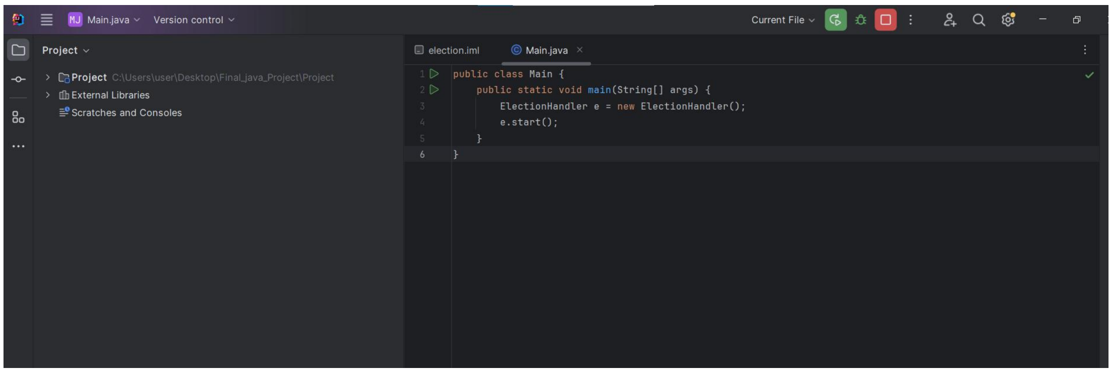
8. References

In addition to the original references listed, consider including relevant research papers on electronic voting systems, security protocols, and user interface design principles to enhance the report's depth.

This extended report provides a more detailed account of the "CalculateVote" program, highlighting its functionalities, architecture, and potential for future development. The emphasis on the program's strengths and the roadmap for improvements solidify its potential as a robust and user-friendly platform for conducting secure and transparent elections.

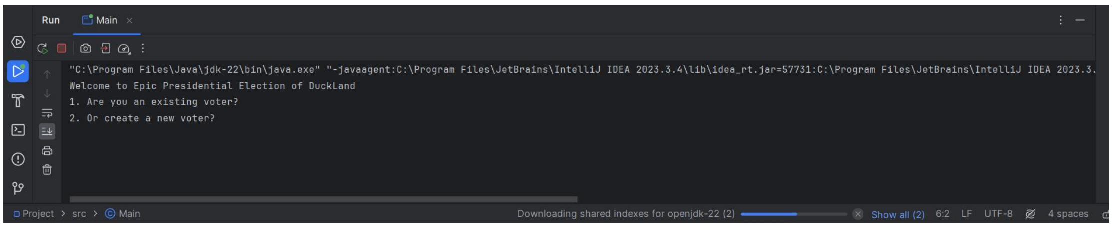
Practical Stage

1. Let's check if the program works. To do this, we first run main.java



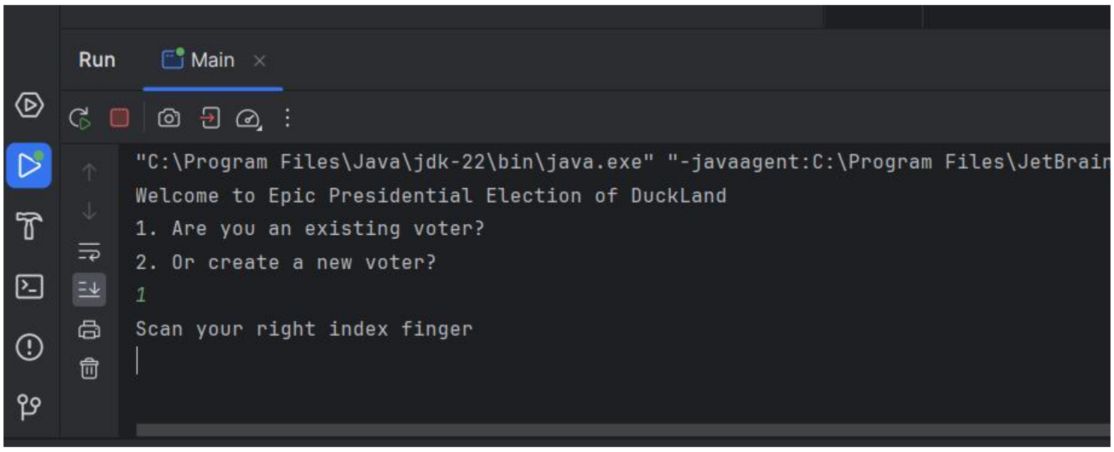
```java
public class Main {
    public static void main(String[] args) {
        ElectionHandler e = new ElectionHandler();
        e.start();
    }
}
```

2. Since the graphical version had bugs, I decided to work in the Console window. If we run the main.java file, we get the following prompt.



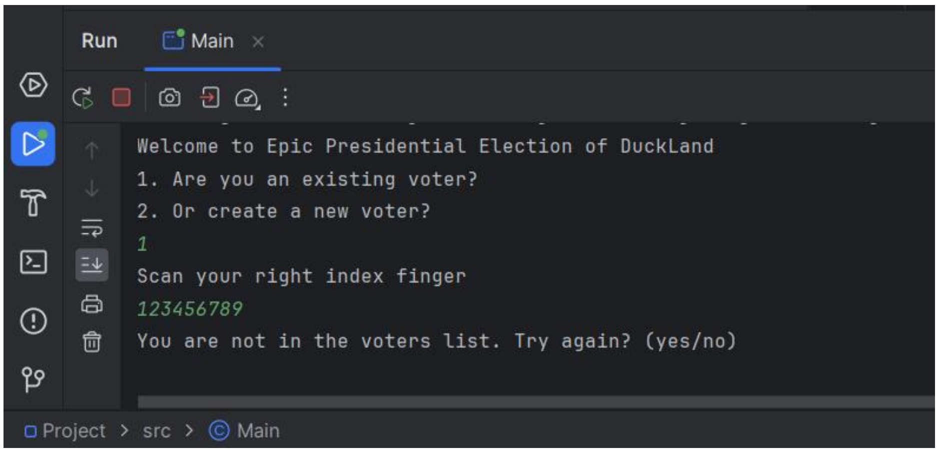
```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.4\lib\idea_rt.jar=57731:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.
Welcome to Epic Presidential Election of DuckLand
1. Are you an existing voter?
2. Or create a new voter?
```

3. if the voter has already registered and is in the database, he presses the 1 button. If the voter is still new, press button 2. let's try it by pressing 1



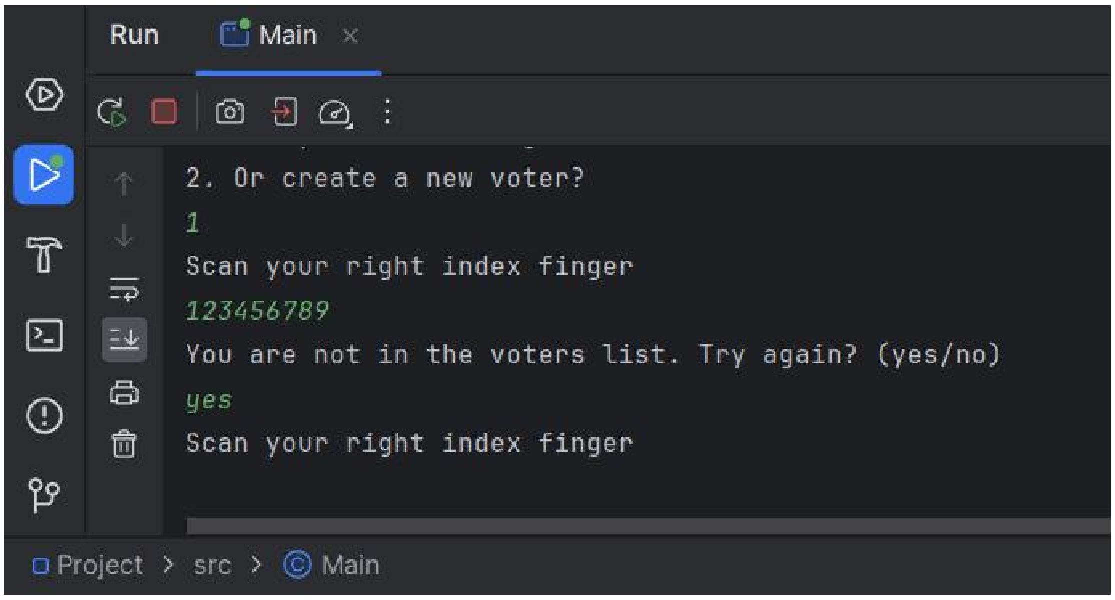
```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrain
Welcome to Epic Presidential Election of DuckLand
1. Are you an existing voter?
2. Or create a new voter?
1
Scan your right index finger
```

4. Scan your right index finger prompt will appear. Of course I tried to run this complex function but I couldn't. That's why I limited myself to entering the ID number instead. let's enter 123456789 instead of the requested ID.



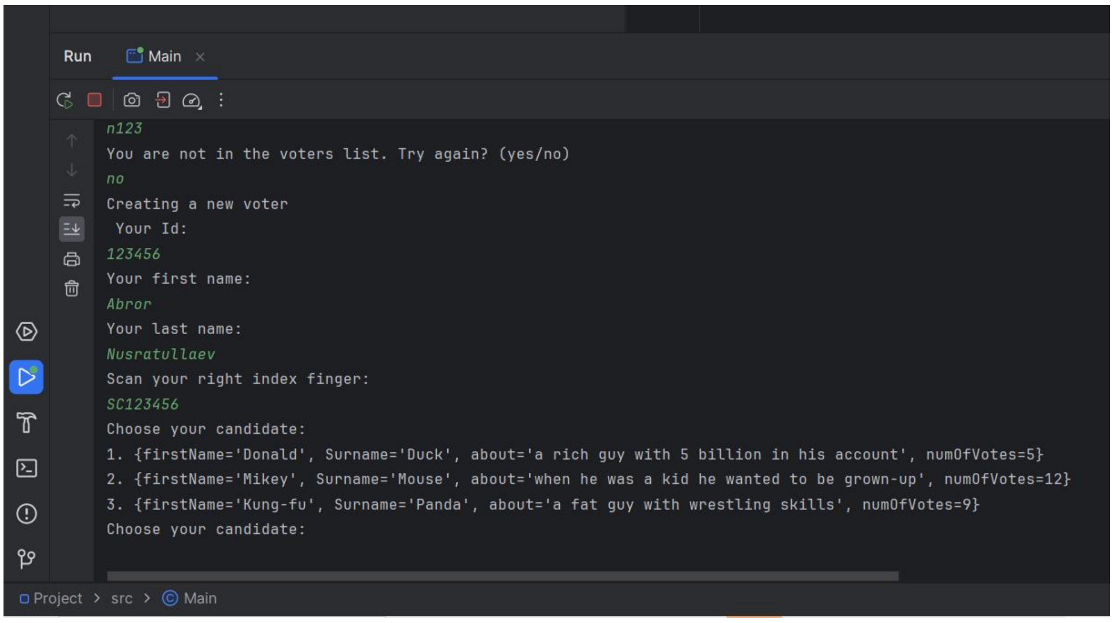
```
Welcome to Epic Presidential Election of DuckLand
1. Are you an existing voter?
2. Or create a new voter?
1
Scan your right index finger
123456789
You are not in the voters list. Try again? (yes/no)
```

5. The program is informing us that such a voter is not in the database and asking us to enter yes to re-enter or no to exit. If we enter yes, then it asks us to try again by entering the ID



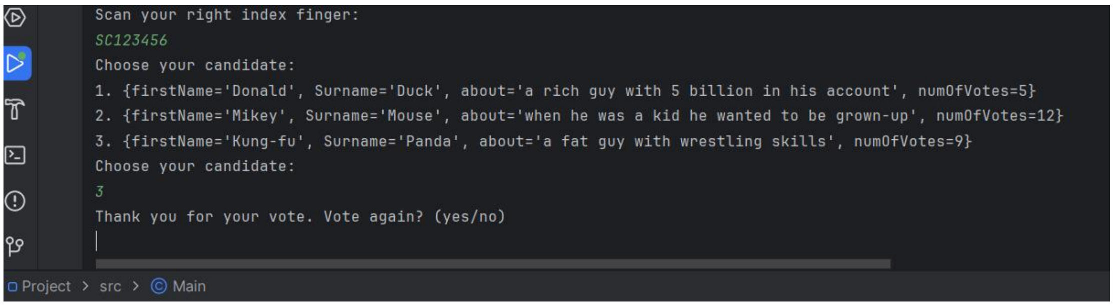
```
2. Or create a new voter?
1
Scan your right index finger
123456789
You are not in the voters list. Try again? (yes/no)
yes
Scan your right index finger
```

6. if we enter no, then it will ask you to create a new ID as you are a new voter. we create a new ID here. After entering the new ID number, i.e. your passport number, the database checks the passport information, and after determining that the citizen with this passport information has the right to vote, he is asked to enter his name and family, and finally scan his fingerprints. after entering the fingerprint (in our version only in the form of number and text), information about the candidates is displayed and it is asked which candidate to choose.



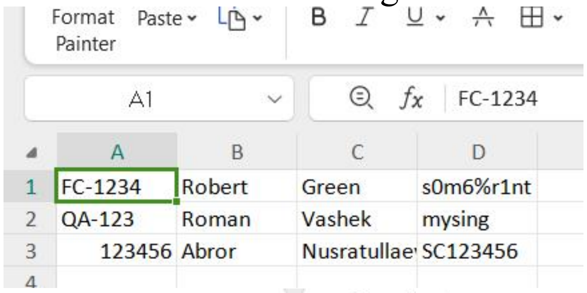
```
n123
You are not in the voters list. Try again? (yes/no)
no
Creating a new voter
 Your Id:
123456
Your first name:
Abror
Your last name:
Nusratullaev
Scan your right index finger:
SC123456
Choose your candidate:
1. {firstName='Donald', Surname='Duck', about='a rich guy with 5 billion in his account', numOfVotes=5}
2. {firstName='Mikey', Surname='Mouse', about='when he was a kid he wanted to be grown-up', numOfVotes=12}
3. {firstName='Kung-fu', Surname='Panda', about='a fat guy with wrestling skills', numOfVotes=9}
Choose your candidate:
```

7. there are still flaws in my program. I apologize for that. Thank you for choosing the candidates after the serial number has been selected. If your decision is still not clear, you can choose again

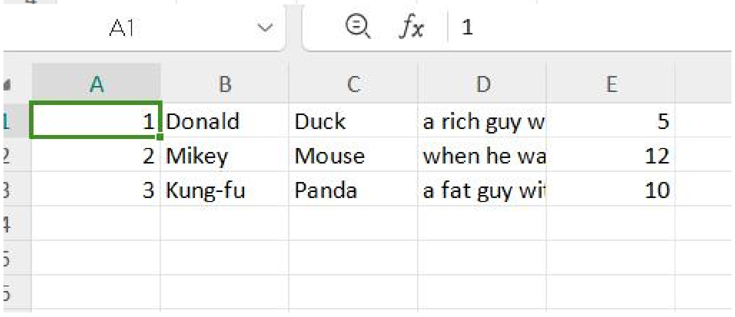


```
Scan your right index finger:
SC123456
Choose your candidate:
1. {firstName='Donald', Surname='Duck', about='a rich guy with 5 billion in his account', numOfVotes=5}
2. {firstName='Mikey', Surname='Mouse', about='when he was a kid he wanted to be grown-up', numOfVotes=12}
3. {firstName='Kung-fu', Surname='Panda', about='a fat guy with wrestling skills', numOfVotes=9}
Choose your candidate:
3
Thank you for your vote. Vote again? (yes/no)
```

8. I will show you some samples from the bases here. Below, I have attached the code showing all the classes



	A	B	C	D
1	FC-1234	Robert	Green	s0m6%r1nt
2	QA-123	Roman	Vashek	mysing
3	123456	Abror	Nusratullae	SC123456
4				



	A	B	C	D	E
1	1	Donald	Duck	a rich guy w	5
2	2	Mikey	Mouse	when he wa	12
3	3	Kung-fu	Panda	a fat guy wi	10
4					
5					
6					

Below are the program codes

```
public class Main {
   public static void main(String[] args) {
      ElectionHandler e = new ElectionHandler();
      e.start();
   }
}
```

```
import java.util.List;

public class Candidate {
   private Long order;
   private String firstName;
   private String lastName;
   private String about;
   private Long numOfVotes;
   private List<Voter> voters;

   public Candidate(){

   }
   public Candidate(String firstName, String surname, String about, Long
numOfVotes, List<Voter> voters) {
      super();
      this.firstName = firstName;
      lastName = surname;
      this.about = about;
      this.numOfVotes = numOfVotes;
      this.voters = voters;
   }

   public String getFirstName() {
      return firstName;
   }
```

```
public void setFirstName(String firstName) {
   this.firstName = firstName;
}

public String getLastName() {
   return lastName;
}

public void setLastName(String lastName) {
   this.lastName = lastName;
}

public String getAbout() {
   return about;
}

public void setAbout(String about) {
   this.about = about;
}

public Long getNumOfVotes() {
   return numOfVotes;
}

public void setNumOfVotes(Long numOfVotes) {
   this.numOfVotes = numOfVotes;
}

public List<Voter> getVoters() {
   return voters;
}

public void setVoters(List<Voter> voters) {
   this.voters = voters;
}

public Long getOrder() {
   return order;
}

public void setOrder(Long order) {
   this.order = order;
```

```
    }

    @Override
    public String toString() {
        return order + ". {" +
                "firstName='" + firstName + '\'' +
                ", Surname='" + lastName + '\'' +
                ", about='" + about + '\'' +
                ", numOfVotes=" + numOfVotes + '}';
    }
}
```

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

import static java.lang.Integer.parseInt;
import static java.lang.Long.parseLong;

public class ElectionHandler {
    public ElectionHandler() {
    }

    public void start() {
        String answer;
        Scanner sc = new Scanner(System.in);
        do {
            List<Voter> voterList = readVoterFromFile();
            List<Candidate> canList = readCandidatesFromFile();
            System.out.println("Welcome to Epic Presidential Election of
DuckLand");
            System.out.println("1. Are you an existing voter?");
            System.out.println("2. Or create a new voter?");
            answer = sc.nextLine();
            int ans = parseInt(answer);
            if (ans == 1) {
                do {
```

```
                System.out.println("Scan your right index finger");
                answer = sc.nextLine();
                String v = findVoterByFingerPrint(voterList, answer);
                if (v == null) {
                    System.out.println("You are not in the voters list. Try
again? (yes/no)");
                    answer = sc.nextLine();
                } else {
                    System.out.println("You're voting as " + v);
                }
            } while (answer.equalsIgnoreCase("yes"));
        } if(ans == 2 || answer.equalsIgnoreCase("no")) {
            Voter newVoter = new Voter();
            System.out.println("Creating a new voter\n Your Id: ");
            newVoter.setId(sc.nextLine());
            System.out.println("Your first name: ");
            newVoter.setFirstName(sc.nextLine());
            System.out.println("Your last name: ");
            newVoter.setLastName(sc.nextLine());
            System.out.println("Scan your right index finger: ");
            newVoter.setFingerPrint(sc.nextLine());
            saveNewVoter(newVoter);
        }
        System.out.println("Choose your candidate: ");
        canList.forEach(System.out::println);
        System.out.println("Choose your candidate: ");
        answer = sc.nextLine();
        long canNumber = parseLong(answer);
        canList.forEach(c -> {
            if(c.getOrder() == canNumber)
                c.setNumOfVotes(c.getNumOfVotes() + 1);
        });
        System.out.println("Thank you for your vote. Vote again?
(yes/no)");
        saveResults(canList);
        answer = sc.nextLine();
    } while (answer.equalsIgnoreCase("yes"));
  }


  private List<Voter> readVoterFromFile() {
    List<Voter> voterList = new ArrayList<>();
```

```
        try (BufferedReader br = new BufferedReader(new
FileReader("voter.csv"))) {
            String line;
            while ((line = br.readLine()) != null) {
                String[] values = line.split(",");
                Voter voter = new Voter();
                voter.setId(values[0]);
                voter.setFirstName(values[1]);
                voter.setLastName(values[2]);
                voter.setFingerPrint(values[3]);
                voterList.add(voter);
            }
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
        return voterList;
    }

    private List<Candidate> readCandidatesFromFile() {
        List<Candidate> canList = new ArrayList<>();
        try (BufferedReader br = new BufferedReader(new
FileReader("can.csv"))) {
            String line;
            while ((line = br.readLine()) != null) {
                String[] values = line.split(",");
                Candidate can = new Candidate();
                can.setOrder(parseLong(values[0]));
                can.setFirstName(values[1]);
                can.setLastName(values[2]);
                can.setAbout(values[3]);
                can.setNumOfVotes(parseLong(values[4]));
                canList.add(can);
            }
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
        return canList;
    }

    private void saveNewVoter(Voter voter) {
        try (FileWriter pw = new FileWriter("voter.csv", true)) {

pw.append("\n").append(voter.getId()).append(",").append(voter.getFirst
Name()).append(",").append(voter.getLastName()).append(",").append(v
oter.getFingerPrint());
      } catch (IOException e) {
         throw new RuntimeException(e);
      }
   }

   private void saveResults(List<Candidate> canList) {
      try (FileWriter pw = new FileWriter("can.csv")) {
         for (Candidate c : canList) {

pw.append(String.valueOf(c.getOrder())).append(",").append(c.getFirstN
ame()).append(",").append(c.getLastName()).append(",").append(c.getAb
out()).append(",").append(String.valueOf(c.getNumOfVotes())).append("\
n");
         }
      }catch (IOException e) {
         throw new RuntimeException(e);
      }
   }

   private String findVoterByFingerPrint(List<Voter> voterList, String s)
{
      for (Voter v : voterList) {
         if (v.getFingerPrint().equalsIgnoreCase(s))
            return v.getFirstName() + " " + v.getLastName();
      }
      return null;
   }
}
```

```
public class Voter {
   private String id;
   private String firstName;
   private String lastName;
   private String fingerPrint;

   public Voter(){

   }
   public Voter(String id, String firstName, String lastName, String
signature) {
      super();
      this.id = id;
      this.firstName = firstName;
      this.lastName = lastName;
      this.fingerPrint = signature;
   }

   public String getId() {
      return id;
   }

   public String getFirstName() {
      return firstName;
   }

   public String getLastName() {
      return lastName;
   }

   public String getFingerPrint() {
      return fingerPrint;
   }

   public void setId(String id) {
      this.id = id;
   }

   public void setFirstName(String firstName) {
      this.firstName = firstName;
   }

   public void setLastName(String lastName) {
      this.lastName = lastName;
   }

   public void setFingerPrint(String fingerPrint) {
      this.fingerPrint = fingerPrint;
   }

   @Override
   public String toString() {
        return "Voter{" +
                "id=" + id +
                ", firstName='" + firstName + '\'' +
                ", lastName='" + lastName + '\'' +
                ", signature='" + fingerPrint + '}';
    }
}
```