```
import java.util.*;

class Person {
    protected String name;

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

    public String getName() {
        return name;
    }
}

class Voter extends Person {
    private int voterId;
    private boolean hasVoted;

    public Voter(int voterId, String name) {
        super(name);
        this.voterId = voterId;
        this.hasVoted = false;
    }

    public int getVoterId() {
        return voterId;
    }

    public boolean hasVoted() {
        return hasVoted;
    }

    public void setVoted(boolean voted) {
        this.hasVoted = voted;
    }
}

class Candidate extends Person {
    private int candidateId;
    private int votes;

    public Candidate(int candidateId, String name) {
        super(name);
        this.candidateId = candidateId;
        this.votes = 0;
    }

    public int getCandidateId() {
        return candidateId;
    }

    public void addVote() {
        votes++;
    }

    public int getVotes() {
        return votes;
    }
}

class Election {
    private List<Voter> voters = new ArrayList<>();
    private List<Candidate> candidates = new ArrayList<>();

    public void addVoter(Voter voter) {
        voters.add(voter);
    }

    public void addCandidate(Candidate candidate) {
        candidates.add(candidate);
    }

    public Voter loginVoter(int voterId) {
        for (Voter v : voters) {
            if (v.getVoterId() == voterId) {
                return v;
            }
        }
        return null;
    }

    public void showCandidates() {
        System.out.println("\nCandidates:");
        for (Candidate c : candidates) {
            System.out.printf("%d. %s\n", c.getCandidateId(), c.getName());
        }
    }
    public void castVote(Voter voter, int candidateId) {
        if (voter.hasVoted()) {
            System.out.println("You have already voted!");
            return;
        }

        for (Candidate c : candidates) {
            if (c.getCandidateId() == candidateId) {
                c.addVote();
                voter.setVoted(true);
                System.out.println("Vote casted successfully!");
                return;
            }
        }
        System.out.println("Invalid Candidate ID!");
    }

    public void showResults() {
        System.out.println("\n----- ELECTION RESULTS -----");
        Candidate winner = null;
        int maxVotes = 0;
        boolean tie = false;

        for (Candidate c : candidates) {
            System.out.printf("%d. %s - Votes: %d\n", c.getCandidateId(), c.getName(),
c.getVotes());

            if (winner == null || c.getVotes() > maxVotes) {
                winner = c;
                maxVotes = c.getVotes();
                tie = false;
            } else if (c.getVotes() == maxVotes) {
                tie = true;
            }
        }
        if (winner != null) {
            if (tie && maxVotes > 0) {
                System.out.println("\nIt's a tie!");
            } else if (maxVotes == 0) {
                System.out.println("\nNo votes have been cast yet.");
            } else {
                System.out.printf("\nWinner: %s 🎉\n", winner.getName());
            }
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Election election = new Election();

        election.addCandidate(new Candidate(1, "Candidate A"));
        election.addCandidate(new Candidate(2, "Candidate B"));
        election.addCandidate(new Candidate(3, "Candidate C"));

        election.addVoter(new Voter(101, "Pravin"));
        election.addVoter(new Voter(102, "Ravi"));
        election.addVoter(new Voter(103, "Meena"));

        Scanner sc = new Scanner(System.in);

        String cont;
        do {
            System.out.print("Enter Voter ID to Login: ");
            int voterId = sc.nextInt();

            Voter voter = election.loginVoter(voterId);

            if (voter == null) {
                System.out.println("Invalid Voter ID! Access Denied.");
            } else {
                System.out.println("Welcome " + voter.getName() + "!");
                election.showCandidates();

                System.out.print("Enter Candidate ID to Vote: ");
                int candidateId = sc.nextInt();

                election.castVote(voter, candidateId);
            }

            System.out.print("\nContinue voting? (yes/no): ");
            cont = sc.next().trim().toLowerCase();
        } while (cont.equals("yes"));

        election.showResults();
        sc.close();
    }
}
```