```java
package model;

import data.Database;

import java.util.List;
import java.util.Scanner;

/**
 * User class
 */
public class User {
    private String name, streetname, dob, telephone, cpr, houseNumber, role;
    private String username, password;
    private int id, postcode;
    boolean err = true;
    char dash;
    private static int counter;

    Scanner input = new Scanner(System.in);

    /**
     * Database Constructor
     *
     * @param newId ID of the User
     * @param newRole Role of the User (administrator, customer, car owner)
     * @param newName Full name of the user
     * @param newStreetname Street name of the user's home address
     * @param newHouseNumber House number of the user's home address
     * @param newPostcode Post code of the user's home address
     * @param newDob Date of Birth of the user
     * @param newTelephone Telephone number of the user
     * @param newCpr CPR of the user
     * @param newUsername user name of the user
     * @param newPassword password of the user
     */
    public User(int newId, String newRole, String newName, String
        newStreetname, String newHouseNumber, int newPostcode,
            String newDob, String newTelephone, String newCpr, String
                newUsername, String newPassword) {
        id          = newId;
        role        = newRole;
        name        = newName;
        streetname  = newStreetname;
        houseNumber     = newHouseNumber;
        postcode    = newPostcode;
        dob             = newDob;
        telephone   = newTelephone;
        cpr             = newCpr;
        username    = newUsername;
        password    = newPassword;
        counter++;
    }

    /**
     * In-App Constructor
     *
     * @param newRole Role of the User (administrator, customer, car owner)
     * @param newName Full name of the user
     * @param newStreetname Street name of the user's home address
     * @param newHouseNumber House number of the user's home address
```

```
         * @param newPostcode Post code of the user's home address
         * @param newDob Date of Birth of the user
         * @param newTelephone Telephone number of the user
         * @param newCpr CPR of the user
         */
        public User (String newRole, String newFirstname, String newLastname,
            String newStreetname, String newHouseNumber, int newPostcode,
                String newDob, String newTelephone, String newCpr) {
            id          = ++counter;
            role        = newRole;
            name        = newFirstname+" "+newLastname;
            streetname  = newStreetname;
            houseNumber     = newHouseNumber;
            postcode    = newPostcode;
            dob             = newDob;
            telephone   = newTelephone;
            cpr             = newCpr;
            username    = makeUsername(newFirstname, newLastname);
            password    = makePassword(newFirstname, newLastname);
        }

        /**
         * Writes a user object (line) to the databse
         */
        public void toDB() {
            Database.write("users", this.id+";"+this.role+";"+this.name+";"+this
                .streetname+";"+this.houseNumber+";"+this.postcode+";"+

                        this.dob+";"+this.telephone+";"+this.cpr+";"+this.
                        username+";"+this.password+" ");
        }

        // Start getter
        public int getId() {
            return id;
        }

        public String getRole() {
            return role;
        }

        public String getName() {
            return name;
        }

        public String getUsername() {
            return username;
        }

        public String getPassword() {
            return password;
        }

        public int getID() {
            return id;
        }

        public String getAddress() {
            return streetname + " " + houseNumber + " " + postcode;
        }
```

```
    public String getDOB() {
        return dob;
    }

    public String getTelephone() {
        return username;
    }

    public String getCPR() {
        return cpr;
    }
    // End getter

    // Start setters
    public void setStreetname(String newStreetname) {
        streetname = newStreetname;
    }

    public void setHouseNumber(String newHouseNumber) {
        houseNumber = newHouseNumber;
    }

    public void setPostcode(int newPostcode) {
        postcode = newPostcode;
    }

    public void setTelephone(String newTelephone) {
        telephone = newTelephone;
    }

    public void newPassword(String newPassword) {
        password = newPassword;
    }
    // End setters

    /**
     * Validate credentials method
     *
     * @param testUsername User name that will be tested against this
         instance's password
     * @param testPassword Password that will be tested against this
         instance's password
     * @return
     */
    public Boolean checkCredentials(String testUsername, String testPassword
        ) {
        if ( username.equals(testUsername) && password.equals(testPassword)
            ) {
            return true;
        } else {
            return false;
        }
    }

    /**
     * Create user name from first letter of the first name and the first
         three letters from
     * the last name.
     *
     * Critical Problem: Users with similar names might get the same user
         name.
     *
     * To Do: Make recursive unique-validator
     *
     * @param firstname User's first name
     * @param lastname User's last name
     * @return User name
     */
    private String makeUsername(String firstname, String lastname) {
        String username = firstname.substring(0, 1).toLowerCase() + lastname
            .substring(0, 3).toLowerCase();
        System.out.printf("Your username is %s\n", username);
        return username;
    }

    /**
     * Create password from the first three letters of the last name and the
         last four numbers
     * of the CPR.
     *
     * Critical Problem: Knowing the last name and the CPR of a user enables
         hackers to get
     * access to foreign accounts.
     *
     * To Do: Make password less predictable (safer)
     *
     * @param firstname User's first name
     * @param lastname User's last name
     * @return String password
     */
    private String makePassword(String firstname, String lastname) {
        String password = lastname.substring(0, 3).toLowerCase() + cpr.
            substring(cpr.length() - 4).toLowerCase();
        System.out.printf("Your password is %s.\n", password);
        return password;
    }

    // Start print methods
    public static String printTableHeader() {
        return String.format("\t  | %-2s | %-15s | %-20s | %-11s | %-8s |\n"
            ,
                "ID", "Role", "Name", "CPR", "Username");
    }

    public static String printLine(User user) {
        return String.format("\t  | %-2s | %-15s | %-20s | %-11s | %-8s |\n"
            ,
                user.getId(), user.getRole(), user.getName(), user.getCPR(),
                    user.getUsername());
    }

    public static String toString(List<User> users) {
        String out = String.format("\n\n\tUSERS:\n");
        out += printTableHeader();
        for ( User user : users ) {
            out += printLine(user);
        }
        return out;
    }
    public static String toString(User user) {
        String out = String.format("\n\n\tACCOUNT:\n");
        out += printTableHeader();
        out += printLine(user);
        return out;
    }

    public String toString() {
        String out = String.format("\n\n\tUSER:\n");
        out += printTableHeader();
        out += printLine(this);
        return out;
    }

    public static String report() {
        return toString(Database.getUsers());
    }
    // End print methods
}
```