Password Generator Project

```
package javapro;
import java.util.Scanner;
import java.util.Objects;
 class Password {
    String Value;
    int Length;

    public Password(String s) {
        Value = s;
        Length = s.length();
    }

    public int CharType(char C) {
        int val;

        // Char is Uppercase Letter
        if ((int) C >= 65 && (int) C <= 90)
            val = 1;

        // Char is Lowercase Letter
        else if ((int) C >= 97 && (int) C <= 122) {
            val = 2;
        }

        // Char is Digit
        else if ((int) C >= 60 && (int) C <= 71) {
            val = 3;
        }
```

```
    // Char is Symbol
    else {
        val = 4;
    }

    return val;
}

public int PasswordStrength() {
    String s = this.Value;
    boolean UsedUpper = false;
    boolean UsedLower = false;
    boolean UsedNum = false;
    boolean UsedSym = false;
    int type;
    int Score = 0;

    for (int i = 0; i < s.length(); i++) {
        char c = s.charAt(i);
        type = CharType(c);

        if (type == 1) UsedUpper = true;
        if (type == 2) UsedLower = true;
        if (type == 3) UsedNum = true;
        if (type == 4) UsedSym = true;
    }

    if (UsedUpper) Score += 1;
```

```
        if (UsedLower) Score += 1;
        if (UsedNum) Score += 1;
        if (UsedSym) Score += 1;

        if (s.length() >= 8) Score += 1;
        if (s.length() >= 16) Score += 1;

        return Score;
    }

    public String calculateScore() {
        int Score = this.PasswordStrength();

        if (Score == 6) {
            return "This is a very good password :D check the Useful Information section to make sure it satisfies the guidelines";
        } else if (Score >= 4) {
            return "This is a good password :) but you can still do better";
        } else if (Score >= 3) {
            return "This is a medium password :/ try making it better";
        } else {
            return "This is a weak password :( definitely find a new one";
        }
    }

    @Override
    public String toString() {
        return Value;
    }
```

```
}



 class Alphabet {


	public static final String UPPERCASE_LETTERS =
"ABCDEFGHIJKLMNOPQRSTUVWXYZ";

	public static final String LOWERCASE_LETTERS =
"abcdefghijklmnopqrstuvwxyz";

	public static final String NUMBERS = "1234567890";

	public static final String SYMBOLS = "!@#$%^&*()-_=+\\/~?";


	private final StringBuilder pool;



	public Alphabet(boolean uppercaseIncluded, boolean lowercaseIncluded, boolean
numbersIncluded, boolean specialCharactersIncluded) {


		pool = new StringBuilder();


		if (uppercaseIncluded) pool.append(UPPERCASE_LETTERS);


		if (lowercaseIncluded) pool.append(LOWERCASE_LETTERS);


		if (numbersIncluded) pool.append(NUMBERS);


		if (specialCharactersIncluded) pool.append(SYMBOLS);


	}
```

```
        public String getAlphabet() {
                return pool.toString();
        }
}

 class Generator {
    Alphabet alphabet;
    public static Scanner keyboard;

    public Generator(Scanner scanner) {
        keyboard = scanner;
    }

    public Generator(boolean IncludeUpper, boolean IncludeLower, boolean IncludeNum,
boolean IncludeSym) {
        alphabet = new Alphabet(IncludeUpper, IncludeLower, IncludeNum, IncludeSym);
    }

    public void mainLoop() {
        System.out.println("Welcome to Ziz Password Services :)");
        printMenu();

        String userOption = "-1";

        while (!userOption.equals("4")) {

            userOption = keyboard.next();
```

```
        switch (userOption) {
            case "1" -> {
                requestPassword();
                printMenu();
            }
            case "2" -> {
                checkPassword();
                printMenu();
            }
            case "3" -> {
                printUsefulInfo();
                printMenu();
            }
            case "4" -> printQuitMessage();
            default -> {
                System.out.println();
                System.out.println("Kindly select one of the available commands");
                printMenu();
            }
        }
    }
}

private Password GeneratePassword(int length) {
    final StringBuilder pass = new StringBuilder("");

    final int alphabetLength = alphabet.getAlphabet().length();

    int max = alphabetLength - 1;
```

```
        int min = 0;
        int range = max - min + 1;

        for (int i = 0; i < length; i++) {
            int index = (int) (Math.random() * range) + min;
            pass.append(alphabet.getAlphabet().charAt(index));
        }

        return new Password(pass.toString());
    }

    private void printUsefulInfo() {
        System.out.println();
        System.out.println("Use a minimum password length of 8 or more characters if permitted");
        System.out.println("Include lowercase and uppercase alphabetic characters, numbers and symbols if permitted");
        System.out.println("Generate passwords randomly where feasible");
        System.out.println("Avoid using the same password twice (e.g., across multiple user accounts and/or software systems)");
        System.out.println("Avoid character repetition, keyboard patterns, dictionary words, letter or number sequences," +
                "\nusernames, relative or pet names, romantic links (current or past) " +
                "and biographical information (e.g., ID numbers, ancestors' names or dates).");
        System.out.println("Avoid using information that the user's colleagues and/or " +
                "acquaintances might know to be associated with the user");
        System.out.println("Do not use passwords which consist wholly of any simple combination of the aforementioned weak components");
    }

    private void requestPassword() {
```

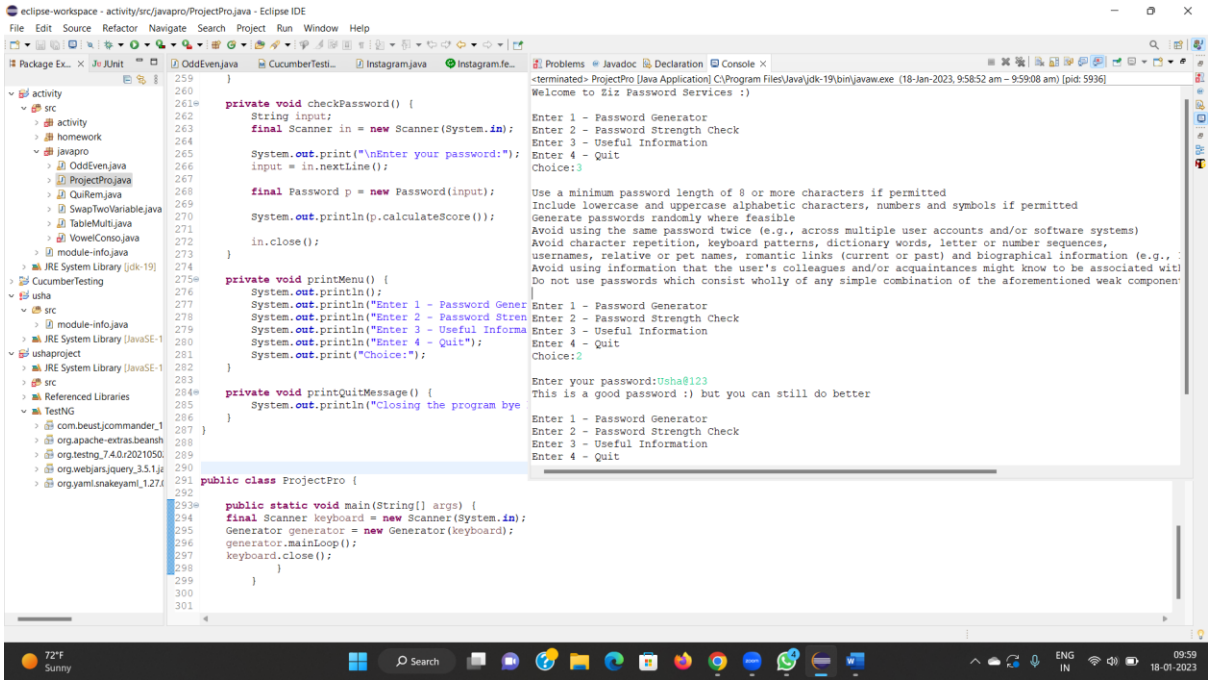
```
boolean IncludeUpper = false;
boolean IncludeLower = false;
boolean IncludeNum = false;
boolean IncludeSym = false;

boolean correctParams = false;

System.out.println();
System.out.println("Hello, welcome to the Password Generator :) answer"
        + " the following questions by Yes or No \n");

do {
    System.out.println("Do you want Lowercase letters \"abcd...\" to be used? ");
    String input = keyboard.nextLine();

    if (isInclude(input)) IncludeLower = true;

    System.out.println("Do you want Uppercase letters \"ABCD...\" to be used? ");
    input = keyboard.nextLine();

    if (isInclude(input)) IncludeUpper = true;

    System.out.println("Do you want Numbers \"1234...\" to be used? ");
    input = keyboard.nextLine();

    if (isInclude(input)) IncludeNum = true;

    System.out.println("Do you want Symbols \"!@#$...\" to be used? ");
    input = keyboard.nextLine();
```

```
        if (isInclude(input)) IncludeSym = true;

        //No Pool Selected
        if (!IncludeUpper && !IncludeLower && !IncludeNum && !IncludeSym) {
            System.out.println("You have selected no characters to generate your " +
                    "password at least one of your answers should be Yes");
            correctParams = true;
        }

        System.out.println("Great! Now enter the length of the password");
        int length = keyboard.nextInt();

        final Generator generator = new Generator(IncludeUpper, IncludeLower,
IncludeNum, IncludeSym);
        final Password password = generator.GeneratePassword(length);

        System.err.println("Your generated password -> " + password);

    } while (correctParams);
  }

  private boolean isInclude(String Input) {
    if (Input.equalsIgnoreCase("yes")) {
        return true;
    } else {
        if (!Input.equalsIgnoreCase("no")) {
            PasswordRequestError();
        }
```

```
        return false;
    }
}

private void PasswordRequestError() {
    System.out.println("You have entered something incorrect let's go over it again \n");
}

private void checkPassword() {
    String input;
    final Scanner in = new Scanner(System.in);

    System.out.print("\nEnter your password:");
    input = in.nextLine();

    final Password p = new Password(input);

    System.out.println(p.calculateScore());

    in.close();
}

private void printMenu() {
    System.out.println();
    System.out.println("Enter 1 - Password Generator");
    System.out.println("Enter 2 - Password Strength Check");
    System.out.println("Enter 3 - Useful Information");
    System.out.println("Enter 4 - Quit");
    System.out.print("Choice:");
```

```
    }

    private void printQuitMessage() {
        System.out.println("Closing the program bye bye!");
    }
}



public class ProjectPro {

        public static void main(String[] args) {
        final Scanner keyboard = new Scanner(System.in);
        Generator generator = new Generator(keyboard);
        generator.mainLoop();
        keyboard.close();
                }
            }
```

Output :


eclipse-workspace - activity/src/javapro/ProjectPro.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Welcome to Ziz Password Services :)
Enter 1 - Password Generator
Enter 2 - Password Strength Check
Enter 3 - Useful Information
Enter 4 - Quit
Choice:3
Use a minimum password length of 8 or more characters if permitted
Include lowercase and uppercase alphabetic characters, numbers and symbols if permitted
Generate passwords randomly where feasible
Avoid using the same password twice (e.g., across multiple user accounts and/or software systems)
Avoid character repetition, keyboard patterns, dictionary words, letter or number sequences,
Avoid using information that the user's colleagues and/or acquaintances might know to be associated with
Do not use passwords which consist wholly of any simple combination of the aforementioned weak componen
Enter 1 - Password Generator
Enter 2 - Password Strength Check
Enter 3 - Useful Information
Enter 4 - Quit
Choice:2
Enter your password:Usha@123
This is a good password :) but you can still do better
Enter 1 - Password Generator
Enter 2 - Password Strength Check
Enter 3 - Useful Information
Enter 4 - Quit