**Virtual Assistant**



A project report submitted to

Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal

towards partial fulfillment of

the degree of

**MASTER OF COMPUTER APPLICATION**

**{2020-21}**

**Submitted by:**

Aman Jain

0801CA191001

**Guided By:**

Mr Upendra Singh

*Department of Information*

*Techmology*

Department of Information Technology

**SHRI G.S. INSTITUTE OF TECHNOLOGY AND SCIENCE**

**INDORE (M.P.)**

**SHRI G.S. INSTITUTE OF TECHNOLOGY AND SCIENCE**

**INDORE (M.P.)**



**Recommendation**

The project report entitled **“*Virtual Assistant*”** submitted by **Aman Jain** students of MCA Second year in the session 2020-21, towards partial fulfillment of the degree of **Master of Computer Applications** of Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal, is a satisfactory account of his work and is recommended for the award of degree**.**

**Upendra Singh**

Project Guide

Department of IT

**Head**

Department of IT

**Dean(Academics)**

**S.G.S.I.T.S.,Indore**

**SHRI G.S. INSTITUTE OF TECHNOLOGY AND SCIENCE**

**INDORE (M.P.)**



**Certificate**

The project report entitled **“*Virtual Assistant*”** submitted by **Aman Jain** students of MCA Second year in the session 2020-21, towards partial fulfillment of the degree of **Master of Computer Applications** of Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal, is a satisfactory account of their work and is approved for award of the degree.

**Internal Examiner External Examiner**

**Date**

**Acknowledgement**

I am heartily pleased to acknowledge all those people who have helped me in the successful completion of this project. With great pleasure i express my heartfelt gratitude to our esteemed guide, **Mr. Upendra Singh** Lecturer Department of information Technology, S.G.S.I.T.S. Indore. His persistent encouragement, perpetual motivation, everlasting patience and valuable technical inputs in discussions have enabled the successful completion of this project. His invaluable help, advice and constant encouragement helped us a lot and provide impetus to the progress of the project. We extend our profound indebtedness to the Head of the department **Ms. Sunita Varma,** the word loose their worth for her invaluable guidance, continuous encouragement and cooperation in every respect.

I sincerely wish to express our gratitude to all the members of staff of M.C.A.who have extended their cooperation at all times and have contributed in their own way in developing the project. Successful completion of a project is not an individual effort. It is an outcome of the cumulative effort of a number of persons, each having his own importance to the objective. We are thankful to our parents for being a constant source of encouragement in all our endeavors. Indeed it is their support that helps us through the ups and downs of life. The support and suggestion of our friends are worth appreciation and thankfulness. *A blend of gratitude, pleasure, great satisfaction and indebtedness is what, we feel to convey to all those who have directly or indirectly contributed to the successful completion of our project work.*

**Aman Jain**

**Abstract**

**Java Virtual Assistant is a software build on java and runs on terminal only.**

**A virtual assistant is a program or a software which provides various services to users and simplify things for them which results in saving much time.**

**There are number of things which this virtual assistant program can do, including:**

* **Open calculator**
* **Open Notepad application**
* **Open Microsoft Paint application**
* **Open file explorer**
* **Shutdown/restart system**
* **Check password strength**
* **Store a memo / diary / note**
* **Show system related information**

***Table of Contents***

**Chapter 1.**

**Introduction …1**

1.1 Objective 1

1.2 Scope 1

1.3 Technologies used 1

**Chapter 2.**

**Analysis …2**

2.1 Data Flow Diagram 2

2.2 Class Diagram 2

2.3 Component Diagram 3

2.4 Use Case Diagrams 4

**Coding 5**

**Output 15**

**References 17**

**Chapter 1**

**Introduction**

* 1. **Objective**

The aim of this project is to implement a software which can reduce human work and save time.

* 1. **Scope**

The software can be used to perform certain tasks like open some applications, check password strength and get system information

The benefits of this software are it is easy to use and there is no 3rd party service involvement makes this software secure and reliable .

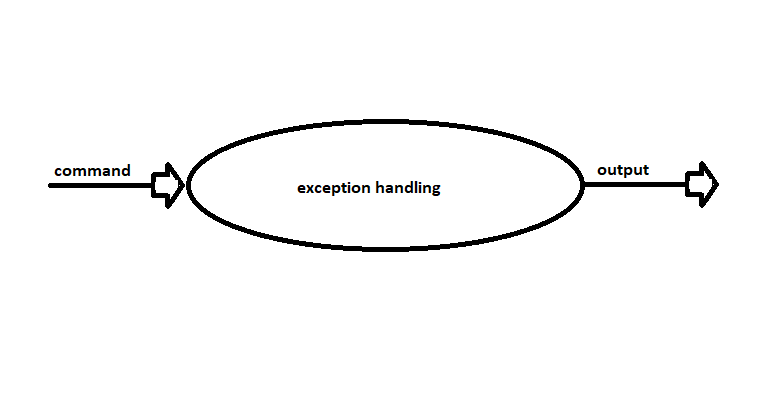
* 1. **Technologies Used**

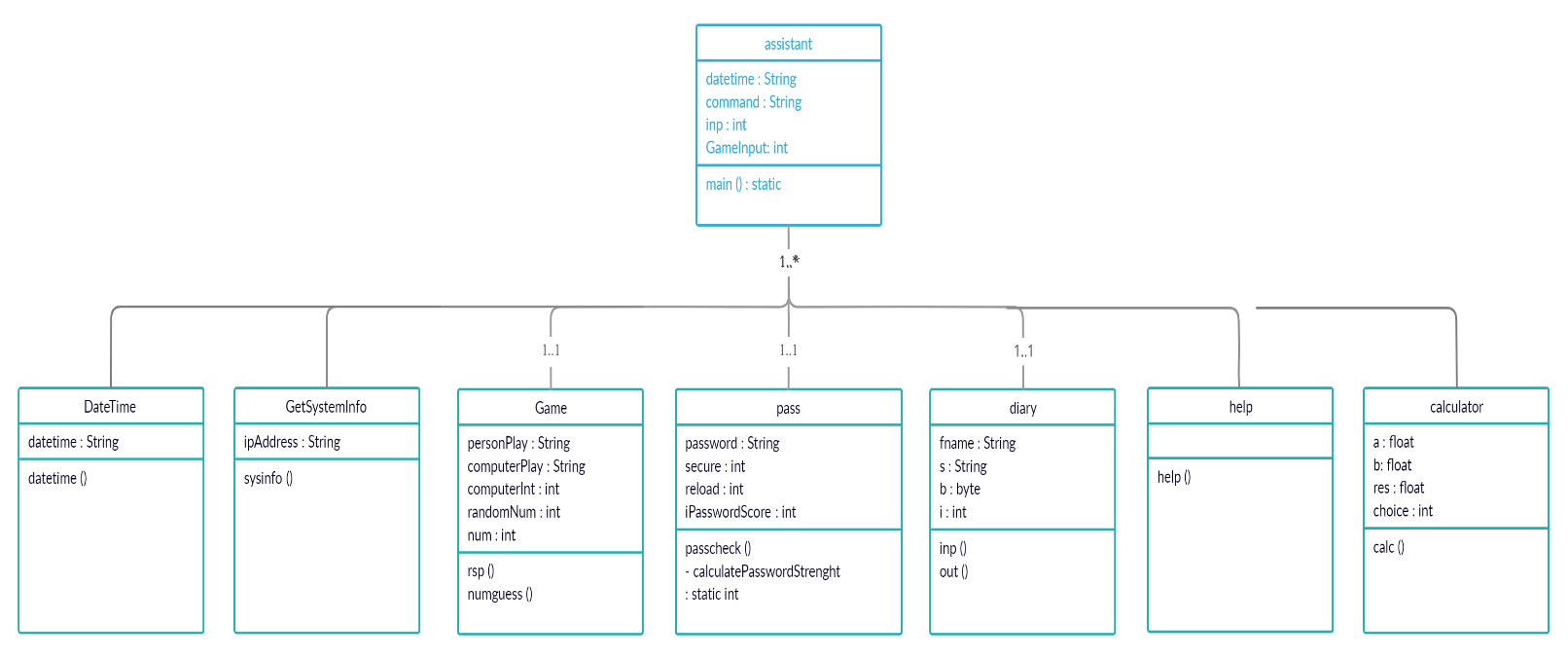
1. Java

**Chapter 2**

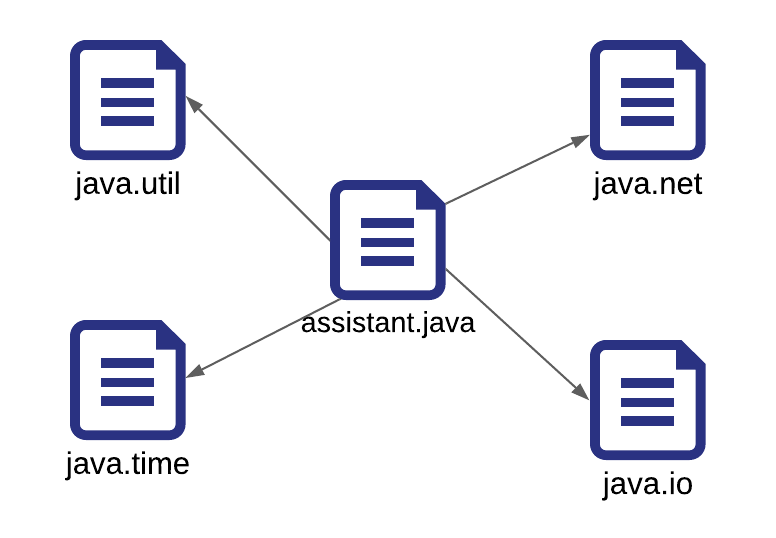
**Analysis**

**DataFlow Diagram**

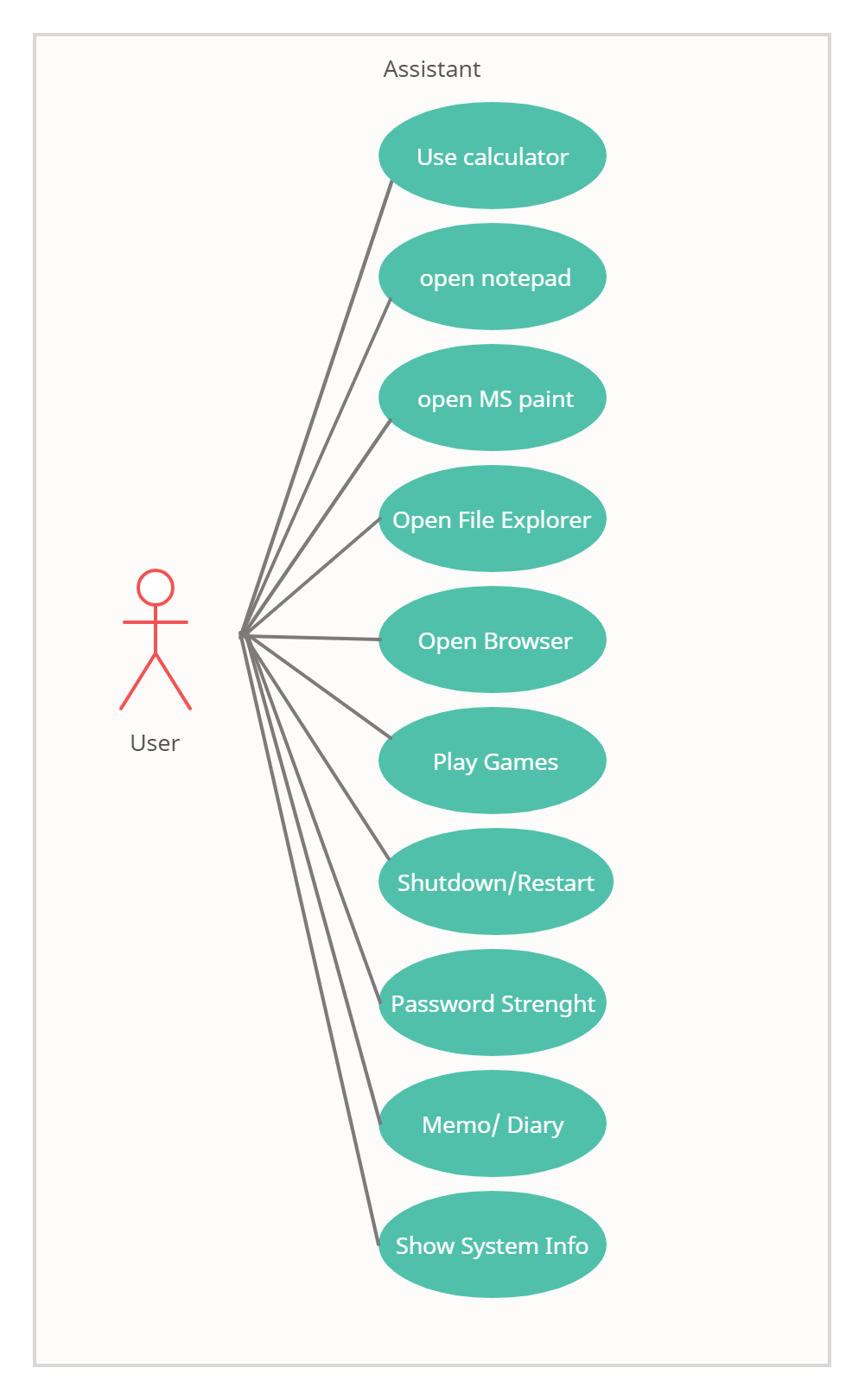
****

**Class Diagram**

**Component Diagram**

****

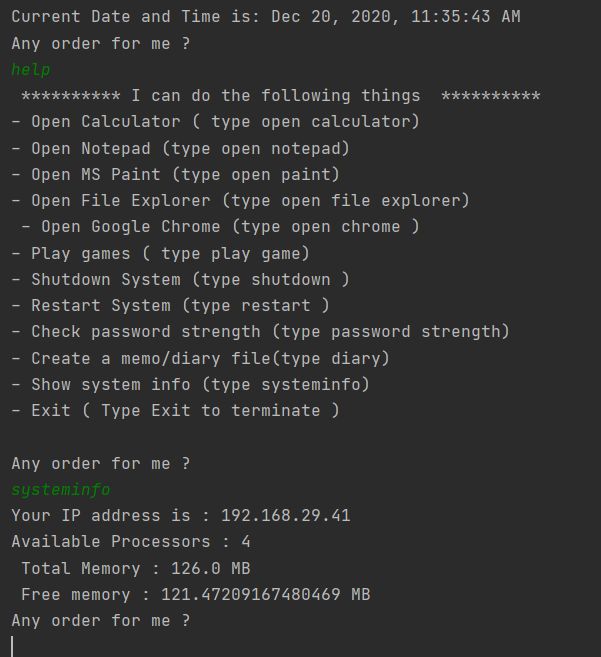
**Use case Diagram**

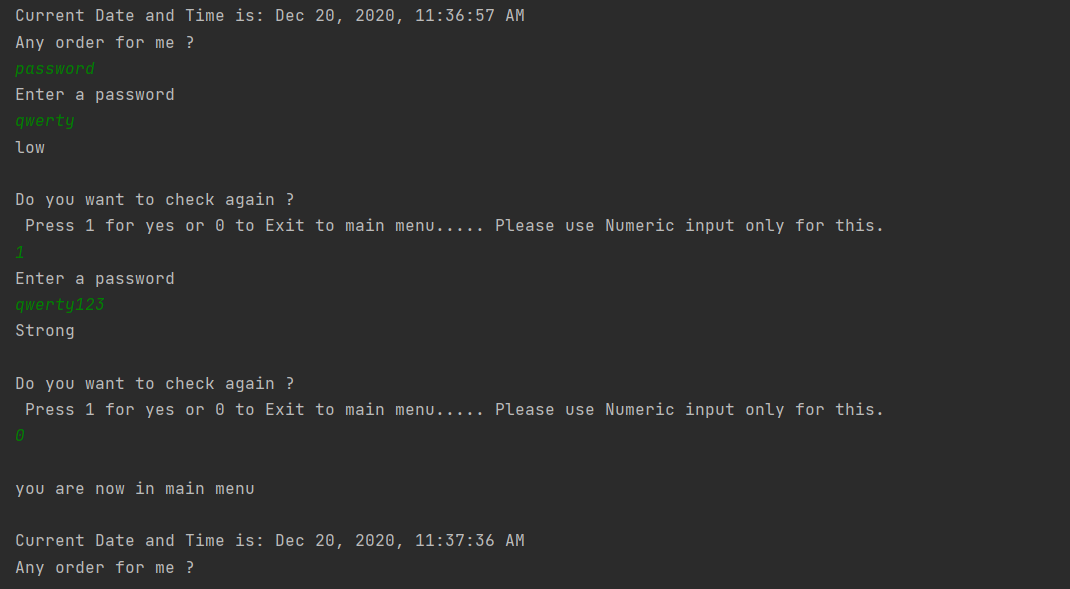
****

**Coding**

import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
import java.time.format.FormatStyle;  
import java.util.Scanner;  
import java.net.InetAddress;  
import java.util.Random;  
import java.io.\*;  
  
public class assistant {  
 static class DateTime {  
 void datetime() {  
 LocalDateTime current = LocalDateTime.*now*();  
  
 DateTimeFormatter formatter = DateTimeFormatter.*ofLocalizedDateTime*(FormatStyle.*MEDIUM*);  
 String datetime = current.format(formatter);  
  
 System.*out*.println("Current Date and Time is: " + datetime);  
 }  
 }  
  
  
  
 static class GetSystemInfo {  
 void sysinfo() {  
 try {  
  
 InetAddress inetAddress = InetAddress.*getLocalHost*();  
  
 String ipAddress = inetAddress.getHostAddress();  
 System.*out*.println("Your IP address is : " + ipAddress);  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 static class Game {  
 void rsp() {  
  
 String personPlay; //User's play -- "R", "P", or "S"  
 String computerPlay = ""; //Computer's play -- "R", "P", or "S"  
 int computerInt; //Randomly generated number used to determine  
 //computer's play  
  
  
 Scanner scan = new Scanner(System.*in*);  
 Random generator = new Random();  
  
 System.*out*.println("Hey, let's play Rock, Paper, Scissors!\n" +  
 "Please enter a move.\n" + "Rock = R, Paper" +  
 "= P, and Scissors = S.");  
  
 System.*out*.println();  
  
 //Generate computer's play (0,1,2)  
 computerInt = generator.nextInt(3) + 1;  
  
 //Translate computer's randomly generated play to  
 //string using if //statements  
  
 if (computerInt == 1)  
 computerPlay = "R";  
 else if (computerInt == 2)  
 computerPlay = "P";  
 else if (computerInt == 3)  
 computerPlay = "S";  
  
 //Get player's play from input-- note that this is  
 // stored as a string  
 System.*out*.println("Enter your play: ");  
 personPlay = scan.next();  
  
 //Make player's play uppercase for ease of comparison  
 personPlay = personPlay.toUpperCase();  
  
 //Print computer's play  
 System.*out*.println("Computer play is: " + computerPlay);  
  
 //See who won. Use nested ifs  
  
 if (personPlay.equals(computerPlay))  
 System.*out*.println("It's a tie!");  
 else if (personPlay.equals("R"))  
 if (computerPlay.equals("S"))  
 System.*out*.println("Rock crushes scissors. You win!!");  
 else if (computerPlay.equals("P"))  
 System.*out*.println("Paper eats rock. You lose!!");  
 else if (personPlay.equals("P"))  
 if (computerPlay.equals("S"))  
 System.*out*.println("Scissor cuts paper. You lose!!");  
 else if (computerPlay.equals("R"))  
 System.*out*.println("Paper eats rock. You win!!");  
 else if (personPlay.equals("S"))  
 if (computerPlay.equals("P"))  
 System.*out*.println("Scissor cuts paper. You win!!");  
 else if (computerPlay.equals("R"))  
 System.*out*.println("Rock breaks scissors. You lose!!");  
 else {  
 System.*out*.println("Invalid user input.");  
 }  
  
 System.*out*.println("\nDo you want to play again ? \n" +  
 " Press 1 for yes or 0 to Exit to main menu....." +  
 " Please use Numeric input only for this. ");  
 int replay = scan.nextInt();  
  
 if (replay == 1)  
 rsp();  
 else {  
 System.*out*.println(" \nyou are now in main menu \n");  
 *main*(null);  
 }  
 }  
  
 void numguess() {  
 Scanner scan = new Scanner(System.*in*);  
 Random generator = new Random();  
 //Generate computer's number (0,1,2,3,4)  
 int randomNum = generator.nextInt(5) + 1;  
 System.*out*.println("enter a number between 1 to 5");  
 int num = scan.nextInt();  
  
 if (num == randomNum) {  
 System.*out*.println(" Hurrey.... Number matched. You win \n");  
 } else {  
 System.*out*.println("Oops, numbers do not matched. You lost \n");  
 System.*out*.println("The number was " + randomNum);  
 }  
 System.*out*.println("\nDo you want to play again ? \n" +  
 " Press 1 for yes or 0 to Exit to main menu....." +  
 " Please use Numeric input only for this. ");  
 int replay = scan.nextInt();  
  
 if (replay == 1)  
 numguess();  
 else {  
 System.*out*.println(" \nyou are now in main menu \n");  
 *main*(null);  
 }  
 }  
 }  
  
 static class pass {  
  
 void passcheck() {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.println("Enter a password");  
 String password = scan.nextLine();  
 int secure = *calculatePasswordStrength*(password);  
 if (secure < 5)  
 System.*out*.println("low");  
 else if (secure > 5 && secure < 8)  
 System.*out*.println("Medium");  
 else  
 System.*out*.println("Strong");  
  
  
 System.*out*.println("\nDo you want to check again ? \n" +  
 " Press 1 for yes or 0 to Exit to main menu....." +  
 " Please use Numeric input only for this. ");  
 int reload = scan.nextInt();  
  
 if (reload == 1)  
  
 passcheck();  
 else {  
 System.*out*.println(" \nyou are now in main menu \n");  
 *main*(null);  
 }  
 }  
  
 private static int calculatePasswordStrength(String password) {  
  
 //total score of password  
 int iPasswordScore = 0;  
  
 if (password.length() < 8)  
 return 0;  
 else if (password.length() >= 10)  
 iPasswordScore += 2;  
 else  
 iPasswordScore += 1;  
  
 //if it contains one digit, add 2 to total score  
 if (password.matches("(?=.\*[0**-**9]).\*"))  
 iPasswordScore += 2;  
  
 //if it contains one lower case letter, add 2 to total score  
 if (password.matches("(?=.\*[a**-**z]).\*"))  
 iPasswordScore += 2;  
  
 //if it contains one upper case letter, add 2 to total score  
 if (password.matches("(?=.\*[A**-**Z]).\*"))  
 iPasswordScore += 2;  
  
 //if it contains one special character, add 2 to total score  
 if (password.matches("(?=.\*[~!@#$%^&\*()\_-]).\*"))  
 iPasswordScore += 2;  
  
 return iPasswordScore;  
  
 }  
 }  
 static class diary{  
 void inp()  
 {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Enter file name with .txt extension");  
 String fname = sc.nextLine();  
 try  
 {  
 FileOutputStream fout=new FileOutputStream(fname);  
 System.*out*.println("write here (please write in single line only)........... \n");  
 String s= sc.nextLine();  
 byte b[]=s.getBytes();//converting string into byte array  
 fout.write(b);  
 fout.close();  
 System.*out*.println("successfully written...........\n");  
 }  
 catch(Exception e){System.*out*.println(e);}  
 }  
 void out()  
 {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Enter file name to open with .txt extension");  
 String fname = sc.nextLine();  
 try  
 {  
 FileInputStream fin=new FileInputStream(fname);  
 int i=0;  
 while((i=fin.read())!=-1){  
 System.*out*.print((char)i);  
 }  
 System.*out*.println("\n");  
 fin.close();  
 }  
 catch(Exception e){System.*out*.println("File not found. please try again with correct file name. \n");  
 }  
  
 }  
 }  
 static class Help {  
 void help() {  
 System.*out*.println(" \*\*\*\*\*\*\*\*\*\* I can do the following things \*\*\*\*\*\*\*\*\*\* \n" +  
 "- Open Calculator ( type open calculator) \n" +  
 "- Open Notepad (type open notepad) \n" +  
 "- Open MS Paint (type open paint) \n" +  
 "- Open File Explorer (type open file explorer) \n " +  
 "- Open Google Chrome (type open chrome ) \n" +  
 "- Play games ( type play game) \n" +  
 "- Shutdown System (type shutdown )\n" +  
 "- Restart System (type restart ) \n" +  
 "- Check password strength (type password strength) \n" +  
 "- Create a memo/diary file(type diary)\n" +  
 "- Show system info (type systeminfo)\n" +  
 "- Exit ( Type Exit to terminate )\n");  
  
 }  
 }  
 static class calculator  
 {  
 void calc()  
 {  
 float a,b,res;  
 int choice;  
 Scanner S=new Scanner(System.*in*);  
  
 do  
 {  
  
 System.*out*.println("\n\nCalculator : \n" +  
 "Press 1 for Addition\n" +  
 "Press 2 for Subtraction\n" +  
 "Press 3 for Division\n" +  
 "Press 4 for Multiplication\n" +  
 "Press 5 to exit to main menu\n");  
  
 // enter the choice  
 System.*out*.print("Enter your choice : ");  
  
 // read the input choice value.  
 choice=S.nextInt();  
  
 switch(choice)  
 {  
 case 1:System.*out*.print("Enter two numbers : ");  
 a=S.nextFloat();  
 b=S.nextFloat();  
 res=a+b;  
 System.*out*.print("Result : " +res);  
 break;  
  
 case 2:System.*out*.print("Enter two numbers : ");  
 a=S.nextFloat();  
 b=S.nextFloat();  
 res=a-b;  
 System.*out*.print("Result : " +res);  
 break;  
  
 case 3:System.*out*.print("Enter two numbers : ");  
 a=S.nextFloat();  
 b=S.nextFloat();  
 res=a/b;  
 System.*out*.print("Result : " +res);  
 break;  
  
 case 4:System.*out*.print("Enter two numbers : ");  
 a=S.nextFloat();  
 b=S.nextFloat();  
 res=a\*b;  
 System.*out*.print("Result : " +res);  
 break;  
  
 case 5:  
 System.*out*.println("You are being redirected to main menu\n" +  
 "You are in main menu now\n");  
 *main*(null);  
 break;  
 default : System.*out*.print("Wrong Choice.....\n");  
 break;  
 }  
 }  
 // loop works till the number 5 not selected.  
 while(choice!=5);  
 }  
 }  
  
 public static void main(String[] args) {  
 DateTime dt = new DateTime();  
 dt.datetime();  
 Runtime app = Runtime.*getRuntime*();  
 Help hp = new Help();  
 GetSystemInfo sinfo = new GetSystemInfo();  
 Game gm = new Game();  
 pass p = new pass();  
 diary d = new diary();  
 calculator cal = new calculator();  
  
 while (true) {  
 System.*out*.println("Any order for me ? ");  
 Scanner input = new Scanner(System.*in*);  
  
 String command = input.nextLine();  
 try {  
  
 if (command.toLowerCase().contains("calculator")) {  
// app.exec("calc"); // open calculator app  
 cal.calc();//runs calculator function  
 } else if (command.toLowerCase().contains("help"))  
 hp.help(); // help menu  
 else if (command.toLowerCase().contains("password"))  
 p.passcheck(); // check password strenght  
 else if (command.toLowerCase().contains("ip address"))  
 sinfo.sysinfo(); // show system ip address  
 else if (command.toLowerCase().contains("notepad") || command.toLowerCase().contains("file"))  
 app.exec("notepad"); // opens notepad app  
 else if (command.toLowerCase().contains("calender"))  
 app.exec("calender"); // opens calender app  
 else if (command.toLowerCase().contains("explorer"))  
 app.exec("explorer"); // opens file explorer  
 else if (command.toLowerCase().contains("paint"))  
 app.exec("mspaint"); // opens file explorer  
 else if (command.toLowerCase().contains("chrome"))  
 app.exec("C:\\Program Files (x86)\\Google\\Chrome\\Application\\chrome.exe"); //opens chrome  
 else if (command.toLowerCase().contains("no")) {  
 System.*out*.println("Exiting, Thanks for using me.");  
 System.*exit*(0); // exit function  
 } else if (command.toLowerCase().contains("shutdown")) {  
 System.*out*.println("Are you sure you want to shutdown the system." +  
 " Type Yes to confirm or anything else to terminate"); // shutdown confirmation  
 String confirm = input.nextLine();  
 if (confirm.toLowerCase().contains("Yes")) {  
 Runtime.*getRuntime*().exec("c:\\Windows\\System32\\shutdown -s -t 0");//shudtown command  
 } else  
 System.*out*.println("System Shutdown Terminated");// shutdown terminate message  
 } else if (command.toLowerCase().contains("Restart")) {  
 System.*out*.println("Are you sure you want to Restart the system." +  
 " Type Yes to confirm or anything else to terminate"); // restart message  
 String confirm = input.nextLine();  
 if (confirm.toLowerCase().contains("Yes")) {  
 Runtime.*getRuntime*().exec("shutdown -r -t 0"); // restart command  
 } else  
 System.*out*.println("System Restart Terminated"); // restart terminated message  
 } else if (command.toLowerCase().contains("exit")) {  
 System.*out*.println("Exiting, Thanks for using me.");  
 System.*exit*(0); // exit function  
 } else if (command.toLowerCase().contains("systeminfo"))  
 {  
 sinfo.sysinfo(); // shows system info  
 System.*out*.println("Available Processors : "+app.availableProcessors());  
 System.*out*.printf(" Total Memory : "+app.totalMemory()/(1024.0\*1024.0) +" MB\n");  
 System.*out*.printf(" Free memory : "+app.freeMemory()/(1024.0\*1024.0) +" MB\n");  
 }  
 else if (command.toLowerCase().contains("diary"))  
 {  
 while (true) {  
 System.*out*.println("\nPress 1 to write in a file\n" +  
 "Press 2 to read from a file\n" +  
 "Press 3 to exit to main menu");  
  
 int inp = input.nextInt();  
  
 switch (inp) {  
 case 1:  
 d.inp();  
 break;  
 case 2:  
 d.out();  
 break;  
 case 3 :  
 System.*out*.println("You are in main menu now\n");  
 *main*(null);  
 default:  
 System.*out*.println("enter correct choice");  
 }  
  
 }  
 }  
 else if (command.toLowerCase().contains("game")) {  
 System.*out*.println("Which game you want to play : \n" +  
 " Press 1 for Rock paper Scissor \n" +  
 " Press 2 for number guessing game \n");  
 int GameInput = input.nextInt();  
 switch (GameInput) {  
 case 1:  
 gm.rsp();  
 break;  
 case 2:  
 gm.numguess();  
 break;  
 default:  
 System.*out*.println("Invalid Choice");  
 }  
 }  
  
 } catch (Exception Ex) {  
 System.*out*.println("Sorry, I didn't get that");  
 }  
 }  
 }  
}

**Output –**





**References**

* [www.google.com](http://www.google.com)
* [www.javatpoint.com](http://www.javatpoint.com)
* [www.geeksforgeeks.com](http://www.geeksforgeeks.com)
* [www.stackoverflow.com](http://www.stackoverflow.com)