



Green University of Bangladesh
Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Semester: (Summer, Year: 2021), B.Sc. in CSE (Day)

Course Title: Computer Networking Lab
Course Code: CSE 312 Section: DA

Lab Project Name
Baymath: A Mathematical Chatbot

Student Details

Name	ID
Tarekul Islam	191002300

Submission Date : 14. 09. 2021
Course Teacher's Name : Mr. Palash Roy

Lab Project Status

Marks:

Signature:

Comments:

Date:

Table of Contents

Chapter 1	3
Introduction.....	3
1.1 Introduction.....	3
1.2 Design Goals/Objective	3
1.3 Prerequisites & requirements	4
Chapter 2	5
Design/Development/Implementation of the Project.....	5
2.1 Methodology	5
2.1.2 Project Design.....	5
2.2 Coding.....	6
Server Side:	6
2.2.1 Sever Side Main Method	6
2.2.2 Reading and Conversation Method.....	6
2.2.3 MathLab Method	7
Client Side :	8
2.2.4 Instance Variables in The User Class	8
2.2.5 Main , Connection & Reading Method.....	8
2.2.6 Event Handle Method	9
2.2.7 JavaFX Start Method	9
2.2.8 GUI Create Method.....	9
2.2.9 CSS Style Code.....	10
2.2.10 Module-info.java.....	11
Chapter 3	11
Performance Evaluation	11
3.1 Simulation Environment/ Simulation Procedure.....	11
3.2 Results and Discussions	13
3.2.2 Some Instruction	14
3.2.3 Technology Used:	15
Chapter 4 Conclusion	16
4.1 Introduction.....	16
4.1 Practical Implications	16
4.2 Scope of Future Work.....	16
References.....	16

Chapter 1

Introduction

1.1 Introduction

Chatbots, or conversational interfaces as they are also known, present a new way for individuals to interact with computer systems. Traditionally, to get a question answered by a software program involved using a search engine, or filling out a form. A chatbot allows a user to simply ask questions in the same manner that they would address a human. The most well-known chatbots currently are voice chatbots: Alexa and Siri. However, chatbots are currently being adopted at a high rate on computer chat platforms. The technology at the core of the rise of the chatbot is natural language processing (“NLP”). Recent advances in machine learning have greatly improved the accuracy and effectiveness of natural language processing, making chatbots a viable option for many organizations. This improvement in NLP is firing a great deal of additional research which should lead to continued improvement in the effectiveness of chatbots in the years to come

This is a program that communicates with us. It is a layer on top of, or a gateway to, a service. Sometimes it is powered by machine learning (the chatbot gets smarter the more you interact with it). Or, more commonly, it is driven using intelligent rules (i.e. if the person says this, respond with that). The services a chatbot can deliver are diverse. Important life-saving health messages, to check the weather forecast or to purchase a new pair of shoes, and anything else in between. The term chatbot is synonymous with text conversation but is growing quickly through voice communication... “Alexa, what time is it?” (other voice-chatbots are available!). The chatbot can talk to you through different channels; such as Facebook Messenger, Siri, WeChat, Telegram, SMS, Slack, Skype and many others. Consumers spend lots of time using messaging applications. Therefore, messaging applications are currently the most popular way companies deliver chatbot experiences to consumers.

This project is also a chatbot but not so advance, like that we mention above. our chatbot can communicate with user and can Do any kind of mathematical Operation and also can chat little bit further information.

1.2 Design Goals/Objective

A chatbot is a software application used to conduct an on-line chat conversation via text or text-to-speech, in lieu of providing direct contact with a live human agent. Designed to convincingly simulate the way a human would behave as a conversational partner, chatbot systems typically require continuous tuning and testing, and many in production remain unable to adequately converse or pass the industry

standard Turing test.

Chatbots are used in dialog systems for various purposes including customer service, request routing, or information gathering. While some chatbot applications use extensive word-classification processes, natural language processors, and sophisticated AI, others simply scan for general keywords and generate responses using common phrases obtained from an associated library or database.

Most chatbots are accessed on-line via website popups or through virtual assistants. They can be classified-into-usage-categories-that include: commerce (ecommerce viachat), education, finance, news entertainment, health, and productivity.

We mainly do this project using Java Programming Language. we use some frameworks of java like JavaFX for creating Graphical User Interface. and Java.net Socket programming. This is an offline chatbot. there is no need to Internet connection and also this is a Desktop application. it can be run on Desktop in windows. There are broadly two variants of chatbots. One follows a set of rules, flows, and triggers to respond to very specific commands. The other variant uses machine learning to try to understand the sentiment and meaning of the language used, to not rely on pre-planned commands. So, ours is first variant chatbot. its follows a set of rules, flows and trigger to respond with user message. our chatbot can do Mathematical calculation the user sent by a fixed sequence.

1.3 Prerequisites & requirements

Minimum Software Requirements:

- Any operating system capable of running JVM.
- IntelliJ IDEA
- Java 16
- JAVA FX version 11.0 or higher

Minimum Hardware Requirements:

- Processor – Any processor capable of running IntelliJ IDEA or other IDE.
- Hard Disk – 20 GB
- Memory – 40 GB RAM

Chapter 2

Design/Development/Implementation of the Project

2.1 Methodology

The methodology of developing of project will be a step-by-step sequence to design, develop and deliver the application. In software engineering this methodology called 'waterfall model' which one portion of work follows after another in a linear sequence. Following steps will be followed in this methodology:

- Initiation (Requirement Specification);
- Planning and design;
- Execution (construction and coding);
- Validation (Testing);
- Closure (Installation and Maintenance).

2.1.1 Requirement Specification

By project requirements specifications we can analyze the tasks which going to be done by the system. The function and performance of allocated to software as part of system engineering are refined by establishing a complete information description. A detailed functional and behavioral description of the project and concentrating on requirements and constraints of that will provide and good product.

2.1.2 Project Design

For designing and implementing we used JavaFX for UI Design and for backend we use Socket programming to Communicate between server and user. For Designing we divide the UI interface into three section:

Top Bar:

In this section we initialize our Title: Mathematica using in a Horizontal Box (HBox). Its background color is dark green. We can also use here an image but this time we will ignore it. The Height of the HBox is 50 px. For styling we use here css style class appropriate with javafx.

Middle Box:

This a the section where the conversion of server and user are shown by people. Here we use a TextArea and we do Editable of this section false so that no one can edit the showed conversation. In this section the server reply will show left side and user sent message will show in right side. We set its height 490 px. And it's a Scrollable tab.

Bottom Bar:

In this section we added send button and a Text Field, where user can write their text's and queries. Its height same as Topbar. Here also use css style class appropriate with javafx for styling. And here used another HBox.

2.2 Coding

As mentioned before, project is on Java Socket Programming and it's a windows based application which is coded in IntelliJ IDEA and Run using JVM and JavaFx . So coding exist for all following pages:

Server Side:

2.2.1 Sever Side Main Method

```
public class Server extends Thread {
    public static void main(String[] args) throws IOException {

        ServerSocket serverSocket = new ServerSocket(12);
        Socket socket = null;
        int c = 0;
        while (true) {
            if (c > 10) {
                socket.close();
                serverSocket.close();
                System.exit(0);
                break;
            } else {
                c++;
                socket = serverSocket.accept();
                DataInputStream in = new DataInputStream(socket.getInputStream());
                DataOutputStream dos = new
DataOutputStream(socket.getOutputStream());
                reading(socket, c,in,dos);
                System.out.println("User "+c);
            }
        }
    }
}
```

2.2.2 Reading and Conversation Method

```
private static void reading(Socket socket, int c,DataInputStream
in,DataOutputStream dos) {
    Runnable r1 = () -> {

        while (true) {
            try {
                String msg = in.readUTF();
                String[] a = msg.split(",");
                int s = a.length;

                if(msg.equals("hi") || msg.equals("Hi")){
                    dos.writeUTF("Hi! I am Baymath\nHow can I help you?");
                }
                else if(msg.equals("How are you") || msg.equals("how are you?")){
                    dos.writeUTF("I am Always Fine.\n How can I help you?");
                }
                else if (msg.equals("What do you can?")){
                    dos.writeUTF("I can Do Mathematical Calculations.\nFollow the
given Format (operator,value1,value2)");
                }
                else if (msg.equals("bye")) {

                    dos.writeUTF("bye");
                }
                else if(msg.equals("exit")){

```

```

        socket.close();
        System.exit(0);
    }
    else {
        MathLab(a, s, dos);
    }
} catch (IOException e) {
}
}
};
new Thread(r1).start();
}

```

2.2.3 MathLab Method

```

private static void MathLab(String[] a, int s, DataOutputStream dos) throws
IOException {
    String[] operators = {"+", "-", "*", "/", "%", "root", "abs", "exp", "log",
"log10", "pow", "sin", "cos", "tan", "deg", "rad"};

    if (a[0].equals("+")) {
        dos.writeUTF(String.valueOf(Integer.parseInt(a[1]) +
Integer.parseInt(a[2])));
    } else if (a[0].equals("-")) {
        dos.writeUTF(String.valueOf(Integer.parseInt(a[1]) -
Integer.parseInt(a[2])));
    } else if (a[0].equals("/")) {
        dos.writeUTF(String.valueOf((Double.parseDouble(a[1]) /
Integer.parseInt(a[2]))));
    } else if (a[0].equals("*")) {
        dos.writeUTF(String.valueOf(Integer.parseInt(a[1]) *
Integer.parseInt(a[2])));
    } else if (a[0].equals("%")) {
        dos.writeUTF(String.valueOf(Integer.parseInt(a[1]) %
Integer.parseInt(a[2])));
    } else if (a[0].equals("root")) {
        dos.writeUTF(String.valueOf((Math.sqrt(Integer.parseInt(a[1])))));
    } else if (a[0].equals("abs")) {
        dos.writeUTF(String.valueOf((Math.abs(Integer.parseInt(a[1])))));
    } else if (a[0].equals("exp")) {
        dos.writeUTF(String.valueOf((Math.exp(Integer.parseInt(a[1])))));
    } else if (a[0].equals("log")) {
        dos.writeUTF(String.valueOf((Math.log(Double.parseDouble(a[1])))));
    } else if (a[0].equals("pow")) {
        dos.writeUTF(String.valueOf((Math.pow(Double.parseDouble(a[1]),
Double.parseDouble(a[2])))));
    } else if (a[0].equals("log10")) {
        dos.writeUTF(String.valueOf((Math.log10(Double.parseDouble(a[1])))));
    } else if (a[0].equals("sin")) {
        dos.writeUTF(String.valueOf((Math.sin(Double.parseDouble(a[1])))));
    } else if (a[0].equals("cos")) {
        dos.writeUTF(String.valueOf((Math.cos(Double.parseDouble(a[1])))));
    } else if (a[0].equals("tan")) {
        dos.writeUTF(String.valueOf((Math.tan(Double.parseDouble(a[1])))));
    } else if (a[0].equals("deg")) {
        dos.writeUTF(String.valueOf((Math.toDegrees(Double.parseDouble(a[1])))));
    } else if (a[0].equals("rad")) {
        dos.writeUTF(String.valueOf((Math.toRadians(Double.parseDouble(a[1])))));
    } else {
        dos.writeUTF("Sorry! I can't understand!");
    }
}

```

```

    }
}

```

Client Side :

2.2.4 Instance Variables in The User Class

```

public class User extends Application {
    public static final TextArea showMessage=new TextArea();
    public static final TextField inputMsg=new TextField();
    public static final Button sendBtn=new Button();
    private static final Circle image=new Circle();
    private static final Label Name = new Label();
    private static Socket socket;
    private static Font font=new Font("Roboto",15);
    private static DataInputStream din;
    private static DataOutputStream dos;
}

```

2.2.5 Main , Connection & Reading Method

```

public static void main(String[] args) throws IOException {
    Connection();
    launch(args);
}

private static void Connection() throws IOException {
    socket=new Socket("localhost",12);
    din=new DataInputStream(socket.getInputStream());
    dos=new DataOutputStream(socket.getOutputStream());

    handleEvent();
    reading();
}

private static void reading() {
    Runnable r1=()->{
        while (!socket.isClosed()){
            try {
                String msg=din.readUTF();
                if (msg.equals("bye")){
                    socket.close();
                    System.exit(0);
                    break;
                }
            }
            else{
                showMessage.appendText(" "+msg+"\n");
            }
        }
    } catch (IOException e) {
    }
}

};
new Thread(r1).start();
}

```


2.2.6 Event Handle Method

```
private static void handleEvent() throws IOException {  
  
    inputMsg.setAction(event -> {  
        String msg=inputMsg.getText();  
        showMessage.appendText("\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t"+msg + "\n");  
  
        try {  
            dos.writeUTF(msg);  
            inputMsg.setText("");  
            inputMsg.requestFocus();  
        } catch (IOException e) {  
  
        }  
    });  
  
    sendBtn.setAction(event -> {  
        String msg=inputMsg.getText();  
        showMessage.appendText("\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t"+msg + "\n");  
        try {  
            dos.writeUTF(msg);  
            inputMsg.setText("");  
            inputMsg.requestFocus();  
        } catch (IOException e) {  
  
        }  
    });  
  
}
```

2.2.7 JavaFX Start Method

```
@Override
public void start(Stage primaryStage) throws Exception {

    primaryStage.setScene(new Scene(create()));
    primaryStage.setTitle("Mathematical Robot");
    primaryStage.show();
}
```

2.2.8 GUI Create Method

```
private Parent create() {
    HBox topBar=new HBox(10,Name);
    topBar.setPrefHeight(50);

    topBar.getStylesheets().addAll(this.getClass().getResource("style.css").toExternalForm());

    topBar.getStyleClass().add("bg");
    topBar.setAlignment(Pos.CENTER);
    topBar.paddingProperty().set(new Insets(5,10,5,20));

    showMessage.setPrefHeight(490);
    showMessage.setFont(font);
    showMessage.setEditable(false);
}
```

```

    Name.setText("MATHEMATiCA");
    Name.setFont(Font.font("Britannic Bold", FontWeight.BOLD, 50));
    Name.setTextFill(Paint.valueOf("WHITE"));
    Name.alignmentProperty().set(Pos.CENTER);

    HBox bottomBar=new HBox(20, inputMsg, sendBtn);

    bottomBar.getStylesheets().addAll(this.getClass().getResource("style.css").toExternalForm());
    bottomBar.getStyleClass().add("bg");
    bottomBar.setAlignment(Pos.CENTER_LEFT);
    bottomBar.setPrefHeight(55);

    inputMsg.setPrefHeight(40);
    inputMsg.setPrefWidth(520);
    inputMsg.getStyleClass().add("textfield");

    sendBtn.setText("SEND");
    sendBtn.setTextFill(Paint.valueOf("WHITE"));
    sendBtn.setFont(font);
    sendBtn.setPrefHeight(40);
    sendBtn.setPrefWidth(160);
    sendBtn.getStyleClass().add("btn");

    VBox root=new VBox(0, topBar, showMessage, bottomBar);

    root.setPrefSize(700, 600);
    return root;
}
}

```

2.2.9 CSS Style Code

```

.bg{

    -fx-background-color:linear-gradient(to top, rgb(40, 145, 146), rgb(51, 139, 147));;
}

.textfield{

    -fx-background-radius: 0 30 30 0;
    -fx-background-color:white;

}

.btn{
    -fx-background-color:linear-gradient(to right, rgb(70, 185, 146), rgb(51, 160, 147));;
    -fx-background-radius: 30 0 0 30;
    -fx-padding:0 0 0 20;

}
.file{
    -fx-background-color:linear-gradient(to right, rgb(70, 185, 146), rgb(51, 160, 147));;
    -fx-background-radius: 30 30 30 30;

}
.btn:pressed{

```

```

    -fx-background-color: rgba(255, 255, 255, 0.2);
}

.img{
-fx-background-radius: 30 30 30 30;
}

```

2.2.10 Module-info.java

```

module Mathematical.Robot {
    requires javafx.controls;
    requires javafx.media;
    requires javafx.swing;
    requires javafx.web;
    requires javafx.graphics;
    requires javafx.base;
    requires javafx.fxml;

    opens classes;
}

```

Chapter 3

Performance Evaluation

3.1 Simulation Environment/ Simulation Procedure

For Simulating this project We need javafx runnable Environment. Here we use intellij idea as IDE and JavaFX SDK version 17. And Java Version 16.0.2. To Simulate the Project there should be JavaFX and Java install first.

Procedure:

- Open The Project in IntelliJ IDEA or any Other IDE where JavaFX installed.
- Now Go to file and Click Project Structure Tab
- And Click Global Libraries
- Click the + Icon and Set the JavaFX lib folder Like Fig: 01
- After that click Project Tab and in Project section select available Java SDK version Like fig: 02
- Click ok and in the Classes Package first run the Server like fig: 03
- After that Right click user and Run the User class.

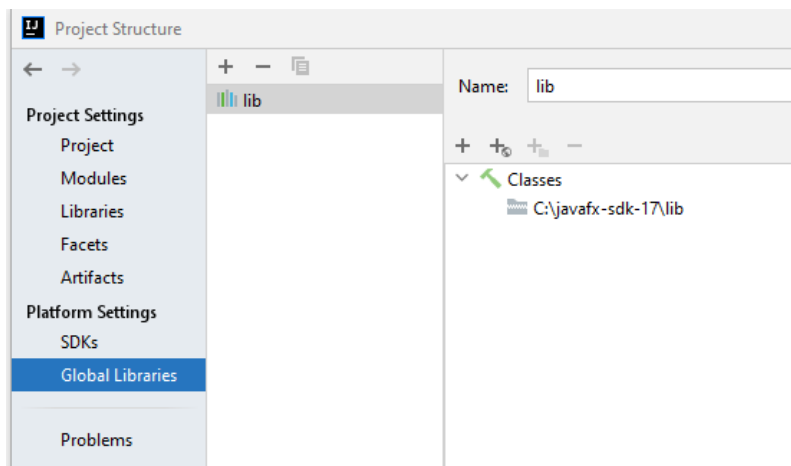


Fig: 01

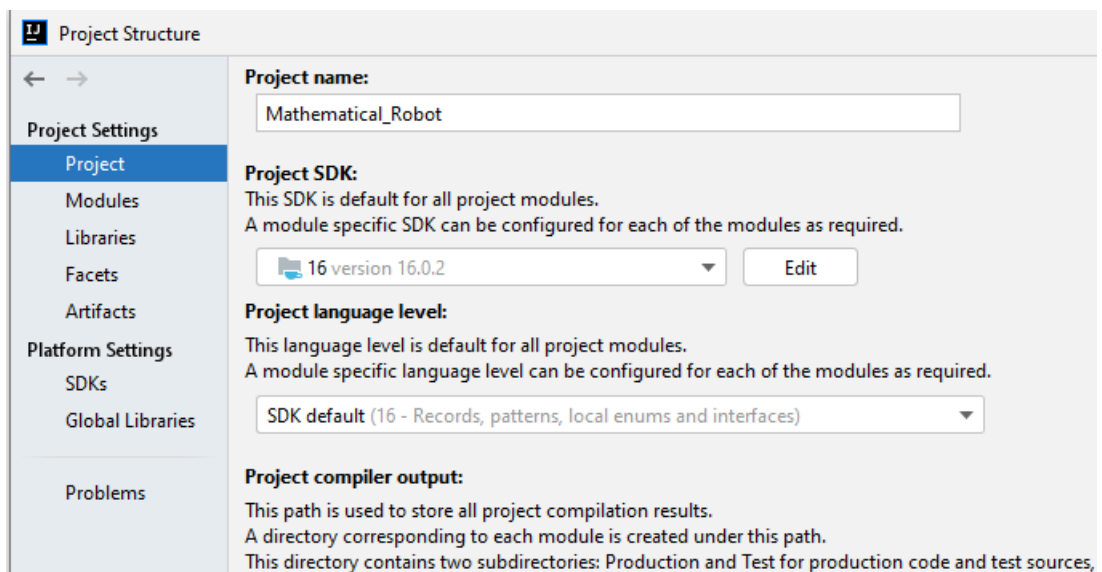


Fig: 02

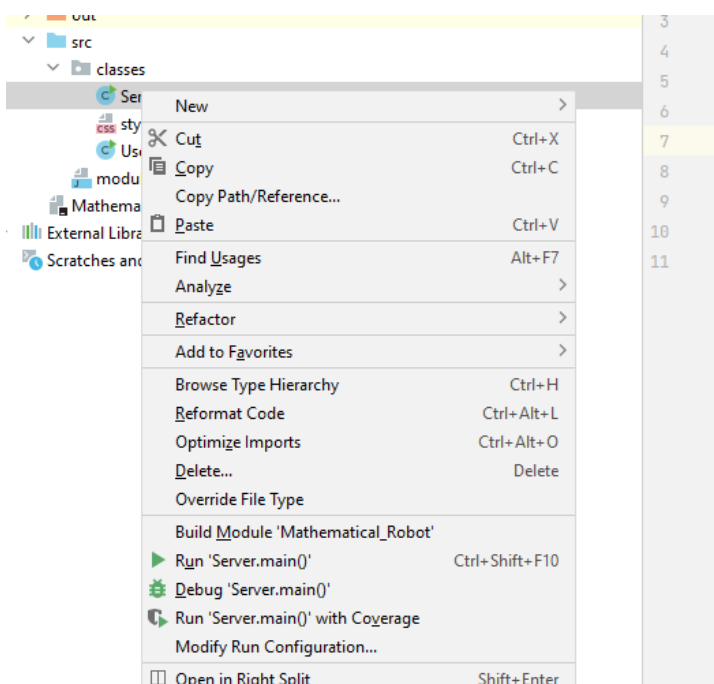


Fig: 03

3.2 Results and Discussions

3.2.1 Results/Testing

After completing the above task we will show the Main Window of chatbot Application. We can successfully Run our application and it will working properly. Some screenshot of our Application is given below:

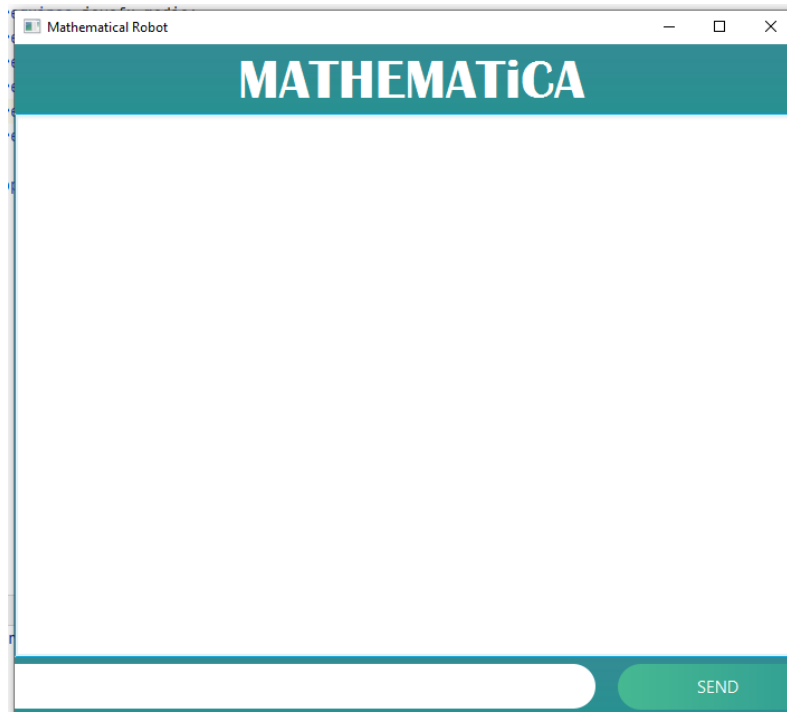


Fig: 04

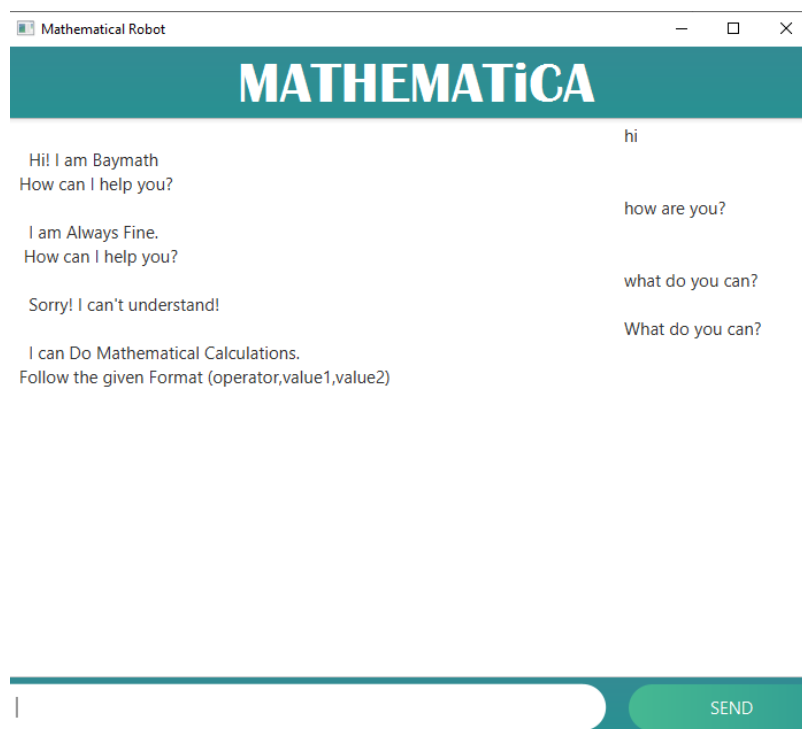


Fig: 05

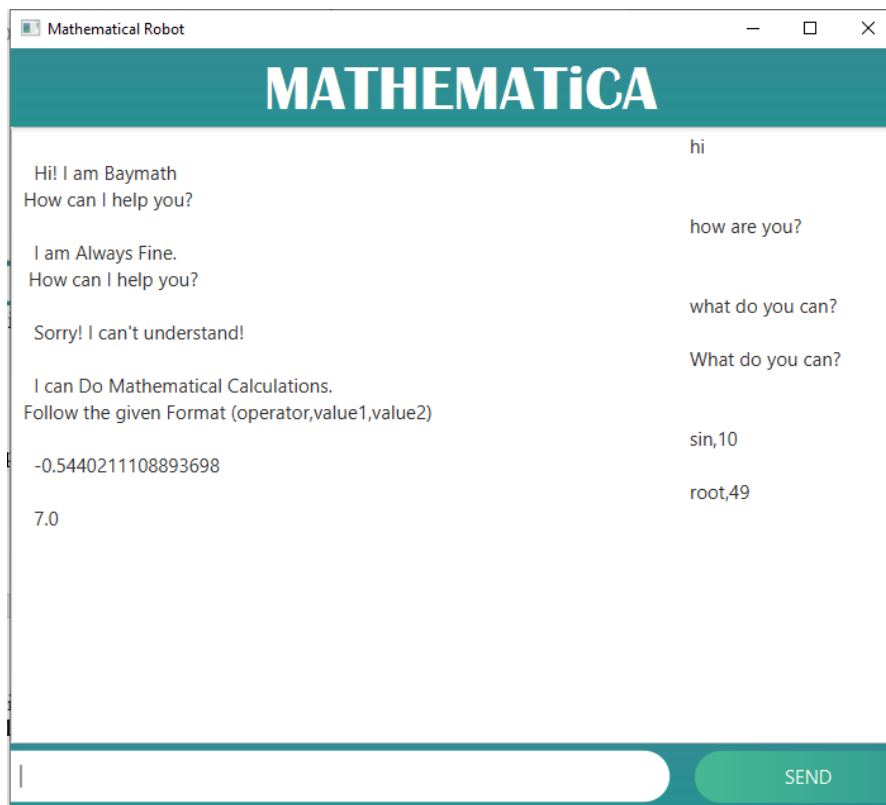


Fig: 06

3.2.2 Some Instruction

You need to follow some keyword. That are given below:

- For Summation = +,value1,value2
- For Subtraction = -,value1,value2
- For Division = /,value1,value2
- For Multiplication = *,value1,value2
- For Mod = %,value1,value2
- For Square Root = root,value1
- For Absolute = abs,value1
- For Exponential = exp,value1
- For logarithm = log,value1
- For 10 based log = log10,value1
- For Square = pow,value1,value2
- For Sin = sin,value1,value2
- For Cos = cos,value1
- For tan = tan,value1
- For Degree = deg,value1
- For Radiant = rad,value1

Some Conversation:

- Hi
- How are you?
- What do you can?
- Bye for Exiting

3.2.2 Analysis and Outcome

The feasibility study is an evaluation and analysis of the potential of a proposed project which is based on extensive investigation and research to support the process of decision making. Depending on the results of the initial investigation the survey is now expanded to a more detailed feasibility study.

3.2.3 Technology Used:

- Java Version 16
- JavaFX version 17
- IntelliJ IDEA version 2021.2.1
- Socket Programming

Chapter 4

Conclusion

4.1 Introduction

In conclusion, Baymath: A Mathematical chatbot is a desktop application that uses Limited functionalities. which will provide answers to the limited question answers and Mathematical calculation of the user. Users will have to follow some format for questioning. It can also be used as normal and traditional and calculating system as well. The main purpose of building this application is to make chatbot a little effort in calculation for students. There are already many similar desktop and Web applications and they are using advanced chatbot system. So this Desktop application tries to help little bit in calculating students and Learners.

4.1 Practical Implications

- Customers Prefer Texting: The primary means of communication with a chatbot is via text. And customers prefer this method
- Chatbot is Instant & 24/7: A chatbot, unlike a customer support agent, never takes breaks, sleeps or goes offline. With chatbots, businesses can provide 24/7 customer support
- Chatbots Answer Questions and Inquiries
- Book Tickets to Events/Shows with Chatbots
- To Find Products, Check Inventory and Recommend Items

4.2 Scope of Future Work

In future we can add AI into this project so that the limitation of this project will overcome much. We create a version for web and also add database etc. Some of the enhancement that will added soon are given below:

- Self-training of chatbot
- Improving training dataset
- Enhancement in response time
- Proper Database connectivity
- More Interactive interface
- AI improvement and bug fixed

References

- [1] <https://docs.oracle.com/javafx/2/>
- [2] <https://www.uio.no/studier/>
- [3] <https://www.geeksforgeeks.org/>.