

FEEES RECEIPT GENERATOR

Y.Baladithya	AP21110010210
Gnana Kartheek	AP21110010238
Ch.Bhargav	AP21110010252
N.Chaitanya Sai	AP21110010253

INTRODUCTION

MODULES USED

- A. GUI
- B. Generate Receipt
- C. Reset Receipt
- D. Print Receipt

Graphical User Interface (GUI)

GUI

- Addnewframe :
 - Instance of JFrame class.
 - Represent main window of application.
- JTextField :
 - Provides single line text field.
 - getText() method return current text value ;
 - setText() method sets the text to be displayed.
- JTextArea :
 - Provides multi text area.
- JButton :
 - Provides clickable buttons.

```
static JFrame addnewframe = new JFrame();
static JTextField amounttext = new JTextField();
static JTextField snametext = new JTextField();
static JTextField sclasstext = new JTextField();
static JTextField fnametext = new JTextField();
static JTextField fmobiletext = new JTextField();
static JTextField mnametext = new JTextField();
static JTextField mmobiletext = new JTextField();
static JTextField saddresstext = new JTextField();
static JTextField smobiletext = new JTextField();
static JTextField dojtext = new JTextField();
static JTextArea receiptarea = new JTextArea();
static JButton reset = new JButton(text:"RESET");
static JButton generatereceipt = new JButton(text:"GENERATE RECEIPT");
static JButton printreceipt = new JButton(text:"PRINT RECEIPT");
static int num =0;
```

Generate Receipt

Generate receipt

```
public static void generatereceipt_action(ActionEvent event)
{

    try
    {
        FileInputStream fin = new FileInputStream(name:"receipt number.txt");
        int i = 0;

        while((i = fin.read())!= -1)
        {
            num=i;
        }

        fin.close();
    }

    catch(Exception e)
    {
        System.out.println(e);
    }
}
```

```
Date obj = new Date();
```

```
String date = obj.toString();
```

```
receiptarea.setText(t+"\t*****\n");
receiptarea.setText(""+receiptarea.getText()+"\t CONCEPT CLASSES\n");
receiptarea.setText(""+receiptarea.getText()+"\t*****\n\n"+date+"    Receipt Number : "+(num+1));
receiptarea.setText(""+receiptarea.getText()+"\n\nSTUDENT NAME : "+sname.getText());
receiptarea.setText(""+receiptarea.getText()+"\n\nFATHER NAME : "+fname.getText());
receiptarea.setText(""+receiptarea.getText()+"\n\nMOTHER NAME : "+mname.getText());
receiptarea.setText(""+receiptarea.getText()+"\n\nADDRESS : "+saddres.getText());
receiptarea.setText(""+receiptarea.getText()+"\n\nDATE OF JOINING : "+doj.getText());
receiptarea.setText(""+receiptarea.getText()+"\n\nCLASS : "+sclas.getText());
receiptarea.setText(""+receiptarea.getText()+"\n\nFATHER MOBILE : "+fmobile.getText());
receiptarea.setText(""+receiptarea.getText()+"\n\nMOTHER MOBILE : "+mmobile.getText());
receiptarea.setText(""+receiptarea.getText()+"\n\nSTUDENT MOBILE : "+smobile.getText());
receiptarea.setText(""+receiptarea.getText()+"\n\nFEE AMOUNT : "+amount.getText()+"\t\t    SIGNATURE");
```

Generate Receipt

- The code snippet you provided shows a static method named `generatereceipt_action` that takes an `ActionEvent` parameter. This method is likely used as an event handler for generating a receipt in your Java Swing application.
- The method attempts to read the content of a file named "receipt number.txt" using a `FileInputStream`. It reads the file byte by byte until reaching the end and stores the last read value in the `num` variable. This is done to retrieve the current receipt number.
- The generated receipt includes details such as the current date, receipt number, student name, father name, mother name, address, date of joining, class, father mobile, mother mobile, student mobile, and fee amount.

Reset Receipt

Reset Receipt

- This reset class for clearing all the input fields.

```
public static void reset_action(ActionEvent event)
{
    amounttext.setText(t: "");
    sname.setText(t: "");
    sclasstext.setText(t: "");
    fname.setText(t: "");
    fmobiletext.setText(t: "");
    mname.setText(t: "");
    mmobiletext.setText(t: "");
    saddresstext.setText(t: "");
    smobiletext.setText(t: "");
    dojtext.setText(t: "");
    receiptarea.setText(t: "");
}
```

Print Receipt

Print Receipt

In the above code, I assume that you have a `TextArea` object named `receiptArea` that contains the receipt text. You can modify it according to your application's specific UI components. The code prints the receipt text to the console using `System.out.println()` and appends it to a file named "receipt.txt" using `FileOutputStream` and `PrintWriter`. The try-catch block handles any `IOException` that may occur during file operations. If an exception occurs, it prints the stack trace.

```
public static void printreceipt_action(ActionEvent event)
{
    try{
        receiptarea.print();
    }
    catch(Exception e)
    {
    }

    try
    {
        FileOutputStream fout = new FileOutputStream(name:"receipt number.txt",append:true);

        num++;
        fout.write(num);
        fout.close();
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
}
```

INTERFACE



FEE RECEIPT

STUDENT NAME :

FATHER'S NAME:

MOTHER'S NAME:

ADDRESS :

DATE OF JOINING :

CLASS :

FATHER'S PH:

MOTHER'S PH:

STUDENT'S PH:

FEE AMOUNT :

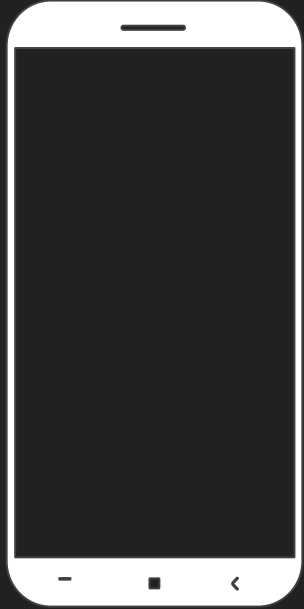
GENERATE RECEIPT

RESET

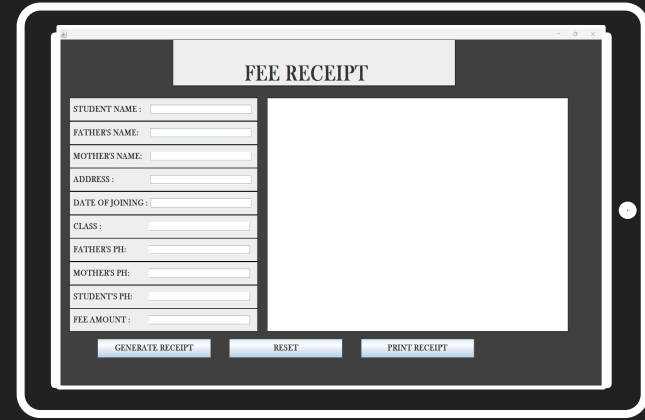
PRINT RECEIPT

FUTURE WORK

Mobile mockup



Tablet mockup



Desktop mockup

Conclusion

In conclusion, the Java project presented here showcases the power of Swing components and event handling to create a seamless student receipt generation system. By leveraging the flexibility of the Java programming language, the project provides a user-friendly interface that enables efficient data input and processing. Through the integration of various JTextField and JTextArea elements, the system allows for easy entry and display of student information, including names, class details, parental contact, and fee amounts. The generated receipts are conveniently presented in a JTextArea, providing a clear overview of the transaction details.

Furthermore, the project incorporates essential functionalities such as receipt resetting, printing, and the maintenance of a unique receipt number. These features contribute to enhanced record-keeping and facilitate efficient administrative tasks within educational institutions.

THANK YOU

Github Repository Link:

https://github.com/chaitanyasai-2021/Fees_Receipt_Generator