

SIKSHA 'O' ANUSANDHAN
DEEMED TO BE UNIVERSITY

Admission Batch: 2019

Session: 2020-2021

End-Term Project Report

Computer Science Workshop 1 (CSE 3141)

Submitted by

Name: Amrit Kafle

Registration No.: 19410121179

Branch: Computer Science And Engineering

Semester: 3 Section: M



Department of Computer Science & Engineering

Faculty of Engineering & Technology (ITER)

Jagamohan Nagar, Jagamara, Bhubaneswar, Odisha - 751030

Table Of Contents

Contents

Table Of Contents	2
Problem Statement	2
• Features Requested	3
• Features added by Developer	3
Problem Analysis	3
• Database	4
• GUI	4
Flow of Programme	5
• Database Flowchart	Error! Bookmark not defined.
• GUI Flowchart	5
Java Code	8
Database Code	8
✦ Database.java	8
GUI Codes	11
✦ App.java	11
Snapshots of Input and Output	33
Some Important Points Regarding Project Development:	44

Problem Statement

Call Management System

- Features Requested

This project is a simple automated Call Management System. In this project you will be learning how to add new customer to the database, how to generate a 10 digit unique phone number for each customer, how to set number as Prepaid/Postpaid type. Also, you will be implementing the following operations add customer, view customer usage details, update call balance (apply call tariffs), calculate due amount and next recharge date, etc..

- Instructions for implementation

- Design GUI based interface for user, by using which a user is able to login to system with proper authentication.
- Specify operation according to type of user i.e. if the user is a Manager he/she is able to insert new customer, update call details, calculate next recharge date and amount. If the user is a customer, he is only able to see his account details and perform recharge operation.
- A User is able to get the call details of the specified number in a nicely formatted text file.
- Use any database (Oracle / MySQL /SQLite) to store the required data.

Problem Analysis

- Database

- ✦ Should have a connection with mySql server .
- ✦ Should have insert method that can insert data into the database.
- ✦ For removing customer there should be a delete method.
- ✦ For updating the info about a customer there should be a update method in database class.
- ✦ There should be a get method for executing select queries which are used get data from database.
- ✦ There should be two tables in the database , one for Storing data of the customers and another for storing Authentication data.

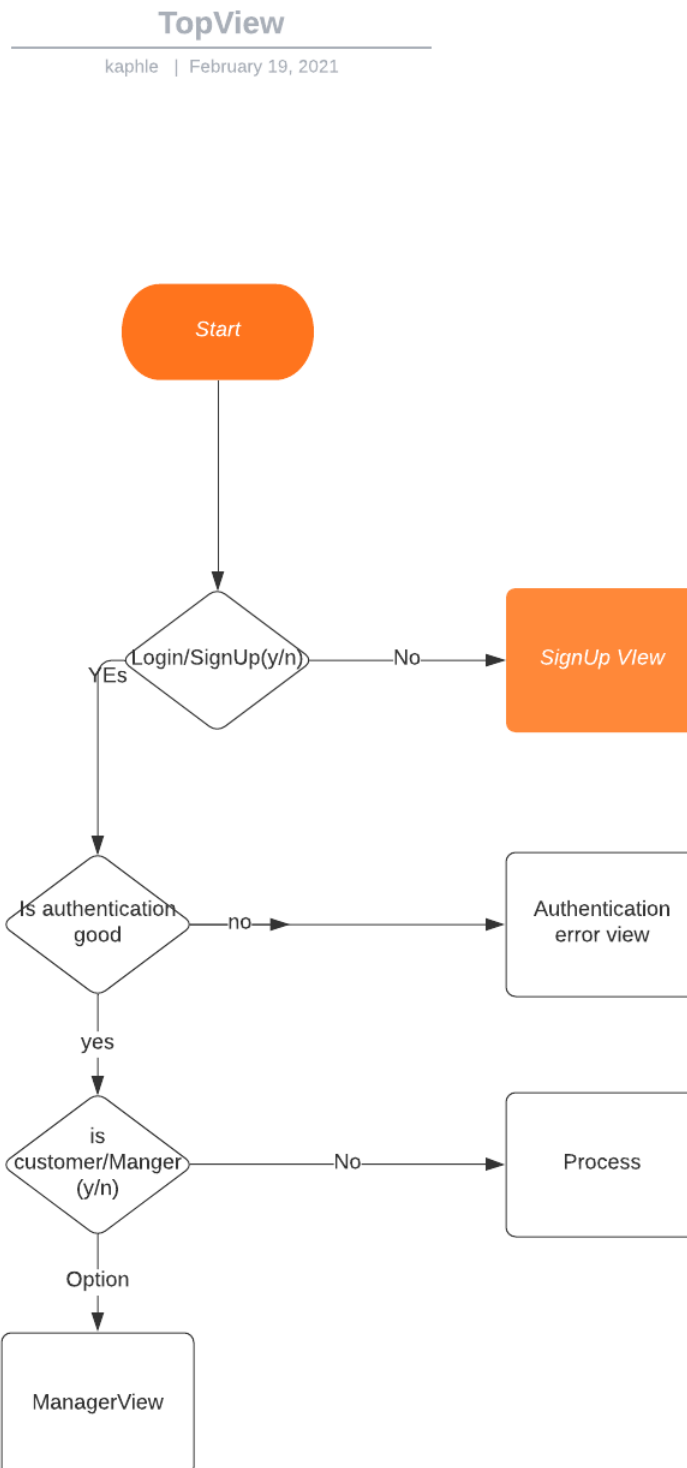
- GUI

- ✦ The GUI should first show the login page with a Button for creating new customer.
- ✦ If the username or password is wrong , it should be Informed .
- ✦ If the person logging in is a manager , he/she should Be prompted to manager View.
- ✦ If the person is a customer , he/she should be prompted to customer View.
- ✦ All the features like recharging, viewing balance, update info, knowing if its prepaid or postpaid, next Recharge date ,details about self, etc. should be provided by Customer View.
- ✦ Features like removing a customer , adding a customer, listing all the customers and all the things that a customer can do should be provided by Manager View.

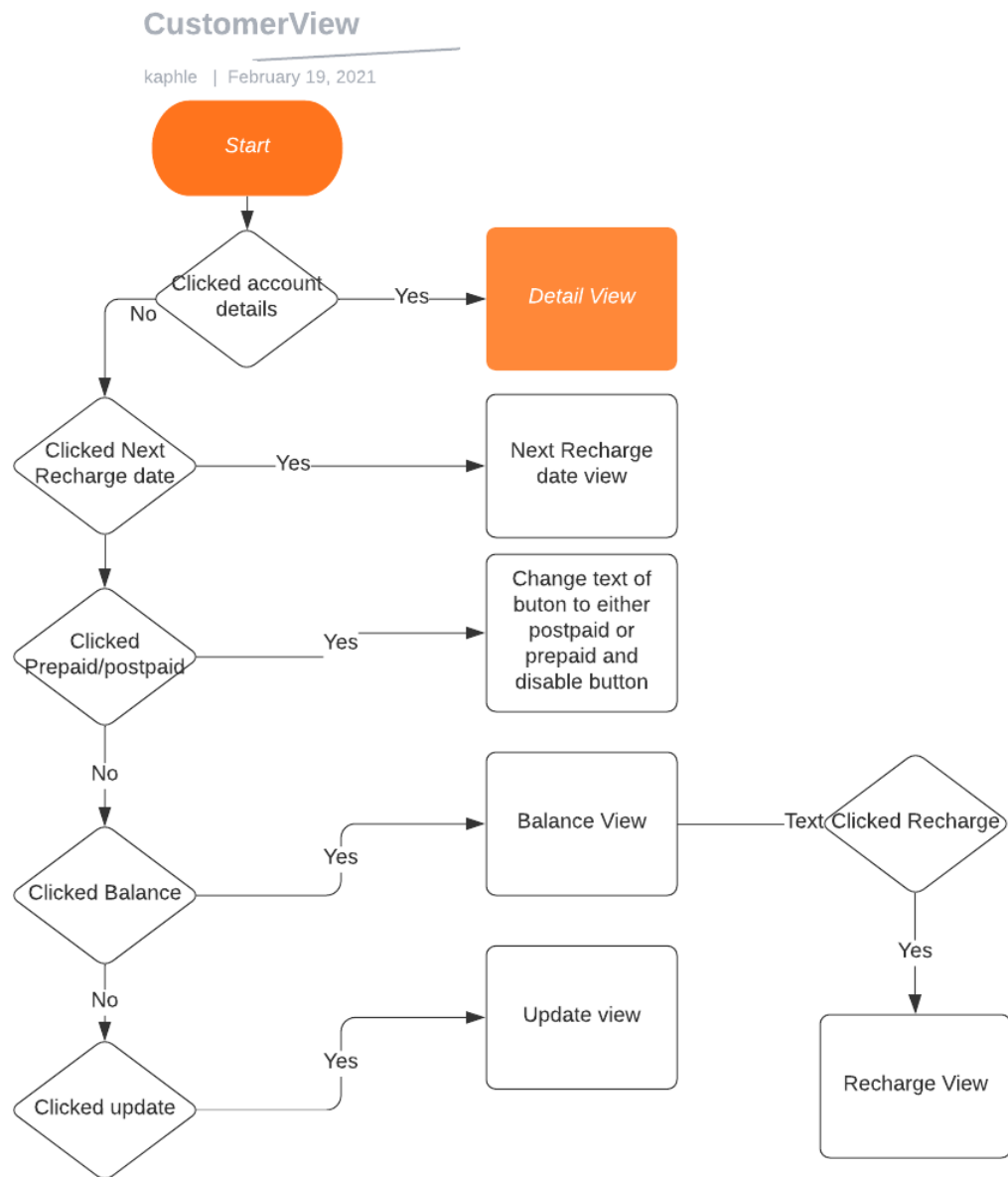
Flow of Program

- GUI flowchart

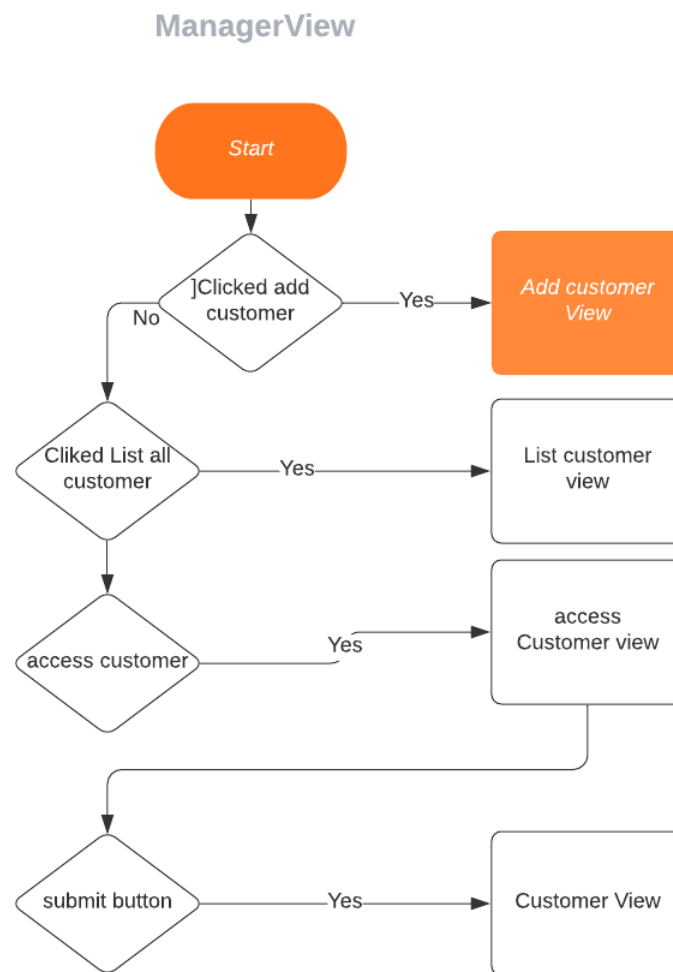
1. Top View:



2. Customer View :



3. Manager View :



Java Code

MySql Code

★ cswProject.sql

```
create database javadb;

use javadb;

create table authTable(
customer_id int not null auto_increment,
username varchar(100) not null,
pass varchar(100) not null,
position varchar(100) not null,
primary key(customer_id)
);

create table myTable (
customer_id int not null auto_increment,
namee varchar(100) not null,
phone varchar(100) not null,
details varchar(100) not null,
primary key(customer_id)
);

insert into authTable(customer_id,username,pass,position) values (1,'admin','admin','manager');
insert into authTable(username,pass,position) values ('nonadmin','nonadmin','customer');

insert into myTable(customer_id, namee, phone, details) values (1,'Ram chhetri','9076545434','good person');
insert into myTable(namee, phone, details) values ('shyam pokhrel','9054652445','with good humour');
insert into myTable(namee, phone, details) values ('hari prasad','90554562445','strong fellow');
```

★ Database.java


```

import java.sql.Statement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class Database {
    static Statement stm;
    static Connection con;
    static PreparedStatement pStm;

    public Database() {
        try {
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/javadb", "
root", "realGeek1!");
            stm = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, 0);
            pStm = con.prepareStatement("insert into myTable (namee,phone,details)" +
" values(?,?,?)");
        } catch (SQLException e) {
            System.out.println(e);
        }
    }

    public ResultSet get(String query) throws SQLException {
        return stm.executeQuery(query);
    }

    public int size() throws SQLException {
        ResultSet rs = stm.executeQuery("SELECT * FROM myTable");
        int n = 0;
        if(rs != null) {
            rs.last();
            n = rs.getRow();
        }
        return n;
    }

    public void insert(String nm , String phone, String details) throws SQLExcep
tion {
        pStm.setString(1, nm);
        pStm.setString(2, phone);
        pStm.setString(3, details);
        pStm.executeUpdate();
    }

    public void authInsert(String username , String password) throws SQLException
    {
        PreparedStatement authPstm = con.prepareStatement("INSERT INTO authTable(
username,pass,position)" +
        "VALUES(?,?, 'customer')");
        authPstm.setString(1, username);
    }

```

```

        authPstm.setString(2, password);
        authPstm.executeUpdate();
    }

    public void delete(int id) throws SQLException {
        PreparedStatement delStm = con.prepareStatement("DELETE FROM myTable WHERE customer_id = " + id);
        PreparedStatement authDelstm = con.prepareStatement("DELETE FROM authTable WHERE customer_id = " + id);

        delStm.executeUpdate();
        authDelstm.executeUpdate();
    }

    public void update(int id, String name, String phone, String details) throws SQLException {
        PreparedStatement upStm = con.prepareStatement("UPDATE myTable SET name = ?, phone = ?, details = ? WHERE customer_id = ?");
        upStm.setString(1, name);
        upStm.setString(2, phone);
        upStm.setString(3, details);
        upStm.setInt(4, id);
        upStm.executeUpdate();
    }
}

```

GUI Code

★ App.java

```
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.time.LocalDate;
import java.util.Arrays;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPasswordField;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;

public class App {
    JFrame f;

    public App() throws SQLException {
        f = new JFrame("Authetication");
        f.getContentPane().setBackground(Color.black);
        f.setSize(400, 400);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel usrLabel = new JLabel("username:");
        JLabel passLabel = new JLabel("password:");
        JTextField username = new JTextField();
        JPasswordField password = new JPasswordField(8);
        JButton login = new JButton("Login");
        JLabel emptyLabel = new JLabel();
        JButton signUp = new JButton("SignUp");
        signUp.setForeground(Color.white);
        signUp.setBackground(Color.blue);
        signUp.setFont(new Font("Helvetica", Font.PLAIN, 18));

        f.getContentPane().add(signUp);
```

```

usrLabel.setBounds(10, 10, 300, 50);
passLabel.setBounds(10, 90, 200, 50);
username.setBounds(30, 50, 200, 20);
password.setBounds(30, 130, 200, 20);
login.setBounds(10,250,200,20);
signUp.setBounds(10, 300, 200, 20);
emptyLabel.setBounds(10, 340, 300, 30);

usrLabel.setFont(new Font("Verdana",Font.PLAIN,18));
usrLabel.setForeground(Color.white);

passLabel.setFont(new Font("Verdana",Font.PLAIN,18));
passLabel.setForeground(Color.white);

username.setFont(new Font("Verdana",Font.PLAIN,15));
password.setFont(new Font("Verdana",Font.PLAIN,15));

login.setFont(new Font("Verdana", Font.PLAIN,18));
login.setBackground(Color.blue);
login.setForeground(Color.white);

f.add(usrLabel);
f.add(passLabel);
f.add(username);
f.add(password);
f.add(login);
f.add(emptyLabel);

signUp.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        signUpView();
        f.dispose();
        f.setVisible(false);
    }
});

login.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        Database myDB = new Database();

        String inpUsername = username.getText();
        char[] inpPass = password.getPassword();
        String position = "invalid";
        try {
            ResultSet rs = myDB.get("SELECT username, pass, position,customer
_id FROM authTable");

```

```

        while (rs.next()) {
            char[] pass = rs.getString(2).toCharArray();
            if (inpUsername.equals(rs.getString(1)) && Arrays.equals(
inpPass,pass) ) {
                position = rs.getString(3);
                break;
            }
        }
        if(position.equals("manager")) {
            managerView();
        }
        else if(position.equals("customer")) {
            customerView(Integer.parseInt(rs.getString(4)));
        }
        else {
            invalidAuth();
        }
    }
    catch (SQLException e2) {
        e2.printStackTrace();
    }
    f.dispose();
    f.setVisible(false);
}
});
f.setVisible(true);

}

public static void main(String[] args) throws SQLException {

    new App();

}

public static void addView() {
    JFrame f = new JFrame();
    f.setSize(400,400);
    f.setLayout(null);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.getContentPane().setBackground(Color.black);

    JLabel nameLabel = new JLabel("Name: ");
    JLabel numLabel = new JLabel("Number: ");
    JLabel aboutLabel = new JLabel("About: ");

    nameLabel.setForeground(Color.white);
    numLabel.setForeground(Color.white);
    aboutLabel.setForeground(Color.white);

```

```

JTextField name = new JTextField();
JTextField num = new JTextField();
JTextField about = new JTextField();
JButton done = new JButton("Done");
done.setBackground(Color.blue);
done.setForeground(Color.white);
JButton cancel = new JButton("Cancel");
cancel.setBackground(Color.blue);
cancel.setForeground(Color.white);
done.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        String inpName = name.getText();
        String inpNum = num.getText();
        String inpAbout = about.getText();

        Database myDB = new Database();
        try {
            myDB.insert(inpName, inpNum, inpAbout);
        } catch (SQLException e1) {
            e1.printStackTrace();
        }
        managerView();
        f.dispose();
        f.setVisible(false);
    }

});

cancel.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        managerView();
        f.dispose();
        f.setVisible(false);
    }

});

nameLabel.setBounds(10,10,300,30);
name.setBounds(20,50,300,30);
numLabel.setBounds(10,90,300,30);
num.setBounds(10,130,300,30);
aboutLabel.setBounds(10,170,300,30);
about.setBounds(10,210,300,30);
done.setBounds(30,250,100,30);
cancel.setBounds(180,250,100,30);

f.add(nameLabel);

```

```

        f.add(name);
        f.add(numLabel);
        f.add(num);
        f.add(aboutLabel);
        f.add(about);
        f.add(done);
        f.add(cancel);

        f.setVisible(true);
    }

    public static void managerView() {
        JFrame f = new JFrame();
        f.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        f.setSize(400,400);
        f.setTitle("Manager");
        f.getContentPane().setBackground(Color.black);

        JButton add = new JButton("Add customer");
        add.setBackground(Color.blue);
        add.setForeground(Color.white);
        add.addActionListener((ActionListener) new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                addView();
                f.dispose();
                f.setVisible(false);
            }
        });

        JButton back = new JButton("Back");
        back.setBackground(Color.blue);
        back.setForeground(Color.white);
        back.addActionListener((ActionListener) new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    new App();
                } catch (SQLException e1) {
                    e1.printStackTrace();
                }
                f.dispose();
                f.setVisible(false);
            }
        });

        JButton accessCustomer = new JButton("Access Customer");
        accessCustomer.setBackground(Color.blue);
        accessCustomer.setForeground(Color.white);
    }

```

```

        accessCustomer.addActionListener((ActionListener) new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                askNumView();
                f.dispose();
                f.setVisible(false);
            }
        });

        JButton listCustomers = new JButton("List all the customers");
        listCustomers.setBackground(Color.blue);
        listCustomers.setForeground(Color.white);

        listCustomers.addActionListener((ActionListener) new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    listView();
                } catch (SQLException e1) {
                    e1.printStackTrace();
                }
                f.dispose();
                f.setVisible(false);
            }
        });

        JLabel free = new JLabel();

        add.setBounds(10,10,300,30);
        accessCustomer.setBounds(10,90,300,30);
        listCustomers.setBounds(10,170,300,30);
        back.setBounds(10,250,300,30);

        free.setBounds(10,290,300,30);

        f.add(add);
        f.add(accessCustomer);
        f.add(listCustomers);
        f.add(back);
        f.add(free);

        f.setVisible(true);
    }

    public static void askNumView() {
        JFrame f= new JFrame();
    }

```



```

f.setSize(400,400);
f.setLayout(null);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.getContentPane().setBackground(Color.black);

JLabel askNumLabel = new JLabel("Enter the customer id to get access");
askNumLabel.setForeground(Color.white);
JTextField askId = new JTextField();
JButton submit = new JButton("Submit");
submit.setBackground(Color.blue);
submit.setForeground(Color.white);
JButton back = new JButton("Back");
back.setBackground(Color.blue);
back.setForeground(Color.white);
back.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        managerView();
        f.dispose();
        f.setVisible(false);
    }

});

submit.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        int inpId = Integer.parseInt(askId.getText());

        managerUserView(inpId);
        f.dispose();
        f.setVisible(false);
    }

});

askNumLabel.setBounds(30,70,200,50);
askId.setBounds(10,160,100,30);
submit.setBounds(10,200,100,30);
back.setBounds(120,200,100,30);

f.add(askId);
f.add(askNumLabel);
f.add(submit);
f.add(back);

f.setVisible(true);

}

public static void managerUserView(int id) {

```

```

JFrame f = new JFrame("Customer with id " + id);
f.setSize(400,400);
f.setLayout(null);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.getContentPane().setBackground(Color.black);

JButton nextRec = new JButton("Next Recharge Date");
nextRec.setBackground(Color.blue);
nextRec.setForeground(Color.white);
JButton prepost = new JButton("Prepaid/Postpaid");
prepost.setBackground(Color.blue);
prepost.setForeground(Color.white);
JButton details = new JButton("account detail");
details.setBackground(Color.blue);
details.setForeground(Color.white);
JButton balance = new JButton("Balance");
balance.setBackground(Color.blue);
balance.setForeground(Color.white);
JButton remove = new JButton("Remove User");
remove.setBackground(Color.blue);
remove.setForeground(Color.white);
JButton back = new JButton("Back");
back.setBackground(Color.blue);
back.setForeground(Color.white);
JButton update = new JButton("Update");
update.setBackground(Color.blue);
update.setForeground(Color.white);

details.setBounds(10,10,300,30);
nextRec.setBounds(10,60,300,30);
prepost.setBounds(10,110,300,30);
balance.setBounds(10,160,300,30);
remove.setBounds(10,210,300,30);
update.setBounds(10, 260, 300, 30);
back.setBounds(10, 310, 300, 30);

f.add(prepost);
f.add(details);
f.add(nextRec);
f.add(balance);
f.add(remove);
f.add(back);
f.add(update);

update.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {

        try {
            updateView(id,"managerUserView");

```

```

        } catch (SQLException e1) {
            e1.printStackTrace();
        }
        f.dispose();
        f.setVisible(false);
    }
});

back.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        askNumView();
        f.dispose();
        f.setVisible(false);
    }
});

remove.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        Database myDB = new Database();
        try {
            myDB.delete(id);
        } catch (SQLException e1) {
            e1.printStackTrace();
        }
        askNumView();
        f.dispose();
        f.setVisible(false);
    }
});

nextRec.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        nextRecView(id,"mangerUserView");
        f.dispose();
        f.setVisible(false);
    }
});

prepost.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        prepost.setText(Math.random()>0.5? "Prepaid" : "Postpaid");
        prepost.setEnabled(false);
    }
});

details.addActionListener((ActionListener) new ActionListener() {

```

```

        @Override
        public void actionPerformed(ActionEvent e) {
            try {
                detailView(id,"managerUserView");
                f.dispose();
                f.setVisible(false);
            } catch (SQLException e1) {
                e1.printStackTrace();
            }
        }
    });
    balance.addActionListener((ActionListener) new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {

            balanceView(id,"managerUserView");
            f.dispose();
            f.setVisible(false);

        }
    });
    f.setVisible(true);
}

public static void nextRecView(int id, String string) {
    JFrame f = new JFrame();
    f.setSize(400, 400);
    f.setLayout(null);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.getContentPane().setBackground(Color.black);

    JLabel nextDate = new JLabel();
    nextDate.setForeground(Color.white);
    nextDate.setFont(new Font("Helvetica",Font.PLAIN,16));
    String s = LocalDate.now().plusDays(28).toString();
    nextDate.setText("<html>" + s + " is the next recharge date</html>");
    JButton back = new JButton("Back");
    back.setForeground(Color.white);
    back.setBackground(Color.blue);
    back.setBounds(10, 230, 300, 30);
    nextDate.setBounds(10, 10, 300, 200);

    back.addActionListener((ActionListener) new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
            if(string.equals("managerUserView")) managerUserView(id); else custom
erView(id);
            f.dispose();
            f.setVisible(false);
        }
    });
}

```

```

        }
    });
    f.add(nextDate);
    f.add(back);

    f.setVisible(true);
}

public void invalidAuth() {
    JFrame f = new JFrame("invalid credentials");
    f.setSize(400, 400);
    f.setLayout(null);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.getContentPane().setBackground(Color.black);
    JLabel invalid = new JLabel("<html>Either the password or username or both are incorrect , please check and try again</html>");
    invalid.setForeground(Color.white);
    invalid.setBounds(10, 50, 300, 100);

    JButton back = new JButton("Back");
    back.setForeground(Color.white);
    back.setBackground(Color.blue);
    back.setBounds(10, 200, 300, 30);

    back.addActionListener((ActionListener) new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
            try {
                new App();
            } catch (SQLException e1) {
                e1.printStackTrace();
            }
            f.dispose();
            f.setVisible(false);
        }
    });
    f.add(invalid);
    f.setVisible(true);
    f.add(back);
}

public static void customerView(int id) {
    JFrame f = new JFrame("CustomerView with id " + id);

    f.setSize(400, 400);
    f.setLayout(null);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.getContentPane().setBackground(Color.black);

```

```

JButton nextRec = new JButton("Next Recharge Date");
nextRec.setBackground(Color.blue);
nextRec.setForeground(Color.white);
JButton prepost = new JButton("Prepaid/Postpaid");
prepost.setBackground(Color.blue);
prepost.setForeground(Color.white);
JButton details = new JButton("account detail");
details.setBackground(Color.blue);
details.setForeground(Color.white);
JButton balance = new JButton("Balance");
balance.setBackground(Color.blue);
balance.setForeground(Color.white);
JButton back = new JButton("Back");
back.setBackground(Color.blue);
back.setForeground(Color.white);
JButton update = new JButton("Update");
update.setBackground(Color.blue);
update.setForeground(Color.white);

details.setBounds(10, 10, 300, 30);
nextRec.setBounds(10, 60, 300, 30);
prepost.setBounds(10, 110, 300, 30);
balance.setBounds(10, 160, 300, 30);
update.setBounds(10, 210, 300, 30);
back.setBounds(10, 260, 300, 30);

f.add(prepost);
f.add(details);
f.add(nextRec);
f.add(balance);
f.add(back);
f.add(update);

update.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {

        try {
            updateView(id,"customerView");
        } catch (SQLException e1) {
            e1.printStackTrace();
        }
        f.dispose();
        f.setVisible(false);
    }
});

back.addActionListener((ActionListener) new ActionListener() {

    @Override

```

```

        public void actionPerformed(ActionEvent e) {
            try {
                new App();
            } catch (SQLException e1) {
                e1.printStackTrace();
            }
            f.dispose();
            f.setVisible(false);
        }
    });

    nextRec.addActionListener((ActionListener) new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
            nextRecView(id,"customerView");
            f.dispose();
            f.setVisible(false);
        }
    });

    prepost.addActionListener((ActionListener) new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
            prepost.setText(Math.random() > 0.5 ? "Prepaid" : "Postpaid");
            prepost.setEnabled(false); // here
        }
    });

    details.addActionListener((ActionListener) new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
            try {
                detailView(id,"customerView");
                f.dispose();
                f.setVisible(false);
            } catch (SQLException e1) {
                e1.printStackTrace();
            }
        }
    });

    balance.addActionListener((ActionListener) new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {

            balanceView(id,"customerView");
            f.dispose();
            f.setVisible(false);

        }
    });

```

```

        f.setVisible(true);
    }

    public static void listView() throws SQLException {
        JFrame f = new JFrame("ListView");
        f.setSize(400, 400);
        f.setLayout(null);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.getContentPane().setBackground(Color.black);

        Database myDB = new Database();
        int n = myDB.size();
        String[][] data = new String[n][n];

        String[] cols = { "Name", "Phone", "Details" };
        ResultSet rs = myDB.get("SELECT namee, phone, details FROM myTable");
        int i = 0;
        while (i < n) {
            rs.next();
            data[i][0] = rs.getString(1);
            data[i][1] = rs.getString(2);
            data[i][2] = rs.getString(3);
            i++;
        }
        JTable table = new JTable(data, cols);
        table.setBounds(10, 10, 380, 300);

        JScrollPane sp = new JScrollPane(table);
        sp.setBounds(15, 15, 380, 300);

        JButton back = new JButton("Back");
        back.setBackground(Color.blue);
        back.setForeground(Color.white);
        back.setBounds(15, 330, 380, 30);
        JLabel empty = new JLabel();
        empty.setBounds(30, 300, 300, 30);

        back.addActionListener((ActionListener) new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                managerView();
                f.dispose();
                f.setVisible(false);
            }
        });

        f.add(sp);
        f.add(empty);
        f.add(back);
        f.setVisible(true);
    }

```



```

}

public static void detailView(int id, String string) throws SQLException {
    JFrame f = new JFrame();
    f.setSize(400, 400);
    f.setLayout(null);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    JLabel detail = new JLabel();
    detail.setText(
        "<html><h2>Details</h2>This field should be filled with some of the d
etailed information of the customer.This field should be filled.</html>";
        + "</html>");
    JButton back = new JButton("Back");

    f.getContentPane().setBackground(Color.black);
    detail.setForeground(Color.white);
    detail.setFont(new Font("Helvetica", Font.PLAIN, 16));
    detail.setBounds(10, 10, 320, 300);

    back.setBackground(Color.blue);
    back.setForeground(Color.white);
    back.setFont(new Font("Helvetica", Font.PLAIN, 16));
    back.setBounds(10, 320, 320, 30);

    back.addActionListener((ActionListener) new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
            System.out.println("reached 6");

            if(string.equals("managerUserView")) managerUserView(id); else custom
erView(id);
            f.dispose();
            f.setVisible(false);
        }
    });
    f.add(detail);
    f.add(back);

    f.setVisible(true);
}

public static void balanceView(int id, String string) {
    JFrame f = new JFrame();
    f.setSize(400, 400);
    f.setLayout(null);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    double balance = Math.random() * 500;
    JLabel bal = new JLabel();
    bal.setText("<html><h2>Balance</h2> you have total balance of " + balance + "
</html>");
}

```

```

JButton back = new JButton("Back");
JButton recharge = new JButton("Recharge");

f.getContentPane().setBackground(Color.black);
bal.setForeground(Color.white);
bal.setFont(new Font("Helvetica", Font.PLAIN, 16));
bal.setBounds(10, 10, 300, 280);
recharge.setBounds(10, 300, 320, 30);

recharge.setBackground(Color.blue);
recharge.setForeground(Color.white);
back.setBackground(Color.blue);
back.setForeground(Color.white);
back.setFont(new Font("Helvetica", Font.PLAIN, 16));
back.setBounds(10, 340, 320, 30);

recharge.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        rechargeView(id);
        f.dispose();
        f.setVisible(false);
    }
});

back.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {

        if(string.equals("managerUserView")) managerUserView(id); else customerV
iew(id);
        f.dispose();
        f.setVisible(false);
    }
});
f.add(bal);
f.add(back);
if(string.equals("customerView")) f.add(recharge);

f.setVisible(true);
}

public static void updateView(int id, String string) throws SQLException {
    JFrame f = new JFrame();
    f.setSize(400,400);
    f.setLayout(null);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.getContentPane().setBackground(Color.black);

```

```

JLabel nameLabel = new JLabel("Name: ");
JLabel numLabel = new JLabel("Number: ");
JLabel aboutLabel = new JLabel("About: ");

nameLabel.setForeground(Color.white);
numLabel.setForeground(Color.white);
aboutLabel.setForeground(Color.white);

Database myDb = new Database();
ResultSet rs = myDb.get("SELECT namee,phone,details FROM myTable WHERE custom
er_id = " + id);
rs.next();

JTextField name = new JTextField(rs.getString(1));
JTextField num = new JTextField(rs.getString(2));
JTextField about = new JTextField(rs.getString(3));
JButton done = new JButton("Done");
done.setBackground(Color.blue);
done.setForeground(Color.white);
JButton cancel = new JButton("Cancel");
cancel.setBackground(Color.blue);
cancel.setForeground(Color.white);

done.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        String inpName = name.getText();
        String inpNum = num.getText();
        String inpAbout = about.getText();

        Database myDB = new Database();
        try {
            myDB.update(id,inpName, inpNum, inpAbout);
        } catch (SQLException e1) {
            e1.printStackTrace();
        }
        if(string.equals("managerUserView")) managerUserView(id); else custom
erView(id);
        f.dispose();
        f.setVisible(false);
    }

});

cancel.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        if(string.equals("managerUserView")) managerUserView(id); else custom
erView(id);

```

```

        f.dispose();
        f.setVisible(false);
    }

});

nameLabel.setBounds(10,10,300,30);
name.setBounds(20,50,300,30);
numLabel.setBounds(10,90,300,30);
num.setBounds(10,130,300,30);
aboutLabel.setBounds(10,170,300,30);
about.setBounds(10,210,300,30);
done.setBounds(30,250,100,30);
cancel.setBounds(180,250,100,30);

f.add(nameLabel);
f.add(name);
f.add(numLabel);
f.add(num);
f.add(aboutLabel);
f.add(about);
f.add(done);
f.add(cancel);

f.setVisible(true);
}

public static void signUpView() {
    JFrame f = new JFrame();
    f.setSize(400,400);
    f.setLayout(null);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.getContentPane().setBackground(Color.black);

    JLabel nameLabel = new JLabel("Name: ");
    JLabel numLabel = new JLabel("Number: ");
    JLabel aboutLabel = new JLabel("About: ");
    JLabel usernameLabel = new JLabel("Username: ");
    JLabel passwordLabel = new JLabel("Password: ");
    JLabel empty = new JLabel();

    nameLabel.setForeground(Color.white);
    numLabel.setForeground(Color.white);
    aboutLabel.setForeground(Color.white);
    usernameLabel.setForeground(Color.white);
    passwordLabel.setForeground(Color.white);

    JTextField name = new JTextField();

```

```

JTextField num = new JTextField();
JTextField about = new JTextField();
JTextField username = new JTextField();
JPasswordField password = new JPasswordField();

JButton create = new JButton("Create");
create.setBackground(Color.blue);
create.setForeground(Color.white);
JButton cancel = new JButton("Cancel");
cancel.setBackground(Color.blue);
cancel.setForeground(Color.white);

create.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        String inpName = name.getText();
        String inpNum = num.getText();
        String inpAbout = about.getText();
        String inpUsername = username.getText();
        String inpPassword = String.valueOf(password.getPassword());

        Database myDB = new Database();
        try {
            myDB.insert(inpName, inpNum, inpAbout);
            myDB.authInsert(inpUsername, inpPassword);
            new App();
            f.dispose();
            f.setVisible(false);
        } catch (SQLException e1) {
            e1.printStackTrace();
        }
    }

});

cancel.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            new App();
            f.dispose();
            f.setVisible(false);
        } catch (SQLException e1) {
            e1.printStackTrace();
        }
    }

});

nameLabel.setBounds(10,5,250,20);

```

```

name.setBounds(20,30,250,25);
numLabel.setBounds(10,60,250,20);
num.setBounds(10,85,250,25);
aboutLabel.setBounds(10,115,250,20);
about.setBounds(10,140,250,25);
usernameLabel.setBounds(10,170,250,20);
username.setBounds(10,195,250,25);
passwordLabel.setBounds(10,225,250,20);
password.setBounds(10,250,250,25);

create.setBounds(30,300,100,30);
cancel.setBounds(180,300,100,30);
empty.setBounds(30,340,1000,30);

f.add(empty);
f.add(nameLabel);
f.add(name);
f.add(numLabel);
f.add(num);
f.add(aboutLabel);
f.add(about);
f.add(create);
f.add(cancel);
f.add(username);
f.add(usernameLabel);
f.add(password);
f.add(passwordLabel);

f.setVisible(true);
}

public static void rechargeView(int id) {
    JFrame f = new JFrame();
    f.setSize(400, 400);
    f.setLayout(null);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    // double balance = Math.random() * 500;
    JLabel regLabel = new JLabel("Recharge Amount");
    JButton back = new JButton("Back");
    JButton recharge = new JButton("Recharge");
    JTextField amount = new JTextField();
    f.getContentPane().setBackground(Color.black);
    regLabel.setForeground(Color.white);
    regLabel.setFont(new Font("Helvetica", Font.PLAIN, 16));
    regLabel.setBounds(10, 10, 300, 30);
    amount.setBounds(10,60,300,30);
    recharge.setBounds(10,120,100,30);
    back.setBounds(10, 230, 100, 30);

    recharge.setBackground(Color.blue);

```

```
recharge.setForeground(Color.white);
back.setBackground(Color.blue);
back.setForeground(Color.white);
back.setFont(new Font("Helvetica", Font.PLAIN, 16));

recharge.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        customerView(id);
        f.dispose();
        f.setVisible(false);
    }
});

back.addActionListener((ActionListener) new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {

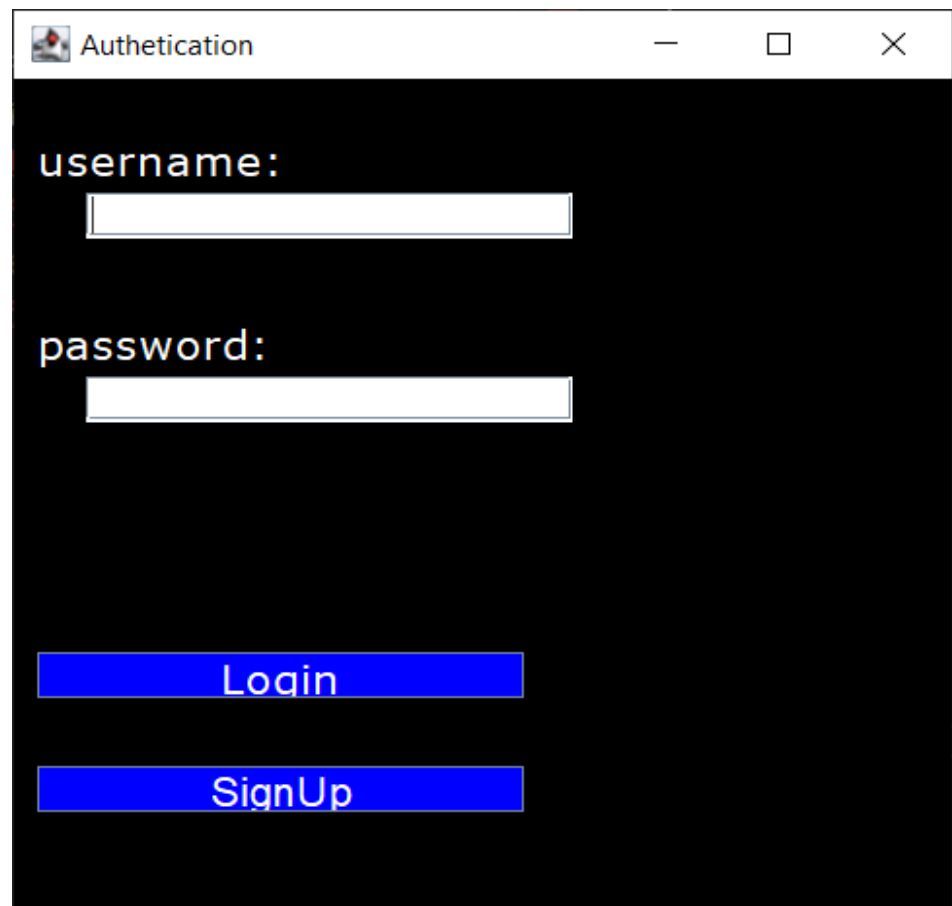
        customerView(id);
        f.dispose();
        f.setVisible(false);
    }
});
f.add(regLabel);
f.add(back);
f.add(recharge);
f.add(amount);

f.setVisible(true);
}
}
```


Snapshots of Input and Output

Server-Side

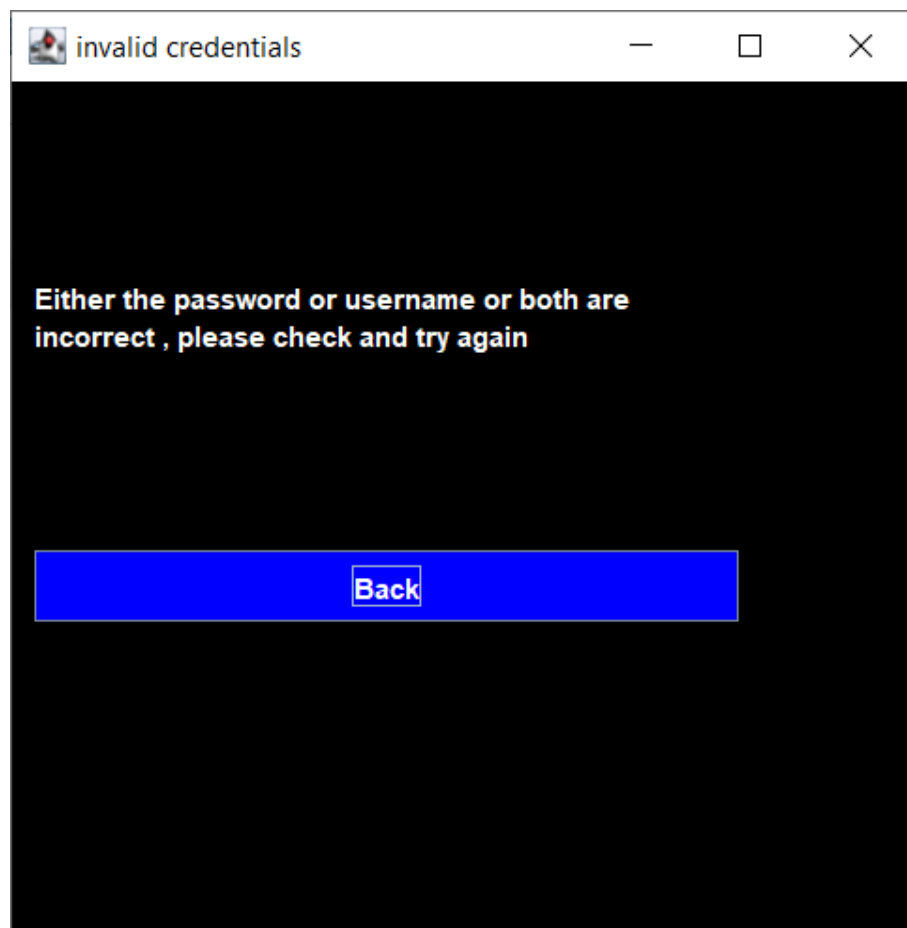
Snapshot of the first login page. (SNAP 1)



The image shows a web browser window titled "Authentication". The background is black. There are two white input fields. The first is labeled "username:" and the second is labeled "password:". Below the password field, there are two blue buttons with white text: "Login" and "SignUp".

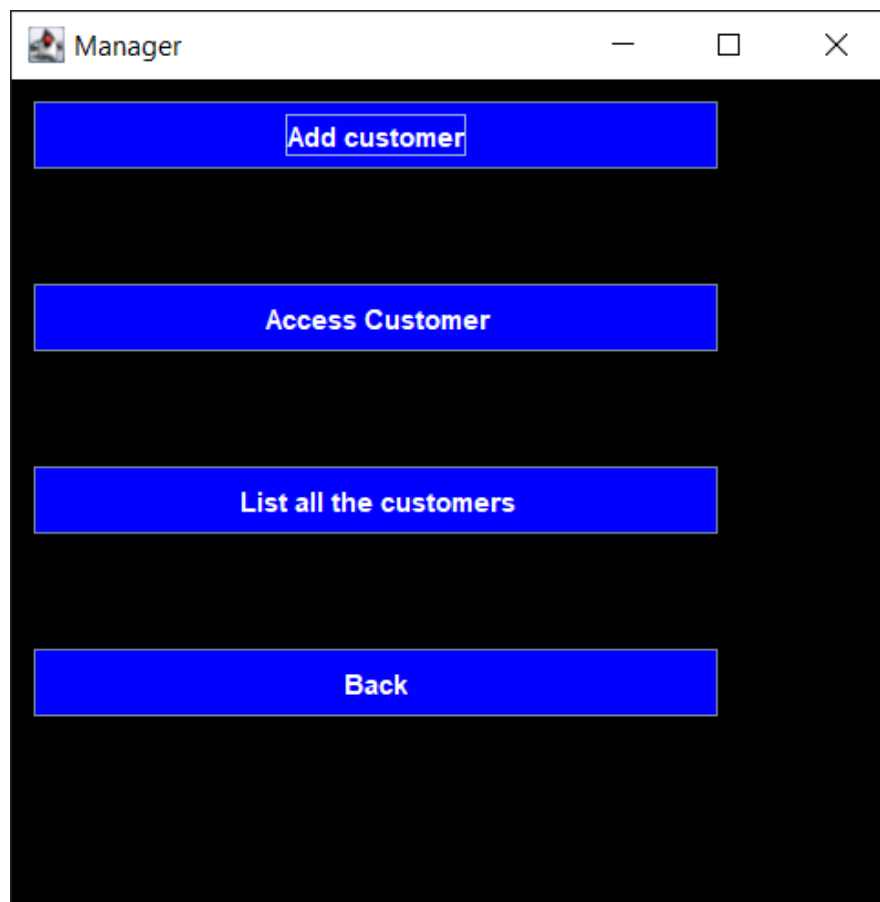
1. After Clicking => Login : if id, password is incorrect -> (SNAP 2)
: if id, password is of manger -> (SNAP 3)
:if id, password is of customer -> (SNAP 4)
2. After Clicking => SignUp -> (SNAP 5)

Snapshot of bad authentication(SNAP 2) :



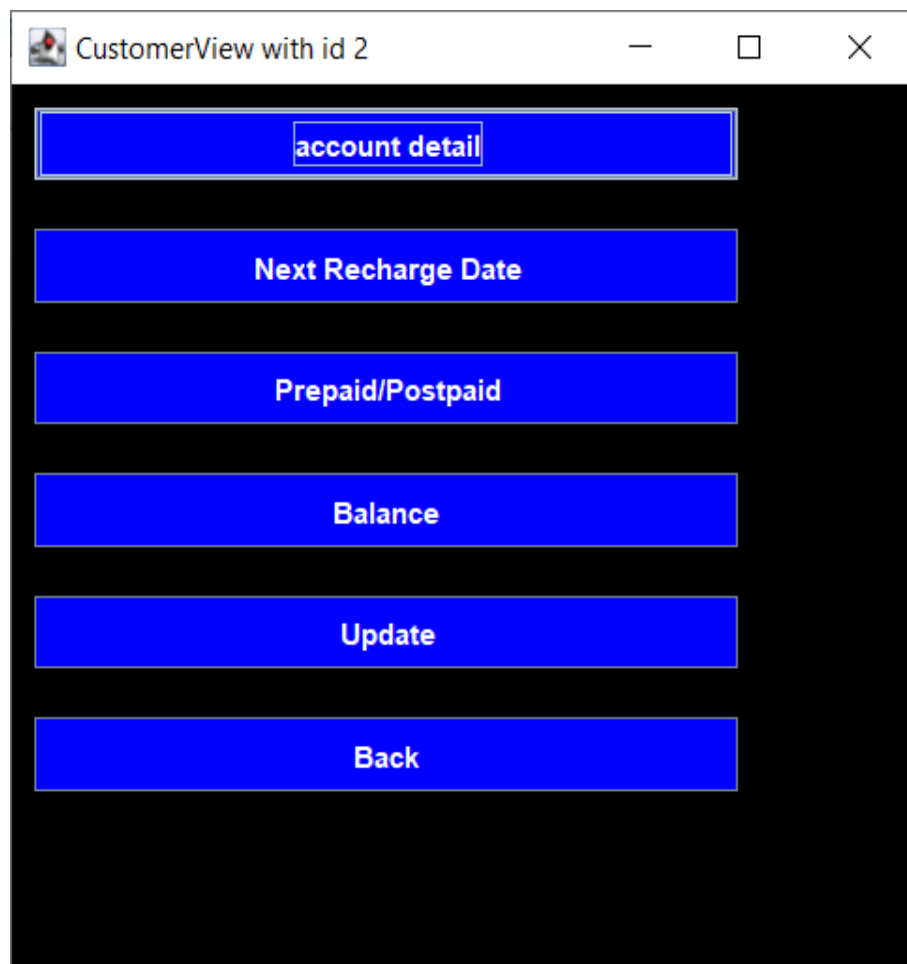
1. After Clicking => Back -> SNAP 1

Snapshot of manager View (SNAP 3):



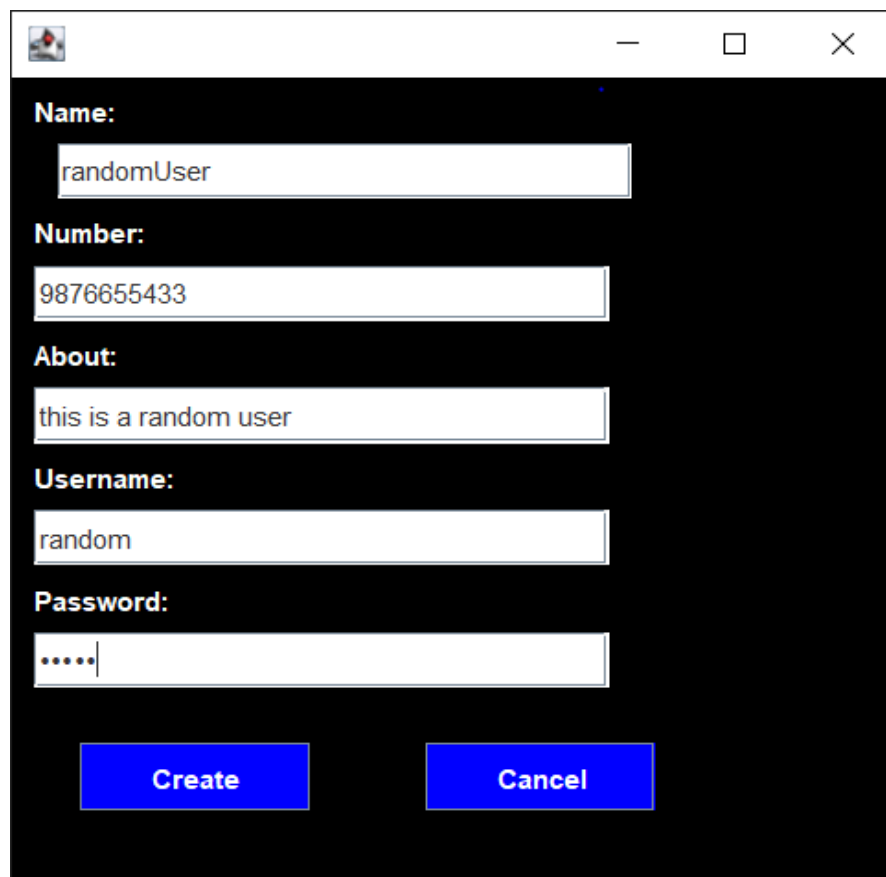
1. After Clicking => Add customer -> SNAP 6
2. After Clicking => Access Customer-> SNAP 7
3. After Clicking => List all the customers -> SNAP 8
4. After Clicking => Back -> SNAP 1

Snapshot of customer View(SNAP 4) :



1. After Clicking => account detail -> SNAP 9
2. After Clicking => Next Recharge Date -> SNAP 10
3. After Clicking => Prepaid/Postpaid -> SNAP 11
4. After Clicking => Balance -> SNAP 12
5. After Clicking => Update -> SNAP 13
6. After Clicking => back -> SNAP 1

Snapshot of signUp View (SNAP 5):



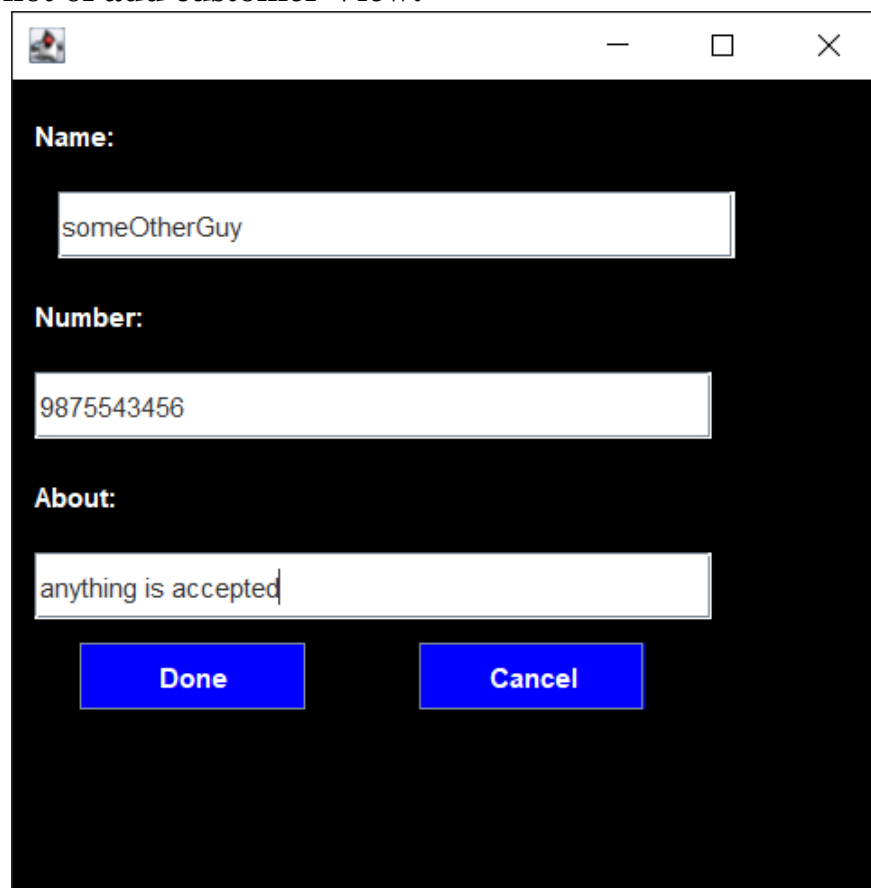
A screenshot of a user creation dialog box. The dialog has a title bar with a small icon, a minus button, a maximize button, and a close button. The background is black. The form contains the following fields and labels:

- Name:** A text input field containing "randomUser".
- Number:** A text input field containing "9876655433".
- About:** A text input field containing "this is a random user".
- Username:** A text input field containing "random".
- Password:** A password input field with five dots and a cursor.

At the bottom, there are two blue buttons: "Create" and "Cancel".

1. After Clicking => create -> SNAP 1
2. After Clicking => cancel -> SNAP 1

Snapshot of add customer View:



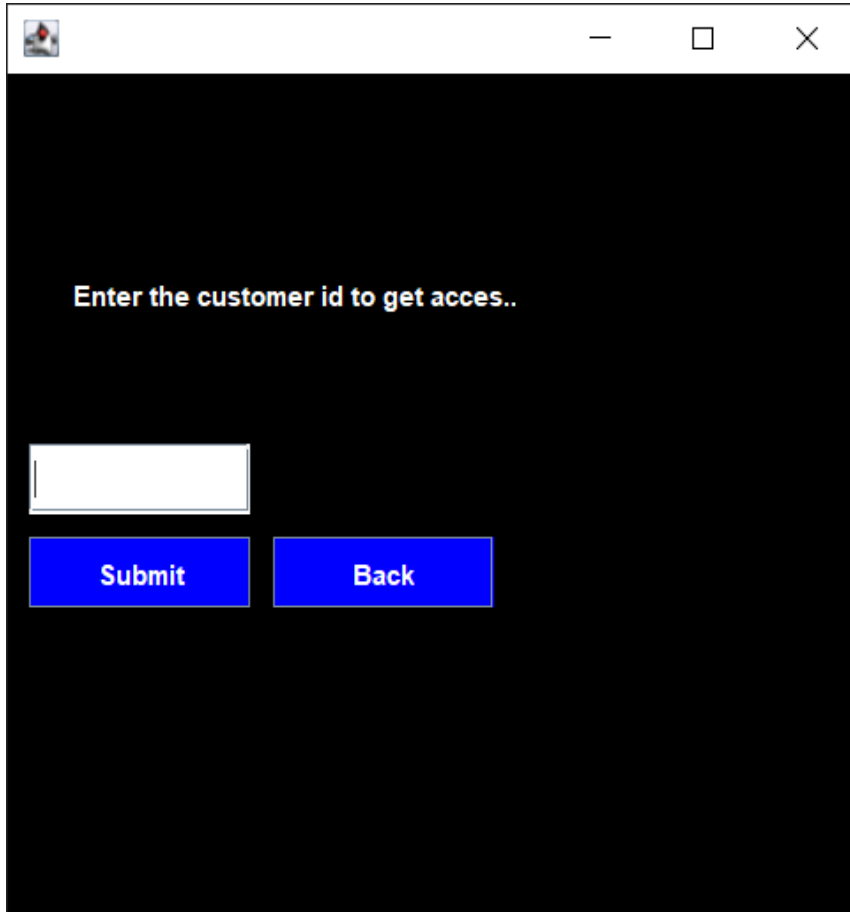
A screenshot of an add customer dialog box. The dialog has a title bar with a small icon, a minus button, a maximize button, and a close button. The background is black. The form contains the following fields and labels:

- Name:** A text input field containing "someOtherGuy".
- Number:** A text input field containing "9875543456".
- About:** A text input field containing "anything is accepted".

At the bottom, there are two blue buttons: "Done" and "Cancel".

1. Clicking => Done -> snap 3
2. Clicking => Cancel -> snap 3

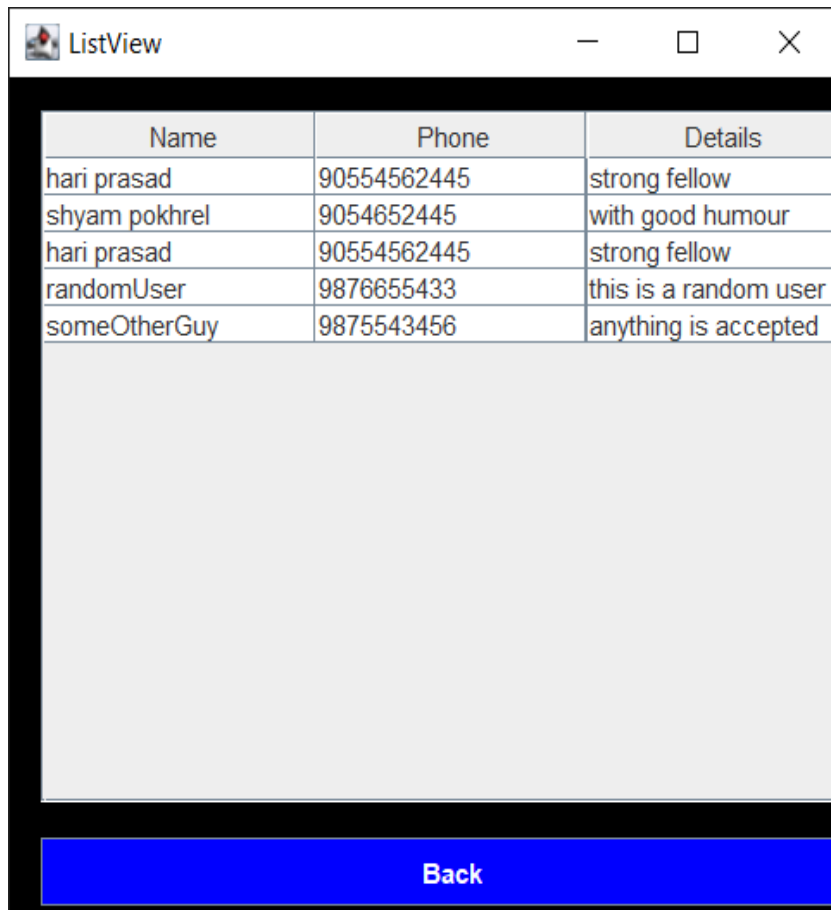
Snapshot of access Customer (SNAP 7) :



The screenshot shows a web application window with a black background. At the top, there is a white header bar containing a small icon on the left and three window control buttons (minimize, maximize, close) on the right. Below the header, the text "Enter the customer id to get acces.." is displayed in a white, sans-serif font. Underneath the text, there is a white rectangular text input field. Below the input field, there are two blue rectangular buttons with white text: "Submit" on the left and "Back" on the right.

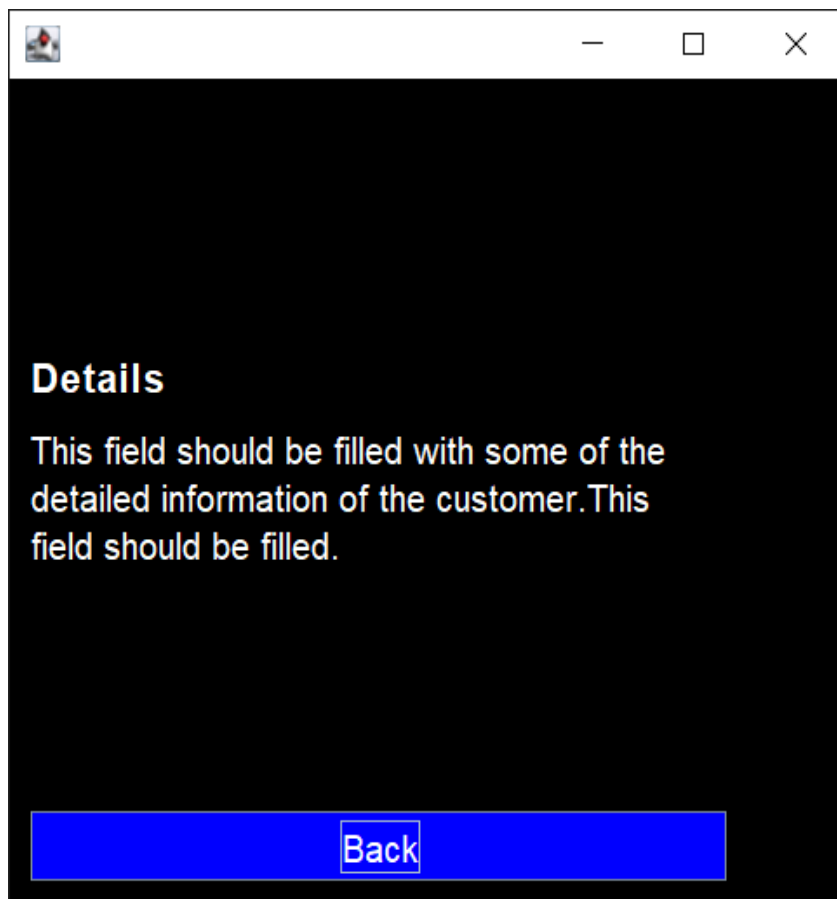
1. Clicking => Submit -> SNAP 14
2. Clicking => Back -> SNAP 3

Snapshot of list customer view (SNAP 8) :



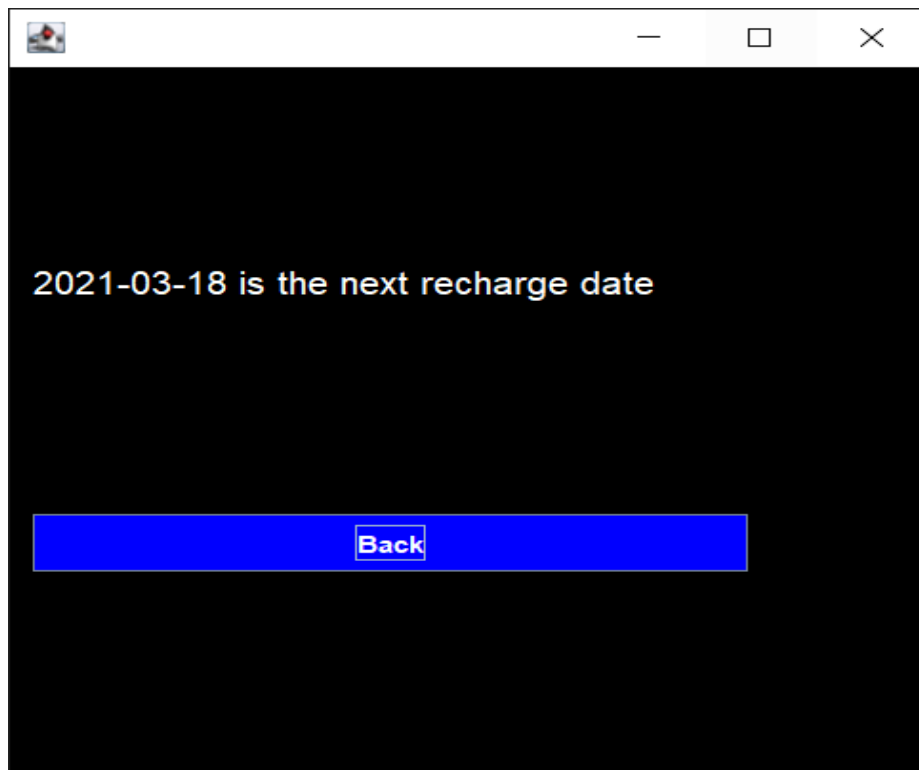
1. Clicking => Back -> SNAP 3

Snapshot of account detail view (SNAP 9) :



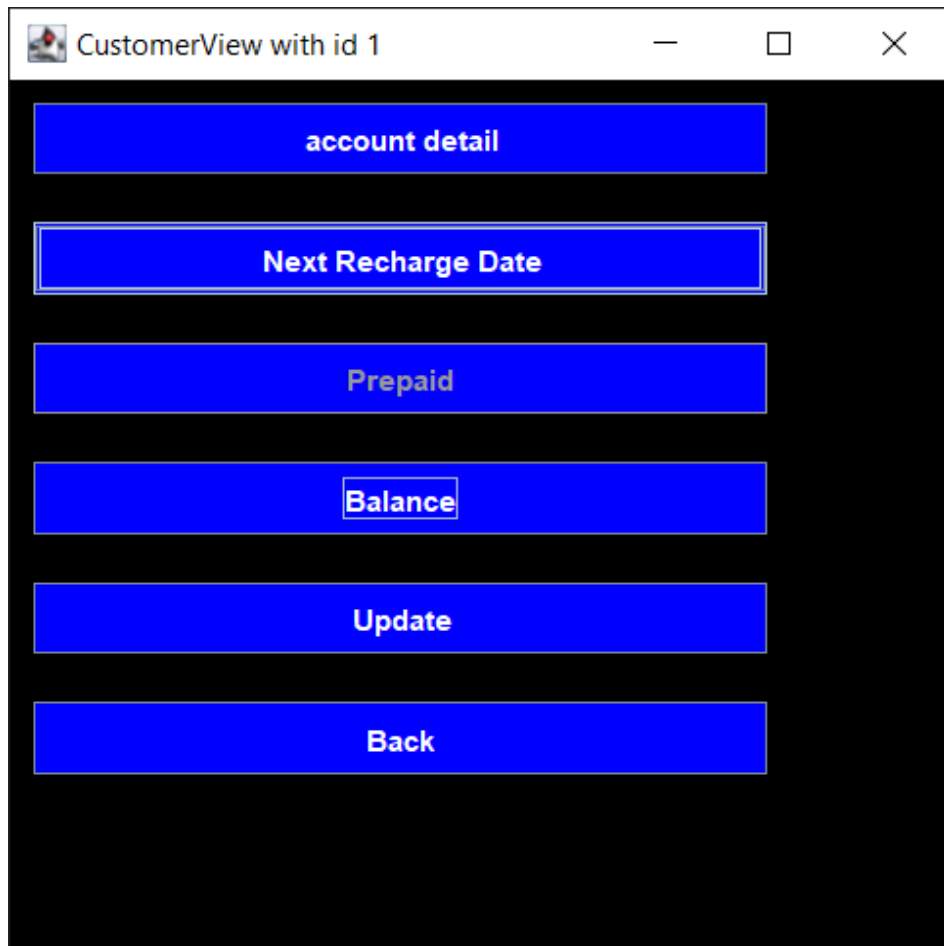
1. Clicking => back -> SNAP 4

Snapshot of next recharge view (SNAP 10):

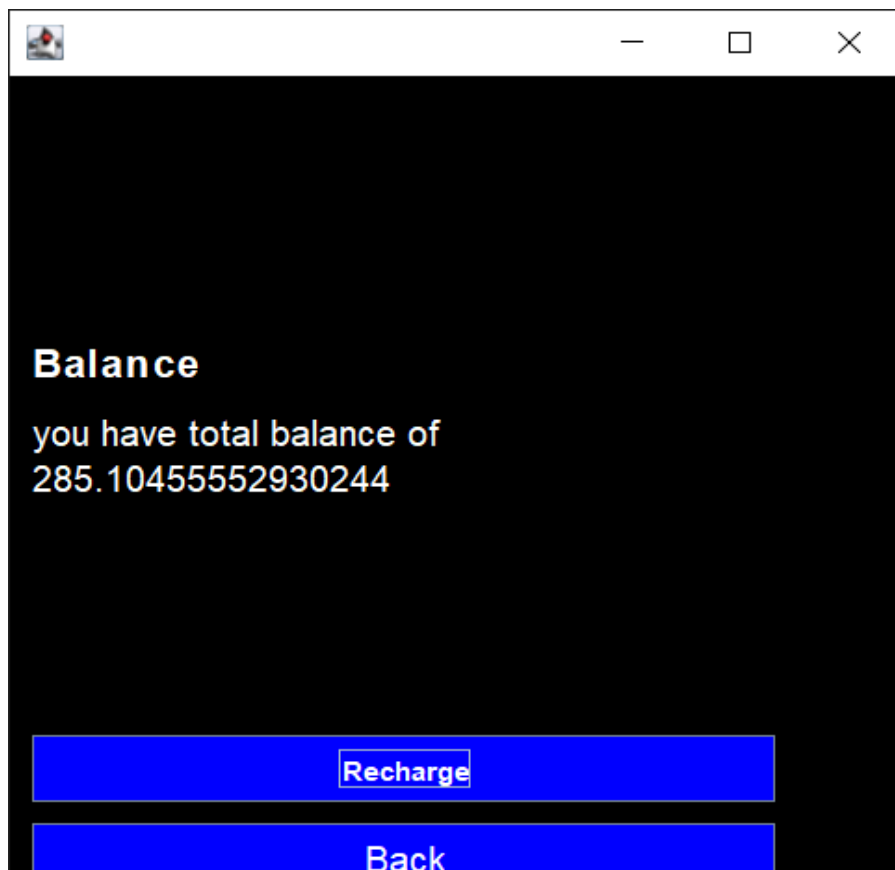


1. Clicking => Back -> SNAP 4

Snapshot of prepaid/postpaid (SNAP 11) :

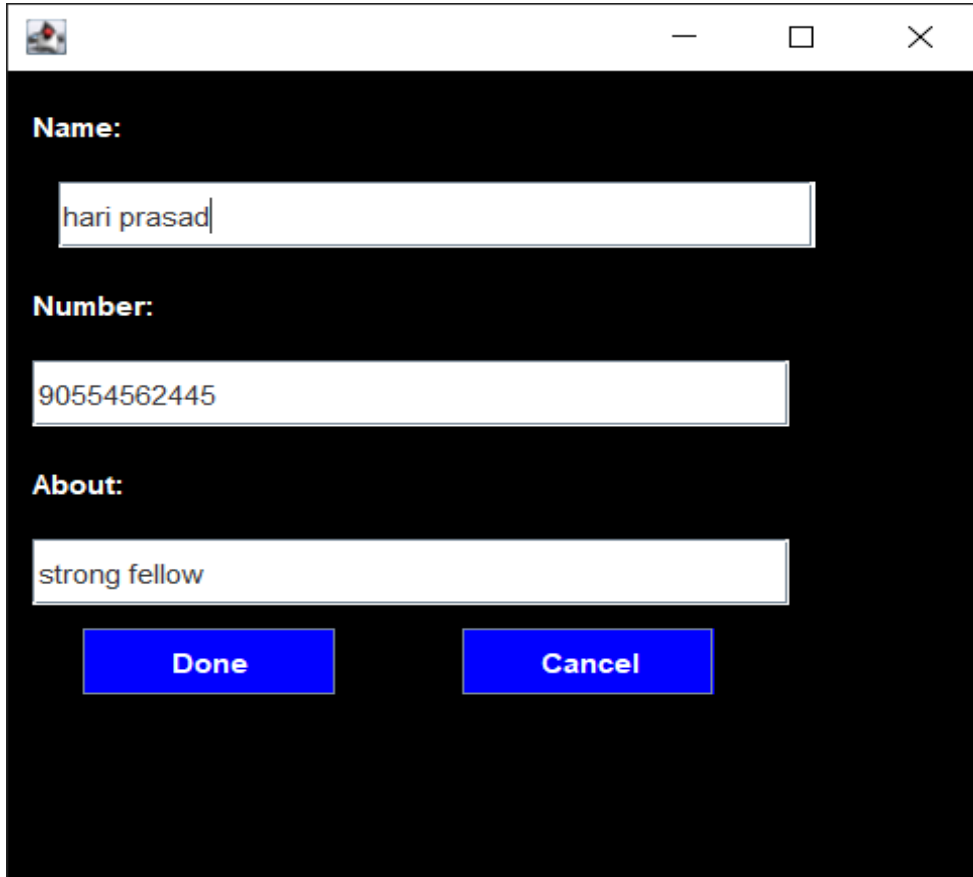


Snapshot of balance View (SNAP 12):



1. Clicking => Recharge -> SNAP 15
2. Clicking => Back -> SNAP 4

Snapshot of update View (SNAP 13):



Name:

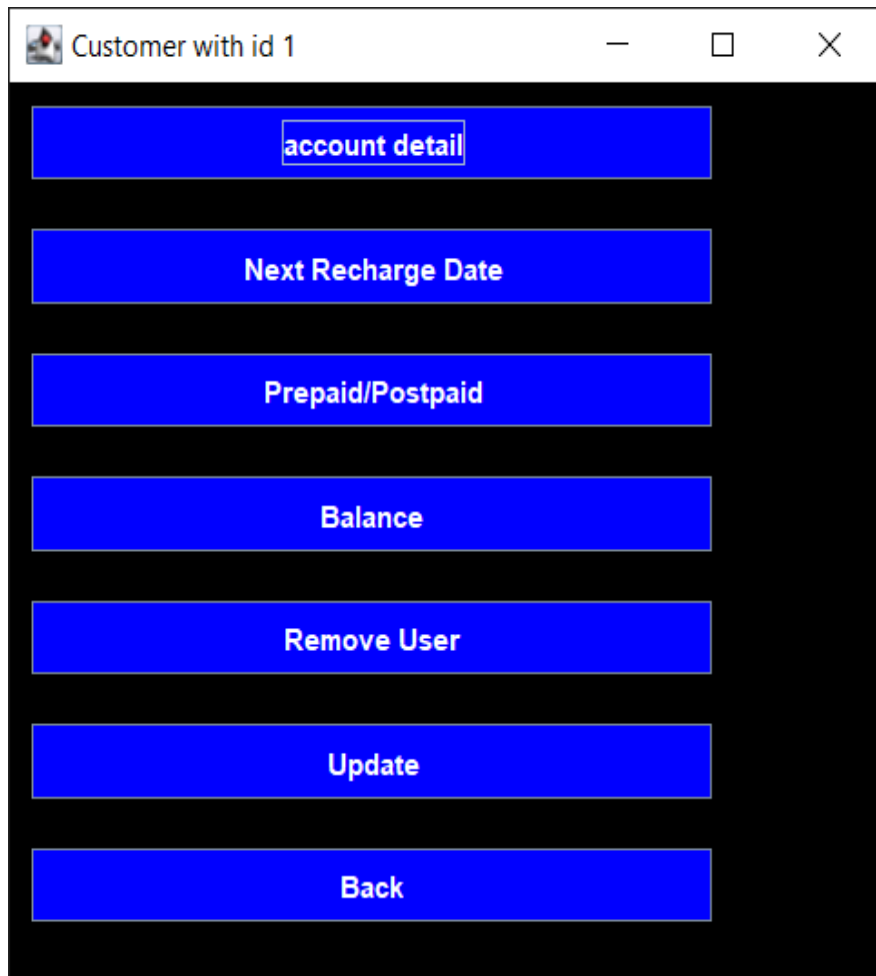
Number:

About:

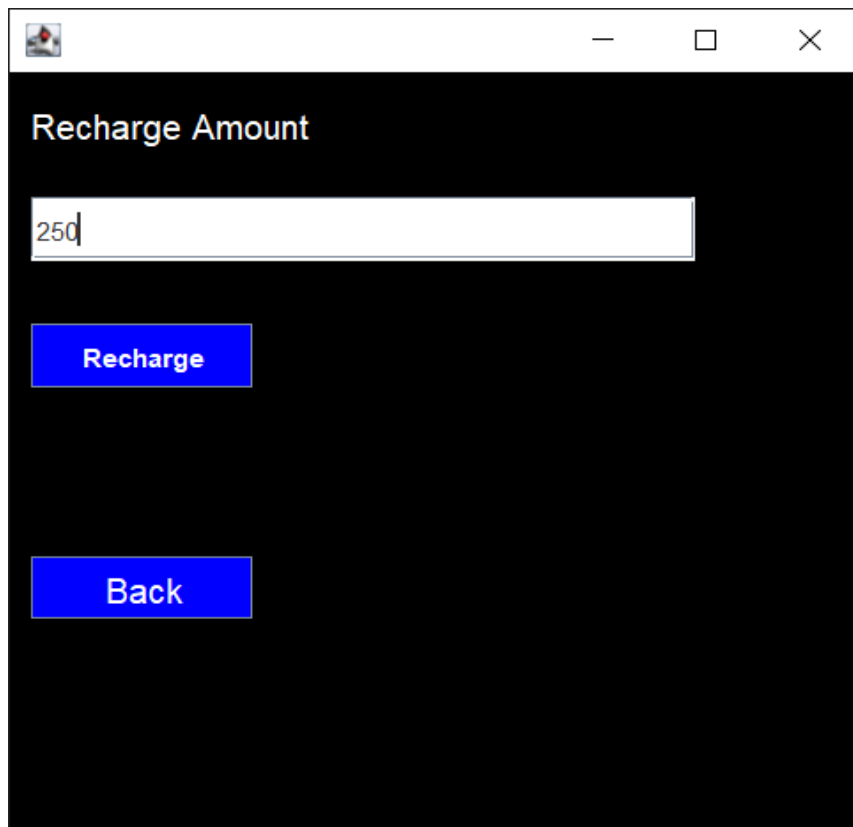
Done **Cancel**

1. Clicking => Done -> SNAP 4
2. Clicking => Cancel -> SNAP 4

Snapshot of manager user View (SNAP 14):



Snapshot of Recharge View (SNAP 15):



1. Clicking => Recharge -> SNAP 11
2. Clicking => Back -> SNAP 11

Source Code Of Project is available on Github and link is given below.
<https://github.com/Kaphleamrit/javaGUIandDb>

Some Important Points Regarding Project Development:

- ★ Project is developed in Microsoft's VS Code.
- ★ Only Java and mySql Code is pasted in lab Record.
- ★ Codes of form (.form) files are not copied to report.
- ★ JDK used => JDK 11 .
- ★ The program starts by running App.java file.
- ★ You should have a mySql server in your computer to run the program.
- ★ You will need to run each statement of cswProject.sql file at first for running the program without error .
- ★ After running the sql scripts , you will get a manager and a customer , Later you can add customers but if you want more manager You need to Do it using mysql client i.e. mysql workbench or mysql cli client.
- ★ For manager , the id is admin and password is also admin.
- ★ For customer, the id is nonadmin and password is also nonadmin.