

Online Parking Reservation

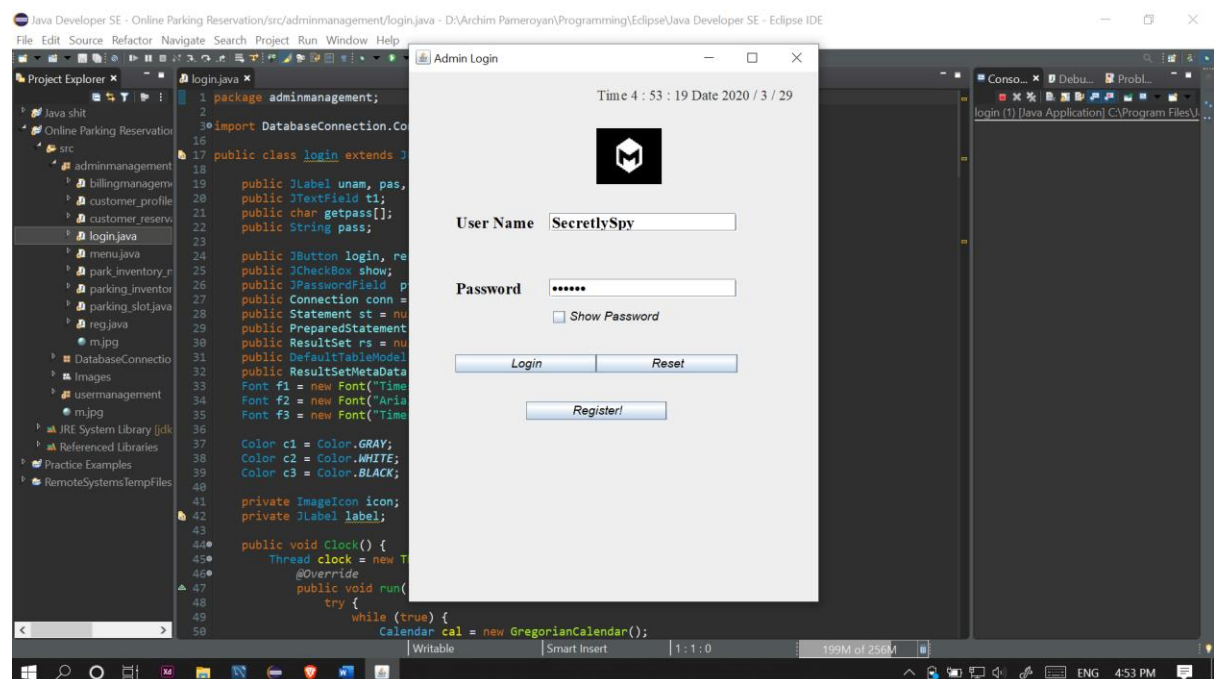
Manalad, Jhon Lawrence S.L.

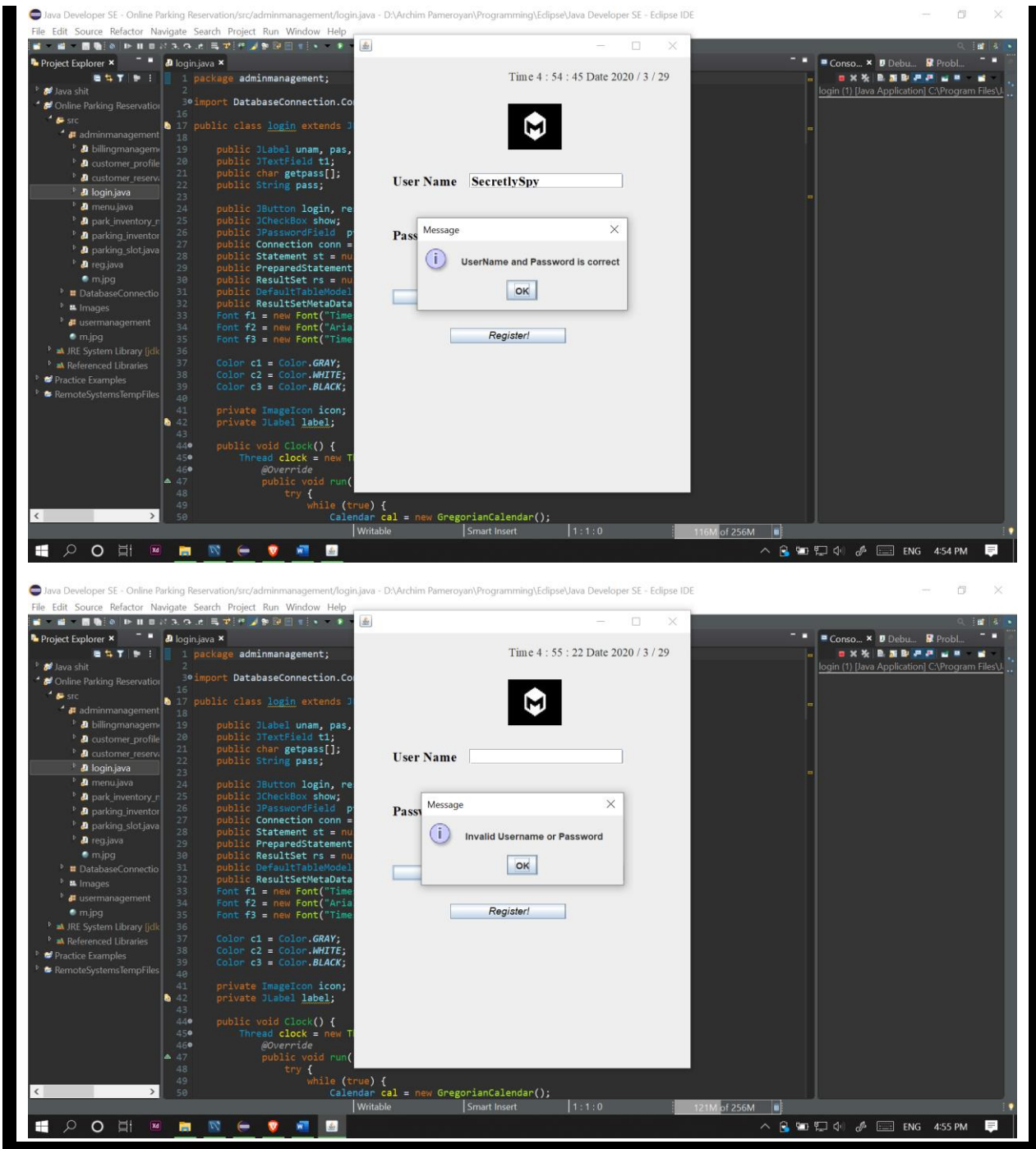
Pameroyan, Archim Paul C.

DICT 2-5

Admin Management

Admin Login





```
package adminmanagement;
```

```
import DatabaseConnection.ConnectionDB;
```

```
import javax.swing.*;
```

```
import javax.swing.table.DefaultTableModel;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
import java.util.Calendar;
import java.util.GregorianCalendar;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.Statement;

public class login extends JFrame implements ActionListener {

    public JLabel unam, pas, lblClock, photo;
    public JTextField t1;
    public char getpass[];
    public String pass;

    public JButton login, res, reg1;
    public JCheckBox show;
    public JPasswordField pfpwd;
    public Connection conn = null;
    public Statement st = null;
    public PreparedStatement ps = null;
    public ResultSet rs = null;
    public DefaultTableModel dm = null;
    public ResultSetMetaData rsmd = null;
    Font f1 = new Font("Times New Roman", Font.BOLD, 17);
    Font f2 = new Font("Arial", Font.ITALIC, 13);
    Font f3 = new Font("Times New Roman", Font.PLAIN, 15);
```

```

Color c1 = Color.GRAY;

Color c2 = Color.WHITE;

Color c3 = Color.BLACK;


private ImageIcon icon;

private JLabel label;


public void Clock() {
    Thread clock = new Thread() {
        @Override
        public void run() {
            try {
                while (true) {
                    Calendar cal = new GregorianCalendar();
                    int day = cal.get(Calendar.DAY_OF_MONTH);
                    int month = cal.get(Calendar.MONTH);
                    int year = cal.get(Calendar.YEAR);

                    int second = cal.get(Calendar.SECOND);
                    int minute = cal.get(Calendar.MINUTE);
                    int hour = cal.get(Calendar.HOUR);

                    lblClock.setText("Time " + hour + " : " + minute
+ " : " + second + " Date " + year + " / "
                                + (month + 1) + " / " + day);

                    sleep(1000);
                }
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    };
}

```

```

        }

    };

    clock.start();
}

public login() {

    conn = ConnectionDB.getConnection();

    icon = new ImageIcon("src/m.jpg");

    Container con = getContentPane();
    con.setLayout(null);

    photo = new JLabel() {

        public void paintComponent(Graphics g) {

            g.drawImage(icon.getImage(), 0, 0, null);

            super.paintComponent(g);

        }

    };

    photo.setOpaque(false);
    con.add(photo);
    photo.setBounds(200, 60, 150, 100);

    con.add(lblClock = new JLabel(""));
    lblClock.setBounds(200, 10, 230, 30);
    lblClock.setFont(f3);

    unam = new JLabel("User Name");

```

```
pas = new JLabel("Password");

t1 = new JTextField();
pfpwd = new JPasswordField();

login = new JButton("Login");
show = new JCheckBox("Show Password");
res = new JButton("Reset");
reg1 = new JButton("Register!");

login.addActionListener(this);
show.addActionListener(this);
res.addActionListener(this);
reg1.addActionListener(this);

unam.setBounds(50, 150, 100, 20);
pas.setBounds(50, 220, 100, 20);
t1.setBounds(150, 150, 200, 20);
pfpwd.setBounds(150, 220, 200, 20);
show.setBounds(150, 250, 150, 20);
login.setBounds(50, 300, 150, 20);
res.setBounds(200, 300, 150, 20);
reg1.setBounds(125, 350, 150, 20);

unam.setFont(f1);
pas.setFont(f1);
t1.setFont(f1);
pfpwd.setFont(f1);
```

```
login.setFont(f2);
```

```
res.setFont(f2);
```

```
show.setFont(f2);
```

```
reg1.setFont(f2);
```

```
unam.setForeground(c3);
```

```
pas.setForeground(c3);
```

```
t1.setForeground(c3);
```

```
pfpwd.setForeground(c3);
```

```
login.setForeground(c3);
```

```
res.setForeground(c3);
```

```
show.setForeground(c3);
```

```
reg1.setForeground(c3);
```

```
con.add(unam);
```

```
con.add(pas);
```

```
con.add(t1);
```

```
con.add(pfpwd);
```

```
con.add(login);
```

```
con.add(res);
```

```
con.add(show);
```

```
con.add(reg1);
```

```
Clock();
```

```
}
```

```
@Override
```

```

public void actionPerformed(ActionEvent e) {

    getpass = pfpwd.getPassword();
    pass = String.valueOf(getpass);
    // reg button papuntang regform

    if (e.getSource() == reg1) {
        reg app = new reg();
        app.setTitle("Admin Registration Form");
        app.setVisible(true);
        app.setSize(1300, 450);
        app.setLocationRelativeTo(null);
        login.this.dispose();
    }
    if (e.getSource() == login) {
        // Login Button

        getpass = pfpwd.getPassword();
        pass = String.valueOf(getpass);

        try {
            String sql = "select * from adminaccount where Username=?
and Password=?";

            ps = conn.prepareStatement(sql);
            ps.setString(1, t1.getText());
            ps.setString(2, pass);
            rs = ps.executeQuery();
            int count = 0;
            while (rs.next()) {

```



```

        count = count + 1;
    }
    if (count == 1) {
        JOptionPane.showMessageDialog(null,
"UserName and Password is correct");

        JOptionPane.showMessageDialog(this, "Login
Successful");

        // login button punta ng home/menu
        login.addActionListener(new ActionListener() {
            public void
actionPerformed(ActionEvent evt) {

                menu app = new menu();
                app.setTitle("Admin Menu");
                app.setVisible(true);
                app.setSize(1000, 400);
                app.setLocationRelativeTo(null);
                login.this.dispose();

            }
        });
    } else if (count > 1) {
        JOptionPane.showMessageDialog(null,
"Duplicate UserName and Password");
    } else {
        JOptionPane.showMessageDialog(null,
"UserName and Password is not correct Try Again...");

        JOptionPane.showMessageDialog(this, "Invalid
Username or Password");

        t1.setText("");
        pfpwd.setText("");
    }
}

```

```
        rs.close();
        ps.close();
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, ex);
    }
}

// Reset Button
if (e.getSource() == res) {
    t1.setText("");
    pfpwd.setText("");
}

// Show Password
if (e.getSource() == show) {
    if (show.isSelected()) {
        pfpwd.setEchoChar((char) 0);
    } else {
        pfpwd.setEchoChar('*');
    }
}
}
```

```
}
```

```
static login app = new login();
```

```
public static void main(String[] args) {
```

```
    app.setTitle("Admin Login");
```

```
    app.setSize(450, 600);
```

```
    app.setVisible(true);
```

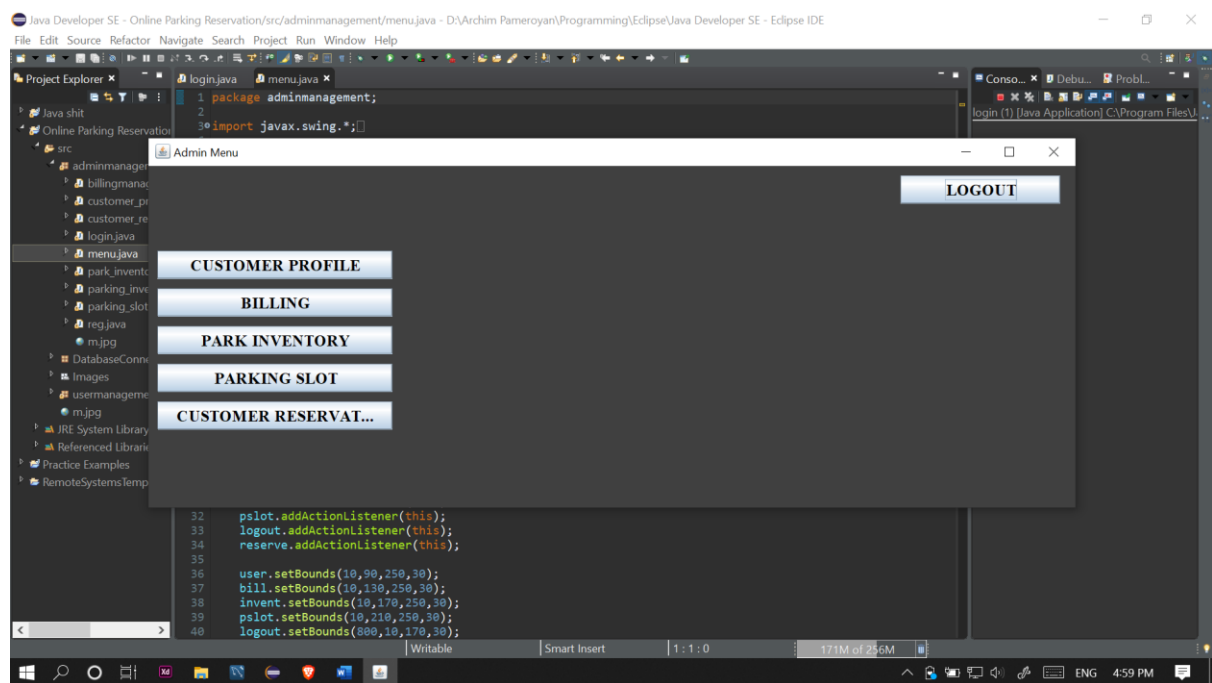
```
    app.setLocationRelativeTo(null);
```

```
    app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
}
```

```
}
```

Admin Menu



```
package adminmanagement;
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class menu extends JFrame implements ActionListener{

    public JButton user, bill, invent, pslot, logout, reserve;

    Font f1 = new Font("Times New Roman", Font.BOLD,17);
    Font f2 = new Font("Arial", Font.ITALIC,13);
    Color c1 = Color.GRAY;
    Color c2 = Color.BLACK;
    Color c3 = Color.WHITE;
    Color c4 = Color.DARK_GRAY;

    public menu() {
        Container con = getContentPane();
        con.setLayout(null);

        user = new JButton ("CUSTOMER PROFILE");
        bill = new JButton ("BILLING");
        invent = new JButton ("PARK INVENTORY");
        pslot = new JButton ("PARKING SLOT");
        logout = new JButton ("LOGOUT");
        reserve = new JButton ("CUSTOMER RESERVATION");

        user.addActionListener(this);
        bill.addActionListener(this);
        invent.addActionListener(this);
```

```
pslot.addActionListener(this);  
logout.addActionListener(this);  
reserve.addActionListener(this);
```

```
user.setBounds(10,90,250,30);  
bill.setBounds(10,130,250,30);  
invent.setBounds(10,170,250,30);  
pslot.setBounds(10,210,250,30);  
logout.setBounds(800,10,170,30);  
reserve.setBounds(10,250,250,30);
```

```
user.setFont(f1);  
bill.setFont(f1);  
invent.setFont(f1);  
pslot.setFont(f1);  
logout.setFont(f1);  
reserve.setFont(f1);
```

```
user.setForeground(c2);  
bill.setForeground(c2);  
invent.setForeground(c2);  
pslot.setForeground(c2);  
logout.setForeground(c2);  
reserve.setForeground(c2);
```

```
con.setBackground(c4);
```

```
con.add(user);  
con.add(bill);
```

```
con.add(invent);
con.add(pslot);
con.add(logout);
con.add(reserve);

}
```

@Override

```
public void actionPerformed(ActionEvent e) {
    // TODO Auto-generated method stub
    //paglogout
    if (e.getSource() == logout) {
        int dialog = JOptionPane.showConfirmDialog(null, "Are you sure you want to
        LOGOUT?", "WARNING",
        JOptionPane.YES_NO_OPTION,
        JOptionPane.WARNING_MESSAGE);
        if (dialog == JOptionPane.YES_OPTION) {
            //pabalik ng login.java
            login app = new login();
            app.setVisible(true);
            app.setSize(450,600);
            app.setVisible(true);
            app.setLocationRelativeTo(null);
            menu.this.dispose();
        }
        if (dialog == JOptionPane.NO_OPTION) {
            menu.this.show();
        }
    }
}
```

```
//papuntang billing
```

```
    if (e.getSource() == bill) {  
        billingmanagement app = new billingmanagement();  
        app.setTitle("Billing Management");  
        app.setVisible(true);  
        app.setSize(1000,700);  
        app.setLocationRelativeTo(null);  
        menu.this.dispose();  
    }
```

```
//papuntang inventory
```

```
    if (e.getSource() == invent) {  
        park_inventory_management app = new  
park_inventory_management();  
        app.setTitle("Parking Inventory Management");  
        app.setVisible(true);  
        app.setSize(1000,700);  
        app.setLocationRelativeTo(null);  
        menu.this.dispose();  
    }
```

```
//papuntang customer profile
```

```
    if (e.getSource() == user) {  
        customer_profile app = new customer_profile();  
        app.setTitle("Customer Profile");  
        app.setSize(700, 600);  
        app.setVisible(true);  
        app.setLocationRelativeTo(null);  
        menu.this.dispose();  
    }
```

```

    }

    //papuntang slot
    if (e.getSource() == pslot) {
        parking_slot app = new parking_slot();
        app.setTitle("Parking Slot");
        app.setSize(700, 600);
        app.setVisible(true);
        app.setLocationRelativeTo(null);
        menu.this.dispose();
    }

    //papuntang reservation
    if (e.getSource() == reserve) {
        customer_reservation app = new customer_reservation();
        app.setTitle("Customer Reservation");
        app.setSize(1000, 400);
        app.setVisible(true);
        app.setLocationRelativeTo(null);
        menu.this.dispose();
    }
}

```

```

static menu app = new menu();
public static void main(String[] args) {
    app.setTitle("Admin Menu");
    app.setSize(1000,400);
    app.setVisible(true);
}

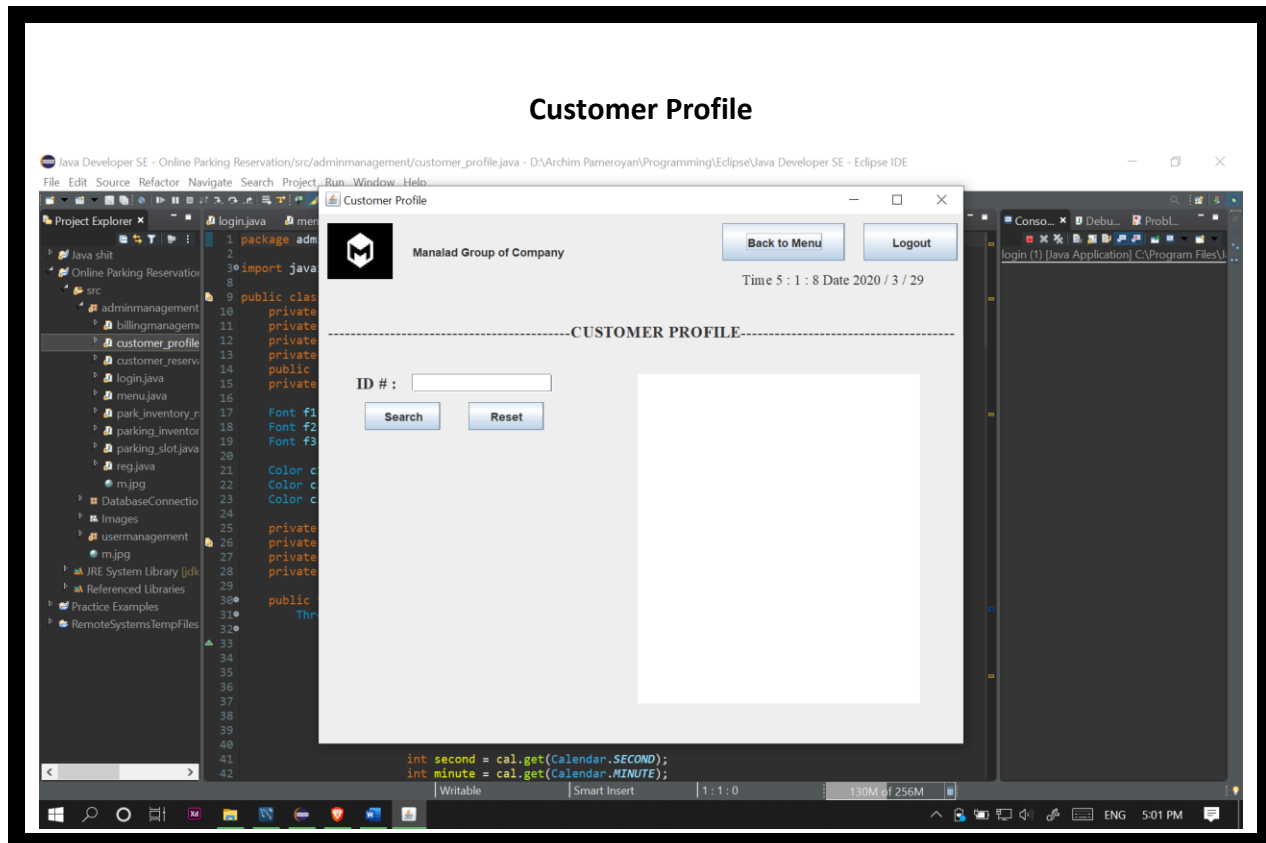
```



```
app.setLocationRelativeTo(null);
```

```
}
```

```
}
```



```
package adminmanagement;
```

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.util.Calendar;
```

```
import java.util.GregorianCalendar;
```

```
public class customer_profile extends JFrame implements ActionListener{
```

```
    private JLabel id;
```

```
    private JTextField tid;
```

```
    private JTextArea table;
```



```

        int minute = cal.get(Calendar.MINUTE);
        int hour = cal.get(Calendar.HOUR);

        lblClock.setText("Time " + hour + " : " + minute
+ " : " + second + " Date " + year + " / "
                                + (month + 1) + " / " + day);

        sleep(1000);
    }
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

};

clock.start();
}

```

```

public customer_profile () {

    icon = new ImageIcon("src/m.jpg");

    Container con = getContentPane();
    con.setLayout(null);

    photo = new JLabel() {
        public void paintComponent(Graphics g) {
            g.drawImage(icon.getImage(), 0, 0, null);
            super.paintComponent(g);
        }
    };
}

```

```
photo.setOpaque(false);
con.add(photo);
photo.setBounds(10,10,100,100);

title = new JLabel ("Manalad Group of Company");
title.setBounds(100,-10,200,100);
con.add(title);

con.add(lblClock = new JLabel(""));
lblClock.setBounds(450,10,300,120);
lblClock.setFont(f3);

logout = new JButton("Logout");
logout.addActionListener(this);
logout.setBounds(580,10,100,40);
con.add(logout);

search = new JButton("Search");
search.addActionListener(this);
search.setBounds(50,200,80,30);
con.add(search);

reset = new JButton("Reset");
reset.addActionListener(this);
reset.setBounds(160,200,80,30);
con.add(reset);

menu1 = new JButton ("Back to Menu");
menu1.addActionListener(this);
```

```

        menu1.setBounds(430,10,130,40);
        con.add(menu1);

        id = new JLabel ("ID # :");
        id.setFont(f1);
        id.setBounds(40,170,60,20);
        con.add(id);

        tid = new JTextField ();
        tid.setFont(f3);
        tid.setBounds(100,170,150,20);
        con.add(tid);

        table = new JTextArea ();
        table.setFont(f3);
        table.setBounds(340,170,300,350);
        con.add(table);

        cusprof = new JLabel ("-----CUSTOMER
PROFILE-----");
        cusprof.setFont(f1);
        cusprof.setBounds(10,100,750,50);
        con.add(cusprof);

        Clock();
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub

```

```

        //Reset Button
        if (e.getSource() == reset) {
            tid.setText("");
        }

        //pag logout
        if (e.getSource() == logout) {
            int dialog = JOptionPane.showConfirmDialog(null, "Are you sure you
want to LOGOUT?", "WARNING",
JOptionPane.YES_NO_OPTION,
JOptionPane.WARNING_MESSAGE);
            if (dialog == JOptionPane.YES_OPTION) {
                //pabalik ng login
                login app = new login();
                app.setTitle("Admin Login");
                app.setVisible(true);
                app.setSize(450,600);
                app.setVisible(true);
                app.setLocationRelativeTo(null);
                customer_profile.this.dispose();

            }
            if (dialog == JOptionPane.NO_OPTION) {
                customer_profile.this.show();
            }
        }

        //pagbalik sa menu
        if (e.getSource() == menu1) {

```

```

        //menu
        menu app = new menu();
            app.setTitle("Admin Menu");
            app.setVisible(true);
            app.setSize(1000,400);
            app.setVisible(true);
            app.setLocationRelativeTo(null);
            customer_profile.this.dispose();
        }

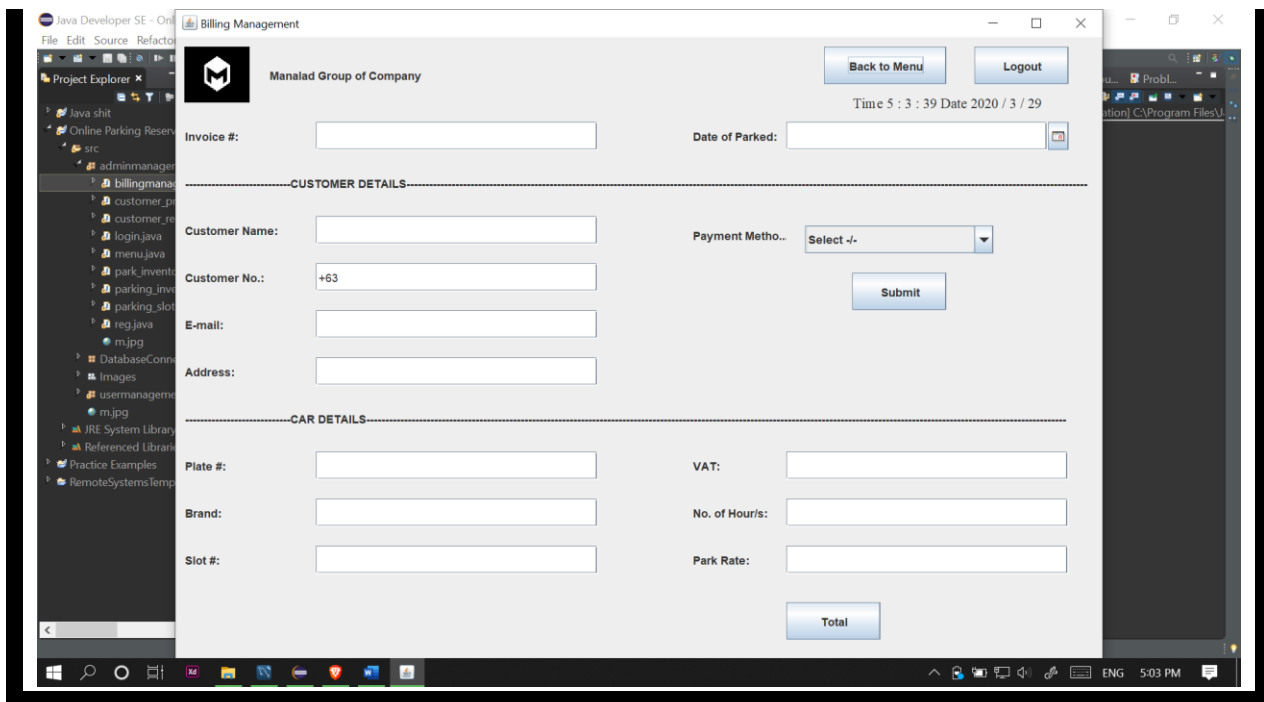
    }

    static customer_profile app = new customer_profile();
    public static void main(String[] args) {
        app.setTitle("Customer Profile");
        app.setSize(700, 600);
        app.setVisible(true);
        app.setLocationRelativeTo(null);
        app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }

}

```



```
package adminmanagement;
```

```
import javax.swing.*;
```

```
import com.toedter.calendar.JDateChooser;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.util.Calendar;
```

```
import java.util.GregorianCalendar;
```

```
public class billingmanagement extends JFrame implements ActionListener{
```

```
    public JLabel photo, title, invoc, cusdet, cusnam, contact, email, add, cardet,  
                platno, brand, slotno, date, paymeth, tax, hour, parkrate;
```

```
    public JTextField tinvoc, tcusnam, tcontact, temail, tadd, tplatno, tbrand, tslotno,  
                ttax, tdiscount, tparkrate;
```



```

public JComboBox <?> pay;

public JButton submit, logout, total, menu1;


Font f3 = new Font("Times New Roman", Font.PLAIN,15);


private com.toedter.calendar.JDateChooser Date;

private ImageIcon icon;

private JLabel label;

private JLabel lblClock;


public void Clock() {
    Thread clock = new Thread() {
        @Override
        public void run() {
            try {
                while (true) {
                    Calendar cal = new GregorianCalendar();
                    int day = cal.get(Calendar.DAY_OF_MONTH);
                    int month = cal.get(Calendar.MONTH);
                    int year = cal.get(Calendar.YEAR);

                    int second = cal.get(Calendar.SECOND);
                    int minute = cal.get(Calendar.MINUTE);
                    int hour = cal.get(Calendar.HOUR);

                    lblClock.setText("Time " + hour + " : " + minute
+ " : " + second + " Date " + year + " / "
                                + (month + 1) + " / " + day);
                    sleep(1000);
                }
            }
        }
    };
}

```

```

        }
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

};

clock.start();
}

```

```

public billingmanagement() {

```

```

    icon = new ImageIcon("src/m.jpg");

```

```

    Container con = getContentPane();
    con.setLayout(null);

```

```

    con.add(lblClock = new JLabel(""));
    lblClock.setBounds(720,10,300,120);
    lblClock.setFont(f3);

```

```

    photo= new JLabel() {
        public void paintComponent(Graphics g) {
            g.drawImage(icon.getImage(), 0, 0, null);
            super.paintComponent(g);
        }
    };

```

```
photo.setOpaque(false);
con.add(photo);
```

```
Date = new JDateChooser();
```

```
//photo = new JLabel ("/billingmanagement/imahe/m.jpg");
```

```
title = new JLabel ("Manalad Group of Company");
```

```
invoc = new JLabel ("Invoice #:");
```

```
cusdet = new JLabel ("-----CUSTOMER DETAILS-----  
-----");
```

```
cusnam = new JLabel ("Customer Name:");
```

```
contact = new JLabel ("Customer No.:");
```

```
email = new JLabel ("E-mail:");
```

```
add = new JLabel ("Address:");
```

```
cardet = new JLabel ("-----CAR DETAILS-----  
-----");
```

```
platno = new JLabel ("Plate #:");
```

```
brand = new JLabel ("Brand:");
```

```
slotno = new JLabel ("Slot #:");
```

```
date = new JLabel ("Date of Parked:");
```

```
paymeth = new JLabel ("Payment Method:");
```

```
tax = new JLabel ("VAT:");
```

```
hour = new JLabel ("No. of Hour/s:");
```

```
parkrate = new JLabel ("Park Rate:");
```

```
tinvoc = new JTextField ();
```

```
tcusnam = new JTextField ();
```

```
tcontact = new JTextField (" +63 ");
```

```
temail = new JTextField ();
tadd = new JTextField ();
tplatno = new JTextField ();
tbrand = new JTextField ();
tslotno = new JTextField ();
```

```
ttax = new JTextField ();
tdiscount = new JTextField ();
tparkrate = new JTextField ();
```

```
String Select[]
    = { "Select -/-", "Cash", "GCash", "7/11 Clique", "Credit/Debit
Card",
        "Paymaya", "Beep"};
```

```
pay = new JComboBox(Select);
```

```
submit = new JButton("Submit");
logout = new JButton("Logout");
total = new JButton("Total");
menu1 = new JButton ("Back to Menu");
```

```
submit.addActionListener(this);
logout.addActionListener(this);
total.addActionListener(this);
menu1.addActionListener(this);
```

```
//photo.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/billingmanagement/imahe/m.jpg"))));
photo.setBounds(10,10,100,100);
```

```
title.setBounds(100,-10,200,100);
invoc.setBounds(10,100,200,10);
cusdet.setBounds(10,150,1000,10);
cusnam.setBounds(10,200,100,10);
contact.setBounds(10,250,100,10);
email.setBounds(10,300,100,10);
add.setBounds(10,350,100,10);
cardet.setBounds(10,400,1000,10);
platno.setBounds(10,450,100,10);
brand.setBounds(10,500,100,10);
slotno.setBounds(10,550,100,10);
date.setBounds(550,100,100,10);
paymeth.setBounds(550,200,100,20);
tax.setBounds(550,450,100,10);
hour.setBounds(550,500,150,10);
parkrate.setBounds(550,550,100,10);
```

```
tinvoc.setBounds(150,90,300,30);
tcusnam.setBounds(150,190,300,30);
tcontact.setBounds(150,240,300,30);
temail.setBounds(150,290,300,30);
tadd.setBounds(150,340,300,30);
tplatno.setBounds(150,440,300,30);
tbrand.setBounds(150,490,300,30);
tslotno.setBounds(150,540,300,30);
Date.setBounds(650,90,300,30);
ttax.setBounds(650,440,300,30);
tdiscount.setBounds(650,490,300,30);
tparkrate.setBounds(650,540,300,30);
```

```
pay.setBounds(670,200,200,30);
```

```
logout.setBounds(850,10,100,40);
```

```
menu1.setBounds(690,10,130,40);
```

```
total.setBounds(650,600,100,40);
```

```
submit.setBounds(720,250,100,40);
```

```
// con.add(photo);
```

```
con.add(title);
```

```
con.add(invoc);
```

```
con.add(cusdet);
```

```
con.add(cusnam);
```

```
con.add(contact);
```

```
con.add(email);
```

```
con.add(add);
```

```
con.add(cardet);
```

```
con.add(platno);
```

```
con.add(brand);
```

```
con.add(slotno);
```

```
con.add(date);
```

```
con.add(paymeth);
```

```
con.add(tax);
```

```
con.add(hour);
```

```
con.add(parkrate);
```

```
con.add(tinvoc);
```

```
con.add(tcusnam);
```

```
con.add(tcontact);
```

```
        con.add(temail);
        con.add(tadd);
        con.add(tplatno);
        con.add(tbrand);
        con.add(tslotno);
        con.add(Date);
        con.add(ttax);
        con.add(tdiscount);
        con.add(tparkrate);

        con.add(pay);

        con.add(submit);
        con.add(logout);
        con.add(total);
        con.add(menu1);

        Clock();
    }
```

@Override

```
public void actionPerformed(ActionEvent e) {
    // TODO Auto-generated method stub
    //paglogout
    if (e.getSource() == logout) {
        int dialog = JOptionPane.showConfirmDialog(null, "Are you sure you
        want to LOGOUT?", "WARNING",
        JOptionPane.YES_NO_OPTION,
        JOptionPane.WARNING_MESSAGE);
        if (dialog == JOptionPane.YES_OPTION) {
```

```
//pabalik ng login
    login app = new login();
    app.setTitle("Admin Login");
    app.setVisible(true);
    app.setSize(450,600);
    app.setVisible(true);
    app.setLocationRelativeTo(null);
    billingmanagement.this.dispose();

}
if (dialog == JOptionPane.NO_OPTION) {
    billingmanagement.this.show();
}
}
```

//para sa date

```
//pagbalik sa menu
if (e.getSource() == menu1) {
    //menu
    menu app = new menu();
    app.setTitle("Admin Menu");
    app.setVisible(true);
    app.setSize(1000,400);
    app.setVisible(true);
    app.setLocationRelativeTo(null);
    billingmanagement.this.dispose();
}
```



```
}
```

```
static billingmanagement app = new billingmanagement();
```

```
public static void main(String[] args) {
```

```
    app.setTitle("Billing Management");
```

```
    app.setSize(1000, 700);
```

```
    app.setVisible(true);
```

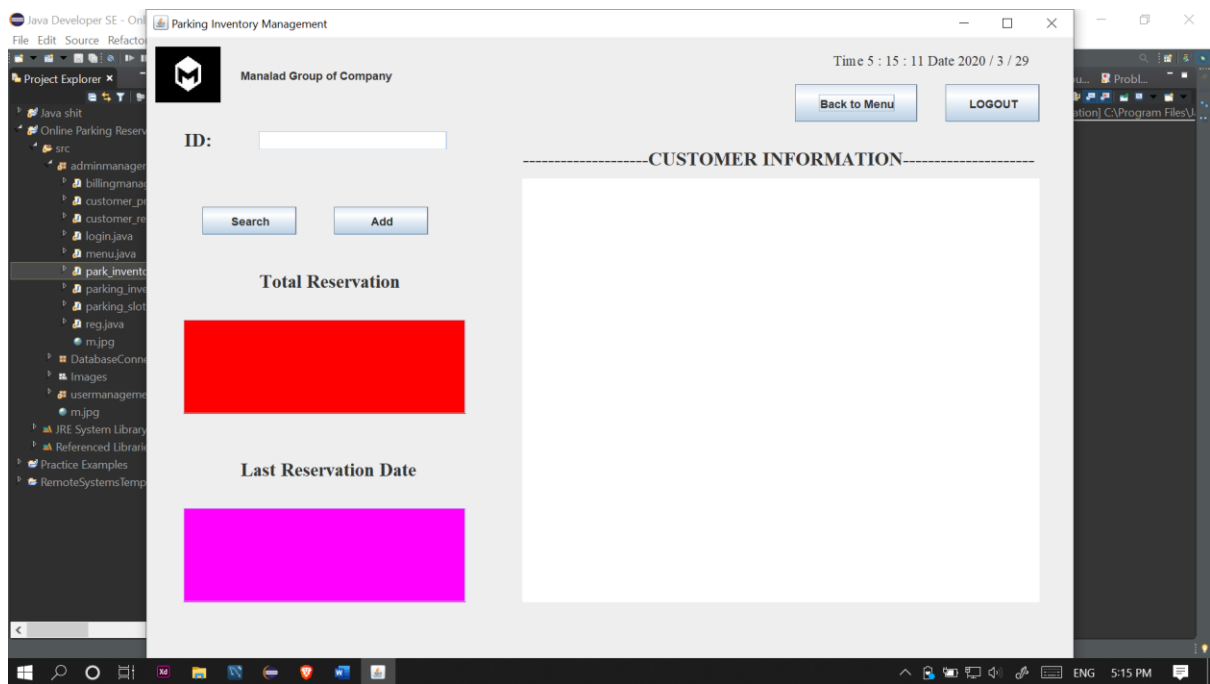
```
    app.setLocationRelativeTo(null);
```

```
    app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
}
```

```
}
```

Parking Inventory



```
package adminmanagement;
```

```
import javax.swing.*;
```

```
import javax.swing.table.DefaultTableModel;
```

```
import DatabaseConnection.ConnectionDB;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.sql.Connection;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import java.sql.ResultSetMetaData;
```

```
import java.sql.Statement;
```

```
import java.util.Calendar;
```

```
import java.util.GregorianCalendar;
```

```
public class park_inventory_management extends JFrame implements ActionListener {
```

```
    public JLabel id, cus, tr, lrd;
```

```
    public JTextField t1, total, date;
```

```
    public JButton search, add1, logout, menu1;
```

```
    public JTable table;
```

```
    public Connection conn = null;
```

```
    public Statement st = null;
```

```
    public PreparedStatement ps = null;
```

```
    public ResultSet rs = null;
```

```
    public DefaultTableModel dm = null;
```

```
    public ResultSetMetaData rsmd = null;
```

```
    Font f1 = new Font("Times New Roman",Font.PLAIN, 15);
```

```
    Font f2 = new Font("Times New Roman",Font.BOLD, 20);
```

```
    Font f3 = new Font("Times New Roman", Font.PLAIN,15);
```

```
    Color c1 = Color.WHITE;
```

```
    Color c2 = Color.RED;
```

```
    Color c3 = Color.MAGENTA;
```

```
    private JLabel lblClock;
```

```
    private ImageIcon icon;
```

```
    private JLabel label;
```

```
    private JLabel photo;
```

```
    private JLabel title;
```

```
    public void Clock() {
```

```
        Thread clock = new Thread() {
```

```
            @Override
```

```
            public void run() {
```

```

        try {
            while (true) {
                Calendar cal = new GregorianCalendar();
                int day = cal.get(Calendar.DAY_OF_MONTH);
                int month = cal.get(Calendar.MONTH);
                int year = cal.get(Calendar.YEAR);

                int second = cal.get(Calendar.SECOND);
                int minute = cal.get(Calendar.MINUTE);
                int hour = cal.get(Calendar.HOUR);

                lblClock.setText("Time " + hour + " : " + minute
+ " : " + second + " Date " + year + " / "
                                + (month + 1) + " / " + day);

                sleep(1000);
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }

    };

    clock.start();
}

```

```

public park_inventory_management() {
    conn = ConnectionDB.getConnection();
    icon = new ImageIcon("src/m.jpg");

    Container con = getContentPane();
    con.setLayout(null);
}

```

```
photo= new JLabel() {  
    public void paintComponent(Graphics g) {  
        g.drawImage(icon.getImage(), 0, 0, null);  
        super.paintComponent(g);  
    }  
};
```

```
photo.setOpaque(false);  
con.add(photo);  
photo.setBounds(10,10,100,100);
```

```
con.add(lblClock = new JLabel(""));  
lblClock.setBounds(730, 10, 230, 30);  
lblClock.setFont(f3);
```

```
title = new JLabel ("Manalad Group of Company");  
title.setBounds(100,-10,200,100);  
con.add(title);
```

```
id = new JLabel ("ID:");  
id.setFont(f2);  
cus = new JLabel ("-----CUSTOMER INFORMATION-----");  
cus.setFont(f2);  
tr = new JLabel ("Total Reservation");  
lrd = new JLabel ("Last Reservation Date");  
t1 = new JTextField ();  
total = new JTextField ();
```

```
date = new JTextField ();
```

```
tr.setFont(f2);
```

```
lrd.setFont(f2);
```

```
t1.setEditable(false);
```

```
t1.setBackground(c1);
```

```
t1.setFont(f1);
```

```
total.setEditable(false);
```

```
total.setBackground(c2);
```

```
total.setFont(f2);
```

```
total.setForeground(c1);
```

```
date.setEditable(false);
```

```
date.setBackground(c3);
```

```
date.setFont(f2);
```

```
date.setForeground(c1);
```

```
search = new JButton ("Search");
```

```
add1 = new JButton ("Add");
```

```
logout = new JButton ("LOGOUT");
```

```
menu1 = new JButton ("Back to Menu");
```

```
search.addActionListener(this);
```

```
add1.addActionListener(this);
```

```
logout.addActionListener(this);
```

```
menu1.addActionListener(this);
```

```
table = new JTable ();
```

```
id.setBounds(40,100,100,20);
cus.setBounds(400,120,1000,20);
tr.setBounds(120,250,300,20);
lrd.setBounds(100,450,300,20);
t1.setBounds(120,100,200,20);
total.setBounds(40,300,300,100);
date.setBounds(40,500,300,100);
search.setBounds(60,180,100,30);
add1.setBounds(200,180,100,30);
table.setBounds(400,150,550,450);
logout.setBounds(850,50,100,40);
menu1.setBounds(690,50,130,40);
```

```
con.add(id);
con.add(cus);
con.add(lrd);
con.add(tr);
con.add(t1);
con.add(total);
con.add(date);
con.add(search);
con.add(add1);
con.add(table);
con.add(logout);
con.add(menu1);
```

```
Clock();
```

```
}
```

```
@Override
```

```
public void actionPerformed(ActionEvent e) {
```

```
    // TODO Auto-generated method stub
```

```
    //paglogout
```

```
        if (e.getSource() == logout) {
```

```
            int dialog = JOptionPane.showConfirmDialog(null, "Are you  
sure you want to LOGOUT?", "WARNING",
```

```
                JOptionPane.YES_NO_OPTION);
```

```
            if (dialog == JOptionPane.YES_OPTION) {
```

```
                //pabalik ng login
```

```
                login app = new login();
```

```
                app.setTitle("Admin Login");
```

```
                app.setVisible(true);
```

```
                app.setSize(450,600);
```

```
                app.setVisible(true);
```

```
                app.setLocationRelativeTo(null);
```

```
                park_inventory_management.this.dispose();
```

```
            }
```

```
            if (dialog == JOptionPane.NO_OPTION) {
```

```
                park_inventory_management.this.show();
```

```
            }
```

```
        }
```

```
    //pabalik ng menu
```

```
    //pagbalik sa menu
```

```
    if (e.getSource() == menu1) {
```

```
        //menu
```



```

        menu app = new menu();
        app.setTitle("Admin Menu");
        app.setVisible(true);
        app.setSize(1000,400);
        app.setVisible(true);
        app.setLocationRelativeTo(null);
        park_inventory_management.this.dispose();
    }

    //add button
    if (e.getSource() == add1) {
        parking_inventory_management_add_button app = new
parking_inventory_management_add_button();
        app.setTitle("Parking Inventory Management");
        app.setVisible(true);
        app.setSize(1200, 750);
        app.setVisible(true);
        app.setLocationRelativeTo(null);
        park_inventory_management.this.dispose();
    }

    //search button
    if (e.getSource() == search){

        String m = JOptionPane.showInputDialog(null, "Type the ID #",
"MANALAD", JOptionPane.INFORMATION_MESSAGE);
        t1.setText(m);
    }

```

```
}
```

```
static park_inventory_management app = new park_inventory_management();
```

```
public static void main(String[] args) {
```

```
    app.setTitle("Parking Inventory Management");
```

```
    app.setSize(1000, 700);
```

```
    app.setVisible(true);
```

```
    app.setLocationRelativeTo(null);
```

```
    app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
}
```

```
}
```

Parking Inventory

CUSTOMER DETAILS

ID:

Full Name:

E-mail:

Gender: ☒ Male ☐ Female

Mobile #:

Birthdate:

PARKING DETAILS

Date of Reservation:

Date of Parked:

Slot #:

Total Hours of Parked:

Time of Park:

Time of Departue:

CAR DETAILS

PLATE #:

Brand:

Color: or

Type of Car:

SUMMARY PROFILE

```
package adminmanagement;
```

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;

import DatabaseConnection.ConnectionDB;
import com.toedter.calendar.JDateChooser;

import java.awt.*;
import java.awt.Container;
import java.awt.event.*;
import java.util.Calendar;
import java.util.GregorianCalendar;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.Statement;

public class parking_inventory_management_add_button extends JFrame implements
ActionListener {

    public JLabel custdet, carprof, id, fnam, lnam, bdate, mno, email, platno, brand,
color, gender;

    public JTextField tid, tfnam, tlnam, tmno, temail, tplatno, tbrand, tcolor;

    public JRadioButton male, female;

    public ButtonGroup gengp;

    public JComboBox<?> date, month, year;

    public Connection conn = null;

    public Statement st = null;

    public PreparedStatement ps = null;
```

```

public ResultSet rs = null;

public DefaultTableModel dm = null;

public ResultSetMetaData rsmd = null;

Font f1 = new Font("Times New Roman", Font.PLAIN, 15);
Font f2 = new Font("Times New Roman", Font.BOLD, 20);
Font f3 = new Font("Times New Roman", Font.PLAIN, 13);
Font f4 = new Font("Times New Roman", Font.CENTER_BASELINE, 15);
// Color c1 = new Color.

private JLabel lblClock;
private ImageIcon icon;
private JLabel label;
private JLabel photo;
private JLabel title;
private JLabel or;
private JButton col;
private JLabel type;
private JComboBox<?> combo;
private JButton sub;
private JButton reset1;
private JTextArea tout;
private JTextArea tout1;
public com.toedter.calendar.JDateChooser Date, Date1;
public JLabel dpark, dres, tot, slot, time, time1;
public JTextField ttot, tslot, ttime, ttime1;

public void Clock() {
    Thread clock = new Thread() {
        @Override

```

```

        public void run() {
            try {
                while (true) {

                    Calendar cal = new GregorianCalendar();

                    int day = cal.get(Calendar.DAY_OF_MONTH);
                    int month = cal.get(Calendar.MONTH);
                    int year = cal.get(Calendar.YEAR);

                    int second = cal.get(Calendar.SECOND);
                    int minute = cal.get(Calendar.MINUTE);
                    int hour = cal.get(Calendar.HOUR);

                    lblClock.setText("Time " + hour + " : " + minute
+ " : " + second + " Date " + year + " / "
                                + (month + 1) + " / " + day);

                    sleep(1000);

                }
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    };

    clock.start();
}

```

```

public parking_inventory_management_add_button() {
    conn = ConnectionDB.getConnection();

    icon = new ImageIcon("src/m.jpg");

    Container con = getContentPane();
}

```

```
con.setLayout(null);

photo = new JLabel() {
    public void paintComponent(Graphics g) {
        g.drawImage(icon.getImage(), 0, 0, null);
        super.paintComponent(g);
    }
};

photo.setOpaque(false);
con.add(photo);
photo.setBounds(10, 10, 100, 100);

con.add(lblClock = new JLabel(""));
lblClock.setBounds(930, 10, 230, 30);
lblClock.setFont(f3);

Date = new JDateChooser();
Date.setBounds(750, 110, 200, 30);
con.add(Date);

Date1 = new JDateChooser();
Date1.setBounds(750, 150, 200, 30);
con.add(Date1);

title = new JLabel("Manalad Group of Company");
title.setBounds(100, -10, 200, 100);
con.add(title);
```

```
        custdet = new JLabel(  
            "-----CUSTOMER DETAILS-----  
-----PARKING DETAILS-----");  
        custdet.setBounds(10, 60, 1200, 50);  
        custdet.setFont(f2);  
        con.add(custdet);  
  
        id = new JLabel("ID:");  
        id.setBounds(40, 110, 50, 30);  
        id.setFont(f1);  
        con.add(id);  
  
        fnam = new JLabel("Full Name:");  
        fnam.setBounds(40, 150, 80, 30);  
        fnam.setFont(f1);  
        con.add(fnam);  
  
        lnam = new JLabel("E-mail:");  
        lnam.setBounds(40, 190, 80, 30);  
        lnam.setFont(f1);  
        con.add(lnam);  
  
        gender = new JLabel("Gender:");  
        gender.setBounds(40, 230, 80, 30);  
        gender.setFont(f1);  
        con.add(gender);  
  
        mno = new JLabel("Mobile #:");  
        mno.setBounds(40, 270, 80, 30);  
        mno.setFont(f1);
```

```
con.add(mno);

carprof = new JLabel(
    "-----CAR DETAILS-----
SUMMARY PROFILE");
carprof.setBounds(10, 330, 1200, 50);
carprof.setFont(f2);
con.add(carprof);

platno = new JLabel("PLATE #");
platno.setBounds(40, 370, 80, 30);
platno.setFont(f1);
con.add(platno);

brand = new JLabel("Brand:");
brand.setBounds(40, 410, 80, 30);
brand.setFont(f1);
con.add(brand);

color = new JLabel("Color:");
color.setBounds(40, 450, 80, 30);
color.setFont(f1);
con.add(color);

or = new JLabel("or");
or.setBounds(310, 450, 20, 30);
or.setFont(f1);
con.add(or);

col = new JButton("Choose Color");
```



```
col.setBounds(330, 450, 120, 30);  
col.setFont(f1);  
col.addActionListener(this);  
con.add(col);
```

```
type = new JLabel("Type of Car:");  
type.setBounds(40, 490, 100, 30);  
type.setFont(f1);  
con.add(type);
```

```
String Select[] = { "Select -/-", "Hatchback", "Sedan", "MPV", "SUV",  
"Crossover", "Coupe", "Convertible",  
"Sports Car", "Utility", "Compact", "Mini Van", "Van", "Pick-Up  
Truck" };
```

```
combo = new JComboBox(Select);  
combo.setFont(new Font("Arial", Font.PLAIN, 15));  
combo.setBounds(150, 490, 100, 30);  
con.add(combo);
```

```
sub = new JButton("Submit");  
sub.setFont(new Font("Arial", Font.PLAIN, 15));  
sub.setBounds(150, 570, 100, 20);  
sub.addActionListener(this);  
con.add(sub);
```

```
reset1 = new JButton("Reset");  
reset1.setFont(new Font("Arial", Font.PLAIN, 15));  
reset1.setBounds(300, 570, 100, 20);  
reset1.addActionListener(this);  
con.add(reset1);
```

```
tout = new JTextArea();  
tout.setFont(new Font("Arial", Font.PLAIN, 15));  
tout.setBounds(550, 370, 250, 300);  
tout.setLineWrap(true);  
tout.setEditable(false);  
con.add(tout);
```

```
tout1 = new JTextArea();  
tout1.setFont(new Font("Arial", Font.PLAIN, 15));  
tout1.setBounds(800, 370, 350, 300);  
tout1.setLineWrap(true);  
tout1.setEditable(false);  
con.add(tout1);
```

```
tid = new JTextField();  
tid.setBounds(150, 110, 200, 30);  
tid.setFont(f1);  
con.add(tid);
```

```
tfnam = new JTextField();  
tfnam.setBounds(150, 150, 300, 30);  
tfnam.setFont(f1);  
con.add(tfnam);
```

```
tlnam = new JTextField();  
tlnam.setBounds(150, 190, 300, 30);  
tlnam.setFont(f1);  
con.add(tlnam);
```

```
tmno = new JTextField("+63");  
tmno.setBounds(150, 270, 300, 30);  
tmno.setFont(f1);  
con.add(tmno);
```

```
tplatno = new JTextField();  
tplatno.setBounds(150, 370, 300, 30);  
tplatno.setFont(f1);  
con.add(tplatno);
```

```
tbrand = new JTextField();  
tbrand.setBounds(150, 410, 300, 30);  
tbrand.setFont(f1);  
con.add(tbrand);
```

```
tcolor = new JTextField();  
tcolor.setBounds(150, 450, 150, 30);  
tcolor.setFont(f1);  
con.add(tcolor);
```

```
male = new JRadioButton("Male");  
male.setFont(new Font("Arial", Font.PLAIN, 15));  
male.setSelected(true);  
male.setBounds(150, 230, 80, 30);  
male.setActionCommand("male");  
con.add(male);
```

```
female = new JRadioButton("Female");
```

```
female.setFont(new Font("Arial", Font.PLAIN, 15));
```

```
female.setSelected(true);
```

```
female.setBounds(230, 230, 80, 30);
```

```
female.setActionCommand("female");
```

```
con.add(female);
```

```
gengp = new ButtonGroup();
```

```
gengp.add(male);
```

```
gengp.add(female);
```

```
bdate = new JLabel("Birthdate:");
```

```
bdate.setFont(new Font("Arial", Font.PLAIN, 15));
```

```
bdate.setBounds(40, 310, 80, 30);
```

```
con.add(bdate);
```

```
String dates[] = { "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12",  
"13", "14", "15", "16", "17",  
"18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28",  
"29", "30", "31" };
```

```
date = new JComboBox(dates);
```

```
date.setFont(new Font("Arial", Font.PLAIN, 15));
```

```
date.setBounds(250, 310, 80, 30);
```

```
con.add(date);
```

```
String months[] = { "Jan", "feb", "Mar", "Apr", "May", "Jun", "July", "Aug",  
"Sep", "Oct", "Nov", "Dec" };
```

```
month = new JComboBox(months);
```

```
month.setFont(new Font("Arial", Font.PLAIN, 15));
```

```
month.setBounds(150, 310, 80, 30);
```

```
con.add(month);
```

```

        String years[] = { "1930", "1931", "1932", "1933", "1934", "1935", "1936",
"1937", "1938", "1939", "1940",

        "1941", "1942", "1943", "1944", "1945", "1946", "1947",
"1948", "1949", "1950", "1951", "1952", "1953",

        "1954", "1955", "1956", "1957", "1958", "1959", "1960",
"1961", "1962", "1963", "1964", "1965", "1966",

        "1967", "1968", "1969", "1970", "1971", "1972", "1973",
"1974", "1975", "1976", "1977", "1978", "1979",

        "1980", "1981", "1982", "1983", "1984", "1985", "1986",
"1987", "1988", "1989", "1990", "1991", "1992",

        "1993", "1994", "1995", "1996", "1997", "1998", "1999",
"2000", "2001", "2002", "2003", "2004", "2005",

        "2006", "2007", "2008", "2009", "2010", "2011", "2012",
"2013", "2014", "2015", "2016", "2017", "2018",

        "2019", "2020" };

        year = new JComboBox(years);

        year.setFont(new Font("Arial", Font.PLAIN, 15));

        year.setBounds(350, 310, 80, 30);

        con.add(year);


        dpark = new JLabel("Date of Parked:");

        dpark.setFont(f1);

        dpark.setBounds(580, 150, 150, 30);

        con.add(dpark);


        dres = new JLabel("Date of Reservation:");

        dres.setFont(f1);

        dres.setBounds(580, 110, 150, 30);

        con.add(dres);


        time = new JLabel("Time of Park:");

```

```
time.setFont(f1);  
time.setBounds(580, 270, 150, 30);  
con.add(time);
```

```
time1 = new JLabel("Time of Departue:");  
time1.setFont(f1);  
time1.setBounds(580, 310, 150, 30);  
con.add(time1);
```

```
tot = new JLabel("Total Hours of Parked:");  
tot.setFont(f1);  
tot.setBounds(580, 230, 150, 30);  
con.add(tot);
```

```
slot = new JLabel("Slot #:");  
slot.setFont(f1);  
slot.setBounds(580, 190, 150, 30);  
con.add(slot);
```

```
ttime = new JTextField();  
ttime.setBounds(750, 190, 150, 30);  
con.add(ttime);
```

```
ttime1 = new JTextField();  
ttime1.setBounds(750, 230, 150, 30);  
con.add(ttime1);
```

```
ttot = new JTextField();  
ttot.setBounds(750, 270, 150, 30);
```

```

        con.add(ttot);

        tslot = new JTextField();
        tslot.setBounds(750, 310, 150, 30);
        con.add(tslot);

        Clock();

    }

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        if (e.getSource() == sub) {

            String data1;

            String data = "          CUSTOMER DETAILS  " + "\n" + "ID # : " +
tid.getText() + "\n" + "Full Name : "
                                + tfnam.getText() + "\n" + "E-mail : " + tlnam.getText() +
"\n" + "Mobile # : " + tmno.getText()
                                + "\n";

            if (male.isSelected())
                data1 = "Gender : Male" + "\n";
            else
                data1 = "Gender : Female" + "\n";

            String data2 = "Date of Birth : " + (String) date.getSelectedItem() + "/"
+ (String) month.getSelectedItem()

```

```

CAR DETAILS " + "\n"
+ "/" + (String) year.getSelectedItem() + "\n" + "\n" + "
+ "PLATE # :" + tplatno.getText() + "\n" + "Brand :" +
tbrand.getText() + "\n"
+ "Color of the Car :" + tcolor.getText() + "\n";

String data3 = "Type of Car : " + (String) combo.getSelectedItem()

+ "\n";

String datas = " PARKING DETAILS " + "\n" + "Date of
Reservation :" + Date.getDate() + "\n"
+ "Date of Parked :" + Date1.getDate() + "\n" + "Slot #
:" + ttime.getText() + "\n"
+ "Total Hours Parked : " + ttime1.getText() + "\n" +
"Time of Park : " + ttot.getText() + "\n"
+ "Time of Departure : " + tslot.getText() + "\n";

tout.setText(data + data1 + data2 + data3);
tout.setEditable(false);
tout1.setText(datas);
tout1.setEditable(false);

try {
String query = "insert into inventory
(ID,FullName,Email,Gender,MobileNumber,Birthdate,PlateNumber,Brand,Color,Type,DOR,D
OP,SlotNumber,THP,TP,TD) values (?,?,?,?,?,?,?,?,?,?,?,?,?,?) ";

ps = conn.prepareStatement(query);
ps.setString(1, tid.getText());
ps.setString(2, fnam.getText());
ps.setString(3, tlnam.getText());
ps.setString(4, gengp.getSelection().getActionCommand());

```



```

        ps.setString(5, tmno.getText());
        ps.setString(6, (String) date.getSelectedItemId());
        ps.setString(6, (String) month.getSelectedItemId());
        ps.setString(6, (String) year.getSelectedItemId());
        ps.setString(7, tplatno.getText());
        ps.setString(8, tbrand.getText());
        ps.setString(9, tcolor.getText());
        ps.setString(10, (String) combo.getSelectedItemId());
        ps.setString(11, (String) Date.getDateFormatString());
        ps.setString(12, (String) Date1.getDateFormatString());
        ps.setString(13, (String) ttime.getText());
        ps.setString(14, (String) ttime1.getText());
        ps.setString(15, (String) tslot.getText());
        ps.setString(16, (String) ttot.getText());
        ps.execute();
        ps.close();

        JOptionPane.showMessageDialog(null, "Data Saved");
        JOptionPane.showMessageDialog(this, "Add Successfully");
    } catch (Exception ex) {
        ex.printStackTrace();
    }
} else {
    tout.setText("");
    tout1.setText("");
}

// pabalik ng parking inventory
park_inventory_management app = new
park_inventory_management();

```

```
app.setTitle("Parking Inventory Management");
app.setVisible(true);
app.setSize(1000, 700);
app.setVisible(true);
app.setLocationRelativeTo(null);
parking_inventory_management_add_button.this.dispose();
```

```
if (e.getSource() == reset1) {
    String def = "";
    tid.setText(def);
    tfnam.setText(def);
    tout.setText(def);
    date.setSelectedIndex(0);
    month.setSelectedIndex(0);
    year.setSelectedIndex(0);
    combo.setSelectedIndex(0);
}
```

```
}
```

```
//main
```

```
static parking_inventory_management_add_button app = new
parking_inventory_management_add_button();
```

```
public static void main(String[] args) {
    app.setTitle("Parking Inventory Management");
    app.setSize(1200, 750);
```

```

app.setVisible(true);

app.setLocationRelativeTo(null);

app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

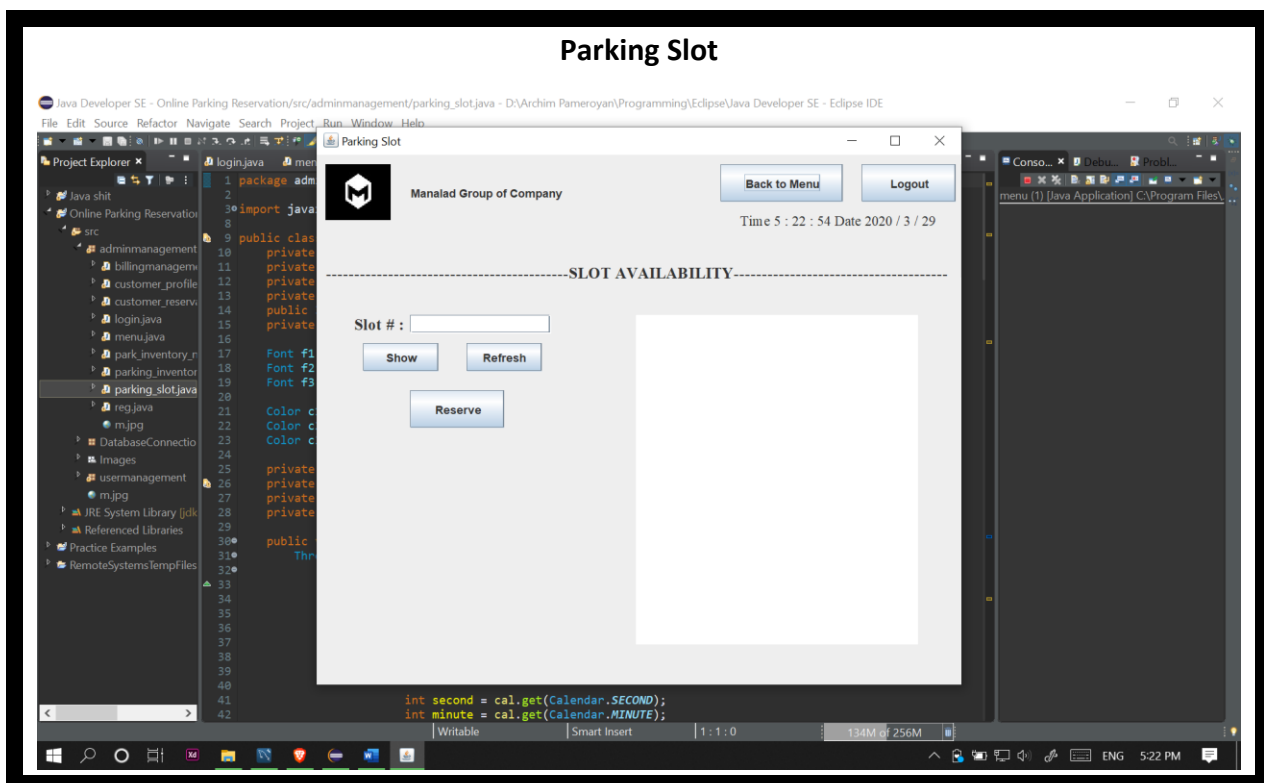
}

```

```

}

```



```

package adminmanagement;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.Calendar;
import java.util.GregorianCalendar;

public class parking_slot extends JFrame implements ActionListener{

```



```

        int year = cal.get(Calendar.YEAR);

        int second = cal.get(Calendar.SECOND);
        int minute = cal.get(Calendar.MINUTE);
        int hour = cal.get(Calendar.HOUR);

        lblClock.setText("Time " + hour + " : " + minute
+ " : " + second + " Date " + year + " / "
                                + (month + 1) + " / " + day);

        sleep(1000);
    }
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

};
clock.start();
}

public parking_slot () {

    icon = new ImageIcon("src/m.jpg");

    Container con = getContentPane();
    con.setLayout(null);

    photo = new JLabel() {
        public void paintComponent(Graphics g) {
            g.drawImage(icon.getImage(), 0, 0, null);
            super.paintComponent(g);
        }
    };
}

```

```
}  
};  
  
photo.setOpaque(false);  
con.add(photo);  
photo.setBounds(10,10,100,100);  
  
title = new JLabel ("Manalad Group of Company");  
title.setBounds(100,-10,200,100);  
con.add(title);  
  
con.add(lblClock = new JLabel(""));  
lblClock.setBounds(450,10,300,120);  
lblClock.setFont(f3);  
  
logout = new JButton("Logout");  
logout.addActionListener(this);  
logout.setBounds(580,10,100,40);  
con.add(logout);  
  
parking = new JButton("Reserve");  
parking.addActionListener(this);  
parking.setBounds(100,250,100,40);  
con.add(parking);  
  
search = new JButton("Show");  
search.addActionListener(this);  
search.setBounds(50,200,80,30);  
con.add(search);
```

```
reset = new JButton("Refresh");
reset.addActionListener(this);
reset.setBounds(160,200,80,30);
con.add(reset);
```

```
menu1 = new JButton ("Back to Menu");
menu1.addActionListener(this);
menu1.setBounds(430,10,130,40);
con.add(menu1);
```

```
id = new JLabel ("Slot # :");
id.setFont(f1);
id.setBounds(40,170,60,20);
con.add(id);
```

```
tid = new JTextField ();
tid.setFont(f3);
tid.setBounds(100,170,150,20);
con.add(tid);
```

```
table = new JTextArea ();
table.setFont(f3);
table.setBounds(340,170,300,350);
con.add(table);
```

```
cusprof = new JLabel ("-----SLOT
AVAILABILITY-----");
cusprof.setFont(f1);
cusprof.setBounds(10,100,750,50);
```

```

        con.add(cusprof);

        Clock();
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        //Reset Button
        if (e.getSource() == reset) {
            tid.setText("");
        }

        //pag logout
        if (e.getSource() == logout) {
            int dialog = JOptionPane.showConfirmDialog(null, "Are you sure you
want to LOGOUT?", "WARNING",
                JOptionPane.YES_NO_OPTION,
JOptionPane.WARNING_MESSAGE);
            if (dialog == JOptionPane.YES_OPTION) {
                //pabalik ng login
                login app = new login();
                app.setTitle("Admin Login");
                app.setVisible(true);
                app.setSize(450,600);
                app.setVisible(true);
                app.setLocationRelativeTo(null);
                parking_slot.this.dispose();

            }

```



```
        if (dialog == JOptionPane.NO_OPTION) {
            parking_slot.this.show();
        }
    }

//pagbalik sa menu
if (e.getSource() == menu1) {
    //menu
    menu app = new menu();
    app.setTitle("Admin Menu");
    app.setVisible(true);
    app.setSize(1000,400);
    app.setVisible(true);
    app.setLocationRelativeTo(null);
    parking_slot.this.dispose();
}

//reserve button
if (e.getSource() == parking) {
    park_inventory_management app = new park_inventory_management();
    app.setTitle("Parking Inventory Management");
    app.setVisible(true);
    app.setSize(1000, 700);
    app.setVisible(true);
    app.setLocationRelativeTo(null);
    parking_slot.this.dispose();
}
}
```

```

static parking_slot app = new parking_slot();

public static void main(String[] args) {

    app.setTitle("Parking Slot");

    app.setSize(700, 600);

    app.setVisible(true);

    app.setLocationRelativeTo(null);

    app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

}

```

```

}

```

Admin Registration

Java Developer SE - Online Parking Reservation/src/adminmanagement/reg.java - D:\Archim Pameroyan\Programming\Eclipse\Java Developer SE - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer: login.java, menu.java, customer_pro..., billingmanag..., park_invento..., parking_inve..., parking_slot..., reg.java


1 package adminmanagement;

Admin Registration Form

Manalad Group of Company

Time 5 : 28 : 55 Date 2020 / 3 /

ADMIN PROFILE

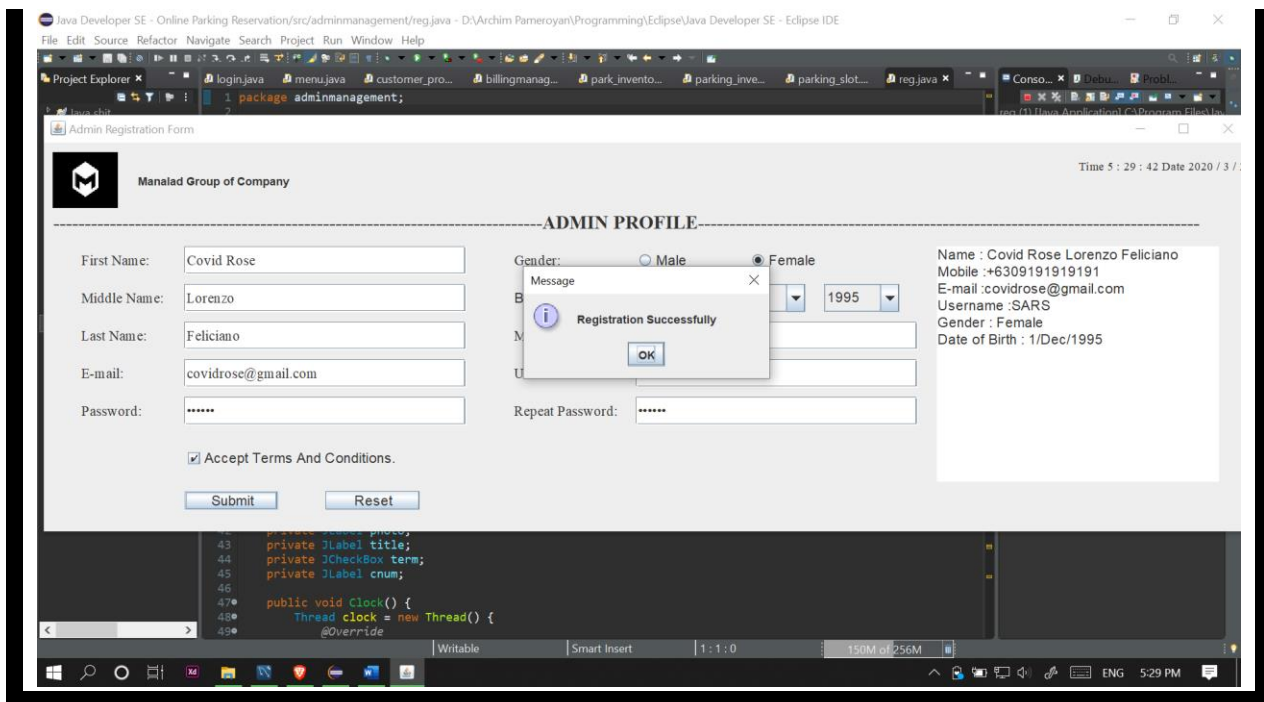
First Name:	<input type="text" value="Covid Rose"/>	Gender:	<input type="radio"/> Male <input checked="" type="radio"/> Female	
Middle Name:	<input type="text" value="Lorenzo"/>	Birthdate:	Dec 1 1995	
Last Name:	<input type="text" value="Feliciano"/>	Mobile #:	+63 09191919191	
E-mail:	<input type="text" value="covidrose@gmail.com"/>	Username:	<input type="text" value="SARS"/>	
Password:	<input type="password" value="*****"/>	Repeat Password:	<input type="password" value="*****"/>	
<input type="checkbox"/> Accept Terms And Conditions.				

```

43 private JLabel photo;
44 private JLabel title;
45 private JCheckBox term;
46 private JLabel cnum;
47
48 public void Clock() {
49     Thread clock = new Thread() {
50         @Override

```

Writable Smart Insert 1:1:0 134M/256M ENG 5:28 PM



```
package adminmanagement;
```

```
import javax.swing.*;
```

```
import javax.swing.table.DefaultTableModel;
```

```
import DatabaseConnection.ConnectionDB;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.sql.Connection;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import java.sql.ResultSetMetaData;
```

```
import java.sql.Statement;
```

```
import java.util.Calendar;
```

```
import java.util.GregorianCalendar;
```

```
public class reg extends JFrame implements ActionListener {
```

```
    String tpass1, trpass2;
```

```
    public char pass1[], pass2[];
```

```

public JLabel adminDet, fnam, mnam, lnam, bdate, mno, email, gender, unam, pass,
rpass;

public JTextField tfnam, tmnam, tlnam, tmno, temail, tunam;

public JRadioButton male, female;

public ButtonGroup gengp;

public JComboBox<?> date, month, year;

public JPasswordField tpass, trpass;

public JButton sub, reset1;

public JTextArea tout;

public Connection conn = null;

public Statement st = null;

public PreparedStatement ps = null;

public ResultSet rs = null;

public DefaultTableModel dm = null;

public ResultSetMetaData rsmd = null;

Font f1 = new Font("Times New Roman", Font.PLAIN, 15);
Font f2 = new Font("Times New Roman", Font.BOLD, 20);
Font f3 = new Font("Times New Roman", Font.PLAIN, 13);
Font f4 = new Font("Times New Roman", Font.CENTER_BASELINE, 15);
// Color c1 = new Color.

private JLabel lblClock;

private ImageIcon icon;

private JLabel label;

private JLabel photo;

private JLabel title;

private JCheckBox term;

private JLabel cnum;

public void Clock() {

```

```

Thread clock = new Thread() {
    @Override
    public void run() {
        try {
            while (true) {
                Calendar cal = new GregorianCalendar();
                int day = cal.get(Calendar.DAY_OF_MONTH);
                int month = cal.get(Calendar.MONTH);
                int year = cal.get(Calendar.YEAR);

                int second = cal.get(Calendar.SECOND);
                int minute = cal.get(Calendar.MINUTE);
                int hour = cal.get(Calendar.HOUR);

                lblClock.setText("Time " + hour + " : " + minute
+ " : " + second + " Date " + year + " / "
                                + (month + 1) + " / " + day);
                sleep(1000);
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
};
clock.start();
}

public reg() {
    conn = ConnectionDB.getConnection();
    icon = new ImageIcon("src/m.jpg");
}

```

```
Container con = getContentPane();
```

```
con.setLayout(null);
```

```
photo = new JLabel() {
```

```
    public void paintComponent(Graphics g) {
```

```
        g.drawImage(icon.getImage(), 0, 0, null);
```

```
        super.paintComponent(g);
```

```
    }
```

```
};
```

```
photo.setOpaque(false);
```

```
con.add(photo);
```

```
photo.setBounds(10, 10, 100, 100);
```

```
con.add(lblClock = new JLabel(""));
```

```
lblClock.setBounds(1100, 10, 230, 30);
```

```
lblClock.setFont(f3);
```

```
title = new JLabel("Manalad Group of Company");
```

```
title.setBounds(100, -10, 200, 100);
```

```
con.add(title);
```

```
admindet = new JLabel(
```

```
    "-----"
```

```
ADMIN PROFILE-----");
```

```
admindet.setBounds(10, 60, 1300, 50);
```

```
admindet.setFont(f2);
```

```
con.add(admindet);
```

```
fnam = new JLabel("First Name:");  
fnam.setBounds(40, 110, 80, 30);  
fnam.setFont(f1);  
con.add(fnam);
```

```
mnam = new JLabel("Middle Name:");  
mnam.setBounds(40, 150, 90, 30);  
mnam.setFont(f1);  
con.add(mnam);
```

```
lnam = new JLabel("Last Name:");  
lnam.setBounds(40, 190, 80, 30);  
lnam.setFont(f1);  
con.add(lnam);
```

```
gender = new JLabel("Gender:");  
gender.setBounds(500, 110, 90, 30);  
gender.setFont(f1);  
con.add(gender);
```

```
mno = new JLabel("Mobile #:");  
mno.setBounds(500, 190, 80, 30);  
mno.setFont(f1);  
con.add(mno);
```

```
cnum = new JLabel("+63");  
cnum.setBounds(630, 190, 30, 30);  
cnum.setFont(f1);  
con.add(cnum);
```

```
unam = new JLabel("Username:");
unam.setBounds(500, 230, 80, 30);
unam.setFont(f1);
con.add(unam);
```

```
rpass = new JLabel("Repeat Password:");
rpass.setBounds(500, 270, 120, 30);
rpass.setFont(f1);
con.add(rpass);
```

```
email = new JLabel("E-mail:");
email.setBounds(40, 230, 80, 30);
email.setFont(f1);
con.add(email);
```

```
pass = new JLabel("Password:");
pass.setBounds(40, 270, 80, 30);
pass.setFont(f1);
con.add(pass);
```

```
String Select[] = { "Select -/-", "Hatchback", " Sedan", "MPV", "SUV",
"Crossover", " Coupe", "Convertible",
                "Sports Car", "Utility", "Compact", "Mini Van", "Van", "Pick-Up
Truck" };
```

```
term = new JCheckBox("Accept Terms And Conditions.");
term.setFont(new Font("Arial", Font.PLAIN, 15));
term.setBounds(150, 320, 300, 30);
con.add(term);
```



```
sub = new JButton("Submit");
sub.setFont(new Font("Arial", Font.PLAIN, 15));
sub.setBounds(150, 370, 100, 20);
sub.addActionListener(this);
con.add(sub);

reset1 = new JButton("Reset");
reset1.setFont(new Font("Arial", Font.PLAIN, 15));
reset1.setBounds(300, 370, 100, 20);
reset1.addActionListener(this);
con.add(reset1);

tfnam = new JTextField();
tfnam.setBounds(150, 110, 300, 30);
tfnam.setFont(f1);
con.add(tfnam);

tmnam = new JTextField();
tmnam.setBounds(150, 150, 300, 30);
tmnam.setFont(f1);
con.add(tmnam);

tlnam = new JTextField();
tlnam.setBounds(150, 190, 300, 30);
tlnam.setFont(f1);
con.add(tlnam);

tmno = new JTextField();
```

```
tmno.setBounds(670, 190, 260, 30);
```

```
tmno.setFont(f1);
```

```
con.add(tmno);
```

```
temail = new JTextField();
```

```
temail.setBounds(150, 230, 300, 30);
```

```
temail.setFont(f1);
```

```
con.add(temail);
```

```
tunam = new JTextField();
```

```
tunam.setBounds(630, 230, 300, 30);
```

```
tunam.setFont(f1);
```

```
con.add(tunam);
```

```
tpass = new JPasswordField();
```

```
tpass.setBounds(150, 270, 300, 30);
```

```
tpass.setFont(f1);
```

```
con.add(tpass);
```

```
trpass = new JPasswordField();
```

```
trpass.setBounds(630, 270, 300, 30);
```

```
trpass.setFont(f1);
```

```
con.add(trpass);
```

```
tout = new JTextArea();
```

```
tout.setFont(new Font("Arial", Font.PLAIN, 15));
```

```
tout.setBounds(950, 110, 300, 250);
```

```
tout.setLineWrap(true);
```

```
tout.setEditable(false);
```

```
con.add(tout);
```

```
male = new JRadioButton("Male");  
male.setFont(new Font("Arial", Font.PLAIN, 15));  
male.setSelected(true);  
male.setBounds(630, 110, 80, 30);  
male.setActionCommand("male");  
con.add(male);
```

```
female = new JRadioButton("Female");  
female.setFont(new Font("Arial", Font.PLAIN, 15));  
female.setSelected(false);  
female.setBounds(750, 110, 80, 30);  
female.setActionCommand("female");  
con.add(female);
```

```
gengp = new ButtonGroup();  
gengp.add(male);  
gengp.add(female);
```

```
bdate = new JLabel("Birthdate:");  
bdate.setFont(new Font("Arial", Font.PLAIN, 15));  
bdate.setBounds(500, 150, 80, 30);  
con.add(bdate);
```

```
String dates[] = { "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12",  
"13", "14", "15", "16", "17",  
"18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28",  
"29", "30", "31" };  
date = new JComboBox(dates);
```

```
date.setFont(new Font("Arial", Font.PLAIN, 15));
```

```
date.setBounds(730, 150, 80, 30);
```

```
con.add(date);
```

```
String months[] = { "Jan", "feb", "Mar", "Apr", "May", "Jun", "July", "Aug",  
"Sep", "Oct", "Nov", "Dec" };
```

```
month = new JComboBox(months);
```

```
month.setFont(new Font("Arial", Font.PLAIN, 15));
```

```
month.setBounds(630, 150, 80, 30);
```

```
con.add(month);
```

```
String years[] = { "1930", "1931", "1932", "1933", "1934", "1935", "1936",  
"1937", "1938", "1939", "1940",
```

```
    "1941", "1942", "1943", "1944", "1945", "1946", "1947",  
"1948", "1949", "1950", "1951", "1952", "1953",
```

```
    "1954", "1955", "1956", "1957", "1958", "1959", "1960",  
"1961", "1962", "1963", "1964", "1965", "1966",
```

```
    "1967", "1968", "1969", "1970", "1971", "1972", "1973",  
"1974", "1975", "1976", "1977", "1978", "1979",
```

```
    "1980", "1981", "1982", "1983", "1984", "1985", "1986",  
"1987", "1988", "1989", "1990", "1991", "1992",
```

```
    "1993", "1994", "1995", "1996", "1997", "1998", "1999",  
"2000", "2001", "2002", "2003", "2004", "2005",
```

```
    "2006", "2007", "2008", "2009", "2010", "2011", "2012",  
"2013", "2014", "2015", "2016", "2017", "2018",
```

```
    "2019", "2020" };
```

```
year = new JComboBox(years);
```

```
year.setFont(new Font("Arial", Font.PLAIN, 15));
```

```
year.setBounds(830, 150, 80, 30);
```

```
con.add(year);
```

```
Clock();
```

```
}
```

```
@Override
```

```
public void actionPerformed(ActionEvent e) {  
    // TODO Auto-generated method stub  
    pass1 = tpass.getPassword();  
    tpass1 = String.valueOf(pass1);  
    pass2 = trpass.getPassword();  
    trpass2 = String.valueOf(pass2);  
    if (e.getSource() == sub) {  
        if (term.isSelected()) {  
  
            String data1;  
            String data = "Name : " + tfnam.getText() + " " +  
tmnam.getText() + " " + tlnam.getText() + "\n"  
                + "Mobile :+63" + tmno.getText() + "\n" + "E-  
mail : " + temail.getText() + "\n" + "Username :"  
                + tunam.getText() + "\n";  
            if (male.isSelected())  
                data1 = "Gender : Male" + "\n";  
            else  
                data1 = "Gender : Female" + "\n";  
            String data2 = "Date of Birth : " + (String)  
date.getSelectedItemAt() + "/"  
                + (String) month.getSelectedItemAt() + "/" +  
(String) year.getSelectedItemAt() + "\n";  
  
            String data3 = "\n";  
            tout.setText(data + data1 + data2 + data3);  
            tout.setEditable(false);  
        }  
    }  
}
```

```

        int dialog = JOptionPane.showConfirmDialog(null,
            "Check the summary if there's missing, if yes
click No", "WARNING", JOptionPane.YES_NO_OPTION,
            JOptionPane.WARNING_MESSAGE);
        if (dialog == JOptionPane.YES_OPTION) {
            JOptionPane.showMessageDialog(this, "Registration
Successfully");

            try {
                String query = "insert into adminaccount
(FirstName,MiddleName,LastName,Email,Gender,Birthdate,MobileNumber,Username,Pass
word,RepeatPassword) values (?, ?, ?, ?, ?, ?, ?, ?, ?) ";

                ps = conn.prepareStatement(query);
                ps.setString(1, tfnam.getText());
                ps.setString(2, tmnam.getText());
                ps.setString(3, tlnam.getText());
                ps.setString(4, temail.getText());
                ps.setString(5,
gengp.getSelection().getActionCommand());
                ps.setString(6, (String) date.getSelectedItemAt());
                ps.setString(6, (String)
month.getSelectedItemAt());

                ps.setString(6, (String) year.getSelectedItemAt());
                ps.setString(7, tmno.getText());
                ps.setString(8, tunam.getText());
                ps.setString(9, tpass.getText());
                ps.setString(10, trpass.getText());

                ps.execute();
                JOptionPane.showMessageDialog(null, "Data
Saved");

```

```

        ps.close();

    } catch (Exception ex) {
        ex.printStackTrace();

    }

    // pabalik ng login.java
    login app = new login();
    app.setTitle("Admin Login");
    app.setVisible(true);
    app.setSize(450, 600);
    app.setLocationRelativeTo(null);
    reg.this.dispose();

    }

    if (dialog == JOptionPane.NO_OPTION) {
        reg.this.show();

    }

    } else {
        tout.setText("");
        JOptionPane.showMessageDialog(this, "Please accept the" + "
terms & conditions!", "Alert",
                                JOptionPane.WARNING_MESSAGE);

    }

    }

else if (e.getSource() == reset1) {
    String def = "";
    tfnam.setText(def);
    tmnam.setText(def);

```

```
        tlnam.setText(def);
        temail.setText(def);
        tunam.setText(def);
        tpass.setText(def);
        trpass.setText(def);
        tmno.setText(def);
        tout.setText(def);
        term.setSelected(false);
        date.setSelectedIndex(0);
        month.setSelectedIndex(0);
        year.setSelectedIndex(0);
    }
}

//main

static reg app = new reg();

public static void main(String[] args) {
    app.setTitle("Admin Registration Form");
    app.setSize(1300, 450);
    app.setVisible(true);
    app.setLocationRelativeTo(null);
    app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

}

}
```


User Management

User Registration

Java Developer SE - Online IDE

Registration Form

Manalad Group of Company

Time 5 : 39 : 20 Date 2020 / 3 / 29

CUSTOMER PROFILE

First Name: Lucy
Middle Name: Reyes
Last Name: Pameroyan
E-mail: lucy@gmail.com
Password: *****
Gender: ☐ Male ☒ Female
Birthdate: feb 14 2020
Mobile #: +63 09125679340
Username: StrawberryShortCake
Repeat Password: *****

CAR PROFILE

PLATE #: l0c1y
Brand: Ferrari
Color: black or Choose Color
Type of Car: SUV
☒ Accept Terms And Conditions
Submit Reset

SUMMARY PROFILE

Java Developer SE - Online IDE

Registration Form

Manalad Group of Company

Time 5 : 39 : 36 Date 2020 / 3 / 29

CUSTOMER PROFILE

First Name: Lucy
Middle Name: Reyes
Last Name: Pameroyan
E-mail: lucy@gmail.com
Password: *****
Gender: ☐ Male ☒ Female
Birthdate: feb 14 2020
Mobile #: +63 09125679340
Username: StrawberryShortCake
Repeat Password: *****

CAR PROFILE

PLATE #: l0c1y
Brand: Ferrari
Color: black or Choose Color
Type of Car: SUV
☒ Accept Terms And Conditions
Submit Reset

SUMMARY PROFILE

ADMIN INFORMATION

Name : Lucy Reyes Pameroyan
Mobile :+6309125679340
E-mail :lucy@gmail.com
Username :StrawberryShortCake
Gender : Female
Date of Birth : 14/feb/2020

CAR INFORMATION

PLATE # :l0c1y
Brand :Ferrari
Color of the Car :black
Type of Car : SUV

Message
Registration Successfully
OK

package usermanagement;

import java.awt.Container;

```
import javax.swing.*;

import javax.swing.table.DefaultTableModel;


import DatabaseConnection.ConnectionDB;


import java.awt.*;
import java.awt.event.*;
import java.util.Calendar;
import java.util.GregorianCalendar;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.Statement;


public class reg extends JFrame implements ActionListener {

    String tpass1, trpass2;

    public char pass1[], pass2[];

    public JLabel custdet, fnam, mnam, lnam, bdate, mno, type, email, platno, brand,
color, gender, unam, pass, rpass,

        carprof, or;

    public JTextField tfnam, tmnam, tlnam, tmno, temail, tplatno, tbrand, tcolor, tunam;
    public JRadioButton male, female;
    public ButtonGroup gengp;
    public JComboBox<?> date, month, year;
    public JPasswordField tpass, trpass;
    public JButton sub, reset1;
    public JTextArea tout;
    public Connection conn = null;
    public Statement st = null;
```

```

public PreparedStatement ps = null;

public ResultSet rs = null;

public DefaultTableModel dm = null;

public ResultSetMetaData rsmd = null;

Font f1 = new Font("Times New Roman", Font.PLAIN, 15);
Font f2 = new Font("Times New Roman", Font.BOLD, 20);
Font f3 = new Font("Times New Roman", Font.PLAIN, 13);
Font f4 = new Font("Times New Roman", Font.CENTER_BASELINE, 15);
// Color c1 = new Color.

private JLabel lblClock;
private ImageIcon icon;
private JLabel label;
private JLabel photo;
private JLabel title;
private JButton col;
private JComboBox<?> combo;
private JCheckBox term;
private JLabel cnum;

public void Clock() {
    Thread clock = new Thread() {
        @Override

        public void run() {
            try {
                while (true) {
                    Calendar cal = new GregorianCalendar();
                    int day = cal.get(Calendar.DAY_OF_MONTH);

```

```

        int month = cal.get(Calendar.MONTH);
        int year = cal.get(Calendar.YEAR);

        int second = cal.get(Calendar.SECOND);
        int minute = cal.get(Calendar.MINUTE);
        int hour = cal.get(Calendar.HOUR);

        lblClock.setText("Time " + hour + " : " + minute
+ " : " + second + " Date " + year + " / "
                                + (month + 1) + " / " + day);

        sleep(1000);
    }
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

clock.start();
}

public reg() {
    conn = ConnectionDB.getConnection();

    icon = new ImageIcon("src/m.jpg");

    Container con = getContentPane();
    con.setLayout(null);

    photo = new JLabel() {
        public void paintComponent(Graphics g) {

```

```

        g.drawImage(icon.getImage(), 0, 0, null);
        super.paintComponent(g);
    }
};

photo.setOpaque(false);
con.add(photo);
photo.setBounds(10, 10, 100, 100);

con.add(lblClock = new JLabel(""));
lblClock.setBounds(730, 10, 230, 30);
lblClock.setFont(f3);

title = new JLabel("Manalad Group of Company");
title.setBounds(100, -10, 200, 100);
con.add(title);

custdet = new JLabel(
    "-----CUSTOMER PROFILE-----");

custdet.setBounds(10, 60, 1200, 50);
custdet.setFont(f2);
con.add(custdet);

fnam = new JLabel("First Name:");
fnam.setBounds(40, 110, 80, 30);
fnam.setFont(f1);
con.add(fnam);

mnam = new JLabel("Middle Name:");

```

```
mnam.setBounds(40, 150, 90, 30);  
mnam.setFont(f1);  
con.add(mnam);
```

```
lnam = new JLabel("Last Name:");  
lnam.setBounds(40, 190, 80, 30);  
lnam.setFont(f1);  
con.add(lnam);
```

```
gender = new JLabel("Gender:");  
gender.setBounds(500, 110, 90, 30);  
gender.setFont(f1);  
con.add(gender);
```

```
mno = new JLabel("Mobile #:");  
mno.setBounds(500, 190, 80, 30);  
mno.setFont(f1);  
con.add(mno);
```

```
cnum = new JLabel("+63");  
cnum.setBounds(630, 190, 30, 30);  
cnum.setFont(f1);  
con.add(cnum);
```

```
unam = new JLabel("Username:");  
unam.setBounds(500, 230, 80, 30);  
unam.setFont(f1);  
con.add(unam);
```

```
rpass = new JLabel("Repeat Password:");  
rpass.setBounds(500, 270, 120, 30);  
rpass.setFont(f1);  
con.add(rpass);
```

```
email = new JLabel("E-mail:");  
email.setBounds(40, 230, 80, 30);  
email.setFont(f1);  
con.add(email);
```

```
pass = new JLabel("Password:");  
pass.setBounds(40, 270, 80, 30);  
pass.setFont(f1);  
con.add(pass);
```

```
carprof = new JLabel(  
    "-----CAR PROFILE-----  
SUMMARY PROFILE");
```

```
carprof.setBounds(10, 330, 1200, 50);  
carprof.setFont(f2);  
con.add(carprof);
```

```
platno = new JLabel("PLATE #:");  
platno.setBounds(40, 370, 80, 30);  
platno.setFont(f1);  
con.add(platno);
```

```
brand = new JLabel("Brand:");  
brand.setBounds(40, 410, 80, 30);  
brand.setFont(f1);
```

```
con.add(brand);
```

```
color = new JLabel("Color:");  
color.setBounds(40, 450, 80, 30);  
color.setFont(f1);  
con.add(color);
```

```
or = new JLabel("or");  
or.setBounds(310, 450, 20, 30);  
or.setFont(f1);  
con.add(or);
```

```
col = new JButton("Choose Color");  
col.setBounds(330, 450, 120, 30);  
col.setFont(f1);  
col.addActionListener(this);  
con.add(col);
```

```
type = new JLabel("Type of Car:");  
type.setBounds(40, 490, 100, 30);  
type.setFont(f1);  
con.add(type);
```

```
String Select[] = { "Select -/-", "Hatchback", " Sedan", "MPV", "SUV",  
"Crossover", " Coupe", "Convertible",  
"Sports Car", "Utility", "Compact", "Mini Van", "Van", "Pick-Up  
Truck" };
```

```
combo = new JComboBox(Select);  
combo.setFont(new Font("Arial", Font.PLAIN, 15));  
combo.setBounds(150, 490, 100, 30);
```



```
con.add(combo);
```

```
term = new JCheckBox("Accept Terms And Conditions.");
```

```
term.setFont(new Font("Arial", Font.PLAIN, 15));
```

```
term.setBounds(150, 530, 300, 30);
```

```
con.add(term);
```

```
sub = new JButton("Submit");
```

```
sub.setFont(new Font("Arial", Font.PLAIN, 15));
```

```
sub.setBounds(150, 570, 100, 20);
```

```
sub.addActionListener(this);
```

```
con.add(sub);
```

```
reset1 = new JButton("Reset");
```

```
reset1.setFont(new Font("Arial", Font.PLAIN, 15));
```

```
reset1.setBounds(300, 570, 100, 20);
```

```
reset1.addActionListener(this);
```

```
con.add(reset1);
```

```
tfnam = new JTextField();
```

```
tfnam.setBounds(150, 110, 300, 30);
```

```
tfnam.setFont(f1);
```

```
con.add(tfnam);
```

```
tmnam = new JTextField();
```

```
tmnam.setBounds(150, 150, 300, 30);
```

```
tmnam.setFont(f1);
```

```
con.add(tmnam);
```

```
tlnam = new JTextField();  
tlnam.setBounds(150, 190, 300, 30);  
tlnam.setFont(f1);  
con.add(tlnam);
```

```
tmno = new JTextField();  
tmno.setBounds(670, 190, 260, 30);  
tmno.setFont(f1);  
con.add(tmno);
```

```
temail = new JTextField();  
temail.setBounds(150, 230, 300, 30);  
temail.setFont(f1);  
con.add(temail);
```

```
tunam = new JTextField();  
tunam.setBounds(630, 230, 300, 30);  
tunam.setFont(f1);  
con.add(tunam);
```

```
tpass = new JPasswordField();  
tpass.setBounds(150, 270, 300, 30);  
tpass.setFont(f1);  
con.add(tpass);
```

```
trpass = new JPasswordField();  
trpass.setBounds(630, 270, 300, 30);  
trpass.setFont(f1);  
con.add(trpass);
```

```
tout = new JTextArea();  
tout.setFont(new Font("Arial", Font.PLAIN, 15));  
tout.setBounds(630, 370, 300, 300);  
tout.setLineWrap(true);  
tout.setEditable(false);  
con.add(tout);
```

```
tplatno = new JTextField();  
tplatno.setBounds(150, 370, 300, 30);  
tplatno.setFont(f1);  
con.add(tplatno);
```

```
tbrand = new JTextField();  
tbrand.setBounds(150, 410, 300, 30);  
tbrand.setFont(f1);  
con.add(tbrand);
```

```
tcolor = new JTextField();  
tcolor.setBounds(150, 450, 150, 30);  
tcolor.setFont(f1);  
con.add(tcolor);
```

```
male = new JRadioButton("Male");  
male.setFont(new Font("Arial", Font.PLAIN, 15));  
male.setSelected(true);  
male.setBounds(630, 110, 80, 30);  
male.setActionCommand("male");  
con.add(male);
```

```
female = new JRadioButton("Female");  
female.setFont(new Font("Arial", Font.PLAIN, 15));  
female.setSelected(false);  
female.setBounds(750, 110, 80, 30);  
female.setActionCommand("female");  
con.add(female);
```

```
gengp = new ButtonGroup();  
gengp.add(male);  
gengp.add(female);
```

```
bdate = new JLabel("Birthdate:");  
bdate.setFont(new Font("Arial", Font.PLAIN, 15));  
bdate.setBounds(500, 150, 80, 30);  
con.add(bdate);
```

```
String dates[] = { "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12",  
"13", "14", "15", "16", "17",  
"18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28",  
"29", "30", "31" };
```

```
date = new JComboBox(dates);  
date.setFont(new Font("Arial", Font.PLAIN, 15));  
date.setBounds(730, 150, 80, 30);  
con.add(date);
```

```
String months[] = { "Jan", "feb", "Mar", "Apr", "May", "Jun", "July", "Aug",  
"Sep", "Oct", "Nov", "Dec" };  
month = new JComboBox(months);  
month.setFont(new Font("Arial", Font.PLAIN, 15));
```

```

        month.setBounds(630, 150, 80, 30);

        con.add(month);

        String years[] = { "1930", "1931", "1932", "1933", "1934", "1935", "1936",
"1937", "1938", "1939", "1940",
        "1941", "1942", "1943", "1944", "1945", "1946", "1947",
"1948", "1949", "1950", "1951", "1952", "1953",
        "1954", "1955", "1956", "1957", "1958", "1959", "1960",
"1961", "1962", "1963", "1964", "1965", "1966",
        "1967", "1968", "1969", "1970", "1971", "1972", "1973",
"1974", "1975", "1976", "1977", "1978", "1979",
        "1980", "1981", "1982", "1983", "1984", "1985", "1986",
"1987", "1988", "1989", "1990", "1991", "1992",
        "1993", "1994", "1995", "1996", "1997", "1998", "1999",
"2000", "2001", "2002", "2003", "2004", "2005",
        "2006", "2007", "2008", "2009", "2010", "2011", "2012",
"2013", "2014", "2015", "2016", "2017", "2018",
        "2019", "2020" };

        year = new JComboBox(years);

        year.setFont(new Font("Arial", Font.PLAIN, 15));

        year.setBounds(830, 150, 80, 30);

        con.add(year);

        Clock();

    }

    @Override

    public void actionPerformed(ActionEvent e) {

        // TODO Auto-generated method stub

        pass1 = tpass.getPassword();

        tpass1 = String.valueOf(pass1);

```

```

pass2 = trpass.getPassword();
trpass2 = String.valueOf(pass2);
if (e.getSource() == sub) {

    if (term.isSelected()) {

        String data1;

        String data = "          ADMIN INFORMATION  " + "\n" + "\n"
+ "Name : " + tfnam.getText() + " "
                                + tmnam.getText() + " " + tlnam.getText() + "\n"
+ "Mobile :+63" + tmno.getText() + "\n"
                                + "E-mail :" + temail.getText() + "\n" +
"Username :" + tunam.getText() + "\n";

        if (male.isSelected())
            data1 = "Gender : Male" + "\n";
        else
            data1 = "Gender : Female" + "\n";

        String data2 = "Date of Birth : " + (String)
date.getSelectedItemAt() + "/"
                                + (String) month.getSelectedItemAt() + "/" +
(String) year.getSelectedItemAt() + "\n" + "\n"
                                + "          CAR INFORMATION  " + "\n" + "\n"
+ "PLATE # :" + tplatno.getText() + "\n"
                                + "Brand :" + tbrand.getText() + "\n" + "Color of
the Car :" + tcolor.getText() + "\n";

        ;

        String data3 = "Type of Car : " + (String)
combo.getSelectedItemAt() + "\n";

        tout.setText(data + data1 + data2 + data3);
        tout.setEditable(false);

        JOptionPane.showMessageDialog(this, "Registration
Successfully");

```

```

try {

    String query = "insert into useraccount
(FirstName,MiddleName,LastName,Email,Gender,Birthdate,MobileNumber,Username,Pass
word,RepeatPassword,PlateNumber,Brand,Color,Type) values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?) ";

    ps = conn.prepareStatement(query);
    ps.setString(1, tfnam.getText());
    ps.setString(2, tmnam.getText());
    ps.setString(3, tlnam.getText());
    ps.setString(4, temail.getText());
    ps.setString(5,
gengp.getSelection().getActionCommand());

    ps.setString(6, (String)date.getSelectedItemAt());
    ps.setString(6, (String)month.getSelectedItemAt());
    ps.setString(6, (String)year.getSelectedItemAt());
    ps.setString(7, tmno.getText());
    ps.setString(8, tunam.getText());
    ps.setString(9, tpass.getText());
    ps.setString(10, trpass.getText());
    ps.setString(11, tplatno.getText());
    ps.setString(12, tbrand.getText());
    ps.setString(13, tcolor.getText());
    ps.setString(14, (String)combo.getSelectedItemAt());

    ps.execute();

    JOptionPane.showMessageDialog(null, "Data Saved");
}

```

```

        ps.close();
    } catch (Exception ex) {
        ex.printStackTrace();
    }

    // pabalik ng login.java
    login app = new login();
    app.setVisible(true);
    app.setSize(450, 600);
    app.setVisible(true);
    app.setLocationRelativeTo(null);
    reg.this.dispose();

    } else {
        tout.setText("");
        JOptionPane.showMessageDialog(this, "Please accept the" + "
terms & conditions!", "Alert",
                                JOptionPane.WARNING_MESSAGE);
    }
}

else if (e.getSource() == reset1) {
    String def = "";
    tfnam.setText(def);
    tmnam.setText(def);
    tlnam.setText(def);
    temail.setText(def);
    tunam.setText(def);
    tpass.setText(def);

```



```
        trpass.setText(def);
        tplatno.setText(def);
        tbrand.setText(def);
        tcolor.setText(def);
        tmno.setText(def);
        tout.setText(def);
        term.setSelected(false);
        date.setSelectedIndex(0);
        month.setSelectedIndex(0);
        year.setSelectedIndex(0);
        combo.setSelectedIndex(0);
    }

}

//main

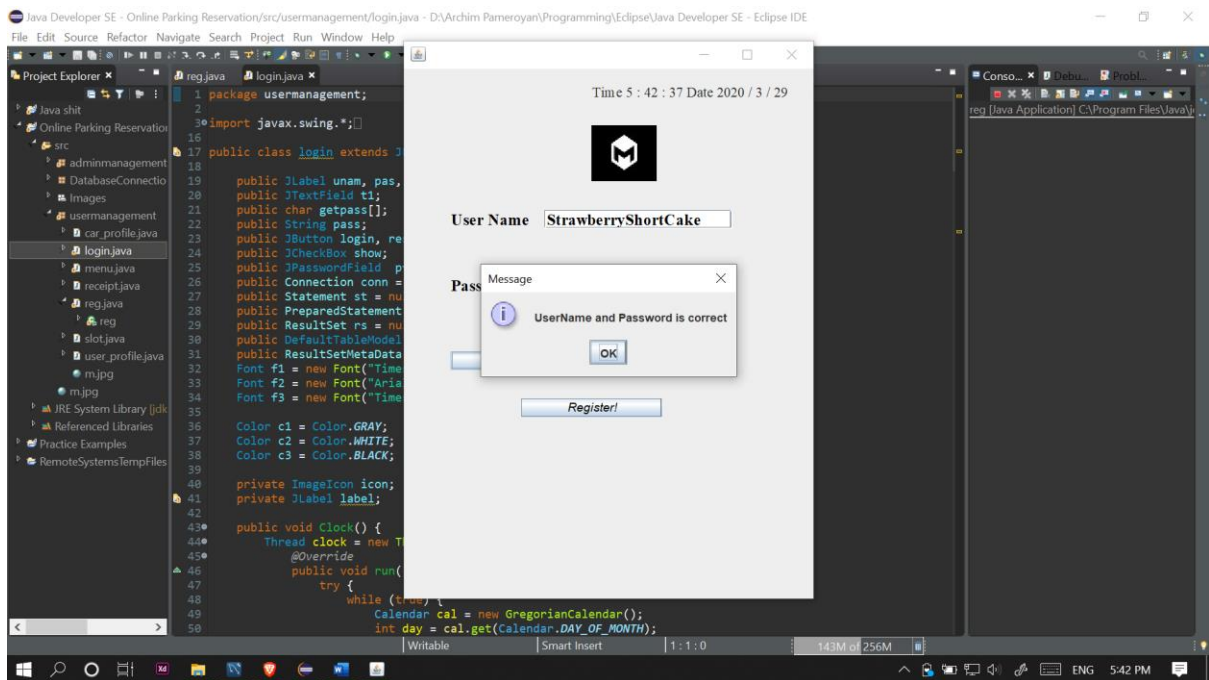
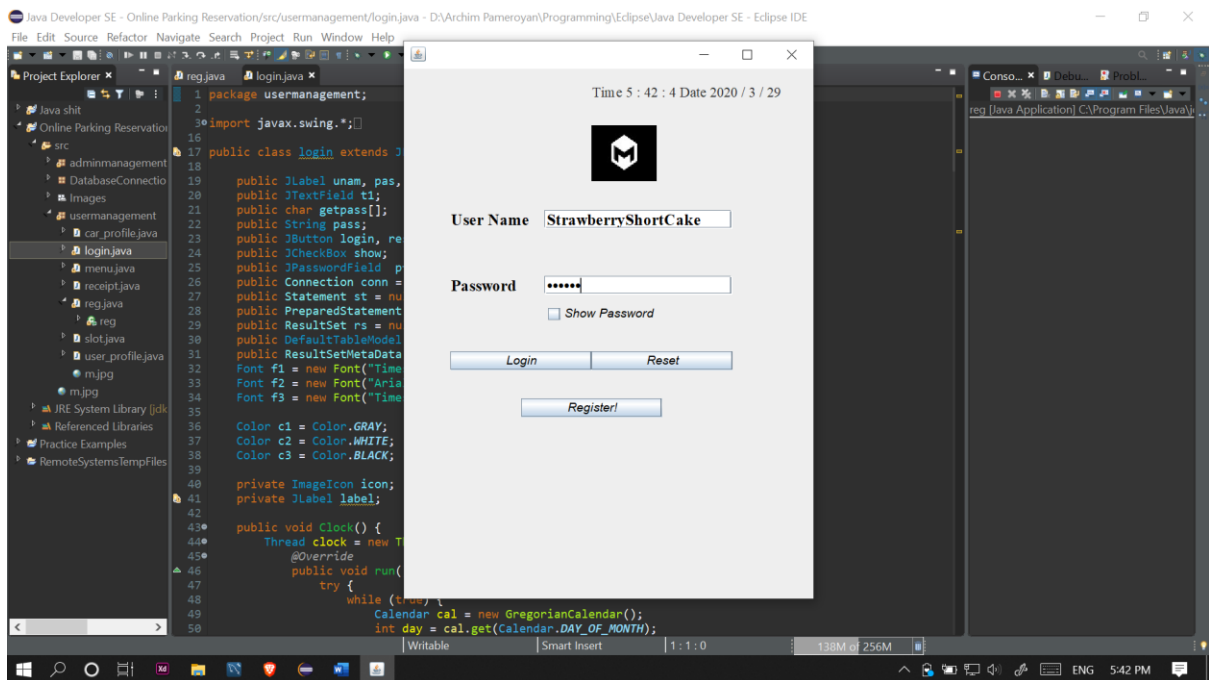
static reg app = new reg();

public static void main(String[] args) {
    app.setTitle("Registration Form");
    app.setSize(950, 750);
    app.setVisible(true);
    app.setLocationRelativeTo(null);
    app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

}

}
```

User Login



```
package usermanagement;
```

```
import javax.swing.*;
```

```
import javax.swing.table.DefaultTableModel;
```

```
import java.awt.*;
import java.awt.event.*;
import java.util.Calendar;
import java.util.GregorianCalendar;
import DatabaseConnection.ConnectionDB;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.Statement;

public class login extends JFrame implements ActionListener {

    public JLabel unam, pas, lblClock, photo;
    public JTextField t1;
    public char getpass[];
    public String pass;
    public JButton login, res, reg1;
    public JCheckBox show;
    public JPasswordField pfpwd;
    public Connection conn = null;
    public Statement st = null;
    public PreparedStatement ps = null;
    public ResultSet rs = null;
    public DefaultTableModel dm = null;
    public ResultSetMetaData rsmd = null;
    Font f1 = new Font("Times New Roman", Font.BOLD, 17);
    Font f2 = new Font("Arial", Font.ITALIC, 13);
    Font f3 = new Font("Times New Roman", Font.PLAIN, 15);
```

```

Color c1 = Color.GRAY;

Color c2 = Color.WHITE;

Color c3 = Color.BLACK;


private ImageIcon icon;

private JLabel label;


public void Clock() {
    Thread clock = new Thread() {
        @Override
        public void run() {
            try {
                while (true) {
                    Calendar cal = new GregorianCalendar();
                    int day = cal.get(Calendar.DAY_OF_MONTH);
                    int month = cal.get(Calendar.MONTH);
                    int year = cal.get(Calendar.YEAR);

                    int second = cal.get(Calendar.SECOND);
                    int minute = cal.get(Calendar.MINUTE);
                    int hour = cal.get(Calendar.HOUR);

                    lblClock.setText("Time " + hour + " : " + minute
+ " : " + second + " Date " + year + " / "
                                + (month + 1) + " / " + day);
                    sleep(1000);
                }
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    };
}

```

```

        }

    }

};

clock.start();

}

public login() {

    conn = ConnectionDB.getConnection();

    icon = new ImageIcon("src/m.jpg");

    Container con = getContentPane();
    con.setLayout(null);

    photo = new JLabel() {

        public void paintComponent(Graphics g) {

            g.drawImage(icon.getImage(), 0, 0, null);

            super.paintComponent(g);

        }

    };

    photo.setOpaque(false);
    con.add(photo);
    photo.setBounds(200, 60, 150, 100);

    con.add(lblClock = new JLabel(""));
    lblClock.setBounds(200, 10, 230, 30);
    lblClock.setFont(f3);

    unam = new JLabel("User Name");

```

```
pas = new JLabel("Password");
```

```
t1 = new JTextField();
```

```
pfpwd = new JPasswordField();
```

```
login = new JButton("Login");
```

```
show = new JCheckBox("Show Password");
```

```
res = new JButton("Reset");
```

```
reg1 = new JButton("Register!");
```

```
login.addActionListener(this);
```

```
show.addActionListener(this);
```

```
res.addActionListener(this);
```

```
reg1.addActionListener(this);
```

```
unam.setBounds(50, 150, 100, 20);
```

```
pas.setBounds(50, 220, 100, 20);
```

```
t1.setBounds(150, 150, 200, 20);
```

```
pfpwd.setBounds(150, 220, 200, 20);
```

```
show.setBounds(150, 250, 150, 20);
```

```
login.setBounds(50, 300, 150, 20);
```

```
res.setBounds(200, 300, 150, 20);
```

```
reg1.setBounds(125, 350, 150, 20);
```

```
unam.setFont(f1);
```

```
pas.setFont(f1);
```

```
t1.setFont(f1);
```

```
pfpwd.setFont(f1);
```

```
login.setFont(f2);
```

```
res.setFont(f2);
```

```
show.setFont(f2);
```

```
reg1.setFont(f2);
```

```
unam.setForeground(c3);
```

```
pas.setForeground(c3);
```

```
t1.setForeground(c3);
```

```
pfpwd.setForeground(c3);
```

```
login.setForeground(c3);
```

```
res.setForeground(c3);
```

```
show.setForeground(c3);
```

```
reg1.setForeground(c3);
```

```
con.add(unam);
```

```
con.add(pas);
```

```
con.add(t1);
```

```
con.add(pfpwd);
```

```
con.add(login);
```

```
con.add(res);
```

```
con.add(show);
```

```
con.add(reg1);
```

```
Clock();
```

```
}
```

```
@Override
```

```

public void actionPerformed(ActionEvent e) {

    // reg button papuntang regform

    if (e.getSource() == reg1) {
        reg app = new reg();
        app.setTitle("Admin Registration Form");
        app.setVisible(true);
        app.setSize(1000, 750);
        app.setLocationRelativeTo(null);
        login.this.dispose();
    }

    // Login Button
    if (e.getSource() == login) {
        getpass = pfpwd.getPassword();
        pass = String.valueOf(getpass);

        try {
            String sql = "select * from useraccount
where Username=? and Password=?";

            ps =
conn.prepareStatement(sql);

            ps.setString(1, t1.getText());
            ps.setString(2, pass);

```



```
rs = ps.executeQuery();  
int count = 0;  
while (rs.next()) {  
    count = count + 1;  
}  
if (count == 1) {
```

```
JOptionPane.showMessageDialog(null, "UserName and Password is correct");
```

```
JOptionPane.showMessageDialog(this, "Login Successful");
```

```
// login button punta ng
```

```
home/menu
```

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

```
public void
```

```
actionPerformed(ActionEvent evt) {
```

```
menu app
```

```
= new menu();
```

```
app.setTitle("Admin Menu");
```

```
app.setVisible(true);
```

```
app.setSize(1000, 400);
```

```
app.setLocationRelativeTo(null);
```

```
login.this.dispose();
```

```

        }
    });

    } else if (count > 1) {

OptionPane.showMessageDialog(null, "Duplicate UserName and Password");

    } else {

OptionPane.showMessageDialog(null, "UserName and Password is not correct Try
Again...");

OptionPane.showMessageDialog(this, "Invalid Username or Password");

        t1.setText("");
        pfpwd.setText("");
    }

    rs.close();
    ps.close();
} catch (Exception ex) {

OptionPane.showMessageDialog(null, ex);

}

}

```

```
// Reset Button
if (e.getSource() == res) {
    t1.setText("");
    pfpwd.setText("");
}

// Show Password
if (e.getSource() == show) {
    if (show.isSelected()) {
        pfpwd.setEchoChar((char) 0);
    } else {
        pfpwd.setEchoChar('*');
    }
}

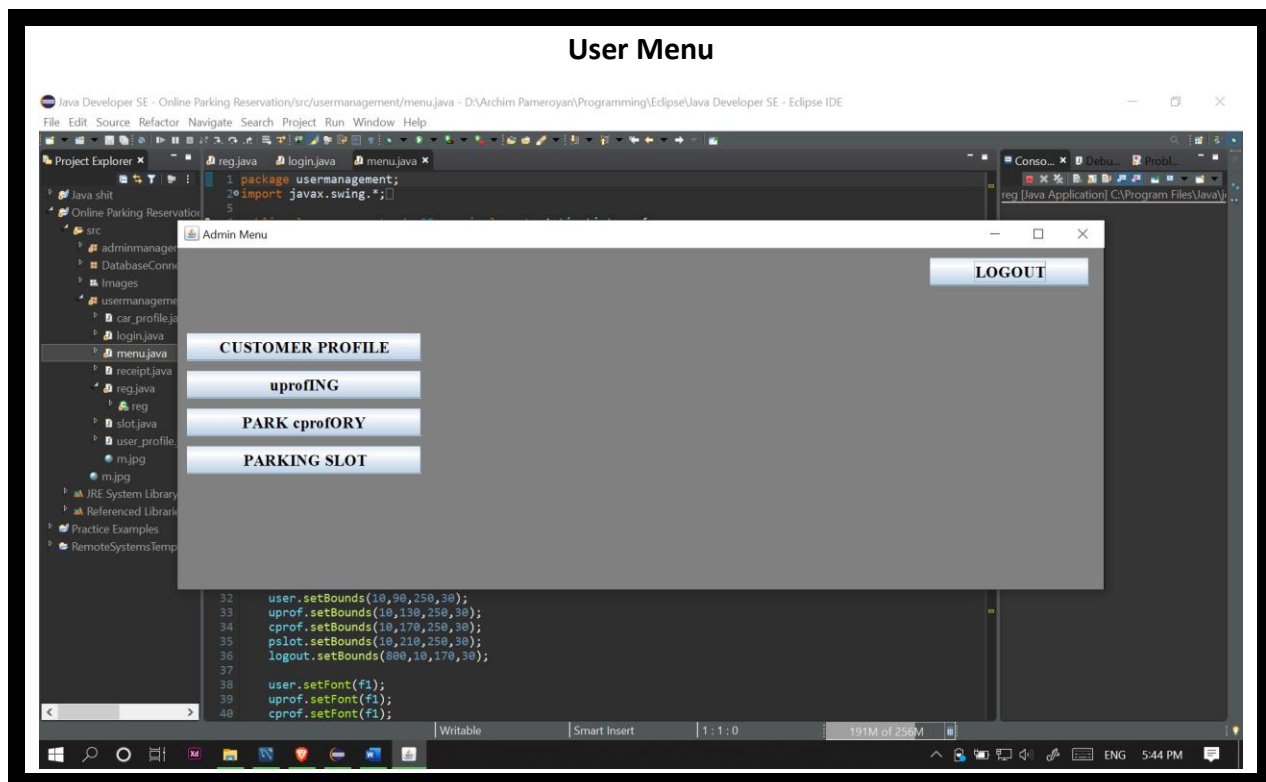
}

static login app = new login();

public static void main(String[] args) {
    app.setTitle("User Login");
    app.setSize(450, 600);
    app.setVisible(true);
    app.setLocationRelativeTo(null);
    app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

}

}



```
package usermanagement;
```

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
public class menu extends JFrame implements ActionListener{
```

```
    public JButton user, uprof, cprof, pslot, logout;
```

```
    Font f1 = new Font("Times New Roman", Font.BOLD,17);
```

```
    Font f2 = new Font("Arial", Font.ITALIC,13);
```

```
    Color c1 = Color.GRAY;
```

```
    Color c2 = Color.BLACK;
```

```
    Color c3 = Color.WHITE;
```

```
public menu() {  
    Container con = getContentPane();  
    con.setLayout(null);  
  
    user = new JButton ("CUSTOMER PROFILE");  
    uprof = new JButton ("uprofING");  
    cprof = new JButton ("PARK cprofORY");  
    pslot = new JButton ("PARKING SLOT");  
    logout = new JButton ("LOGOUT");  
  
    user.addActionListener(this);  
    uprof.addActionListener(this);  
    cprof.addActionListener(this);  
    pslot.addActionListener(this);  
    logout.addActionListener(this);  
  
    user.setBounds(10,90,250,30);  
    uprof.setBounds(10,130,250,30);  
    cprof.setBounds(10,170,250,30);  
    pslot.setBounds(10,210,250,30);  
    logout.setBounds(800,10,170,30);  
  
    user.setFont(f1);  
    uprof.setFont(f1);  
    cprof.setFont(f1);  
    pslot.setFont(f1);  
    logout.setFont(f1);  
}
```

```
user.setForeground(c2);
uprof.setForeground(c2);
cprof.setForeground(c2);
pslot.setForeground(c2);
logout.setForeground(c2);
```

```
con.setBackground(c1);
```

```
con.add(user);
con.add(uprof);
con.add(cprof);
con.add(pslot);
con.add(logout);
```

```
}
```

@Override

```
public void actionPerformed(ActionEvent e) {
    // TODO Auto-generated method stub
    //paglogout
    if (e.getSource() == logout) {
        int dialog = JOptionPane.showConfirmDialog(null, "Are you sure you want to
        LOGOUT?", "WARNING",
            JOptionPane.YES_NO_OPTION);
        if (dialog == JOptionPane.YES_OPTION) {
            //pabalik ng login.java
            login app = new login();
            app.setVisible(true);
            app.setSize(450,600);
            app.setVisible(true);
```

```

        app.setLocationRelativeTo(null);
        menu.this.dispose();
    }

    if (dialog == JOptionPane.NO_OPTION) {
        menu.this.show();
    }
}

//papuntang user profile
uprof.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        user_profile app = new user_profile();
        // app.setVisible(true);
        // app.setSize(1000,700);
        // app.setLocationRelativeTo(null);
        menu.this.dispose();
    }
});

//papuntang car profile
cprof.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        car_profile app = new car_profile();
        // app.setVisible(true);
        // app.setSize(1000,700);
        // app.setLocationRelativeTo(null);
        menu.this.dispose();
    }
});

```

```
//papuntang slot/reserve/cancel
```

```
//papuntang receipt
```

```
}
```

```
static menu app = new menu();
```

```
public static void main(String[] args) {
```

```
app.setTitle("Admin Menu");
```

```
app.setSize(1000,400);
```

```
app.setVisible(true);
```

```
app.setLocationRelativeTo(null);
```

```
}
```

```
}
```

Connection Database

```
package DatabaseConnection;
```

```
import java.sql.*;
```

```
public class ConnectionDB {
```

```
static String driver = "com.mysql.cj.jdbc.Driver";
```

```
static String url ="jdbc:mysql://localhost:3306/onlineparkingreservation";
```

```
static String username = "root";
```

```
static String password = "kakashijirachisohaer";
```

```
public static Connection getConnection(){
```

```
Connection conn=null;
```

```
try{
```



```
Class.forName(driver);  
conn=DriverManager.getConnection(url,username,password);  
}catch(Exception er){System.out.println(er);}  
return conn;  
}  
}
```