PASSPORT AUTOMATION SYSTEM

AIM

To develop the Passport Automation System using Dia diagrams, Java and MySql

1. PROBLEM ANALYSIS AND PROJECT PLAN

To simplify the process of applying passport, software has been created by designing through rational rose tool, using visual basic as a front end and Microsoft access as a back end. Initially the applicant login the passport automation system and submits his details. These details are stored in the database and verification process done by the passport administrator, regional administrator and police the passport is issued to the applicant.

PROBLEM STATEMENT

a. Passport Automation System is used in the effective dispatch of passport to all of the applicants. This system adopts a comprehensive approach to minimize the manual work and schedule resources, time in a cogent manner.
b. The core of the system is to get the online registration form (with details such as name, address etc.,) filled by the applicant whose testament is verified for its genuineness by the Passport Automation System with respect to the already existing information in the database.
c. This forms the first and foremost step in the processing of passport application. After the first round of verification done by the system, the information is in turn forwarded to the regional administrator's (Ministry of External Affairs) office.
d. The application is then processed manually based on the report given by the system, and any forfeiting identified can make the applicant liable to penalty as per the law.
e. The system forwards the necessary details to the police for its separate verification whose report is then presented to the administrator. After all the necessary criteria have been met, the original information is added to the database and the passport is sent to the applicant.

2. SOFTWARE REQUIREMENTS SPECIFICATION

1.0 Introduction

1.1 Purpose

1.2 Scope

1.3 Definition, Acronyms

1.4 and Abbreviations

1.0 INTRODUCTION

Passport Automation System is an interface between the Applicant and the Authority responsible for the Issue of Passport. It aims at improving the efficiency in the Issue of Passport and reduces the complexities involved in it to the maximum possible extent.

1.1 PURPOSE

If the entire process of 'Issue of Passport' is done in a manual manner then it would take several months for the passport to reach the applicant. Considering the fact that the number of applicants for passport is increasing every year, an Automated System becomes essential to meet the demand. So this system uses several programming and database techniques to elucidate the work involved in this process. As this is a matter of National Security, the system has been carefully verified and validated in order to satisfy it.

1.2 SCOPE

a. The System provides an online interface to the user where they can fill in their personal details

b. The authority concerned with the issue of passport can use this system to reduce his workload and process the application in a speedy manner. • Provide a communication platform between the applicant and the administrator. Transfer of data between the Passport Issuing Authority and the Local Police for verification of applicant's information.

1.3 DEFINITIONS, ACRONYMS AND THE ABBREVIATIONS •

Administrator - Refers to the super user who is the Central Authority who has been vested with the privilege to manage the entire system. It can be any higher official in the Regional Passport Office of Ministry of External Affairs. • Applicant - One who wishes to obtain the Passport. • PAS - Refers to this Passport Automation System.

1.4 REFERENCES IEEE Software Requirement Specification format.

1.5 TECHNOLOGIES TO BE USED • Microsoft Visual Basic 6.0

1.6 TOOLS TO BE USED • Rational Rose tool (for developing UML Patterns)

1.7 OVERVIEW

SRS includes two sections overall description and specific requirements - Overall description will describe major role of the system components and interconnections. Specific requirements will describe roles & functions of the actors.

2 OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

The PAS acts as an interface between the 'applicant' and the 'administrator'. This system tries to make the interface as simple as possible and at the same time not risking the security of data stored in. This minimizes the time duration in which the user receives the passport.

2.2 SOFTWARE INTERFACE

a. Front End Client - The applicant and Administrator online interface is built using Java

b. Back End – MS Access database.

2.3 HARDWARE INTERFACE

The server is directly connected to the client systems. The client systems have access to the database in the server.

2.4 SYSTEM FUNCTIONS

a. Secure Registration of information by the Applicants.

b. Message box for Passport Application Status Display by the Administrator.

c. Administrator can generate reports from the information and is the only authorized personnel to add the eligible application information to the database.

2.5 USER CHARACTERISTICS

a. Applicant - They are the people who desires to obtain the passport and submit the information to the database.

b. Administrator - He has the certain privileges to add the passportstatus and to approve the issue of passport. He may contain a group of persons under him to verify the documents and give suggestion whether or not to approve the dispatch of passport.

c. Police - He is the person who upon receiving intimation from the PAS, perform a personal verification of the applicant and see if he has any criminal case against him before or at present. He has been vetoed with the power to decline an application by suggesting it to the Administrator if he finds any discrepancy with the applicant. He communicates via this PAS.

2.6 CONSTRAINTS

o The applicants require a computer to submit their information.

o Although the security is given high importance, there is always a chance of intrusion in the web world which requires constant monitoring.

o The user has to be careful while submitting the information. Much care is required.

2.7 ASSUMPTIONS AND DEPENDENCIES

o The Applicants and Administrator must have basic knowledge of computers and English Language.

o The applicants may be required to scan the documents and send.

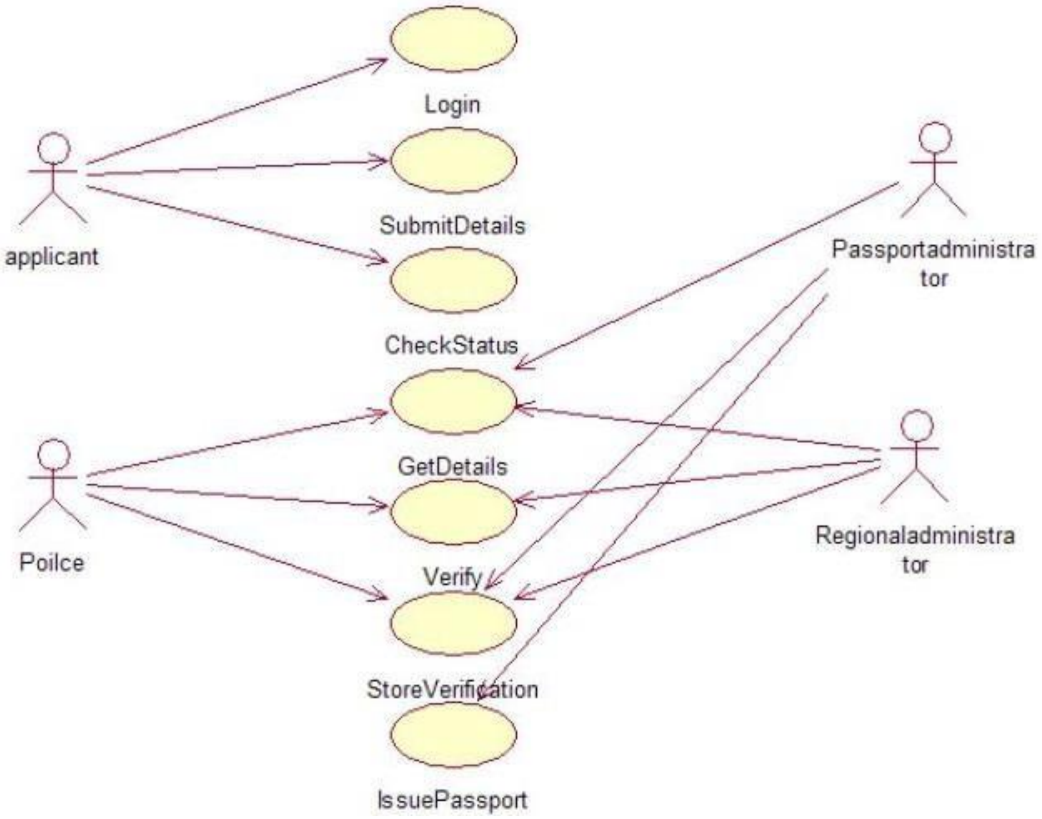
3. UML DIAGRAMS

Sno	**UML DIAGRAMS**
1	Use Case diagram

2	Class diagram
3	Interaction diagram
4	Sequence diagram
5	Collaboration diagram
6	State Chart diagram
7	Activity diagram
8	Component diagram
9	Deployment diagram
10	Package diagram

USE CASE DIAGRAM

Use case is shown as an ellipse containing the name of use case .An actor is shown as a stick figure with the name below it. Use case diagram is a graph of actors.



DOCUMENTATION OF USECASE DIAGRAM

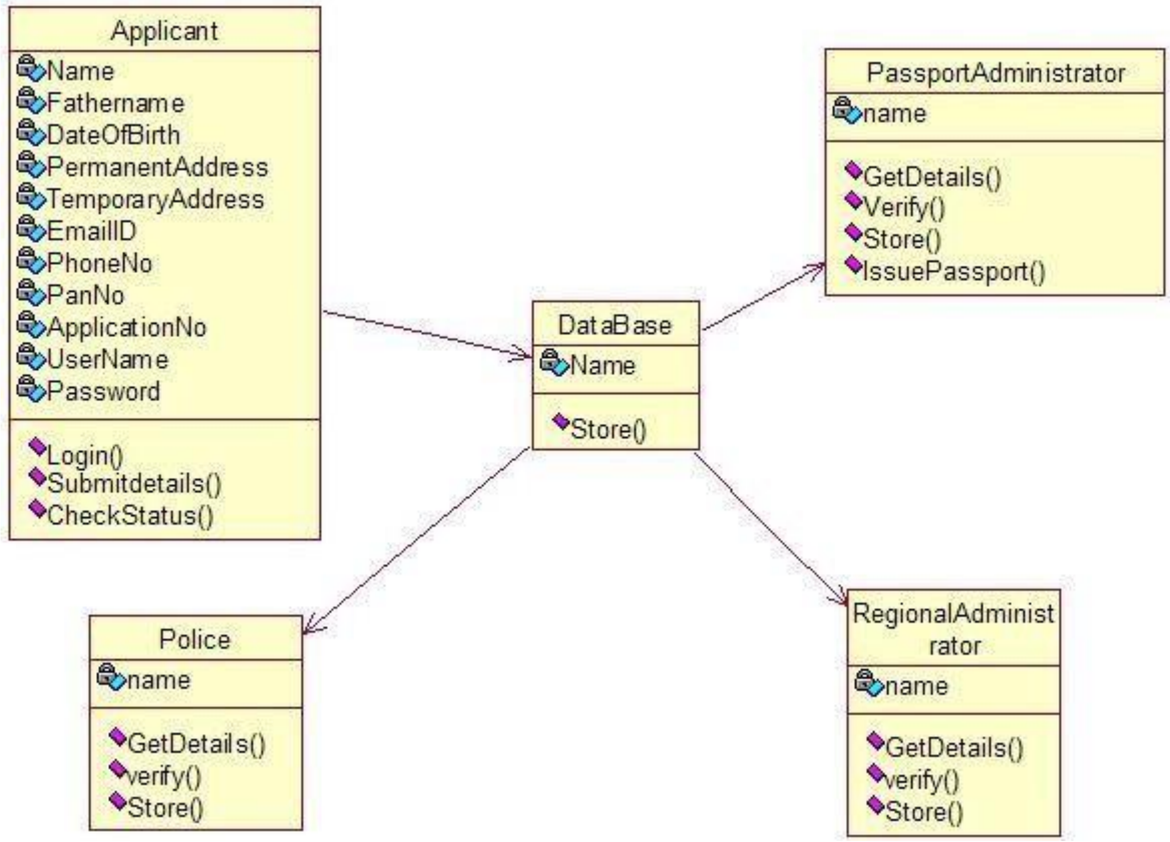
a. The actors in use case diagram are Applicant, regional administrator, database, passport Administrator, Police.

b. The use cases are Login, givedetails, logout, collectdetails, verification, issue.

c. The actors use the use case are denoted by the arrow

CLASSDIAGRAM

A class is drawn as rectangle box with three compartments or components separated by horizontal lines. The top compartment holds the class name and middle compartment holds the attribute and bottom compartment holds list of operations.



DOCUMENTATION OF CLASS DIAGRAM

- The classes are Applicant, database, regional administrator, passport administrator, and police.

- The applicant has attribute such as name and password and operations are login, givedetails and logout.

- The database has attribute such as name and operation is store.

- The regional administrator has attribute such as name and operation are get details, verify details and send.

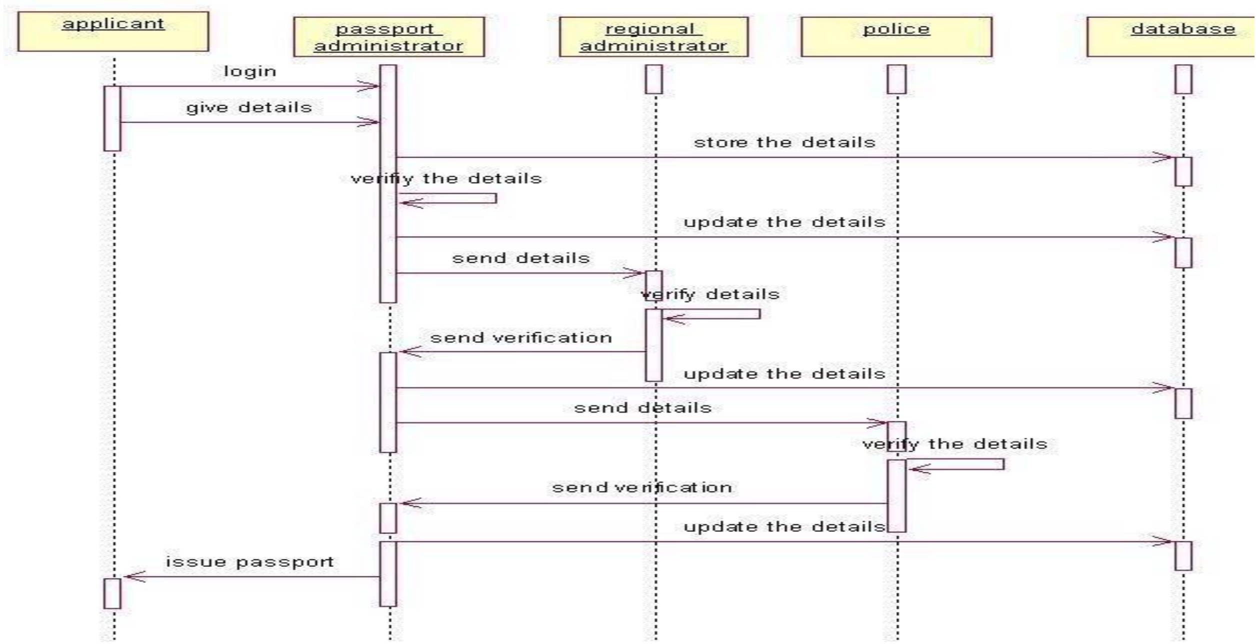
- The passport administrator has attribute such as name and operation are get details, verify details and issue.

- The police has attribute such as name and operation are get details, verify details and send.

SEQUENCE DIAGRAM

A sequence diagram shows an interaction arranged in time sequence,

It shows object participating in interaction by their lifeline by the message they exchange arranged in time sequence. Vertical dimension represent time and horizontal dimension represent object.



DOCUMENTATION OF SEQUENCE DIAGRAM.

- The applicant login the database and give his details and database store the details.

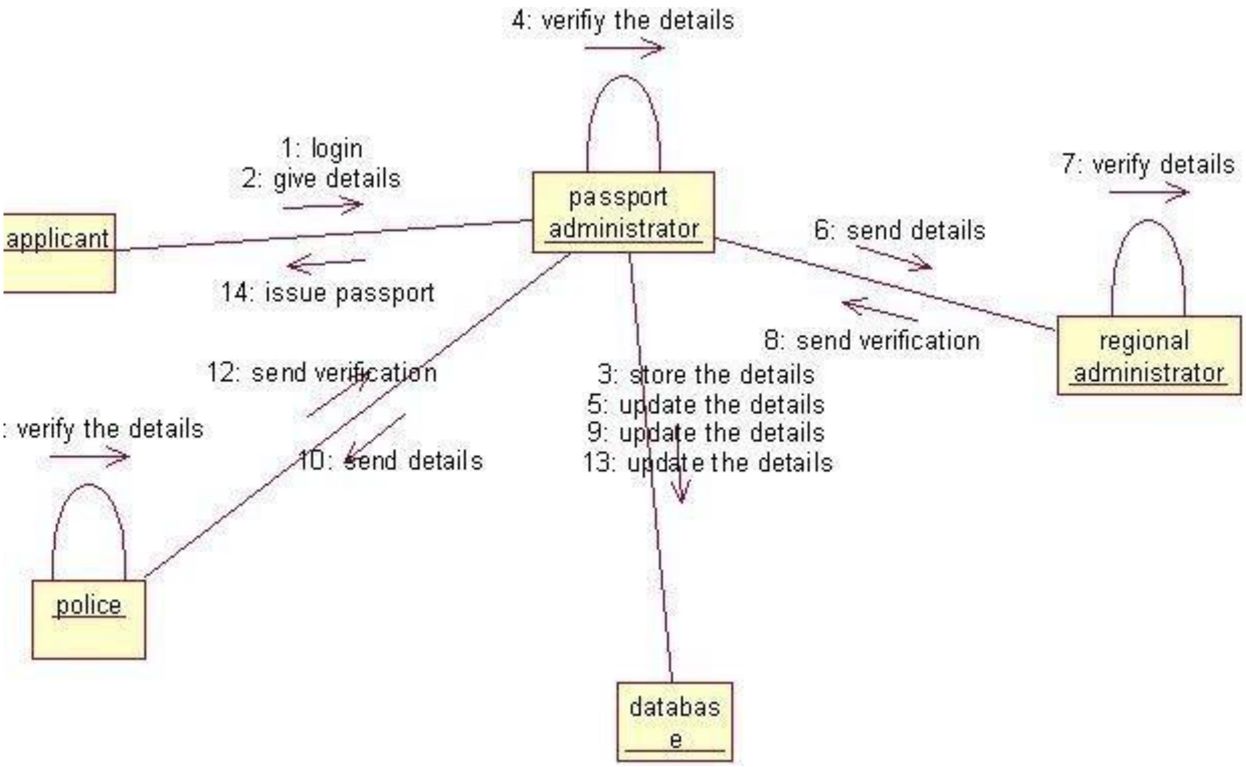
- The passport administrator get the details from the database and do verification and the forward to regional administrator.

- The regional administrator get details form passport administrator and perform verification and send report to passport administrator.

- The police get the details form passport administrator and perform verification and send report to passport administrator.

COLLABORATION DIAGRAM

A collaboration diagram is similar to sequence diagram but the message in number format. In a collaboration diagram sequence diagram is indicated by the numbering the message



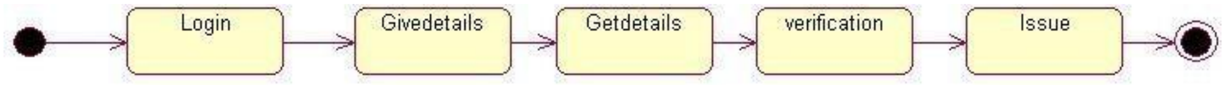
DOCUMENTATION OF COLLABORATION DIAGRAM

a. The applicant, passport administrator, regional administrator, police and database functions are show in sequence number

b. The applicant first login the passport automation system and submit his details the passport administrator, regional administrator and police verification are denoted.

STATE CHART DIAGRAM

The state chart diagram contains the states in the rectangle boxes and starts in indicated by the dot and finish is indicated by dot encircled. The purpose of state chart diagram is to understand the algorithm in the performing method.



DOCUMENTATION OF STATE CHART DIAGRAM

a. The states of the passport automation system are denoted in the state chart diagram

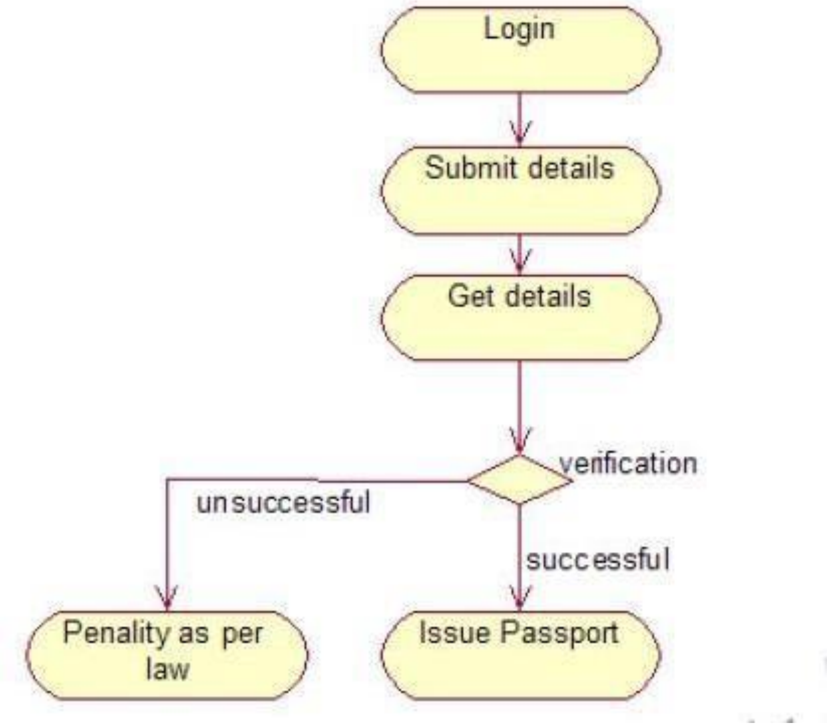
b. Login state represent authentication for login the passport automation system.

c. In this state, it checks whether the applicant has provided all the details that is required.

d. Police, regional administrator and passport administrator get necessary details and verification of the applicant are denoted from the Get detail state and verification state

ACTIVITY DIAGRAM

An activity diagram is a variation or special case of a state machine in which the states or activity representing the performance of operation and transitions are triggered by the completion of operation.
The purpose is to provide view of close and what is going on inside a use case or among several classes. An activity is shown as rounded box containing the name of operation.



DOCUMENTATION OF ACTIVITY DIAGRAM

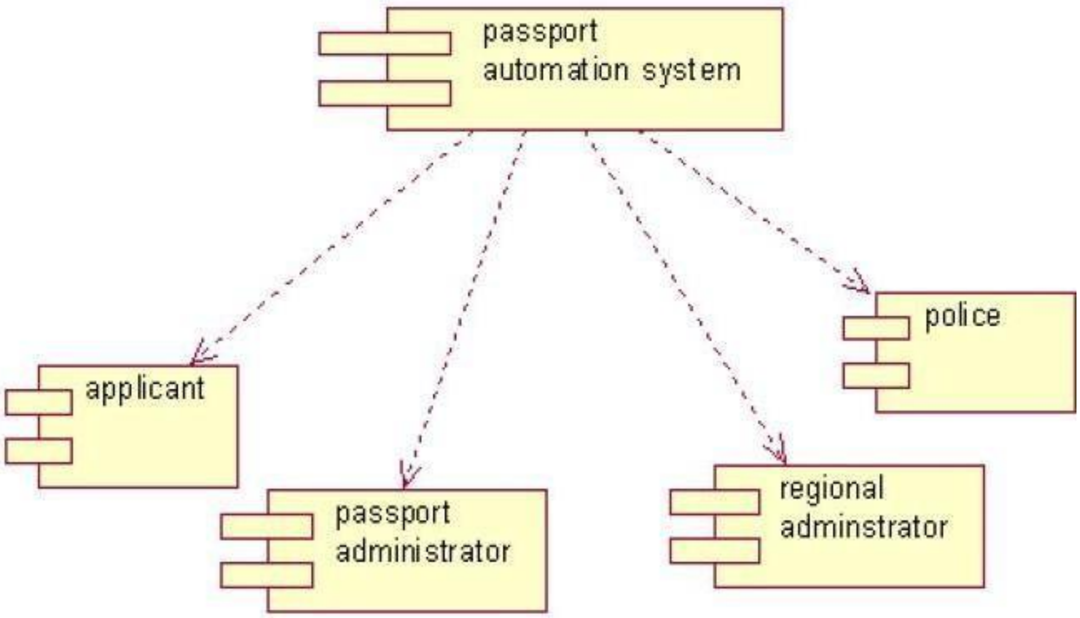
a. The activities in the passport automation system are login, submit details, get details, issue passport and penalty and verification.

b. In the login activity applicant give username and password and then login into the passport automation system after then fill the details that are required for application.

c. After the verification procedure completed successfully the passport is issued to the applicant.

COMPONENT DIAGRAM

The component diagram is represented by figure dependency and it is a graph of design of figure dependency.



DOCUMENTATION OF COMPONENT DIAGRAM

a. The components in the passport automation system are passport automation system, applicant, passport administrator, regional administrator, and police.

b. Applicant ,passport administrator, regional administrator and police are dependent on passport automation system are shown by the dotted arrow

DEPLOYMENT DIAGRAM

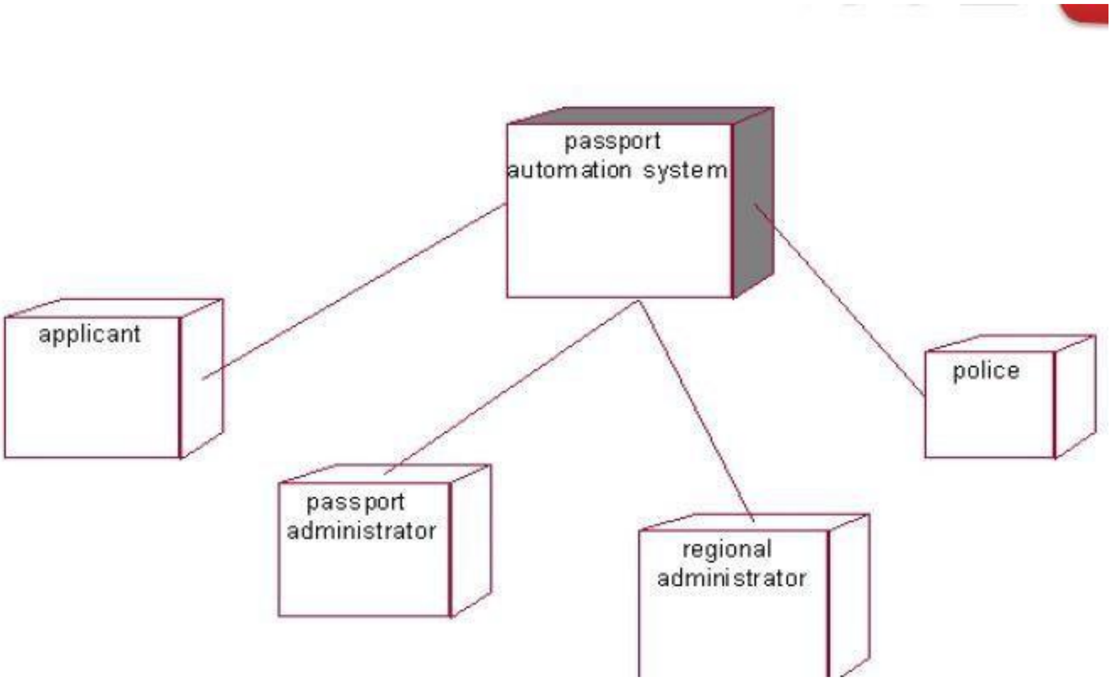
It is a graph of nodes connected by communication association. It is represented by a three dimensional box. The basic element of a deployment diagram is a node of two types

DEVICE NODE

A physical computing resource with processing and memory service to execute software, such as a typical computer or a mobile phone.

EXECUTION ENVIRONMENT NODE

This is a software computing resource that runs within an outer node and which itself provides a service to host an execute other executable software element.

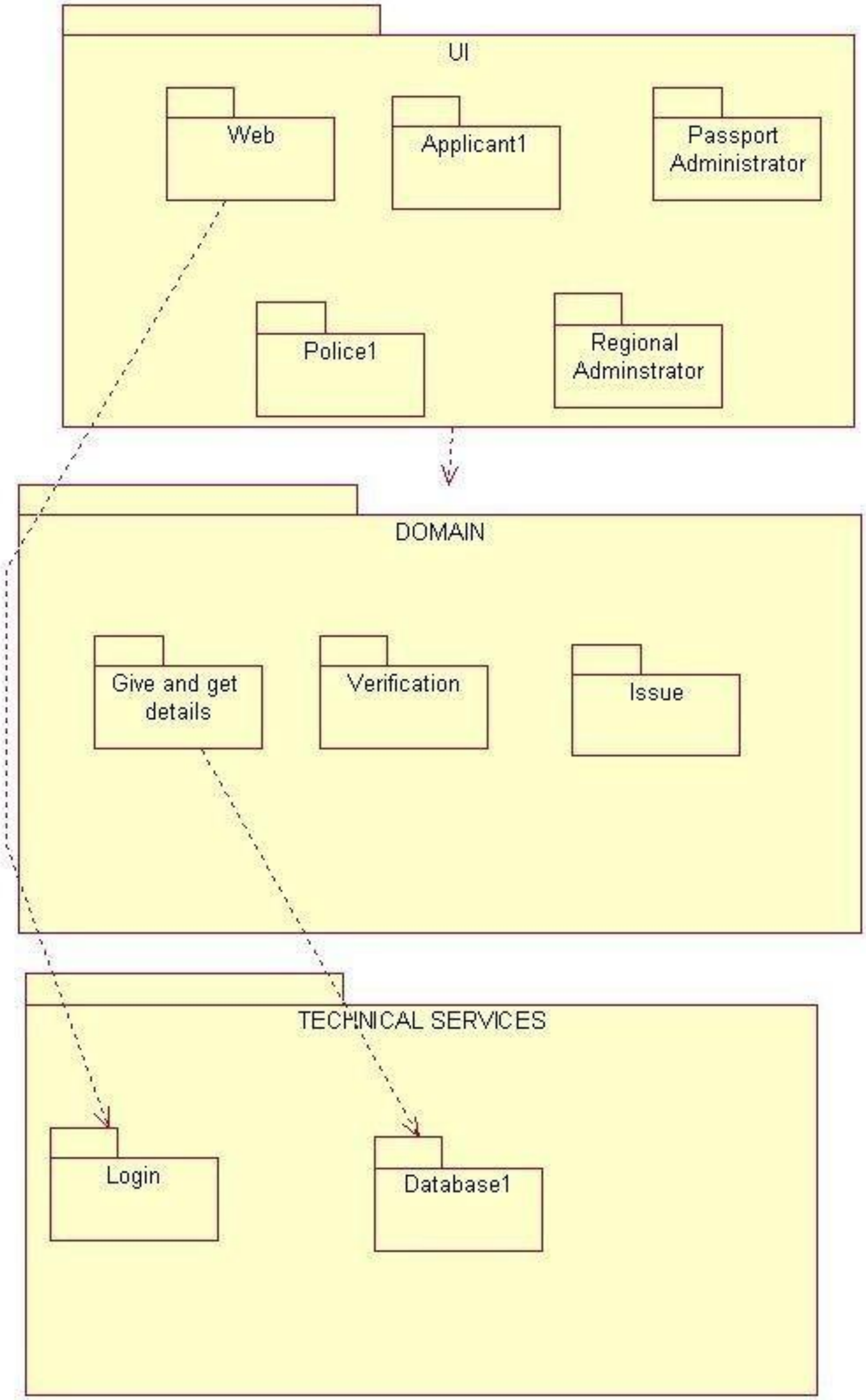


DOCUMENTATION OF DEPLOYMENT DIAGRAM

The device node is passport automation system and execution environment node are applicant passport administrator, regional administrator, and police.

PACKAGE DIAGRAM

A package diagram is represented as a folder shown as a large rectangle with a top attached to its upper left corner. A package may contain both sub ordinate package and ordinary model elements. All uml models and diagrams are organized into package


UI
Web
Applicant1
Passport Administrator
Police1
Regional Adminstrator
DOMAIN
Give and get details
Verification
Issue
TECHNICAL SERVICES
Login
Database1

DOCUMENTATION OF PACKAGE DIAGRAM

a. The three layer in the passport automation system are user interface layer, domain layer, technical service layer

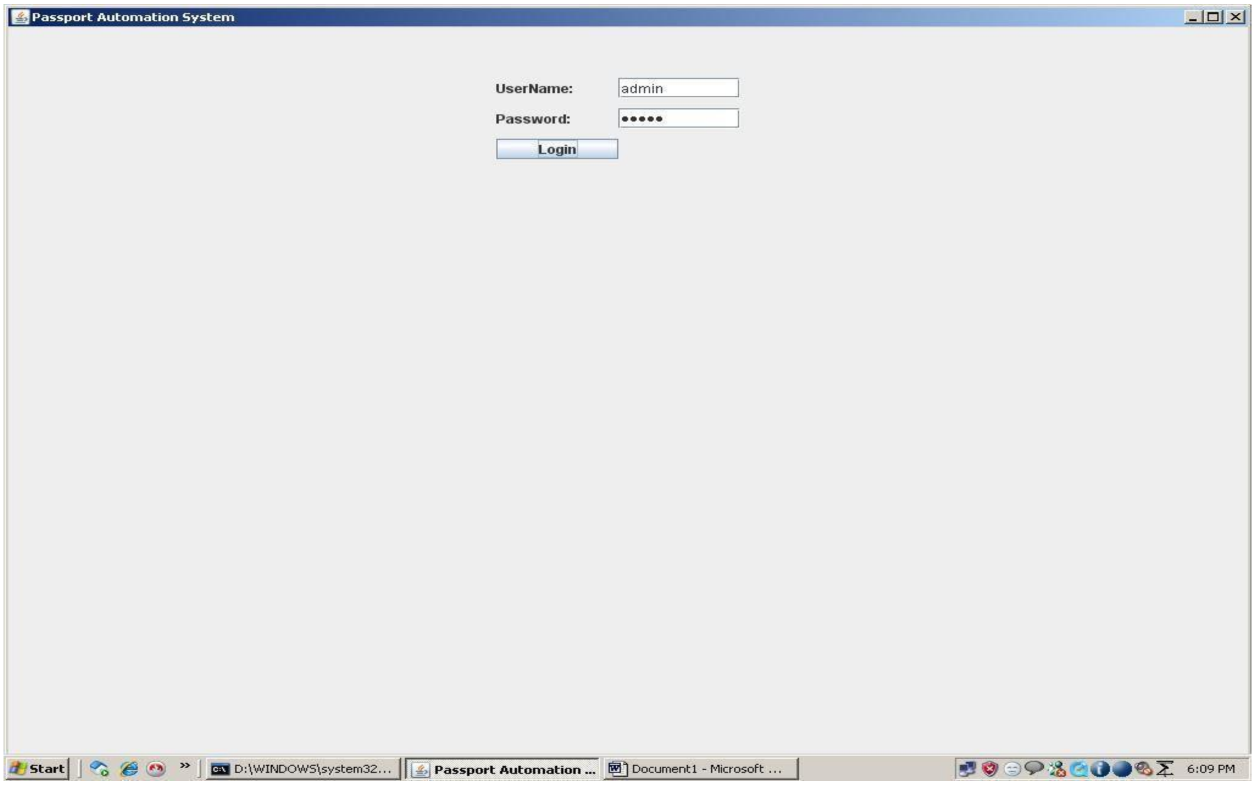
b. The user interface layer represents the user interface components such as web, applicant, passport administrator, police, and regional administrator.

c. The domain layer has major actions such as give and get details, verification and issues.

d. Technical service layer, authenticated user only can access the technical services.

4. FORMS

FORM 1

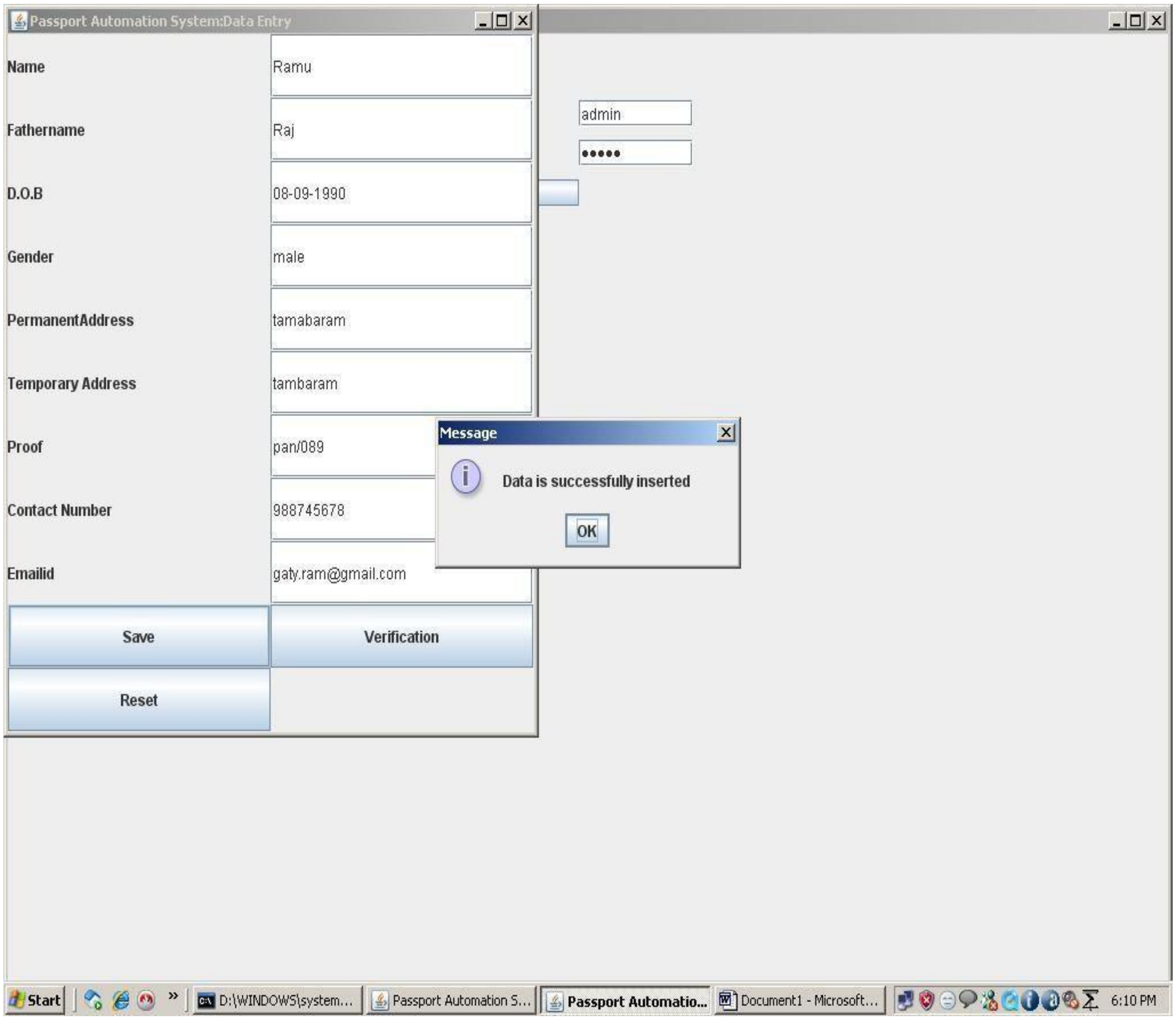


Passport Automation System

UserName: admin

Password: •••••

FORM 2



Name	Ramu
Fathername	Raj
D.O.B	08-09-1990
Gender	male
PermanentAddress	tamabaram
Temporary Address	tambaram
Proof	pan/089
Contact Number	988745678
Emailid	gaty.ram@gmail.com

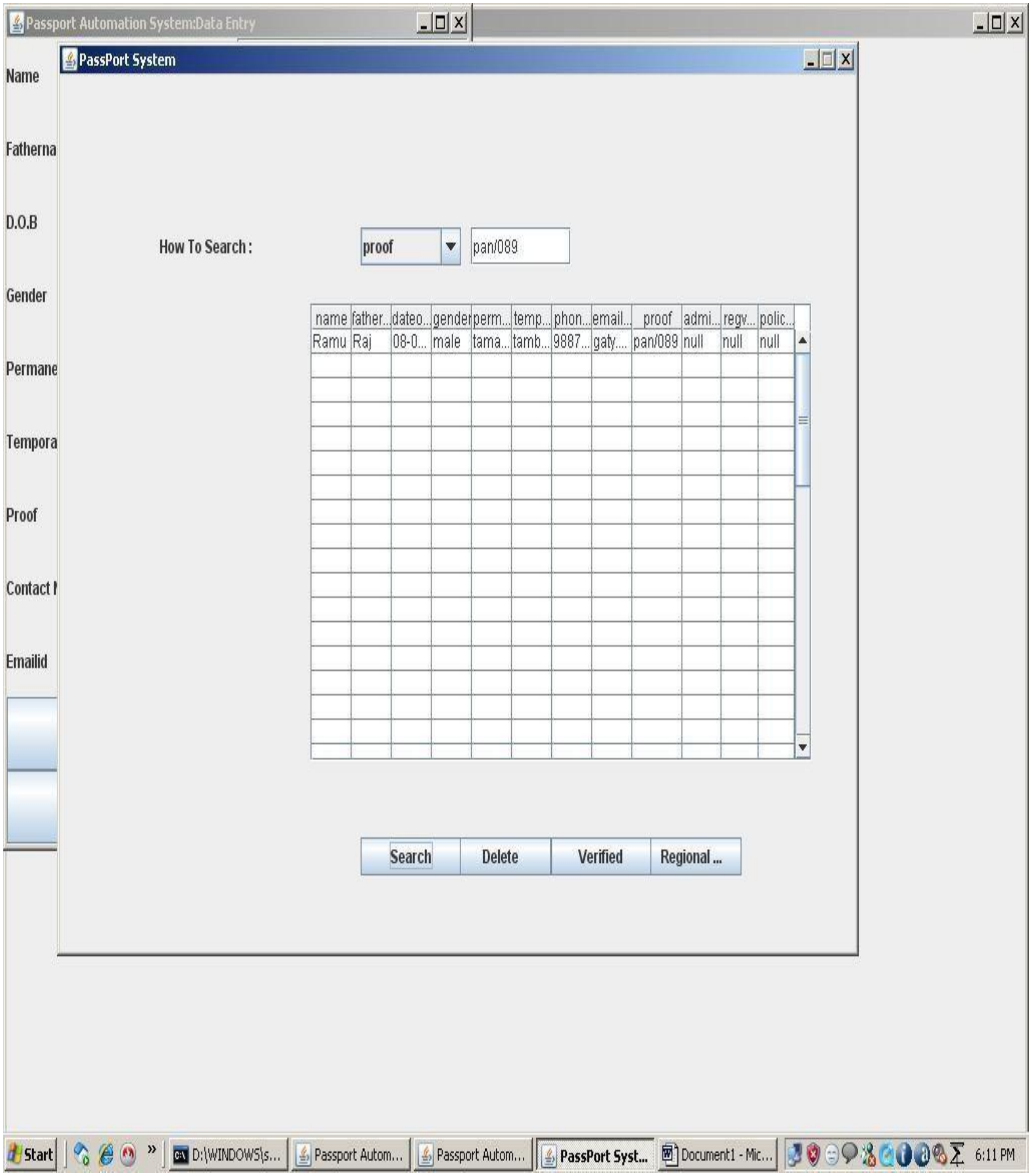
admin

•••••

Message

Data is successfully inserted

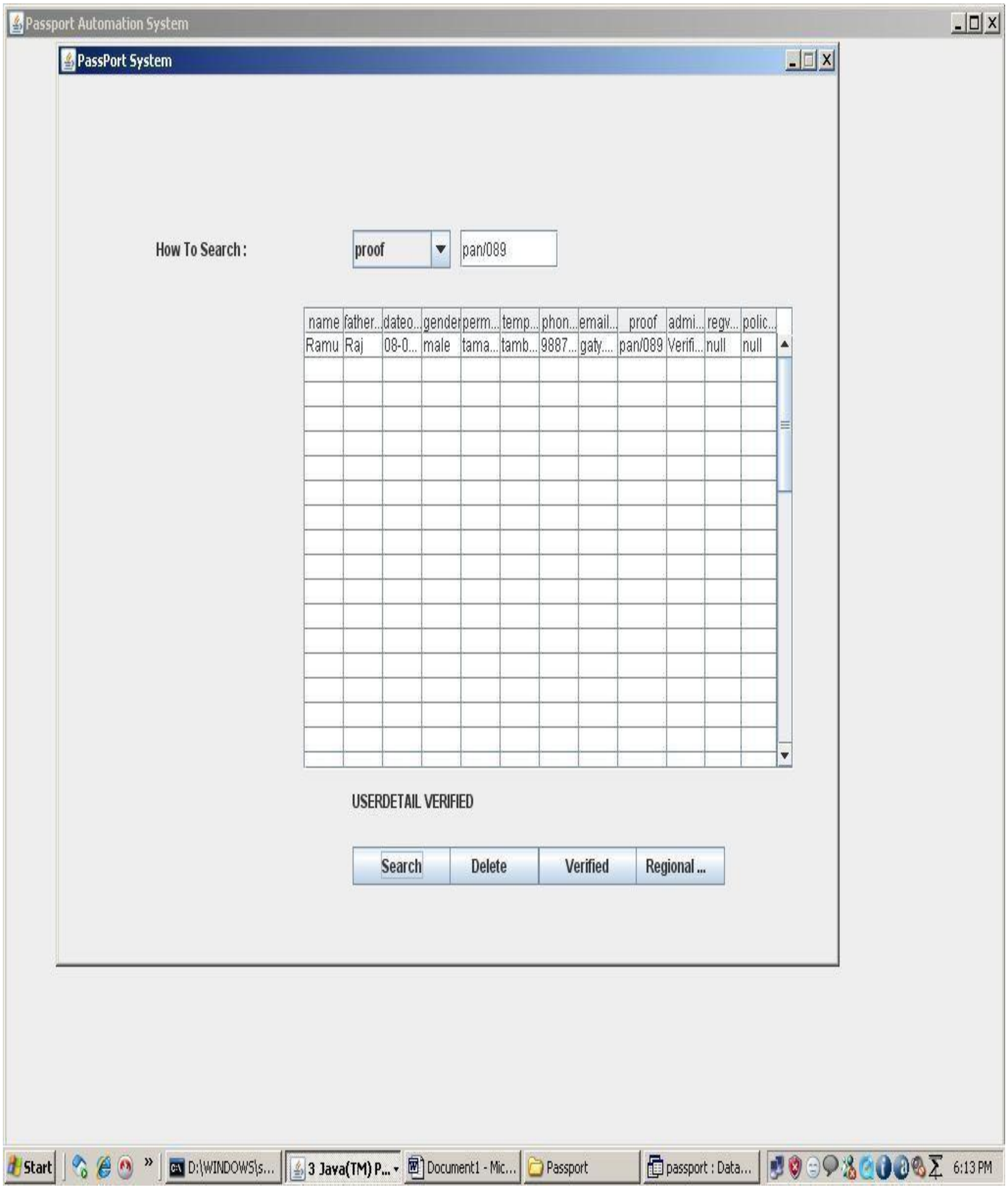
FORM 3



How To Search : proof pan/089

name	father...	dateo...	gender	perm...	temp...	phon...	email...	proof	admi...	regv...	polic...
Ramu	Raj	08-0...	male	tama...	tamb...	9887...	gaty....	pan/089	null	null	null

FORM 4

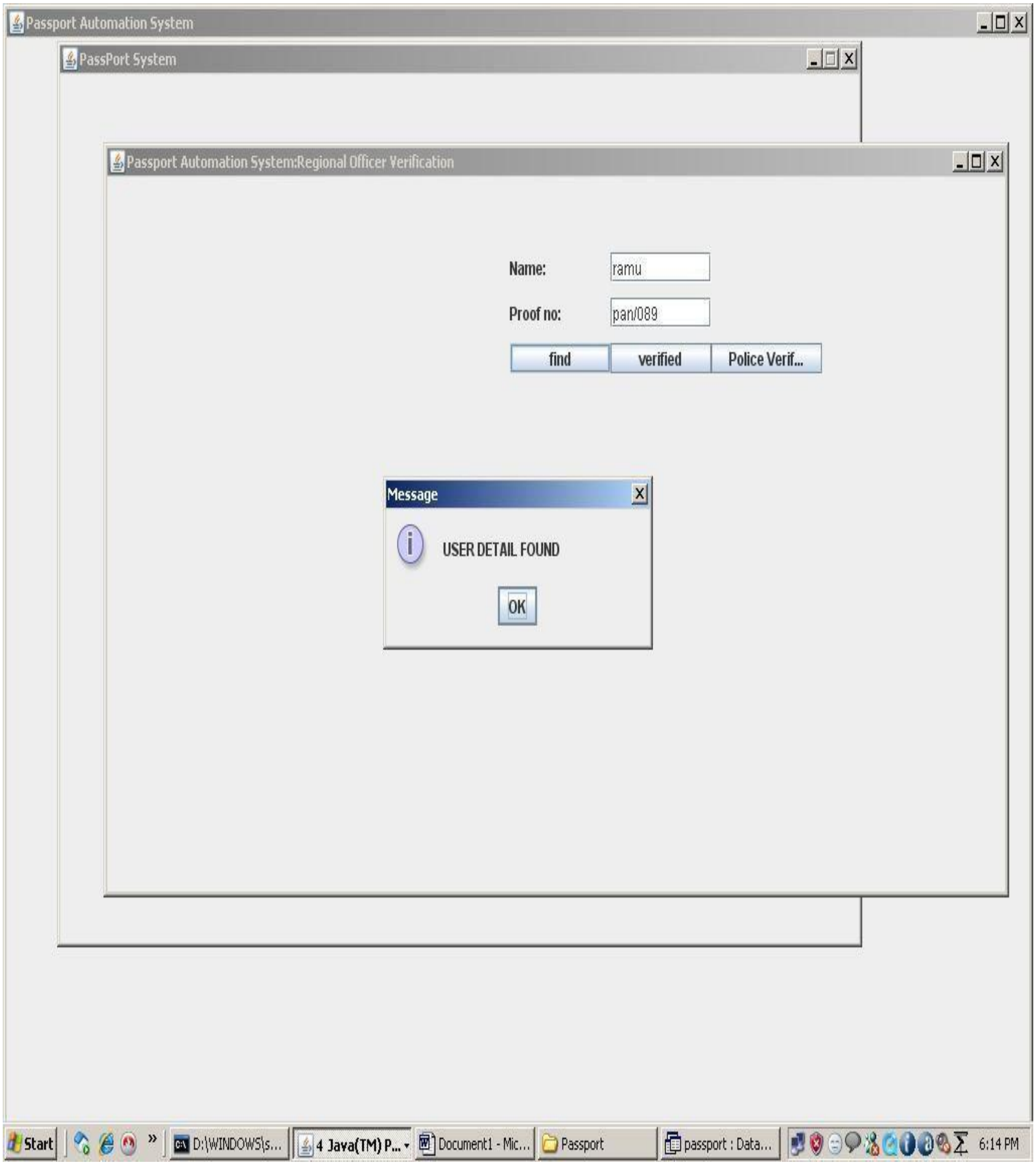


How To Search : proof pan/089

name	father...	dateo...	gender	perm...	temp...	phon...	email...	proof	admi...	regv...	polic...
Ramu	Raj	08-0...	male	tama...	tamb...	9887...	gaty....	pan/089	Verifi...	null	null

USERDETAIL VERIFIED

FORM 5



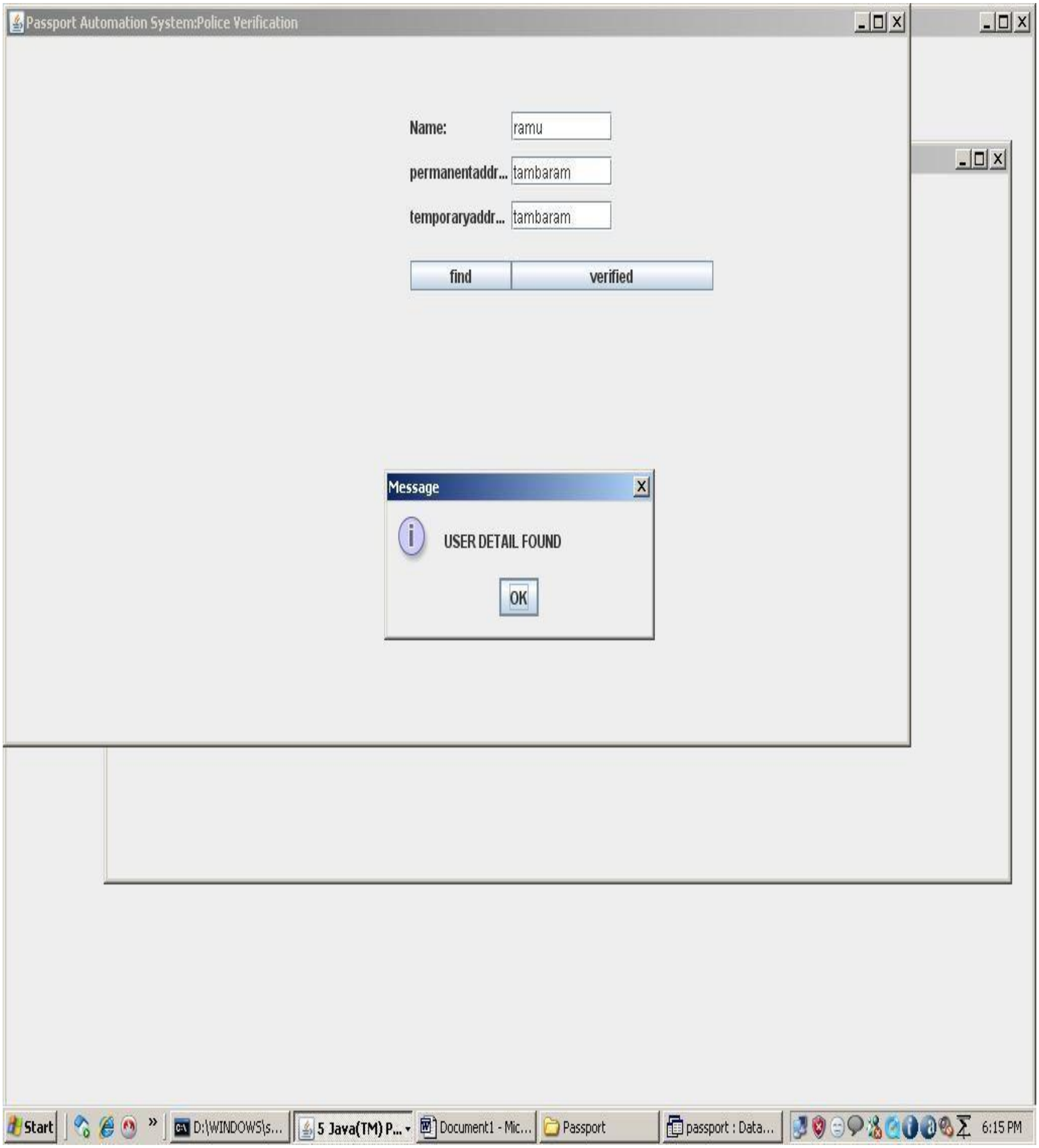
Passport Automation System
PassPort System
Passport Automation System:Regional Officer Verification

Name: ramu

Proof no: pan/089

USER DETAIL FOUND

FORM 6



Passport Automation System:Police Verification

Name: ramu

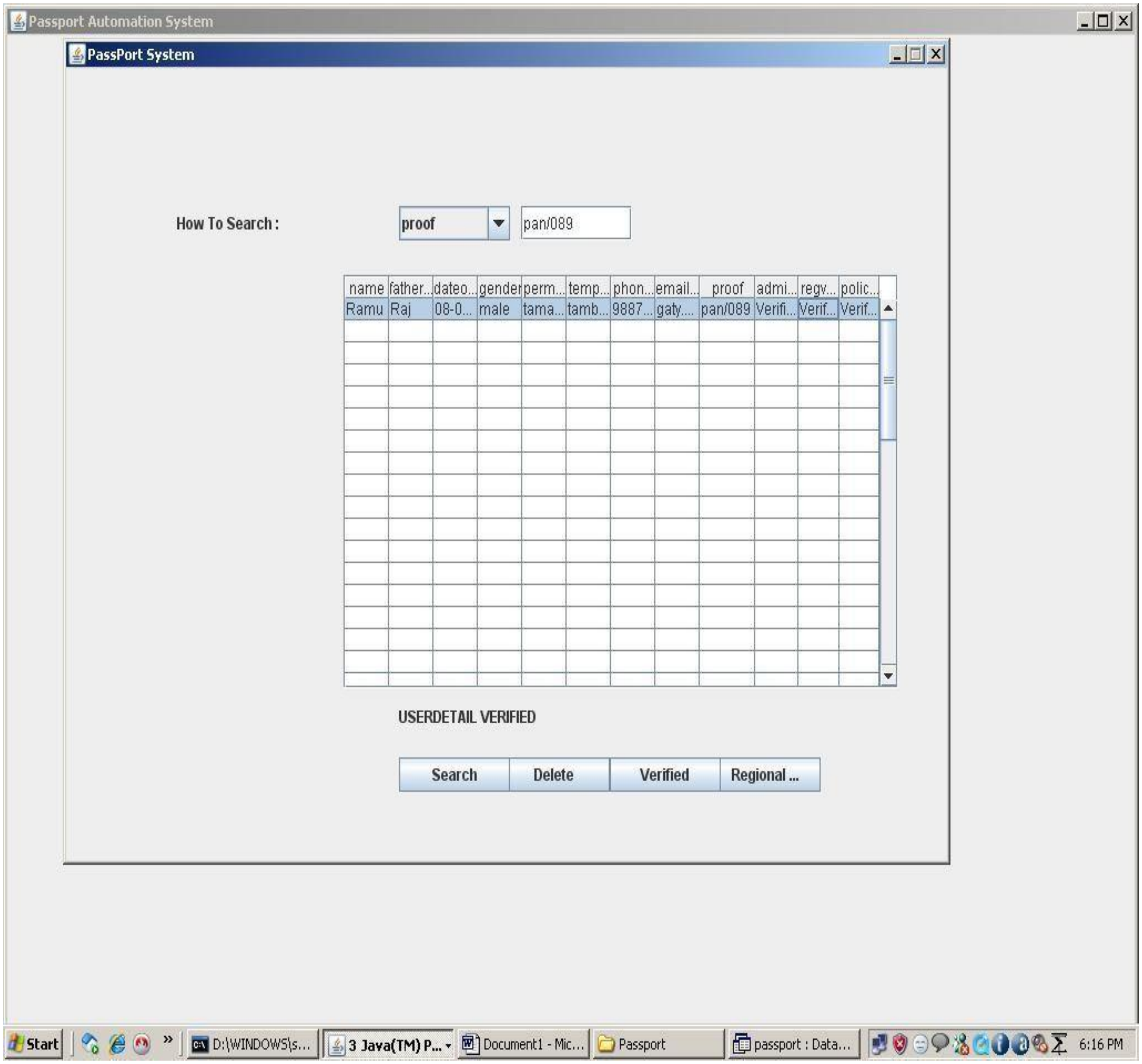
permanentaddr... tambaram

temporaryaddr... tambaram

Message

USER DETAIL FOUND

FORM 7



How To Search : proof pan/089

name	father...	dateo...	gender	perm...	temp...	phon...	email...	proof	admi...	regv...	polic...
Ramu	Raj	08-0...	male	tama...	tamb...	9887...	gaty....	pan/089	Verifi...	Verif...	Verif...

USERDETAIL VERIFIED

SOURCE CODE

LoginDemo.java \\LoginForm

```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
class LoginDemo
{
JButton SUBMIT;
JFrame f;
JLabel label1,label2;
final JTextField text1;
final JPasswordField text2;
LoginDemo()
{
f=new JFrame();
f.getContentPane().setLayout(null);
label1 = new JLabel();
label1.setText("UserName:");
label1.setBounds(400,50,100,20);
text1 = new JTextField(25);
text1.setBounds(500,50,100,20);
label2 = new JLabel();
label2.setText("Password:");
label2.setBounds(400,80,100,20);
text2 = new JPasswordField(25);
text2.setBounds(500,80,100,20);
SUBMIT=new JButton("Login");
SUBMIT.setBounds(400,110,100,20);
// NEWUSER=new JButton("Create Account");
//NEWUSER.setBounds(500,110,200,20);
f.add(label1);
f.add(text1);
f.add(label2);
f.add(text2);
f.add(SUBMIT);
//f.add(NEWUSER);
f.setSize(1024,768);
f.setTitle("Passport Automation System");
f.setVisible(true);
SUBMIT.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
String value1=text1.getText();
String value2=text2.getText();
String user1="";
String pass1="";
String user2="";
String pass2="";
try
{ Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con = DriverManager.getConnection("Jdbc:Odbc:pass","","");
Statement st = con.createStatement();
ResultSet res = st.executeQuery("SELECT * FROM login where
username='"+value1+"' and password='"+value2+"'");
while (res.next())
{user1 = res.getString("username");
pass1 = res.getString("password");
}if(value1.equals(user2) && value2.equals(pass2))
{ JOptionPane.showMessageDialog(null,"Incorrect login or
password","Error",JOptionPane.ERROR_MESSAGE);
}else if(value1.equals(user1) && value2.equals(pass1))
{ CreateAccount acc=new CreateAccount();
acc.setTitle("Passport Automation System:Data Entry");
} else
{
JOptionPane.showMessageDialog(null,"Incorrect login or
password","Error",JOptionPane.ERROR_MESSAGE);
} }
catch(Exception e)
{
System.out.println(e.getMessage());
} } }); }
public static void main(String arg[])
{
LoginDemo frame=new LoginDemo();
}}
```

CreateAccount.java

```
import javax.swing.*;
import java.awt.*;
import java.sql.*;
import java.awt.event.*;
import java.lang.String.*;
class CreateAccount extends JFrame
{
JTextField text1,text2,text3,text31,text4,text5,text6,text7,text8;
JLabel label1,label2,label3,label31,label4,label5,label6,label7,label8;
JPanel panel;
JButton button1,button2,button3;
String re=" ";
CreateAccount()
{
text1=new JTextField(15);
text2=new JTextField(15);
text3=new JTextField(15);
text31=new JTextField(15);
text4=new JTextField(15);
text5=new JTextField(15);
text6=new JTextField(15);
text7=new JTextField(15);
text8=new JTextField(15);
label1=new JLabel("Name");
label2=new JLabel("Fathername");
label3=new JLabel("D.O.B");
label31=new JLabel("Gender");
label4=new JLabel("PermanentAddress");
label5=new JLabel("Temporary Address");
label6=new JLabel("Proof");
label7=new JLabel("Contact Number");
label8=new JLabel("Emailid");
button1=new JButton("Save");
button2=new JButton("Verification");
button3=new JButton("Reset");
panel=new JPanel(new GridLayout(11,2));
panel.add(label1);
panel.add(text1);
panel.add(label2);
panel.add(text2);
panel.add(label3);
panel.add(text3);
panel.add(label31);
panel.add(text31);
panel.add(label4);
panel.add(text4);
panel.add(label5);
panel.add(text5);
panel.add(label6);
panel.add(text6);
panel.add(label7);
panel.add(text7);
panel.add(label8);
panel.add(text8);
panel.add(button1);
panel.add(button2);
panel.add(button3);
button1.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
String value1=text1.getText();
String value2=text2.getText();
String value3=text3.getText();
String value31=text31.getText();
String value4=text4.getText();
String value5=text5.getText();
String value6=text6.getText();
String value7=text7.getText();
String value8=text8.getText();
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con = DriverManager.getConnection("Jdbc:Odbc:pass","","");
Statement st = con.createStatement();
int k=st.executeUpdate("insert into
userdetails(name,fathername,dateofbirth,gender,permanentaddress,temporaryaddre
ss,phoneno,emailid,proof)
values('"+value1+"','"+value2+"','"+value3+"','"+value31+"','"+value4+"','"+value
5
+"','"+value7+"','"+value8+"','"+value6+"')");
JOptionPane.showMessageDialog(null,"Data is successfully inserted");
}
catch(Exception e)
{
System.out.println(e);
}
}
});
add(panel);
setSize(200,400);
setVisible(true);
```

```
button2.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(ae.getSource()==button2)
{
a f2=new a();
f2.setSize(800,600);
f2.setTitle("Passport Administrator Verification");
f2.show();
}
else
{
}
}
});
button3.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
text1.setText(re);
text2.setText(re);
text3.setText(re);
text31.setText(re);
text4.setText(re);
text5.setText(re);
text6.setText(re);
text7.setText(re);
text8.setText(re);
}
});
}
public static void main(String args[])
{
CreateAccount acc=new CreateAccount();
}
}
```

a.java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
class a extends JFrame implements ActionListener,ItemListener
{
JButton search,del,reg,verify;
JLabel name,find;
JComboBox list;
JTextField text;
Container con,con1;
String searchtext,searchfield,sql;
ResultSet rs;
ResultSet rs1;
JTable table;
Object rows[][];
int tval=0,tval1=0;
JScrollPane scrollPane;
String ver="Verified";
a()
{
con=getContentPane();
con.setLayout(null);
con1=getContentPane();
con1.setLayout(null);
rows=new Object[50][13];
Object headers[] =
{"name","fathername","dateofbirth","gender","permanentaddress","temporaryaddress","phoneno","emailid","proof","adminverify","regverify","policeverify"};
table = new JTable(rows, headers);
scrollPane = new JScrollPane(table);
scrollPane.setBounds(250,150,500,300);
scrollPane.setBackground(Color.WHITE);
con.add(scrollPane);
search=new JButton("Search");
search.setBounds(300,500,100,25);
con.add(search);
search.addActionListener(this);
del=new JButton("Delete");
del.setBounds(390,500,100,25);
con.add(del);
del.addActionListener(this);
verify=new JButton("Verified");
verify.setBounds(490,500,100,25);
con.add(verify);
verify.addActionListener(this);
```

```
reg=new JButton("Regional Verification");
reg.setBounds(580,500,100,25);
con.add(reg);
reg.addActionListener(this);
name=new JLabel("How To Search :");
name.setBounds(100,100,200,25);
con.add(name);
find=new JLabel("");
find.setBounds(300,450,350,40);
con.add(find);
text=new JTextField();
text.setBounds(410,100,100,25);
con1.add(text);
text.addActionListener(this);
list=new JComboBox();
list.setModel(new DefaultComboBoxModel(new String[] { "Select","proof" }));
list.setBounds(300,100,100,25);
con1.add(list);
list.addItemListener(this);
}
public void actionPerformed(ActionEvent ae)
{
if(ae.getSource()==text)
find.setText("");
if(ae.getSource()==del)
{
String getdel=JOptionPane.showInputDialog(search, "Enter the Proofno
","PassPort",1);
try{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection cntn3=DriverManager.getConnection("Jdbc:Odbc:pass","","");
Statement ste3=cntn3.createStatement();
ste3.executeUpdate("delete from userdetails where proof ='"+getdel+"'");
find.setText("USERDETAIL DELETED");
}
catch(Exception dele)
{
}
}
if(ae.getSource()==verify)
{
try{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection cntn3=DriverManager.getConnection("Jdbc:Odbc:pass","","");
Statement ste3=cntn3.createStatement();
```

```
String rt=text.getText();
ste3.executeUpdate("UPDATE userdetails SET adminverify='"+ver+"' where
proof='"+rt+"'");
find.setText("USERDETAIL VERIFIED");
}
catch(Exception dele)
{
}
}
if(ae.getSource()==reg)
{
try
{
b g2=new b();
g2.setTitle("Regional Officer Verification");
}
catch(Exception reg)
{
}
}
if(ae.getSource()==search)
{
if(searchfield==null)
find.setText("Please Select Search Category..,");
else
{
sql="select * from userdetails where ";
sql+=searchfield;
sql+="='"+text.getText()+"'";
System.out.println(sql); // the query for sql statement
try{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection cntn=DriverManager.getConnection("Jdbc:Odbc:pass","","");
Statement ste1=cntn.createStatement();
rs1=ste1.executeQuery(sql);
tval=0;
while(rs1.next())
{
table.setValueAt(""+rs1.getString(1),tval,0);
table.setValueAt(""+rs1.getString(2),tval,1);
table.setValueAt(""+rs1.getString(3),tval,2);
table.setValueAt(""+rs1.getString(4),tval,3);
table.setValueAt(""+rs1.getString(5),tval,4);
table.setValueAt(""+rs1.getString(6),tval,5);
table.setValueAt(""+rs1.getString(7),tval,6);
table.setValueAt(""+rs1.getString(8),tval,7);
table.setValueAt(""+rs1.getString(9),tval,8);
table.setValueAt(""+rs1.getString(10),tval,9);
table.setValueAt(""+rs1.getString(11),tval,10);
table.setValueAt(""+rs1.getString(12),tval,11);
tval++;
}
if(tval==0)
find.setText("Details Not Availabel( "+searchfield+" : "+text.getText()+" )\nTri
Again...,");
}
catch(Exception e)
{
JOptionPane.showMessageDialog(search,"Sorry,DataBase Problem,","PassPort
System,",JOptionPane.INFORMATION_MESSAGE);
}
}
}
}
public void itemStateChanged(ItemEvent ie)
{
find.setText("");
text.setText("");
list.removeItem("Select");
for(int i=0;i<tval;i++)
{
table.setValueAt("",i,0);
//table.setValueAt("",i,1);
//table.setValueAt("",i,2);
}
searchfield=""+ie.getItem();
setSize(800,600);
setTitle("PassPort System");
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setVisible(true);
setResizable(false);
}
public static void main(String[] argv)
{
a f2=new a();
}
}
```

b.java

```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
class b extends JFrame implements ActionListener
{
JButton SUBMIT,verify,police;
JFrame f;
JLabel label1,label2;
final JTextField text1, text2;
b()
{
f=new JFrame();
f.getContentPane().setLayout(null);
label1 = new JLabel();
label1.setText("Name:");
label1.setBounds(400,50,100,20);
text1 = new JTextField(25);
text1.setBounds(500,50,100,20);
label2 = new JLabel();
label2.setText("Proof no:");
label2.setBounds(400,80,100,20);
text2 = new JTextField(25);
text2.setBounds(500,80,100,20);
SUBMIT=new JButton("find");
SUBMIT.setBounds(400,110,100,20);
verify=new JButton("verified");
verify.setBounds(500,110,100,20);
police=new JButton("Police Verification");
police.setBounds(600,110,110,20);
police.addActionListener(this);
f.add(label1);
f.add(text1);
f.add(label2);
f.add(text2);
f.add(SUBMIT);
f.add(verify);
f.add(police);
f.setTitle("Passport Automation System:Regional Officer Verification");
f.setSize(900,500);
f.setVisible(true);
```

```
SUBMIT.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
String value1=text1.getText();
String value2=text2.getText();
String user1="";
String pass1="";
String user2="";
String pass2="";
String ver="Verified";
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con = DriverManager.getConnection("Jdbc:Odbc:pass","","");
Statement st = con.createStatement();
ResultSet res = st.executeQuery("SELECT * FROM RegionalDatabase where
Name='"+value1+"' and Proof='"+value2+"'");
while (res.next())
{
user1 = res.getString("Name");
pass1 = res.getString("Proof");
}
if(value1.equals(user2) && value2.equals(pass2))
{
JOptionPane.showMessageDialog(null,"Type the name and
Proof","Error",JOptionPane.ERROR_MESSAGE);
}
else if(value1.equals(user1) && value2.equals(pass1))
{
JOptionPane.showMessageDialog(null,"USER DETAIL FOUND");
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con2 = DriverManager.getConnection("Jdbc:Odbc:pass","","");
Statement st2 = con2.createStatement();
ResultSet res2 = st2.executeQuery("UPDATE userdetails SET regverify='"+ver+"'
where name='"+value1+"' and Proof='"+value2+"'");
}
else
{
JOptionPane.showMessageDialog(null,"DETAILS NOT
FOUND","Error",JOptionPane.ERROR_MESSAGE);
}
}
catch(Exception e)
{
```

```
// System.out.println(e.getMessage());
}
}
});
}
public void actionPerformed(ActionEvent ae)
{
if(ae.getSource()==police)
{
try
{
c g3=new c();
}
catch(Exception police)
{
}
}
}
public static void main(String arg[])
{
b g2=new b();
}
}
```

c.java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
class c
{
JButton SUBMIT,verify;
JFrame f;
JLabel label1,label2,label3;
final JTextField text1, text2,text3;
c()
{
f=new JFrame();
f.getContentPane().setLayout(null);
label1 = new JLabel();
label1.setText("Name:");
label1.setBounds(400,50,100,20);
```

```
text1 = new JTextField(25);
text1.setBounds(500,50,100,20);
label2 = new JLabel();
label2.setText("permanentaddress");
label2.setBounds(400,80,100,20);
text2 = new JTextField(25);
text2.setBounds(500,80,100,20);
label3 = new JLabel();
label3.setText("temporaryaddress");
label3.setBounds(400,110,100,20);
text3 = new JTextField(25);
text3.setBounds(500,110,100,20);
SUBMIT=new JButton("find");
SUBMIT.setBounds(400,150,100,20);
verify=new JButton("verified");
verify.setBounds(500,150,200,20);
f.add(label1);
f.add(text1);
f.add(label2);
f.add(text2);
f.add(label3);
f.add(text3);
f.add(SUBMIT);
f.add(verify);
f.setTitle("Passport Automation System:Police Verification");
f.setSize(900,500);
f.s etVisible(true);
SUBMIT.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
String value1=text1.getText();
String value2=text2.getText();
String value3=text3.getText();
String user1="";
String pass1="";
String pass3="";
String user2="";
String pass2="";
String pass4="";
String ver="Verified";
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con = DriverManager.getConnection("Jdbc:Odbc:pass","","");
Statement st = con.createStatement();
ResultSet res = st.executeQuery("SELECT * FROM PoliceDb where
Name='"+value1+"' and permanentaddress='"+value2+"' and
temporaryaddress='"+value3+"'");
while (res.next())
{
user1 = res.getString("Name");
pass1 = res.getString("permanentaddress");
pass3 =res.getString("temporaryaddress");
}
if(value1.equals(user2) && value2.equals(pass2) && value3.equals(pass4) )
{
JOptionPane.showMessageDialog(null,"Type the name and
Proof","Error",JOptionPane.ERROR_MESSAGE);
}
else if(value1.equals(user1) && value2.equals(pass1) && value3.equals(pass3))
{
JOptionPane.showMessageDialog(null,"USER DETAIL FOUND");
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con2 = DriverManager.getConnection("Jdbc:Odbc:pass","","");
Statement st2 = con2.createStatement();
ResultSet res2 = st2.executeQuery("UPDATE userdetails SET
policeverify='"+ver+"' where name='"+value1+"' and
permanentaddress='"+value2+"' and temporaryaddress='"+value3+"'");
}
else
{
JOptionPane.showMessageDialog(null,"DETAILS NOT
FOUND","Error",JOptionPane.ERROR_MESSAGE);
}
}
catch(Exception e)
{
// System.out.println(e.getMessage());
}
}
});
}
public static void main(String arg[])
{ c g3=new c();}}
```

RESULT:

Thus the project to implement Passport Automation System using java has been successfully designed.