

**PHINMA UNIVERSITY OF ILOILO**

**College of Information Technology Education**

**APARTMENT MANAGEMENT SYSTEM(AMS)**

**A Mini Capstone Project presented to the College of Information Technology  
Education PHINMA University of Iloilo  
Iloilo City**

**Submitted in partial fulfillment of the requirements for the degree of  
Bachelor of Science in Information Technology  
(IT Project Management, Information Systems(including Database  
Fundamentals)  
and Object-Oriented Programming).**

**MARTINEZ, VALERIE  
BENGAN, RHEA  
DEFANTE, THERENCE HILCEE  
SOLILAP, OSMHAR LENOUX  
DELLOMES, MARVI DIANE**

**October 2023**

**PHINMA UNIVERSITY OF ILOILO**

**College of Information Technology Education**

**I. PROJECT DESCRIPTION**

Our proposed project title is about Apartment Management System (AMS), a digital solution specifically designed to simplify interactions between tenants and especially the landlords, thereby enhancing the overall apartment living and management experience. The AMS will be developed as a desktop application, using Java and Java GUI for the user interface, with the development process facilitated by NetBeans. The system will consist of three primary modules: Room Information, Tenant Information Management, and Payment Management.

**II. PROJECT OBJECTIVES**

**General:**

To design and implement a robust Apartment Management System(AMS) that simplifies and enhances the management of apartments.

**Specific:**

1. To create a user-friendly interface for tenants and especially landlords.
2. To automate processes like rent payments.
3. To facilitate communication between all parties involved.
4. To incorporate a personalized profiling system that tailors the user experience.

**III. Scope and Delimitation**

The Apartment Management System (AMS) will be accessible as a web-based platform, making it compatible across various operating systems. The primary users of this system will be the tenants and property owners.

# **PHINMA UNIVERSITY OF ILOILO**

## **College of Information Technology Education**

### **IV. Software Development Model**

In the development of the Apartment Management System(AMS), our team has chosen the Agile Development Model as a method in our process. The Agile model allows us to break down tasks into smaller, manageable parts, eliminating the need for extensive long-term planning.

The AMS is designed to benefit both tenants and landlords. For tenants, it provides a base for asking questions and receiving clarifications from landlords. For landlords, it simplifies the monitoring of payments and tenants.

### **V. Project Budget**

Category	Cost	Value
People	<ul style="list-style-type: none"><li>· Salary of project Manager</li><li>· Salary of Programmer</li><li>· Salary for the Quality Assurance personnel</li><li>· Salary for Project Designer</li><li>· Salary for the Documentation personnel</li></ul>	<ul style="list-style-type: none"><li>· Php 168 per hour as an average salary</li><li>· Php 168 per hour as an average salary</li></ul>
Physical	<ul style="list-style-type: none"><li>· Mobile devices</li></ul>	<ul style="list-style-type: none"><li>· Php 500 and above</li></ul>

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education

	<ul style="list-style-type: none"><li>· Cablings and computers for project team</li><li>· Office/room for project team</li><li>· Internet Access</li></ul>	<ul style="list-style-type: none"><li>· Php 25,000 and above per unit</li><li>· Php 2,500 per/sqm (10x14)</li><li>· Php 2,000 for monthly internet access</li></ul>
Marketing	<ul style="list-style-type: none"><li>· Advertising</li></ul>	<ul style="list-style-type: none"><li>· Php 30,000 /mo of Digital marketing with freelancers</li></ul>
Organizational	<ul style="list-style-type: none"><li>· System bug down/crash</li></ul>	<ul style="list-style-type: none"><li>· Approximately Php100,000 sales loss per day</li></ul>

### VI. Project Roles & Responsibilities:

NAME	ROLE	RESPONSIBILITY
Osmhar Lenoux Solilap	Project Manager	He is the one who monitors the progress of the project and ensures that it is completed on time and within the budget. He may also develop and maintain the project plan such as implementation, work plans, and resources.

## **PHINMA UNIVERSITY OF ILOILO**

### **College of Information Technology Education**

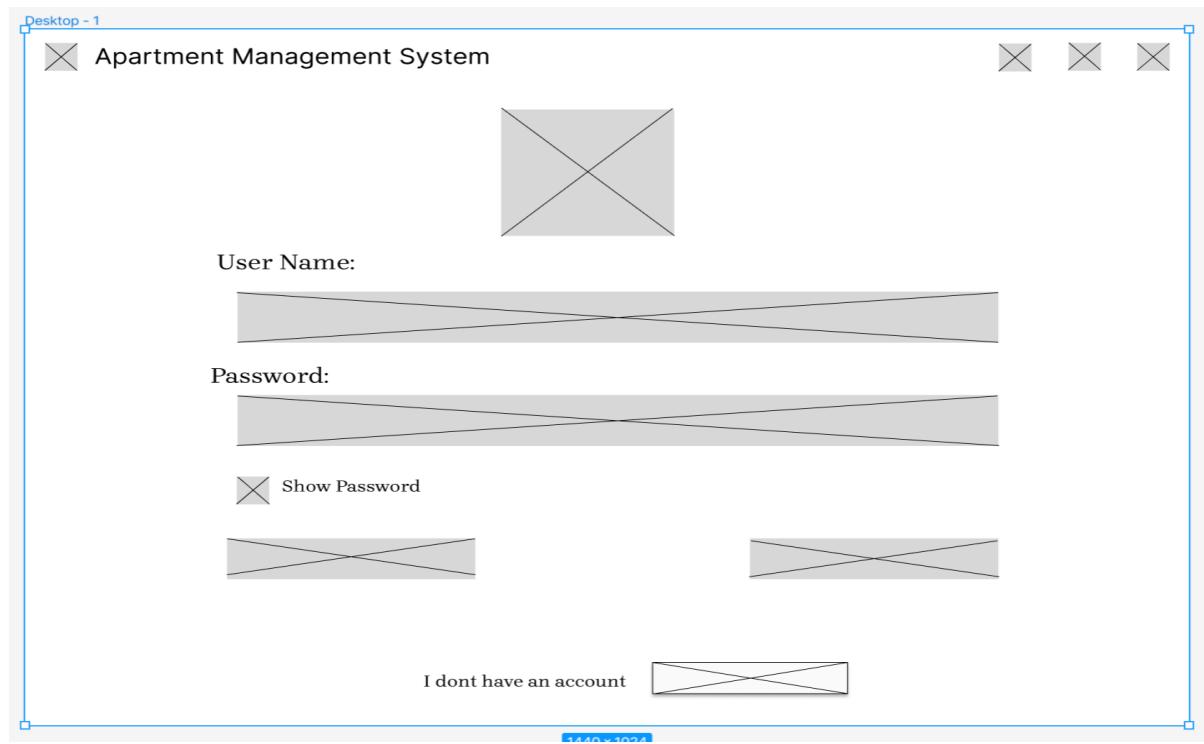
Valerie M. Martinez	Programmer	She is the one who's responsible for making the program.
Therence Hilcee Defante	Quality Assurance	She is the one who's responsible for the visual aspects of our program. The one who creates what will be the visual representation of our application.
Rhea O. Bengan	Designer	She is the one who's responsible for the visual aspects of our program. The one who creates what will be the visual representation of our application.
Marvi Diane Delloomes	Documenter	She is the one who documents everything that's happening to our project from the beginning until we finish it. The one who records and documents all the success and failure in making our program.

### **VII. Implementation**

Wireframe:

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education

Desktop - 5

Registration Form

X X X

### Registration Form

First Name:

Last Name:

Address:

Age:

Gender:

Contact No.:

User Name:

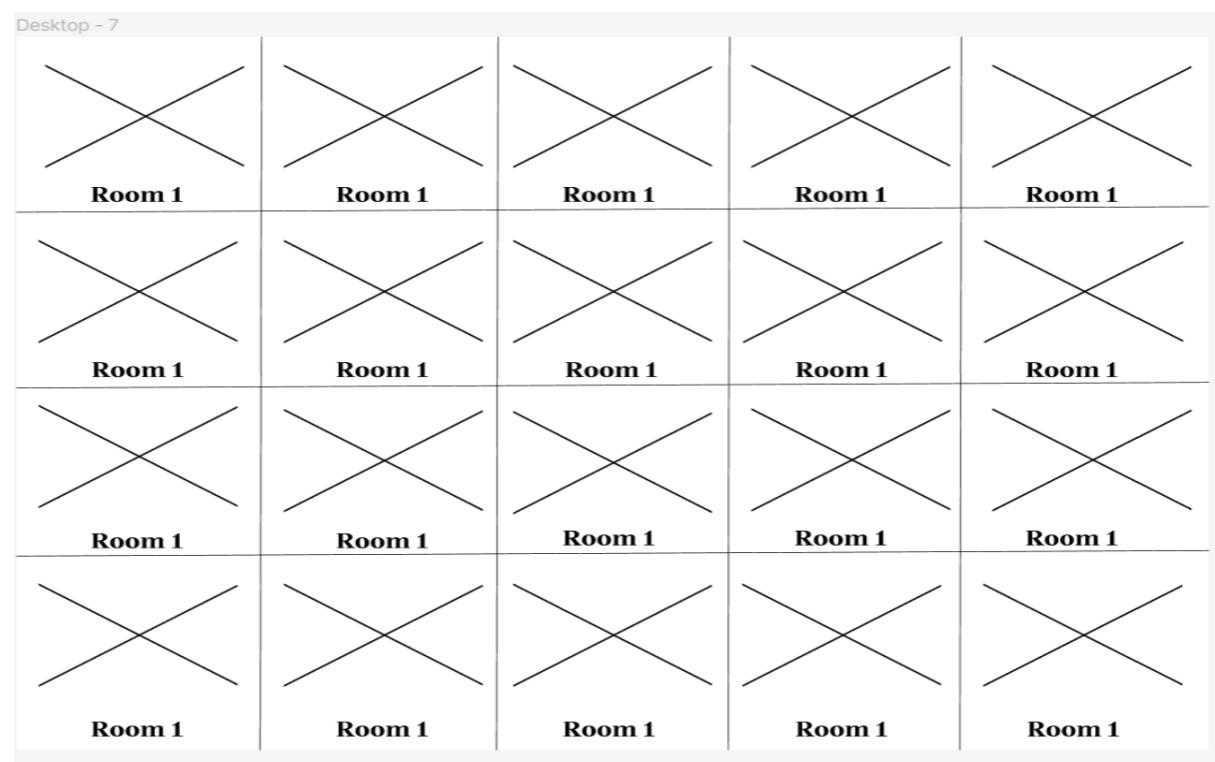
Password:

View Password

X

PHINMA UNIVERSITY OF ILOILO

**College of Information Technology Education**

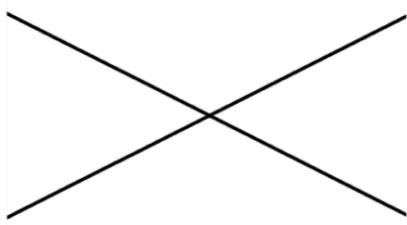


**PHINMA UNIVERSITY OF ILOILO**

**College of Information Technology Education**

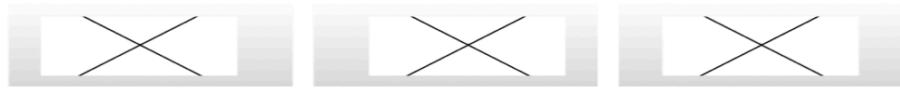
Desktop - 8

**A**



**M**

**S**



# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education

Desktop - 10

Tenant Payment Management

Tenant ID	Amount	Date Paid

First Name:

Last Name:

Room Price:

Date Occupied:

Tenant ID:  ▼ Payment Amount:  Add Payment Payment Lists

**PHINMA UNIVERSITY OF ILOILO**

**College of Information Technology Education**

Payment List

Tenant Payment Management

Tenant ID	Amount	Date Paid
-----------	--------	-----------

First Name:

Last Name:

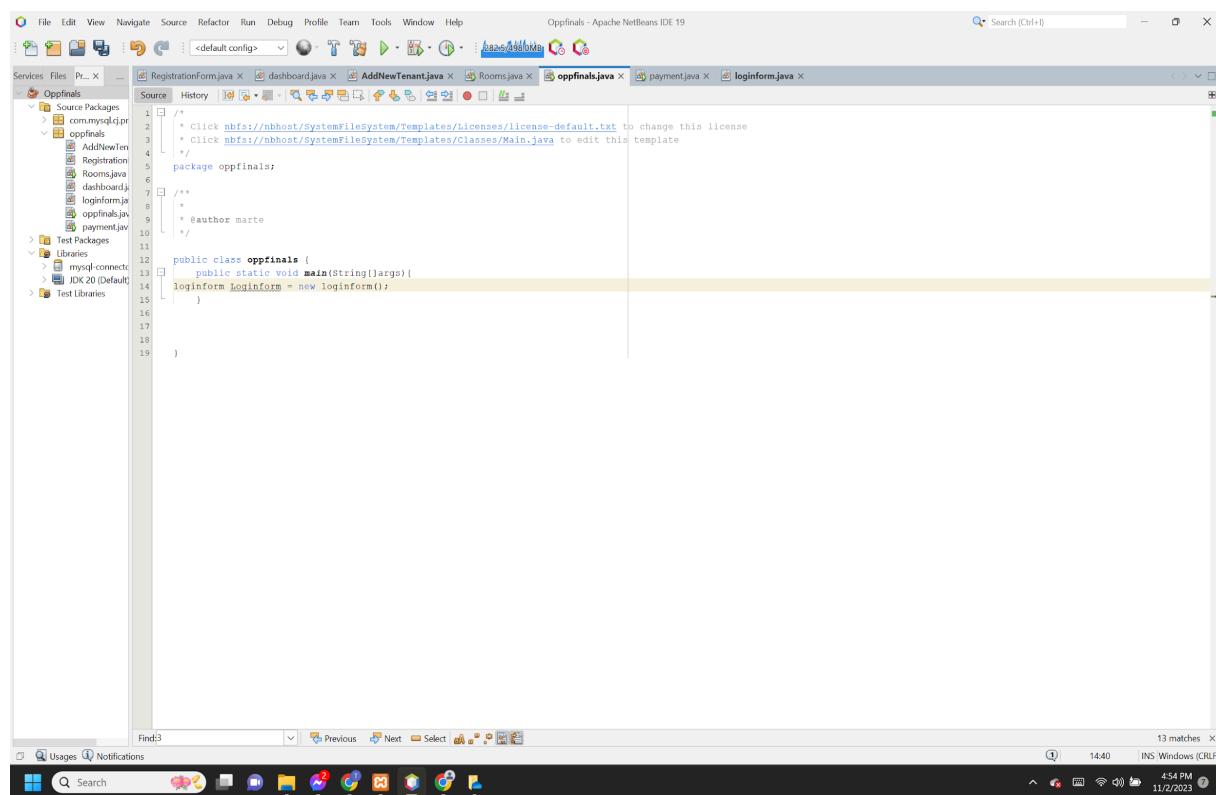
Room Price:

Date Occupied:

Tenant ID:  ▼ Payment Amount:  Add Payment Payment Lists

Java code:

**PHINMA UNIVERSITY OF ILOILO**  
**College of Information Technology Education**



The screenshot shows the Apache NetBeans IDE 19 interface. The left sidebar displays the project structure under the 'Source Packages' section, which includes 'oppfinals' and its sub-packages: 'com.mysql.cj.pr', 'AddNewTen', 'Rooms', 'loginform', 'oppfinals', and 'payment'. The main editor window shows the 'oppfinals.java' file with the following code:

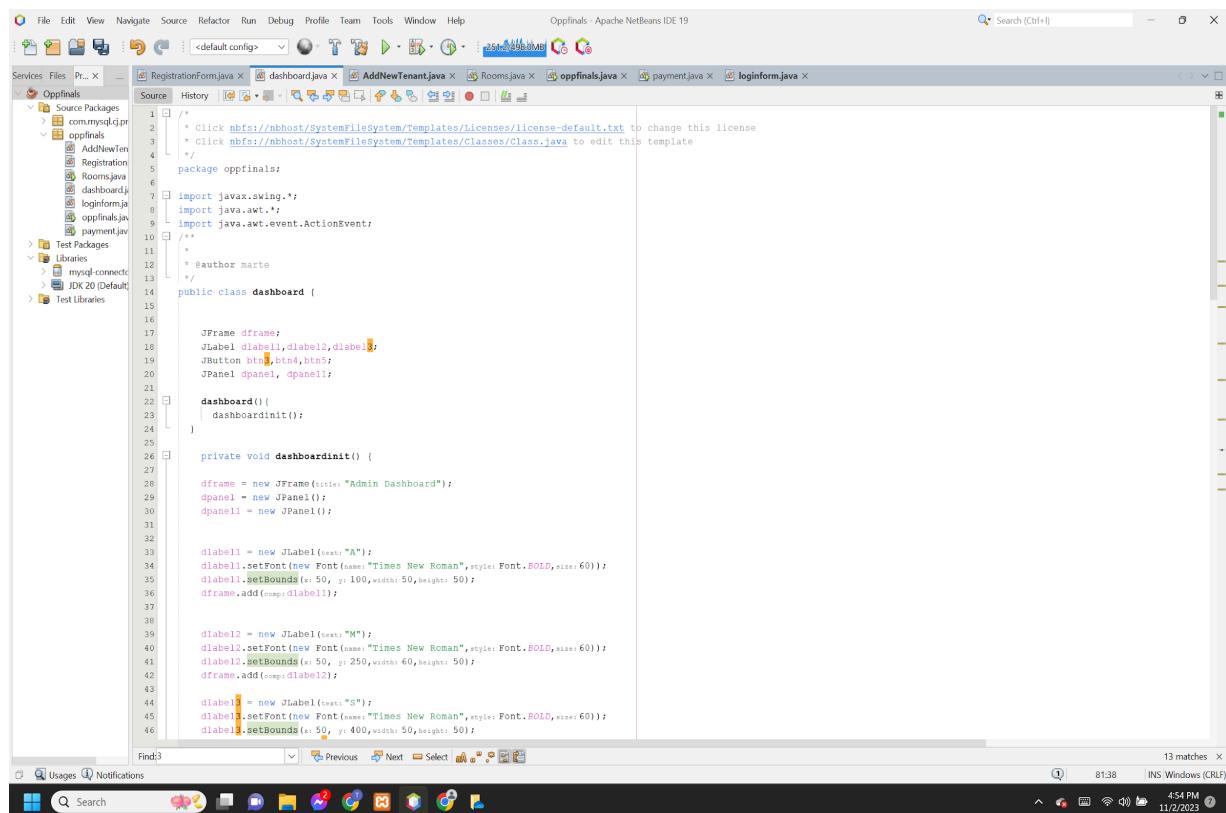
```
/*
 * Click nbfs://nhost/systemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nhost/systemFileSystem/Templates/Classes/Main.java to edit this template
 */
package oppfinals;

/**
 *
 * @author marte
 */
public class oppfinals {
    public static void main(String[] args) {
        loginform loginform = new loginform();
    }
}
```

The code editor has a search bar at the top labeled 'Search (Ctrl+F)' and a status bar at the bottom showing 'Find:3' and '13 matches'. The system tray at the bottom right indicates the date and time as '11/2/2023 4:54 PM'.

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



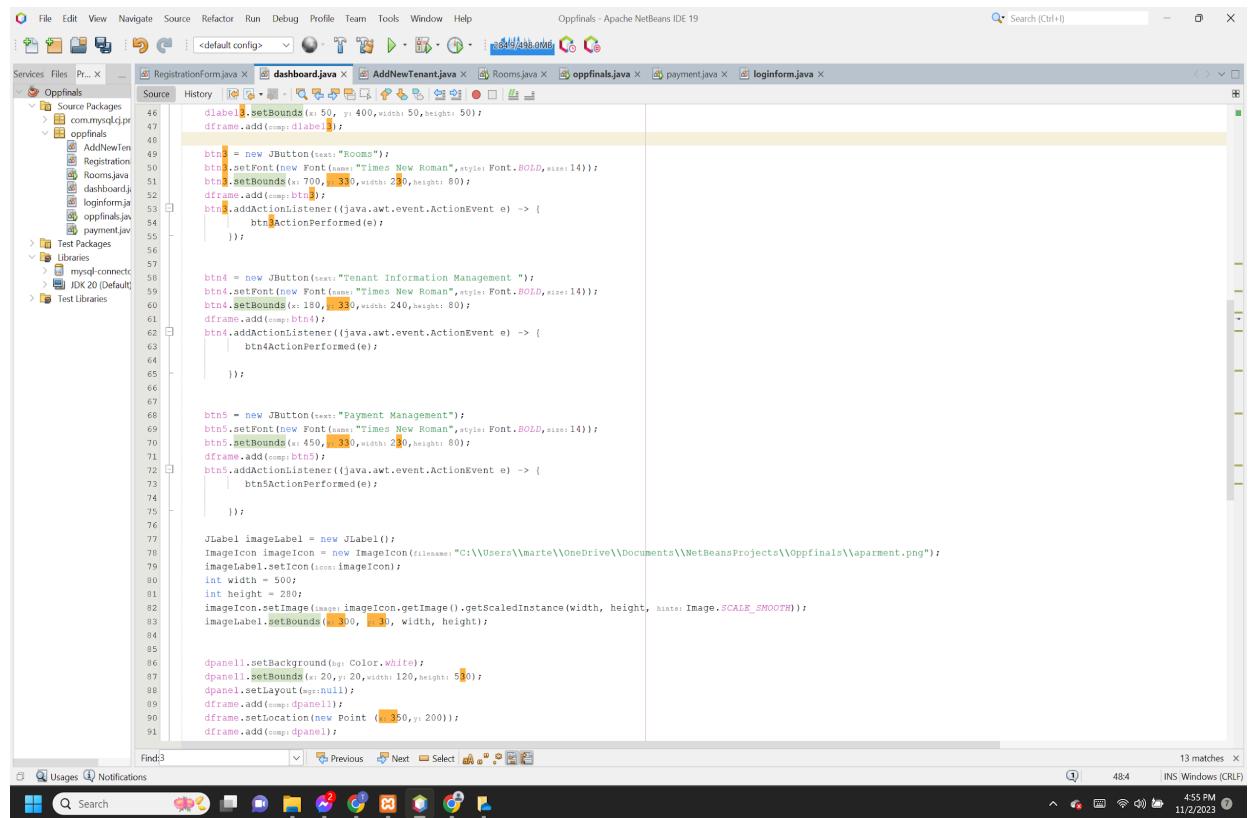
The screenshot shows the Apache NetBeans IDE 19 interface. The left sidebar displays the project structure under 'Source Packages' for the 'Oppfinals' package, which includes files like 'AddNewTen.java', 'Rooms.java', 'dashboard.java', 'loginform.java', 'oppfinals.java', and 'payment.java'. The main editor window shows the 'dashboard.java' file. The code is as follows:

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package oppfinals;
6
7  import javax.swing.*;
8  import java.awt.*;
9  import java.awt.event.ActionEvent;
10 
11 /**
12  * @author marte
13  */
14 public class dashboard {
15 
16     JFrame dframe;
17     JLabel dlabel1,dlabel2,dlabel3;
18     JButton btn1,btn2,btn3,btn4,btn5;
19     JPanel dpanel, dpanel1;
20 
21     dashboard(){
22         dpanel1();
23     }
24 
25     private void dpanel1() {
26 
27         dframe = new JFrame(title: "Admin Dashboard");
28         dpanel = new JPanel();
29         dpanel1 = new JPanel();
30 
31 
32         dlabel1 = new JLabel(text:"A");
33         dlabel1.setFont(new Font(name:"Times New Roman",style:Font.BOLD,size:60));
34         dlabel1.setBounds(x: 50, y: 100,width:50,height: 50);
35         dframe.add(dpanel,dlabel1);
36 
37 
38         dlabel2 = new JLabel(text:"W");
39         dlabel2.setFont(new Font(name:"Times New Roman",style:Font.BOLD,size:60));
40         dlabel2.setBounds(x: 50, y: 250,width:60,height: 50);
41         dframe.add(dpanel,dlabel2);
42 
43 
44         dlabel3 = new JLabel(text:"S");
45         dlabel3.setFont(new Font(name:"Times New Roman",style:Font.BOLD,size:60));
46         dlabel3.setBounds(x: 50, y: 400,width:50,height: 50);
47         dframe.add(dpanel,dlabel3);
48 
49     }
50 
51     public static void main(String[] args) {
52         // TODO code application logic here
53     }
54 }
```

The code defines a 'dashboard' class with a constructor that initializes a frame and adds three large labels ('A', 'W', 'S') to it. The labels are positioned at (x: 50, y: 100), (x: 50, y: 250), and (x: 50, y: 400) respectively, with a size of 60.

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education

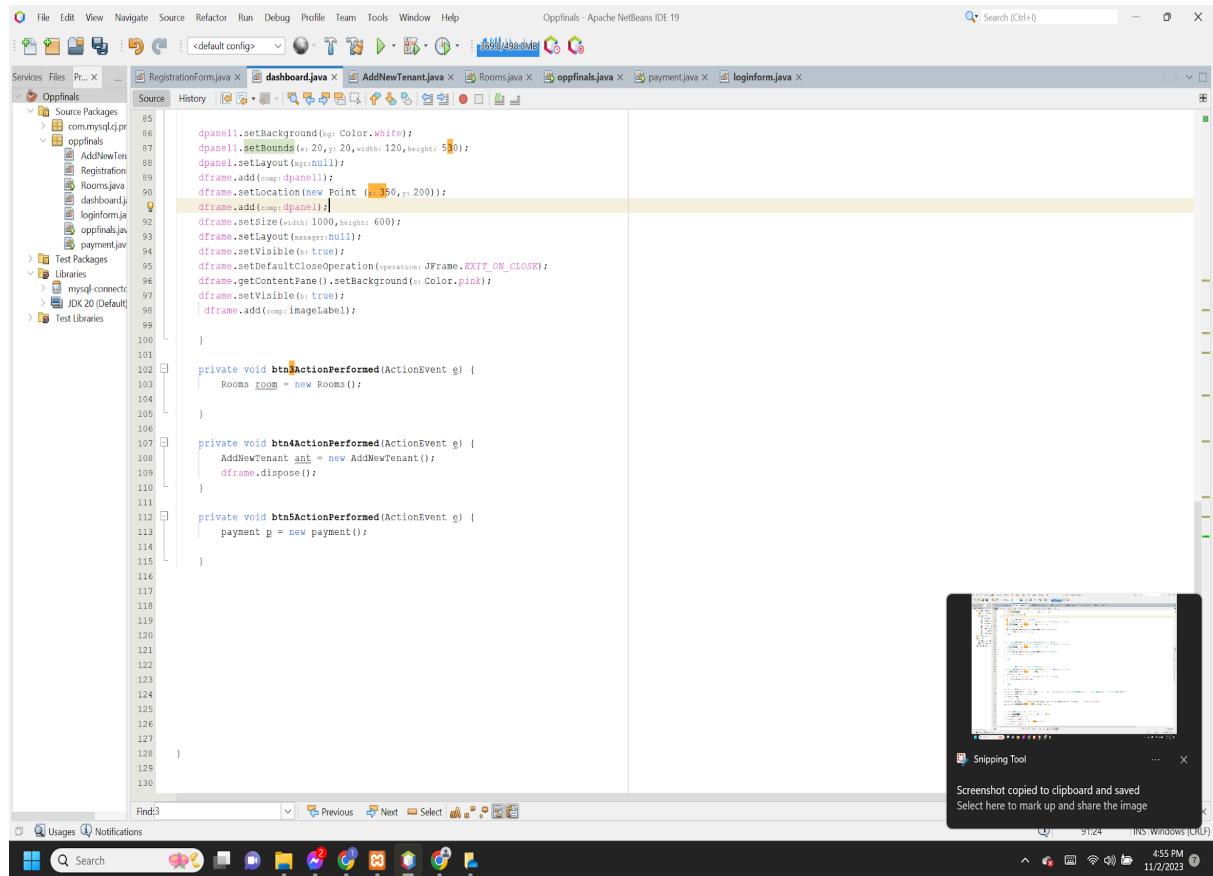


The screenshot shows the Apache NetBeans IDE 19 interface with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Toolbar:** Standard NetBeans toolbar with icons for file operations, search, and project navigation.
- Search Bar:** Search (Ctrl+F) at the top right.
- Project Explorer:** Shows the package structure "Oppfinals" containing "Source Packages" like "com.mysql.jdbc" and "oppfinals" which includes files such as "AddNewTenent.java", "Rooms.java", "dashboard.java", "loginform.java", "oppfinals.java", "payment.java", and "Registration.java".
- Code Editor:** The main editor window displays Java code for a GUI application. The code creates several buttons ("btn1" through "btn5") and a label ("imageLabel"). It sets their text, font (Times New Roman, bold, size 14), and bounds (x, y, width, height). It also adds action listeners to each button. The code ends with setting the background of a panel to white and adding it to a frame.
- Toolbars:** Bottom toolbar with icons for file operations, search, and notifications.
- Status Bar:** Shows "13 matches X" (likely from a previous search), the system clock (4:44 PM), and the date (11/2/2023).

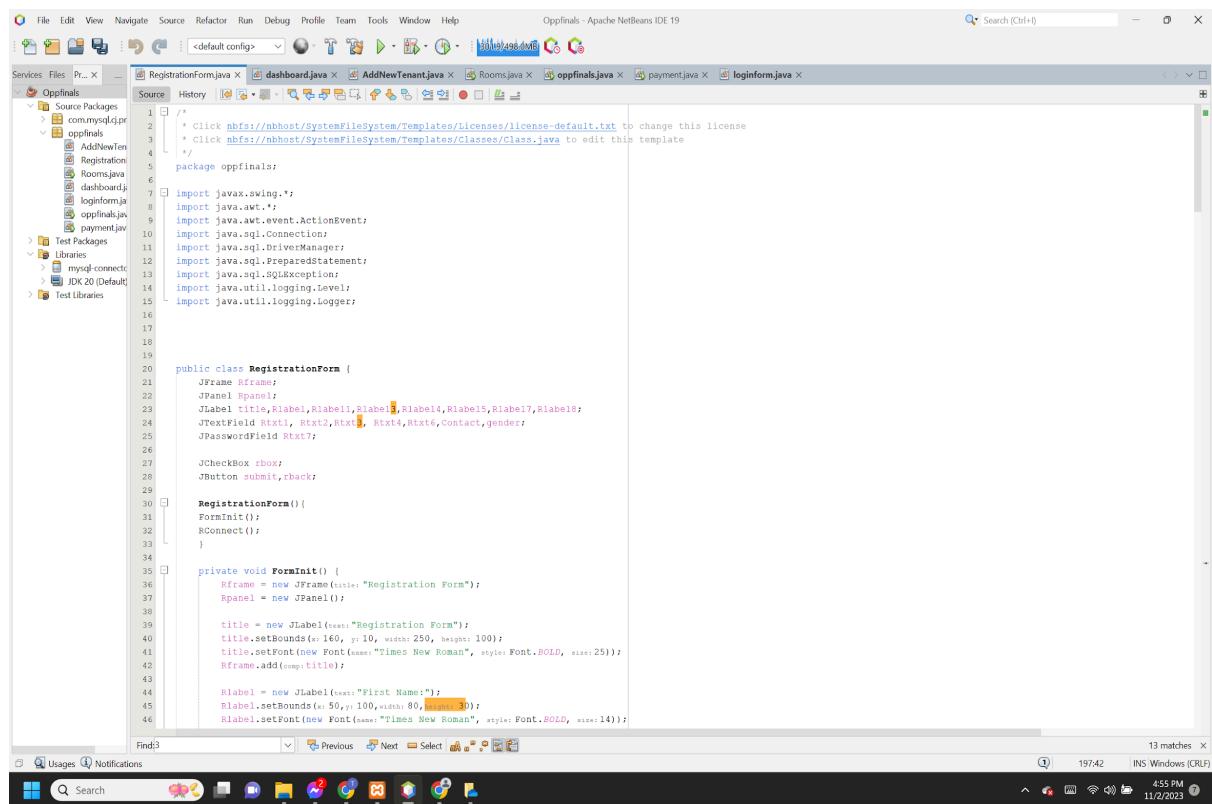
# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



The screenshot shows the Apache NetBeans IDE 19 interface with the file `RegistrationForm.java` open. The code implements a registration form window with various labels, text fields, and buttons. The code uses Java Swing components like `JFrame`, `JPanel`, and `JLabel`. It includes imports for `java.awt`, `java.awt.event`, `java.sql`, and `java.util.logging`. The `FormInit()` method sets up the frame and panel, while the `RegistrationForm()` constructor initializes the form's components.

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package oppfinals;
6
7  import javax.swing.*;
8  import java.awt.*;
9  import java.awt.event.ActionEvent;
10 import java.awt.Connection;
11 import java.sql.DriverManager;
12 import java.sql.PreparedStatement;
13 import java.sql.SQLException;
14 import java.util.logging.Level;
15 import java.util.logging.Logger;
16
17
18
19
20 public class RegistrationForm {
21     JFrame Rframe;
22     JPanel Rpanel;
23     JLabel title, Rlabel1, Rlabel2, Rlabel3, Rlabel4, Rlabel5, Rlabel6, Rlabel7, Rlabel8;
24     JTextField Rtxt1, Rtxt2, Rtxt3, Rtxt4, Rtxt5, Contact, gender;
25     JPasswordField Rtxt7;
26
27     JCheckBox Rcheckbox;
28     JButton submit, Rback;
29
30     RegistrationForm() {
31         FormInit();
32         RConnect();
33     }
34
35     private void FormInit() {
36         Rframe = new JFrame("Registration Form");
37         Rpanel = new JPanel();
38
39         title = new JLabel("Registration Form");
40         title.setBounds(x: 160, y: 10, width: 250, height: 100);
41         title.setFont(new Font("Times New Roman", Font.BOLD, size: 25));
42         Rframe.add(title);
43
44         Rlabel = new JLabel("First Name:");
45         Rlabel.setBounds(x: 50, y: 100, width: 80, height: 30);
46         Rlabel.setFont(new Font("Times New Roman", Font.BOLD, size: 14));
```

**PHINMA UNIVERSITY OF ILOILO**

# **College of Information Technology Education**

The screenshot shows the Apache NetBeans IDE 19 interface with the following details:

- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Toolbar:** Standard Java development tools like New Project, Open, Save, Run, Stop, etc.
- Project Explorer:** Shows the project structure under "Oppfinals".
- Code Editor:** The main window displays the `RegistrationForm.java` file with approximately 92 lines of Java code. The code is used to create a registration form with various labels, text fields, and checkboxes.
- Search Bar:** Top right corner with the text "Search (Ctrl+F)".
- Status Bar:** Bottom right corner showing "50:34 INS Windows (CRU)" and the date "11/2/2023".

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Oppfinals - Apache NetBeans IDE 19
Search (Ctrl+F)
Services Files Pr... X
Source Packages Oppfinals
  +-- Source Packages
    +-- Oppfinals
      +-- AddNewTen...
      +-- Registration
      +-- Rooms.java
      +-- dashboard.java
      +-- loginform.ja...
      +-- oppfinals.java
      +-- payment.java
      +-- RegistrationForm.java
      +-- AddNewTenant.java
      +-- Rooms.java
      +-- oppfinals.java
      +-- payment.java
      +-- loginform.java
  +-- Test Packages
  +-- Libraries
    +-- mysql-connector...
  +-- JDK 20 (Default)
  +-- Test Libraries

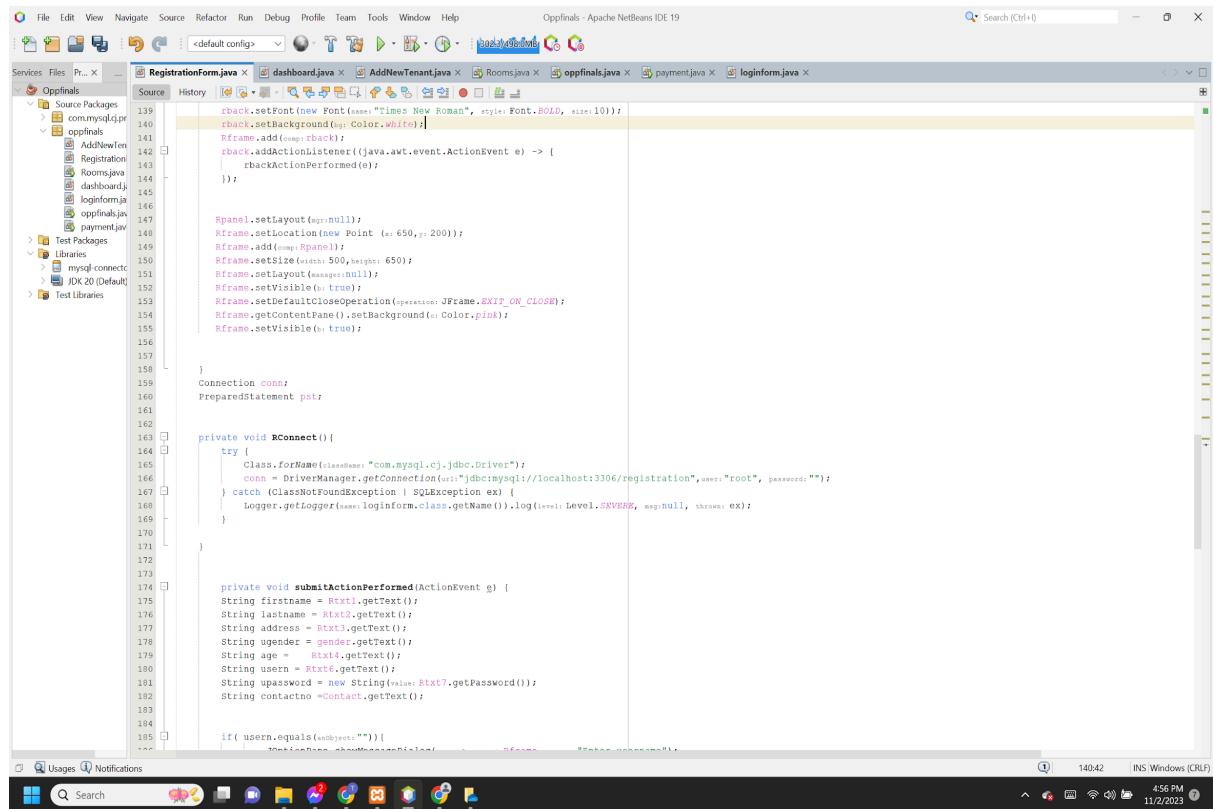
<default config> <...> Services Files Pr... X
Source History Source Packages Oppfinals
  +-- Source Packages
    +-- Oppfinals
      +-- AddNewTen...
      +-- Registration
      +-- Rooms.java
      +-- dashboard.java
      +-- loginform.ja...
      +-- oppfinals.java
      +-- payment.java
      +-- RegistrationForm.java
      +-- AddNewTenant.java
      +-- Rooms.java
      +-- oppfinals.java
      +-- payment.java
      +-- loginform.java
  +-- Test Packages
  +-- Libraries
    +-- mysql-connector...
  +-- JDK 20 (Default)
  +-- Test Libraries

RegistrationForm.java X dashboard.java X AddNewTenant.java X Rooms.java X oppfinals.java X payment.java X loginform.java X
46     JLabel1 = new JLabel(text:"First Name:");
47     JLabel1.setFont(new Font("Times New Roman", style:Font.BOLD, size:14));
48     Rframe.add(comp:LLabel1);
49
50     Rtxt1 = new JTextField();
51     Rtxt1.setBounds(x: 150,y: 100,width: 300,height: 30);
52     Rframe.add(comp:Rtxt1);
53
54     Llabel2 = new JLabel(text:"Last Name:");
55     Llabel2.setFont(new Font("Times New Roman", style:Font.BOLD, size:14));
56     Rframe.add(comp:Llabel2);
57
58     Rtxt2 = new JTextField();
59     Rtxt2.setBounds(x: 150,y: 150,width: 300,height: 30);
60     Rframe.add(comp:Rtxt2);
61
62     Rlabel3 = new JLabel(text:"Address:");
63     Rlabel3.setBounds(x: 50,y: 200,width: 80,height: 30);
64     Rlabel3.setFont(new Font("Times New Roman", style:Font.BOLD, size:14));
65     Rframe.add(comp:Rlabel3);
66
67     Rtxt3 = new JTextField();
68     Rtxt3.setBounds(x: 150,y: 200,width: 300,height: 30);
69     Rframe.add(comp:Rtxt3);
70
71     Rlabel4 = new JLabel(text:"Age:");
72     Rlabel4.setBounds(x: 50,y: 250,width: 80,height: 30);
73     Rlabel4.setFont(new Font("Times New Roman", style:Font.BOLD, size:14));
74     Rframe.add(comp:Rlabel4);
75
76     Rtxt4 = new JTextField();
77     Rtxt4.setBounds(x: 150,y: 250,width: 300,height: 30);
78     Rframe.add(comp:Rtxt4);
79
80     Rlabel5 = new JLabel(text:"Gender:");
81     Rlabel5.setBounds(x: 50,y: 300,width: 80,height: 30);
82     Rlabel5.setFont(new Font("Times New Roman", style:Font.BOLD, size:14));
83     Rframe.add(comp:Rlabel5);
84
85     gender = new JTextField();
86     gender.setBounds(x: 150,y: 300,width: 300,height: 30);
87     Rframe.add(comp:gender);
88
89
90     Contact = new JTextField();
91     Contact.setBounds(x: 150,y: 350,width: 300,height: 30);
92     Rframe.add(comp:Contact);

 50:34 INS Windows (CRU) 11/2/2023 4:56 PM
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



The screenshot shows the Apache NetBeans IDE 19 interface with the 'RegistrationForm.java' file open in the editor. The code is a Java Swing application for user registration. It includes methods for connecting to a MySQL database, handling form submission, and performing various validations. The code uses annotations like @Override, @FXML, and @PostConstruct. It also imports several Java packages such as java.awt, javax.swing, and com.mysql.cj.jdbc.

```
139     rback.setFont(new Font("Times New Roman", Font.BOLD, size:10));
140     rback.setBackground(Color.white);
141     Rframe.add(comp: rback);
142     Rframe.addActionListener(java.awt.event.ActionEvent e) -> {
143         rbackActionPerformed(e);
144     };
145
146     Rpanel.setLayout(null);
147     Rframe.setLocation(new Point (x: 650,y: 200));
148     Rframe.add(comp: Rpanel);
149     Rframe.setSize(width: 500,height: 650);
150     Rframe.setLayout(manager: null);
151     Rframe.setVisible(b: true);
152     Rframe.setDefaultCloseOperation(operation: JFrame.EXIT_ON_CLOSE);
153     Rframe.getContentPane().setBackground(c: Color.pink);
154     Rframe.setVisible(b: true);
155
156
157 }
158 Connection conn;
159 PreparedStatement pst;
160
161
162 private void RConnect() {
163     try {
164         Class.forName(className: "com.mysql.cj.jdbc.Driver");
165         conn = DriverManager.getConnection(url:"jdbc:mysql://localhost:3306/registration",user: "root", password: "");
166     } catch (ClassNotFoundException | SQLException ex) {
167         Logger.getLogger(name: loginform.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
168     }
169 }
170
171
172
173
174 private void submitActionPerformed(ActionEvent g) {
175     String fname = Rtxt1.getText();
176     String lname = Rtxt2.getText();
177     String address = Rtxt3.getText();
178     String gender = gender.getText();
179     String age = Rtxt4.getText();
180     String user = Rtxt6.getText();
181     String upassword = new String(password: Rtxt7.getPassword());
182     String contactno = Contact.getText();
183
184     if( user.equals(subject: "")){
185         JOptionPane.showMessageDialog(null, "User Name is Required!");
186     }
187     else if( upassword.equals(subject: "")){
188         JOptionPane.showMessageDialog(null, "Password is Required!");
189     }
190     else if( !user.equals(subject: "") && !upassword.equals(subject: "") ){
191         try {
192             conn = DriverManager.getConnection(url:"jdbc:mysql://localhost:3306/registration",user: "root", password: "");
193             Statement st = conn.createStatement();
194             String query = "insert into users values ('"+user+"','"+upassword+"','"+fname+"','"+lname+"','"+address+"','"+gender+"','"+age+"','"+contactno+"')";
195             st.executeUpdate(query);
196             JOptionPane.showMessageDialog(null, "User Registered!");
197             Rtxt1.setText("");
198             Rtxt2.setText("");
199             Rtxt3.setText("");
200             gender.setText("");
201             Rtxt4.setText("");
202             Rtxt6.setText("");
203             Rtxt7.setText("");
204             Contact.setText("");
205         } catch (SQLException ex) {
206             Logger.getLogger(name: loginform.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
207         }
208     }
209 }
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education

The screenshot shows the Apache NetBeans IDE 19 interface with the file `RegistrationForm.java` open. The code implements a registration logic using JDBC. It checks if a password is provided, prepares an SQL statement for insertion, and then executes it. If successful, it shows a success message and creates a new `loginform` instance. If failed, it shows an error message. A try-catch block handles `SQLException`.

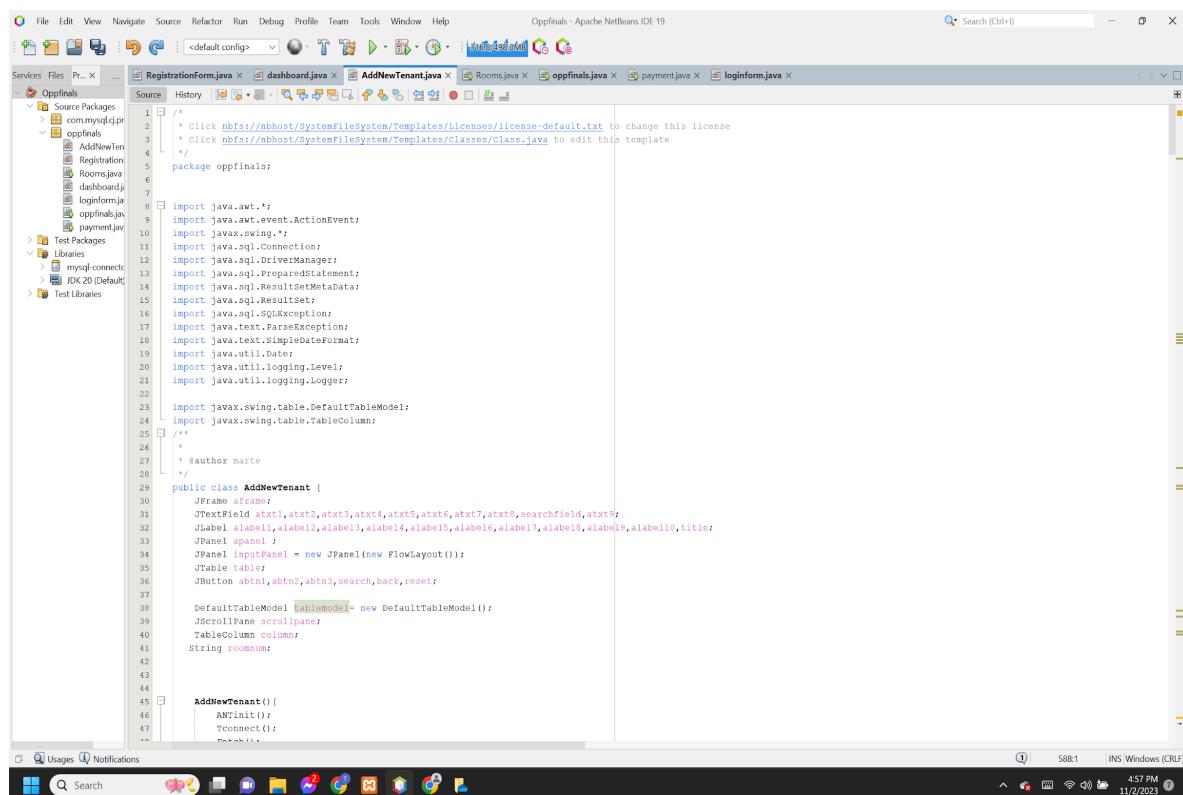
```
187     else if(!upassword.equals("")){
188         JOptionPane.showMessageDialog(parentComponent,Rframe,message:"Enter password");
189     }
190
191     try {
192         pst = conn.prepareStatement("INSERT INTO `userregistrationform`(`first_name`, `last_name`, `gender`, `address`, `age`, `contactnumber`, `username`, `password`)");
193         pst.setString(1,firstname);
194         pst.setString(2,lastname);
195         pst.setString(3,ugender);
196         pst.setString(4,address);
197         pst.setString(5,age);
198         pst.setString(6,contactno);
199         pst.setString(7,username);
200         pst.setString(8,upassword);
201
202         int k = pst.executeUpdate();
203
204         if(k==1){
205             JOptionPane.showMessageDialog(parentComponent,Rframe,message:"success");
206             Rtxt1.setText("");
207             Rtxt1.setEchoChar('');
208             gender.setText("");
209             Rtxt1.setEchoChar('');
210             Rtxt1.setText("");
211             Rtxt1.setEchoChar('');
212             Contact.setText("");
213             Contact.setEchoChar('');
214             Rtxt1.setEchoChar('');
215
216             Rframe.setVisible(false);
217             loginform lgform = new loginform();
218
219         }
220         else{
221             JOptionPane.showMessageDialog(parentComponent,Rframe,message:"registration failed!");
222         }
223
224
225
226     } catch (SQLException ex) {
227         Logger.getLogger(RegistrationForm.class.getName()).log(Level.SEVERE, null, ex);
228     }
229 }
230
231
232     private void rbackActionPerformed(ActionEvent g) {
233         loginform lgform = new loginform();
234         Rframe.dispose();
235
236
237
238
239     private void rboxActionPerformed(ActionEvent g) {
240         if(box.isSelected()){
241             Rtxt7.setEchoChar((char)0);
242         }
243         else{
244             Rtxt7.setEchoChar('*');
245         }
246
247
248
249     }
250
251 }
```

The screenshot shows the Apache NetBeans IDE 19 interface with the file `RegistrationForm.java` open. The code is identical to the one in the previous screenshot, implementing a registration logic using JDBC. It checks if a password is provided, prepares an SQL statement for insertion, and then executes it. If successful, it shows a success message and creates a new `loginform` instance. If failed, it shows an error message. A try-catch block handles `SQLException`.

```
221     else{
222         JOptionPane.showMessageDialog(parentComponent,Rframe,message:"registration failed!");
223     }
224
225
226     } catch (SQLException ex) {
227         Logger.getLogger(RegistrationForm.class.getName()).log(Level.SEVERE, null, ex);
228     }
229 }
230
231
232     private void rbackActionPerformed(ActionEvent g) {
233         loginform lgform = new loginform();
234         Rframe.dispose();
235
236
237
238
239     private void rboxActionPerformed(ActionEvent g) {
240         if(box.isSelected()){
241             Rtxt7.setEchoChar((char)0);
242         }
243         else{
244             Rtxt7.setEchoChar('*');
245         }
246
247
248
249     }
250
251 }
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



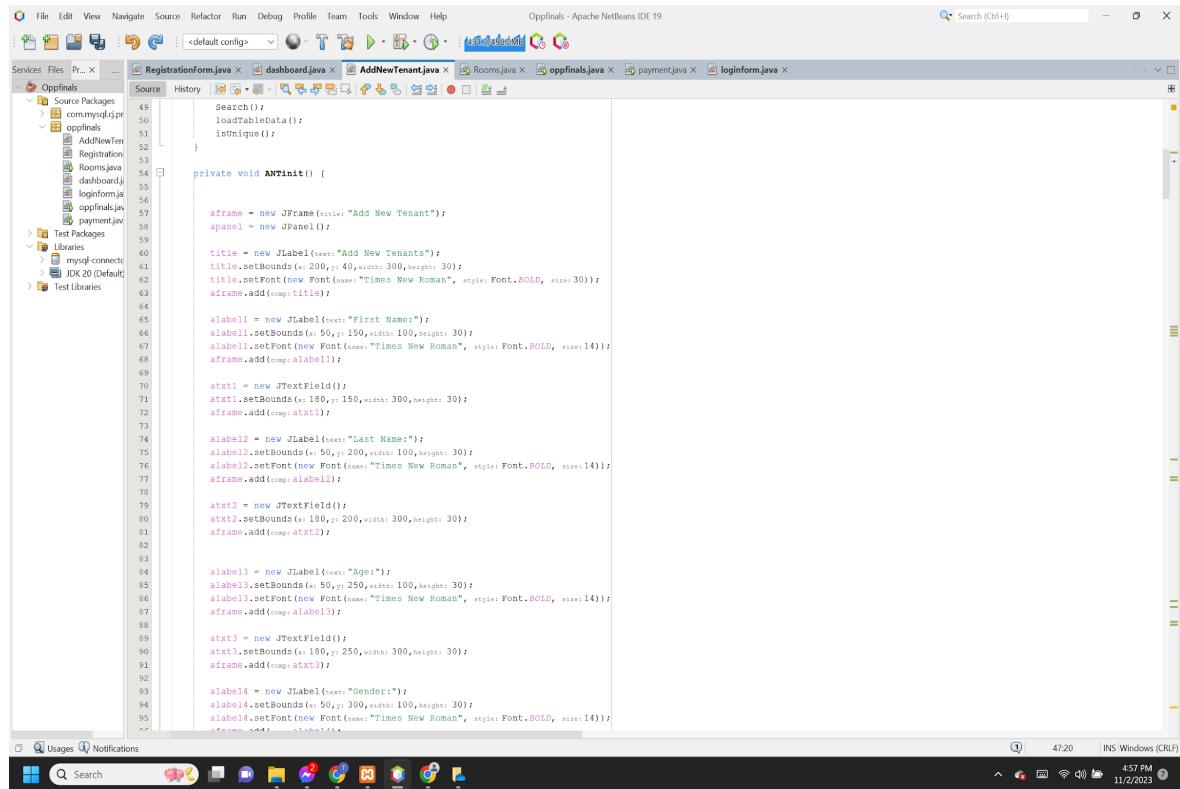
The screenshot shows the Apache NetBeans IDE 19 interface. The title bar reads "Oppfinals - Apache NetBeans IDE 19". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has icons for New Project, Open Project, Save, Run, Stop, and Exit. The left sidebar shows the project structure under "Source Packages" for the "Oppfinals" project, containing files like Registration.java, Rooms.java, oppfinals.java, payment.java, and loginform.java. The right sidebar shows tabs for "Search (Ctrl+F)" and "Find in Path". The main editor area displays the Java code for "AddNewTenant.java". The code imports various Java packages including awt, event, sql, swing, and util. It defines a class "AddNewTenant" with methods for initializing the frame, connecting to the database, and setting up the input panel. A copyright notice at the bottom of the code specifies "© 2012-2013 Mariano G. Diaz". The status bar at the bottom shows "588:1 INS Windows (CR/LF)" and the system tray indicates the date and time as "11/2/2023 4:57 PM".

```
/*
 * Click nbfs://nbshost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbshost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package oppfinals;

import java.awt.*;
import java.awt.event.ActionEvent;
import javax.swing.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSetMetaData;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableColumn;
*/
/*
 * @author marte
 */
public class AddNewTenant {
    JFrame aframe;
    JTextField atxt1,atxt2,atxt3,atxt4,atxt5,atxt6,atxt7,atxt8,searchfield,atxt9;
    JLabel alabel1,alabel2,alabel3,alabel4,alabel5,alabel6,alabel7,alabel8,alabel9,alabel10,title;
    JPanel panel;
    JPanel inputPanel = new JPanel(new FlowLayout());
    JTable table;
    JButton abtn1,abtn2,abtn3,search,back,reset;
    DefaultTableModel tablamodel= new DefaultTableModel();
    JScrollPane scrollpane;
    TableColumn column;
    String roomnum;
    public AddNewTenant() {
        ANInit();
        Tconnect();
        Panel();
    }
}
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



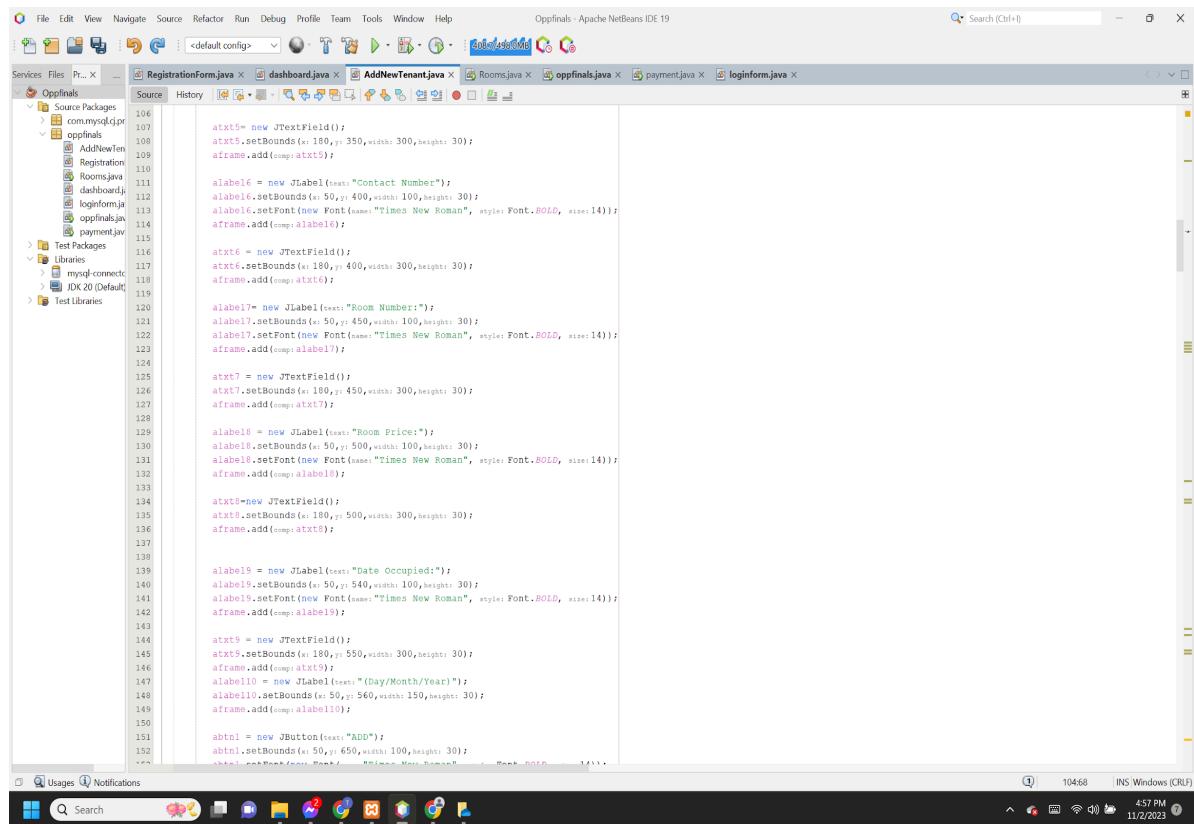
The screenshot shows the Apache NetBeans IDE interface with the following details:

- Title Bar:** PHINMA UNIVERSITY OF ILOILO - Oppfinals - Apache NetBeans IDE 19
- Toolbar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Search Bar:** Search (Ctrl+F)
- Project Explorer:** Shows packages like Source Packages (oppfinals), Test Packages, Libraries, and Test Libraries.
- Code Editor:** Displays the Java code for `AddNewTenant.java`. The code is for creating a GUI window to add new tenants, using `JFrame`, `JLabel`, and `JTextField` components. It includes imports for `java.awt`, `java.awt.event`, and `java.sql`.
- Status Bar:** 47:20, INS Windows (CRF), 4:57 PM, 11/2/2023

```
48     Search();
49     loadTableData();
50     isUnique();
51 }
52 
53 private void ANTinit() {
54 
55     aframe = new JFrame("Add New Tenant");
56     JPanel apanel = new JPanel();
57 
58     title = new JLabel(text:"Add New Tenant");
59     title.setBounds(x: 200, y: 40, width: 300, height: 30);
60     title.setFont(new Font(name:"Times New Roman", style:Font.BOLD, size:30));
61     aframe.add(title);
62 
63     alabel1 = new JLabel(text:"First Name:");
64     alabel1.setBounds(x: 50, y: 150, width: 100, height: 30);
65     alabel1.setFont(new Font(name:"Times New Roman", style:Font.BOLD, size:14));
66     aframe.add(alabel1);
67 
68     atxt1 = new JTextField();
69     atxt1.setBounds(x: 180, y: 150, width: 300, height: 30);
70     aframe.add(alabel1);
71 
72     alabel2 = new JLabel(text:"Last Name:");
73     alabel2.setBounds(x: 50, y: 180, width: 100, height: 30);
74     alabel2.setFont(new Font(name:"Times New Roman", style:Font.BOLD, size:14));
75     aframe.add(alabel2);
76 
77     atxt2 = new JTextField();
78     atxt2.setBounds(x: 180, y: 180, width: 300, height: 30);
79     aframe.add(alabel2);
80 
81     alabel3 = new JLabel(text:"Age:");
82     alabel3.setBounds(x: 50, y: 210, width: 100, height: 30);
83     alabel3.setFont(new Font(name:"Times New Roman", style:Font.BOLD, size:14));
84     aframe.add(alabel3);
85 
86     atxt3 = new JTextField();
87     atxt3.setBounds(x: 180, y: 210, width: 300, height: 30);
88     aframe.add(alabel3);
89 
90     alabel4 = new JLabel(text:"Gender:");
91     alabel4.setBounds(x: 50, y: 240, width: 100, height: 30);
92     alabel4.setFont(new Font(name:"Times New Roman", style:Font.BOLD, size:14));
93     aframe.add(alabel4);
94 
95 }
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



The screenshot shows the Apache NetBeans IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Search Bar:** Search (Ctrl+F).
- Project Tree:** Shows the package structure under "Oppfinals".
- Code Editor:** Displays the Java code for "AddNewTenant.java". The code is a Swing application for adding new tenants, creating a frame with various labels and text fields. Key snippets include:

```
atxt5 = new JTextField();
atxt5.setBounds(x: 180,y: 350,width: 300,height: 30);
aframe.add(comp:atxt5);

label6 = new JLabel(text: "Contact Number");
label6.setBounds(x: 50,y: 400,width: 100,height: 30);
label6.setFont(new Font(name:"Times New Roman", style:Font.BOLD, size:14));
aframe.add(comp:label6);

atxt6 = new JTextField();
atxt6.setBounds(x: 180,y: 400,width: 300,height: 30);
aframe.add(comp:atxt6);

label7 = new JLabel(text: "Room Number");
label7.setBounds(x: 50,y: 450,width: 100,height: 30);
label7.setFont(new Font(name:"Times New Roman", style:Font.BOLD, size:14));
aframe.add(comp:label7);

atxt7 = new JTextField();
atxt7.setBounds(x: 180,y: 450,width: 300,height: 30);
aframe.add(comp:atxt7);

label8 = new JLabel(text: "Room Price:");
label8.setBounds(x: 50,y: 500,width: 100,height: 30);
label8.setFont(new Font(name:"Times New Roman", style:Font.BOLD, size:14));
aframe.add(comp:label8);

atxt8=new JTextField();
atxt8.setBounds(x: 180,y: 500,width: 300,height: 30);
aframe.add(comp:atxt8);

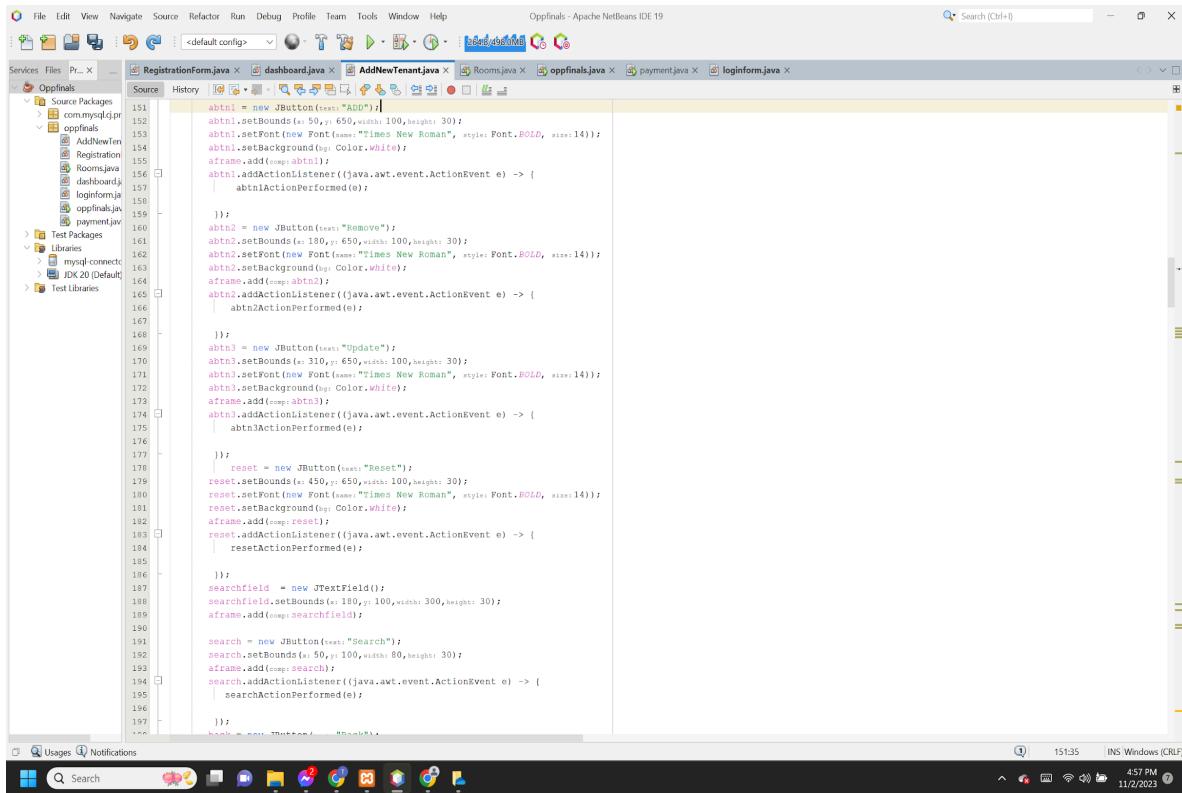
label9 = new JLabel(text:"Date Occupied:");
label9.setBounds(x: 50,y: 540,width: 100,height: 30);
label9.setFont(new Font(name:"Times New Roman", style:Font.BOLD, size:14));
aframe.add(comp:label9);

atxt9 = new JTextField();
atxt9.setBounds(x: 180,y: 550,width: 300,height: 30);
aframe.add(comp:atxt9);
label10 = new JLabel(text:"(Day/Month/Year)");
label10.setBounds(x: 50,y: 560,width: 150,height: 30);
aframe.add(comp:label10);

abtn1 = new JButton(text:"ADD");
abtn1.setBounds(x: 50,y: 650,width: 100,height: 30);
aframe.add(comp:abtn1);
```
- Bottom Status Bar:** 104:68, INS Windows (CR/LF), 457 PM, 11/2/2023.

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education

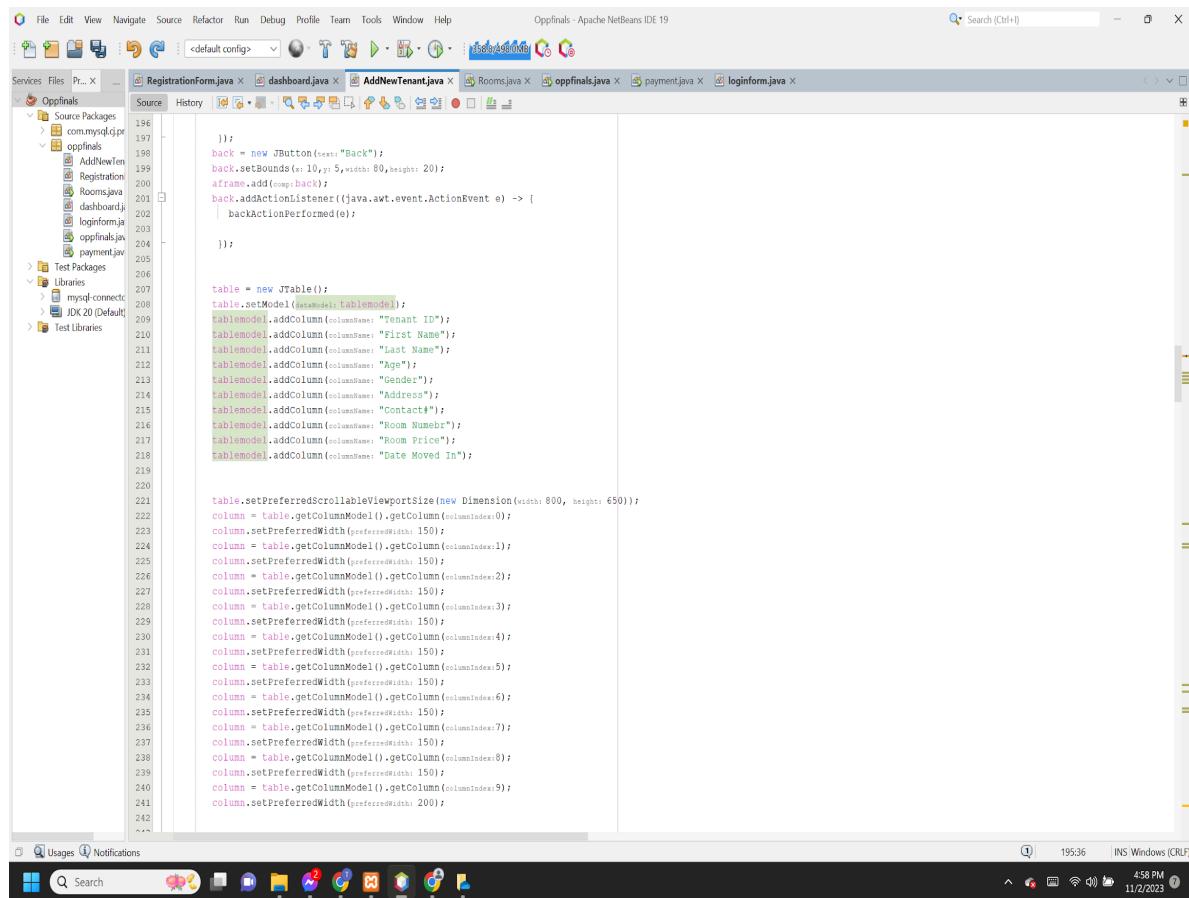


The screenshot shows the Apache NetBeans IDE 19 interface with the title "Oppfinals - Apache NetBeans IDE 19". The main window displays the Java code for "RegistrationForm.java". The code is a part of the "oppfinals" project, specifically the "Source Packages" section. The code defines a class named "RegistrationForm" that extends "JFrame". It contains several JButton instances ("abtn1", "abtn2", "abtn3", "reset", "search") and a JTextField instance ("searchfield"). The code sets various properties for these components, such as text, font, and background color (Color.white). It also adds action listeners to each button to handle events like button presses and search actions.

```
151     abtn1 = new JButton(text:"ADD");
152     abtn1.setBounds(x: 50,y: 650,width: 100,height: 30);
153     abtn1.setFont(new Font(name:"Times New Roman", style:Font.BOLD, size:14));
154     abtn1.setBackground(bg: Color.white);
155     aframe.add(component:abtn1);
156     abtn1.addActionListener((java.awt.event.ActionEvent e) -> {
157         | abtn1ActionPerformed(e);
158     });
159
160     abtn2 = new JButton(text:"Remove");
161     abtn2.setBounds(x: 130,y: 650,width: 100,height: 30);
162     abtn2.setFont(new Font(name:"Times New Roman", style:Font.BOLD, size:14));
163     abtn2.setBackground(bg: Color.white);
164     aframe.add(component:abtn2);
165     abtn2.addActionListener((java.awt.event.ActionEvent e) -> {
166         | abtn2ActionPerformed(e);
167     });
168
169     abtn3 = new JButton(text:"Update");
170     abtn3.setBounds(x: 210,y: 650,width: 100,height: 30);
171     abtn3.setFont(new Font(name:"Times New Roman", style:Font.BOLD, size:14));
172     abtn3.setBackground(bg: Color.white);
173     aframe.add(component:abtn3);
174     abtn3.addActionListener((java.awt.event.ActionEvent e) -> {
175         | abtn3ActionPerformed(e);
176     });
177
178     | reset = new JButton(text:"Reset");
179     reset.setBounds(x: 450,y: 650,width: 100,height: 30);
180     reset.setFont(new Font(name:"Times New Roman", style:Font.BOLD, size:14));
181     reset.setBackground(bg: Color.white);
182     aframe.add(component:reset);
183     reset.addActionListener((java.awt.event.ActionEvent e) -> {
184         | resetActionPerformed(e);
185     });
186
187     | searchfield = new JTextField();
188     searchfield.setBounds(x: 100,y: 100,width: 300,height: 30);
189     aframe.add(component:searchfield);
190
191     search = new JButton(text:"Search");
192     search.setBounds(x: 50,y: 100,width: 80,height: 30);
193     aframe.add(component:search);
194     search.addActionListener((java.awt.event.ActionEvent e) -> {
195         | searchActionPerformed(e);
196     });
197
198     | back = new JButton(text:"Back");
199     back.setBounds(x: 10,y: 100,width: 80,height: 30);
200     aframe.add(component:back);
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



The screenshot shows the Apache NetBeans IDE 19 interface. The left sidebar displays the project structure under the package 'Oppfinals'. The main editor window shows Java code for setting up a JTable. The code creates a table model and adds columns for Tenant ID, First Name, Last Name, Age, Gender, Address, Contact, Room Number, Room Price, and Date Moved In. Each column has a preferred width of 150 pixels. The table is set to have a scrollable view port size of 800x650.

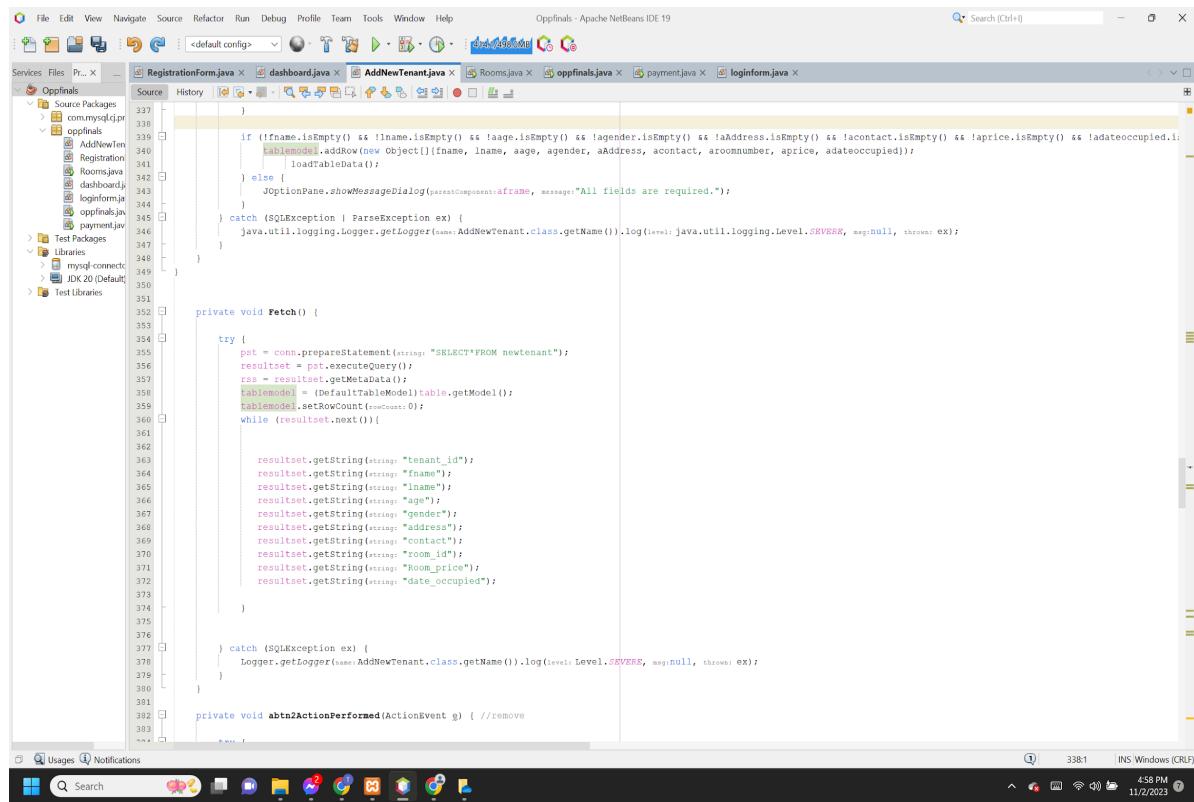
```
197     ];
198     back = new JButton("Back");
199     back.setBounds(10, 5, 80, 20);
200     afame.add(back);
201     back.addActionListener((java.awt.event.ActionEvent e) -> {
202         | backActionPerformed(e);
203     });
204
205
206
207     table = new JTable();
208     table.setModel(tablemodel);
209     tablemodel.addColumn("Tenant ID");
210     tablemodel.addColumn("First Name");
211     tablemodel.addColumn("Last Name");
212     tablemodel.addColumn("Age");
213     tablemodel.addColumn("Gender");
214     tablemodel.addColumn("Address");
215     tablemodel.addColumn("Contact");
216     tablemodel.addColumn("Room Number");
217     tablemodel.addColumn("Room Price");
218     tablemodel.addColumn("Date Moved In");
219
220
221     table.setPreferredScrollableViewportSize(new Dimension(width: 800, height: 650));
222     column = table.getColumnModel().getColumn(columnIndex:0);
223     column.setPreferredWidth(preferredWidth: 150);
224     column = table.getColumnModel().getColumn(columnIndex:1);
225     column.setPreferredWidth(preferredWidth: 150);
226     column = table.getColumnModel().getColumn(columnIndex:2);
227     column.setPreferredWidth(preferredWidth: 150);
228     column = table.getColumnModel().getColumn(columnIndex:3);
229     column.setPreferredWidth(preferredWidth: 150);
230     column = table.getColumnModel().getColumn(columnIndex:4);
231     column.setPreferredWidth(preferredWidth: 150);
232     column = table.getColumnModel().getColumn(columnIndex:5);
233     column.setPreferredWidth(preferredWidth: 150);
234     column = table.getColumnModel().getColumn(columnIndex:6);
235     column.setPreferredWidth(preferredWidth: 150);
236     column = table.getColumnModel().getColumn(columnIndex:7);
237     column.setPreferredWidth(preferredWidth: 150);
238     column = table.getColumnModel().getColumn(columnIndex:8);
239     column.setPreferredWidth(preferredWidth: 150);
240     column = table.getColumnModel().getColumn(columnIndex:9);
241     column.setPreferredWidth(preferredWidth: 200);
242
243
```

**PHINMA UNIVERSITY OF ILOILO**

**College of Information Technology Education**

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education

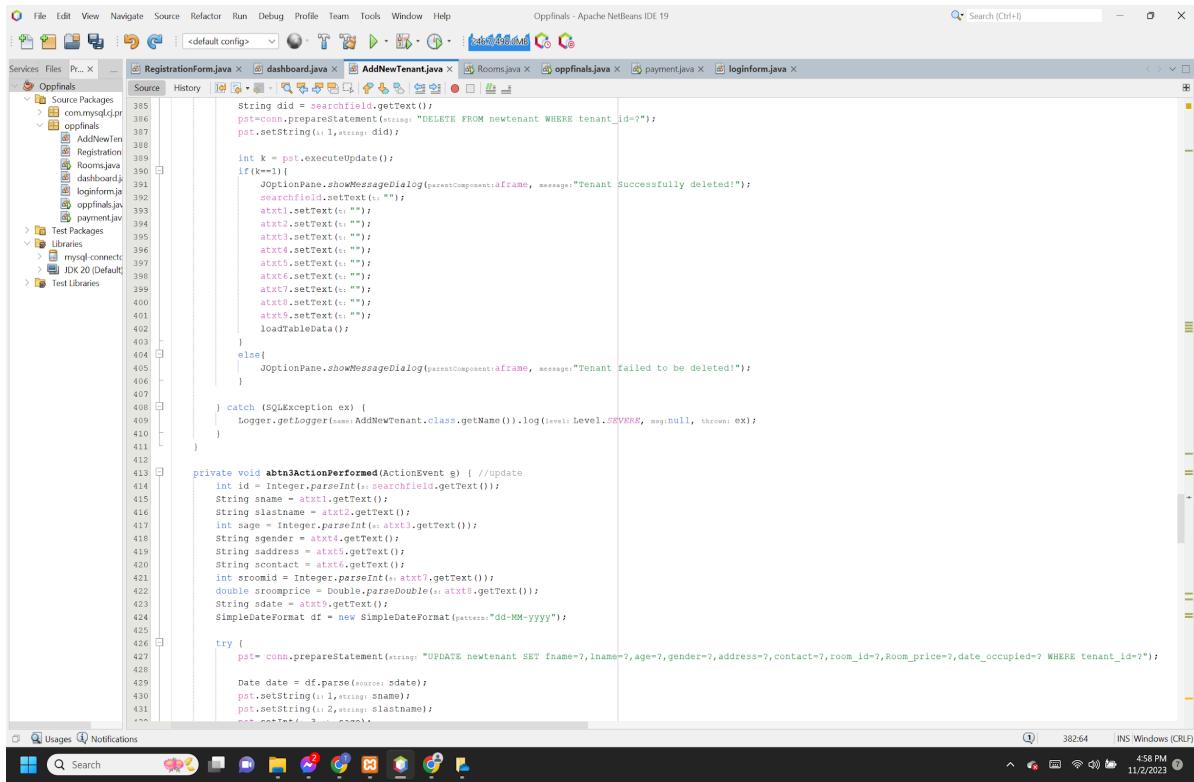


The screenshot shows the Apache NetBeans IDE 19 interface with the project "Oppfinals" open. The code editor displays the file `AddNewTenant.java`. The code implements a database query to fetch tenant information from a table named `newtenant`. The code uses JDBC to prepare a statement, execute it, and then iterate through the result set to print each row's values. Error handling is included to log SQL exceptions.

```
337     }
338     if (!fname.isEmpty() && !lname.isEmpty() && !age.isEmpty() && !gender.isEmpty() && !address.isEmpty() && !contact.isEmpty() && !aprice.isEmpty() && !adateoccupied.isEmpty()) {
339         tablemodel.addRow(new Object[]{fname, lname, age, gender, address, contact, roomnumber, aprice, adateoccupied});
340     } else {
341         JOptionPane.showMessageDialog(parentComponent, "All fields are required.");
342     }
343     catch (SQLException | ParseException ex) {
344         java.util.logging.Logger.getLogger(AddNewTenant.class.getName()).log(Level.SEVERE, null, ex);
345     }
346 }
347
348 private void Fetch() {
349     try {
350         pst = conn.prepareStatement("SELECT * FROM newtenant");
351         resultSet = pst.executeQuery();
352         rss = resultSet.getMetaData();
353         tablemodel = (DefaultTableModel)table.getModel();
354         tablemodel.setRowCount(0);
355         while (resultSet.next()) {
356             resultSet.getString("tenant_id");
357             resultSet.getString("fname");
358             resultSet.getString("lname");
359             resultSet.getString("age");
360             resultSet.getString("gender");
361             resultSet.getString("address");
362             resultSet.getString("contact");
363             resultSet.getString("room_id");
364             resultSet.getString("room_price");
365             resultSet.getString("date_occupied");
366         }
367     }
368     catch (SQLException ex) {
369         Logger.getLogger(AddNewTenant.class.getName()).log(Level.SEVERE, null, ex);
370     }
371 }
372 private void abtn2ActionPerformed(ActionEvent g) { //remove
373     rss.close();
374 }
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



The screenshot shows the Apache NetBeans IDE 19 interface. The left sidebar displays the project structure under the package 'Oppfinals'. The main editor window shows Java code for managing tenants. The code includes methods for deleting and updating tenant records, interacting with a MySQL database via JDBC, and displaying messages to the user using JOptionPane. The code is well-commented and follows standard Java conventions.

```
String did = searchfield.getText();
pst=conn.prepareStatement(string, "DELETE FROM newtenant WHERE tenant_id=?");
pst.setString(1, string: did);

int k = pst.executeUpdate();
if(k>1) {
    JOptionPane.showMessageDialog(parentComponent:afframe, message:"Tenant Successfully deleted!");
    searchfield.setText("");
    atxt1.setText("");
    atxt2.setText("");
    atxt3.setText("");
    atxt4.setText("");
    atxt5.setText("");
    atxt6.setText("");
    atxt7.setText("");
    atxt8.setText("");
    atxt9.setText("");
} else{
    JOptionPane.showMessageDialog(parentComponent:afframe, message:"Tenant failed to be deleted!");
}

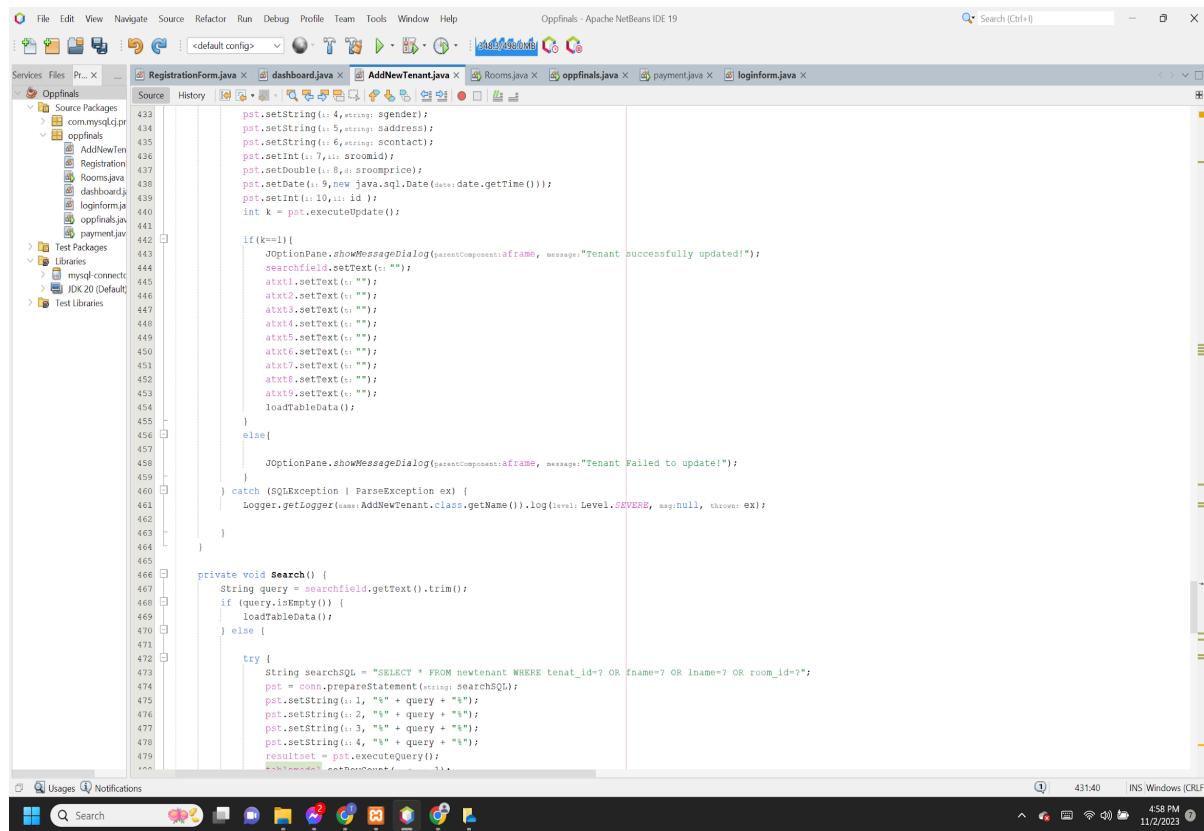
} catch (SQLException ex) {
    Logger.getLogger(AddNewTenant.class.getName()).log(Level.SEVERE, null, ex);
}

private void abtn3ActionPerformed(ActionEvent e) { //update
    int id = Integer.parseInt(searchfield.getText());
    String sname = atxt1.getText();
    String slastname = atxt2.getText();
    int sage = Integer.parseInt(atxt3.getText());
    String gender = atxt4.getText();
    String saddress = atxt5.getText();
    String scontact = atxt6.getText();
    int sroomid = Integer.parseInt(atxt7.getText());
    double sroomprice = Double.parseDouble(atxt8.getText());
    String sdate = atxt9.getText();
    SimpleDateFormat f = new SimpleDateFormat(pattern:"dd-MM-yyyy");

    try {
        pst= conn.prepareStatement(string, "UPDATE newtenant SET fname=?,lname=?,age=?,gender=?,address=?,contact=?,room_id=?,Room_price=?,date_occupied=? WHERE tenant_id=?");
        Date date = df.parse(sdate);
        pst.setString(1, string: sname);
        pst.setString(2, string: slastname);
        pst.setString(3, string: sage);
        pst.setString(4, string: gender);
        pst.setString(5, string: saddress);
        pst.setString(6, string: scontact);
        pst.setInt(7, integer: sroomid);
        pst.setDouble(8, double: sroomprice);
        pst.setDate(9, date);
        pst.setString(10, string: sdate);
    } catch (SQLException ex) {
        Logger.getLogger(AddNewTenant.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education

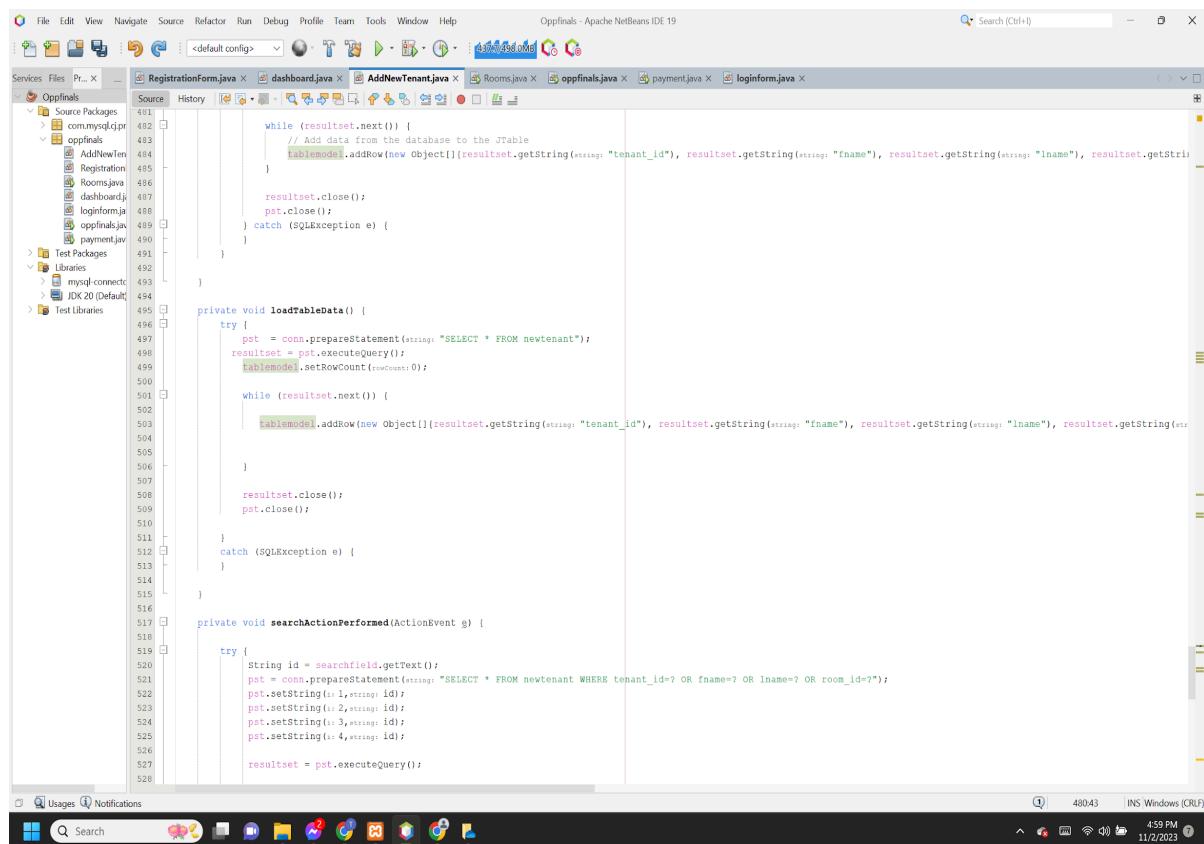


The screenshot shows the Apache NetBeans IDE 19 interface. The left sidebar displays the project structure under the package 'Oppfinals'. The main editor window shows the Java code for the file 'AddNewTenant.java'. The code is a method that updates a database table 'newtenant' based on user input from several JTextFields. It uses PreparedStatement to set values for columns like sgender, saddress, scontact, sroomid, sroomprice, and date. It also handles a search function and logs errors if a SQL exception occurs. The code is annotated with line numbers from 432 to 479.

```
432     pst.setString(4,sgender);
433     pst.setString(5,saddress);
434     pst.setString(6,scontact);
435     pst.setString(7,sroomid);
436     pst.setDouble(8,sroomprice);
437     pst.setDate(9,new java.sql.Date(date.getTime()));
438     pst.setInt(10,id );
439     int k = pst.executeUpdate();
440
441     if(k>=1){
442         JOptionPane.showMessageDialog(parentComponent:aframe, message:"Tenant successfully updated!");
443         searchfield.setText("");
444         axtl1.setText("");
445         axtl2.setText("");
446         axtl3.setText("");
447         axtl4.setText("");
448         axtl5.setText("");
449         axtl6.setText("");
450         axtl7.setText("");
451         axtl8.setText("");
452         axtl9.setText("");
453         axtl10.setText("");
454         loadTableData();
455     }
456     else{
457         JOptionPane.showMessageDialog(parentComponent:aframe, message:"Tenant Failed to update!");
458     }
459 } catch (SQLException | ParseException ex) {
460     Logger.getLogger(AddNewTenant.class.getName()).log(Level.SEVERE, null, ex);
461 }
462 }
463 }
464
465 private void Search() {
466     String query = searchfield.getText().trim();
467     if (query.isEmpty()) {
468         loadTableData();
469     } else {
470
471         try {
472             String searchSQL = "SELECT * FROM newtenant WHERE tenant_id=? OR fname=? OR lname=? OR room_id=?";
473             pst = conn.prepareStatement(searchSQL);
474             pst.setString(1, "%" + query + "%");
475             pst.setString(2, "%" + query + "%");
476             pst.setString(3, "%" + query + "%");
477             pst.setString(4, "%" + query + "%");
478             resultSet = pst.executeQuery();
479         } catch (SQLException ex) {
480             ex.printStackTrace();
481         }
482     }
483 }
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education

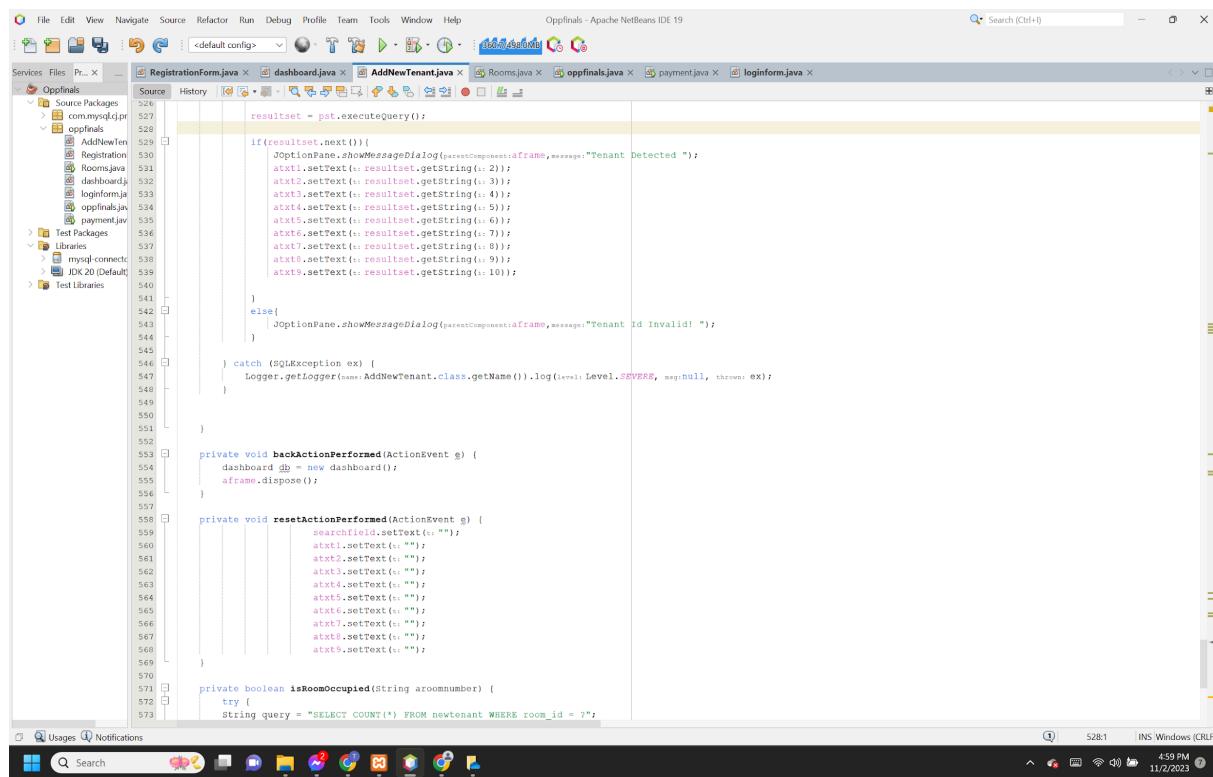


The screenshot shows the Apache NetBeans IDE 19 interface. The title bar reads "Oppfinals - Apache NetBeans IDE 19". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has icons for New, Open, Save, Run, Stop, and others. The left sidebar shows the project structure under "Oppfinals" with packages like com.mysql.jdbc.pr, oppfinals, and sub-packages like AddNewTen, Registration, Rooms, dashboard, loginform, oppfinals, payment, and paymentjav. The main editor window displays Java code for the "AddNewTenant.java" file. The code implements methods for adding new tenant data to a database and performing a search action. The code uses JDBC to execute SQL statements and handle result sets. The bottom status bar shows system information: 48043, INS Windows (CRLF), 4:59 PM, 11/2/2023.

```
491     while (resultset.next()) {
492         // Add data from the database to the JTable
493         tablemodel.addRow(new Object[]{resultset.getString("tenant_id"), resultset.getString("fname"), resultset.getString("lname"), resultset.getString("room_id")});
494     }
495     resultset.close();
496     pst.close();
497     catch (SQLException e) {
498     }
499 }
500 }
501 private void loadTableData() {
502     try {
503         pst = conn.prepareStatement("SELECT * FROM newtenant");
504         resultSet = pst.executeQuery();
505         tablemodel.setRowCount(0);
506
507         while (resultSet.next()) {
508             tablemodel.addRow(new Object[]{resultSet.getString("tenant_id"), resultSet.getString("fname"), resultSet.getString("lname"), resultSet.getString("room_id")});
509         }
510
511         resultSet.close();
512         pst.close();
513     }
514     catch (SQLException e) {
515     }
516 }
517 private void searchActionPerformed(ActionEvent g) {
518     try {
519         String id = searchfield.getText();
520         pst = conn.prepareStatement("SELECT * FROM newtenant WHERE tenant_id=? OR fname=? OR lname=? OR room_id=?");
521         pst.setString(1, id);
522         pst.setString(2, id);
523         pst.setString(3, id);
524         pst.setString(4, id);
525         resultSet = pst.executeQuery();
526
527         while (resultSet.next()) {
528             tablemodel.addRow(new Object[]{resultSet.getString("tenant_id"), resultSet.getString("fname"), resultSet.getString("lname"), resultSet.getString("room_id")});
529         }
530
531         resultSet.close();
532         pst.close();
533     }
534     catch (SQLException e) {
535     }
536 }
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



The screenshot shows the Apache NetBeans IDE 19 interface with the following details:

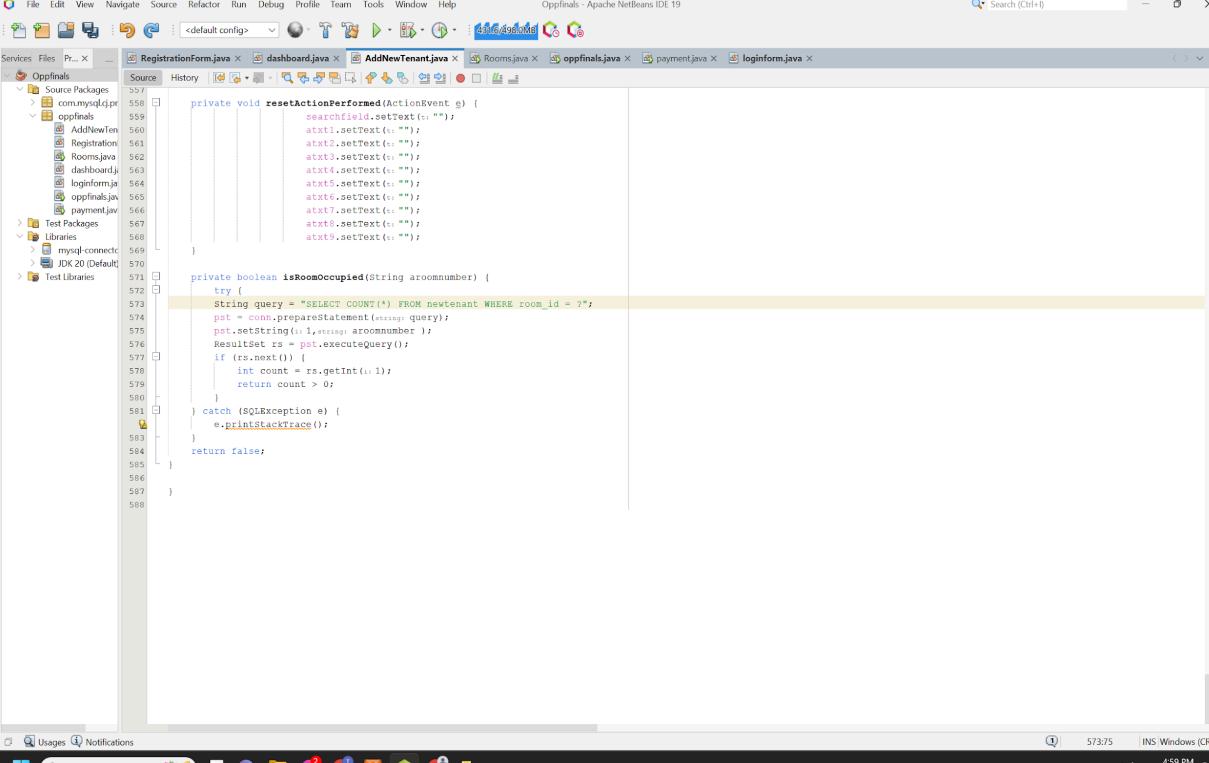
- Title Bar:** Oppinias - Apache NetBeans IDE 19
- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help
- Toolbar:** Standard NetBeans toolbar with icons for file operations.
- Project Explorer:** Shows the project structure under Services. It includes a Source Packages node containing the package `Oppinias` with files: `AddNewTenants.java`, `Dashboard.java`, `LoginForm.java`, `Rooms.java`, `Rooms.java`, `Oppinias.java`, `Payment.java`, and `RegistrationForm.java`. There are also Test Packages, Libraries (including MySQL connector), and Test Libraries.
- Code Editor:** The main window displays the `RegistrationForm.java` file. The code handles tenant detection and room occupancy logic. Key snippets include:

```
resultset = ps.executeQuery();
if(resultset.next()){
    JOptionPane.showMessageDialog(parentComponent,aframe,message,"Tenant Detected ");
    atxt1.setText(resultset.getString(2));
    atxt2.setText(resultset.getString(3));
    atxt3.setText(resultset.getString(4));
    atxt4.setText(resultset.getString(5));
    atxt5.setText(resultset.getString(6));
    atxt6.setText(resultset.getString(7));
    atxt7.setText(resultset.getString(8));
    atxt8.setText(resultset.getString(9));
    atxt9.setText(resultset.getString(10));
}
else{
    JOptionPane.showMessageDialog(parentComponent,aframe,message,"Tenant Id Invalid! ");
}
```

```
private void backActionPerformed(ActionEvent g) {
    dashboard db = new dashboard();
    aframe.dispose();
}
private void resetActionPerformed(ActionEvent g) {
    searchfield.setText("");
    atxt1.setText("");
    atxt2.setText("");
    atxt3.setText("");
    atxt4.setText("");
    atxt5.setText("");
    atxt6.setText("");
    atxt7.setText("");
    atxt8.setText("");
    atxt9.setText("");
}
private boolean isRoomOccupied(String aroomnumber) {
    try {
        String query = "SELECT COUNT(*) FROM newtenant WHERE room_id = ?";
```
- Bottom Status Bar:** Shows system information: 528:1, INS Windows (CRLF), 4:59 PM, 11/2/2023.

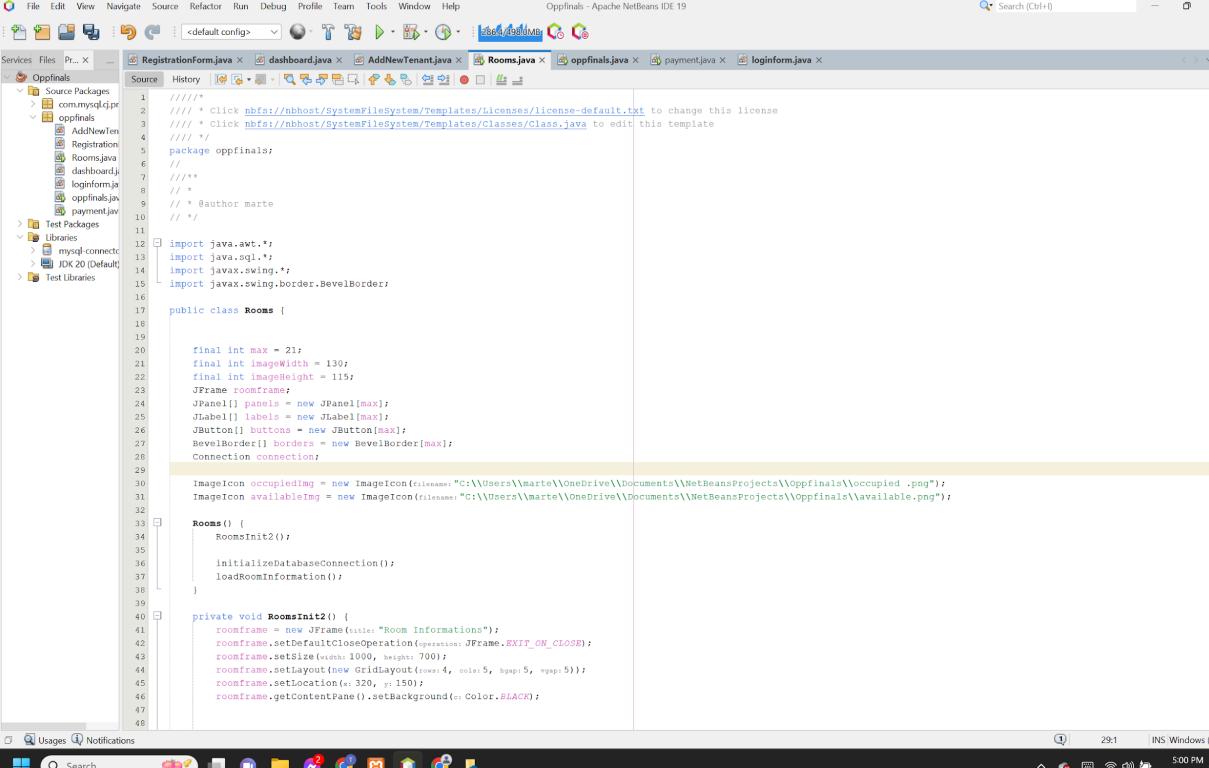
# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



```
private void resetActionPerformed(ActionEvent e) {
    searchfield.setText("");
    atext1.setText("");
    atext2.setText("");
    atext3.setText("");
    atext4.setText("");
    atext5.setText("");
    atext6.setText("");
    atext7.setText("");
    atext8.setText("");
    atext9.setText("");
    atext10.setText("");
    atext11.setText("");
}

private boolean isRoomOccupied(String aroomnumber) {
    try {
        String query = "SELECT COUNT(*) FROM newtenant WHERE room_id = ?";
        pst = conn.prepareStatement(query);
        pst.setString(1, aroomnumber);
        ResultSet rs = pst.executeQuery();
        if (rs.next()) {
            int count = rs.getInt(1);
            return count > 0;
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return false;
}
```



```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package oppfinals;

import java.awt.*;
import java.sql.*;
import javax.swing.*;
import javax.swing.border.BevelBorder;

public class Rooms {

    final int max = 21;
    final int imageWidth = 130;
    final int imageHeight = 115;
    JFrame roomframe;
    JPanel[] panels = new JPanel[max];
    JLabel[] labels = new JLabel[max];
    JButton[] buttons = new JButton[max];
    BevelBorder[] border = new BevelBorder[max];
    Connection connection;

    ImageIcon occupiedImg = new ImageIcon("C:\\Users\\marte\\OneDrive\\Documents\\NetBeansProjects\\oppfinals\\occupied.png");
    ImageIcon availableImg = new ImageIcon("C:\\Users\\marte\\OneDrive\\Documents\\NetBeansProjects\\oppfinals\\available.png");

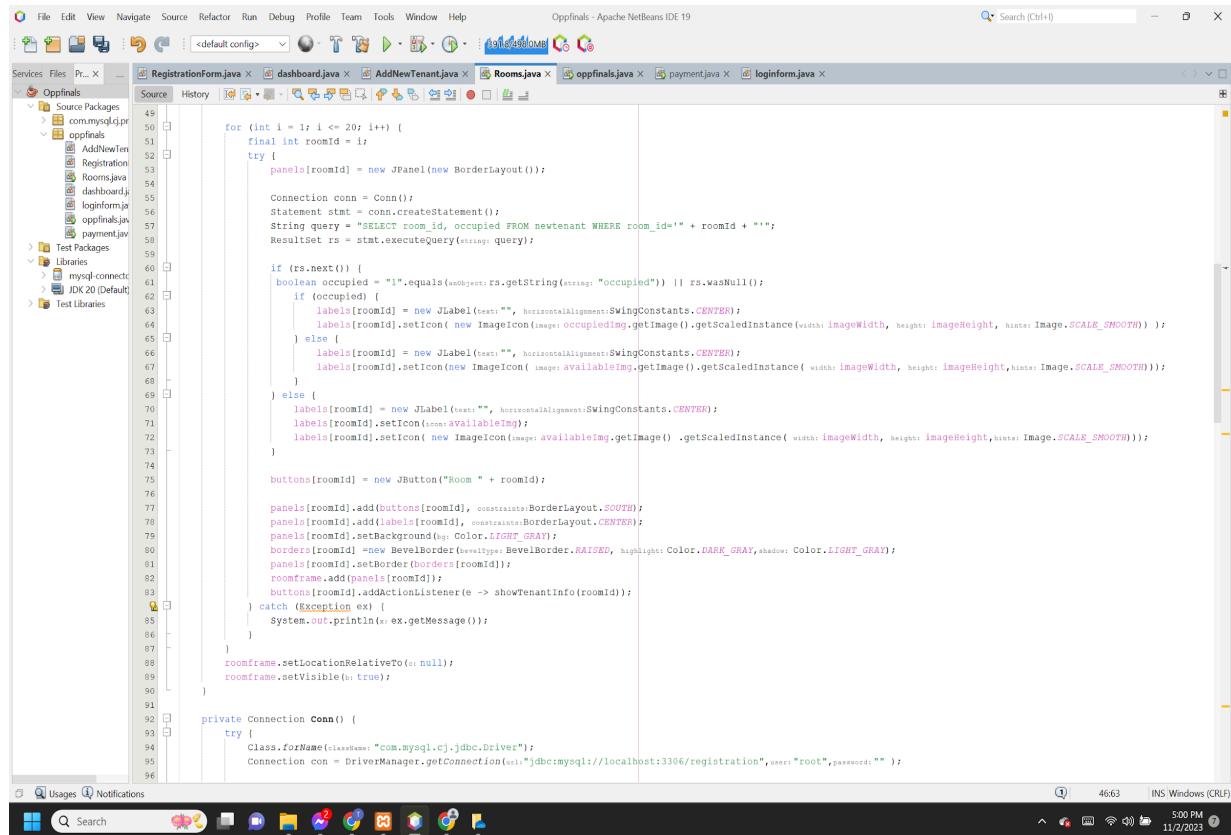
    Rooms() {
        RoomsInit2();
        initializeDatabaseConnection();
        loadRoomInformation();
    }

    private void RoomsInit2() {
        roomframe = new JFrame("Room Informations");
        roomframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        roomframe.setSize(width: 1000, height: 700);
        roomframe.setLayout(new GridLayout(rows: 4, cols: 5, hgap: 5, vgap: 5));
        roomframe.setLocation(x: 320, y: 150);
        roomframe.getContentPane().setBackground(Color.BLACK);
    }

    private void RoomsInit1() {
        roomframe = new JFrame("Room Informations");
        roomframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        roomframe.setSize(width: 1000, height: 700);
        roomframe.setLayout(new GridLayout(rows: 4, cols: 5, hgap: 5, vgap: 5));
        roomframe.setLocation(x: 320, y: 150);
        roomframe.getContentPane().setBackground(Color.BLACK);
    }
}
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



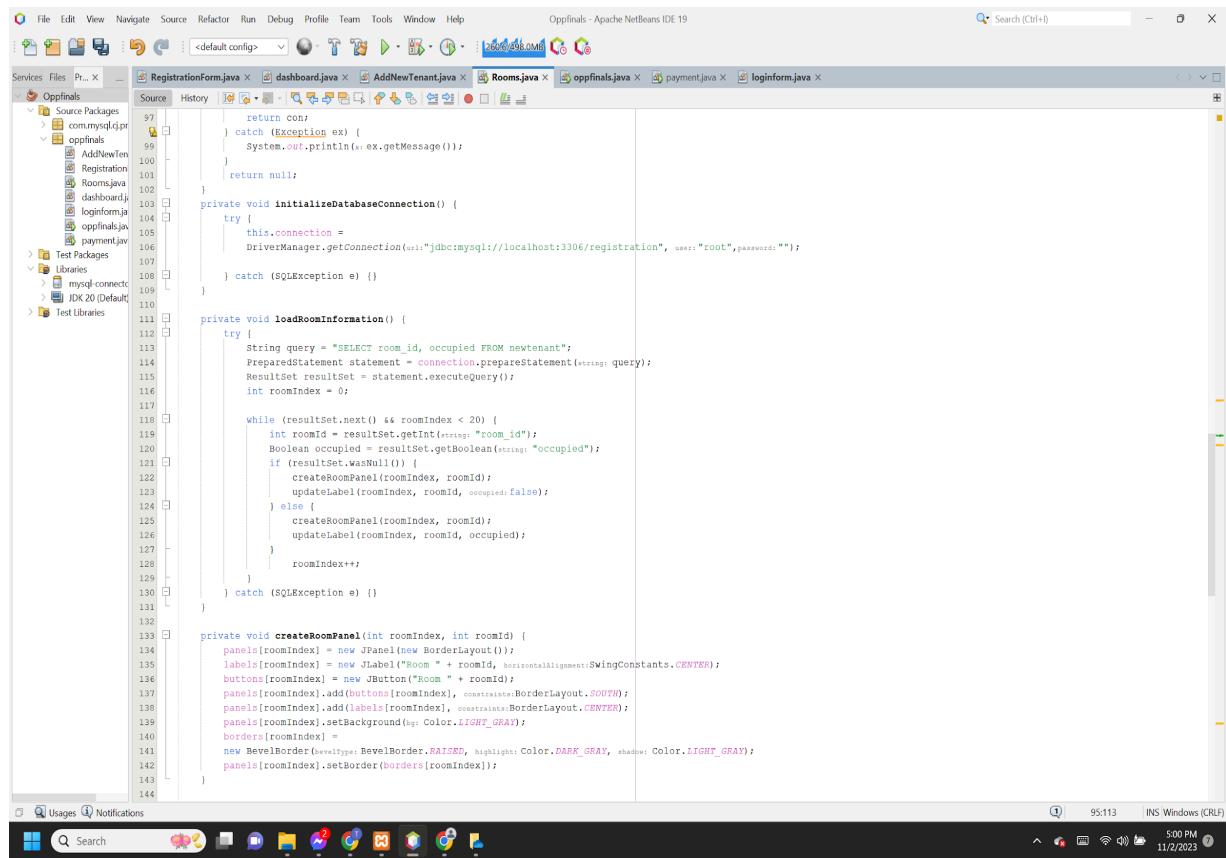
The screenshot shows the Apache NetBeans IDE interface with the following details:

- Title Bar:** Oppfinals - Apache NetBeans IDE 19
- Toolbar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Search Bar:** Search (Ctrl+F)
- Project Explorer:** Shows the project structure under "Oppfinals".
- Code Editor:** Displays the Java code for `RegistrationForm.java`. The code implements a user interface for displaying room availability information. It uses JDBC to query a MySQL database and Swing components like JPanel, JLabel, and JButton.
- Bottom Status Bar:** Shows system information: 4663, INS Windows (CRLF), 5:00 PM, 11/2/2023.

```
45
46     for (int i = 1; i <= 20; i++) {
47         final int roomId = i;
48         try {
49             JPanel[roomId] = new JPanel(new BorderLayout());
50
51             Connection conn = Conn();
52             Statement stat = conn.createStatement();
53             String query = "SELECT room_id, occupied FROM newtenant WHERE room_id=" + roomId + "!";
54             ResultSet rs = stat.executeQuery(string: query);
55
56             if (rs.next()) {
57                 boolean occupied = "1".equals(rs.getString(string: "occupied")) || rs.wasNull();
58                 if (occupied) {
59                     labels[roomId] = new JLabel(text: "", horizontalAlignment: SwingConstants.CENTER);
60                     labels[roomId].setIcon(new ImageIcon(image: occupiedImg.getImage().getScaledInstance(width: imageWidth, height: imageHeight, hints: Image.SCALE_SMOOTH)));
61                 } else {
62                     labels[roomId] = new JLabel(text: "", horizontalAlignment: SwingConstants.CENTER);
63                     labels[roomId].setIcon(new ImageIcon(image: availableImg.getImage().getScaledInstance(width: imageWidth, height: imageHeight, hints: Image.SCALE_SMOOTH)));
64                 }
65             } else {
66                 labels[roomId] = new JLabel(text: "", horizontalAlignment: SwingConstants.CENTER);
67                 labels[roomId].setIcon(icon: availableImg);
68                 labels[roomId].setIcon(new ImageIcon(image: availableImg.getImage().getScaledInstance(width: imageWidth, height: imageHeight, hints: Image.SCALE_SMOOTH)));
69             }
70
71             JButton[roomId] = new JButton("Room " + roomId);
72
73             JPanel[roomId].add(buttons[roomId], constraints: BorderLayout.SOUTH);
74             JPanel[roomId].add(labels[roomId], constraints: BorderLayout.CENTER);
75             JPanel[roomId].setBackground(Color.LIGHT_GRAY);
76             JPanel[roomId].setBorder(borders[roomId]);
77             JPanel[roomId].setBorder(borders[roomId]);
78             roomframe.add(JPanel[roomId]);
79             buttons[roomId].addActionListener(e -> showTenantInfo(roomId));
80
81         } catch (Exception ex) {
82             System.out.println(ex.getMessage());
83         }
84     }
85
86     roomframe.setLocationRelativeTo(null);
87     roomframe.setVisible(true);
88 }
89
90
91 private Connection Conn() {
92     try {
93         Class.forName(className: "com.mysql.cj.jdbc.Driver");
94         Connection con = DriverManager.getConnection(url: "jdbc:mysql://localhost:3306/registration", user: "root", password: "");
95     }
96 }
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



The screenshot shows the Apache NetBeans IDE interface with the following details:

- Title Bar:** PHINMA UNIVERSITY OF ILOILO - Oppfinals - Apache NetBeans IDE 19
- Toolbar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Project Explorer:** Shows the project structure under "Oppfinals".
- Code Editor:** Displays the Java code for the `Rooms.java` file. The code handles database connections, initializes database connections, loads room information, and creates room panels.

```
private void loadRoomInformation() {
    try {
        String query = "SELECT room_id, occupied FROM newtenant";
        PreparedStatement statement = connection.prepareStatement(query);
        ResultSet resultSet = statement.executeQuery();
        int roomIndex = 0;

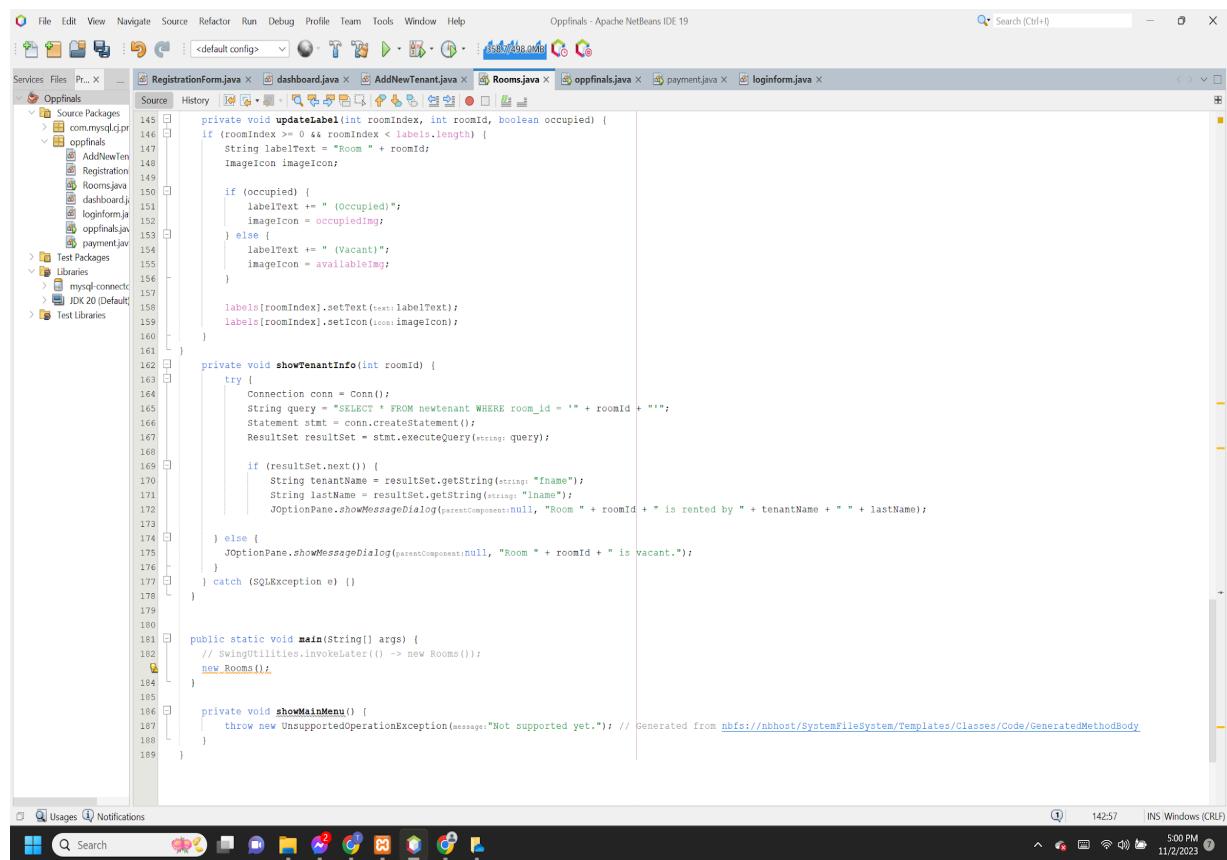
        while (resultSet.next()) {
            int roomId = resultSet.getInt("room_id");
            Boolean occupied = resultSet.getBoolean("occupied");
            if (resultSet.wasNull()) {
                createRoomPanel(roomIndex, roomId);
                updateLabel(roomIndex, roomId, occupied);
            } else {
                createRoomPanel(roomIndex, roomId);
                updateLabel(roomIndex, roomId, occupied);
            }
            roomIndex++;
        }
    } catch (SQLException e) {}
}

private void createRoomPanel(int roomIndex, int roomId) {
    panels[roomIndex] = new JPanel(new BorderLayout());
    labels[roomIndex] = new JLabel("Room " + roomId, SwingConstants.CENTER);
    buttons[roomIndex] = new JButton("Room " + roomId);
    panels[roomIndex].add(buttons[roomIndex], constraints.BorderLayout.SOUTH);
    panels[roomIndex].add(labels[roomIndex], constraints.BorderLayout.CENTER);
    panels[roomIndex].setBackground(Color.LIGHT_GRAY);
    borders[roomIndex] =
        new BevelBorder(BevelBorder.RAISED, highlight:Color.DARK_GRAY, shadow:Color.LIGHT_GRAY);
    panels[roomIndex].setBorder(borders[roomIndex]);
}
```

- Bottom Status Bar:** Shows system information: 95:113, INS Windows (CRLF), 500 PM, 11/2/2023.

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



```
private void updateLabel(int roomIndex, int roomId, boolean occupied) {
    if (roomIndex >= 0 && roomIndex < labels.length) {
        String labelText = "Room " + roomId;
        ImageIcon imageIcon;

        if (occupied) {
            labelText += " (Occupied)";
            imageIcon = occupiedImg;
        } else {
            labelText += " (Vacant)";
            imageIcon = availableImg;
        }

        labels[roomIndex].setText(labelText);
        labels[roomIndex].setIcon(icon:imageIcon);
    }
}

private void showTenantInfo(int roomId) {
    try {
        Connection conn = Conn();
        String query = "SELECT * FROM neuentenant WHERE room_id = '" + roomId + "'";
        Statement stmt = conn.createStatement();
        ResultSet resultSet = stmt.executeQuery(query);

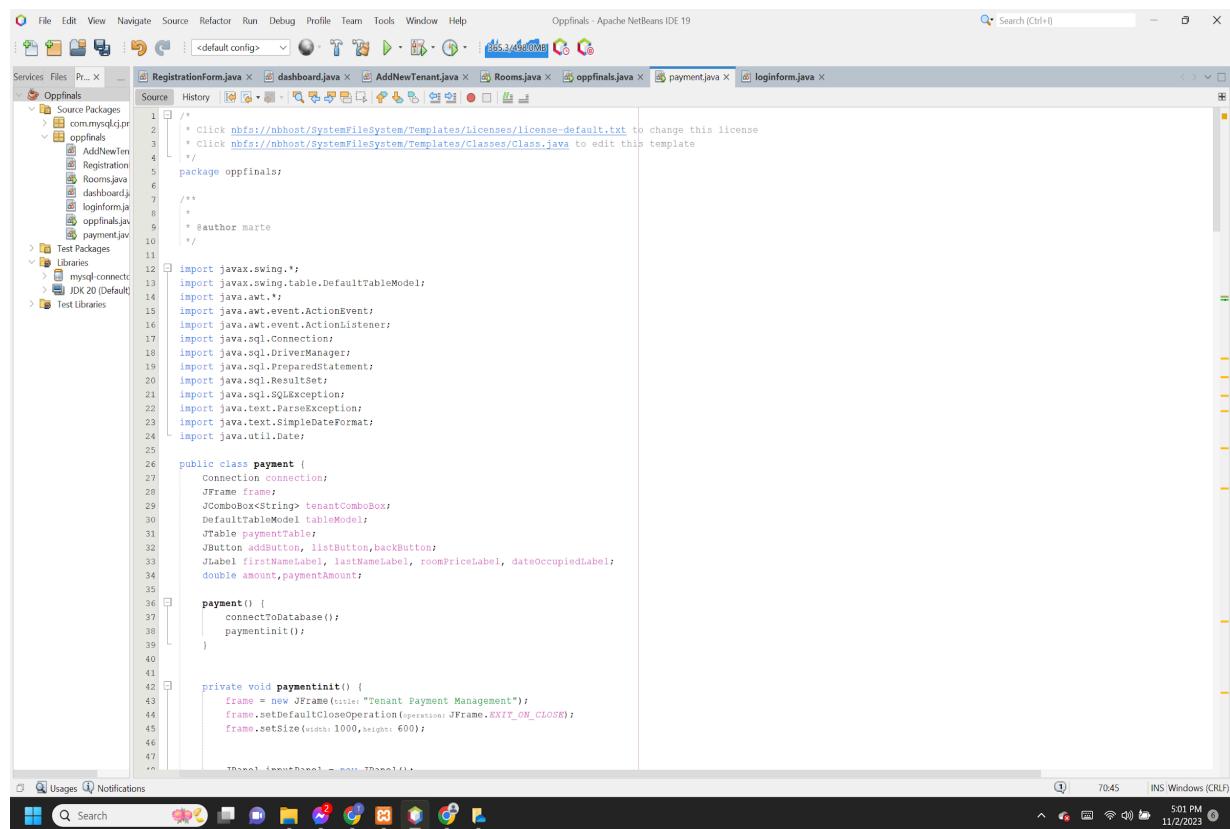
        if (resultSet.next()) {
            String tenantName = resultSet.getString("fname");
            String lastName = resultSet.getString("lname");
            JOptionPane.showMessageDialog(parentComponent:null, "Room " + roomId + " is rented by " + tenantName + " " + lastName);
        } else {
            JOptionPane.showMessageDialog(parentComponent:null, "Room " + roomId + " is vacant.");
        }
    } catch (SQLException e) {}
}

public static void main(String[] args) {
    // SwingUtilities.invokeLater(() -> new Rooms());
    new_Rooms();
}

private void showMainMenu() {
    throw new UnsupportedOperationException(message:"Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/GeneratedMethodBody
}
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



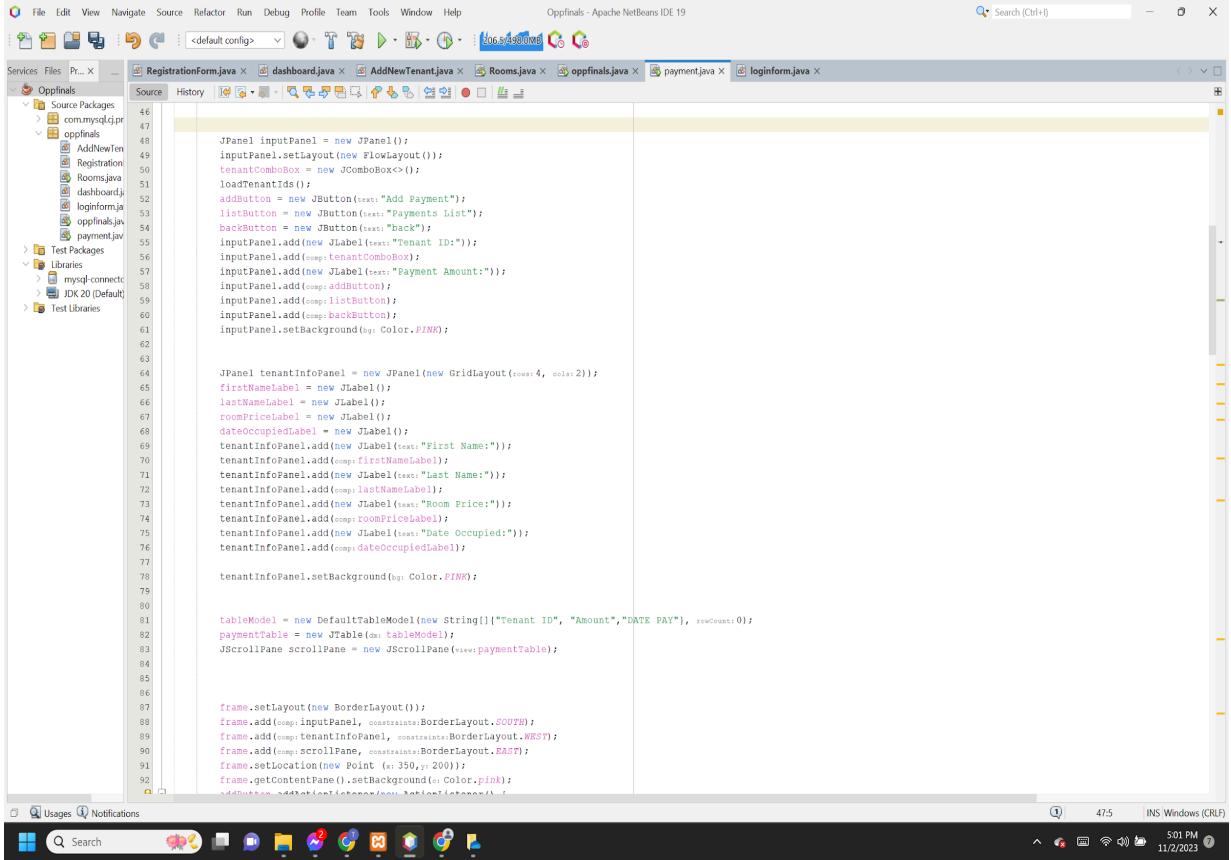
The screenshot shows the Apache NetBeans IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Toolbar:** Standard NetBeans toolbar with icons for file operations, search, and project navigation.
- Project Explorer:** Shows the project structure under "Oppfinals".
  - Source Packages: com.mysql.jdbc, oppfinals (containing AddNewTen, Registration, Rooms.java, dashboard.java, loginform.java, oppfinals.java, payment.java).
  - Test Packages, Libraries, Test Libraries.
- Code Editor:** The main editor window displays the `payment.java` file. The code defines a `payment` class with methods `payment()` and `paymentinit()`. It imports various Java packages related to JDBC, Swing, and AWT.
- Toolbars:** History, Search, Find, Replace, etc.
- Status Bar:** Shows the current file is "Oppfinals - Apache NetBeans IDE 19", the line count is 7045, and the status is "INS Windows (CRLF)".
- Taskbar:** Shows the Windows taskbar with various pinned icons like File Explorer, Edge, and File Manager.
- System Tray:** Shows the date and time as "11/2/2023 5:01 PM".

```
1 // Click nbfs://nbhost/SystemFileSystem/templates/licenses/license-default.txt to change this license
2 // Click nbfs://nbhost/SystemFileSystem/templates/classes/Class.java to edit this template
3 /*
4  * @author marte
5  */
6
7 /**
8  * 
9  */
10 package oppfinals;
11
12 import javax.swing.*;
13 import javax.swing.table.DefaultTableModel;
14 import java.awt.*;
15 import java.awt.event.ActionEvent;
16 import java.awt.event.ActionListener;
17 import java.sql.Connection;
18 import java.sql.DriverManager;
19 import java.sql.PreparedStatement;
20 import java.sql.ResultSet;
21 import java.sql.SQLException;
22 import java.text.ParseException;
23 import java.text.SimpleDateFormat;
24 import java.util.Date;
25
26 public class payment {
27     Connection connection;
28     JFrame frame;
29     JComboBox<String> tenantComboBox;
30     DefaultTableModel tableModel;
31     JTable paymentTable;
32     JButton addButton, listButton, backButton;
33     JLabel firstNameLabel, lastNameLabel, roomPriceLabel, dateOccupiedLabel;
34     double amount, paymentAmount;
35
36     payment() {
37         connectToDatabase();
38         paymentInit();
39     }
40
41     private void paymentInit() {
42         frame = new JFrame("Tenant Payment Management");
43         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
44         frame.setSize(width: 1000, height: 600);
45
46         JPanel inputPanel = new JPanel();
47     }
48 }
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



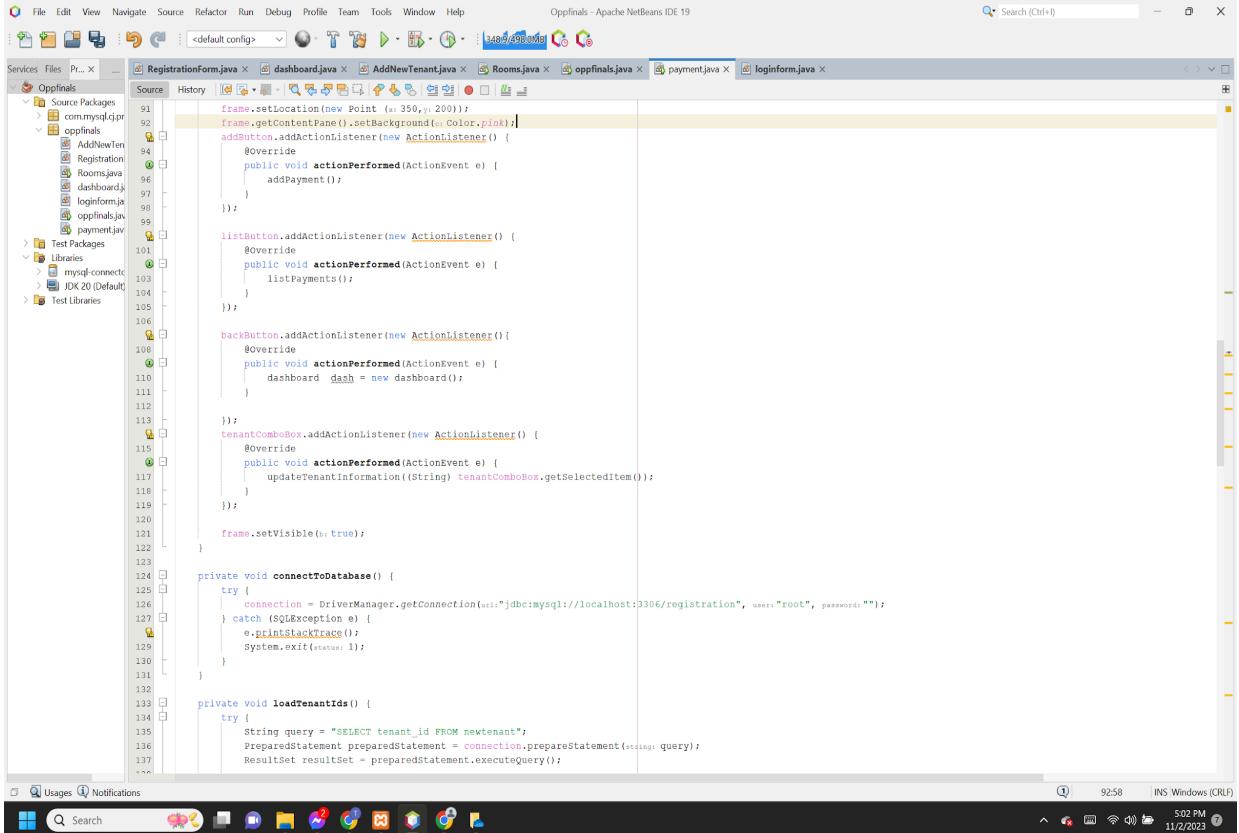
The screenshot shows the Apache NetBeans IDE interface with the following details:

- Title Bar:** Oppfinals - Apache NetBeans IDE 19
- Toolbar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help
- Project Explorer:** Shows the project structure under "Oppfinals".
- Code Editor:** The "payment.java" file is open, displaying Java code for a payment interface.
- Bottom Status Bar:** Shows system information: 47.5, INS Windows (CRF), 5:01 PM, 11/2/2023.

```
46 JPanel inputPanel = new JPanel();
47 inputPanel.setLayout(new FlowLayout());
48 tenantComboBox = new JComboBox<>();
49 loadTenantsIDs();
50 addButton = new JButton(text:"Add Payment");
51 listButton = new JButton(text:"Payments List");
52 backButton = new JButton(text:"Back");
53 inputPanel.add(new JLabel(text:"Tenant ID:"));
54 inputPanel.add(tenantComboBox);
55 inputPanel.add(new JLabel(text:"Payment Amount:"));
56 inputPanel.add(comp.addButton);
57 inputPanel.add(comp.listButtons);
58 inputPanel.add(comp.backButtons);
59 inputPanel.setBackground(bg, Color.PINK);
60
61
62
63 JPanel tenantInfoPanel = new JPanel(new GridLayout(rows:4, cols:2));
64 firstNameLabel = new JLabel();
65 lastNameLabel = new JLabel();
66 roomPriceLabel = new JLabel();
67 dateOccupiedLabel = new JLabel();
68 tenantInfoPanel.add(new JLabel(text:"First Name:"));
69 tenantInfoPanel.add(comp.firstNameLabel);
70 tenantInfoPanel.add(new JLabel(text:"Last Name:"));
71 tenantInfoPanel.add(comp.lastNameLabel);
72 tenantInfoPanel.add(new JLabel(text:"Room Price:"));
73 tenantInfoPanel.add(comp.roomPriceLabel);
74 tenantInfoPanel.add(new JLabel(text:"Date Occupied:"));
75 tenantInfoPanel.add(comp.dateOccupiedLabel);
76
77
78 tenantInfoPanel.setBackground(bg, Color.PINK);
79
80
81 tableModel = new DefaultTableModel(new String[][]{"Tenant ID", "Amount", "DATE PAY"}, rowCount: 0);
82 paymentTable = new JTable(m, tableModel);
83 JScrollPane scrollPane = new JScrollPane(paymentTable);
84
85
86
87 frame.setLayout(new BorderLayout());
88 frame.add(comp.inputPanel, constraints:BorderLayout.SOUTH);
89 frame.add(comp.tenantInfoPanel, constraints:BorderLayout.WEST);
90 frame.add(comp.scrollPane, constraints:BorderLayout.EAST);
91 frame.setLocation(new Point(x: 350, y: 200));
92 frame.getContentPane().setBackground(bg, Color.pink);
93 frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education

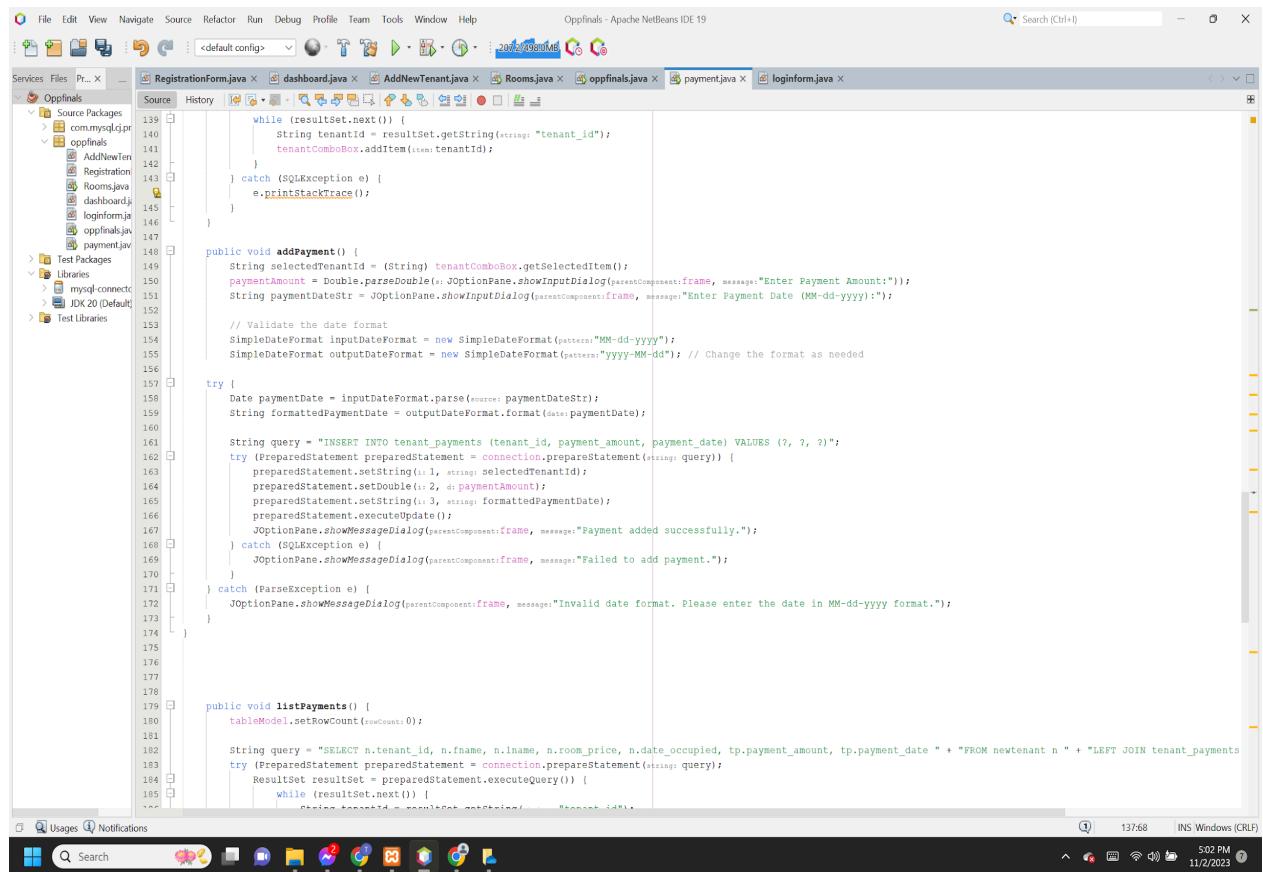


The screenshot shows the Apache NetBeans IDE 19 interface. The left sidebar displays the project structure under the 'Oppfinals' package, which contains several Java files: RegistrationForm.java, dashboard.java, AddNewTenant.java, Rooms.java, oppfinals.java, payment.java, and loginform.java. The main editor window shows the 'oppfinals.java' file, specifically the 'Oppfinals' class. The code implements various ActionListener interfaces for buttons like addButton, listButton, and backButton, as well as a tenantComboBox. It also includes methods for connecting to a MySQL database and executing SQL queries to load tenant IDs. The status bar at the bottom right indicates the system is running on Windows (CRLF) at 9:58 PM on 11/2/2023.

```
91     frame.setLocation(new Point (x: 350, y: 200));
92     frame.getContentPane().setBackground(Color.pink);
93     addButton.addActionListener(new ActionListener() {
94         @Override
95         public void actionPerformed(ActionEvent e) {
96             addPayment();
97         }
98     });
99
100    listButton.addActionListener(new ActionListener() {
101        @Override
102        public void actionPerformed(ActionEvent e) {
103            listPayments();
104        }
105    });
106
107    backButton.addActionListener(new ActionListener() {
108        @Override
109        public void actionPerformed(ActionEvent e) {
110            dashboard dash = new dashboard();
111        }
112    });
113
114    tenantComboBox.addActionListener(new ActionListener() {
115        @Override
116        public void actionPerformed(ActionEvent e) {
117            updateTenantInformation((String) tenantComboBox.getSelectedItem());
118        }
119    });
120
121    frame.setVisible(true);
122 }
123
124 private void connectToDatabase() {
125     try {
126         connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/registration", "root", "");
127     } catch (SQLException e) {
128         e.printStackTrace();
129         System.exit(status: 1);
130     }
131 }
132
133 private void loadTenantids() {
134     try {
135         String query = "SELECT tenant_id FROM newtenant";
136         PreparedStatement preparedStatement = connection.prepareStatement(string: query);
137         ResultSet resultSet = preparedStatement.executeQuery();
138     }
139 }
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



The screenshot shows the Apache NetBeans IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Window, Help.
- Toolbar:** Standard NetBeans toolbar with icons for file operations, search, and project navigation.
- Project Explorer:** Shows the project structure under "Oppfinals".
- Code Editor:** Displays Java code for "payment.java".
- Toolbars:** Standard Java development toolbars for code navigation and refactoring.
- Bottom Status Bar:** Shows system information: 13768, INS Windows (CRLF), 502 PM, 11/2/2023.

```
139     while (resultSet.next()) {
140         String tenantId = resultSet.getString("tenant_id");
141         tenantComboBox.addItem(item(tenantId));
142     }
143 } catch (SQLException e) {
144     e.printStackTrace();
145 }
146
147 public void addPayment() {
148     String selectedTenantId = (String) tenantComboBox.getSelectedItem();
149     paymentAmount = Double.parseDouble(JOptionPane.showInputDialog(parentComponent, frame, "Enter Payment Amount:"));
150     String paymentDateStr = JOptionPane.showInputDialog(parentComponent, frame, "Enter Payment Date (MM-dd-yyyy):");
151
152     // Validate the date format
153     SimpleDateFormat inputDateFormat = new SimpleDateFormat("MM-dd-yyyy");
154     SimpleDateFormat outputDateFormat = new SimpleDateFormat("yyyy-MM-dd"); // Change the format as needed
155
156     try {
157         Date paymentDate = inputDateFormat.parse(paymentDateStr);
158         String formattedPaymentDate = outputDateFormat.format(paymentDate);
159
160         String query = "INSERT INTO tenant_payments (tenant_id, payment_amount, payment_date) VALUES (?, ?, ?)";
161         try (PreparedStatement preparedStatement = connection.prepareStatement(query)) {
162             preparedStatement.setString(1, selectedTenantId);
163             preparedStatement.setDouble(2, paymentAmount);
164             preparedStatement.setString(3, formattedPaymentDate);
165             preparedStatement.executeUpdate();
166             JOptionPane.showMessageDialog(frame, "Payment added successfully.");
167         } catch (SQLException e) {
168             JOptionPane.showMessageDialog(frame, "Failed to add payment.");
169         }
170     } catch (ParseException e) {
171         JOptionPane.showMessageDialog(frame, "Invalid date format. Please enter the date in MM-dd-yyyy format.");
172     }
173 }
174
175
176
177
178 public void listPayments() {
179     tableView.setModel(tableViewModel);
180     tableViewModel.setRowCount(0);
181
182     String query = "SELECT n.tenant_id, n.fname, n.lname, n.room_price, n.date_occupied, tp.payment_amount, tp.payment_date " +
183     "FROM newtenant n " +
184     "LEFT JOIN tenant_payments tp ON n.tenant_id = tp.tenant_id";
185
186     try (PreparedStatement preparedStatement = connection.prepareStatement(query)) {
187         ResultSet resultSet = preparedStatement.executeQuery();
188         while (resultSet.next()) {
189             String tenantId = resultSet.getString("tenant_id");
190             ...
191         }
192     }
193 }
```

# PHINMA UNIVERSITY OF ILOILO

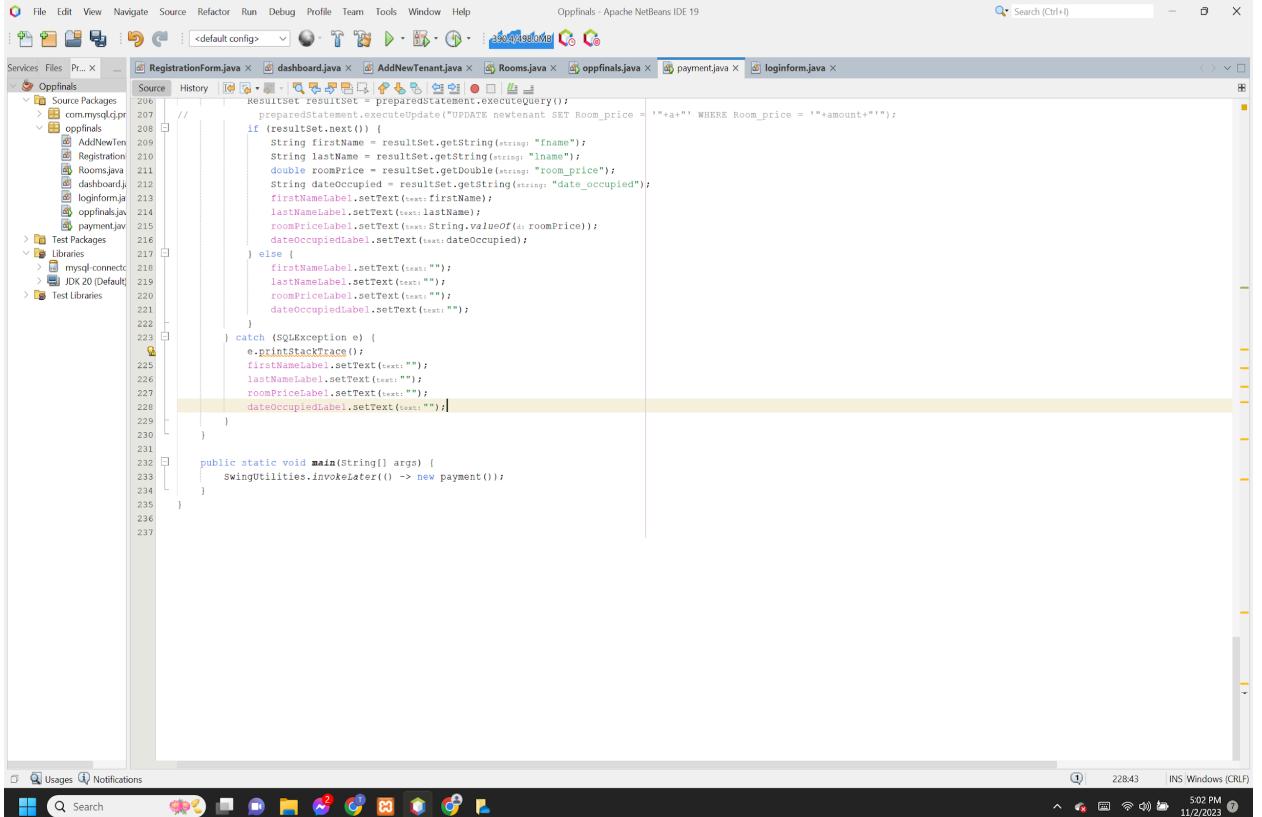
## College of Information Technology Education

The screenshot shows the Apache NetBeans IDE 19 interface. The title bar reads "Oppfinals - Apache NetBeans IDE 19". The left sidebar shows the project structure under "Source Packages" for the "Oppfinals" package, containing files like RegistrationForm.java, AddNewTenant.java, Rooms.java, loginform.java, oppfinals.java, and payment.java. The main editor window displays Java code for the "RegistrationForm.java" file. The code handles database operations using JDBC, including preparing statements, executing queries, and updating tenant information. The code uses try-catch blocks to handle SQL exceptions. The bottom status bar shows the time as 18:54 and date as 11/2/2023.

```
184     ResultSet resultSet = preparedStatement.executeQuery();
185     while (resultSet.next()) {
186         String tenantId = resultSet.getString("tenant_id");
187         String fname = resultSet.getString("fname");
188         String lastName = resultSet.getString("lname");
189         double roomPrice = resultSet.getDouble("room_price");
190         String dateOccupied = resultSet.getString("date_occupied");
191         String paymentDate = resultSet.getString("payment_date");
192         amount = resultSet.getDouble("payment_amount");
193         tableModel.addRow(new Object[]{tenantId, amount, paymentDate});
194     }
195     catch (SQLException e) {
196         e.printStackTrace();
197     }
198 }
199
200 private void updateTenantInformation(String tenantId) {
201     String query = "SELECT fname, lname, Room_price, date_occupied FROM newtenant WHERE tenant_id = ?";
202     // double a = amount - paymentAmount;
203     try (PreparedStatement preparedStatement = connection.prepareStatement(query)) {
204         preparedStatement.setString(1, tenantId);
205         ResultSet resultSet = preparedStatement.executeQuery();
206         preparedStatement.executeUpdate("UPDATE newtenant SET Room_price = "+a+" WHERE Room_price = "+amount+"");
207         if (resultSet.next()) {
208             String firstName = resultSet.getString("fname");
209             String lastName = resultSet.getString("lname");
210             double roomPrice = resultSet.getDouble("room_price");
211             String dateOccupied = resultSet.getString("date_occupied");
212             firstNameLabel.setText(firstName);
213             lastNameLabel.setText(lastName);
214             roomPriceLabel.setText(String.valueOf(roomPrice));
215             dateOccupiedLabel.setText(dateOccupied);
216         } else {
217             firstNameLabel.setText("");
218             lastNameLabel.setText("");
219             roomPriceLabel.setText("");
220             dateOccupiedLabel.setText("");
221         }
222     } catch (SQLException e) {
223         e.printStackTrace();
224         firstNameLabel.setText("");
225         lastNameLabel.setText("");
226         roomPriceLabel.setText("");
227         dateOccupiedLabel.setText("");
228     }
229 }
230 }
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



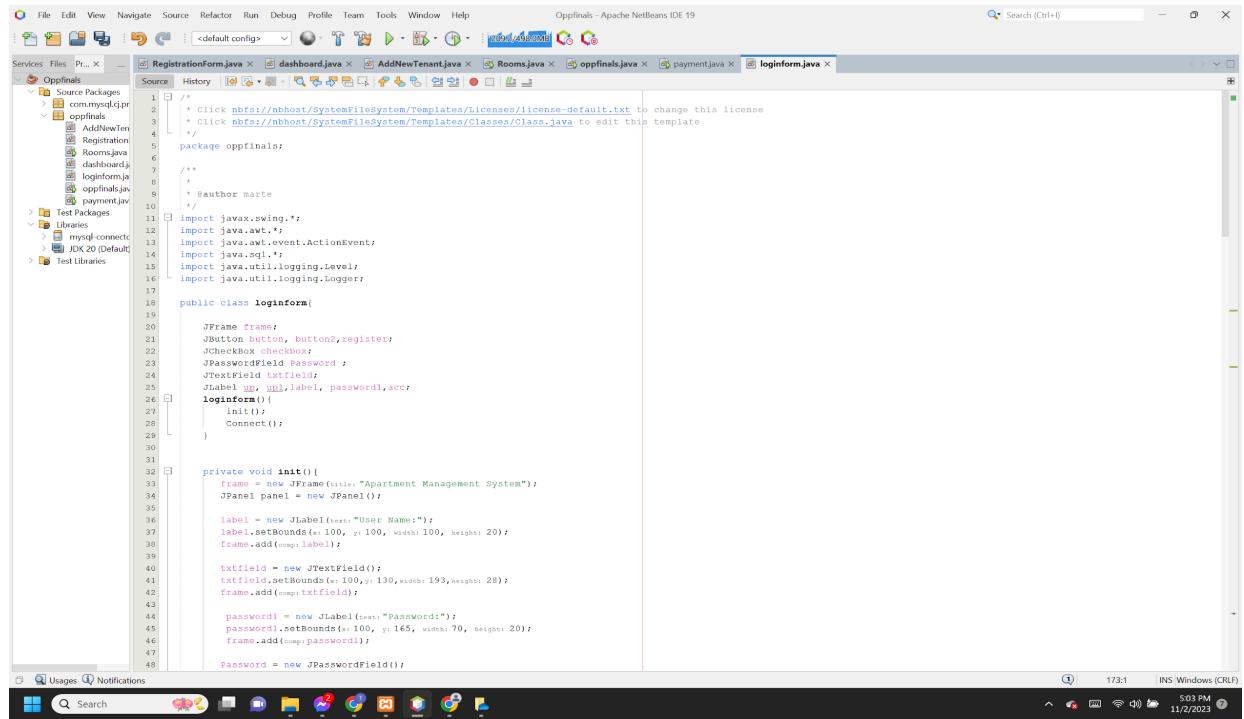
The screenshot shows the Apache NetBeans IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Search Bar:** Search (Ctrl+F).
- Project Tree:** Shows the project structure under "Oppfinals".
  - Source Packages: com.mysql.jdbc, oppfinals (containing AddNewTen, Registration, Rooms, dashboard, loginform, oppfinals, payment).
  - Test Packages: mysql-connector, JDX 20 (Default).
  - Libraries: MySQL connector.
  - Test Libraries: None.
- Code Editor:** The "payment.java" file is open, displaying Java code for a payment application. The code includes database queries for updating room prices and handling occupied dates.
- Status Bar:** Shows 22843, INS Windows (CRLF), and a timestamp of 5:02 PM on 11/2/2023.

```
206     // If the update was successful
207     if (resultSet.next()) {
208         String firstName = resultSet.getString("fname");
209         String lastName = resultSet.getString("lname");
210         double roomPrice = resultSet.getDouble("room_price");
211         String dateOccupied = resultSet.getString("date_occupied");
212         firstNameLabel.setText(firstName);
213         lastNameLabel.setText(lastName);
214         roomPriceLabel.setText(String.valueOf(roomPrice));
215         dateOccupiedLabel.setText(dateOccupied);
216     } else {
217         firstNameLabel.setText("");
218         lastNameLabel.setText("");
219         roomPriceLabel.setText("");
220         dateOccupiedLabel.setText("");
221     }
222     } catch (SQLException e) {
223         e.printStackTrace();
224         firstNameLabel.setText("");
225         lastNameLabel.setText("");
226         roomPriceLabel.setText("");
227         dateOccupiedLabel.setText("");
228     }
229 }
230
231 public static void main(String[] args) {
232     SwingUtilities.invokeLater(() -> new payment());
233 }
234 }
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



The screenshot shows the Apache NetBeans IDE interface with the file `loginform.java` open in the source editor. The code implements a login form for an apartment management system. It includes imports for Java Swing and AWT, and defines a `loginform` class with methods `init()` and `Connect()`. The `init()` method sets up the frame, labels, and text fields. The `Connect()` method is currently empty.

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package oppfinals;

/**
 *
 * @author marte
 */
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;

public class loginform {

    JFrame frame;
    JButton button, button2,register;
    JTextField txtname;
    JPasswordField Password ;
    JTextField txtfield;
    JLabel up, unl,label, password1,acc;
    loginform()
    {
        init();
        Connect();
    }
    public void init()
    {
        frame = new JFrame("Apartment Management System");
        JPanel panel = new JPanel();
        up = new JLabel(text:"User Name:");
        up.setBounds(x: 100, y: 100, width: 100, height: 20);
        frame.add(unl,label);

        txtfield = new JTextField();
        txtfield.setBounds(x: 100,y: 130,width: 193,height: 28);
        frame.add(unl,txtfield);

        password1 = new JLabel(text:"Password:");
        password1.setBounds(x: 100, y: 165, width: 70, height: 20);
        frame.add(unl,password1);

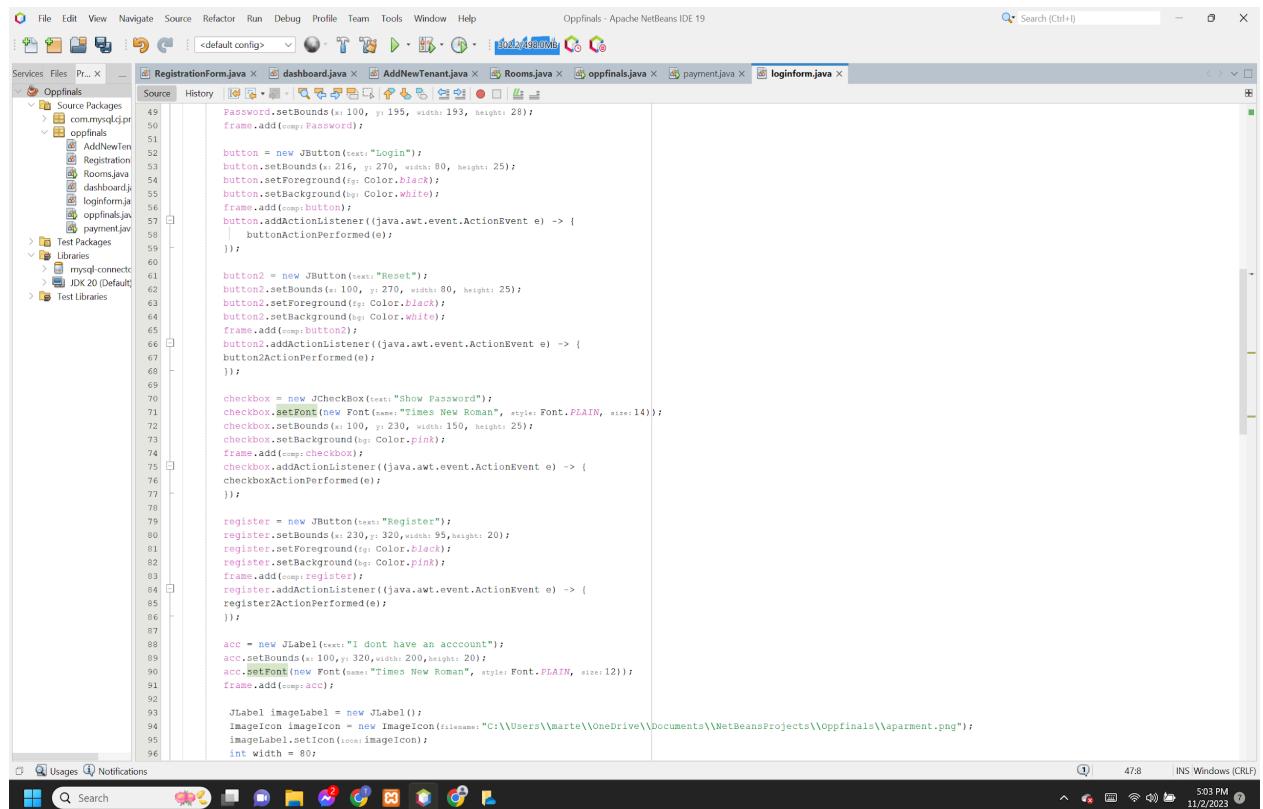
        Password = new JPasswordField();
    }

    private void init()
    {
        frame = new JFrame(title:"Apartment Management System");
        JPanel panel = new JPanel();
        up = new JLabel(text:"User Name:");
        up.setBounds(x: 100, y: 100, width: 100, height: 20);
        frame.add(unl,label);

        txtfield = new JTextField();
        txtfield.setBounds(x: 100,y: 130,width: 193,height: 28);
        frame.add(unl,txtfield);

        password1 = new JLabel(text:"Password:");
        password1.setBounds(x: 100, y: 165, width: 70, height: 20);
        frame.add(unl,password1);

        Password = new JPasswordField();
    }
}
```



The screenshot shows the Apache NetBeans IDE interface with the file `loginform.java` open in the source editor. The code adds more UI components to the login form, including a `button`, a `button2` for reset, a `checkbox` for password visibility, and a `register` button. The `button` and `button2` are set to black foreground and white background. The `checkbox` has a pink background. The `register` button is located at the bottom left. The `register` button's action listener is currently empty.

```
        button = new JButton(text:"Login");
        button.setBounds(x: 216, y: 270, width: 80, height: 25);
        button.setForeground(bg: Color.black);
        button.setBackground(bg: Color.white);
        frame.add(unl,button);
        button.addActionListener((java.awt.event.ActionEvent e) -> {
            button2ActionPerformed(e);
        });

        button2 = new JButton(text:"Reset");
        button2.setBounds(x: 100, y: 270, width: 80, height: 25);
        button2.setForeground(bg: Color.black);
        button2.setBackground(bg: Color.white);
        frame.add(unl,button2);
        button2.addActionListener((java.awt.event.ActionEvent e) -> {
            button2ActionPerformed(e);
        });

        checkbox = new JCheckBox(text:"Show Password");
        checkbox.setFont(new Font(name: "Times New Roman", style: Font.PLAIN, size:14));
        checkbox.setBounds(x: 100, y: 230, width: 150, height: 25);
        checkbox.setBackground(bg: Color.pink);
        frame.add(unl,checkbox);
        checkbox.addActionListener((java.awt.event.ActionEvent e) -> {
            checkboxActionPerformed(e);
        });

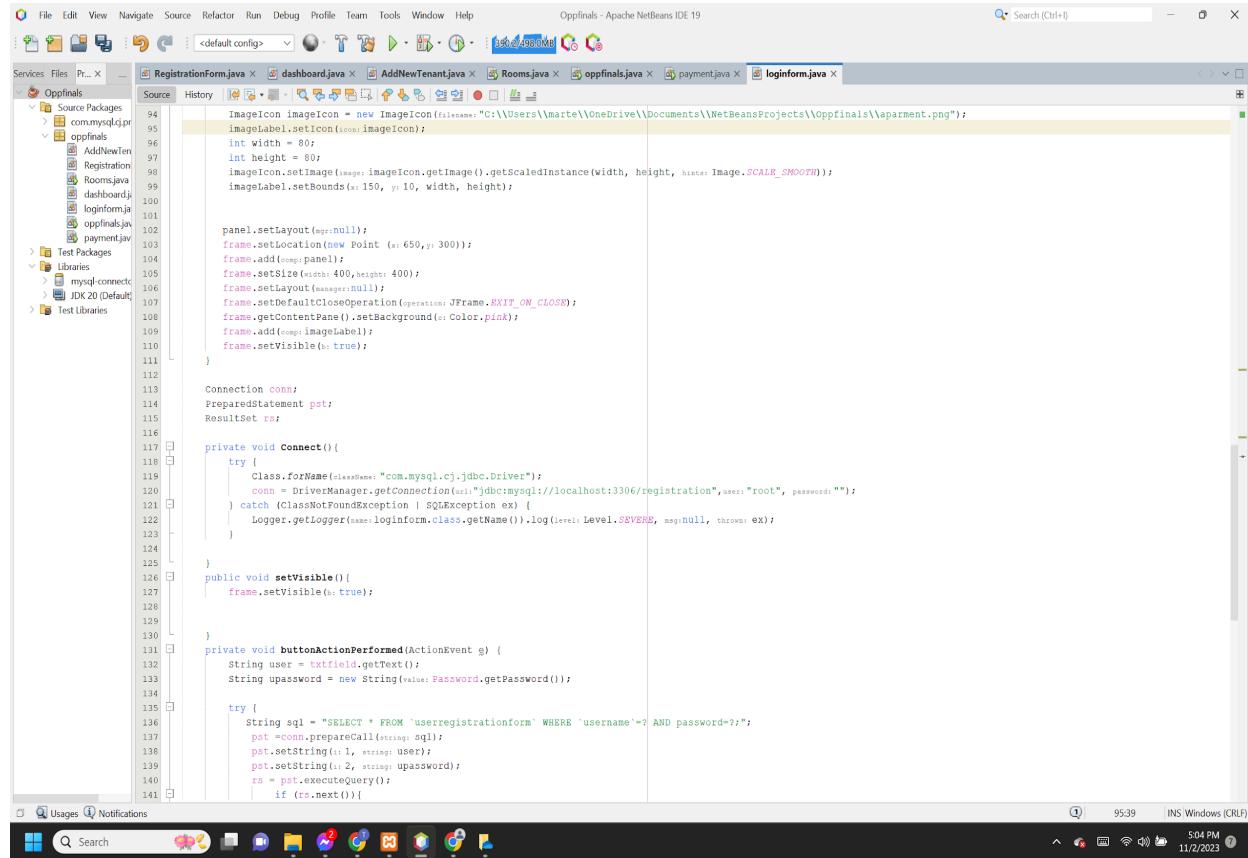
        register = new JButton(text:"Register");
        register.setBounds(x: 230, y: 320, width: 95,height: 20);
        register.setForeground(bg: Color.black);
        register.setBackground(bg: Color.pink);
        frame.add(unl,register);
        register.addActionListener((java.awt.event.ActionEvent e) -> {
            register2ActionPerformed(e);
        });

        acc = new JLabel(text:"I dont have an account");
        acc.setBounds(x: 100,y: 320, width: 200,height: 20);
        acc.setFont(new Font(name: "Times New Roman", style: Font.PLAIN, size: 12));
        frame.add(unl,acc);

        JLabel imagedlabel = new JLabel();
        ImageIcon imageIcon = new ImageIcon(filename: "C:\\\\Users\\\\marte\\\\OneDrive\\\\Documents\\\\NetBeansProjects\\\\Oppfinals\\\\aparment.png");
        imageLabel.setIcon(imageIcon);
        int width = 80;
```

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education

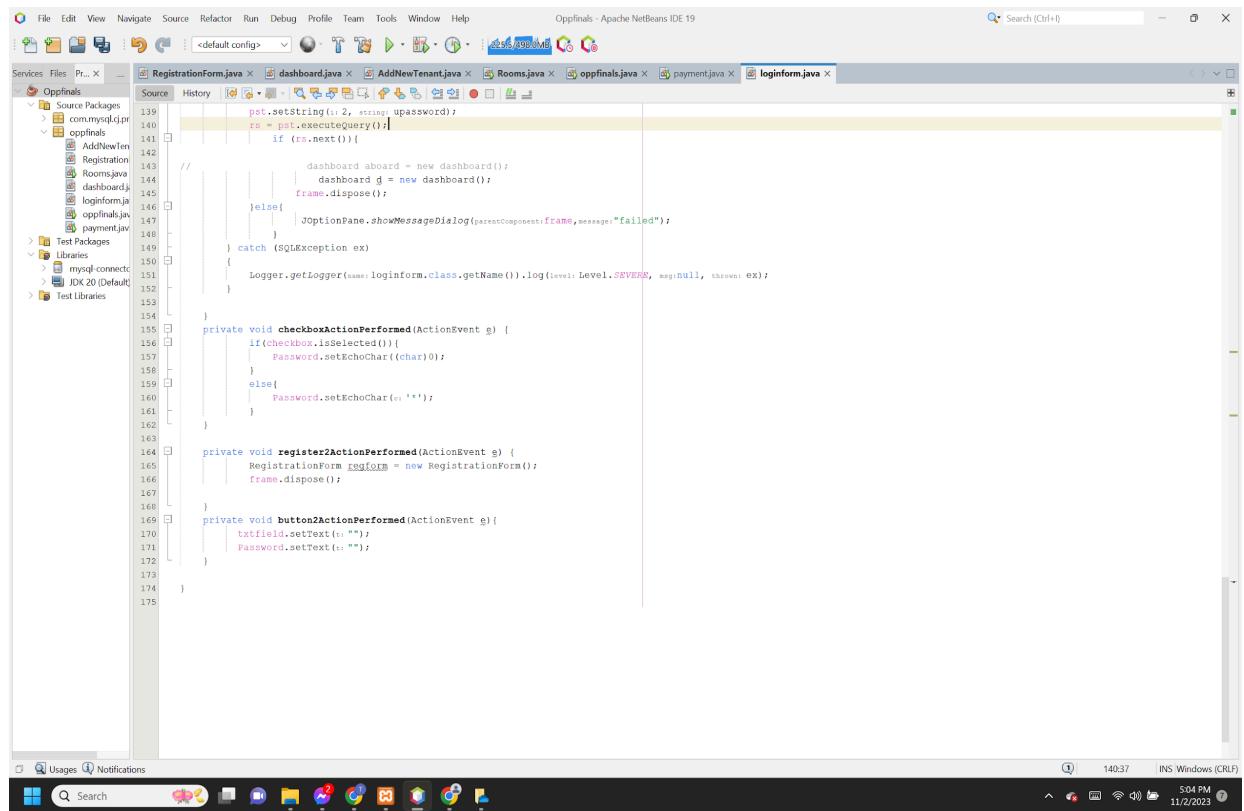


The screenshot shows the Apache NetBeans IDE interface with the following details:

- Project Structure:** The left sidebar shows the project structure under "Oppfinals". It includes a "Source Packages" node containing "oppfinals" (with files like AddNewTen, Registration, Room.java, dashboard.java, loginform.java, oppfinals.java, and payment.java) and "Test Packages" (with mysql-connector and JDK 20 (Default)).
- Code Editor:** The main window displays the "loginform.java" file. The code is a Java Swing application for user login. It includes imports for java.awt, javax.swing, and javax.imageio.ImageIcon. The code sets up a frame, adds an image icon to the title bar, and handles button action events to execute SQL queries against a MySQL database.
- Toolbars and Status Bar:** The top toolbar has standard options like File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and a search bar. The status bar at the bottom right shows system information: 95.39, 5:04 PM, 11/2/2023, and INS Windows (CRLF).

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education



The screenshot shows the Apache NetBeans IDE 19 interface. The title bar reads "Oppfinals - Apache NetBeans IDE 19". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has icons for New, Open, Save, Cut, Copy, Paste, Find, and Run. The left sidebar shows the project structure under "Source Packages": Oppfinals (containing com.mysql.jdbc.Driver, oppfinals, AddNewTenent, Registration, Rooms.java, dashboard.java, loginform.java, oppfinals.java, payment.java) and Test Packages (empty). The right sidebar shows the current file "RegistrationForm.java" with its code:

```
139     prt.setString(1, email);
140     rs = ps.executeQuery();
141     if (rs.next()){
142         // ...
143         dashboard aboard = new dashboard();
144         dashboard d = new dashboard();
145         frame.dispose();
146     } else{
147         JOptionPane.showMessageDialog(parentComponent, frame, "failed");
148     }
149 } catch (SQLException ex){
150 {
151     Logger.getLogger(loginform.class.getName()).log(Level.SEVERE, null, ex);
152 }
153 }
154 private void checkboxActionPerformed(ActionEvent e) {
155     if (checkbox.isSelected()){
156         Password.setEchoChar((char)0);
157     } else{
158         Password.setEchoChar((char)'*');
159     }
160 }
161
162 private void register2ActionPerformed(ActionEvent e) {
163     RegistrationForm regform = new RegistrationForm();
164     frame.dispose();
165 }
166
167 private void buttonActionPerformed(ActionEvent e){
168     txtId.setText("");
169     Password.setText("");
170 }
```

Data Flow Diagram:

# PHINMA UNIVERSITY OF ILOILO

## College of Information Technology Education

