

KK14203

OBJECT-ORIENTED PROGRAMMING

HOTEL MANAGEMENT SYSTEM

DAPHNE YONG ZHI XUAN

BI19110065

SECTION 1

LECTURER:

MOHD SHAMRIE SAININ

FKI

**CONTENTS**

[**INTRODUCTION** 3](#_Toc47204558)

[**JAVA CODE** 4](#_Toc47204559)

[**OBJECT ORIENTED CONCEPT IMPLEMENTATION** 16](#_Toc47204560)

[**READ AND WRITE IMPLEMENTATION** 17](#_Toc47204561)

[**USER MANUAL** 19](#_Toc47204562)

# **INTRODUCTION**

This report is written to show a simple hotel management system with Graphical User Interface (GUI) implemented. In this report, the java code of the system is written. For further information, a short and simple explanation of object oriented concept , read and write implementation in the java code are shown. A user manual is provided for explanation on how to use the system.

The hotel in the java code is given a name called ”Fortune Hotel”. The user needs to login before able to access the main menu. The username of the code is ”user” and the password is ”1234”. In the main menu, user can choose to enter a new customer data or view the register record of the hotel.

# **JAVA CODE**

//DAPHNE YONG ZHI XUAN  
//BI19110065  
//KK14203 INDIVIDUAL PROJECT  
//HOTEL MANAGEMENT SYSTEM  
import java.util.Scanner;  
import java.awt.\*;  
import java.awt.event.\*;  
import javax.swing.\*;  
import javax.swing.event.\*;  
import javax.swing.BorderFactory;  
import javax.swing.border.Border;  
import java.io.File;  
import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.PrintWriter;  
import java.io.BufferedWriter;  
import java.io.BufferedReader;  
import java.io.IOException;  
  
class login extends JPanel {  
 private JLabel line1;  
 private JLabel line2;  
 private JLabel line3;  
 JPasswordField password;  
 JTextField username;  
 private JButton b\_login;  
  
 public login() {  
 line1 = new JLabel ("Login");  
 line2 = new JLabel ("Username");  
 line3 = new JLabel ("Password");  
 password = new JPasswordField (20);  
 username = new JTextField (20);  
 b\_login = new JButton ("Login");  
  
 setPreferredSize (new Dimension (624, 366));  
 setLayout (null);  
  
 add (line1);  
 add (line2);  
 add (line3);  
 add (password);  
 add (username);  
 add (b\_login);  
  
 line1.setBounds (260, 40, 100, 25);  
 line2.setBounds (130, 85, 100, 25);  
 line3.setBounds (130, 125, 100, 25);  
 password.setBounds (230, 125, 200, 25);  
 username.setBounds (230, 85, 200, 25);  
 b\_login.setBounds (235, 175, 100, 25);  
   
 b\_login.addActionListener(new ActionListener(){  
 public void actionPerformed(ActionEvent e){  
 if(username.getText().equals("user")&&password.getText().equals("1234")){  
 JOptionPane.showMessageDialog(null, "Login successfully");   
 JFrame f2 = new JFrame("Main menu");  
 f2.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 menu m=new menu();  
 f2.add(m,BorderLayout.CENTER);  
 f2.setSize(400,300);  
 f2.setVisible (true);  
 setVisible(false);  
 }  
 else  
 JOptionPane.showMessageDialog(null, "Username and password incorrect","ERROR",JOptionPane.ERROR\_MESSAGE);  
 }  
 });   
 }  
}  
  
class menu extends JPanel {  
 private JLabel line1;  
 private JButton b\_new;  
 private JButton b\_view;  
   
 String output="";  
 String filePath="data.txt";  
  
 public menu() {  
 line1 = new JLabel ("Menu");  
 b\_new = new JButton ("Create new data");  
 b\_view = new JButton ("Check register record");  
  
 setPreferredSize (new Dimension (336, 197));  
 setLayout (null);  
  
 add (line1);  
 add (b\_new);  
 add (b\_view);  
  
 line1.setBounds (140, 20, 100, 25);  
 b\_new.setBounds (75, 60, 200, 25);  
 b\_view.setBounds (75, 105, 200, 25);  
   
 b\_new.addActionListener(new ActionListener(){  
 public void actionPerformed(ActionEvent e){  
 JFrame f3 = new JFrame("Create new data");  
 f3.setDefaultCloseOperation(JFrame.HIDE\_ON\_CLOSE);  
 create c=new create();  
 f3.add(c,BorderLayout.CENTER);  
 f3.setSize(700,500);  
 f3.setVisible (true);  
 }  
 });  
   
 b\_view.addActionListener(new ActionListener(){  
 public void actionPerformed(ActionEvent e){  
 JFrame f4 = new JFrame("Record");  
 f4.setDefaultCloseOperation(JFrame.HIDE\_ON\_CLOSE);  
 view v = new view();  
 f4.add(v, BorderLayout.CENTER);  
 f4.setSize(700,500);  
 f4.setVisible(true);  
 }  
 });  
 }  
}  
  
class create extends JPanel {  
 private JLabel line1;  
 private JTextField name;  
 private JLabel line2;  
 private JTextField tel;  
 private JLabel line3;  
 private JTextField day;  
 private JLabel line4;  
 private JLabel line5;  
 private JButton b\_submit;  
 private JRadioButton r1;  
 private JRadioButton r2;  
 private JRadioButton r3;  
 JLabel output\_field;  
 Border border = BorderFactory.createLineBorder(Color.BLACK, 1);  
 String output="";  
 int room=0;  
 String roomtype="";  
  
 public create() {  
 line1 = new JLabel ("Customer Name");  
 name = new JTextField (30);  
 line2 = new JLabel ("Telephone Number");  
 tel = new JTextField (20);  
 line3 = new JLabel ("Days Stayed");  
 day = new JTextField (5);  
 line4 = new JLabel ("Booking for Hotel");  
 b\_submit = new JButton ("Submit");  
 line5 = new JLabel ("Type of room");  
 r1 = new JRadioButton ("Single (RM170)");  
 r2 = new JRadioButton ("Double (RM220)");  
 r3 = new JRadioButton ("Family(RM300)");  
 output\_field = new JLabel("");  
 output\_field.setPreferredSize(new Dimension(15, 20));  
   
 setPreferredSize (new Dimension (667, 609));  
 setLayout (null);  
  
 add (line1);  
 add (name);  
 add (line2);  
 add (tel);  
 add (line3);  
 add (day);  
 add (output\_field);  
 add (line4);  
 add (b\_submit);  
 add (line5);  
 add (r1);  
 add (r2);  
 add (r3);  
  
 line1.setBounds (105, 65, 100, 25);  
 name.setBounds (230, 65, 200, 25);  
 line2.setBounds (105, 95, 111, 25);  
 tel.setBounds (230, 95, 200, 25);  
 line3.setBounds (105, 125, 100, 25);  
 day.setBounds (230, 125, 200, 25);  
 output\_field.setBounds (35, 210, 600, 270);  
 line4.setBounds (250, 10, 256, 25);  
 b\_submit.setBounds (245, 180, 100, 25);  
 line5.setBounds (105, 150, 100, 25);  
 r1.setBounds (230, 155, 120, 25);  
 r2.setBounds (350, 155, 120, 25);  
 r3.setBounds (470, 155, 120, 25);  
 output\_field.setVerticalAlignment(JLabel.TOP);  
 output\_field.setBorder(border);  
  
 ButtonGroup bg = new ButtonGroup();  
 bg.add(r1);  
 bg.add(r2);  
 bg.add(r3);  
   
 b\_submit.addActionListener(new ActionListener(){  
 public void actionPerformed(ActionEvent e){  
 if (r1.isSelected()){  
 roomtype="Single room";  
 room=1;  
 }  
 else if (r2.isSelected()){  
 roomtype="Double room";  
 room=2;  
 }  
 else if (r3.isSelected()){  
 roomtype="Family room";  
 room=3;  
 }  
 else  
 room=0;  
 if(printOutput()){  
 writeInput();   
 name.setText("");  
 tel.setText("");  
 day.setText("");  
 bg.clearSelection();  
 }  
 }  
 });   
 }  
   
 public boolean printOutput(){  
 output = "<html>";  
 output += "Thank you for choosing Fortune Hotel<br><br>";   
 output += "Customer Name: " + name.getText() + "<br>";  
 if(name.getText().equals("")){  
 output\_field.setText("Customer name is not entered!");  
 return false;  
 }  
 output += "Telephone number: " + tel.getText() + "<br>";  
 if(tel.getText().equals("")){  
 output\_field.setText("Telephone number is not entered!");  
 return false;  
 }  
 if(isInteger(day.getText()))  
 output += "Day(s) Stayed: " + day.getText() + "<br>";  
 else  
 output\_field.setText("Days stayed needs to be number!");  
 if(day.getText().equals("")){  
 output\_field.setText("Days stayed is not selected!");  
 return false;  
 }  
 output += "Room Type: " + roomtype + "<br>";  
 if(room==0){  
 output\_field.setText("Room type is not selected!");  
 return false;  
 }  
 output += "Price: RM" + calculateprice(day.getText(),room) +"<br>";  
 output += "</html>";   
 JOptionPane.showMessageDialog(null, "The data is saved successfully.", "INFORMATION",JOptionPane.INFORMATION\_MESSAGE);   
 output\_field.setText(output);  
 return true;  
 }  
   
 private boolean isInteger(String input) {  
 try {  
 Integer.parseInt( input );  
 return true;  
 }catch( Exception e ) {  
 return false;  
 }  
 }  
   
 private int calculateprice(String day, int room){  
 int daynum = Integer.parseInt(day);  
 int price;  
 if(room==1)  
 price = daynum\*170;  
 else if(room==2)  
 price = daynum\*220;  
 else  
 price = daynum\*300;  
 return price;  
 }  
   
 public void writeInput(){  
 BufferedReader reader;  
 BufferedWriter br = null;  
 String data="";  
 try {  
 reader = new BufferedReader(new FileReader("data.txt"));  
 String line = reader.readLine();  
 while (line != null) {  
 data += line+"\n";  
 line = reader.readLine();  
 }  
 reader.close();  
 } catch (IOException io) {  
 output\_field.setText(io.toString());  
 }  
  
 String input = name.getText() + ", " + tel.getText() + ", " + day.getText() + ","+roomtype+", RM" + calculateprice(day.getText(), room);  
 try {  
 br = new BufferedWriter(new FileWriter("data.txt"));  
 br.write(data+input);  
 } catch (IOException e) {   
 output\_field.setText(e.toString());  
 } finally {  
 try {  
 br.close();  
 } catch (IOException e) {  
 output\_field.setText(e.toString());  
 }  
 }  
 }  
}  
  
class view extends JPanel{  
 private JLabel line1;  
 private JLabel sale;  
 JScrollPane jsp;  
 Border border = BorderFactory.createLineBorder(Color.BLACK, 1);  
  
 public view(){  
 line1 = new JLabel("Fortune Hotel Record");  
 sale = new JLabel("");  
 sale.setPreferredSize(new Dimension(667, 366));  
 sale.setBorder(border);  
 sale.setVerticalAlignment(JLabel.TOP);  
 jsp = new JScrollPane(sale);  
 String output="<html>";  
 add (line1);  
 add (sale);  
 add (jsp);  
   
 line1.setBounds(210, 10, 200, 25);  
 sale.setBounds(45, 50, 500, 205);  
 BufferedReader reader;  
 try {  
 reader = new BufferedReader(new FileReader("data.txt"));  
 String line = reader.readLine();  
 while (line != null) {  
 output += line+"<br>";  
 line = reader.readLine();  
 }  
 reader.close();  
 } catch (IOException io) {  
 JOptionPane.showMessageDialog(null,io.toString());  
 }  
 sale.setText(output);  
 }  
}  
  
public class HotelManagement{   
 public static void main(String[]args){  
 JFrame f1 = new JFrame("Fortune Hotel Login");  
 f1.setDefaultCloseOperation(JFrame.HIDE\_ON\_CLOSE);  
 login l=new login();  
 f1.add(l, BorderLayout.CENTER);  
 f1.setSize(550,350);  
 f1.setVisible(true);  
 }  
}

## **OBJECT ORIENTED CONCEPT IMPLEMENTATION**

The first object oriented concept implemented is encapsulation. Encapsulation is achieved when each object keeps its state private, inside a class. For example, in the java code, JButton, Radio Buttons and most of the JLabel are stated as private. The user cannot change any data in the private state method or data. The calculate price method is mentioned as private state causing no other class can change the content in the calculate price method. The user can only use the method to get the price but cannot change how to calculate the price of the hotel room.

The next object-oriented concept implemented is abstraction. Abstraction is the natural extension of encapsulation. The user just needs to press the button but the code runs and performs the work. For example, by pressing the ‘Submit’ button in the ‘Create new data’ page, the data entered to each text field is processed and provide a receipt while saving the data to a text file known as ’data.txt’.

Moreover, inheritance is shown in the java code. To inherit a class, the keyword ‘extends’ is used. The classes such as login, menu and create inherit the JPanel class which enable them to form a JPanel. They inherits the state and behaviors of the JPanel class. The login, menu and create classes act as child class while the JPanel class acts as the parent class.

The next object-oriented concept is polymorphism. The classes that inherit from the JPanel class keeps their own method. The login and menu class share the same parent class but they do not use the same method. Both classes have their own code and different from the others.

Other than that, class is implemented in the java code. There are many classes can be found in the java code. Five main classes are prepared for the code which is the Hotel Management, login, menu, create and view. Small classes such as print and write are also used in the code.

Furthermore, interface concept is implemented. The Graphical User Interface (GUI) is implemented.

Finally, method can be found in the java code. The calculate price method receives two data which is the number of days and the type of room. Then, the method provides an outcome of the price of the customer needs to pay. The method can be reused again and again for different data provided.

# **READ AND WRITE IMPLEMENTATION**

The code for reading the data in the text file is:

BufferedReader reader;  
try {  
 reader = new BufferedReader(new FileReader("data.txt"));  
 String line = reader.readLine();  
 while (line != null) {  
 output += line+"<br>";  
 line = reader.readLine();  
 }  
 reader.close();  
 } catch (IOException io) {  
 JOptionPane.showMessageDialog(null,io.toString());  
 }

sale.setText(output);

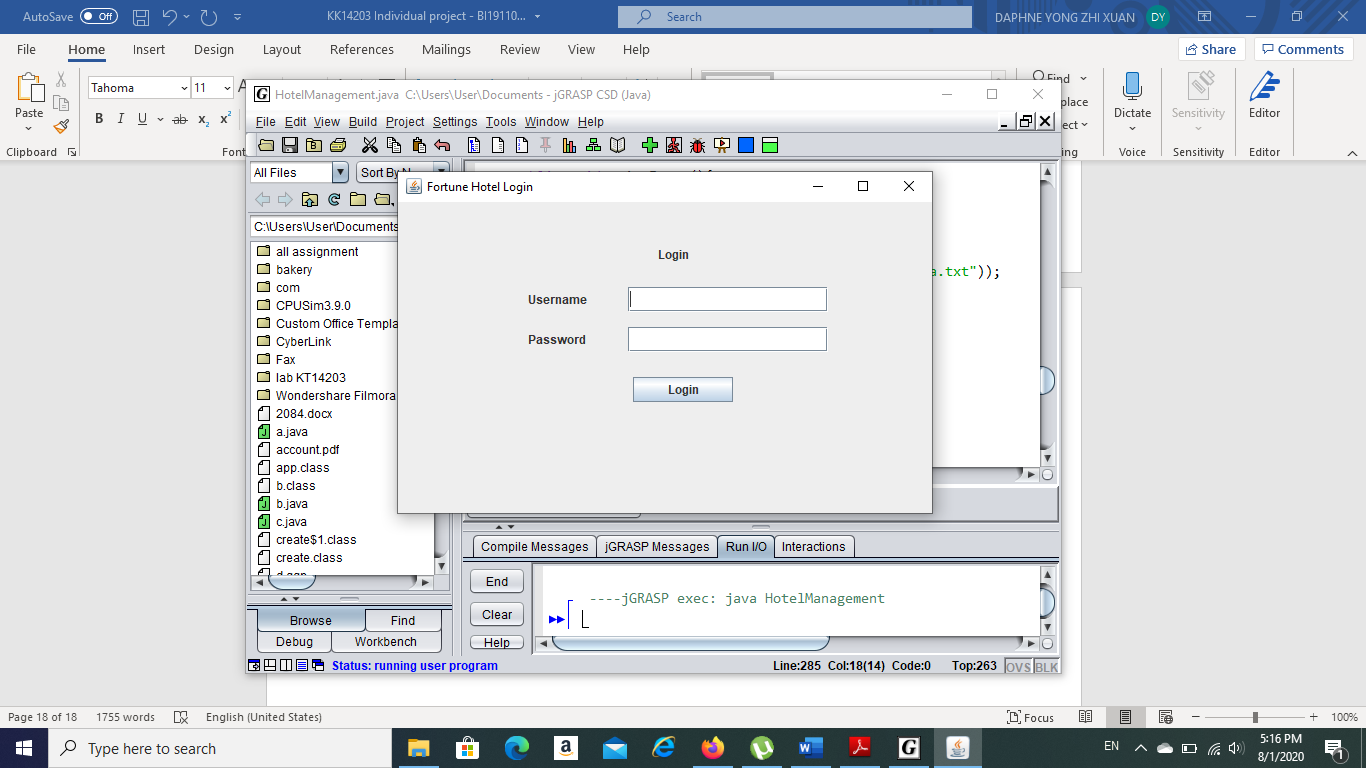
The buffered reader is called to read the data in the file. If the file is not found, an IOException message is pop-up. The readLine is used to read a line in the text file. The data read is then added to a variable called ‘output’. The readLine is repeated until no data can be read.

The code for writing the data to the text file is:

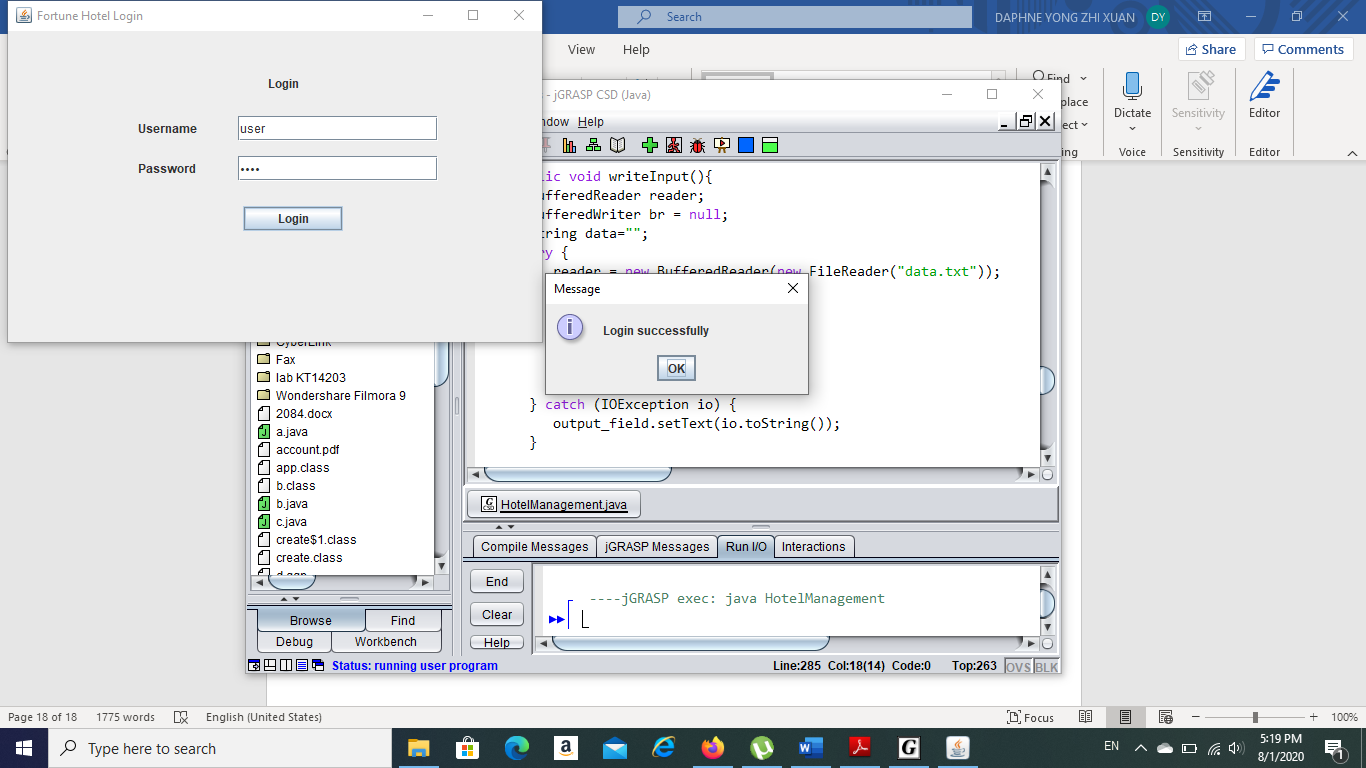
public void writeInput(){  
 BufferedReader reader;  
 BufferedWriter br = null;  
 String data="";  
 try {  
 reader = new BufferedReader(new FileReader("data.txt"));  
 String line = reader.readLine();  
 while (line != null) {  
 data += line+"\n";  
 line = reader.readLine();  
 }  
 reader.close();  
 } catch (IOException io) {  
 output\_field.setText(io.toString());  
 }  
  
 String input = name.getText() + ", " + tel.getText() + ", " + day.getText() + ", RM" + calculateprice(day.getText(), room);  
 try {  
 br = new BufferedWriter(new FileWriter("data.txt"));  
 br.write(data+input);  
 } catch (IOException e) {   
 output\_field.setText(e.toString());  
 } finally {  
 try {  
 br.close();  
 } catch (IOException e) {  
 output\_field.setText(e.toString());  
 }  
 }  
 }

A class called writeInput is used to write the data to the text file. First, the data is read from the text file and the ‘Create new file’ interface. Then, the data is combined and written into the text file. The data is read from the text file before writing to avoid the exist data is erased by the second and later inputs.

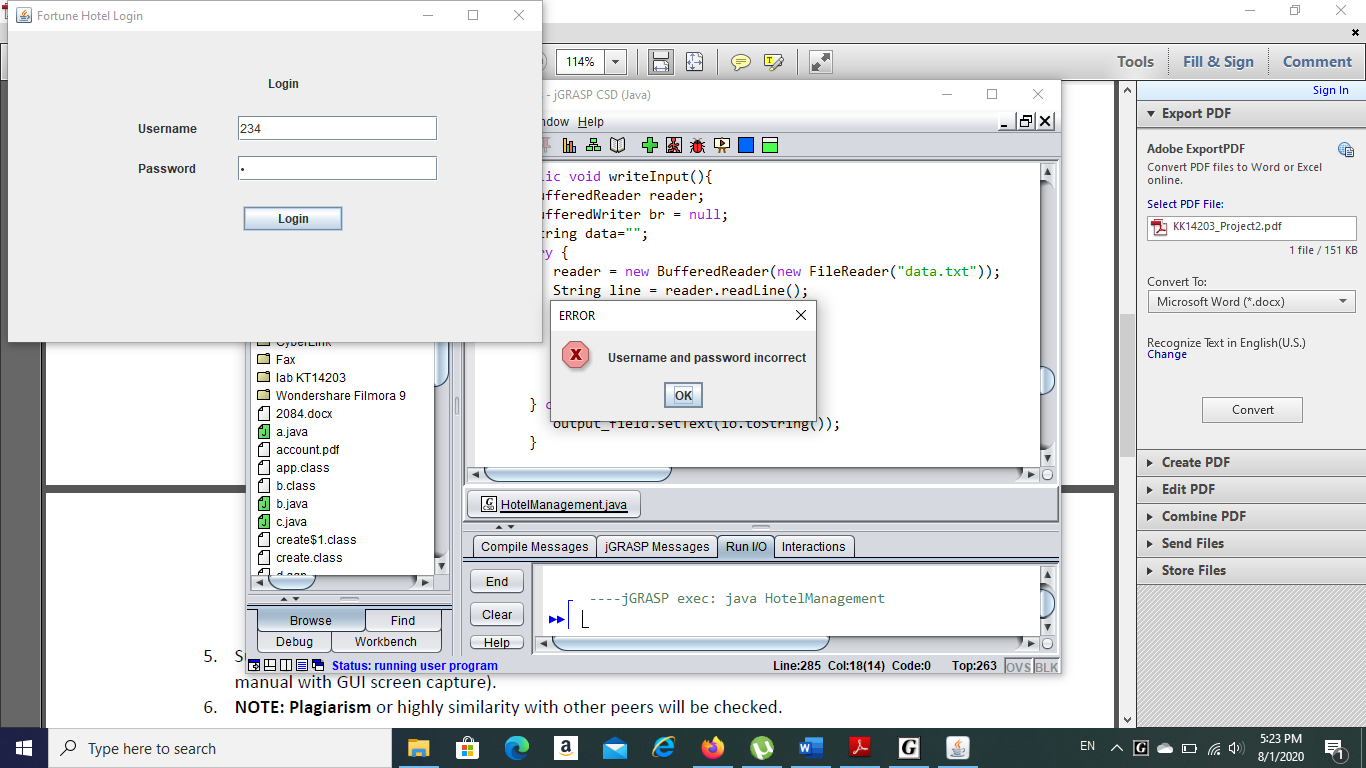
# **USER MANUAL**



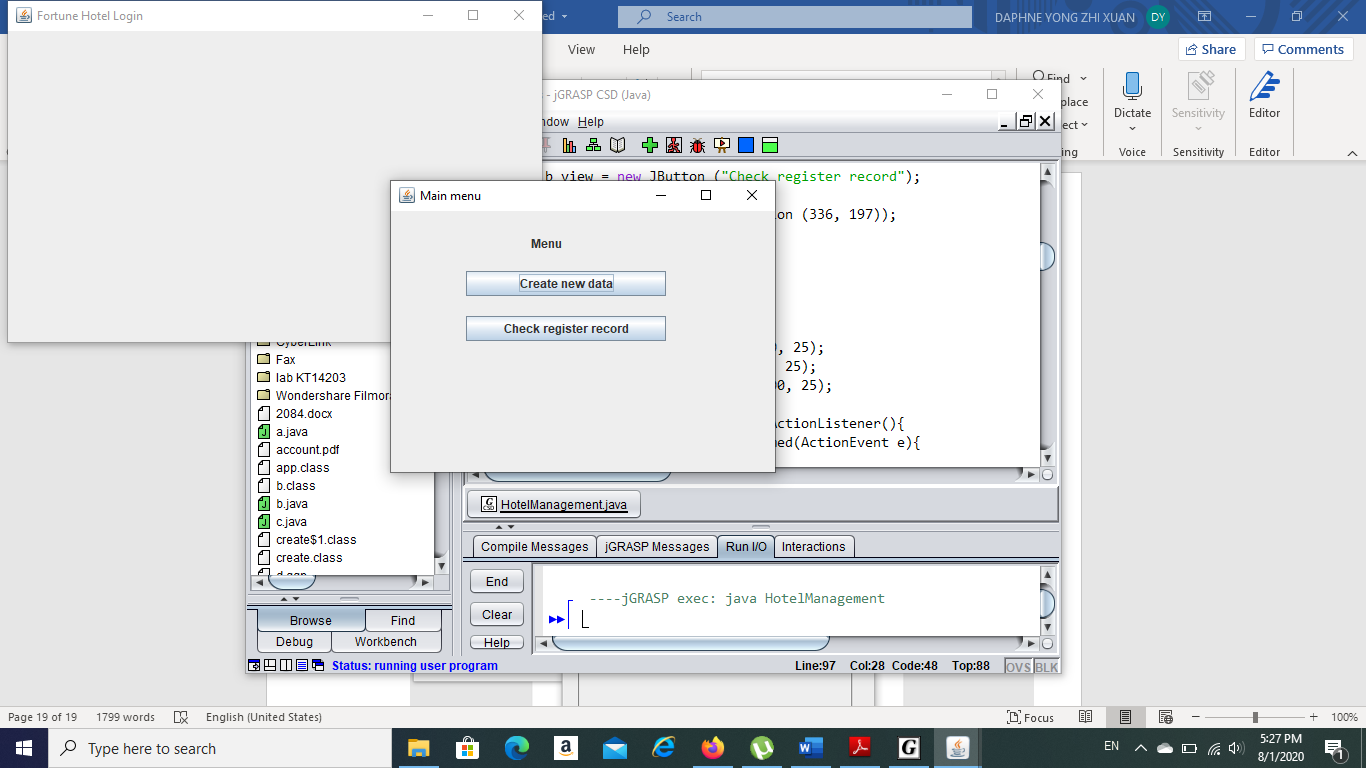
1. Enter the Username ‘user’ and Password ‘1234’.
2. Press the ‘Login’ button.
3. If the Username and Password are correct, a dialog box is open as below.

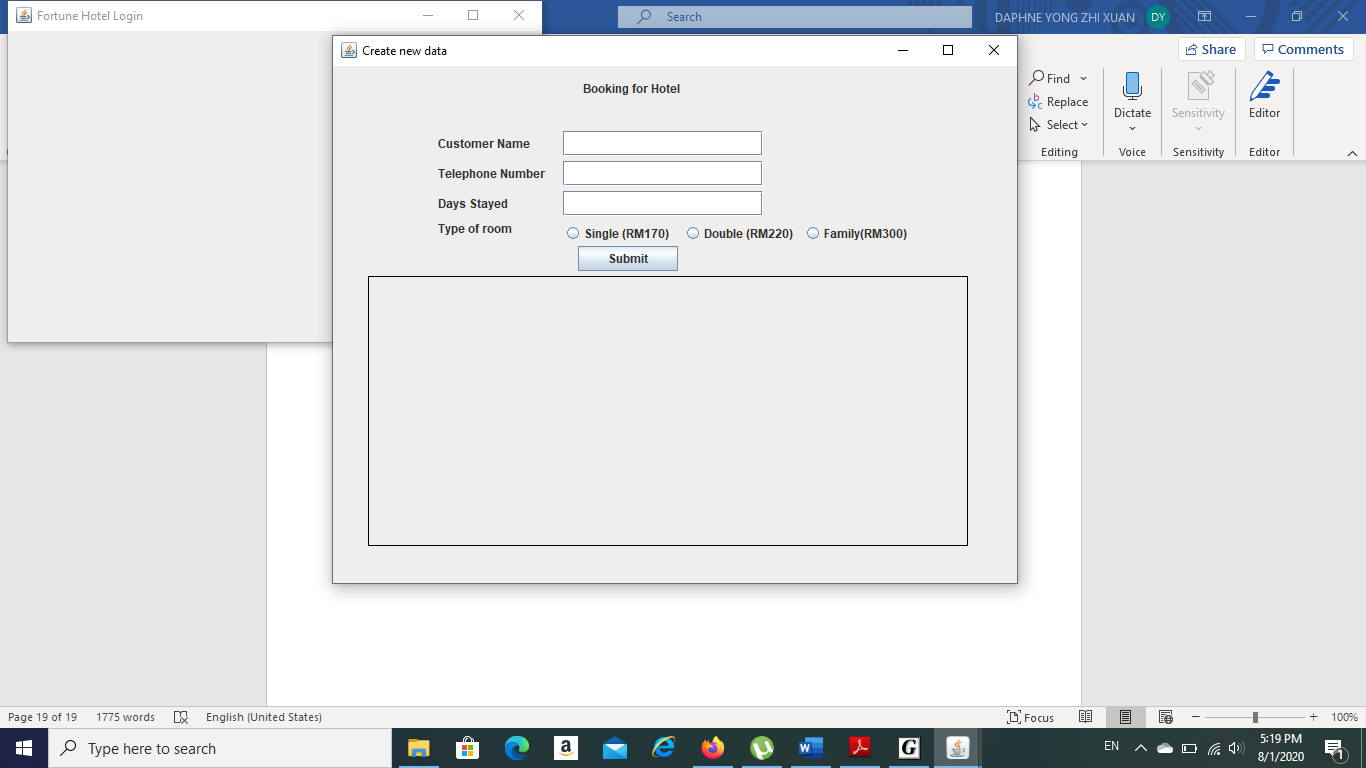


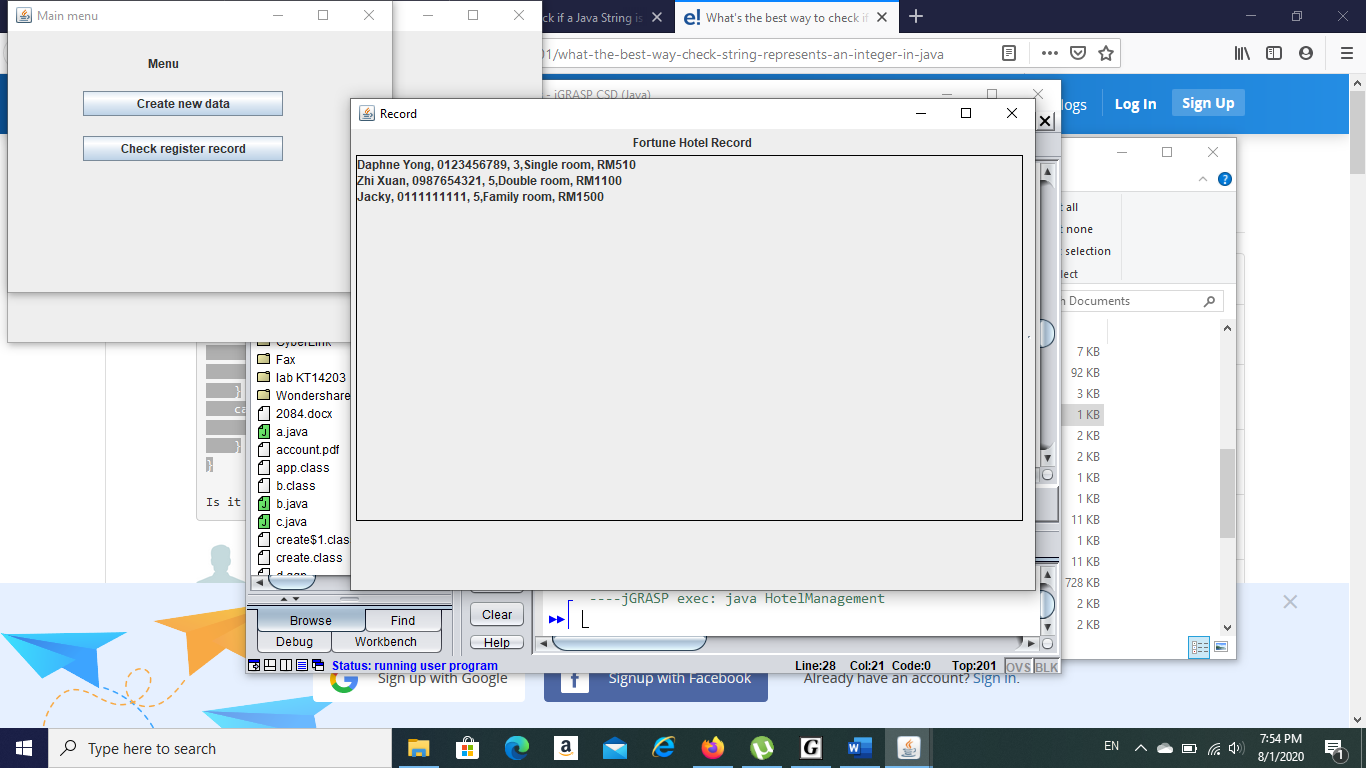
1. If the Username and Password are incorrect, an error dialog is open as below.



1. Main menu pops up after login successfully.



1. Click on the ‘Create new data’ to key-in the customer details.
2. Key-in the customer name, telephone number, days stayed and type of room. Click on the ‘Submit’ button to submit the data. 
3. Close the window after finish key-in the data.
4. Click on the ‘Check register record’ in the main menu window to view the data in the form of ‘customer name’, ‘telephone number’, ‘days stayed’, ‘Type of rooms’ and ‘Paid amount’ .



ATTENTION

1. The java code MUST save with the name ‘HotelManagement.java’.
2. The text file MUST save with the name ‘data.txt’.
3. The application CLOSE when exit from the main menu window.
4. The Username is ‘user’ and Password is ‘1234’.