UNIVERSITI TEKNOLOGI MARA

Fakulti Sains Komputer Dan Matematik

CSC435 – OBJECT-ORIENTED PROGRAMMING

Mini Project Report

Lecturer:

En. Nizam Bin Othman

Title: UiTM Perlis Sport Centre

Group Name	Computer Scientist
Team Member	Muhammad Zainul Muttaqin Mohd Zainudin (2020828578) - Leader Muhammad Akmal Ahmad Nazri (2020819092) Wan Amirul Afiq Wan Alias (2020627832)
Group	RCS2514A

Date Submitted: 6/2/2022

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

Our project is about a system where students can access to sport management through the internet easily without needing to go to the sports centre by walking to it, which consumes energy and needs to queue up when there are many people. So we came up with making UiTM Perlis Sport Centre where students can make court reservations, trainer reservation and sports equipment renting easily. At first opening this system, students need to sign up to UiTM Perlis Sport Centre by entering their first name, last name, student ID and password then after that they need to sign in to the system using the student ID and password they just entered before. Then the main menu will appear and shows some options; court reservation, rental sport equipment, trainer reservation and print receipt. For the first option which is court reservation, users will need to choose which sport will be played. Six sports have been set and users need to input how many hours of playing. Secondly is rental sport equipment, users are given the opportunity to rent basic necessities for playing any game for a reasonable price. Various equipment can be used for six types of sport available. For each item users can demand any quantity they want. Prices are based on each item whether they come in pairs or pieces. Next is trainer reservation, where users can choose their preferred trainer available in particular sports on the system. They also can request how many hours they want to be trained by the chosen trainer. Lastly is the print receipt, bills of those reservations and rentals will be calculated into one receipt and print it out to inform the user, total bill to be paid.

OBJECTIVE

Our project is for the purpose of making sport a seamless activity with less hustle by going to the sport facility, queueing up and waiting for just a simple matter such as to rent equipment and make court or coach reservations. The C++ program we developed could cut the queues due to the fact that the application could save their time so that they could have a longer time in the fields. Other than that, the sport reservation made by the student is systematic where the student just needs to access through the internet and click at UiTM Perlis Online Sport Centre for booking any court, sport equipment and trainer. Plus, the system is user friendly because it is easier to use. Thus could help create a healthier environment in UiTM Perlis.

SYSTEM SCOPE

In UiTM, students will need to go to the UiTM Sport Centre for any booking for sport facilities which will take time for a student. In this project we would like to create a program that may help the student. Students just need to login into UiTM Sport Centre portal for a booking for court, tools and trainer. Students will be provided a price for each facility. Students may also know any sport facilities being prepared in UiTM Sport Centre by login into the portal. Any problem can be directly said in the portal and will be solved as soon as possible by the UiTM Sport Centre.

STUDENT TASK

In our system, we emphasise the application of GUI in developing this project. As an addition we added on with application of file handling and also polymorphism. Our project is divided into 2 main options after the user has entered the credentials which are equipment rental menu and court reservation. This process uses one superclass Student and 2 subclasses which are Equipment Rental and Court Reservation. Court resrvation is handled by Muhammad Akmal while Equipment Rental is handled by Wan Amirul Afiq. Both subclasses are then combined by Zainul Muttaqin using a polymorphism approach. The main method of our project is handled by all three members whereas Muhammad Akmal handled the part when the user opts for court reservations and Wan Amirul Afiq takes the part when the user opts for equipment rental. To complete the system, Zainul Muttaqin held the part which handles the superclass Student input and outputs and also compiled all the necessary lines of coding to complete the system and also the debugging process.

Chapter 2

INPUT

The input to run our system consists of 2 ways. First is the input from the user directly through GUI. Second is through text file where the prices are taken from the text file of the particular service either equipment rental or court reservation. The text file input eases the staff to change the rates anytime they want without having to search the code for price lines.

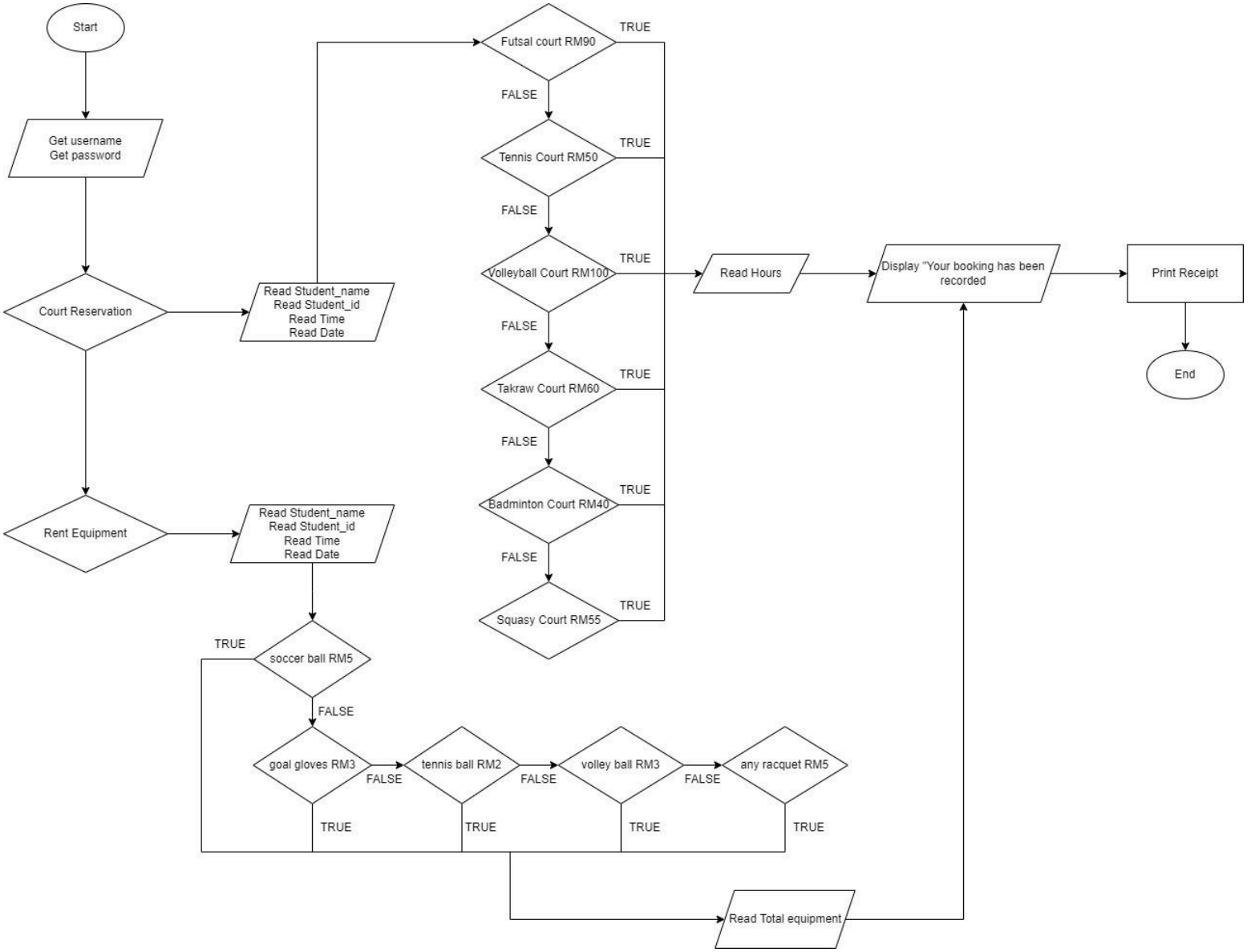
OUTPUT

After completing all the reservations and rental, the system will notify the user that their booking has been recorded and it will then print out a receipt out of the items that the user opted for in a form of a text file. The receipt will contain everything covered from the user's name, ID, date, time and total that has to be paid.

PROCESS

The system we created is simple and user friendly where the user shall opt for everything one by one without any hustle. First is the login page. The user has to enter the credentials such as the username and password to the system. Only then the system will grant access to the services that are Equipment rental or to Court reservations. After being prompted to the desired option, the user has to fill up the blank spaces that demand for the user's credentials such as name, ID, date and the time reserved. Next, the difference will prevail in the next options either the options for court reservations or the available equipment for rental are displayed.

CHAPTER 3


Start
Get username
Get password
Court Reservation
Read Student_name
Read Student_id
Read Time
Read Date
Futsal court RM90
TRUE
FALSE
Tennis Court RM50
TRUE
FALSE
Volleyball Court RM100
TRUE
FALSE
Takraw Court RM60
TRUE
FALSE
Badminton Court RM40
TRUE
FALSE
Squasy Court RM55
TRUE
Read Hours
Display "Your booking has been recorded
Print Receipt
End
Rent Equipment
Read Student_name
Read Student_id
Read Time
Read Date
TRUE
soccer ball RM5
FALSE
goal gloves RM3
FALSE
tennis ball RM2
FALSE
volley ball RM3
FALSE
any racquet RM5
TRUE
TRUE
TRUE
TRUE
Read Total equipment

CHAPTER 4 : SOURCE CODE

Main class: studentApp

```
package miniProject;

public class studentApp {

	public static void main(String[] args) {
		// TODO Auto-generated method stub\
		PasswordID PID = new PasswordID();
		loginPage LP = new loginPage(PID.getLoginInfo());

	}

}
```

Class : PasswordID

```
package miniProject;
import java.util.HashMap;

public class PasswordID {
	//instance member...
	private HashMap<String, String> loginInfo = new HashMap<String, String>();
	//instance member

	//default constructor...
	public PasswordID() {
		loginInfo.put("Afiq", "2020627832");
		loginInfo.put("Zainul", "2020828578");
		loginInfo.put("Akmal", "2020819092");
		loginInfo.put("Farhana", "2020828182");
	}
	//default constructor

	//getter and setter...
	public HashMap<String, String> getLoginInfo() {
		return loginInfo;
	}

	public void setLoginInfo(HashMap<String, String> loginInfo) {
		this.loginInfo = loginInfo;
	}
	//getter and setter

}
```

Class : loginPage

```
package miniProject;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;

public class loginPage implements ActionListener{
	//instance members...
	private JFrame frame = new JFrame();
	private JButton loginButton = new JButton("Login");
	private JButton resetButton = new JButton("Reset");
	private JTextField usernameField = new JTextField();
	private JPasswordField userPasswordField = new JPasswordField();
	private JLabel uitmLabel = new JLabel("Welcome to UiTM Perlis Sport Centre!");
	private JLabel usernameLabel = new JLabel("Username:");
	private JLabel userPasswordLabel = new JLabel("Password:");
	private JLabel messageLabel = new JLabel();
	private HashMap<String, String> loginInfo = new HashMap<String, String>();
	//instance members

	//normal constructors...
	public loginPage(HashMap<String, String> LI) {
		loginInfo = LI;

		//label(x, y, width, height)...
		uitmLabel.setFont(new Font(null, Font.BOLD, 18));
		uitmLabel.setBounds(30, 0, 500, 100);
		usernameLabel.setBounds(50, 100, 75, 25);
		userPasswordLabel.setBounds(50, 150, 75, 25);
		//label(x, y, length, height)

		//popup messages...
		messageLabel.setBounds(125, 250, 250, 35);
		messageLabel.setFont(new Font(null, Font.ITALIC, 25));
		//popup messages

		//input field...
		usernameField.setBounds(125, 100, 200, 25);
		userPasswordField.setBounds(125, 150, 200, 25);
		//input field

		//button...
		loginButton.setBounds(125, 200, 100, 25);
		loginButton.setFocusable(false); //clear small box around the text
		loginButton.addActionListener(this); //action performed
		resetButton.setBounds(225, 200, 100, 25);
		resetButton.setFocusable(false); //clear small box around the text
		resetButton.addActionListener(this); //action performed
```

```
		//button

		//to make the label and button visible...
		frame.add(uitmLabel);
		frame.add(usernameLabel);
		frame.add(userPasswordLabel);
		frame.add(messageLabel);
		frame.add(usernameField);
		frame.add(userPasswordField);
		frame.add(loginButton);
		frame.add(resetButton);
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame.setSize(420, 420);
		frame.setTitle("Login Page");
		frame.setLayout(null);
		frame.setVisible(true);
		//to make the label and button visible

	}
	//normal constructors

	//user interaction...
	public void actionPerformed(ActionEvent aE) {
		//reset to null...
		if(aE.getSource()==resetButton) {
			usernameField.setText("");
			userPasswordField.setText("");
		}
		//reset to null

		if(aE.getSource()==loginButton) {
			String username = usernameField.getText();
			String password = String.valueOf(userPasswordField.getPassword());
//convert JPassword into string

			//check info from PasswordID...
			if(loginInfo.containsKey(username)) {
				if(loginInfo.get(username).equals(password)) {
					messageLabel.setForeground(Color.green);
					messageLabel.setText("Login succussful!");
					frame.dispose();
					mainMenu mM = new mainMenu(username);
				}
				else {
					messageLabel.setForeground(Color.red);
					messageLabel.setText("Wrong password!");
				}
			}
```

```
                        else {
                                messageLabel.setForeground(Color.red);
                                messageLabel.setText("Username not found!");
                        }
                        //check info from PasswordID
                }
        }
        //user interaction

}
```

Class : mainMenu

```
package miniProject;
import java.awt.*;
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JFrame; //used if extend JFrame
import javax.swing.JButton;

public class mainMenu implements ActionListener{
        //instance members...
        private JFrame frame = new JFrame();
        private JLabel optionMessage = new JLabel("Select an option:");
        private JLabel welcomeLabel = new JLabel();
        private JButton buttonCourt = new JButton("Court Reservation");
        private JButton buttonEquipment = new JButton("Rent Equipments");
        //instance members

        //normal contructor...
        public mainMenu(String username) {
                welcomeLabel.setBounds(0, 0, 500, 35);
                welcomeLabel.setFont(new Font(null, Font.PLAIN, 25));
                welcomeLabel.setText("Welcome student, " + username + "!");
                optionMessage.setBounds(125, 50, 150, 50);
                optionMessage.setFont(new Font(null, Font.PLAIN, 15));

                //option button...
                buttonCourt.setBounds(125, 100, 150, 50);
                buttonCourt.setFocusable(false);
                buttonCourt.addActionListener(this);
                buttonEquipment.setBounds(125, 200, 150, 50);
                buttonEquipment.setFocusable(false);
                buttonEquipment.addActionListener(this);
                //option button

                //add frame...
                frame.add(optionMessage);
```

```
		frame.add(buttonCourt);
		frame.add(buttonEquipment);
		frame.add(welcomeLabel);
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame.setSize(420, 420);
		frame.setTitle("Main Menu");
		frame.setLayout(null);
		frame.setVisible(true);
		//add frame
	}
	//normal constructor

	//getter and setter...
	public JButton getButtonCourt() {
		return buttonCourt;
	}

	public void setButtonCourt(JButton buttonCourt) {
		this.buttonCourt = buttonCourt;
	}

	public JButton getButtonEquipment() {
		return buttonEquipment;
	}

	public void setButtonEquipment(JButton buttonEquipment) {
		this.buttonEquipment = buttonEquipment;
	}
	//getter and setter

	//button action...
	public void actionPerformed(ActionEvent E) {
		if(E.getSource()==buttonCourt) {
			frame.dispose();
			new CourtGUI();
		}
		else if(E.getSource()==buttonEquipment) {
			frame.dispose();
			new EquipmentGUI();
		}
	}
	//button action

}
```

Super Class : Student

```
package miniProject;
```

```
public abstract class Student {
	private String studentName;
	private int studentID;
	private String date;
	private String time;

	public Student() {
		studentName = "";
		studentID = 0;
		date = "";
		time = "";
	}

	public Student (String sN, int ID, String dt, String tm) {
		studentName = sN;
		studentID = ID;
		date = dt;
		time = tm;
	}

	public void setstudentName (String sN) {studentName = sN;}
	public void setstudentID (int ID ) {studentID = ID;}
	public void setdate (String dt) {date = dt;}
	public void settime (String tm) {time = tm;}

	public String getstudentName () {return studentName;}
	public int getstudentID () {return studentID;}
	public String getdate () {return date;}
	public String gettime () {return time;}

	public String toString() {
		return "\n\nName: "+studentName+
				"\nID: "+studentID+
				"\nDate: "+date+
				"\nTime: "+time;
	}

	public abstract double calctotal();
}
```

Subclass : courtReservation

```
 package miniProject;
 import java.util.*;
 import java.io.*;

public class courtReservation extends Student{
	//instance members...
	private String sport;
```

```
	private double courtbill;
	private double hour;
	//instance members

	//default constructor...
	public courtReservation() {
		super();
		sport = "";
		courtbill = 0;
		hour = 0;
	}
	//default constructor

	//normal constructor...
	public courtReservation(String sN, int ID, String dt, String tm, String sp, double cb,
double hr){
	  super(sN, ID, dt, tm);
	  sport = sp;
	  courtbill = cb;
	  hour = hr;
	}
	//normal constructor

	  //setter
	  public void setsport(String sp){sport=sp;}
	  public void setcourtbill(double cb){courtbill = cb;}
	  public void sethour(double hr){hour=hr;}

	  //getter
	  public String getsport() {return sport;}
	  public double getcourtbill() {return courtbill;}
	  public double gethour() {return hour;}

	  public double calctotal() {
			double total = 0;

			total = courtbill*hour;
			return total;
		}

	  //toString
	  public String toString(){
	    return super.toString()+"\nSport:"+sport+"\nHour: "+hour+"\nCourt Bill: "+courtbill;
	  }
}
```

Subclass : Equipment

```
package miniProject;
```

```
public class Equipment extends Student{
	private String equipment;
	private double price;
	private int quantity;

	public Equipment (){
		super();
		equipment = "";
		price = 0;
	}

	public Equipment (String sN, int ID, String dt, String tm, String eqp, double pr, int qt) {
		super(sN,ID,dt,tm);
		equipment = eqp;
		price = pr;
		quantity = qt;
	}

	public void setequipment (String eqp) {equipment = eqp;}
	public void setprice (double pr) {price = pr;}
	public void setquantity (int qt) {quantity = qt;}

	public String getequipment() {return equipment;}
	public double getprice() {return price;}
	public int getquantity() {return quantity;}

	public double calctotal() {
		double total = 0;

		total = price*quantity;
		return total;
	}
	public String toString() {
		return super.toString()+ "\nEquipment: "+equipment+
				"\nPrice per piece: RM"+price+
				"\nQuantity# : "+quantity;
	}

}
```

Court Reservation GUI Class : CourtGUI

```
package miniProject;
import java.awt.*;
import java.io.*;
import java.util.StringTokenizer;
import javax.swing.*;
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;

public class CourtGUI implements ActionListener{
	//instance members...
	private double courtBill;
	private String sport;
	private String [] court = new String[6];
	private String [] fee = new String[6];

	private JLabel message = new JLabel("Fill in the details");
	private JLabel studentName = new JLabel("Student's name:");
	private JLabel studentID = new JLabel("Student's ID:");
	private JLabel time = new JLabel("Time:");
	private JLabel date = new JLabel("Date:");
	private JTextField nameField = new JTextField();
	private JTextField idField = new JTextField();
	private JTextField timeField = new JTextField();
	private JTextField dateField = new JTextField();

	private JLabel message1 = new JLabel("Choose court to reserve:");
	private JButton proceed = new JButton("Proceed");
	private JLabel messageHours = new JLabel("Total hours:");
	private JTextField hoursField = new JTextField();
	private JCheckBox [] tick = new JCheckBox[6];
	//private JLabel [] item = new JLabel[6];
	private JFrame frame = new JFrame();

	private JFrame frame1 = new JFrame();
	private JLabel messageDetails = new JLabel("Your booking has been recorded!");
	//instance members

	//default constructors...
	public CourtGUI() {
		//message...
		message.setBounds(50, 25, 200, 25);
		message.setFont(new Font(null, Font.BOLD, 18));
		frame.add(message);
		//message

		//fill in the blanks...
		studentName.setBounds(10, 75, 200, 25);
		frame.add(studentName);
		studentID.setBounds(10, 125, 200, 25);
		frame.add(studentID);
		date.setBounds(10, 225, 200, 25);
		frame.add(date);
		time.setBounds(10, 175, 200, 25);
		frame.add(time);
```

```
nameField.setBounds(105, 75, 200, 25);
frame.add(nameField);
idField.setBounds(105, 125, 200, 25);
frame.add(idField);
dateField.setBounds(105, 175, 200, 25);
frame.add(dateField);
timeField.setBounds(105, 225, 200, 25);
frame.add(timeField);
//fill in the blanks

//message...
message1.setBounds(10, 280, 300, 45);
message1.setFont(new Font(null, Font.BOLD, 20));
frame.add(message1);
//message

//input hours...
messageHours.setBounds(10, 615, 200, 25);
frame.add(messageHours);
hoursField.setBounds(85, 615, 100, 25);
frame.add(hoursField);
//input hours

//button...
proceed.setBounds(200, 615, 100, 25);
proceed.setFocusable(false);
proceed.addActionListener(this);
frame.add(proceed);
//button

  BufferedReader in = null;

  int j=0;

  try {
        in = new BufferedReader(new
FileReader("D:\\eclipse-workspace\\Mini Project\\src\\miniProject\\court.txt"));

        String inData = null;

        while((inData=in.readLine())!=null) {
              StringTokenizer sT = new StringTokenizer(inData, ";");
              court[j] = sT.nextToken();
              fee[j] = sT.nextToken().trim();
              j++;
        }
```

```
                        int y=300;
                        for(int i=0; i<j; i++) {
                                y+=45;
                                JCheckBox fromString = new JCheckBox(court[i] + " RM" +
fee[i]);
                                tick[i] = fromString;
                                tick[i].setBounds(10, y, 300, 25);
                                tick[i].addActionListener(this);
                                tick[i].setFocusable(false);
                                frame.add(tick[i]);
                        }
                        in.close();
                }

                catch(FileNotFoundException fe) {
                        System.out.print(fe.getMessage());
                }
                catch(NumberFormatException fe) {
                        System.out.print(fe.getMessage());
                }
                catch(Exception e) {
                        System.out.print("Problem: " + e.getMessage());
                }

                //add frame...
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                frame.setSize(420, 700);
                frame.setTitle("Court Reservation");
                frame.setLayout(null);
                frame.setVisible(true);
                //add frame
        }
        //default constructor

        public void actionPerformed(ActionEvent e) {
                if(e.getSource()==tick[0]) {
                        tick[1].setSelected(false);
                        tick[2].setSelected(false);
                        tick[3].setSelected(false);
                        tick[4].setSelected(false);
                        tick[5].setSelected(false);
                }
                else if(e.getSource()==tick[1]) {
                        tick[0].setSelected(false);
                        tick[2].setSelected(false);
                        tick[3].setSelected(false);
                        tick[4].setSelected(false);
                        tick[5].setSelected(false);
```

```
		}
		else if(e.getSource()==tick[2]) {
			tick[0].setSelected(false);
			tick[1].setSelected(false);
			tick[3].setSelected(false);
			tick[4].setSelected(false);
			tick[5].setSelected(false);
		}
		else if(e.getSource()==tick[3]) {
			tick[0].setSelected(false);
			tick[2].setSelected(false);
			tick[1].setSelected(false);
			tick[4].setSelected(false);
			tick[5].setSelected(false);
		}
		else if(e.getSource()==tick[4]) {
			tick[0].setSelected(false);
			tick[2].setSelected(false);
			tick[3].setSelected(false);
			tick[1].setSelected(false);
			tick[5].setSelected(false);
		}
		else if(e.getSource()==tick[5]) {
			tick[0].setSelected(false);
			tick[2].setSelected(false);
			tick[3].setSelected(false);
			tick[4].setSelected(false);
			tick[1].setSelected(false);
		}

		for(int i=0; i<6; i++) {
			if(e.getSource()==tick[i]) {
				sport = court[i];
				courtBill = Double.parseDouble(fee[i]);
			}
		}

		if(e.getSource()==proceed) {
			double totalHours = Double.parseDouble(hoursField.getText());

			Student [] sT = new Student[1];
			String name = nameField.getText();
			int id = Integer.parseInt(idField.getText());
			String date = dateField.getText();
			String time = timeField.getText();

			sT[0] = new courtReservation(name, id, date, time, sport, courtBill,
totalHours);
```

```
                        frame.dispose();

                        PrintWriter receipt = null;
                        try {
                                receipt = new PrintWriter(new BufferedWriter(new
FileWriter("D:\\eclipse-workspace\\Mini Project\\src\\miniProject\\receipt.txt")));

                                receipt.println("RECEIPT DETAILS");
                                receipt.println("====================");

                                receipt.println(sT[0].toString());
                                if(sT[0] instanceof courtReservation) {
                                        courtReservation cR = (courtReservation)sT[0];
                                        receipt.println("Total: RM" + cR.calctotal());
                                        }

                                receipt.close();
                        }
                        catch(FileNotFoundException fe) {
                                System.out.println(fe.getMessage());
                        }
                        catch(NumberFormatException fe) {
                                System.out.println(fe.getMessage());
                        }
                        catch(Exception ex) {
                                System.out.println("Problem: " + ex.getMessage());
                        }

                        messageDetails.setBounds(10, 50, 300, 25);
                        frame1.add(messageDetails);

                        frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                        frame1.setSize(300, 200);
                        frame1.setTitle("Receipt");
                        frame1.setLayout(null);
                        frame1.setVisible(true);
                }

        }

}
```

Rental Equipments GUI Class : EquipmentGUI

```
package miniProject;
import java.awt.*;
import java.io.*;
import java.util.StringTokenizer;
```

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

public class EquipmentGUI implements ActionListener{
	//instance members...
	private double price;
	private String equipment;
	private String [] tool = new String[5];
	private String [] fee = new String[5];

	private JLabel message = new JLabel("Fill in the details");
	private JLabel studentName = new JLabel("Student's name:");
	private JLabel studentID = new JLabel("Student's ID:");
	private JLabel time = new JLabel("Time:");
	private JLabel date = new JLabel("Date:");
	private JTextField nameField = new JTextField();
	private JTextField idField = new JTextField();
	private JTextField timeField = new JTextField();
	private JTextField dateField = new JTextField();

	private JLabel message1 = new JLabel("Choose equipment to reserve:");
	private JButton proceed = new JButton("Proceed");
	private JLabel messageTotal = new JLabel("Total equipments:");
	private JTextField totalField = new JTextField();
	private JCheckBox [] tick = new JCheckBox[5];
	private JFrame frame = new JFrame();

	private JFrame frame1 = new JFrame();
	private JLabel messageDetails = new JLabel("Your booking has been recorded!");
	//instance members

	//default constructor...
	public EquipmentGUI() {
		//message...
		message.setBounds(50, 25, 200, 25);
		message.setFont(new Font(null, Font.BOLD, 18));
		frame.add(message);
		//message

		//fill in the blanks...
		studentName.setBounds(10, 75, 200, 25);
		frame.add(studentName);
		studentID.setBounds(10, 125, 200, 25);
		frame.add(studentID);
		date.setBounds(10, 225, 200, 25);
		frame.add(date);
		time.setBounds(10, 175, 200, 25);
```

```
		frame.add(time);

		nameField.setBounds(105, 75, 200, 25);
		frame.add(nameField);
		idField.setBounds(105, 125, 200, 25);
		frame.add(idField);
		dateField.setBounds(105, 175, 200, 25);
		frame.add(dateField);
		timeField.setBounds(105, 225, 200, 25);
		frame.add(timeField);
		//fill in the blanks

		//message...
		message1.setBounds(10, 280, 300, 45);
		message1.setFont(new Font(null, Font.BOLD, 20));
		frame.add(message1);
		//message

		//input hours...
		messageTotal.setBounds(10, 575, 200, 25);
		frame.add(messageTotal);
		totalField.setBounds(115, 575, 100, 25);
		frame.add(totalField);
		//input hours

		//button...
		proceed.setBounds(230, 575, 100, 25);
		proceed.setFocusable(false);
		proceed.addActionListener(this);
		frame.add(proceed);
		//button

		  BufferedReader in = null;

		  int j=0;

		  try {
			in = new BufferedReader(new
FileReader("D:\\eclipse-workspace\\Mini Project\\src\\miniProject\\equipments.txt"));

			String inData = null;

			while((inData=in.readLine())!=null) {
				StringTokenizer sT = new StringTokenizer(inData, ":");
				tool[j] = sT.nextToken();
				fee[j] = sT.nextToken().trim();
				j++;
			}
```

```
                        int y=300;
                        for(int i=0; i<j; i++) {
                                y+=45;
                                JCheckBox fromString = new JCheckBox(tool[i] + " RM" +
fee[i]);
                                tick[i] = fromString;
                                tick[i].setBounds(10, y, 300, 25);
                                tick[i].addActionListener(this);
                                tick[i].setFocusable(false);
                                frame.add(tick[i]);
                        }
                        in.close();
                  }

                  catch(FileNotFoundException fe) {
                        System.out.print(fe.getMessage());
                  }
                  catch(NumberFormatException fe) {
                        System.out.print(fe.getMessage());
                  }
                  catch(Exception e) {
                        System.out.print("Problem: " + e.getMessage());
                  }

                //add frame...
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                frame.setSize(420, 700);
                frame.setTitle("Equipment Reservation");
                frame.setLayout(null);
                frame.setVisible(true);
                //add frame

        }
        //default constructor

        public void actionPerformed(ActionEvent e) {
                if(e.getSource()==tick[0]) {
                        tick[1].setSelected(false);
                        tick[2].setSelected(false);
                        tick[3].setSelected(false);
                        tick[4].setSelected(false);
                }
                else if(e.getSource()==tick[1]) {
                        tick[0].setSelected(false);
                        tick[2].setSelected(false);
                        tick[3].setSelected(false);
                        tick[4].setSelected(false);
```

```
		}
		else if(e.getSource()==tick[2]) {
			tick[0].setSelected(false);
			tick[1].setSelected(false);
			tick[3].setSelected(false);
			tick[4].setSelected(false);
		}
		else if(e.getSource()==tick[3]) {
			tick[0].setSelected(false);
			tick[2].setSelected(false);
			tick[1].setSelected(false);
			tick[4].setSelected(false);
		}
		else if(e.getSource()==tick[4]) {
			tick[0].setSelected(false);
			tick[2].setSelected(false);
			tick[3].setSelected(false);
			tick[1].setSelected(false);
		}

		for(int i=0; i<5; i++) {
			if(e.getSource()==tick[i]) {
				equipment = tool[i];
				price = Double.parseDouble(fee[i]);
			}
		}

		if(e.getSource()==proceed) {
			int totalEQ = Integer.parseInt(totalField.getText());

			Student [] sT = new Student[1];
			String name = nameField.getText();
			int id = Integer.parseInt(idField.getText());
			String date = dateField.getText();
			String time = timeField.getText();

			sT[0] = new Equipment(name, id, date, time, equipment, price,
totalEQ);

			frame.dispose();

			PrintWriter receipt = null;
			try {
				receipt = new PrintWriter(new BufferedWriter(new
FileWriter("D:\\eclipse-workspace\\Mini Project\\src\\miniProject\\receipt.txt")));

				receipt.println("RECEIPT DETAILS");
				receipt.println("====================");
```

```
                    receipt.println(sT[0].toString());
                    if(sT[0] instanceof Equipment) {
                        Equipment eQ = (Equipment)sT[0];
                        receipt.println("Total: RM" + eQ.calctotal());
                        }

                    receipt.close();
                }
                catch(FileNotFoundException fe) {
                    System.out.println(fe.getMessage());
                }
                catch(NumberFormatException fe) {
                    System.out.println(fe.getMessage());
                }
                catch(Exception ex) {
                    System.out.println("Problem: " + ex.getMessage());
                }

                messageDetails.setBounds(10, 50, 300, 25);
                frame1.add(messageDetails);

                frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                frame1.setSize(300, 200);
                frame1.setTitle("Receipt");
                frame1.setLayout(null);
                frame1.setVisible(true);
        }
        }
}
```

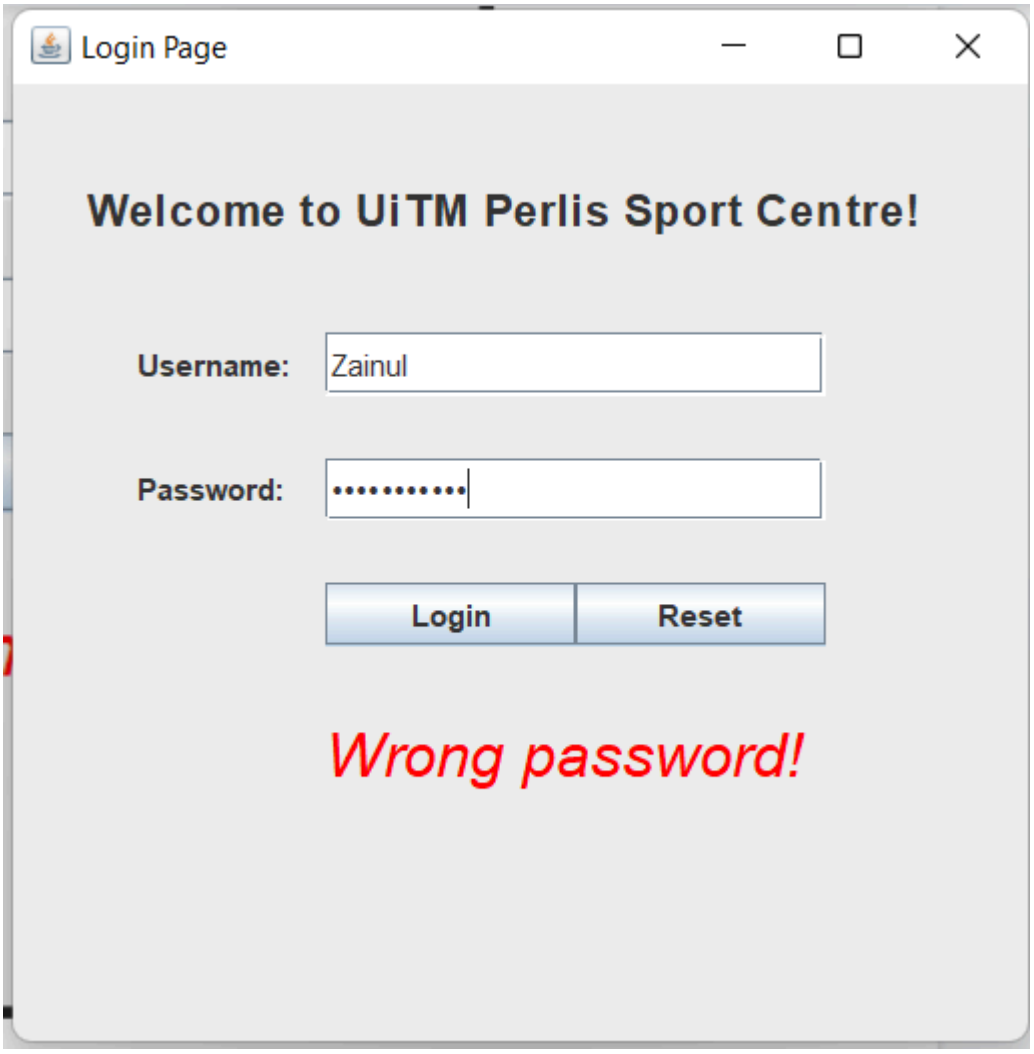
Court Reservation Text File : court.txt

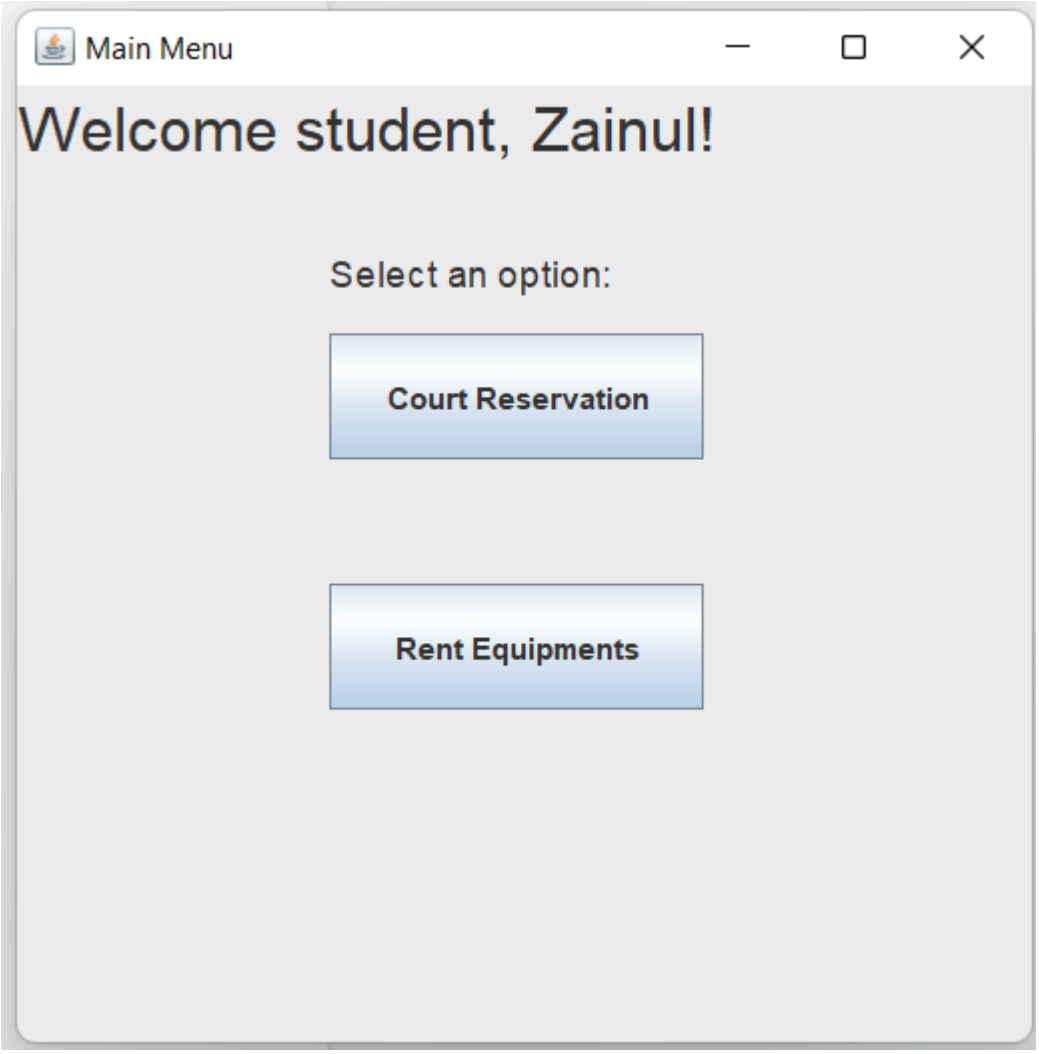
```
Futsal Court;90
Tennis Court;50
Volleyball Court;100
Takraw Court;60
Badminton Court;40
Squasy Court;55
```

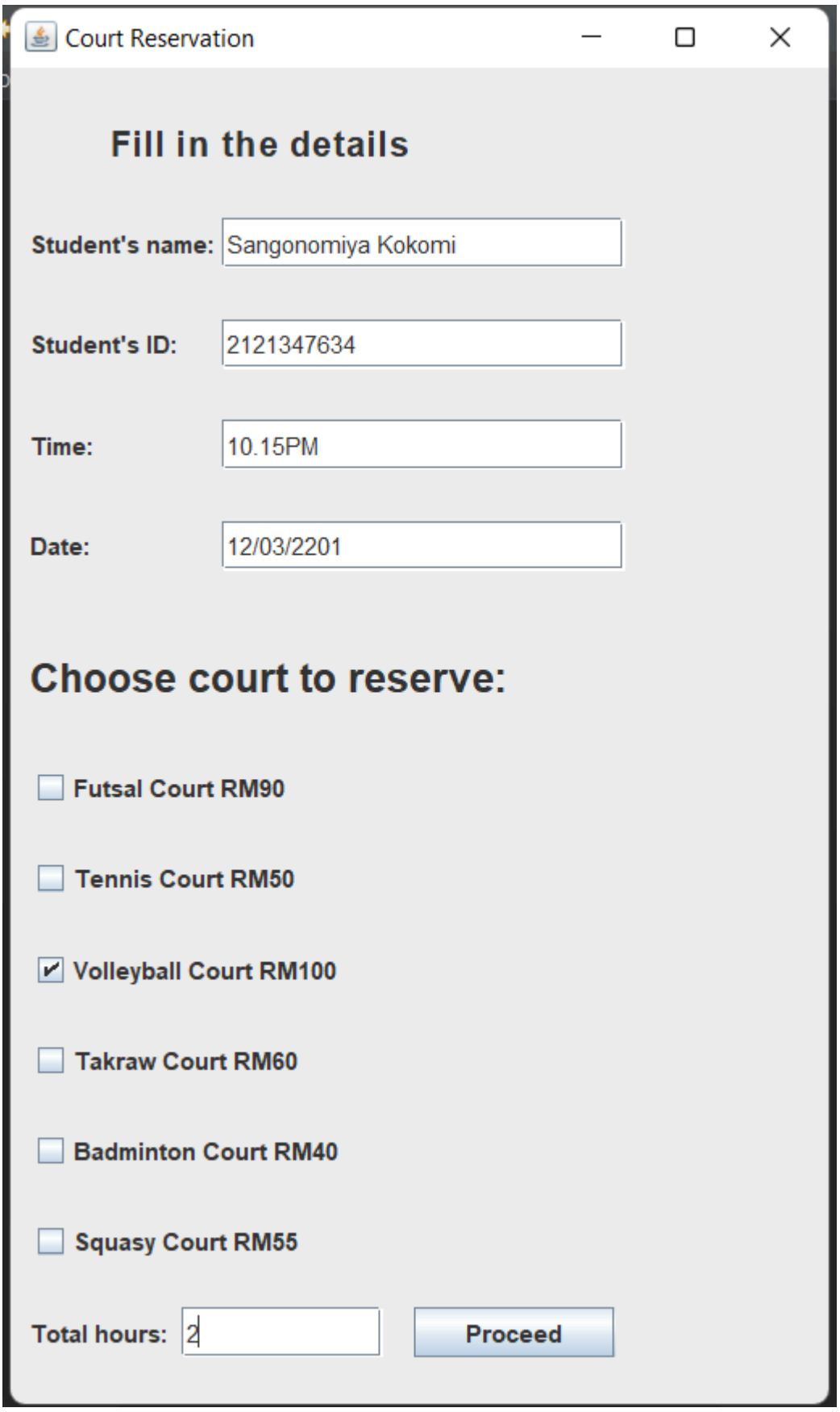
Rental Equipment Text File : equipments.txt

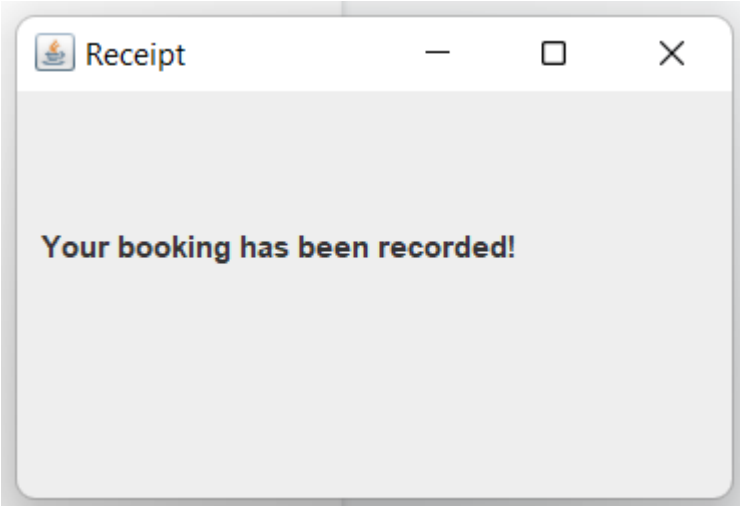
```
soccer ball:5
goal gloves:3
tennis balls:2
volley ball:3
any racquet:5
```

CHAPTER 5 : SAMPLE OF INPUT/OUTPUT

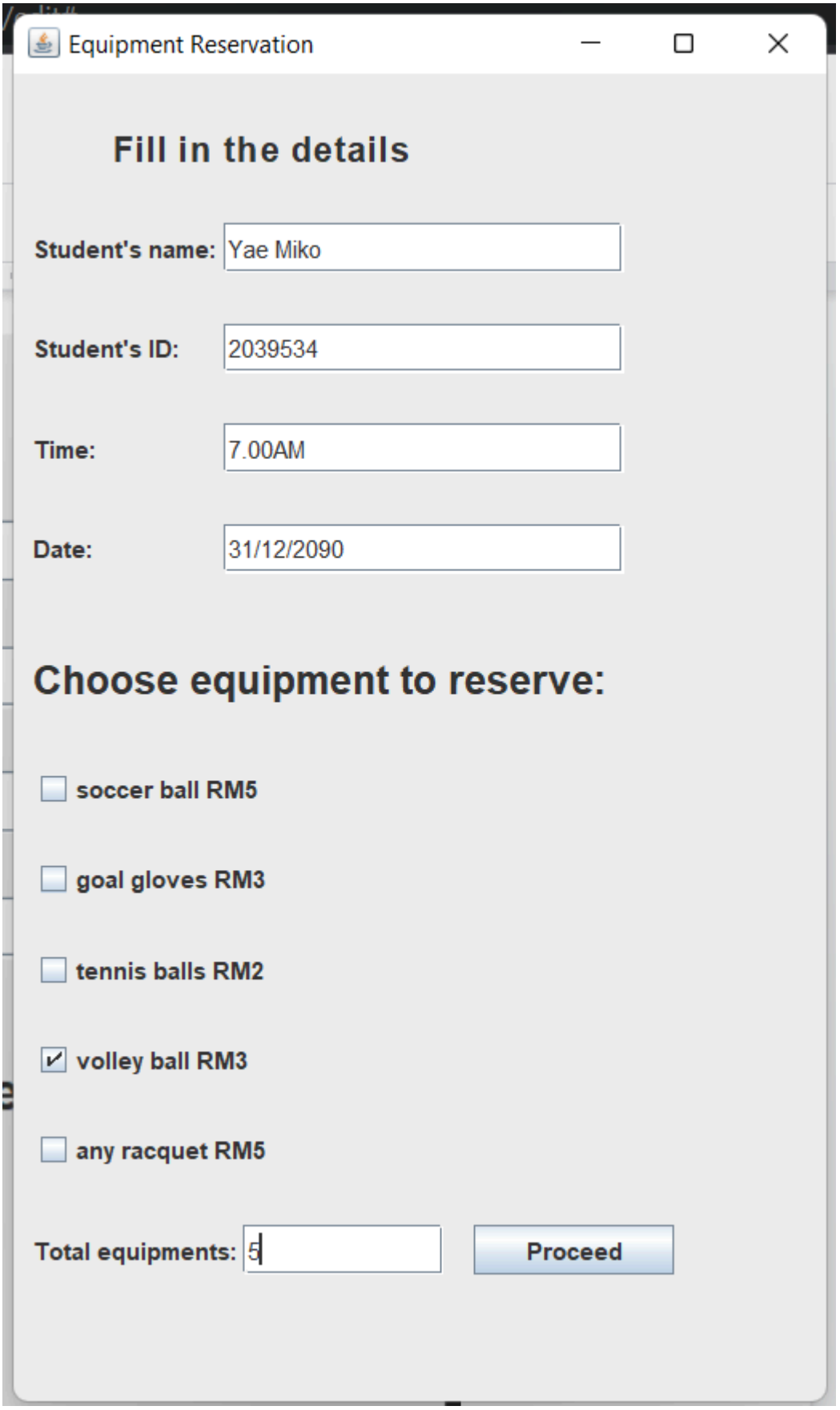


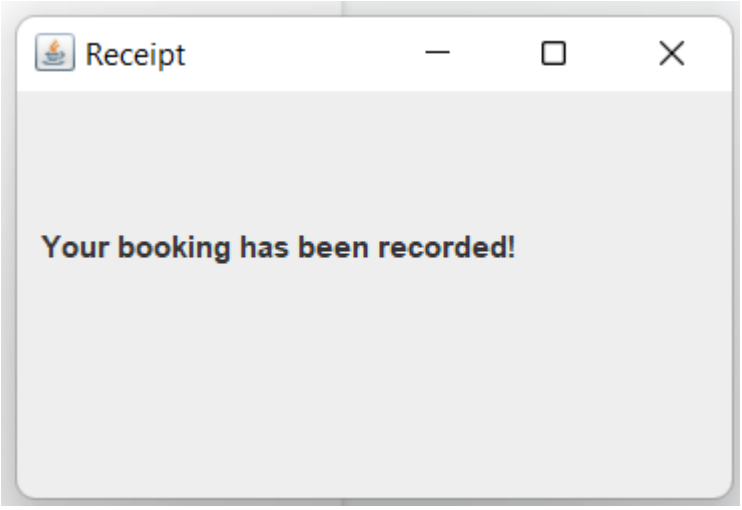

Main Menu
Welcome student, Zainul!
Select an option:
Court Reservation
Rent Equipments


Court Reservation
Fill in the details
Student's name:
Sangonomiya Kokomi
Student's ID:
2121347634
Time:
10.15PM
Date:
12/03/2201
Choose court to reserve:
Futsal Court RM90
Tennis Court RM50
Volleyball Court RM100
Takraw Court RM60
Badminton Court RM40
Squasy Court RM55
Total hours:
2
Proceed


Receipt
Your booking has been recorded!


receipt - Notepad
File Edit Format View Help
RECEIPT DETAILS
==========
Name: Sangonomiya Kokomi
ID: 2121347634
Date: 12/03/2201
Time: 10.15PM
Sport:Volleyball Court
Hour: 2.0
Court Bill: 100.0
Total: RM200.0


Equipment Reservation
Fill in the details
Student's name:
Yae Miko
Student's ID:
2039534
Time:
7.00AM
Date:
31/12/2090
Choose equipment to reserve:
soccer ball RM5
goal gloves RM3
tennis balls RM2
volley ball RM3
any racquet RM5
Total equipments:
5
Proceed


Receipt
Your booking has been recorded!


receipt - Notepad
File Edit Format View Help
RECEIPT DETAILS
==========
Name: Yae Miko
ID: 2039534
Date: 31/12/2090
Time: 7.00AM
Equipment: volley ball
Price per piece: RM3.0
Quantity# : 5
Total: RM15.0