SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY

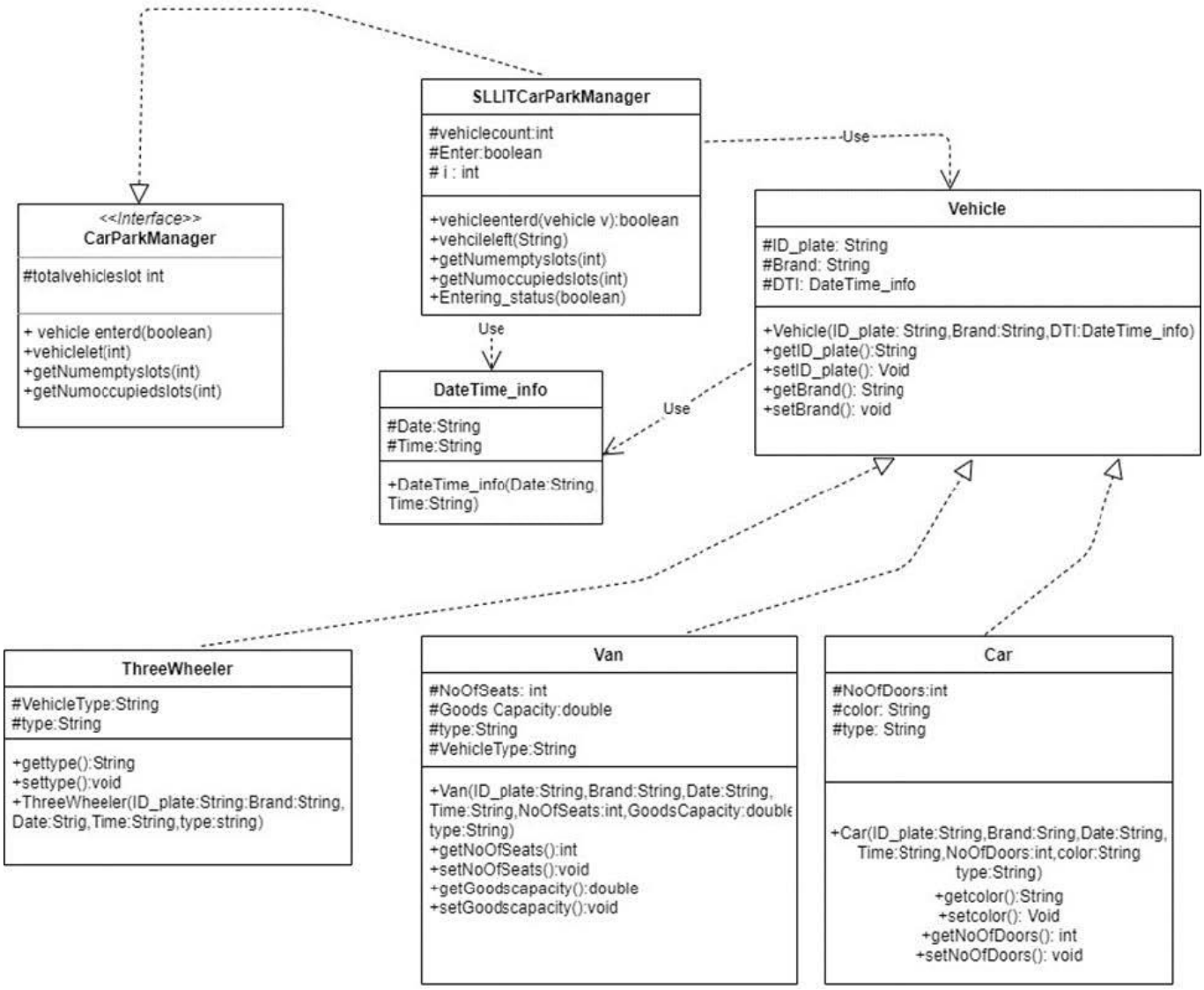
Faculty of Engineering

EC2492 - Object oriented programming

final assigment-Car park management

Jayawardene.M.V.G.J
EN21466830

UMA diagram


<<Interface>>
CarParkManager
#totalvehicleslot int
+ vehicle enterd(boolean)
+vehiclelet(int)
+getNumemptyslots(int)
+getNumoccupiedslots(int)
SLLITCarParkManager
#vehiclecount:int
#Enter:boolean
i : int
+vehicleenterd(vehicle v):boolean
+vehcileleft(String)
+getNumemptyslots(int)
+getNumoccupiedslots(int)
+Entering_status(boolean)
Use
Use
Vehicle
#ID_plate: String
#Brand: String
#DTI: DateTime_info
+Vehicle(ID_plate: String,Brand:String,DTI:DateTime_info)
+getID_plate():String
+setID_plate(): Void
+getBrand(): String
+setBrand(): void
DateTime_info
#Date:String
#Time:String
+DateTime_info(Date:String,
Time:String)
Use
ThreeWheeler
#VehicleType:String
#type:String
+gettype():String
+settype():void
+ThreeWheeler(ID_plate:String:Brand:String,
Date:Strig,Time:String,type:string)
Van
#NoOfSeats: int
#Goods Capacity:double
#type:String
#VehicleType:String
+Van(ID_plate:String,Brand:String,Date:String,
Time:String,NoOfSeats:int,GoodsCapacity:double
type:String)
+getNoOfSeats():int
+setNoOfSeats():void
+getGoodscapacity():double
+setGoodscapacity():void
Car
#NoOfDoors:int
#color: String
#type: String
+Car(ID_plate:String,Brand:Sring,Date:String,
Time:String,NoOfDoors:int,color:String
type:String)
+getcolor():String
+setcolor(): Void
+getNoOfDoors(): int
+setNoOfDoors(): void

Code

```
public abstract class Vehicle { // create a abstract class
	protected String ID_plate; // create a String for ID_plate
	protected String Brand; // Create a String For Brand
	protected DateTime_info DTI;

    public Vehicle(String ID_plate,String Brand,String Date,String Time) { // create
a overloading constructor for vehicle class
	this.ID_plate=ID_plate;
	this.Brand=Brand;
	DTI = new DateTime_info(Date, Time);
}
	public String getID_plate() {// get the data for ID_plate
		return ID_plate;

	}

	public void setID_plate(String ID_plate) {// set the data for ID_plate
		this.ID_plate = ID_plate;

	}

	public String getBrand() { // get the data for Brand
		return Brand;

	}

	public void setBrand(String Brand) { // set the data for ID_plat
		this.Brand = Brand;

	}

	abstract void Displaying_Details();// make a abstract method for Displaying
Details
}
```

```
public class Car extends Vehicle { // extends Car class as a sub class of Vehicle
	private int NoOfDoors; // Creating integer for number of doors private
	String Color; // create a String type for color private String type
	="Car"; // create a String for type variable

	public Car(String ID_plate, String Brand, String Date,String Time, int
NoOfDoors, String Color) { //make a constructor of a car class
		super(ID_plate, Brand, Date, Time); this.NoOfDoors
		= NoOfDoors;
		this.Color = Color;


	}
public int getNoOfDoors() { //get the data for No of doors
	return NoOfDoors;

}
public void setNoOfDoors(int NoOfDoors) { // set the data for No of doors
	this.NoOfDoors = NoOfDoors;

}
public String getColor() { //get the data for No of doors
	return Color;

}
public void setColor(String Color) { // set the data for setting the color
	this.Color = Color;

}

public void Displaying_Details() { System.out.println("Vehicle type
	= " + this.type);
	System.out.println("Vehicle ID = " + super.getID_plate());
	System.out.println("Vehicle Brand = " + super.getBrand());
	System.out.println("Number of Doors = " +getNoOfDoors());
	System.out.println("Color of the car = " + this.Color);
}




}
```

```
public class Van extends Vehicle{// extends Van class as a sub class of Vehicle
	private int NoOfSeats; //Creating integer for number of seats
	private double GoodsCapacity;//Creating double variable type for Goods
capacity
	private String type; // Create a String for Type of the van
	private String Vehicletype ="Van";// create String for Type of a vehicle

	public Van(String ID_plate, String Brand, String Date, String Time, int
NoOfSeats, double GoodsCapacity,String type) {//make a constructor of a van class
		super(ID_plate, Brand, Date, Time);
		this.NoOfSeats = NoOfSeats; this.GoodsCapacity
		= GoodsCapacity; this.type=type;

	}

	public int getNoOfSeats() { //get the data for No of seats
		return NoOfSeats;

	}

	public void setNoOfSeats(int NoOfSeats) { //set the data for No of seats
		this.NoOfSeats = NoOfSeats;

	}

	public double getGoodsCapacity() { //get the data for goods capacity
		return GoodsCapacity;

	}

	public void setGoodsCapacity(double GoodsCapacity){ //set the data for goods
capacity

		this.GoodsCapacity = GoodsCapacity;
	}

	public void Displaying_Details() {
		System.out.println("Vehicle type = "+ this.Vehicletype);
		System.out.println("Vehicle ID = " + super.getID_plate());
		System.out.println("Van Type = " +this.type);
		System.out.println("Vehicle Brand = " + super.getBrand());
		System.out.println("Number of Seats = " +getNoOfSeats());
		System.out.println("Goods Capacity(kg) = " +getGoodsCapacity());
}




}
```

```
public class ThreeWheeler extends Vehicle {// extends ThreeWheeler class as a sub
class of Vehicle
	private String Vehicletype = "ThreeWheeler"; // Create a String type for type
of the vehicle
	private String type; // create another String type for type of the
ThreeWheeler

	public ThreeWheeler(String ID_plate, String Brand, String Date, String Time,
String type) { // make a constructor of ThreeWheeler class
		super(ID_plate, Brand, Date, Time); this.type
		= type;

	}

	public String gettype() { //get the data for type of the ThreeWheeler
		return type;

	}

	public void settype(String type) { //set the data for type of the ThreeWheeler
		this.type = type;

	}
public void Displaying_Details() {
	System.out.println("Vehicle type = " + this.Vehicletype);
	System.out.println("Vehicle ID = " + super.getID_plate());
	System.out.println("Vehicle Brand = " + super.getBrand());
	System.out.println("ThreeWheeler Type = " + this.type);

}
}
public class DateTime_info { // create a class name as DateTime_info
	private String Date; // create a String for Date
	private String Time; // create a String for Time

	public DateTime_info(String Date,String Time) { // make a constructor for
DateTime_info class
		this.Date=Date;
		this.Time=Time;
}
public String getDate() { // get the data for the Date
	return Date;

}

public void setDate(String Date) { //set the data for the Date
	 this.Date = Date;
```

```
}
public String getTime() { //get the data for the Time
	return Time;

}

public void setTime(String Time) {//set the data for the Time
	this.Time = Time;

}
}
public interface CarParkManager { // create a interface name as CarParkManager
	final int totalVehicleSlots =20;
	boolean vehicleEntered(Vehicle v);
	int vehicleLeft(String noPlate);
	int getNumEmptySlots();
	int getNumOccupiedSlots();


}
public class SLIITCarParkManager implements CarParkManager {
private static int vehicleCount = 0; //Create a Vehicle Count Variable

	private Vehicle []slot = new Vehicle[totalVehicleSlots]; // Create An Array
for 20 Slots
	private boolean Enter; // Create a boolean function

	public boolean vehicleEntered(Vehicle v) {

		if (vehicleCount < totalVehicleSlots){
	// set a if condition ,if there are equal or less than 20
vehicles

	int i=0;
	while(i <= vehicleCount){ // Set a while loop
		if(slot[i] == null){
			slot[i] = v;
			vehicleCount = vehicleCount + 1; //increase the

vehicle count by one
					System.out.println("Vehicle entered slot number

:"+(i+1)); // Displaying entered slot number
					System.out.println("Remaining slots
:"+getNumEmptySlots()); //Displaying Remaining slots
					Enter= true;
					break;


}
				i++;




}
```

```
}
  else  {


          Enter=false;
       }


return  Entering_Status(Enter);
}
       public int vehicleLeft(String NumPlate) { //checking left vehicle by its id
plate number

              int i=0;
              while(i <= totalVehicleSlots){
                     String No = slot[i].getID_plate();
                     if(No == NumPlate) {
                            System.out.println("Lefted Vehicle slot No-"+(i+1)+"
Details:"); //searching the slot number of the left vehicle
                            slot[i].Displaying_Details(); // displaying the Details
                            vehicleCount--; // vehicle count decrement one by one
                            System.out.println("Remaining slots:"+getNumEmptySlots()
+"\n"); // print remaining slots
                            slot[i] = null;
                            break;

                     }
                     i++
                     ;
              }

              return getNumEmptySlots();
       }

       public int getNumEmptySlots() { //create a method

              return totalVehicleSlots -vehicleCount; // Subtract vehicle count from
total vehicle slots
       }
       public int getNumOccupiedSlots() { // create a method

              return vehicleCount;
       }

       private boolean Entering_Status(boolean Enter) { //create a boolean for
entering_status
                  if(Enter==true) {
                                System.out.println("completing vehicle entering
process"+"\n");

                  }else if(Enter==false){
                                System.out.println("Dont completing vehicle entering

process"+"\n");

                   }
                   return Enter;
             }
}
```

```
public class Main { // create a main class

	public static void main(String[] args) {
		SLIITCarParkManager SP = new SLIITCarParkManager(); // create a object
From SLIITCarParkManager
		Vehicle V; // create a object from Vehicle class

		V = new Van("HY-1231", "Hyundai", "01-01-2022", "08.00 AM",
12,1000,"Passenger Van");
		SP.vehicleEntered(V); // calling a vehicle entered method

		V = new ThreeWheeler("QCA-1369", "Bajaj", "1-01-2022", "08.10 AM",
"NotForHire");
		 SP.vehicleEntered(V); // calling a vehicle entered method


		  V = new Car("ABC-1367", "Mercedes-Benz", "01-01-2022", "08.30 AM", 4,
"red");
		  SP.vehicleEntered(V);

		  SP.vehicleLeft("HY-1231"); // calling a vehicle left method


		  V = new ThreeWheeler("AAB-1524", "TVS King", "01-01-2022", "10.00 AM",

"Taxi");

		SP.vehicleEntered(V);

		SP.vehicleLeft("ABC-1367"); // calling a vehicle left method

		SP.vehicleLeft("AAB-1524"); // calling a vehicle left method


		 V = new Car("AAD-1964", "Toyota", "01-01-2022", "11.00 AM", 4, "Black");
		SP.vehicleEntered(V);


		 V = new Van("LC-4678", "Nissan", "01-01-2022", "11.45 AM", 15,
1500,"Goods Van");
		 SP.vehicleEntered(V);
}

}
```