UTP
UNIVERSITI TEKNOLOGI PETRONAS

OBJECT ORIENTED PROGRAMMING

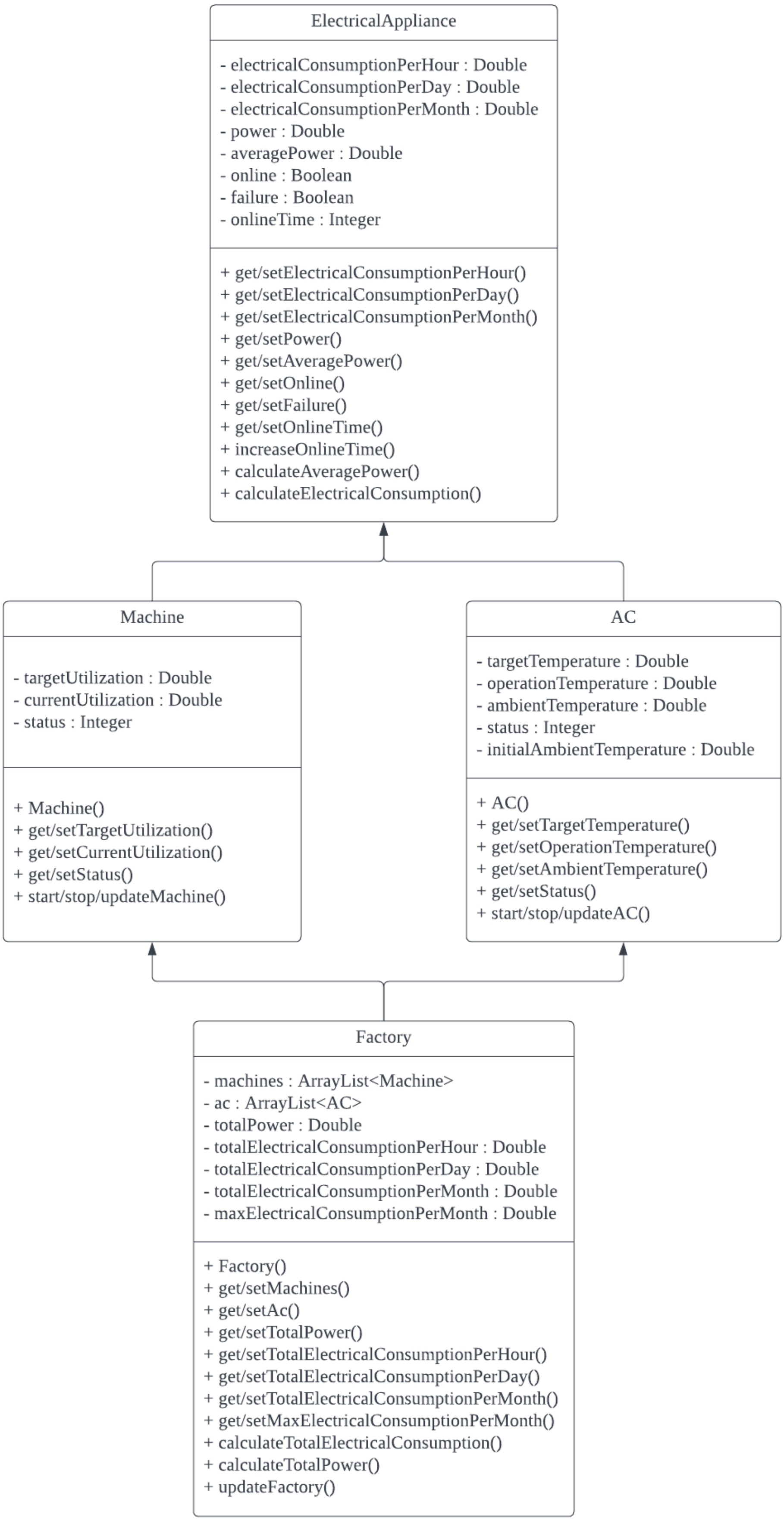
SEPTEMBER 2022

COURSE PROJECT

GROUP 6

STUDENT NAME	STUDENT ID
WONG XING HAO	21000612
LOO PEI XIN	21000483
NUR ADRIANA BINTI MUHAMAD AIZAM	21000162
NAJWA NADIRA BINTI HELMI	21000406

UML DIAGRAM



ElectricalAppliance

- electricalConsumptionPerHour : Double
- electricalConsumptionPerDay : Double
- electricalConsumptionPerMonth : Double
- power : Double
- averagePower : Double
- online : Boolean
- failure : Boolean
- onlineTime : Integer

+ get/setElectricalConsumptionPerHour()
+ get/setElectricalConsumptionPerDay()
+ get/setElectricalConsumptionPerMonth()
+ get/setPower()
+ get/setAveragePower()
+ get/setOnline()
+ get/setFailure()
+ get/setOnlineTime()
+ increaseOnlineTime()
+ calculateAveragePower()
+ calculateElectricalConsumption()

Machine

- targetUtilization : Double
- currentUtilization : Double
- status : Integer

+ Machine()
+ get/setTargetUtilization()
+ get/setCurrentUtilization()
+ get/setStatus()
+ start/stop/updateMachine()

AC

- targetTemperature : Double
- operationTemperature : Double
- ambientTemperature : Double
- status : Integer
- initialAmbientTemperature : Double

+ AC()
+ get/setTargetTemperature()
+ get/setOperationTemperature()
+ get/setAmbientTemperature()
+ get/setStatus()
+ start/stop/updateAC()

Factory

- machines : ArrayList<Machine>
- ac : ArrayList<AC>
- totalPower : Double
- totalElectricalConsumptionPerHour : Double
- totalElectricalConsumptionPerDay : Double
- totalElectricalConsumptionPerMonth : Double
- maxElectricalConsumptionPerMonth : Double

+ Factory()
+ get/setMachines()
+ get/setAc()
+ get/setTotalPower()
+ get/setTotalElectricalConsumptionPerHour()
+ get/setTotalElectricalConsumptionPerDay()
+ get/setTotalElectricalConsumptionPerMonth()
+ get/setMaxElectricalConsumptionPerMonth()
+ calculateTotalElectricalConsumption()
+ calculateTotalPower()
+ updateFactory()

Our team is focusing on developing a factory monitoring system. The UML diagram above representing the factory system part. We create four class for our system. First, ElectricalAppliance class as a fundamental class for all appliances appear in a factory. Then, we created two class which inherit from the ElectricalAppliance class. At current situation, we created Machine class and AC class as we assume a factory should basically have these electrical appliances. Finally, we create a Factory class which is the class that representing a factory which include Machine class and AC class.

Our factory monitoring system mostly focus on monitoring the power and electrical consumption. Hence, the ElectricalAppliance class encapsulated attributes about electrical consumption and power. It also has attributes for online, failure and online time which representing its status. Then, Machine class contain all attributes from ElectricalAppliance class since inherited from it, same as AC class. But Machine class and AC class also has its own attributes. Machine class consists of target utilization, current utilization, and its own status. While AC class encapsulated attributes like target temperature, ambient temperature, operation temperature, initial ambient temperature and also its own status. Finally, the Factory class contain attributes of a ArrayList of Machine and AC class, total power, total and max electrical consumption of a factory.

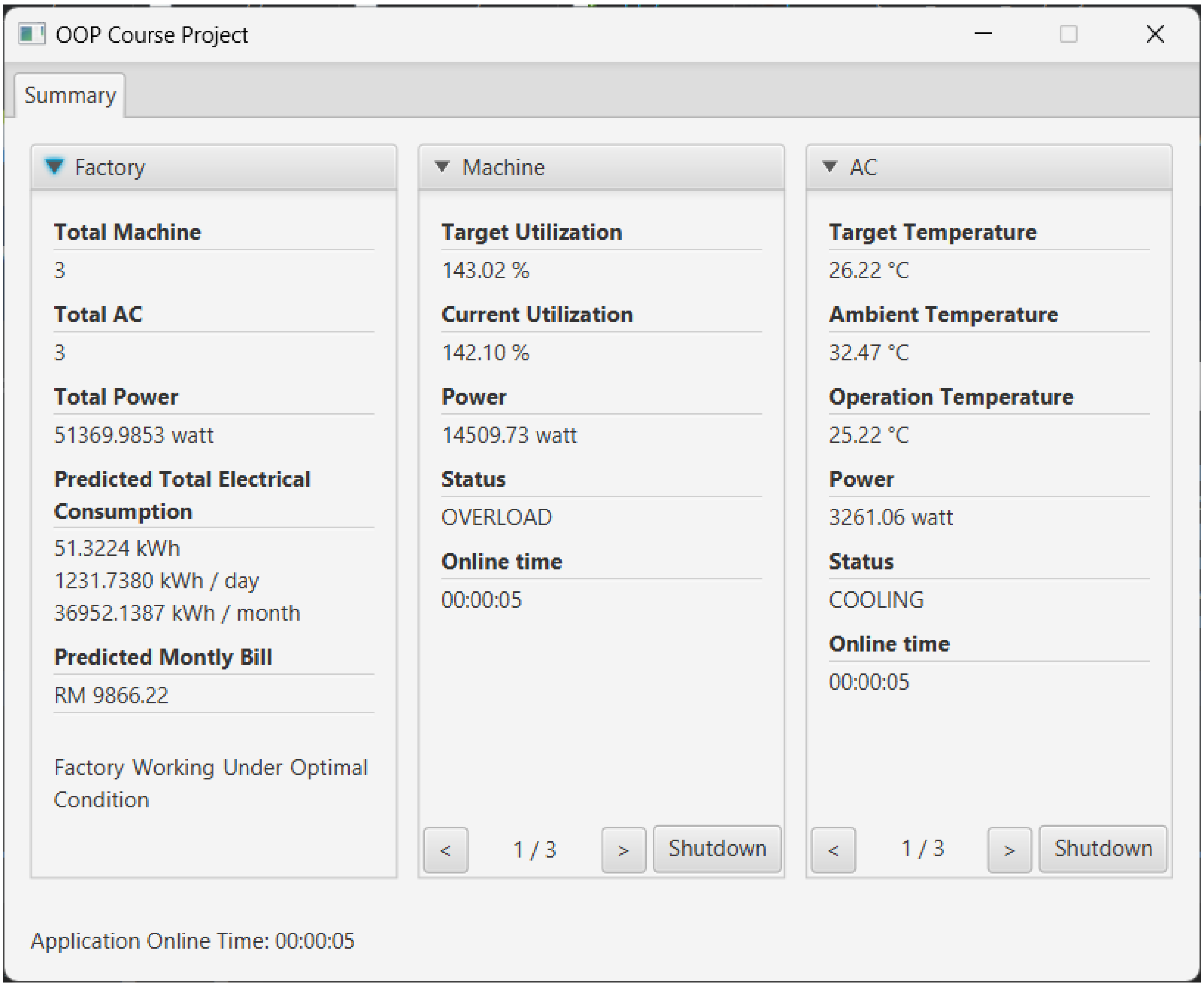
Every class contain method that related to its attributes and also itself and basically each class except for ElectricalAppliance class has a constructor. All class contain setter and getter for almost each attribute. ElectricalAppliance class also have method in increasing online time, calculate average power and calculate electrical consumption. Then, Machine class and AC class have start, stop, update method for controlling itself. Lastly, Factory class have method to calculate total electrical consumption and power, and a update method to update factory status.

CODE

https://github.com/xinghao2003/OOP-Course-Project/tree/main/OOP-Course-Project

We selected Apache NetBeans 15 to development this factory management system.

INTERFACE



Total Machine
3
Total AC
3
Total Power
51369.9853 watt
Predicted Total Electrical Consumption
51.3224 kWh
1231.7380 kWh / day
36952.1387 kWh / month
Predicted Montly Bill
RM 9866.22

Factory Working Under Optimal Condition

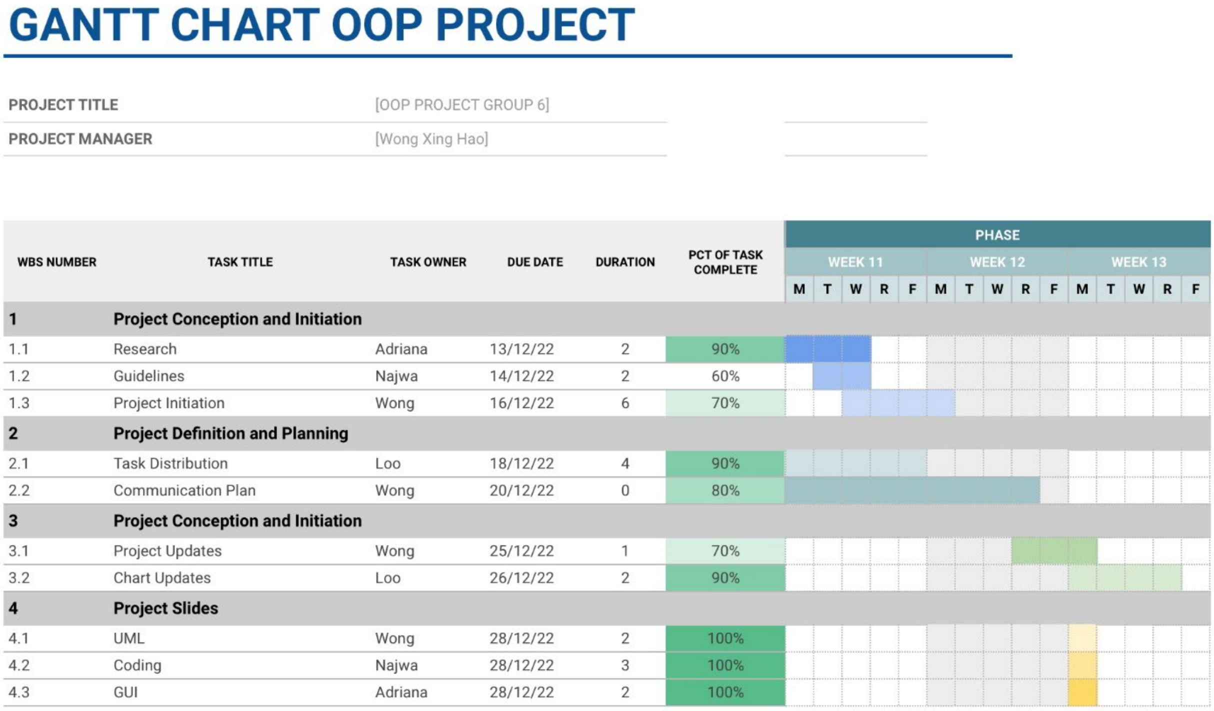
Target Utilization
143.02 %
Current Utilization
142.10 %
Power
14509.73 watt
Status
OVERLOAD
Online time
00:00:05

Target Temperature
26.22 °C
Ambient Temperature
32.47 °C
Operation Temperature
25.22 °C
Power
3261.06 watt
Status
COOLING
Online time
00:00:05

Application Online Time: 00:00:05

Our system interface is designed using JavaFX. Our design is to display all information in same page with multiple section. Our interface is a summary consists of 3 section which representing the factory, machine, and AC. The machine and AC section have buttons to change from one machine/AC to another machine/AC. Each machine and AC can be shutdown separately. The application will refresh the information every second until application being closed.

PROJECT MANAGEMENT



GANTT CHART OOP PROJECT

PROJECT TITLE	[OOP PROJECT GROUP 6]
PROJECT MANAGER	[Wong Xing Hao]

WBS NUMBER	TASK TITLE	TASK OWNER	DUE DATE	DURATION	PCT OF TASK COMPLETE	PHASE														
						WEEK 11					WEEK 12					WEEK 13				
						M	T	W	R	F	M	T	W	R	F	M	T	W	R	F
1	**Project Conception and Initiation**																			
1.1	Research	Adriana	13/12/22	2	90%	■	■	■												
1.2	Guidelines	Najwa	14/12/22	2	60%		■	■												
1.3	Project Initiation	Wong	16/12/22	6	70%			■	■	■	■									
2	**Project Definition and Planning**																			
2.1	Task Distribution	Loo	18/12/22	4	90%	■	■	■	■	■										
2.2	Communication Plan	Wong	20/12/22	0	80%	■	■	■	■	■	■	■	■	■						
3	**Project Conception and Initiation**																			
3.1	Project Updates	Wong	25/12/22	1	70%									■	■	■				
3.2	Chart Updates	Loo	26/12/22	2	90%											■	■	■	■	
4	**Project Slides**																			
4.1	UML	Wong	28/12/22	2	100%											■				
4.2	Coding	Najwa	28/12/22	3	100%											■				
4.3	GUI	Adriana	28/12/22	2	100%											■				

We used Gantt Chart to perform project management effectively and efficiently.