FACULTY OF COMPUTING
UTM Johor Bahru

Semester 2 2023/2024

Subject: SECJ1023 Programming Technique II

Section: 05

Group Name: Helios

Task: Group Project Final

Group Member

	Name	Matric Number
1	FAM QAI ZEN	A23CS0223
2	MAH WILSON	A23CS0243
3	TAN JIAN MING	A23CS0275

TABLE OF CONTENTS

1.0 Introduction

In this day and age of rapid urbanization, widespread industrialization, and wasteful consumer behaviors, we are facing an energy crisis as our energy demand continues to rise. In recent decades, the energy crisis has posed a serious threat to sustainability, particularly in developing countries and communities. For example, between 1990 and 2000, developing countries in Asia Pacific saw a 27% increase in conventional energy consumption, while global consumption increased by 11%. The increase in energy demand is expected to continue. Non-renewable energy sources such as coal, oil, and natural gas have been the primary source of energy since the dawn of technology. However, the depletion of these non-renewable energy sources necessitates the development of long-term, sustainable solutions. As a result, renewable energy sources are one of the most effective ways to address this issue and ensure humanity's future. In conjunction with that, our group is striving to contribute to this just cause by developing a system in line with the principle of clean and affordable energy based on the 7th Sustainable Development Goals aimed at helping future users of solar panels that are looking to make an environmentally-friendly investment to forecast the performance and crucial analytical data from the solar panels to make an informed decision. This system will also include tracking tools for users of solar panels for better management and monitoring.

2.0 Problem Background and Proposed Solution

Problem Background

1. **High upfront cost**

   Homeowners may feel intimidated and overwhelmed when making the decision to purchase solar panels for their home due to the high upfront cost of solar panels, as the market for solar panels has not matured and scaled to bring the price down to an affordable level. As a result, potential customers may be hesitant to spend a lump sum on an investment for which they are unsure of the returns because they cannot estimate the potential returns of a solar panel using traditional methods and metrics.

2. **Waste of produced energy**

   Energy that is produced but not used or stored is converted to heat and dissipated, essentially wasting energy. This is a common situation for many homeowners who have installed solar panels. This is due to the timing of supply and demand, as all of the energy produced by the solar panels occurs during the day when sunlight is abundant and the weather is clear, but there is no demand for energy to be used in the household, resulting in energy waste.

3. **Malfunctions and downtime**

   Solar panels are known for their dependability and durability. This is by design of the manufacturers; however, without consistent and regular maintenance and care by homeowners, solar panels will occasionally malfunction or have low efficiencies. This is a common problem among solar panel owners and operators because they are unable to assess the condition of the solar panels and determine when maintenance or repairs are required because the majority of the working mechanisms are static and hidden beneath the surface.

Proposed Solution

1. **Installment plan formulator and system (class Installment)**

   Through user-friendly interfaces and advanced algorithms, we will integrate a feature into the system to assist homeowners in creating an installment plan tailored to their needs, helping them understand the practical and financial requirements of purchasing a solar system and managing the steep costs associated with it.

2. **Return of investments and cost savings tracker (not yet Financial)**

   To enable existing and future solar panel users to precisely measure and monitor the energy savings derived from using the energy solar panels to meet household energy demands of the household and cut down energy reliance from the grid, hence saving and reducing the monthly electric bills. These will help users to be well-informed and forecast whether their purchase or future purchase is worth it or not.

3. **Optimal energy management (got liao under SolarPanel parent class)**

   To help solar panels users in managing their energy optimally, we will implement a feature in which the system will be automated to detect when there is energy demand, the energy produced by the solar panels will be prioritized to meet demand first and any unfulfilled energy demand will be compensated by relying on the energy grid. Any excess energy generated during low or no demand can be resold back to the grid to help offset the household’s energy bill.

4. **Maintenance (got liao)**

   We plan to include a feature that will monitor and regularly inspect the conditions of the solar panels and its main components such as the photovoltaic cells, the inverters and its wiring. The system will inform the users when a regular maintenance is due or when any urgent repairs are required.

In short, we want to make our system as accessible to as many people as possible in order to achieve sustainability and make long-term investments that are both climate friendly and affordable. This will help to promote the concept of sustainability while also meeting their own household's energy needs through the energy production of solar panels, saving non-renewables and electric bills.

3.0 Objectives

1. **Energy Monitoring and Management**

   By accurately calculating the energy output, we can monitor the performance of the solar panel system. This data aids in identifying potential issues, allowing for repairs or further improvements to improve performance. Ultimately, tracking energy generation is critical for maximizing return on investment and ensuring the long-term reliability and sustainability of solar power systems.

2. **Environmental Impact Assessment**

   By quantifying energy production, we can evaluate their environmental impact and contributions to sustainability goals. Solar panels contribute to climate change mitigation by generating electricity without emitting greenhouse gasses or relying on finite fossil fuels. The amount of energy produced is directly proportional to the environmental benefits gained, such as lower air and water pollution and less reliance on environmentally harmful energy sources.

3. **Financial and Return of Investment Analysis**

   The system provides a reliable source of data for analysis and understanding of the financials of solar energy generation. It allows users to be informed of the savings and returns associated with the solar panels so that they can determine whether they have made a good financial choice or not.

To summarize, these approaches highlight the importance of data-driven strategies in accelerating the implementation and efficacy of solar energy solutions for a more sustainable future.

4.0 Scopes

Our project scope includes several key points and areas in order to achieve our objective while living up to the customers expectations.

1. **Energy Monitoring and Management**

   Through our system, we can inform and update users on actionable data regarding the energy production, efficiency and any potential issues regarding the conditions of the solar panels, allowing users can have a higher level of involvement and information about their solar panels so that they can maintain the solar panels at their most optimal conditions to maximize efficiency and output.

2. **Environmental Impact Assessment**

   By analyzing the energy production data and the percentage of energy demand met by solar panels, we can calculate and quantify our role in reducing environmental impact and carbon footprint because the energy generated by solar panels is clean and sustainable.

3. **Financial and Return of Investment Analysis**

   The system is also optimized to provide detailed information about energy output, which can be used to calculate savings and returns on investment by reducing reliance on the energy grid and reselling excess energy to offset energy bills and grid stress. Long-term, users can use this system to track how long it took for the solar panels to recoup their initial investment and the subsequent benefits.

In summary, our comprehensive energy monitoring and management system provides users with actionable insights for in-depth analysis and better decision-making. Together, these capabilities highlight the critical role of data-driven strategies in promoting both environmental sustainability and financial prudence in solar energy adoption.

5.0 Class Design

5.1 Class Diagram (Association, Aggregation, and Composition)

1. User (show user details)
2. Installment (calculate installment plan)
3. Energy (calculate energy produced, energy used)
4. Maintenance (monitor performance)

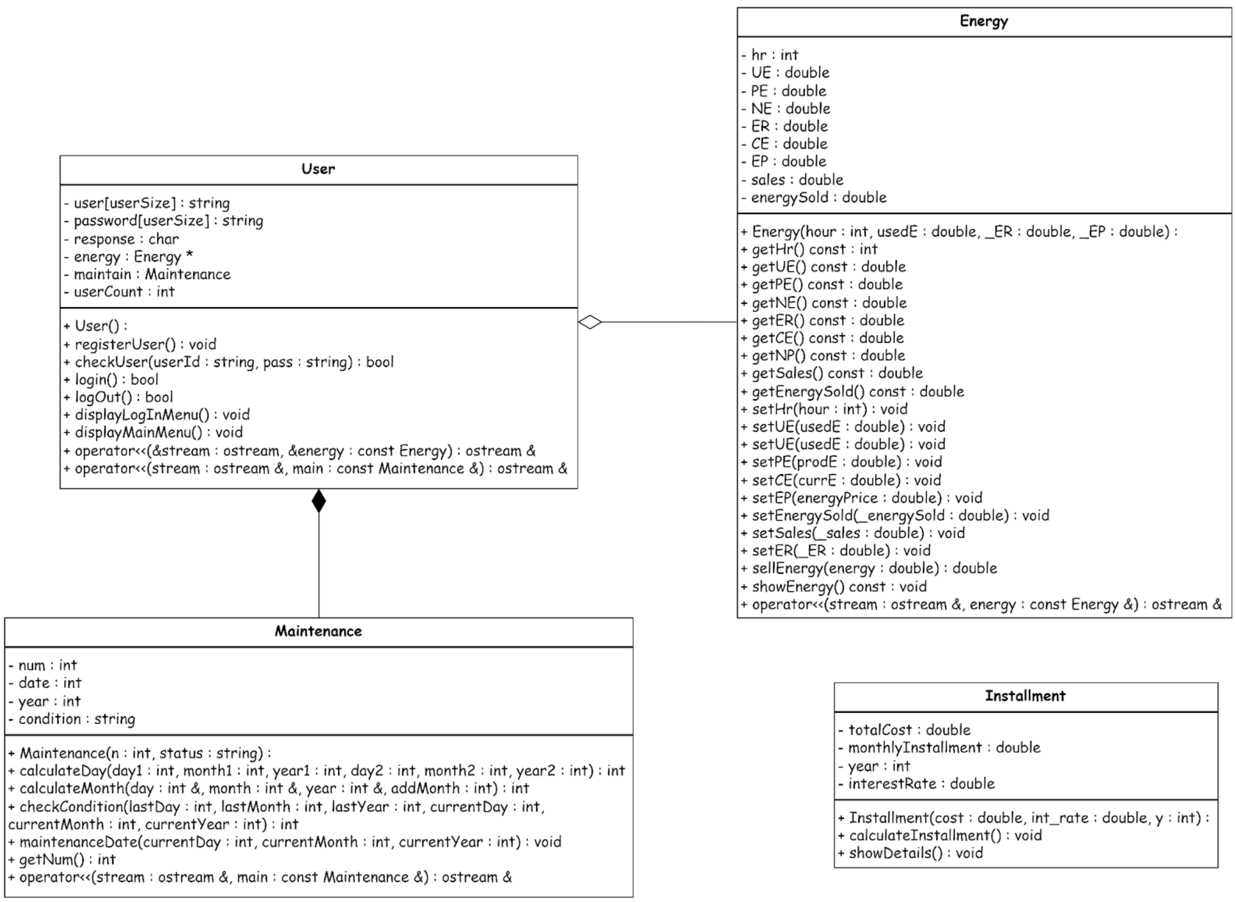

Figure 1: Class Diagram

```
#include <iostream>
#include <string>
using namespace std;

const int userSize = 5;

class Energy;
```

```
class Maintenance;

class User {
    private:
        string user[userSize];
        string password[userSize];
        char response;
        Energy *energy;
        Maintenance *maintain;
        int userCount;

    public:
        User() : userCount(0), energy(nullptr), maintain(nullptr) {}

        void registerUser() {
            if(userCount < userSize) {
                string userId, pass;
                cout << "Enter User Id: ";
                cin >> userId;
                cout << "Enter Password: ";
                cin >> pass;

                user[userCount] = userId;
                password[userCount] = pass;
                userCount++;
                cout << "User registered successfully!" << endl;
            }
            else {
                cout << "Number of Registered Users has reached max 
capacity" << endl;
            }
        }

        bool checkUser(string userId, string pass) {
            for(int i = 0; i < userCount; i++) {
                if((user[i] == userId) && (password[i] == pass)) {
                    return true;
                }
            }
            return false;
```

```
        }

        bool login() {
            string id, pass;
            cout << "User Id: ";
            cin >> id;
            cout << "Password: ";
            cin >> pass;

            if(checkUser(id, pass)) {
                cout << "Login successful!" << endl;
                return true;
            }
            else {
                cout << "Invalid id or password. Please enter again." <<
endl;
                return false;
            }
        }

        bool logOut() {
            cout << "Do you want to log out? (Y/N)" << endl;
            cin >> response;
            if(response == 'Y' || response == 'y') {
                cout << "Logout successful!" << endl;
                return true;
            }
            else if(response == 'N' || response == 'n') {
                return false;
            }
        }

        void displayLogInMenu() {
            cout << "\n===== Helios Solar Panel System =====" << endl;
            cout << "1. Register" << endl;
            cout << "2. Log In" << endl;
            cout << "3. Exit" << endl;
            cout << "=====================================" << endl;
        }
```

```
        void displayMainMenu() {
            cout << "\n===== Helios Solar Panel System =====" << endl;
            cout << "1. Solar Panel Information" << endl;
            cout << "2. Financial Tracking" << endl;
            cout << "3. Energy Tracking" << endl;
            cout << "4. Installment" << endl;
            cout << "5. Maintenance" << endl;
            cout << "6. Log Out" << endl;
            cout << "=====================================" << endl;
        }

        friend ostream &operator<<(ostream &stream, const Energy &energy);
        friend ostream &operator<<(ostream &stream, const Maintenance
&main);
};

class Installment {
    private:
        double totalCost;
        double monthlyInstallment;
        int year;
        double interestRate;

    public:
        Installment(double cost = 0, double int_rate = 0, int y = 0) :
totalCost(cost), interestRate(int_rate), year(y) {}

        void calculateInstallment() {
            cout << "\nInstallment Calculator" << endl;
            cout << "Price: ";
            cin >> totalCost;
            cout << "Number of Years: ";
            cin >> year;
            cout << "Interest Rate: ";
            cin >> interestRate;
            for(int i = 0; i < year; i++) { // compounded interest
                totalCost += totalCost * interestRate;
            }
            monthlyInstallment = totalCost / (year * 12.0);
        }
```

```
        void showDetails() {
            cout << "Total cost: RM" << totalCost << endl;
            cout << "Monthly Installment: RM" << monthlyInstallment <<
endl;
            cout << "Duration: " << year << " year(s)" << endl;
            cout << "Interest Rate: " << interestRate << endl;
        }
};

class Maintenance {
    private:
        int num, day, month, year;
        string condition;

    public:
        Maintenance(int n = 0, string status = "") : num(n),
condition(status), day(0), month(0), year(0) {}

        int calculateDay(int day1, int month1, int year1, int day2, int
month2, int year2) {
            int days1 = year1 * 365 + day1;
            for(int i = 0; i < month1 - 1; i++) {
                days1 += 30;
            }

            int days2 = year2 * 365 + day2;
            for(int i = 0; i < month2 - 1; i++) {
                days2 += 30;
            }
            return abs(days2 - days1);
        }

        void calculateMonth(int &day, int &month, int &year, int addMonth)
{
            month += addMonth;
            while(month > 12) {
                month -= 12;
                year += 1;
            }
```

```
        }

        void checkCondition(int lastDay, int lastMonth, int lastYear, int
currentDay, int currentMonth, int currentYear) {
            int days = calculateDay(lastDay, lastMonth, lastYear,
currentDay, currentMonth, currentYear);

            if(days > 0 && days <= 180) {
                condition = "green";
                cout << "Solar panel's health condition: green" << endl;
            }
            else if (days > 180 && days <= 365) {
                condition = "yellow";
                cout << "Solar panel's health condition: yellow" << endl;
            }
            else {
                condition = "red";
                cout << "Solar panel's health condition: red" << endl;
            }
        }

    void maintenanceDate(int currentDay, int currentMonth, int
currentYear) {
        if(condition == "green") {
            calculateMonth(currentDay, currentMonth, currentYear, 6);
            num++;
            cout << "Your new maintenance date is on " << (currentDay < 10
? "0" : "") << currentDay << "/" << (currentMonth < 10 ? "0" : "") <<
currentMonth << "/" << currentYear << endl;
        } else if(condition == "yellow") {
            calculateMonth(currentDay, currentMonth, currentYear, 3);
            num++;
            cout << "Your new maintenance date is on " << (currentDay < 10
? "0" : "") << currentDay << "/" << (currentMonth < 10 ? "0" : "") <<
currentMonth << "/" << currentYear << endl;
        } else if(condition == "red") {
            cout << "You should maintain the panel immediately." << endl;
        }
    }
```

```
    int getNum() const {
        return num;
    }

    friend ostream &operator<<(ostream &stream, const Maintenance &main) {
        stream << "Maintenance Summary\nMaintenance count: " << main.num
<< "\nCondition: " << main.condition;
        return stream;
    }
};

class Energy {
    private:
        int hr;
        double UE; // Amount of Energy Used (Energy Bought and/or
Produced)
        double PE; // Produced Energy
        double NE; // Net Energy = Energy Produced - Energy Used
        double ER; // Rate of Energy Produced kWh passed from Model type
        double CE; // Current Energy Levels of Energy Storage system
        double EP; // Energy Price
        double sales; // use for ROI
        double energySold; // use for energy savings

    public:
        Energy(int hour = 0, double usedE = 0, double _ER = 0, double _EP
= 0) : hr(hour), UE(usedE), ER(_ER), EP(_EP), CE(0.0), energySold(0.0),
sales(0.0) {}

        int getHr() const { return hr; }
        double getUE() const { return UE; }
        double getPE() const { return hr * ER; }
        double getNE() const { return getPE() - UE; }
        double getCE() const { return CE; }
        double getEP() const { return EP; }
        double getER() const { return ER; }
        double getSales() const { return sales; }
        double getEnergySold() const { return energySold; }

        void setHr(int hour) { hr = hour; }
```

```
        void setUE(double usedE) { UE = usedE; }
        void setPE(double prodE) { PE = prodE; }
        void setCE(double currE) { CE = currE; }
        void setEP(double energyPrice) { EP = energyPrice; }
        void setEnergySold(double _energySold) { energySold = _energySold;
}
        void setSales(double _sales) { sales = _sales; }
        void setER(double _ER) { ER = _ER; }

        double sellEnergy(double energy) {
            return energy * EP;
        }

        void showEnergy() const {
            cout << "Current Energy Levels: " << CE << " kWh" << endl;
            cout << "Energy Used: " << UE << " kWh" << endl;
            cout << "Energy Rate: " << ER << " kWh" << endl;
            cout << "Produced Energy: " << getPE() << " kWh" << endl;
            cout << "Net Energy: " << getNE() << " kWh" << endl;
        }

        friend ostream &operator<<(ostream &stream, const Energy &energy)
{
            stream << "Energy Summary\nEnergy Produced: " <<
energy.getPE() << "\nEnergy Used: " << energy.getUE() << "\nNet Energy: "
<< energy.getNE();
            return stream;
        }
};
```

Figure 2: C++ codes based on class diagram

Figure 1 depicts four classes: user, installation, maintenance, and energy. Class User is used to obtain the user's information, such as their user ID and password, so that they can log into the solar panel system. Next, the Installment class is implemented to assist users in calculating the solar panel system's installment plan. They can tailor the plan to their specific needs, which will help them understand the practical and financial requirements of purchasing a solar system and managing the high costs associated with it. Furthermore, the goal of implementing class

Maintenance is to monitor the health of the solar panel system and notify users if there are any issues that require repair. Additionally, class Energy functions as a calculator for calculating energy used, energy produced, net energy, and rate of energy. Last but not least, figure 2 shows the C++ codes based on the figure 5.1 class diagram.

5.2 Extended Class Diagram

1. SolarPanel (information of solar panel)
2. ModelA (different version of solar panel)
3. ModelB (different version of solar panel)

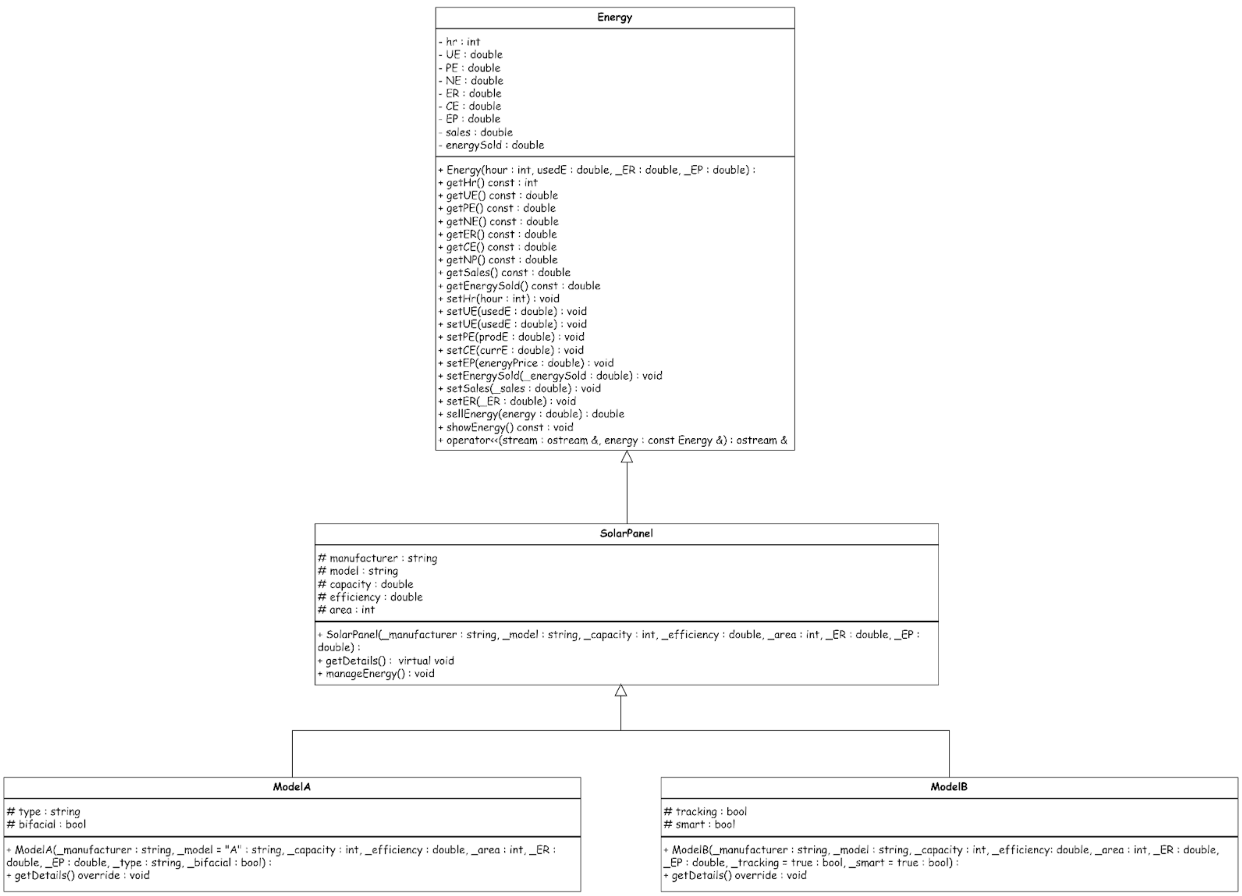

Figure 3: Extended class diagram

```
class SolarPanel : public Energy {
    protected:
        string manufacturer;
        string model;
```

```
        double capacity; // kW
        double efficiency; // %
        int area; // m2

    public:
        SolarPanel(string _manufacturer = "Helios", string _model = "",
double _capacity = 0, double _efficiency = 0.0, int _area = 0, double _ER
= 0.0, double _EP = 0.0) : Energy(0, 0, _ER, _EP),
manufacturer(_manufacturer), model(_model), capacity(_capacity),
efficiency(_efficiency), area(_area) {}

        virtual void getDetails() {
            cout << "Manufacturer: " << manufacturer << endl;
            cout << "Model: " << model << endl;
            cout << "Capacity: " << capacity << " kW" << endl;
            cout << "Efficiency: " << efficiency << " %" << endl;
            cout << "Area: " << area << " sq. meters" << endl;
        }

        void manageEnergy() {
            for(int i = 1; i <= getHr(); i++) {
                setCE(getCE() + (1 * getER()) * efficiency);
                if (getCE() > capacity) {
                    double excessEnergy = getCE() - capacity;
                    setCE(capacity);
                    setEnergySold(getEnergySold() + excessEnergy);
                    setSales(getSales() + sellEnergy(excessEnergy));
                }
                cout << endl;
                cout << "#" << i << " hour: " << endl;
                cout << "Current Energy Level = " << getCE() << endl;
                cout << "Total Energy Sold = " << getEnergySold() << " kW"
<< endl;
                cout << "Total Sales = RM" << getSales() << endl;
            }
        }
};

class ModelA: public SolarPanel {
    protected:
```

```
        string type; // monocrytalline +15% efficiency, polycrystalline
+5% efficiency
        bool bifacial; // true = bifacial(twosided), false = not bifacial

    public:
        ModelA(string _manufacturer = "Helios", string _model = "A",
double _capacity = 300, double _efficiency = 25,
        int _area = 25, double _ER = 2.0, double _EP = 0.1, string _type =
"mono", bool _bifacial = true)
        : SolarPanel(_manufacturer, _model, _capacity, _efficiency + 15,
_area, _ER, _EP), type(_type), bifacial(_bifacial) {}

        void getDetails() override {
            SolarPanel::getDetails();
            cout << "Type: " << type << endl;
            cout << "Bifacial: " << (bifacial ? "Yes" : "No") << endl;
        }
};

class ModelB: public SolarPanel {
    protected:
        bool tracking; // allows panels to track the sun
        bool smart; // power optimizers and more accurate monitoring
capability

    public:
        ModelB(string _manufacturer = "Helios", string _model = "B",
double _capacity = 500, double _efficiency = 35,
        int _area = 15, double _ER = 5.0, double _EP = 2.0, bool _tracking
= true, bool _smart = true)
        : SolarPanel(_manufacturer, _model, _capacity, _efficiency + 15,
_area, _ER, _EP), tracking(_tracking), smart(_smart) {}

        void getDetails() override {
            SolarPanel::getDetails();
            cout << "Tracking: " << (tracking ? "Yes" : "No") << endl;
            cout << "Smart: " << (smart ? "Yes" : "No") << endl;
        }
};
```

Figure 4: C++ codes based on extended class diagram

In Figure 3, we created an extended class diagram that makes use of inheritance relationships between class diagrams. First, we implemented the SolarPanel class, which is the primary function of the solar panel system. It includes the basic details and functions of a solar panel, such as the time of manufacture, model, capacity, efficiency, area, and production rate. In this class, we also used some methods to learn about solar panels. In addition, we created two classes, ModelA and ModelB, which inherit from the SolarPanel class. They inherit the old attributes and methods from the SolarPanel class while adding new attributes and methods to create a unique version of the solar panel. For example, ModelA uses monocrystalline solar cells and is double-sized, with lower capacity, lower efficiency, larger area, and lower production rate than ModelB. Meanwhile, ModelB has a smart AI system to track solar energy, with greater capacity, efficiency, smaller area, and higher production rate than ModelA. Both models have unique characteristics that can entice buyers to purchase them.

6.0 Benefit and Summary of the Proposed System

The benefit of the proposed system is to make it accessible and practical for the homeowners. The main feature is the installment plan option, which helps to manage the high upfront costs by creating personalized payment plans. So, this can make it easier for more people to afford and invest in solar energy, breaking down financial barriers that might otherwise prevent them from making this environmentally-friendly choice.

Moreover, the benefit of the system is the system's ability to track return on investment and cost savings. By monitoring the energy savings and financial benefit in real time, users can see how much money they are saving and assess the effectiveness of their solar panels. This transparency empowers users to make informed decisions about their energy usage and investment, ensuring they get the most out of their solar panels.

Other than that, the system includes advanced energy management features that optimize the use of solar energy. It prioritizes solar energy during high demand periods and sells excess energy back to the grid when demand is low, reducing waste and maximizing efficiency. In addition, regular maintenance and monitoring ensure that the solar panels remain in good condition, minimizing downtime and extending their lifespan. In conclusion, this system promotes environmental sustainability by reducing reliance on non-renewable energy sources while offering significant financial benefits to users.

7.0 Link to recorded Video Demo (Public Access)

https://drive.google.com/file/d/12_pfBPjIrTHFO3IFAB1V0tV2cw_dR8FH/view?usp=sharing

8.0 Github Team Working Assessment

https://github.com/Jianming03/Project_SECJ1023