UOW MALAYSIA
KDU PENANG
UNIVERSITY COLLEGE

PART OF THE UNIVERSITY OF WOLLONGONG AUSTRALIA GLOBAL NETWORK

ASSIGNMENT COVER PAGE

Programme	**Course Code and Title**
Diploma in Information Technology	DOP1254 Fundamentals of Object Oriented Programming
Student's name / student's id	**Lecturer's name**
ADRIAN TAN YEE HORNG 0205183 NAVEENRAAJ A/L P THINARTHAN 0205034 OOI ZI XUAN 0204951	Tan Phit Huan

Date issued	**Submission Deadline**	**Indicative Weighting**
Week 6 -03/05/2021	Week 11 – 09/04/2021	20%
Assignment 2 title	Product management system	

This assessment assesses the following course learning outcomes

# as in Course Guide	UOWM KDU Penang University College Learning Outcome
CLO2	Apply modularization and array in programming.
CLO4	Apply object oriented programming concepts in software development.

Student's declaration

I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

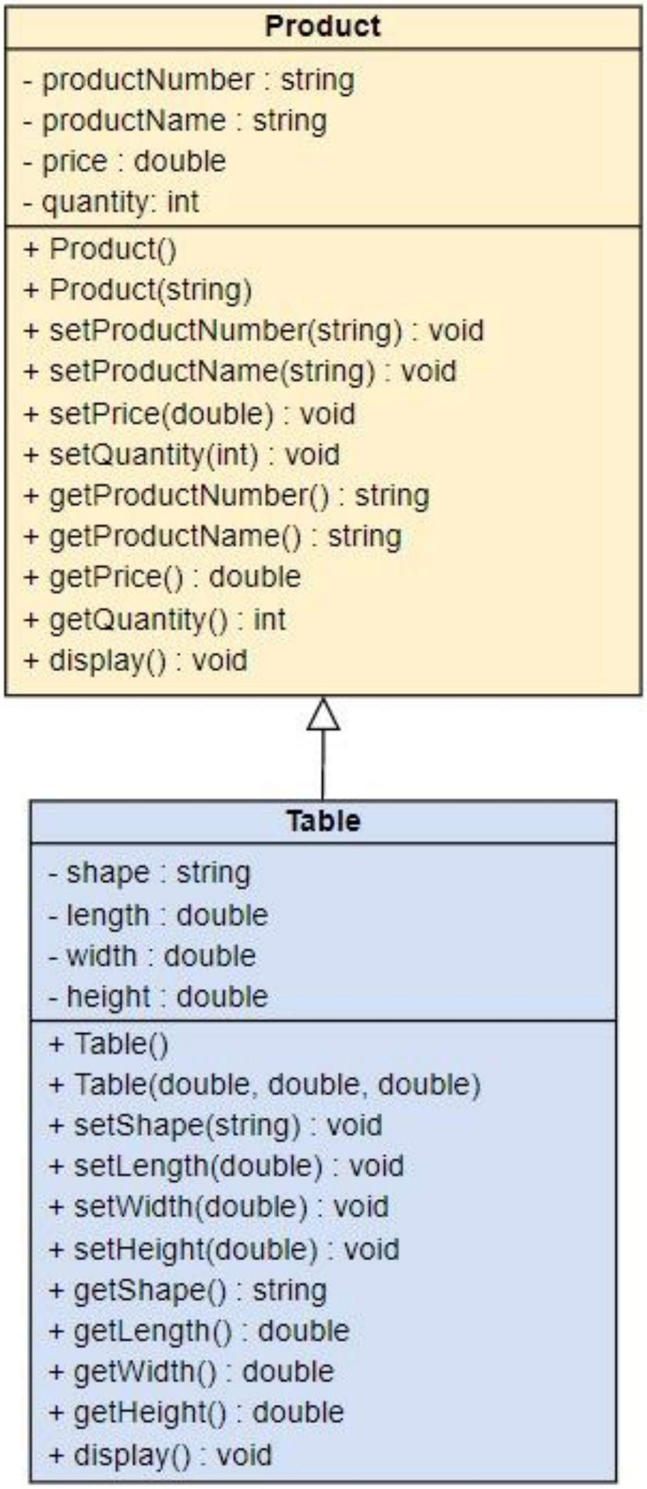
Student's signature: ADRIAN TAN YEE HORNG

Submission Date: 08-04-2021

TABLE OF CONTENTS

Task 1

Class Diagram



Task 2

Complete Program

Product.h

```
#ifndef product_h
#define product_h
#include <iostream>
#include <iomanip>
using namespace std;

class Product{
    private:
        //declare variables
        string productNumber, productName;
        int quantity;
        double price;
    public:
        //default constructor
        Product(){
            productNumber = "-1";
            productName = "Name Unknown";
            price = 0;
            quantity = 0;
        }
        //parameterized constructor
        Product(string name){
            productNumber = "-1";
            productName = name;
            price = 0;
            quantity = 0;
}

//setter
void setProductNumber(string code){
	productNumber = code;
}
void setProductName(string name){
	productName = name;
}
void setPrice(double rrp){
	price = rrp;
}
void setQuantity(int qty){
	quantity = qty;
}

//getter
string getProductNumber(){
	return productNumber;
}
string getProductName(){
	return productName;
}
double getPrice(){
	return price;
}
int getQuantity(){
	return quantity;
}
```

```
        //query
        //set the suitable decimal places
        void display(){
            cout << "Product Number: " << productNumber << endl
                << "Product Name: " << productName << endl
                << fixed << showpoint << setprecision(2)
                << "Price: " << price << endl
                << "Quantity: " << quantity << endl
                << fixed << showpoint << setprecision(3);
        }
};
#endif
```

<u>Table.h</u>

```
#ifndef table_h
#define table_h
#include "Product.h"
#include <iostream>
using namespace std;

class Table : public Product{
        private:
                //declare variables
                string shape;
                double length, width, height;
        public:
                //default constructor
                Table() : Product(){
                        shape = "Shape Unknown";
                        length = 1.0;
                        width = 1.0;
                        height = 1.0;
                }
                //parameterized constructor
                Table(double le, double wi, double he) : Product(){
                        length = le;
                        width = wi;
                        height = he;
                }

                //setter
                void setShape(string sh){
                        shape = sh;
```

```
}
void setLength(double le){
    length = le;
}
void setWidth(double wi){
    width = wi;
}
void setHeight(double he){
    height = he;
}

//getter
string getShape(){
    return shape;
}
double getLength(){
    return length;
}
double getWidth(){
    return width;
}
double getHeight(){
    return height;
}

//query
void display(){
    Product::display();
    cout << "Shape: " << shape << endl
        << "Length: " << length << endl
```

```
                << "Width: " << width << endl
                << "Height: " << height << endl;
        }
};
#endif
```

Function.h

```
#ifndef function_h
#define function_h
#include <iostream>
using namespace std;

//functions are arranged in alphabetical order
void divider(){
	//act as divider
	cout << "=================================================" << endl;
}

void error_emptyRecord(){
	//display error message for no Table record stored
	cout << endl << "No Table record stored. " << endl;
}

void feature1_measurementHeight(){
	//ask the user to input height
	cout << "Height: ";
}

void feature1_measurementLength(){
	//ask the user to input length
	cout << endl
		<< "Kindly input: " << endl
		<< "Length: ";
}

void feature1_measurementsAsk(){
    //ask the user whether wants to input the measurements or not
    cout << endl
        << "Do you have the length, width, and height for the Table? " << endl
        << "If yes, kindly reply '1'. Otherwise, reply any other integer value. " << endl
        << "Reply: ";
}

void feature1_measurementWidth(){
    //ask the user to input width
    cout << "Width: ";
}

void feature1_productName(){
    //ask the user to input product name
    cout << endl << "Kindly input Product Name: ";
}

void feature1_productNameAsk(){
    //ask the user whether wants to input the product name or not
    cout << endl
        << "Do you have the Product Name? " << endl
        << "If yes, kindly reply '1'. Otherwise, reply any other integer value. " << endl
        << "Reply: ";
}

void feature1_productNumberAsk(){
    //ask the user to input product number
    cout << "Kindly input Product Number: ";
}
```

```
void feature1_productPrice(){
	//ask the user to input price
	cout << endl
		<< "Kindly input: " << endl
		<< "Price: ";
}

void feature1_productPriceAndQuantityAsk(){
	//ask the user whether wants to input the price and quantity or not
	cout << endl
		<< "Do you have the Price and Quantity for the Table? " << endl
		<< "If yes, kindly reply '1'. Otherwise, reply any other integer value. " << endl
		<< "Reply: ";
}

void feature1_productQuantity(){
	//ask the user to input quantity
	cout << "Quantity: ";
}

void feature1_success(){
	//display message for a new Table record created
	cout << endl << "New Table record created. " << endl;
}

void feature1_tableShape(){
	//ask the user to input shape
	cout << endl
		<< "Kindly input: " << endl
		<< "Shape: ";
}

void feature1_tableShapeAsk(){
	//ask the user whether wants to input the shape or not
	cout << endl
		<< "Do you have the Shape for the Table? " << endl
		<< "If yes, kindly reply '1'. Otherwise, reply any other integer value. " << endl
		<< "Reply: ";
}

void feature1_title(){
	//title for feature 1
	cout << "NEW TABLE RECORD";
}

void feature2_success(){
	//display message for the new measurements set successfully
	cout << endl << "Length, width, and height are set successfully. " << endl;
}

void feature2_title(){
	//title for feature 2
	cout << "CHANGE THE MEASUREMENTS OF A TABLE RECORD STORED";
}

void feature3_title(){
	//title for feature 3
	cout << "PRINT DETAILS OF A TABLE RECORD STORED";
}
```

```
void feature4_title(){
    //title for feature 4
    cout << "PRINT DETAILS OF ALL TABLE RECORDS STORED";
}

void featureExit(){
    //dislay message for terminating the program
    cout << "No feature selected. Program terminated. " << endl
        << "Thank you for using. Have a nice day. " << endl;
}

void featuresAsk(){
    //ask the user to choose the features
    cout << "Kindly input " << endl
        << "'1' to create a new Table record. " << endl
        << "'2' to change the measurement(s) of a Table record stored using Product Number. " << endl
        << "'3' to print the details of a Table record stored using Product Number. " << endl
        << "'4' to print the details of all Table records stored. " << endl
        << "'0' to terminate the program. " << endl
        << "Input: ";
}

void productNumber_ask(){
    //ask the user to input product number for matching process
    cout << endl << "Kindly give a Product Number: ";
}

void productNumber_invalidError(){
        //display error message for no Table record stored found
        cout << "No Table record found based on the Product Number! " << endl;
}

void productNumber_repeatError(){
        //display error message for repeated product number
        cout << endl
                << "Existing product number is entered! Please create an unique product number. " << endl
                << "Do you want to continue creating a new Table record? " << endl
                << "If yes, kindly reply '1'. Otherwise, reply any other integer value. " << endl
                << "Reply: ";
}

void welcome(){
        //display message to welcome the user for using the program
        cout << "Welcome to Product Management System (PMS). You can create up to 100 Table records. " << endl;
}
#endif
```

Driver.cpp

```
#include "Table.h"
#include "Function.h"
#include <iostream>
using namespace std;

int main(){
    //declare and initialize variables
    //declare array table
    const int RECORD = 100;
    Table table[RECORD];
    int counter = 0, matcher, feature, yesORno, yesORno2 = 1, qty;
    string name, code, number, sh;
    double rrp, le, wi, he;

    //call welcome()
    welcome();

    //allow the user to use the program for unlimited times until feature 0 is chosen
    do{
        //call featuresAsk()
        //set the input as feature
        featuresAsk();
        cin >> feature;

        //clear the screen for a less distracting view
        system("CLS");

        //if the user chooses to create a new Table record (feature 1)
        if(feature == 1){
```

```
//call feature1_title()
feature1_title();

//if the total Table record stored hasn't reached the limit
if(counter < RECORD){
    //call feature1_productNameAsk()
    //set the input as yesORno
    feature1_productNameAsk();
    cin >> yesORno;

    //if the user chooses to enter the product name
    if(yesORno == 1){
        //call feature1_productName()
        //ignore previous input(s)
        //set the whole input as name
        feature1_productName();
        cin.ignore(256, '\n');
        getline(cin, name);
    }
    //if the user chooses not to enter the product name
    else{
        //set the name to "Name Unknown"
        name = "Name Unknown";

        //to make it neat
        cout << endl;
    }

    //call feature1_productNumberAsk()
    //set the input as code
```

```
                feature1_productNumberAsk();
                cin >> code;

                //for loop to access each product number in the array until it reaches
the counter
                for(matcher = 0; matcher < counter; matcher++){
                    //set the product number from the current array as number
for comparision
                    number = table[matcher].getProductNumber();

                    //if the existing product number is found
                    if(code == number){
                        //call productNumber_repeatError()
                        //set the input as yesORno2
                        productNumber_repeatError();
                        cin >> yesORno2;

                        //to make it neat
                        cout << endl;

                        //if the user chooses to reenter the product number
                        if(yesORno2 == 1){
                            //call feature1_productNumberAsk()
                            //set the input as code
                            feature1_productNumberAsk();
                            cin >> code;

                            //reset the matcher to -1 to let the system
chack again the validity of the new product number
                            matcher = -1;
                        }
                        //if the user chooses not to reenter the product number and exit feature 1
                        else{
                            //exit feature 1
                            break;
                        }
                    }
                }

                //skip the functions below if the user chooses to exit feature 1 in previous for loop
                if(yesORno2 == 1){
                    //call feature1_productPriceAndQuantityAsk()
                    //set the input as yesORno
                    feature1_productPriceAndQuantityAsk();
                    cin >> yesORno;

                    //if the user chooses to enter the product price
                    if(yesORno == 1){
                        //call feature1_productPrice() and feature1_productQuantity()
                        //set the inputs as rrp and qty
                        feature1_productPrice();
                        cin >> rrp;
                        feature1_productQuantity();
                        cin >> qty;
                    }

                    //call feature1_tableShapeAsk()
                    //set the input as yesORno
                    feature1_tableShapeAsk();
```

```
cin >> yesORno;

//if the user chooses to enter the shape
if(yesORno == 1){
    //call feature1_tableShape()
    //set the input as sh
    feature1_tableShape();
    cin >> sh;
}
//if the user chooses not to enter the shape
else{
    //set the sh to "Shape Unknown"
    sh = "Shape Unknown";
}

//call feature1_measurementsAsk()
//set the input as yesORno
feature1_measurementsAsk();
cin >> yesORno;

//if the user chooses to enter the measurements
if(yesORno == 1){
    //call feature1_measurementLength(), feature1_measurementWidth(), and feature1_measurementHeight()
    //set the inputs as le, wi, and he
    feature1_measurementLength();
    cin >> le;
    feature1_measurementWidth();
    cin >> wi;
    feature1_measurementHeight();
```

```
                        cin >> he;

                        //create new Table record with length, width, and height

                        table[counter] = Table(le, wi, he);
                    }
                    //if the user chooses not to enter the measurements
                    else{
                        //create new Table record with default constructor
                        table[counter] = Table();
                    }

                    //store the variables in corresponding attributes in the class
                    table[counter].setProductName(name);
                    table[counter].setProductNumber(code);
                    table[counter].setPrice(rrp);
                    table[counter].setQuantity(qty);
                    table[counter].setShape(sh);

                    //increase the counter by 1
                    counter++;

                    //call feature1_success()
                    feature1_success();

                    //
                    //if there is only 1 Table record stored
                    if(counter == 1)
                        cout << "Total of " << counter << " Table is created successfully. " << endl;
                    //if there are more than 1 Table record stored
                    else
                        cout << "Total of " << counter << " Tables are
created successfully. " << endl;
                }

                //reset yesORno2 to 1
                yesORno2 = 1;
            }
            //if the total Table record stored has reached the limit
            else{
                //display error message for reaching the limit of Table record stored
                cout << endl << "You have reached the limit of your Table record
stored! You can only create " << RECORD << " Table records. "
                     << endl << endl;
            }

            //pause the system to let the user views the error message
            //then clear the screen for a less distracting view
            system("pause");
            system("CLS");
        }

        //if the user chooses to change the measurements of a Table record stored (feature
2)
        else if(feature == 2){
            //call feature2_title()
            feature2_title();

            //if there is no Table record stored
            if(counter == 0){
                //to make it neat
                cout << endl;

                //call error_emptyRecord()
                error_emptyRecord();

                //pause the system to let the user views the error message
                system("pause");
            }
            //if there is(are) Table record(s) stored
            else{
                //call productNumber_ask()
                //set the input as code
                productNumber_ask();
                cin >> code;

                //to make it neat
                cout << endl;

                //for loop to access each product number in the array until it reaches the counter
                for(matcher = 0; matcher < counter; matcher++){
                    //set the product number from the current array as number for comparision
                    number = table[matcher].getProductNumber();

                    //if the existing product number is found
                    if(code == number){
                        //ask for measurements
                        //set the inputs as le, wi, and he
```

```
        //store them in corresponding measurements in the class
        cout << "Set the length for " << name << ": ";
        cin >> le;
        table[matcher].setLength(le);
        cout << "Set the width  for " << name << ": ";
        cin >> wi;
        table[matcher].setWidth(wi);
        cout << "Set the height for " << name << ": ";
        cin >> he;
        table[matcher].setHeight(he);

        //call feature2_success()
        feature2_success();

        //pause the system to let the user views the error message
        system("pause");

        //exit for loop
        break;
}

//if the matcher reaches the last counter and no matched product number is found
if(matcher == counter - 1){
        //call productNumber_invalidError()
        productNumber_invalidError();

        //pause the system to let the user views the error message
```

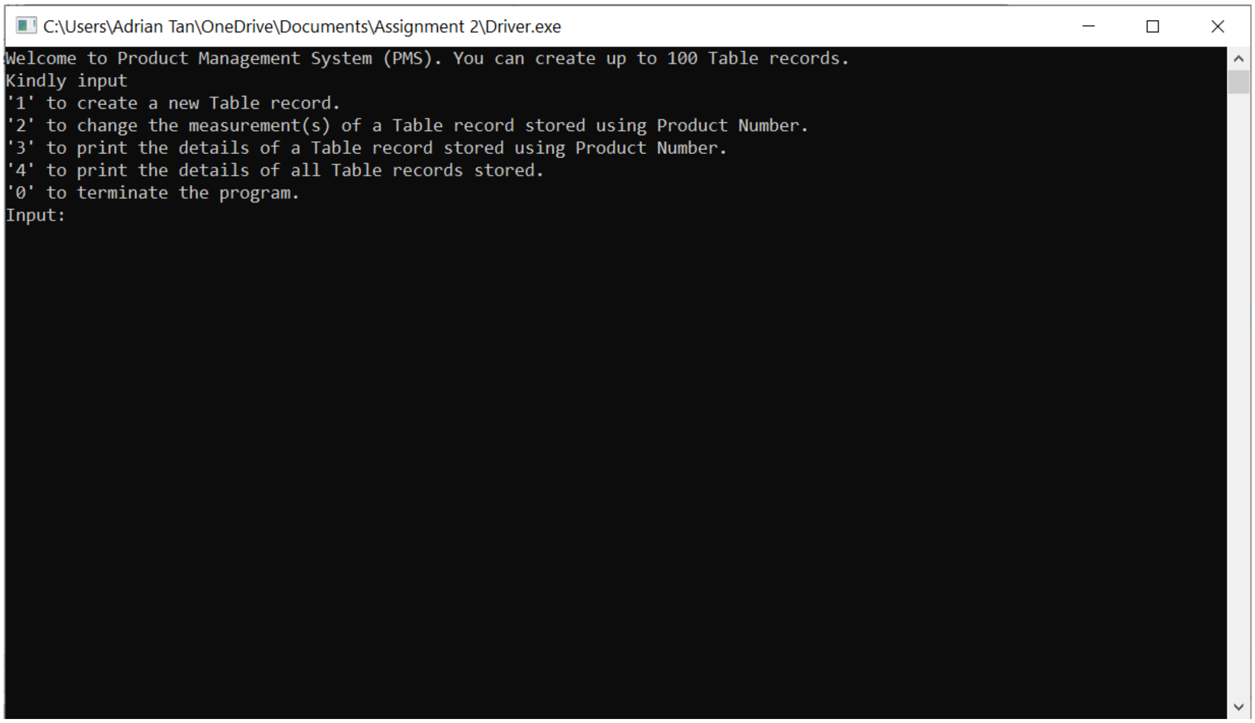
```
                system("pause");
            }
        }
    }

    //clear the screen for a less distracting view
    system("CLS");
}

//if the user chooses to print the details of a Table record stored (feature 3)
else if(feature == 3){
    //call feature3_title()
    feature3_title();

    //if there is no Table record stored
    if(counter == 0){
        //to make it neat
        cout << endl;

        //call error_emptyRecord()
        error_emptyRecord();

        //pause the system to let the user views the error message
        system("pause");
    }
    //if there is(are) Table record(s) stored
    else{
        //call productNumber_ask()
        //set the input as code
        productNumber_ask();
```

```
                cin >> code;

                //to make it neat
                cout << endl;

                //for loop to access each product number in the array until it reaches
the counter
                for(matcher = 0; matcher < counter; matcher++){
                    //set the product number from the current array as number
for comparision
                    number = table[matcher].getProductNumber();

                    //if the existing product number is found
                    if(code == number){
                        //display output
                        cout << endl << "Result " << endl;
                        table[matcher].display();
                        cout << endl;

                        //pause the system to let the user views the error
message
                        system("pause");

                        //exit for loop
                        break;
                    }

                    //if the matcher reaches the last counter and no matched
product number is found
                    if(matcher == counter - 1){
                        //call productNumber_invalidError()
                productNumber_invalidError();

                //pause the system to let the user views the error message
                system("pause");
            }
        }
    }

    //clear the screen for a less distracting view
    system("CLS");
}

//if the user chooses to print the details of all Table record stored(feature 4)
else if(feature == 4){
    //call feature4_title()
    feature4_title();

    //to make it neat
    cout << endl;

    //if there is(are) Table record(s) stored
    if(counter != 0){
        for(int z = 0; z < counter; z++){
            //display output
            divider();
            cout << "Table " << z + 1 << ". " << endl;
            table[z].display();
            divider();
            cout << endl;
```

```
                }
            }
            //if there is no Table record stored
            else{
                //call error_emptyRecord()
                error_emptyRecord();
            }

            //pause the system to let the user views the error message
            system("pause");

            //clear the screen for a less distracting view
            system("CLS");
        }

        //if the user chooses to terminate the program(feature 0)
        else if(feature == 0){
            //call featureExit()
            featureExit();
        }
    }while(feature != 0);

    return 0;
}
```

Description of programs

Main Page



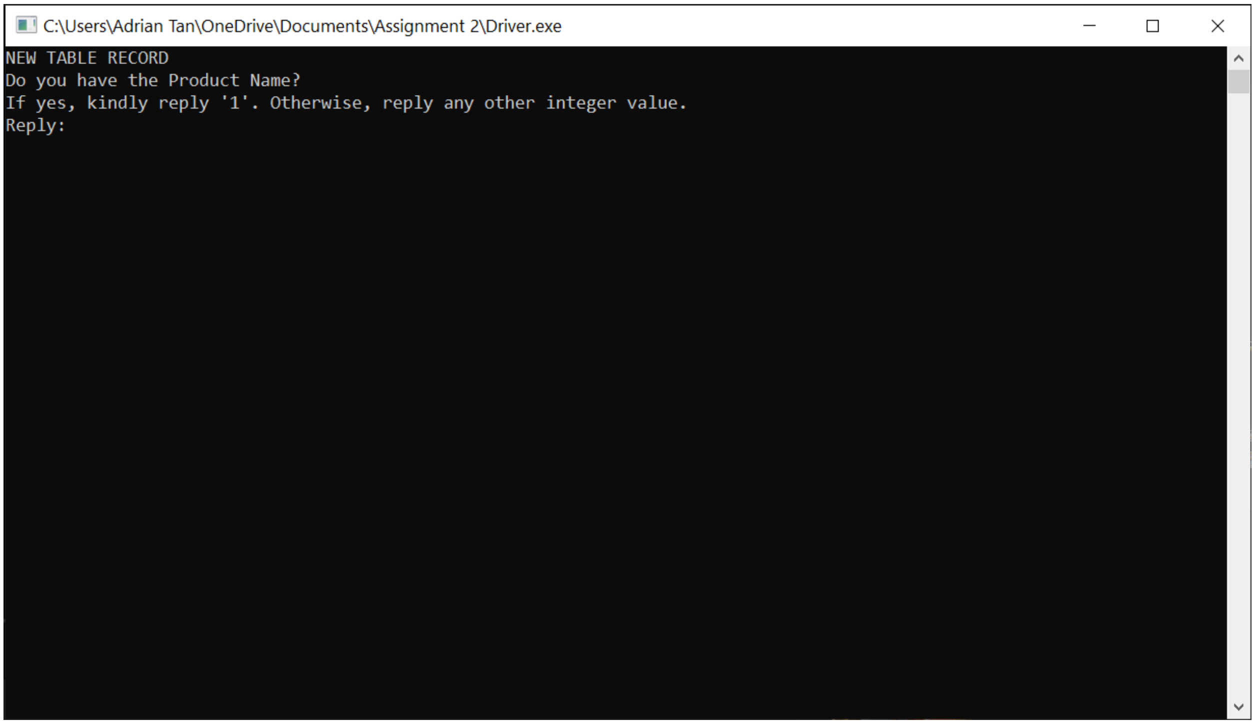
C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
Welcome to Product Management System (PMS). You can create up to 100 Table records.
Kindly input
'1' to create a new Table record.
'2' to change the measurement(s) of a Table record stored using Product Number.
'3' to print the details of a Table record stored using Product Number.
'4' to print the details of all Table records stored.
'0' to terminate the program.
Input:
```

Figure 1

Firstly, the program will welcome the user to use the Product Management System (PMS). The user will be informed that he/she can only create up to 100 Table records. Next, the system will ask the user to choose the features as shown in Figure 1.

Feature 1 (New Table Record)

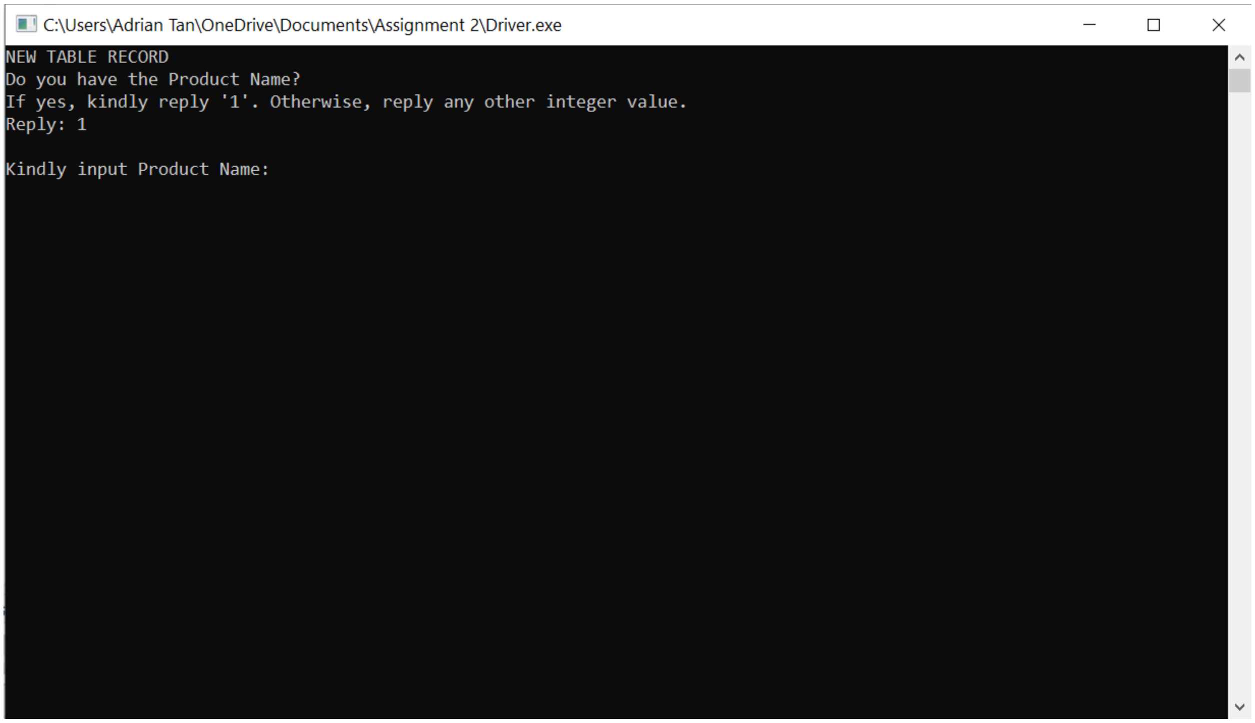


C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
NEW TABLE RECORD
Do you have the Product Name?
If yes, kindly reply '1'. Otherwise, reply any other integer value.
Reply:
```

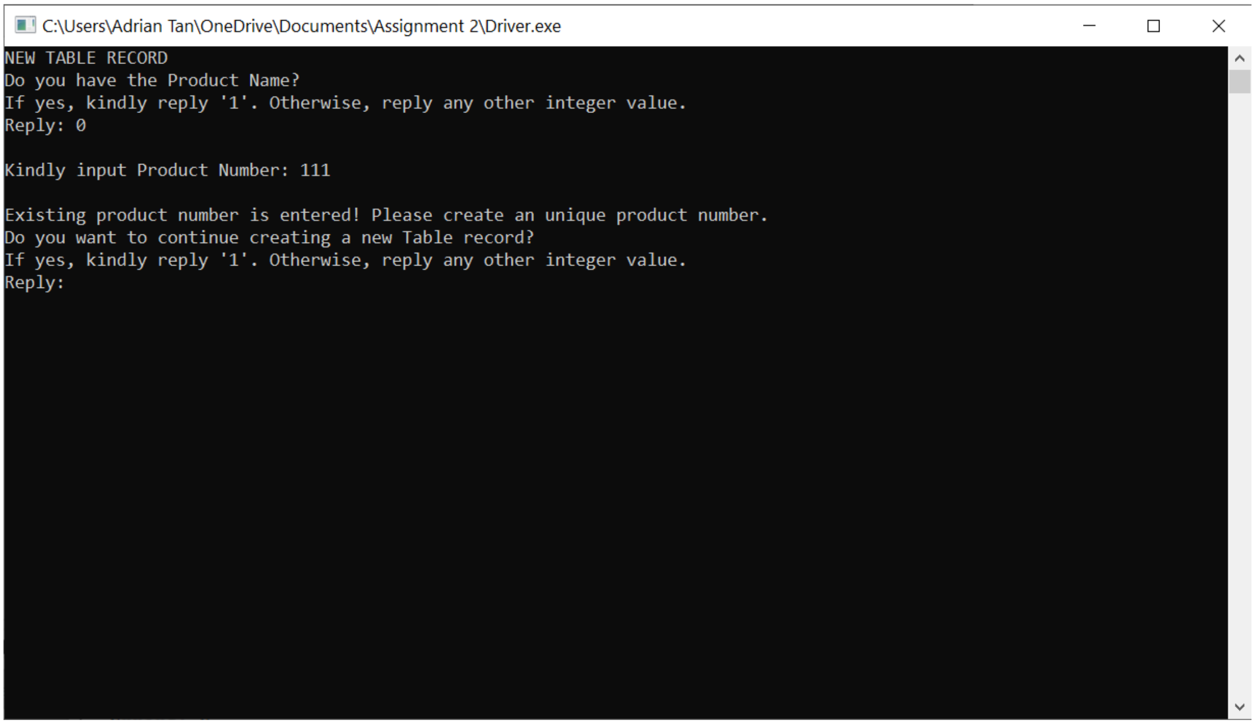
Figure 2.1

If the user enters '1', the user will be led to the New Table Record page. The system will ask the user whether the Table has a product name or not. If the user enters '1' which indicates the Table has a product number, the system will ask the user to enter the product name as shown in Figure 2.2. Then, the user will be asked for the product number. If the user enters any other integer besides '1', The system will straight ask the user to enter the product number.


C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
NEW TABLE RECORD
Do you have the Product Name?
If yes, kindly reply '1'. Otherwise, reply any other integer value.
Reply: 1

Kindly input Product Name:
```

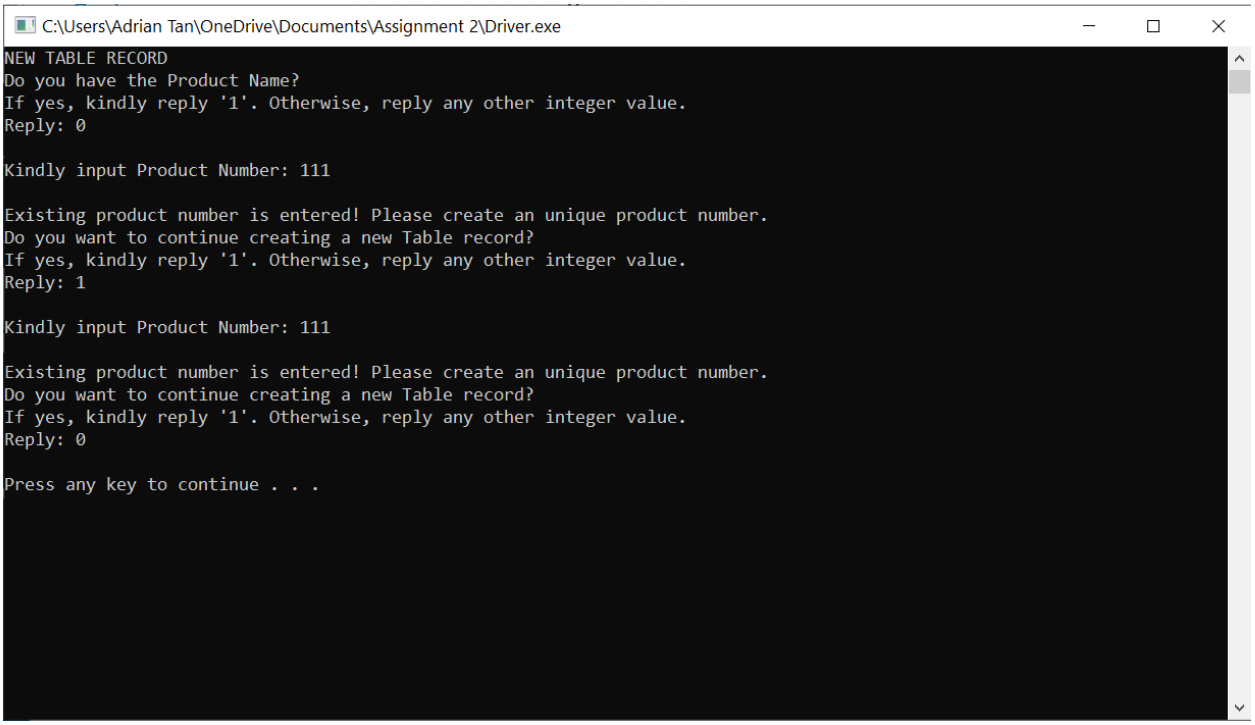
Figure 2.2



C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
NEW TABLE RECORD
Do you have the Product Name?
If yes, kindly reply '1'. Otherwise, reply any other integer value.
Reply: 0

Kindly input Product Number: 111

Existing product number is entered! Please create an unique product number.
Do you want to continue creating a new Table record?
If yes, kindly reply '1'. Otherwise, reply any other integer value.
Reply:
```

Figure 2.3

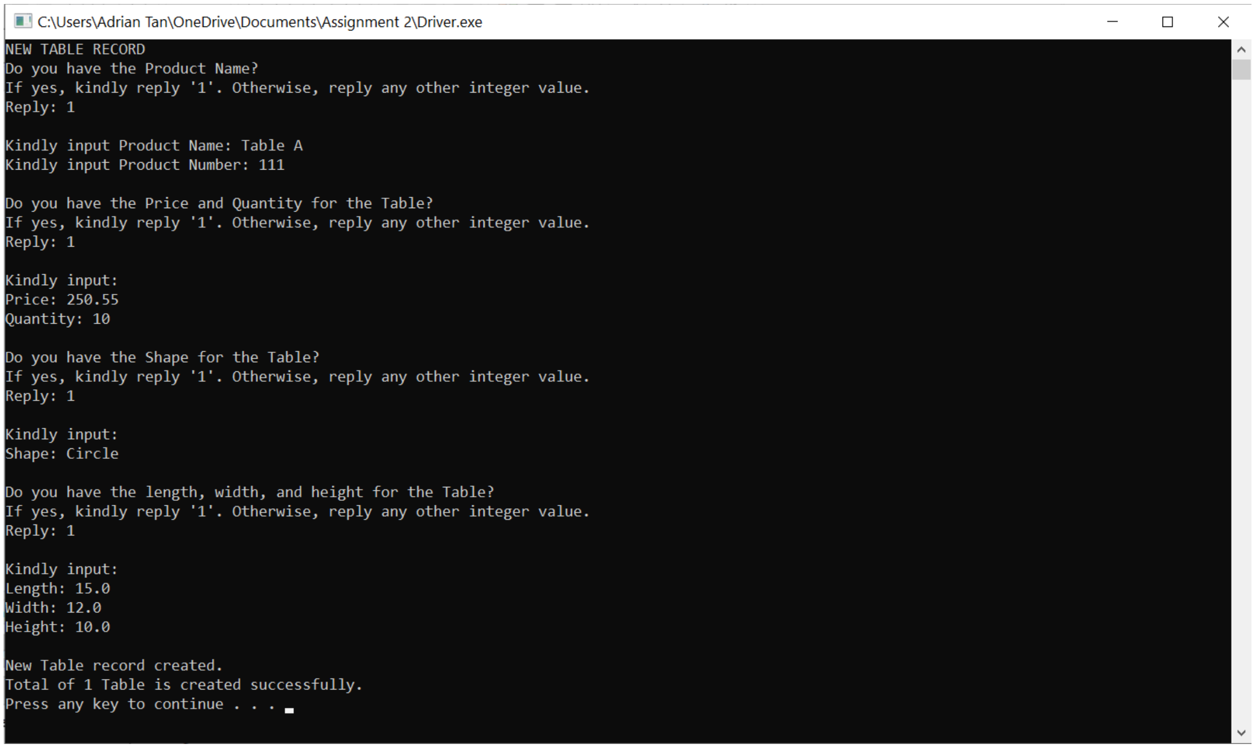


C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
NEW TABLE RECORD
Do you have the Product Name?
If yes, kindly reply '1'. Otherwise, reply any other integer value.
Reply: 0

Kindly input Product Number: 111

Existing product number is entered! Please create an unique product number.
Do you want to continue creating a new Table record?
If yes, kindly reply '1'. Otherwise, reply any other integer value.
Reply: 1

Kindly input Product Number: 111

Existing product number is entered! Please create an unique product number.
Do you want to continue creating a new Table record?
If yes, kindly reply '1'. Otherwise, reply any other integer value.
Reply: 0

Press any key to continue . . .
```

Figure 2.4

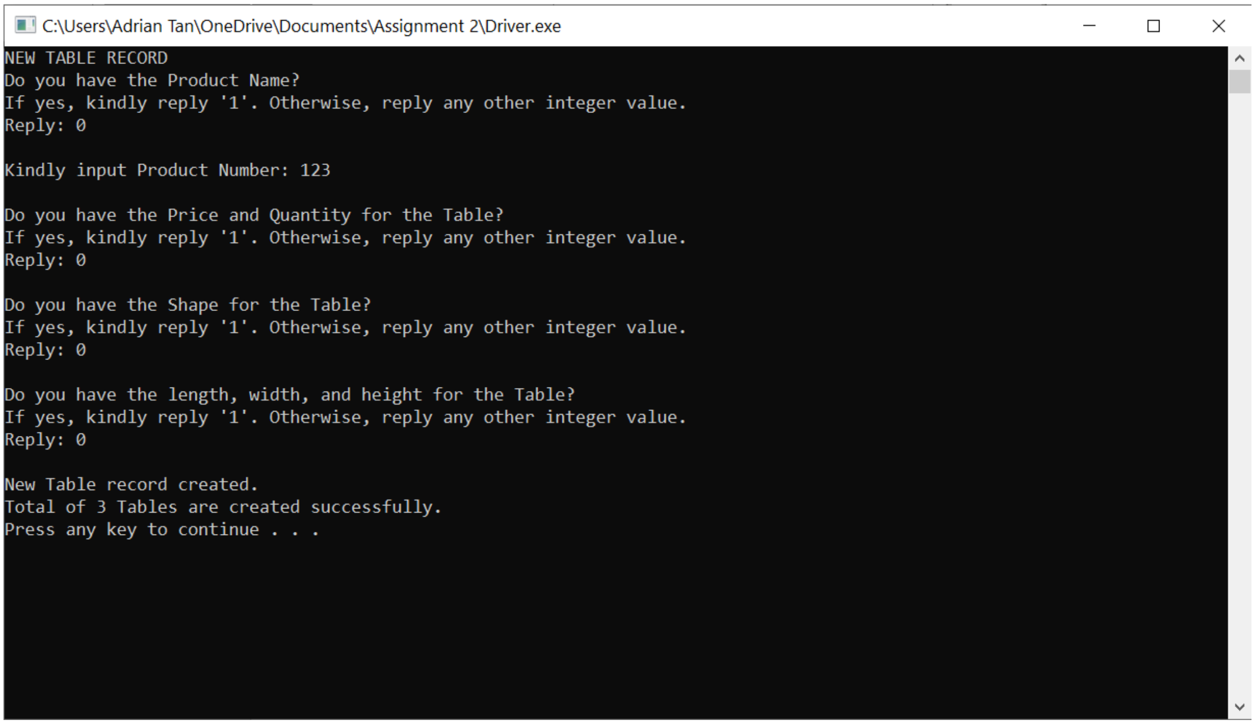
If the product number entered is found repeated, the system will show an error message and ask the user to choose whether reenter the product number or exit the function and return to the main page as shown in Figure 2.3. The system will show the error message if the user still enters a repeated product number until a non-repeated product number is entered or the user chooses to leave the feature. Figure 2.4 shows the situation of the user enter a repeated product number twice and then chooses to leave the feature.


C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
NEW TABLE RECORD
Do you have the Product Name?
If yes, kindly reply '1'. Otherwise, reply any other integer value.
Reply: 1

Kindly input Product Name: Table A
Kindly input Product Number: 111

Do you have the Price and Quantity for the Table?
If yes, kindly reply '1'. Otherwise, reply any other integer value.
Reply: 1

Kindly input:
Price: 250.55
Quantity: 10

Do you have the Shape for the Table?
If yes, kindly reply '1'. Otherwise, reply any other integer value.
Reply: 1

Kindly input:
Shape: Circle

Do you have the length, width, and height for the Table?
If yes, kindly reply '1'. Otherwise, reply any other integer value.
Reply: 1

Kindly input:
Length: 15.0
Width: 12.0
Height: 10.0

New Table record created.
Total of 1 Table is created successfully.
Press any key to continue . . .
```

Figure 2.5

Figure 2.5 shows the running of the entire feature 1 with all the attributes and without any error. The system will then ask the user whether the Table has a price and quantity or not. If the user enters '1', he/she will be asked to enter the price and quantity. Then, the system will ask the same question again and again for the shape, length, width, and height of the Table as shown in Figure 2.5.

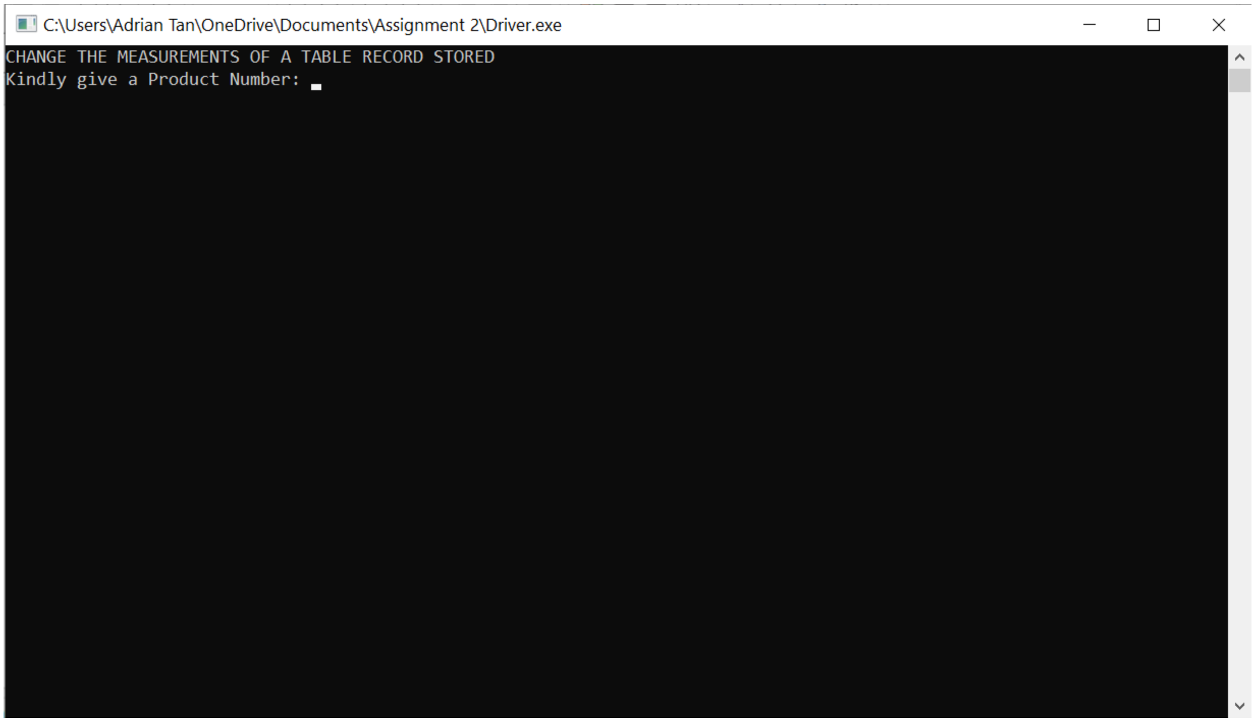

C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
NEW TABLE RECORD
Do you have the Product Name?
If yes, kindly reply '1'. Otherwise, reply any other integer value.
Reply: 0

Kindly input Product Number: 123

Do you have the Price and Quantity for the Table?
If yes, kindly reply '1'. Otherwise, reply any other integer value.
Reply: 0

Do you have the Shape for the Table?
If yes, kindly reply '1'. Otherwise, reply any other integer value.
Reply: 0

Do you have the length, width, and height for the Table?
If yes, kindly reply '1'. Otherwise, reply any other integer value.
Reply: 0

New Table record created.
Total of 3 Tables are created successfully.
Press any key to continue . . .
```

Figure 2.6

Figure 2.6 shows the running of the full feature with only the product number is given and without any error. The system will show a new Table record is created successfully and the total of Table records created at the end of the page.

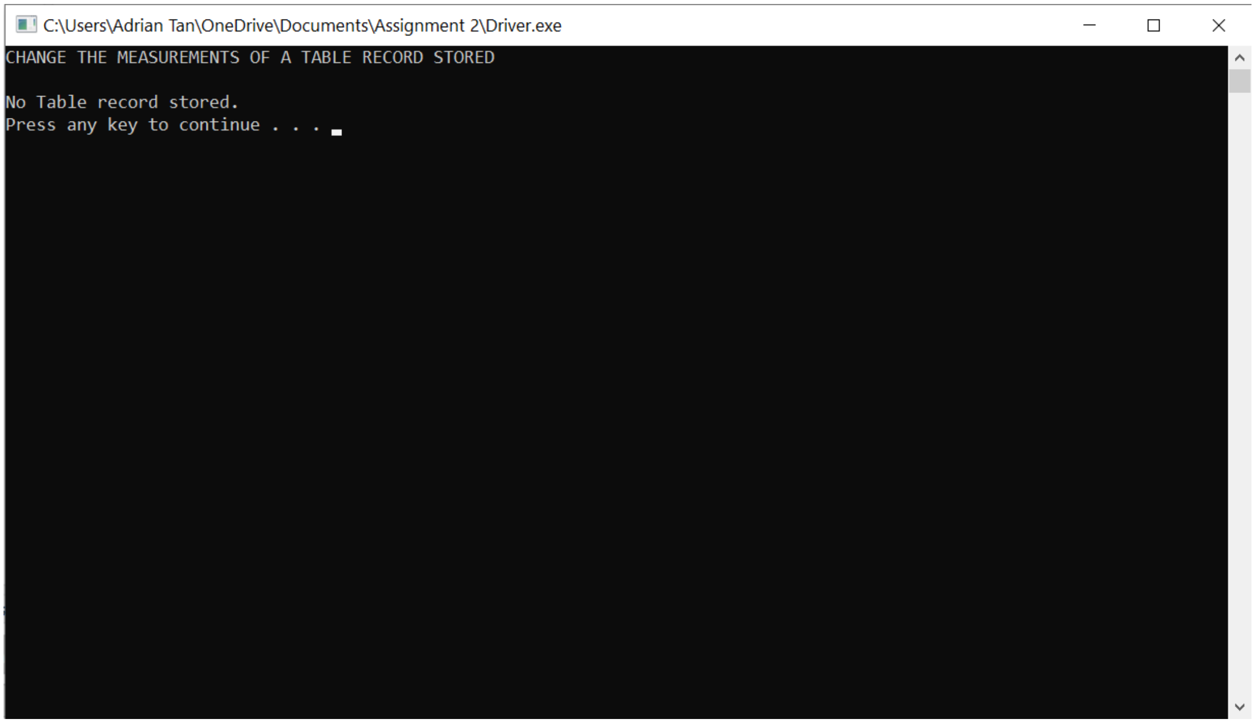
Feature 2 (Change the Measurements of a Table Record Stored)



C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
CHANGE THE MEASUREMENTS OF A TABLE RECORD STORED
Kindly give a Product Number: _
```

Figure 3.1

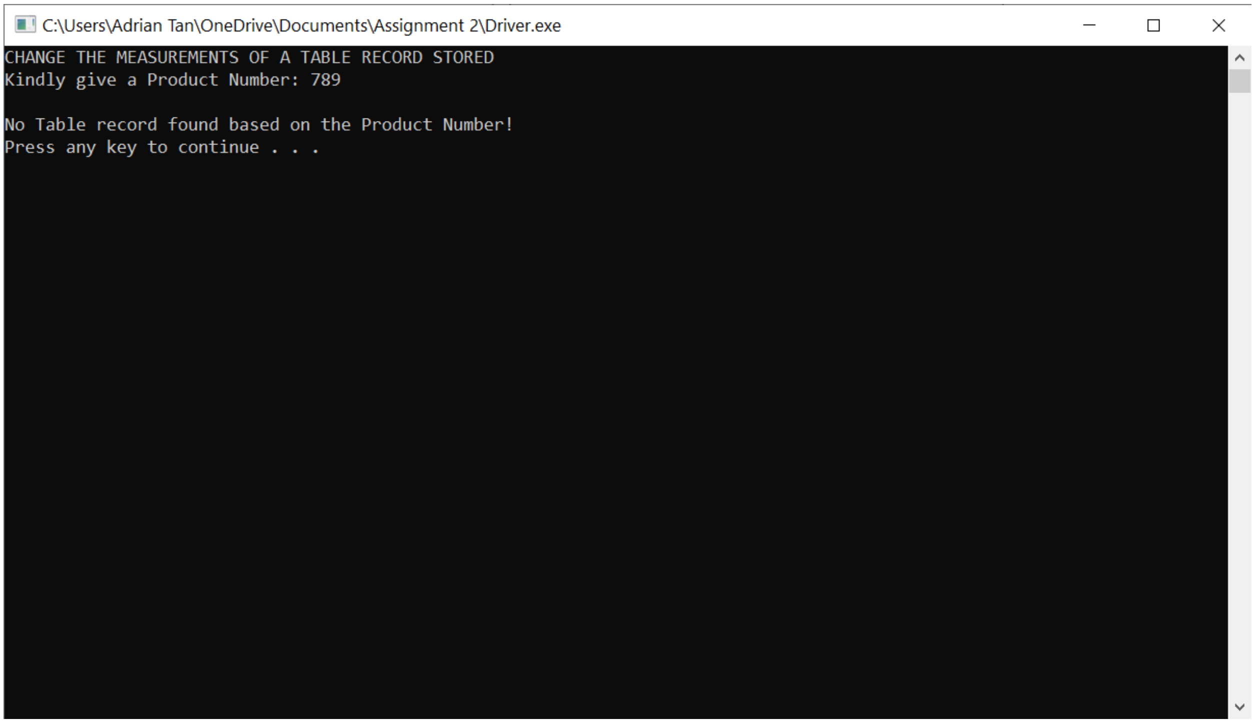


C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
CHANGE THE MEASUREMENTS OF A TABLE RECORD STORED

No Table record stored.
Press any key to continue . . . _
```

Figure 3.2

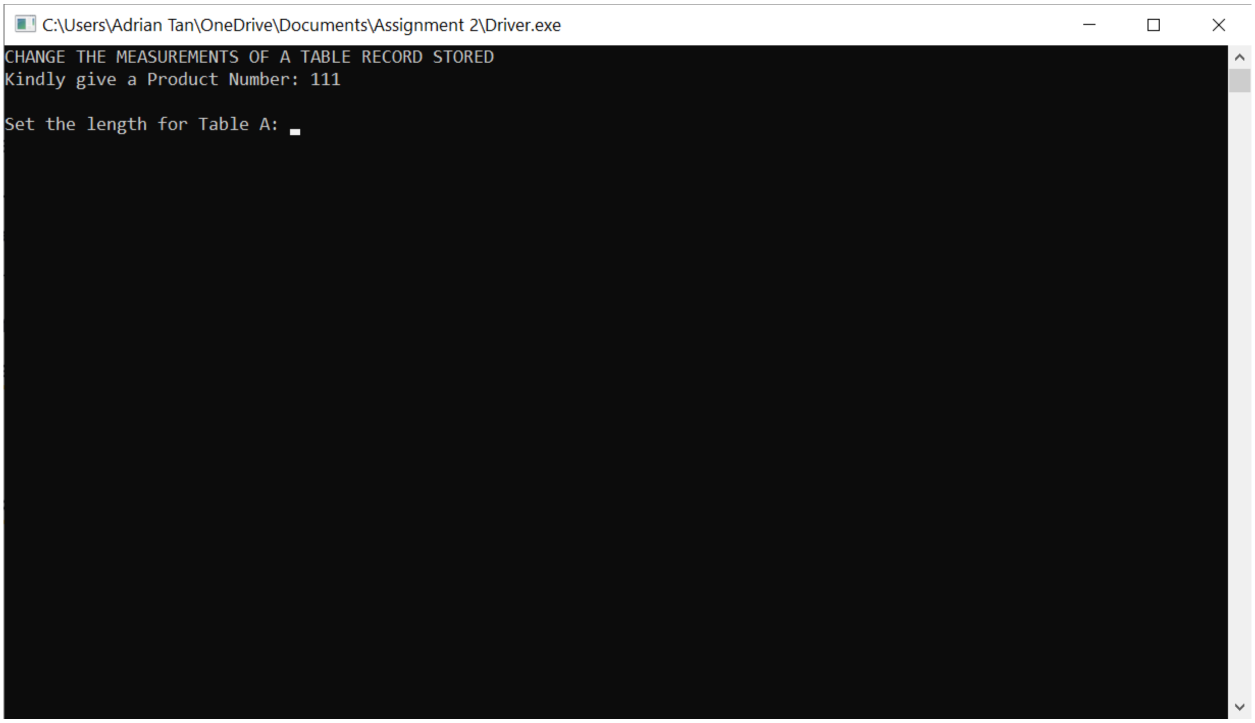
Figure 3.1 shows the page of feature 2 with Table record(s) stored in the system whereas Figure 3.2 shows the page without any Table record stored. If there is a Table record stored, the user will be asked to enter the product number to match with the correct Table record with the system. If the user enters an invalid product number, the system will show an error message as shown in Figure 3.3.



C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
CHANGE THE MEASUREMENTS OF A TABLE RECORD STORED
Kindly give a Product Number: 789

No Table record found based on the Product Number!
Press any key to continue . . .
```

Figure 3.3

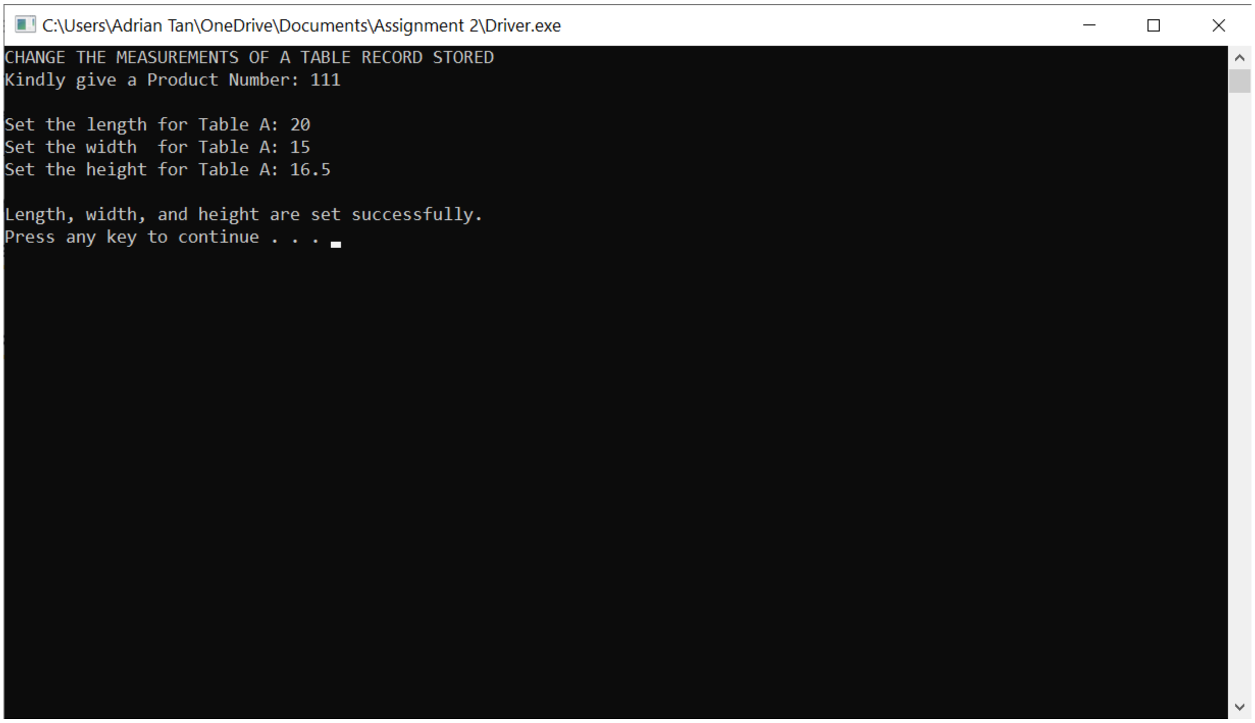

C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
CHANGE THE MEASUREMENTS OF A TABLE RECORD STORED
Kindly give a Product Number: 111

Set the length for Table A: _
```

Figure 3.4

The user will be asked to enter the new length, width, and height of the Table selected after entering a matched product number.

Figure 3.5 shows the running of the full feature without any error. The system will show the new length, width, and height are set successfully at the end of the page.

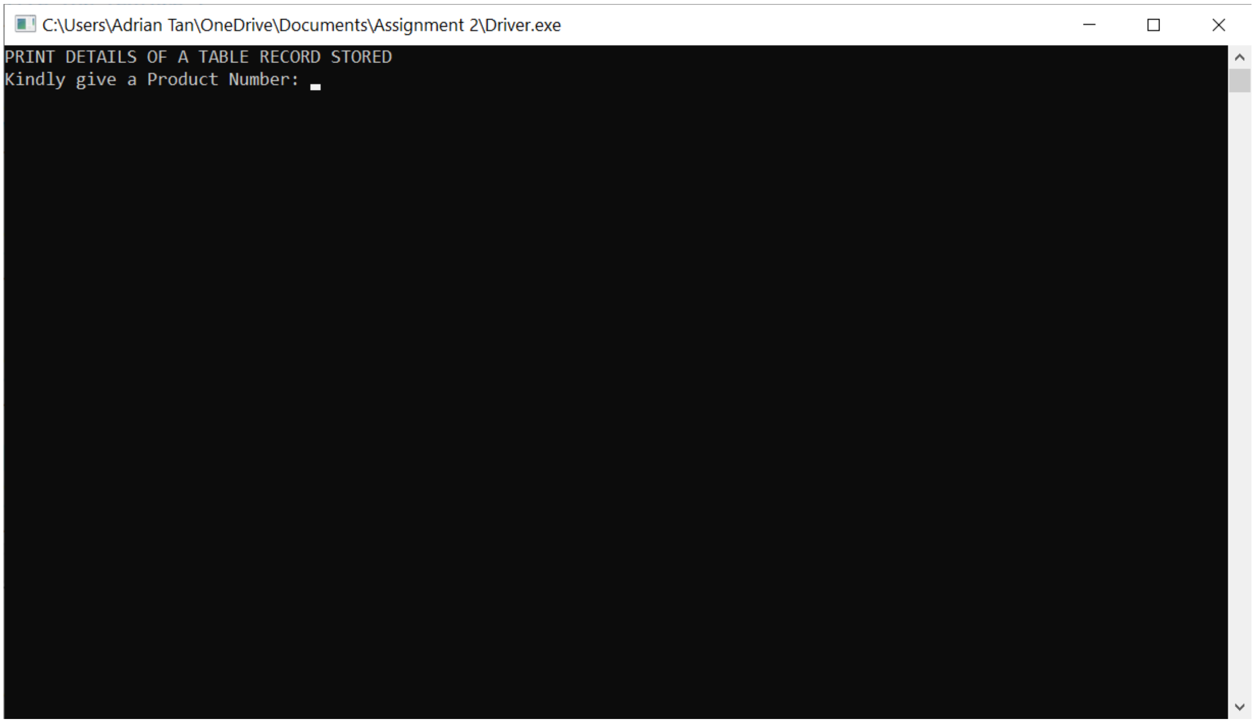


C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
CHANGE THE MEASUREMENTS OF A TABLE RECORD STORED
Kindly give a Product Number: 111

Set the length for Table A: 20
Set the width  for Table A: 15
Set the height for Table A: 16.5

Length, width, and height are set successfully.
Press any key to continue . . .
```

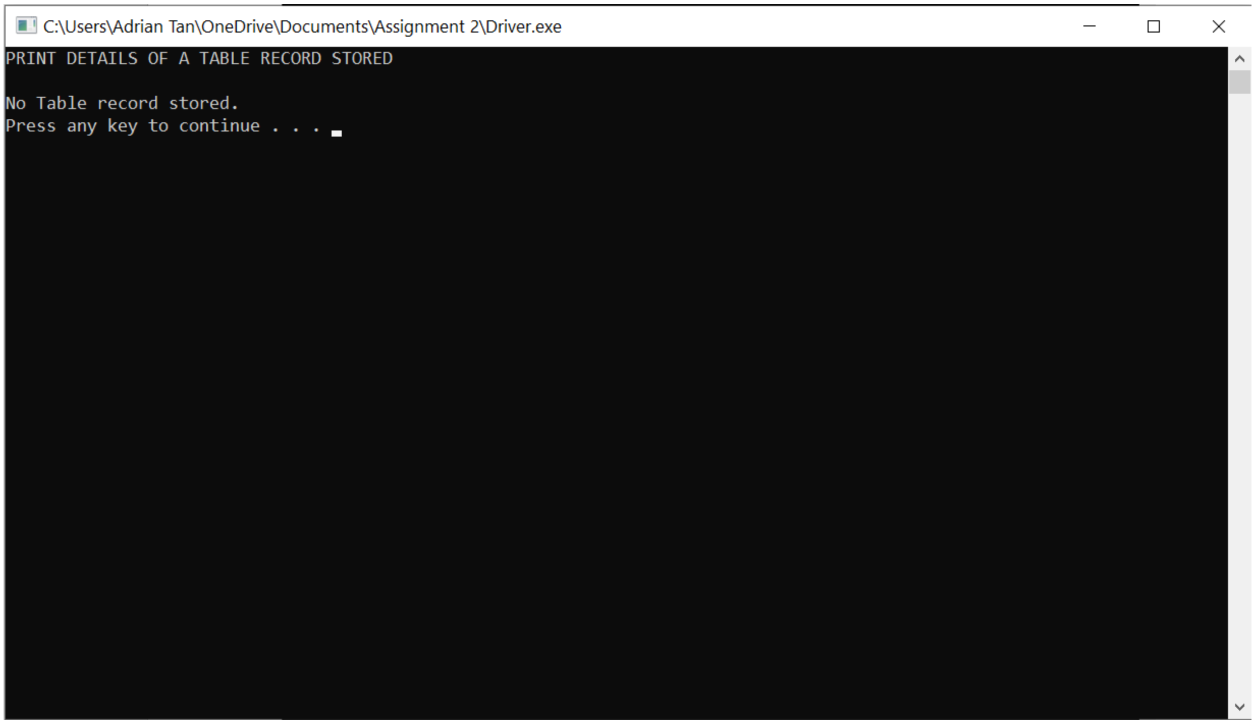
Figure 3.5

Feature 3 (Print Details of a Table Record Stored)


C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
PRINT DETAILS OF A TABLE RECORD STORED
Kindly give a Product Number: 
```

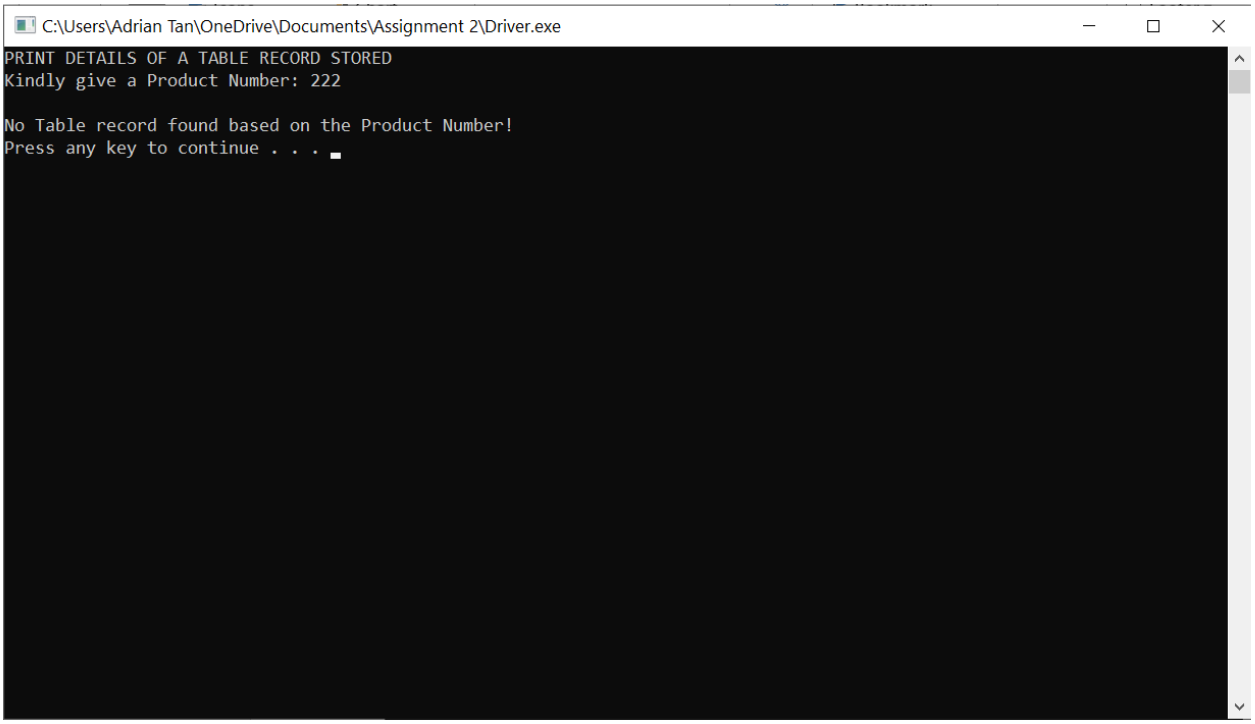
Figure 4.1


C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
PRINT DETAILS OF A TABLE RECORD STORED

No Table record stored.
Press any key to continue . . . 
```

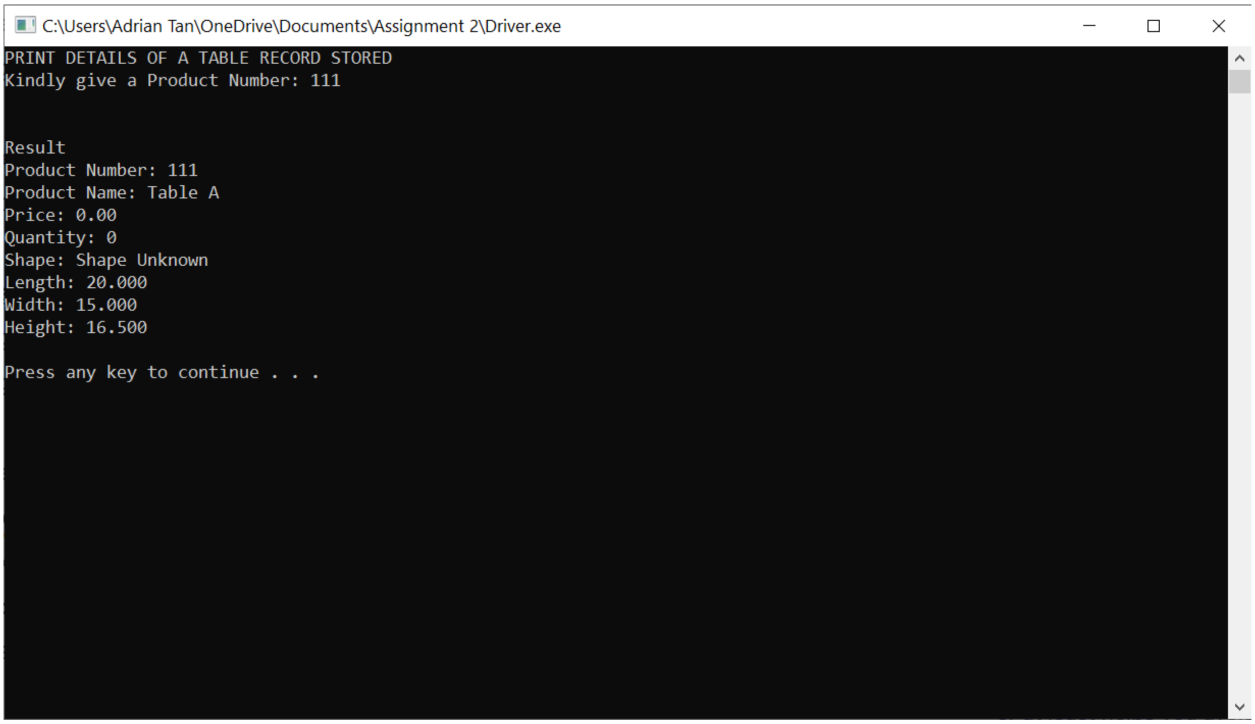
Figure 4.2

Figure 4.1 shows the page of feature 3 with Table record(s) stored in the system whereas Figure 4.2 shows the page without any Table record stored. As in feature 2, if there is a Table record stored, the user will be asked to enter the product number to match with the correct Table record with the system. If the user enters an invalid product number, the system will show an error message as shown in Figure 4.3.


C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
PRINT DETAILS OF A TABLE RECORD STORED
Kindly give a Product Number: 222

No Table record found based on the Product Number!
Press any key to continue . . .
```

Figure 4.3

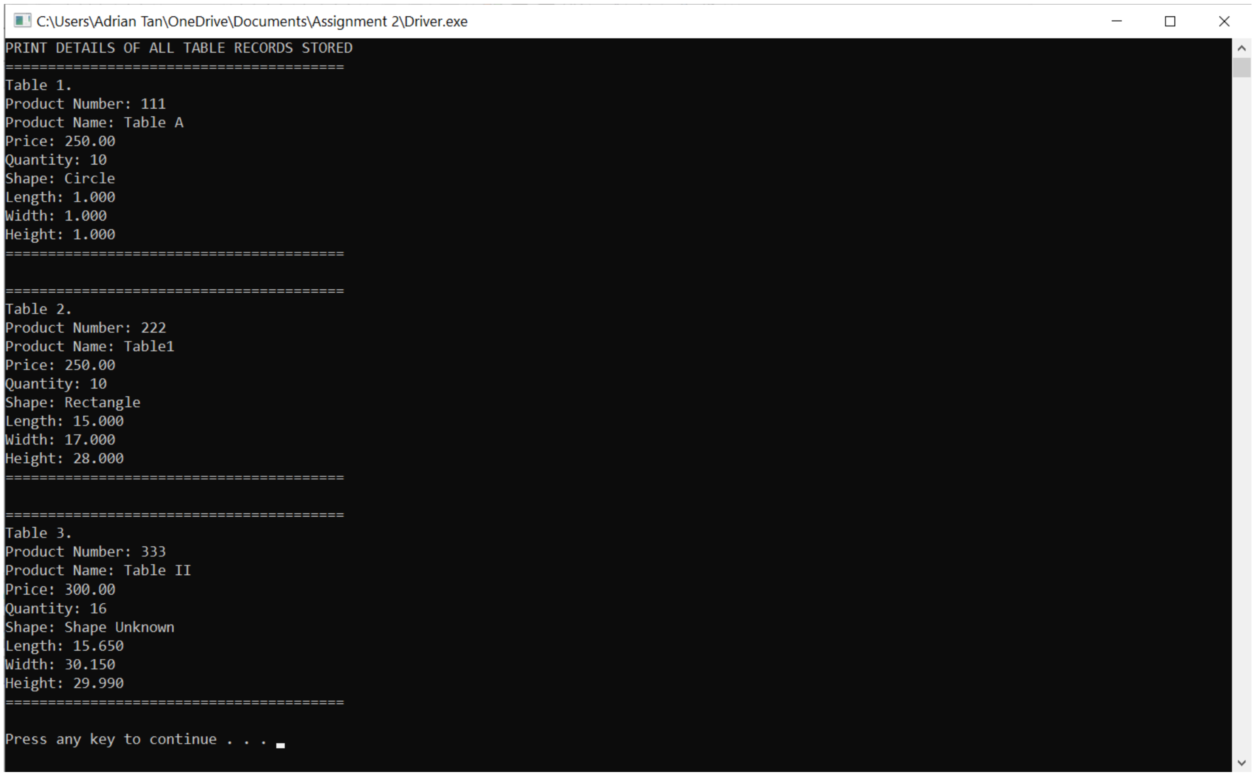

C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
PRINT DETAILS OF A TABLE RECORD STORED
Kindly give a Product Number: 111


Result
Product Number: 111
Product Name: Table A
Price: 0.00
Quantity: 0
Shape: Shape Unknown
Length: 20.000
Width: 15.000
Height: 16.500

Press any key to continue . . .
```

Figure 4.4

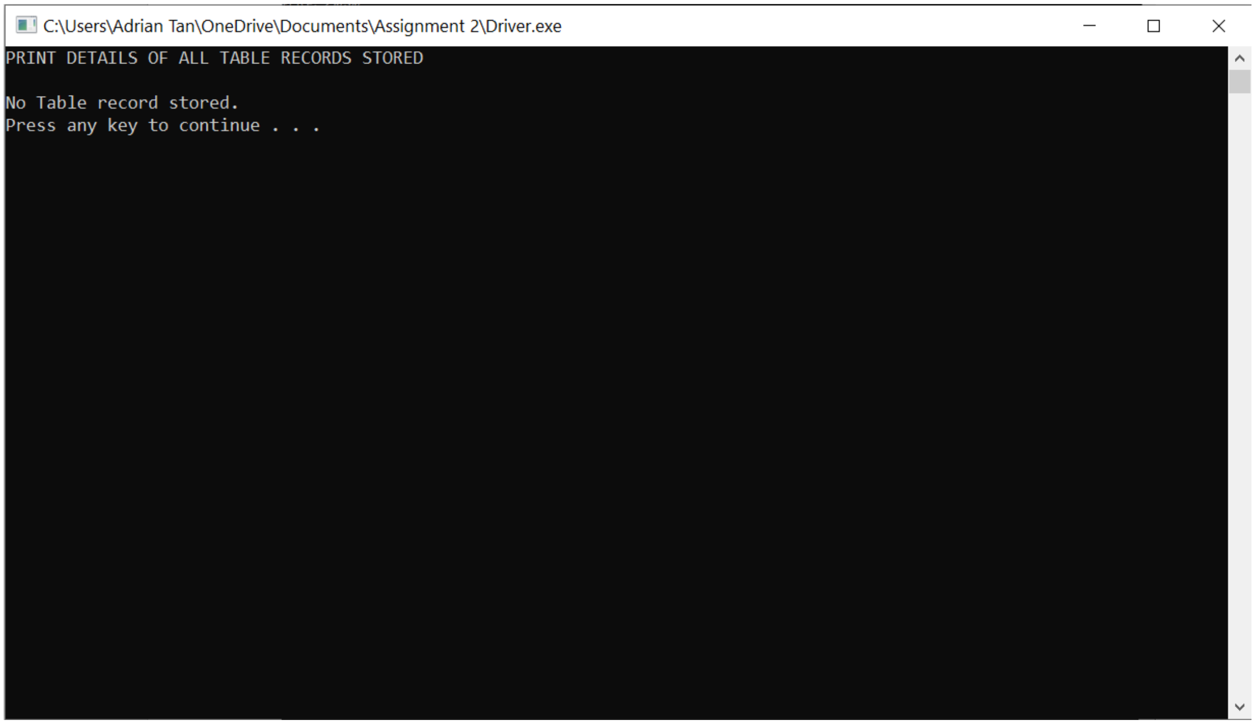
Figure 4.4 shows the running of the full feature without any error. The system will display all the details of the selected Table record with suitable decimal places.

Feature 4 (Print Details of All Table Records Stored)


C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
PRINT DETAILS OF ALL TABLE RECORDS STORED
========================================
Table 1.
Product Number: 111
Product Name: Table A
Price: 250.00
Quantity: 10
Shape: Circle
Length: 1.000
Width: 1.000
Height: 1.000
========================================

========================================
Table 2.
Product Number: 222
Product Name: Table1
Price: 250.00
Quantity: 10
Shape: Rectangle
Length: 15.000
Width: 17.000
Height: 28.000
========================================

========================================
Table 3.
Product Number: 333
Product Name: Table II
Price: 300.00
Quantity: 16
Shape: Shape Unknown
Length: 15.650
Width: 30.150
Height: 29.990
========================================

Press any key to continue . . .
```

Figure 5.1

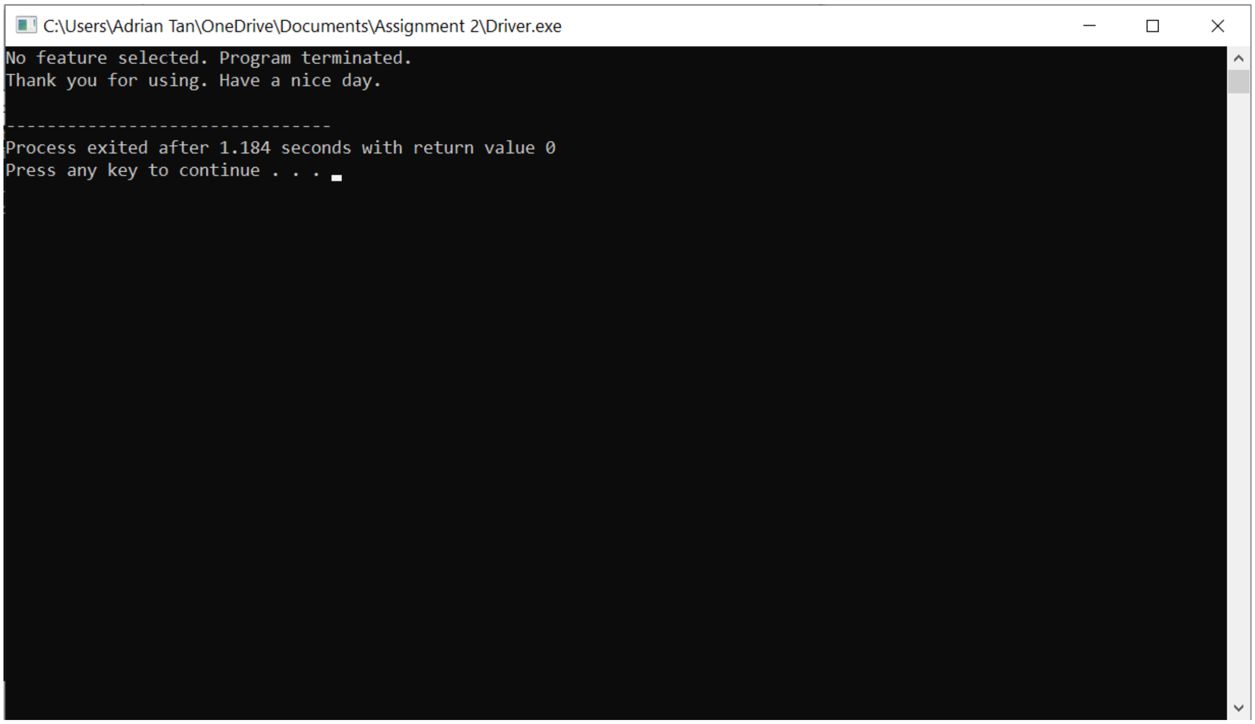

C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
PRINT DETAILS OF ALL TABLE RECORDS STORED

No Table record stored.
Press any key to continue . . .
```

Figure 5.2

Figure 5.1 shows the running of the full feature without any error. The system will display all the details of every Table record stored. Although the system has an array size of 100, it will only display those Table record(s) entered by the user. Those empty arrays will not be displayed for a less distracting view. Figure 5.2 shows the error message as there is no Table record stored.

Feature 4 (Print Details of All Table Records Stored)


C:\Users\Adrian Tan\OneDrive\Documents\Assignment 2\Driver.exe

```
No feature selected. Program terminated.
Thank you for using. Have a nice day.

--------------------------------
Process exited after 1.184 seconds with return value 0
Press any key to continue . . .
```

Figure 6

The system will show some messages before terminating the program including appreciation of using the system.