



Topic : Tourism and Travel Management System

Group No : MLB_WD_CSNE_13_04

Campus : Malabe

Submission Date : 20/05/2022

We declare that this is our own work, and this Assignment does not incorporate without acknowledgment any material previously submitted by anyone else in SLIIT or any other university/Institute. And we declare that each one of us equally contributed to the completion of this Assignment.

Student ID	Name	Contact Number
IT21293344	Wickramaarachchi K. A. S. N.	074 194 9913
IT21352430	Dissanayake B. V. D. P. D.	077 803 2417
IT21360664	Ekanayake K. G. V. M.	076 649 8834
IT21325700	Bandara H. M. S. I. D.	071 221 6631
IT21393570	Mavilmada M. A. S. S.	071 569 8172

Table of Contents

1. System Requirements.....	3
2. Identified Classes	4
3. Methods.....	5
4. CRC Cards	6
5. Class Diagram (UML Notation)	8
6. Exercise 2 : Cording.....	9
6.1. Customer.....	9
6.2. RegisteredCustomer.....	11
6.3. Booking.....	13
6.4. TravellingPackage.....	16
6.5. Hotel.....	18
6.6. Payment.....	20
6.7. Staff.....	22
6.8. Report.....	25
6.9. Main.cpp	27
7. Individual Contribution.....	29

1. System Requirements

- The system should function 24/7/365.
- Customers can overview the system. To use the system, they must register with the system by providing details such as Name, Address, NIC, Email, and contact.
- Registered customers can log into the system by entering the correct username and password.
- The administrative staff should confirm the details.
- Staff can delete or update the status of the package and booking details.
- After confirmation, the system should generate a unique id for the travelling packages and hotels.
- They can book travel packages and hotels under the bookings.
- After booking the package, the registered customer will direct to the payment detail page.
- The payment details page generates Pay ID, Pay date, reference number, and total payment amount to the system.
- Registered customers should make a payment.
- Registered customers must enter their payment details like payment type and card details.
- After the bank or other trusted resources confirm the payment, a report of the payment and booking details for the customer is texted.

2. Identified Classes

- Customer
- Registered customer
- Booking
- Travelling package
- Hotel
- Payment
- Staff

Reasons for rejecting other nouns

Redundant:

An Event or an operation:

Outside scope of system: System, Bank, trusted resources

Meta-language: they

An attribute: Customer Details (Name, Address, NIC, Email, Contact),
Username, password,

3. Methods

- Customer - Register to the system by providing details View the system.
- Registered Customer - Login to the system by entering details.
Logout from the system
- Bookings - Generate book ID
Check availability of packages and hotels
Calculate booking price
- Travelling Package - Generate package id
Display package details
Confirm package type
Add package details
Delete package details
Update package details
- Hotel - Generate hotel ID
Add hotel details
Delete hotel details
Update hotel details
- Payment - Generate pay ID
Check payment details
Calculate total payment
Confirm payments
- Staff - Log into the system
Confirm booking details
Manage registered customer, travelling package and booking details
Logout from the system

4. CRC Cards

Customer	
Responsibility	Collaborators
Register to the system	
Allow to view the System	Travelling package, Hotel

Registered Customer	
Responsibility	Collaborators
Can view and book the Travelling Packages	Travelling Package
Can view and book the Hotels	Hotel
Login to the system	
Logout from the system	

Booking	
Responsibility	Collaborators
Check availability of packages and hotels	Travelling package, Hotel
Calculate booking price	

Travelling Package	
Responsibility	Collaborators
Display package details	
Confirm package type	Staff
Add package details	Staff
Delete package details	Staff
Update package details	Staff

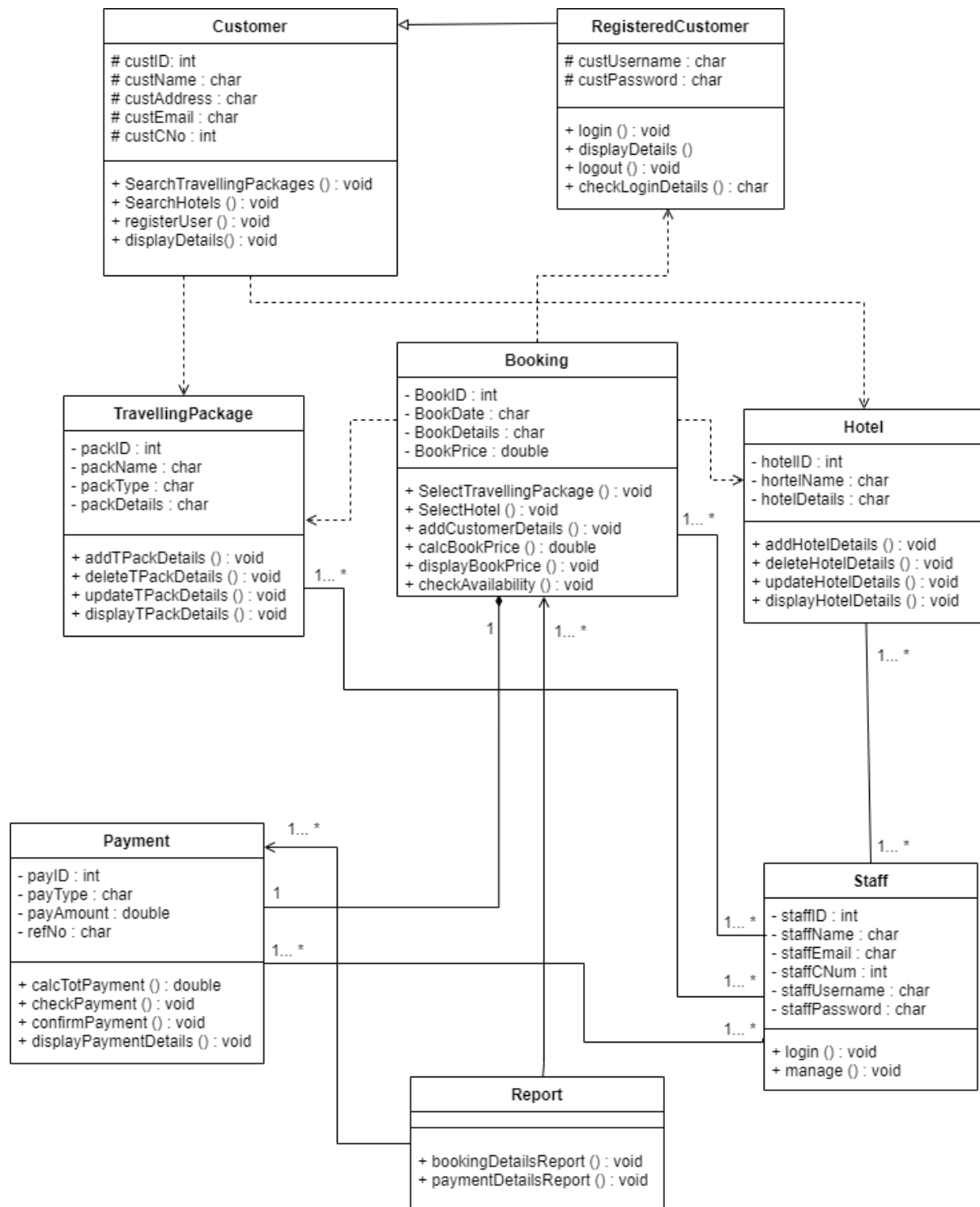
Hotels	
Responsibility	Collaborators
Add hotel details	Staff
Delete hotel details	Staff
Update hotel details	Staff

Payment	
Responsibility	Collaborators
Make a new payment	
Generate Pay ID	Booking
Calculate total payment	Booking
Check payment details	Staff
Confirm payment details	

Staff	
Responsibility	Collaborators
Login to the system	
Confirm booking details	Booking
Manage registered customer, travelling package and booking details	Registered Customer, Travelling package, Booking
Logout from the system	

Report	
Responsibility	Collaborators
Generate Booking details	Booking
Generate Payment details	Payment

5. Class Diagram (UML Notation)



6. Exercise 2 : Cording

6.1. Customer

Customer.h

```
#pragma once
#include "Hotel.h"
#include "TravellingPackage.h"

class Customer
{
protected:
    int custID;
    char custName[20];
    char custAddress[30];
    char custEmail[30];
    int custCNo;

public:
    Customer();
    Customer(int pid, const char pcustName[], const char pcustAddress[],
const char pcustEmail[], int pcustCNo);
    void SearchTravellingPackages (TravellingPackage *cPack);
    void SearchHotels (Hotel *cHotel);
    void registerUser();
    virtual void displayDetails();
    ~Customer();

};
```

Customer.cpp

```
#include "Customer.h"
#include <iostream>
#include <cstring>

using namespace std;

Customer::Customer()
{
    custID = 0;
    strcpy(custName, "");
    strcpy(custAddress, "");
    strcpy(custEmail, "");
    custCNo = 0;
}

Customer::Customer(int pid, const char pcustName[], const char
pcustAddress[], const char pcustEmail[], int pcustCNo)
{
    custID = pid;
    strcpy(custName, pcustName);
```

```
        strcpy(custAddress, pcustAddress);
        strcpy(custEmail, pcustEmail);
        custCNo = pcustCNo;
    }

    void Customer::SearchTravellingPackages (TravellingPackage *cPack)
    {

    }

    void Customer::SearchHotels (Hotel *cHotel)
    {

    }

    void Customer::registerUser()
    {

    }

    void Customer::displayDetails()
    {

    }

    Customer::~~Customer()
    {
        cout << "Destructor Called" << endl;
    }
}
```

6.2. RegisteredCustomer

RegisteredCustomer.h

```
#pragma once
#include "Customer.h"
class RegisteredCustomer :public Customer
{
protected:
    char custUsername[10];
    char custPassword[15];

public:
    RegisteredCustomer();
    RegisteredCustomer(int pid, const char pcustName[], const char
pcustAddress[], const char pcustEmail[], const char pcustCNo[], const char
pcustUsername[], const char pcustPassword[]);
    void displayDetails();
    void login();
    void logout();
    char checkLoginDetails();
    ~RegisteredCustomer();

};
```

RegisteredCustomer.cpp

```
#include "RegisteredCustomer.h"
#include <iostream>
#include <cstring>

using namespace std;

RegisteredCustomer::RegisteredCustomer()
{
    strcpy(custUsername, "");
    strcpy(custPassword, "");
}

RegisteredCustomer::RegisteredCustomer(int pid, const char pcustName[],
const char pcustAddress[], const char pcustEmail[], const char pcustCNo[],
const char pcustUsername[], const char pcustPassword[])
{
    strcpy(custUsername, pcustUsername);
    strcpy(custPassword, pcustPassword);
}

void RegisteredCustomer::displayDetails()
{
}

void RegisteredCustomer::login()
{
}
```

```
}  
  
void RegisteredCustomer::logout()  
{  
  
}  
  
char RegisteredCustomer::checkLoginDetails()  
{  
  
}  
  
RegisteredCustomer::~~RegisteredCustomer()  
{  
  
    cout << "Destructor Called" << endl;  
  
}
```

6.3. Booking

Booking.h

```
#pragma once
#include "Payment.h"
#include "Staff.h"
#include "RegisteredCustomer.h"
#include "Hotel.h"
#include "TravellingPackage.h"

#define SIZE2 2
#define SIZE5 2

class Booking
{
private:
    int BookID;
    char BookDate[15];
    char BookDetails[50];
    double BookPrice;

    Payment *payment[SIZE2];
    Staff *staff[SIZE5];

public:
    Booking();
    Booking(int pBookID, const char pBookDate[], const char pBookDetails[],
double pBookPrice, int pay1_pID, const char pay1_ppayType[], double
pay1_ptotalAmount, const char pay1_prefNo[], int pay2_pID, const char
pay2_ppayType[], double pay2_ptotalAmount, const char pay2_prefNo[], Staff
*bstaff);
    void SelectTravellingPackage (TravellingPackage *tPack);
    void SelectHotel (Hotel *hBooking);
    void addCustomerDetails (RegisteredCustomer *rCust);
    double calcBookPrice(int id, const char pType[], double pAmount);
    void checkAvailability();
    void displayBookPrice();
    ~Booking();

};
```

Booking.cpp

```
#include "Booking.h"
#include <iostream>
#include <cstring>

using namespace std;

Booking::Booking()
{
    BookID = 0;
    strcpy(BookDate, "");
    strcpy(BookDetails, "");
    BookPrice = 0;
```

```

        noOfStaff = 0;

    }

    Booking::Booking(int pBookID, const char pBookDate[], const char
    pBookDetails[], double pBookPrice, int pay1_pID, const char
    pay1_ppayType[], double pay1_ptotalAmount, const char pay1_prefNo[], int
    pay2_pID, const char pay2_ppayType[], double pay2_ptotalAmount, const char
    pay2_prefNo[], Staff *bstaff)
    {

        BookID = pBookID;
        strcpy(BookDate, pBookDate);
        strcpy(BookDetails, pBookDetails);
        BookPrice = pBookPrice;

        payment[0]=new Payment(pay1_pID, pay1_ppayType, pay1_ptotalAmount,
        pay1_prefNo);

        payment[1]=new Payment(pay2_pID, pay2_ppayType, pay2_ptotalAmount,
        pay2_prefNo);

        /* if selection execution */
        if (noOfStaff < SIZE5)
        {

            staff[noOfStaff] = bstaff;

            noOfStaff++;

        } // End if selection execution
    }

    void Booking::SelectTravellingPackage (TravellingPackage *tPack)
    {

    }

    void Booking::SelectHotel (Hotel *hBooking)
    {

    }

    void Booking::addCustomerDetails (RegisteredCustomer *rCust)
    {

    }

    double Booking::calcBookPrice(int id, const char pType[], double pAmount)
    {

    }

    void Booking::checkAvailability()
    {

```

```
}

void Booking::displayBookPrice()
{

}

Booking::~Booking()
{
    for (int i=0; i<SIZE2; i++)
    {
        delete payment[i];
    }
}
```

6.4. TravellingPackage

TravellingPackage.h

```
#pragma once
#include "Staff.h"

#define SIZE5 2

class TravellingPackage
{
private:
    int packID;
    char packName[20];
    char packType[20];
    char packDetails[50];

    Staff *staff[SIZE5];

public:
    TravellingPackage();
    TravellingPackage(int pid, const char ptype[], const char pname[],
const char ppackDetails[], Staff *tstaff);
    void addTPackDetails();
    void deleteTPackDetails();
    void updateTPackDetails();
    void displayTPackDetails();
    ~TravellingPackage();

};
```

TravellingPackage.cpp

```
#include "TravellingPackage.h"
#include <iostream>
#include <cstring>

using namespace std;

TravellingPackage::TravellingPackage()
{
    packID = 0;
    strcpy(packName, "");
    strcpy(packType, "");
    strcpy(packDetails, "");

    noOfStaff = 0;
}

TravellingPackage::TravellingPackage(int pid, const char ptype[], const
char pname[], const char ppackDetails[], Staff *tstaff)
{
    packID = pid;
    strcpy(packName, pname);
    strcpy(packType, ptype);
```



```

strcpy(packDetails, ppackDetails);

/* if selection execution */
if (noOfStaff < SIZE5)
{
    staff[noOfStaff] = tstaff;

    noOfStaff++;

} // End if selection execution
}

void TravellingPackage::addTPackDetails()
{
}

void TravellingPackage::deleteTPackDetails()
{
}

void TravellingPackage::updateTPackDetails()
{
}

void TravellingPackage::displayTPackDetails()
{
}

TravellingPackage::~TravellingPackage()
{
    cout << "Destructor Called" << endl;
}

```

6.5. Hotel

Hotel.h

```
#pragma once
#include "Staff.h"

#define SIZE5 2

class Hotel
{
private:
    int hotelID;
    char hotelName[20];
    char hotelDetails[50];

    Staff *staff[SIZE5];

public:
    Hotel();
    Hotel(int photelID, const char photelName[], const char
photelDetails[], Staff *pstaff);
    void addHotelDetails();
    void deleteHotelDetails();
    void updateHotelDetails();
    void displayHotelDetails();
    ~Hotel();

};
```

Hotel.cpp

```
#include "Hotel.h"
#include <iostream>
#include <cstring>

using namespace std;

Hotel::Hotel()
{
    hotelID = 0;
    strcpy(hotelName, "");
    strcpy(hotelDetails, "");

    noOfStaff = 0;
}

Hotel::Hotel(int photelID, const char photelName[], const char
photelDetails[], Staff *pstaff)
{
    hotelID = photelID;
    strcpy(hotelName, photelName);
    strcpy(hotelDetails, photelDetails);

    if (noOfStaff < SIZE5)
```

```
    {  
  
    staff[noOfStaff] = pstaff;  
    noOfStaff++;  
  
    }  
  
}  
  
void Hotel::addHotelDetails()  
{  
  
}  
  
void Hotel::deleteHotelDetails()  
{  
  
}  
  
void Hotel::updateHotelDetails()  
{  
  
}  
  
void Hotel::displayHotelDetails()  
{  
  
}  
  
Hotel::~~Hotel()  
{  
  
    cout << "Destructor Called" << endl;  
  
}
```

6.6. Payment

Payment.h

```
#pragma once
#include "Staff.h"

#define SIZE5 2

class Payment
{
private:
    int payID;
    char payType[20];
    double totalAmount;
    char refNo[10];

    Staff *staff[SIZE5];

public:
    Payment();
    Payment(int pID, const char ppayType[], double ptotalAmount, const char
prefNo[], Staff *pstaff);
    void checkPayment();
    void confirmPayment();
    void displayPaymentDetails();
    ~Payment();

};
```

Payment.cpp

```
#include "Payment.h"
#include <iostream>
#include <cstring>

using namespace std;

Payment::Payment()
{
    payID = 0;
    strcpy(payType, "");
    totalAmount = 0;
    strcpy(refNo, "");

    noOfStaff = 0;
}

Payment::Payment(int pID, const char ppayType[], double ptotalAmount, const
char prefNo[], Staff *pstaff)
{
    payID = pID;
    strcpy(payType, ppayType);
    totalAmount = ptotalAmount;
    strcpy(refNo, prefNo);
}
```

```

/* if selection execution */
    if (noOfStaff < SIZE5)
    {

        staff[noOfStaff] = pstaff;

        noOfStaff++;

    } // End if selection execution

}

void Payment::checkPayment()
{

}

void Payment::confirmPayment()
{

}

void Payment::displayPaymentDetails()
{

}

Payment::~~Payment()
{

    cout << "Destructor Called" << endl;

}

```

6.7. Staff

Staff.h

```
#pragma once
#include "Booking.h"
#include "TravellingPackage.h"
#include "Hotel.h"
#include "Payment.h"

#define SIZE1 2
#define SIZE2 2
#define SIZE3 2
#define SIZE4 2

class Staff
{
private:
    int staffID;
    char staffName[20];
    char staffEmail[20];
    int staffCNum;
    char staffUserName[20];
    char staffPassword[20];

    Booking *booking[SIZE1];
    Payment *payment[SIZE2];
    TravellingPackage *tPackages[SIZE3];
    Hotel *hotels[SIZE4];

public:
    Staff();
    Staff(int pstaffID, const char pstaffName[], const char pstaffEmail[],
int pstaffCNum, const char pstaffUserName[], const char pstaffPassword[],
Booking *pbooking, TravellingPackages *ptPackages, Hotels *photels, Payment
*ppayment);
    void login(const char stfUserName, const char stfPassword);
    void manage();
    ~Staff();

};
```

Staff.cpp

```
#include "Staff.h"
#include <iostream>
#include <cstring>

using namespace std;

Staff::Staff()
{
    staffID = 0;
    strcpy(staffName, "");
    strcpy(staffEmail, "");
    staffCNum = 0;
    strcpy(staffUserName, "");
```

```

        strcpy(staffPassword, "");

noOfBookings =0;
noOfPayements = 0;
noOfPackages =0;
noOfHotels =0;

}

Staff::Staff(int pstaffID, const char pstaffName[], const char
pstaffEmail[], int pstaffCNum, const char pstaffUserName[], const char
pstaffPassword[], Booking *pbooking, TravellingPackages *ptPackages, Hotels
*photels, Payment *ppayment)
{
    /* if selection execution */
    if (noOfBookings < SIZE1)
    {

        booking[noOfBookings] = pbooking;

        noOfBookings++;

    }// End if selection execution

    /* if selection execution */
    if (noOfPayements < SIZE2)
    {

        payment[noOfPayements] = ppayment;

        noOfPayements++;

    }// End if selection execution

    /* if selection execution */
    if (noOfPackages < SIZE3)
    {

        tPackages[noOfPackages] = ptPackages;

        noOfPackages++;

    }// End if selection execution

    /* if selection execution */
    if (noOfHotels < SIZE4)
    {

        hotels[noOfHotels] = photels;

        noOfHotels++;

    }// End if selection execution

}

void Staff::login(const char stfUserName, const char stfPassword)
{

}

```

```
void Staff::manage()  
{  
  
}  
  
Staff::~~Staff()  
{  
  
    cout << "Destructor Called" << endl;  
  
}
```


6.8. Report

Report.h

```
#pragma once
#include"Booking.h"
#include"Payment.h"
#define SIZE1 2
#define SIZE2 2

class Report
{
private:
    Booking *book[SIZE1];
    Payment *pay[SIZE2];

public:
    Report();
    Report(Booking *pbook[], Payment *ppay[]);
    void bookingDetailsReport();
    void paymentDetailsReport();
    ~Report();

};
```

Report.cpp

```
#include"Report.h"
#include<iostream>

Report::Report()
{
    /* for loop execution */
    for (int i = 0; i < SIZE1; i++)
    {

        book[i] = 0;

    }// End for loop execution

    /* for loop execution */
    for (int j = 0; j < SIZE2; j++)
    {

        pay[j] = 0;

    }// End for loop execution

}

Report::Report(Booking *pbook[], Payment *ppay[])
{
    /* for loop execution */
    for (int i = 0; i < SIZE1; i++)
    {

        book[i] = pbook[i];
```

```

        }// End for loop execution

    /* for loop execution */
    for (int j = 0; j < SIZE2; j++)
    {

        pay[j] = ppay[j];

    }// End for loop execution

}

void Report::bookingDetailsReport()
{

}

void Report::paymentDetailsReport()
{

}

Report::~~Report()
{
    /* for loop execution */
    for (int i = 0; i < SIZE1; i++)
    {

        delete book[i];

    }// End for loop execution

    /* for loop execution */
    for (int j = 0; j < SIZE2; j++)
    {

        delete pay[j];

    }// End for loop execution

}

```

6.9. Main.cpp

```
#include "Booking.h"
#include "Customer.h"
#include "Hotel.h"
#include "RegisteredCustomer.h"
#include "Report.h"
#include "Staff.h"
#include "TravellingPackage.h"

#include <iostream>

using namespace std;

// function main begins program execution
int main()
{
    /* ---- Object Creation ---- */
    Customer *customer = new Customer(); // Object - Customer Class

    RegisteredCustomer *rCustomer = new RegisteredCustomer(); // Object -
    Registered Customer Class

    Booking *booking = new Booking(); // Object - Booking Class

    Hotel *hotel = new Hotel(); // Object - Hotels Class

    TravellingPackage *tPackage = new TravellingPackage(); // Object -
    TravellingPackages Class

    Staff *staff = new Staff(); // Object - Staff Class

    Report *report = new Report(); // Object - Report Class

    /* ---- Method Calling ---- */
    customer->displayDetails();
    customer->registerUser();
    customer->SearchTravellingPackages(booking);
    customer->SearchHotels(hotel);

    rCustomer->displayDetails();
    rCustomer->login();
    rCustomer->logout();

    booking->checkAvailability();
    booking->displayBookPrice();
    booking->SelectTravellingPackage();
    booking->SelectHotel();
    booking->addCustomerDetails();

    hotel->addHotelDetails();
    hotel->deleteHotelDetails();
    hotel->updateHotelDetails();
    hotel->displayHotelDetails();

    tPackage->addTPackDetails();
    tPackage->deleteTPackDetails();
    tPackage->updateTPackDetails();
    tPackage->displayTPackDetails();
}
```

```
        report->bookingDetailsReport();
        report->paymentDetailsReport();

        /* ---- Delete Dynamic Objects ---- */
        delete customer;
        delete rCustomer;
        delete booking;
        delete hotel;
        delete tPackage;
        delete staff;
        delete report;

        return 0;
} //end of function main
```

7. Individual Contribution

Student ID	Name	Contribution
IT21293344	Wickramaarachchi K. A. S. N.	Main.cpp TravellingPackage.h TravellingPackage.cpp
IT21352430	Dissanayake B. V. D. P. D.	Customer.h Customer.cpp RegisteredCustomer.h RegisteredCustomer.cpp
IT21360664	Ekanayake K. G. V. M.	Hotels.h Hotels.cpp Booking.h Booking.cpp
IT21325700	Bandara H. M. S. I. D.	Payment.h Payment.cpp Report.h Report.cpp
IT21393570	Mavilmada M. A. S. S.	Staff.h Staff.cpp