



Topic : Boat trip safari management system

Group no : MLB_10.02_G2_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.

Registration No	Name	Contact Number
IT21380532	Halliyadda H.U.M.S.	0771160889
IT21387326	Rathnaweera W.L.B.	0714238227
IT21387630	Perera V.G.K.B.S.	0704584332
IT21026270	Senarathne D.M.N.P.S.	0702015166
IT21387708	Kumanayaka C.T.	0713389716

Requirements

- 01) All customers can search for boat trips , accommodations and packages.
- 02) Any visitor can sign up on website by providing name, email, username and password.
- 03) All the members can login to the website by providing email and password.
- 04) Once user registered, he/she can view his/her profile, delete profile, or edit profile.
- 05) Registered member can book boat trips, accommodations and he/she can do payments via online by using credit/debit cards and paypal.
- 06) Customer can check if there're any offers available for chosen package.
- 07) If user has any issue regarding payments or need more details about packages he/she can contact our customer service agent through contact us page.
- 08) Customer service agent must handle customers and reply to customer issues.
- 09) If customer wishes to cancel the trip he/she can cancel bookings they've made.
- 10) Any service provider who would prefer to supply boats can register to system as seller by using name, email and password.
- 11) After he/she registered as seller he/she can add new boats to the system and view, edit and update his/her account and boat details.
- 12) When seller supply boats he/she can get payments for service monthly via online.
- 13) Anyone who is willing to drive boats can register to the system as boat driver by filling a form as an employee before drive a boat.
- 14) Manager manages packages and accommodations and manager handles employees.
- 15) Manager can retrieve payments details and calculate total gross profit daily, weekly or monthly.
- 16) System admin also needs to login before do changes on website.
- 17) System admin can update system with new features, edit website, validate memberships and fix errors.

CRC Cards

Customer	
Responsibility	Collaborations
Login to the system	
View profile	
Edit profile	
Delete profile	
Search Trips	Packages, Boat trips, Accommodations
Book trips and accommodations	Packages, Boat trips, Accommodations

Bookings	
Responsibility	Collaborations
Store booking details	Customer, Packages
Display booking details	Customer, Packages

Payment	
Responsibility	Collaborations
Store payment details	
Validate payment details	
Calculate total payments	
Confirm payments	
Cancel payments	
Display payment details	

Card	
Responsibility	Collaborations
Store card payment details	Payment

Pay pal	
Responsibility	Collaborations
Store PayPal payment details	Payment

Boat Seller	
Responsibility	Collaborations
Store boat seller details	
Update boat profile	Boat
Add new boats	Boat
Display recent history	Boat
Reserve payments	Payment

Employee	
Responsibility	Collaborations
Log in to system	
Store employee details	
Display employee details	
Reserve payments	Payment

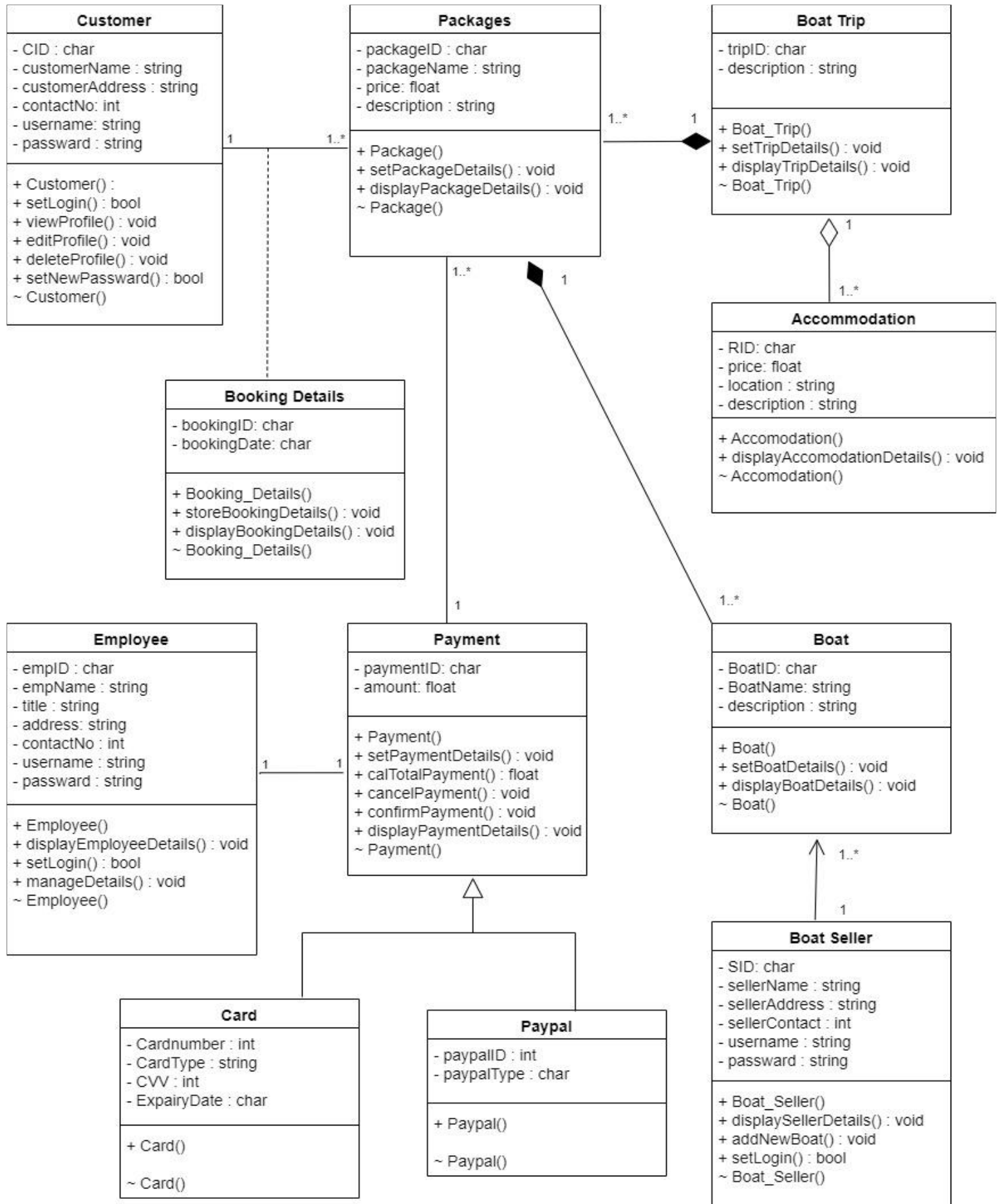
Boat	
Responsibility	Collaborations
Store boat details	
Display boat details	

Packages	
Responsibility	Collaborations
Store package details	Boat trip
Display package details	
Update offers of packages	

Accommodation	
Responsibility	Collaborations
Store accommodation details	
Display accommodation details	
Display booking history	Booking
Add new accommodation details	

Boat trip	
Responsibility	Collaborations
Store trip details	Packages
Display trip details	
Add new trip details	

Exercise 01 - Class diagram



Exercise 02 - Coding

```
#include <iostream>
#include <cstring>

using namespace std;

//Customer class
class Customer
{
private:
    char CID;
    string customerName;
    string customerAddress;
    int contactNo;
    string username;
    string password;
public:
    Customer();
    Customer(char pCID, string cus_Name, string cus_Address, int contact);
    bool setlogin();
    void viewprofile();
    void editprofile();
    void deleteprofile();
    bool setNewPassword();
    ~Customer();
};

//Packages class
class Packages
{
private:
    char packageID;
    string packageName;
    string description;
    float price;
    Payment* pay;
    Boat* b1[10];
public:
    Packages();
    void setPackageDetails(char pID, string pName, string pdesc, float pPrice);
    void displayPackageDetails();
    ~Packages();
};

//BookingDetails class
class BookingDetails
{
private:
    char bookingID;
    string bookingDate;
    Customer* cus1;
    Packages* pack1;
public:
    BookingDetails();
    void storeBookingDetails(char bookID, string date);
    void displayBookingDetails();
    ~BookingDetails();
};

//Boattrip class
class Boattrip {
private:
```

```

        char tripID;
        string description;
        Packages* p[10];
        Accommodation* a[10];
public:
    Boattrip();
    void setTripDetails(char tID, string tDesc);
    void displayTripDetails();
    ~Boattrip();
};

//Accommodation class
class Accommodation
{
private:
    char RID;
    string description;
    string location;
    float price;
public:
    Accommodation();
    Accommodation(char pRID, string aDesc, string aLocation, float aPrice);
    void displayAccommodationDetails();
    ~Accommodation();
};

//Payment class
class Payment
{
protected:
    char paymentID;
    float amount;
    Packages* p[10];
public:
    Payment();
    void setPaymentDetails(char payID, float payAmount);
    float calcTotalPayment();
    void cancelPayment();
    void confirmPayment();
    void displayPaymentDetails();
    ~Payment();
};

//Card class
class Card : public Payment
{
protected:
    int cardNumber;
    char cardType;
    int CVV;
    string exp_date;
public:
    Card();
    Card(int crdNumber, char type, int cvv, string exp);
    ~Card();
};

//Paypal class
class Paypal : public Payment
{
protected:
    int PaypalID;
    char PaypalType;
public:
    Paypal();

```



```

        Paypal(int payID, char payType);
        ~Paypal();
};

//Boat Class
class Boat
{
private:
    char boatID;
    string boatName;
    string description;
public:
    Boat();
    void setBoatDetails(char bID, string bName, string bDesc);
    void displayBoatDetails();
    ~Boat();
};

//Boatseller class
class Boatseller
{
private:
    char SID;
    string sellerName;
    string sellerAddress;
    int sellerContact;
    string username;
    string password;
    Boat* boat;
public:
    Boatseller();
    Boatseller(char sSID, string sName, string sAddress, int sContact);
    void displaySellerDetails();
    void AddNewBoat();
    bool setLogin();
    ~Boatseller();
};

//Employee class
class Employee
{
private:
    char empID;
    string empName;
    string title;
    string address;
    int contactNo;
    string username;
    string password;
public:
    Employee();
    Employee(char pEmpID, string pEmpName, string pTitle, string pEmpAddress, int
pEmpContact);
    void displayEmployeeDetails();
    bool setLogin();
    void manageDetails();
    ~Employee();
};

int main() {

    //Create object for Customer Class
    Customer c1('C001', "Amal", "Colombo", 0771234676);

    c1.setlogin();

```

```

c1.viewprofile();
c1.editprofile();
c1.deleteprofile();
c1.setNewPassword();

//Create object for Package class
Package pack1;

pack1.setPackageDetails('P001', "Family package", "Most famous package", 5000.00);
pack1.displayPackageDetails();

//Create BookingDetails class
BookingDetails BD;

BD.storeBookingDetails('B105', "12/05/2022");
BD.displayBookingDetails();

//Create object for Boattrip class
Boattrip BT1;

BT1.setTripDetails('T002', "Long trip");
BT1.displayTripDetails();

//Create object for Accommodation class
Accommodation a1('R045', "Comfortable", "3rd floor", 1200.00);

//Create object for Boat class
Boat B1;

B1.setBoatDetails('T002', "pearl", "Long trip");
B1.displayBoatDetails();

a1.displayAccommodationDetails();

//Create object for Payment class
Payment p1;

p1.setPaymentDetails('P241', 3500.00);
p1.calculateTotalPayment();
p1.cancelPayment();
p1.confirmPayment();
p1.displayPaymentDetails();

//Create object for Card class
Card crd1(24513, 'VISA', 245, "21/01.2025");

crd1.displayPaymentDetails();

//Create object for Paypal class
Paypal pay1(3425, 'W');

pay1.displayPaymentDetails();

//Create object for Boatseller class(Dynamically)
Boatseller* BS1 = new Boatseller('S005', "Kamal", "Nawala", 0764235167);

BS1->displaySellerDetails();
BS1->setLogin();
BS1->AddNewBoat();

//Create object for Employee class(Dynamically)
Employee* emp1;

emp1 = new Employee('E003', "Amali", "Receptionist", "Nugegoda", 0753452671);

```

```
emp1->setLogin();  
emp1->displayEmployeeDetails();  
emp1->manageDetails();
```

```
delete BS1;  
delete emp1;
```

```
return 0;
```

```
}
```

Contributions

1. H.U.M.S. Halliyadda
 - CRC Cards – Employee,Payment,Card,Paypal,Boatseller
 - Class Diagram – Employee,Payment,Card,Paypal,Boatseller
 - Coding – Employee,Payment,Card,Paypal,Boatseller
2. W.L.B. Rathnaweera
 - CRC Cards – Accommodation,BookingDetails
 - Class Diagram - Accommodation,BookingDetails
 - Coding - Accommodation,BookingDetails
3. V.G.K.B.S. Perera
 - CRC Cards - Boat
 - Class Diagram - Boat
 - Coding - Boat
4. D.M.N.P.S. Senarathne
 - CRC Cards – Packages,Customer
 - Class Diagram - Packages,Customer
 - Coding - Packages,Customer
5. C.T. Kumanayaka
 - CRC Cards - Boattrip
 - Class Diagram - Boattrip
 - Coding - Boattrip