Topic   : Online Pet Care System

Group no   : MLB_01.02_02

Campus   : Malabe

Submission Date: 20/05/2022

Registration No	Name	Contact Number
IT21440236	Kodithuwakku Y. S	076 845 7769
IT21440922	Dewmin T. Y	077 815 5292
IT21704970	Lokuge P. P	070 315 9274
IT21713262	Dissanayake D. C. K	076 932 2812
IT21925818	Peiris H. P	077 033 3314

Requirements Gathering

The proposed system for the online pet care system, Vet Online will focus on providing all the pet related services through a web-based platform where customers are allowed concurrent access to the system and receive facilities such as booking and paying for pet grooming sessions online, participating in online webinar sessions related to pet health and meet other pet owners in the community. In order for these functionalities to exist the system need to implement certain functions and for that we need to identify requirements.

Vet Online is a unique pet care service in Sri Lanka that provides full-service pet care for pet owners who require love and protection for their pets. The suggested system would focus on offering all pet related services such as online booking and payment for pet grooming, pet sitting, and pet training sessions; participating in online webinar sessions related to pet health; and meeting other pet owners in the community via a web-based platform.

A guest can browse the entire website and can be able to view services and customer feedback. Guests can access the pet care centre's social media pages via hyperlinks on the website. If a guest wishes to register as a customer, he/she must supply the following information: name, email, contact number, address, and date of birth. After meeting the conditions, a guest can use their credentials as a registered user to log into the system.

A registered customer can use their credentials to access the website. He/she can book appointments by entering details such as date, time, contact number, and appointment type. Also, registered customers can be able to provide feedback, join webinar sessions, make payments online using credit or debit cards, save payment details, update or modify their account details, and view upcoming and previous appointments.

Staff can confirm appointments and verify payments, which are done by the registered customers. Once the staff member is logged into the system, he/she can validate customer feedback and manage services. Staff are also able to edit/update/remove pet products from the system.

 The system allows for the addition and removal of pet items. The system enables guest users to access the website and create new accounts, register, and access customer feedback. The system should validate the login credentials of

registered users and store the registration details of the members. The system authorizes only registered users to book appointments. The system sends a confirmation email to the customer whenever an appointment is booked. The system always verifies payment details before confirming the booking. The system allows registered users to edit/modify/update their payment information. The system should store all the bookings and display the feedback entered by a registered user. The system allows the registered user to view past appointments.

Furthermore, the system should be accessible 24x7x365, should provide the recovery of login credentials to the customers, should not lose any booking details, and should keep frequent backups. Below listed are the requirements that have been identified for the online pet care system.

- Guest should be able to browse through the entire web site
- Guest should be able to view services and customer feedback
- Guest should be able to create an account and register to the system by providing details such as name, email, contact number, address, and date of birth
- Guest should be able visit the social media pages of the pet care centre through hyperlinks on the website
- Registered user should be able to login to the system using their username and password
- Registered user should be able to book appointments by providing details such as date, time, contact number and type
- Registered user should be able to provide feedback
- Registered user should be able to join webinar sessions
- Registered user should be able to do payments online via credit/debit methods
- Registered user should be able to update/modify their account details
- Registered user should be able to save payment details
- Registered user should be able to view the previous appointments
- Staff should be able to confirm appointments, validate payments, validate customer feedback, and manage services
- Staff should be able to edit/update/remove pet products from the system
- System should allow pet products to be entered/removed from the system.

- System should allow both guest and registered user to view the services offered.
- System should allow guest users to create new accounts and register
- System should allow the guest users to view the customer feedbacks
- System should validate the login credentials of registered user
- System should store the registration details of the members
- System should allow only registered users to book appointments
- System should send a confirmation email to the customer whenever an appointment is booked
- System should verify payment details before confirming the booking
- System should allow registered users to edit/modify/update their profiles
- System should allow registered users to edit/modify/update their payment details
- System should display the given customer feedbacks
- System should store all the bookings done by a registered user
- System should allow the registered user to view past appointments
- The system should be accessible 24x7x365
- System should provide the recovery of login credentials to the customers
- System should not lose any booking details and should keep frequents backups

List of Identified Classes

1. Guest
2. Registered User
3. Appointment
4. Staff
5. Pet products
6. Service
7. Payment
8. Feedback

Parent Class

1. User

CRC Cards for possible classes

Class name: - User	
Responsibilities	**Collaborators**
Input personal details	
Input user ID	
Input username	
Input email address	

Class name: - Guest	
Responsibilities	**Collaborators**
View services	Service
View customer feedback	Feedback
Register to the system	Staff

Class name: - Registered User	
Responsibilities	**Collaborators**
Login to the system	Staff
Update user profile	Staff
Make payments	Payment
Book appointments	Appointment
Provide feedback	Feedback

Class name: - Appointment	
Responsibilities	**Collaborators**
Get appointment type	Service
Get appointment date	
Get appointment time	
Get payment details	Payment
Get user details	Registered User

Class name: - Service	
Responsibilities	**Collaborators**
Add new services	Staff
Delete services	Staff
Update services	Staff
Add service details	Staff

Class name: - Payment	
Responsibilities	**Collaborators**
Add payment methods	Payment
Add appointment charges	Appointment
Add refund details	Payment
Validate payments	Staff
Add payment details	Payment

Class name: - Feedback	
Responsibilities	**Collaborators**
Get service type	Service
Get service details	Service
Get customer details	Registered user

Class name: - Staff	
Responsibilities	**Collaborators**
Confirm appointments	Appointment
Validate payment details	Payment
Validate customer data	Registered user
Manage services	Service

Class name: - Pet Product	
Responsibilities	**Collaborators**
Add product details	Staff
Remove old products	Staff

Class name: - Report	
Responsibilities	**Collaborators**
Create transaction details	Payment
Create Appointment details	Appointment, Registered user
Create feedback report	Feedback, Registered user, Service
Create pet product details report	Pet product
Create service report	Service

Exercise 1

Class Diagram
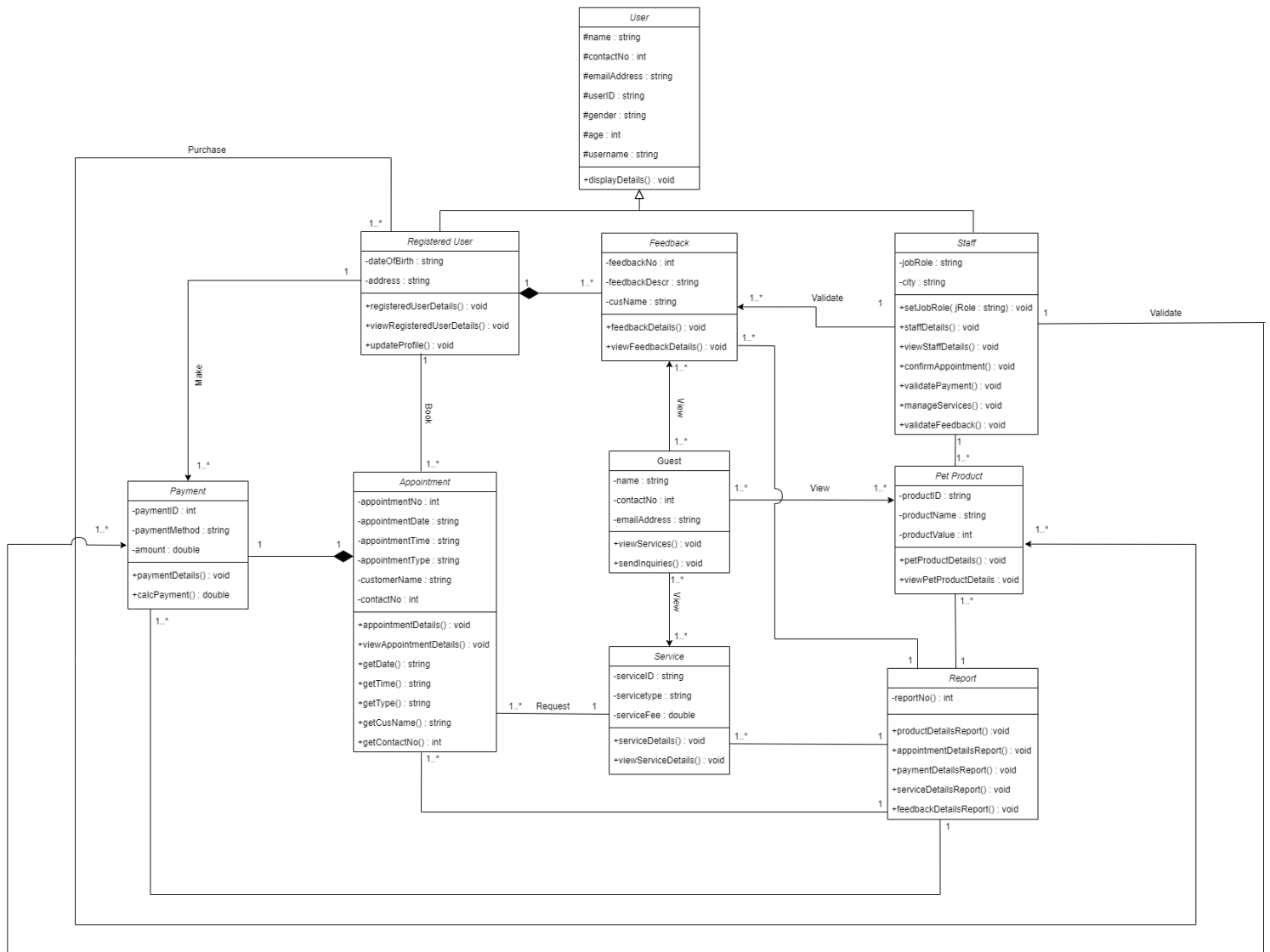
Exercise 2

Coding for the class

```cpp
//MLB_01.02_02_OnlinePetCareSystem

#include <iostream>
#include <string>
#define SIZE1 20
#define SIZE2 10
using namespace std;

//CREATION OF CLASSES
class User;
class RegisteredUser;
class Staff;
class Guest;
class Feedback;
class Appointment;
class Payment;
class Service;
class PetProduct;
class Report;

//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

//DEFINING CLASSES
//IT21440236
class User
{
protected:
    string name;
    string contactNo;
    string emailAddress;
    string userID;
    string gender;
    int age;
    string username;
public:
    User(string uName, string contact, string eAddress, string
uID, string Ugender, int Uage, string user_name);
    void displayDetails();
    ~User();
```

```cpp
};
User::User(string uName, string contact, string eAddress, string
uID, string Ugender, int Uage, string user_name)
{
    name = uName;
    contactNo = contact;
    emailAddress = eAddress;
    userID = uID;
    gender = Ugender;
    age = Uage;
    username = user_name;
}


void User::displayDetails()
{
}


User::~User()
{
}


//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

class RegisteredUser : public User
{
private:
    string dateOfBirth;
    string address;
    Feedback* feedbk[SIZE1];
    PetProduct* ppro[SIZE2];
    Payment* pay[SIZE2];
    Appointment* app[SIZE2];

public:
    RegisteredUser(string uName, string contact, string eAddress,
string uID, string Ugender, int Uage, string user_name,string dob,
string addr,Feedback* fb[SIZE1], PetProduct* petpro[SIZE2], Payment*
payment[SIZE2], Appointment* appoint[SIZE2]);
    RegisteredUser(int fNo1, int fNo2,string fDescr1, string
fDescr2, string cusName1, string cusName2);
    void registeredUserDetails(string dob, string addr);
    void viewRegisteredUserDetails();
    void updateProfile();
    ~RegisteredUser();
};
//Feedback(int fNo, string fDescr, string cName);
RegisteredUser::RegisteredUser(string uName, string contact, string
eAddress, string uID, string Ugender, int Uage, string user_name,
```

```cpp
string     dob,     string     addr,Feedback*     fb[SIZE1],     PetProduct*
petpro[SIZE2], Payment* payment[SIZE2], Appointment* appoint[SIZE2])
: User(uName, contact, eAddress, uID, Ugender, Uage, user_name)
{
      dateOfBirth = dob;
      address = addr;
      for (int i = 0; i < SIZE1; i++) {
           feedbk[i] = fb[i];
      }
      for (int i = 0; i < SIZE2; i++) {
           ppro[i] = petpro[i];
      }
      for (int i = 0; i < SIZE2; i++) {
           pay[i] = payment[i];
      }
      for (int i = 0; i < SIZE2; i++) {
           app[i] = appoint[i];
      }
}
RegisteredUser::RegisteredUser(int fNo1, int fNo2, string fDescr1,
string fDescr2, string cusName1, string cusName2)
{
      feedbk[0] = new Feedback(fNo1,fDescr1,cusName1);
      feedbk[1] = new Feedback(fNo2,fDescr2,cusName2);
}
void RegisteredUser::registeredUserDetails(string dob, string addr)
{
}

void RegisteredUser::viewRegisteredUserDetails()
{
}

void RegisteredUser::updateProfile()
{
}

RegisteredUser::~RegisteredUser()
{
      for (int i = 0; i < SIZE1; i++)
      {
           delete feedbk[i];
      }
}

//|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
//|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

```cpp
//IT21440922
class Staff : public User
{
private:
    string jobRole;
    string city;
    Feedback* feedback[SIZE1];
    Payment* pay[SIZE2];
public:
    Staff(string uName, string contact, string eAddress, string
uID, string Ugender, int Uage, string user_name,string jRole, string
S_city,Feedback* feedbck[SIZE1] ,Payment* payment[SIZE2]);
    void setJobRole(string jRole);
    void staffDetails(string jRole, string S_city);
    void viewStaffDetails();
    void confirmAppointment();
    void validatePayment();
    void manageServices();
    void validateFeedback();
    ~Staff();
};
Staff::Staff(string uName, string contact, string eAddress, string
uID, string Ugender, int Uage, string user_name, string jRole,
string S_city, Feedback* feedbck[SIZE1],Payment* payment[SIZE2]) :
User(uName, contact, eAddress, uID, Ugender, Uage, user_name)
{
    jobRole = jRole;
    city = S_city;
    for (int i = 0; i < SIZE1; i++) {
        feedback[i] = feedbck[i];
    }
    for (int i = 0; i < SIZE2; i++) {
        pay[i] = payment[i];
    }
}

void Staff::setJobRole(string jRole)
{
}


void Staff::staffDetails(string jRole, string S_city)
{
}

void Staff::viewStaffDetails()
{
}
```

```cpp
void Staff::confirmAppointment()
{
}

void Staff::validatePayment()
{
}

void Staff::manageServices()
{
}

void Staff::validateFeedback()
{
}

Staff::~Staff()
{
}

//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

class Guest
{
private:
    string name;
    int contactNo;
    string emailAddress;
    Feedback* fb[SIZE1];
    Service* serv[SIZE2];
    PetProduct* ppro[SIZE2];
public:
    Guest(string gName, int cNo, string eAddress, Feedback*
feedbk[SIZE1], Service* service[SIZE2], PetProduct* petpro[SIZE2]);
    void viewServices();
    void sendInquiries();
    ~Guest();
};
Guest::Guest(string gName, int cNo, string eAddress, Feedback*
feedbk[SIZE1], Service* service[SIZE2], PetProduct* petpro[SIZE2])
{
    name = gName;
    contactNo = cNo;
    emailAddress = eAddress;
    for (int i = 0; i < SIZE1; i++) {
        fb[i] = feedbk[i];
    }
    for (int i = 0; i < SIZE2; i++) {
```

```cpp
            serv[i] = service[i];
        }
        for (int i = 0; i < SIZE2; i++) {
            ppro[i] = petpro[i];
        }
}

void Guest::viewServices()
{
}

void Guest::sendInquiries()
{
}

Guest::~Guest()
{
}

//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

//IT21704970
class Feedback
{
private:
    int feedbackNo;
    string feedbackDescr;
    string cusName;
public:
    Feedback(int fNo, string fDescr, string cName);
    void feedbackDetails();
    void viewFeedbackDetails();
    ~Feedback();
};
Feedback::Feedback(int fNo, string fDescr, string cName)
{
    feedbackNo = fNo;
    feedbackDescr = fDescr;
    cusName = cName;
}

void Feedback::feedbackDetails()
{
}

void Feedback::viewFeedbackDetails()
{
}
```

15

```cpp
Feedback::~Feedback()
{
}

//|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
//|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

class Appointment
{
private:
    int appointmentNo;
    string appointmentDate;
    string appointmentTime;
    string appointmentType;
    string customerName;
    string contactNo;
    RegisteredUser* regUser;
    Service* serv[SIZE2];
    Payment* pay;
public:
    Appointment(int aNo, string aDate, string aTime, string aType,
string aCusName, string cNo, RegisteredUser* reg, Service*
service[SIZE2],Payment* payment);
    Appointment(int pID1,string pMethod1,double amt1);
    void appointmentDetails(int aNo, string aDate, string aTime,
string aType, string aCusName, string cNo);
    void viewAppointmentDetails();
    string getDate();
    string getTime();
    string getType();
    string getCusName();
    int getContactNo();
    ~Appointment();
};
//Payment(int pID, string pMethod, double amt);
Appointment::Appointment(int aNo, string aDate, string aTime, string
aType, string aCusName, string cNo, RegisteredUser* reg, Service*
service[SIZE2], Payment* payment)
{
    appointmentNo = aNo;
    appointmentDate = aDate;
    appointmentTime = aTime;
    customerName = aCusName;
    contactNo = cNo;
    regUser = reg;
    pay = payment;

    for (int i = 0; i < SIZE2; i++) {
```

```cpp
            serv[i] = service[i];
    }
}
Appointment::Appointment(int pID1, string pMethod1, double amt1)
{
    pay = new Payment(pID1,pMethod1,amt1);
}

void Appointment::appointmentDetails(int aNo, string aDate, string
aTime, string aType, string aCusName, string cNo)
{
}

void Appointment::viewAppointmentDetails()
{
}

string Appointment::getDate()
{
    return string();
}

string Appointment::getTime()
{
    return string();
}

string Appointment::getType()
{
    return string();
}

string Appointment::getCusName()
{
    return string();
}

int Appointment::getContactNo()
{
    return 0;
}

Appointment::~Appointment()
{
}


//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

```cpp
//IT21713262
class Payment
{
private:
    int paymentID;
    string paymentMethod;
    double amount;
public:
    Payment(int pID, string pMethod, double amt);
    void paymentDetails();
    double calcPayment();
    ~Payment();
};
Payment::Payment(int pID, string pMethod, double amt)
{
    paymentID = pID;
    paymentMethod = pMethod;
    amount = amt;
}

void Payment::paymentDetails()
{
}

double Payment::calcPayment()
{
}

Payment::~Payment()
{
}

//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

class Service
{
private:
    string serviceID;
    string serviceType;
    double serviceFee;
    Appointment* app[SIZE2];
public:
    Service(string sID, string sType, double sFee, Appointment*
appoint[SIZE2]);
    void serviceDetails(string sID, string sType, double sFee);
    void viewServiceDetails();
    ~Service();
};
```

18

```cpp
Service::Service(string sID, string sType, double sFee, Appointment*
appoint[SIZE2])
{
    serviceID = sID;
    serviceType = sType;
    serviceFee = sFee;
    for (int i = 0; i < SIZE2; i++) {
        app[i] = appoint[i];
    }
}

void Service::serviceDetails(string sID, string sType, double sFee)
{
}

void Service::viewServiceDetails()
{
}

Service::~Service()
{
}

//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

//IT21925818
class PetProduct
{
private:
    string productID;
    string productName;
    double productValue;
public:
    PetProduct(string pID, string pName, double pValue);
    void petProductDetails();
    void viewPetProductDetails();
    ~PetProduct();
};
PetProduct::PetProduct(string pID, string pName, double pValue)
{
    productID = pID;
    productName = pName;
    productValue = pValue;
}

void PetProduct::petProductDetails()
{
}
```

```cpp
void PetProduct::viewPetProductDetails()
{
}

PetProduct::~PetProduct()
{
}
```

//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

```cpp
class Report
{
private:
    int reportNo;
    PetProduct* pPro[SIZE1];
    Appointment* app[SIZE1];
    Payment* pay[SIZE1];
    Service* serv[SIZE1];
    Feedback* fb[SIZE1];
public:
    Report(int   rNo,   PetProduct*   petPro[SIZE1],   Appointment*
appoint[SIZE1], Payment* purchase[SIZE1], Service* service[SIZE1],
Feedback* feedbk[SIZE1]);
    void productDetailsreport();
    void appointmentDetailsreport();
    void paymentDetailsreport();
    void serviceDetailsreport();
    void feedbackDetailsreport();
    ~Report();
};
Report::Report(int   rNo,   PetProduct*   petPro[SIZE1],   Appointment*
appoint[SIZE1], Payment* purchase[SIZE1], Service* service[SIZE1],
Feedback* feedbk[SIZE1])
{
    reportNo = rNo;
    for (int i = 0; i < SIZE1; i++) {
        pPro[i] = petPro[i];
    }
    for (int i = 0; i < SIZE1; i++) {
        app[i] = appoint[i];
    }
    for (int i = 0; i < SIZE1; i++) {
        pay[i] = purchase[i];
    }
    for (int i = 0; i < SIZE1; i++) {
        serv[i] = service[i];
    }
```

```cpp
        for (int i = 0; i < SIZE1; i++) {
            fb[i] = feedbk[i];
        }
}

void Report::productDetailsreport()
{
}

void Report::appointmentDetailsreport()
{
}

void Report::paymentDetailsreport()
{
}

void Report::serviceDetailsreport()
{
}

void Report::feedbackDetailsreport()
{
}

Report::~Report()
{
}

//|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
//|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

//CREATING OBJECTS AND CALLING NECESSARY CONSTRUCTORS
int main()
{
    User *user;
    RegisteredUser *registereduser;
    Staff *staff;
    Guest *guest;
    Feedback *feedback;
    Appointment *appointment;
    Payment *payment;
    Service *service;
    PetProduct *petproduct;
    Report *report;

    //ALLOCATING MEMORY
```

```cpp
User = new
User("Shenaya","0778457765","shenayaJ@gmail.com","US001","Female",22
,"shenayaj2022");

    registereduser  =  new  RegisteredUser("Jude",  "0768457765",
"JudeM@gmail.com",  "US002",  "Male",  28,  "Judemel123","1994-11-
04","98/4,Malabe",feedback,petproduct,payment,appointment);

    staff = new Staff("Jagath", "0718457765", "Jagaths@gmail.com",
"ST001", "Male", 48, "jagathS","Groomer","Matara",feedback,payment);

    guest = new
Guest("Anonymous1","",feedback,service,petproduct);

    feedback = new Feedback(01,"Excellent service", "Shenaya");

    appointment  =  new  Appointment(01,"2022-5-19","18:00","Dog
Grooming","Shenaya","0768455667",registereduser,service,payment);

    payment = new Payment(001,"Visa",9500);

    service = new Service("SR001","Grooming",9500,appointment);

    petproduct = new PetProduct("PR001","Zooloyal Dog
Food",12900);

    report = new
Report(01,petproduct,appointment,payment,service,feedback);



    //RELEASING USED MEMORY
    delete user;
    delete registereduser;
    delete staff;
    delete guest;
    delete feedback;
    delete appointment;
    delete payment;
    delete service;
    delete petproduct;
    delete report;


    return 0;
}
```