Topic : Boat Safari Trip Management System

Group no : PRO_05

Campus : Malabe

Submission Date : 20 / 05 /2022

Registration No	**Name**	**Contact Number**
IT20151560	Lakshan N.M.D	075-3251369
IT20160302	Rajapaksha P.R.I.S	076-5280144

01) System Requirements for Boat Safari Trip Management System

- This system allows all users to search boat for trip
- Any traveler can sign up to become a member
- The boat safari trip management system, all members can access the website by entering their email address and password.
- Once user registered, he/she can view his profile, delete profile, or edit profile details.
- Once a member filled the form, they can add their own information to the website for safari trip management system.
- A member can make payments online through our website via using credit cards & online money transfer.
- Admin also needs to login to the system before do the activities.
- Admin can also generate payment reports & he can accept the tour.

02) Nouns/Verbs analysis

- ➢ Nouns in Red colour
- ➢ Verbs in Blue colour

- This system allows all users to search boat for trip
- Any traveler can sign up to become a member
- The boat safari trip management system, all members can access the website by entering their email address and password.
- Once user registered, he/she can view his profile, delete profile, or edit profile details.
- Once a member filled the form, they can add their own information to the website for safari trip management system.
- A member can make payments online through our website via using credit cards & online money transfer.
- Admin also needs to login to the system before do the activities.
- Admin can also generate payment reports & he can accept the tour.

Identified Classes using Noun Verb Analysis

➢ Identified Classes:

- Admin
- Role
- Member
- Payment
- Tour

➢ Nouns

- User
- Traveler
- Member
- Management System
- Website
- Email address
- Password
- Form
- Information
- Payment
- Credit card
- Online transfer
- System
- Admin

03) CRC Cards for Boat Safari Management System

Role	
Responsibilities	Collaborations
Add roles	
Restock	
Update	

Member	
Responsibilities	Collaborations
Login to the system	
Search boat tour	Tour
View profile	
Delete profile	
Edit profile details	
Request for a tour	Admin
Apply for a tour	Admin
Make Payment	Payment
Request to add Tour	Admin
Buy a boat for trip	Tour

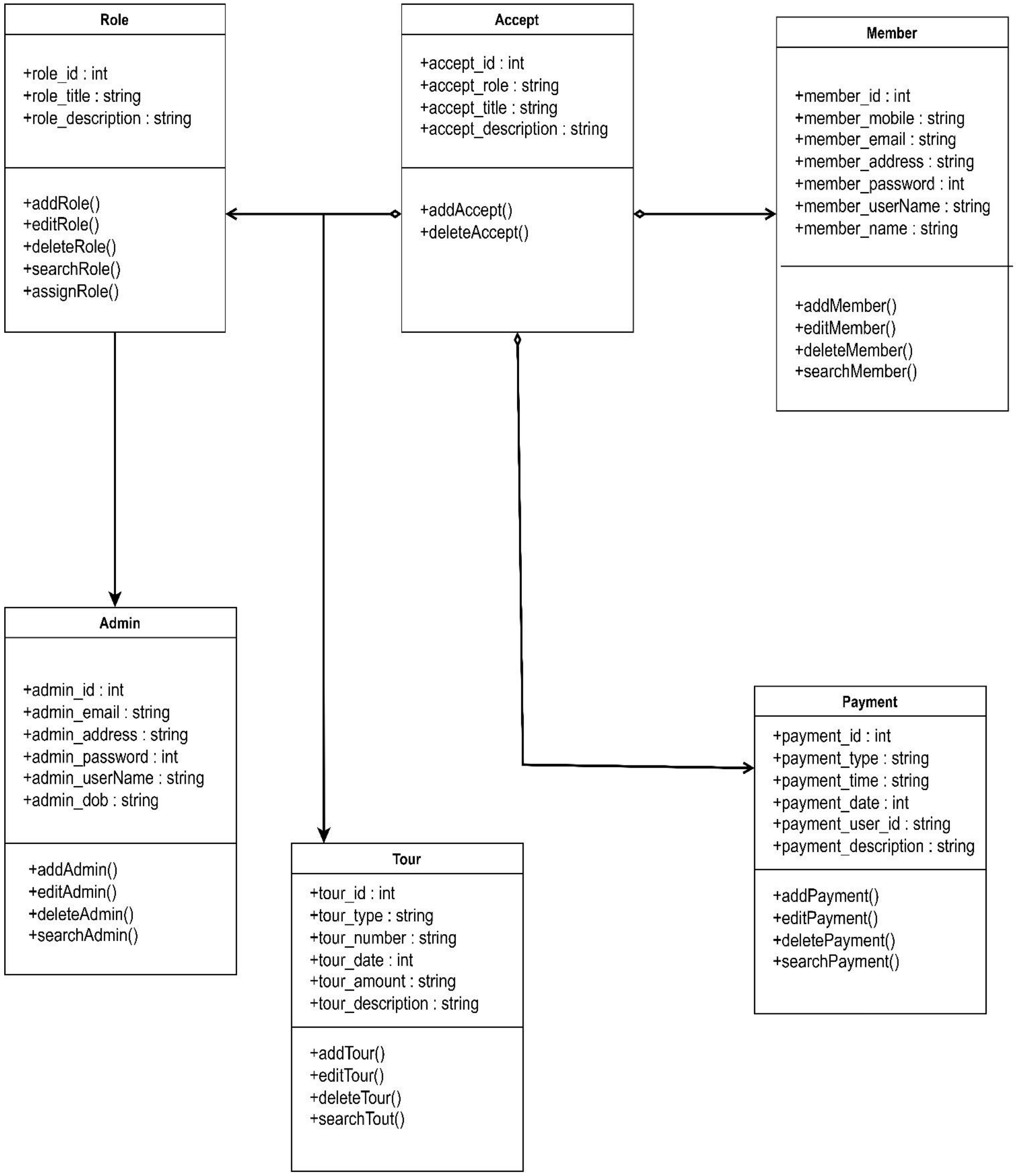
Admin	
Responsibilities	Collaborations
Login to the system	Member
Accept for the member request	Member
Add a member	
Generate report	

Accept	
Responsibilities	Collaborations
Accept tour	Member Payment

Payment	
Responsibilities	Collaborations
View payment details	Member
Validate the payment	
Keep a record of all member members	
Calculate and store amount of deposit returned	
Issue receipt	Member

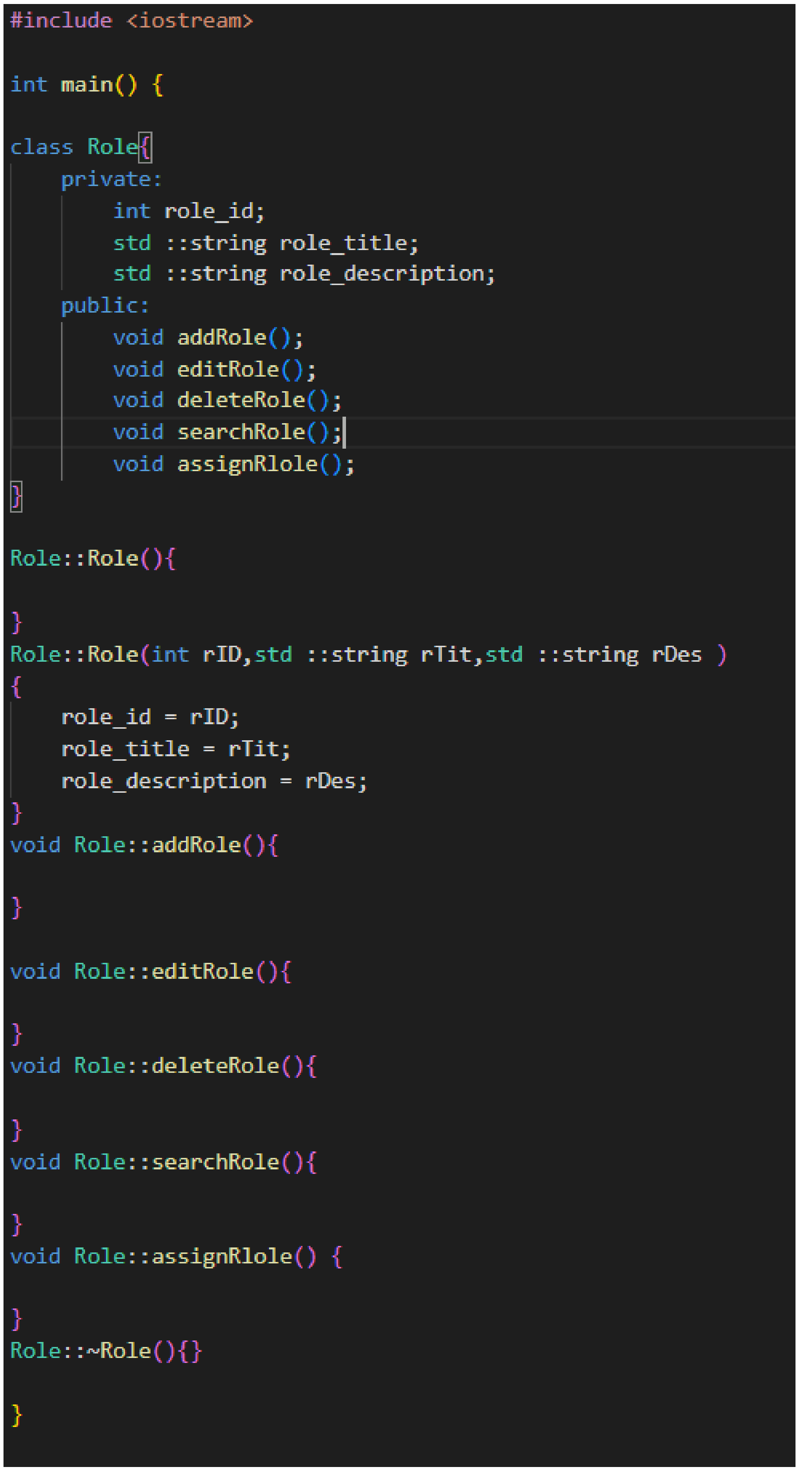
Tour	
Responsibilities	Collaborations
Issue the receipt	Payment
Issue & show the tour policy	Accept

Class Diagram for the Safari Management System



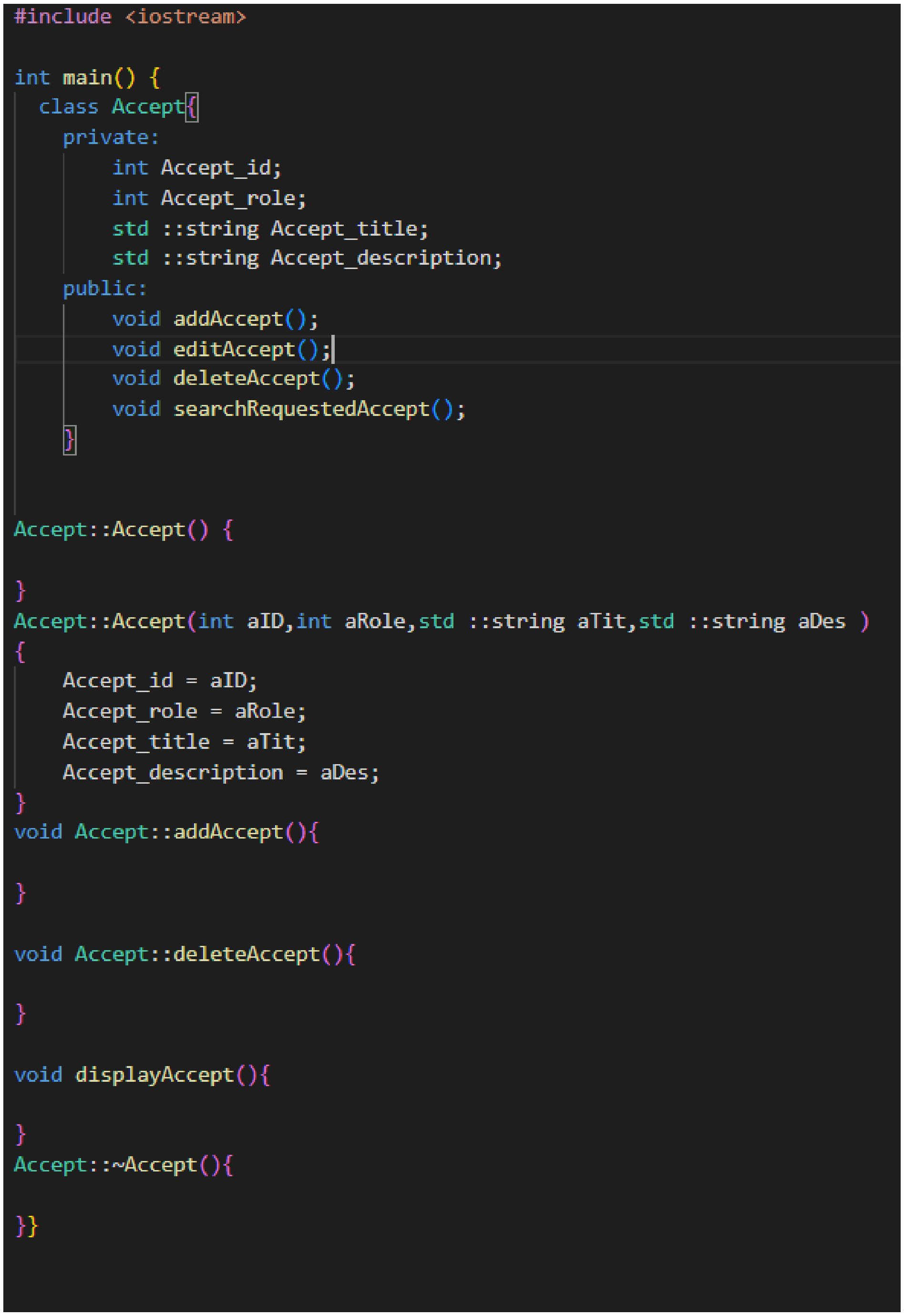
Program Code

Role :



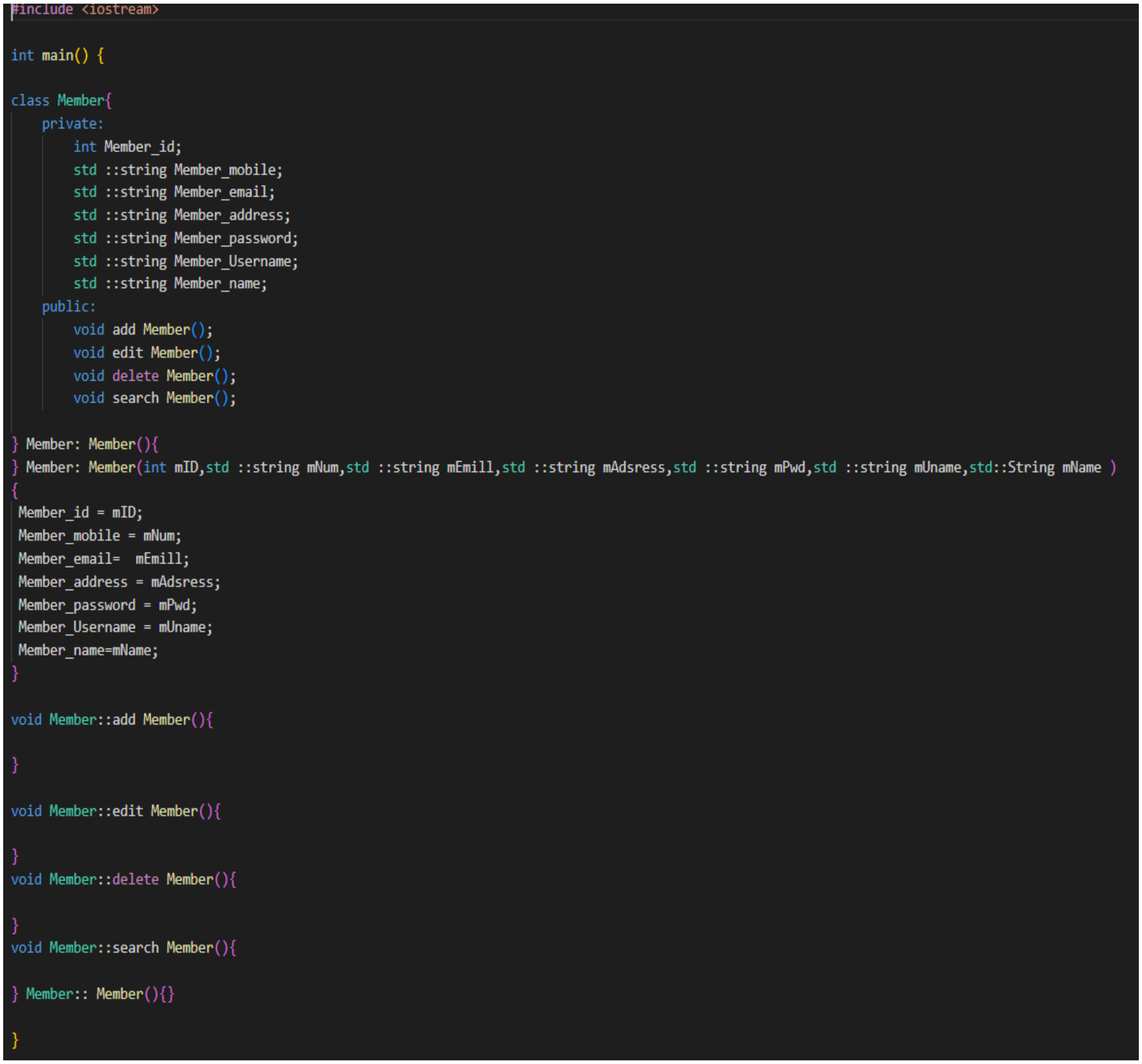
```
#include <iostream>

int main() {

class Role{
    private:
        int role_id;
        std ::string role_title;
        std ::string role_description;
    public:
        void addRole();
        void editRole();
        void deleteRole();
        void searchRole();
        void assignRlole();
}

Role::Role(){

}
Role::Role(int rID,std ::string rTit,std ::string rDes )
{
    role_id = rID;
    role_title = rTit;
    role_description = rDes;
}
void Role::addRole(){

}

void Role::editRole(){

}
void Role::deleteRole(){

}
void Role::searchRole(){

}
void Role::assignRlole() {

}
Role::~Role(){}

}
```

Accept :



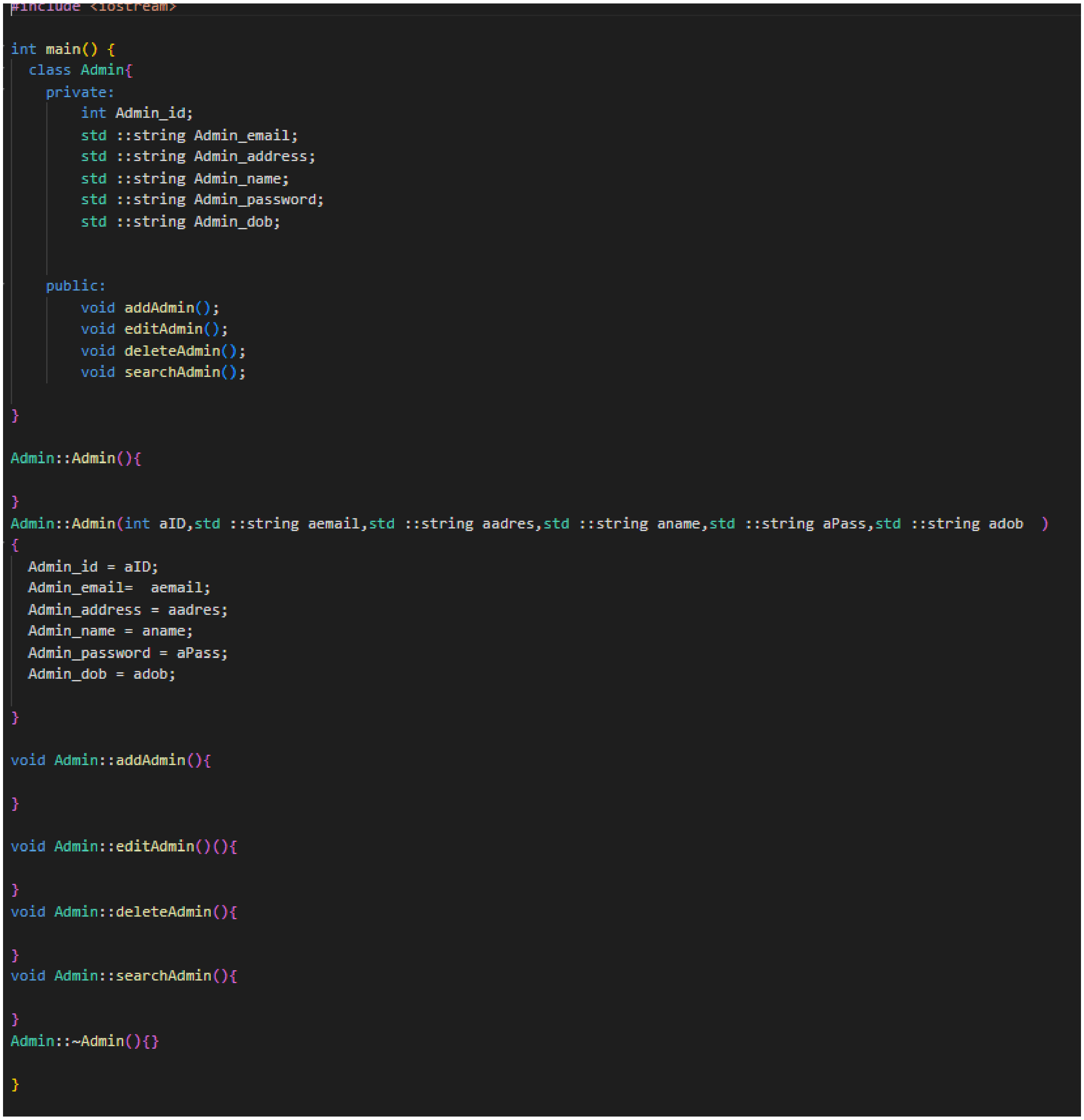
```
#include <iostream>

int main() {
  class Accept{
    private:
        int Accept_id;
        int Accept_role;
        std ::string Accept_title;
        std ::string Accept_description;
    public:
        void addAccept();
        void editAccept();
        void deleteAccept();
        void searchRequestedAccept();
    }


Accept::Accept() {

}
Accept::Accept(int aID,int aRole,std ::string aTit,std ::string aDes )
{
    Accept_id = aID;
    Accept_role = aRole;
    Accept_title = aTit;
    Accept_description = aDes;
}
void Accept::addAccept(){

}

void Accept::deleteAccept(){

}

void displayAccept(){

}
Accept::~Accept(){

}}
```

Member :



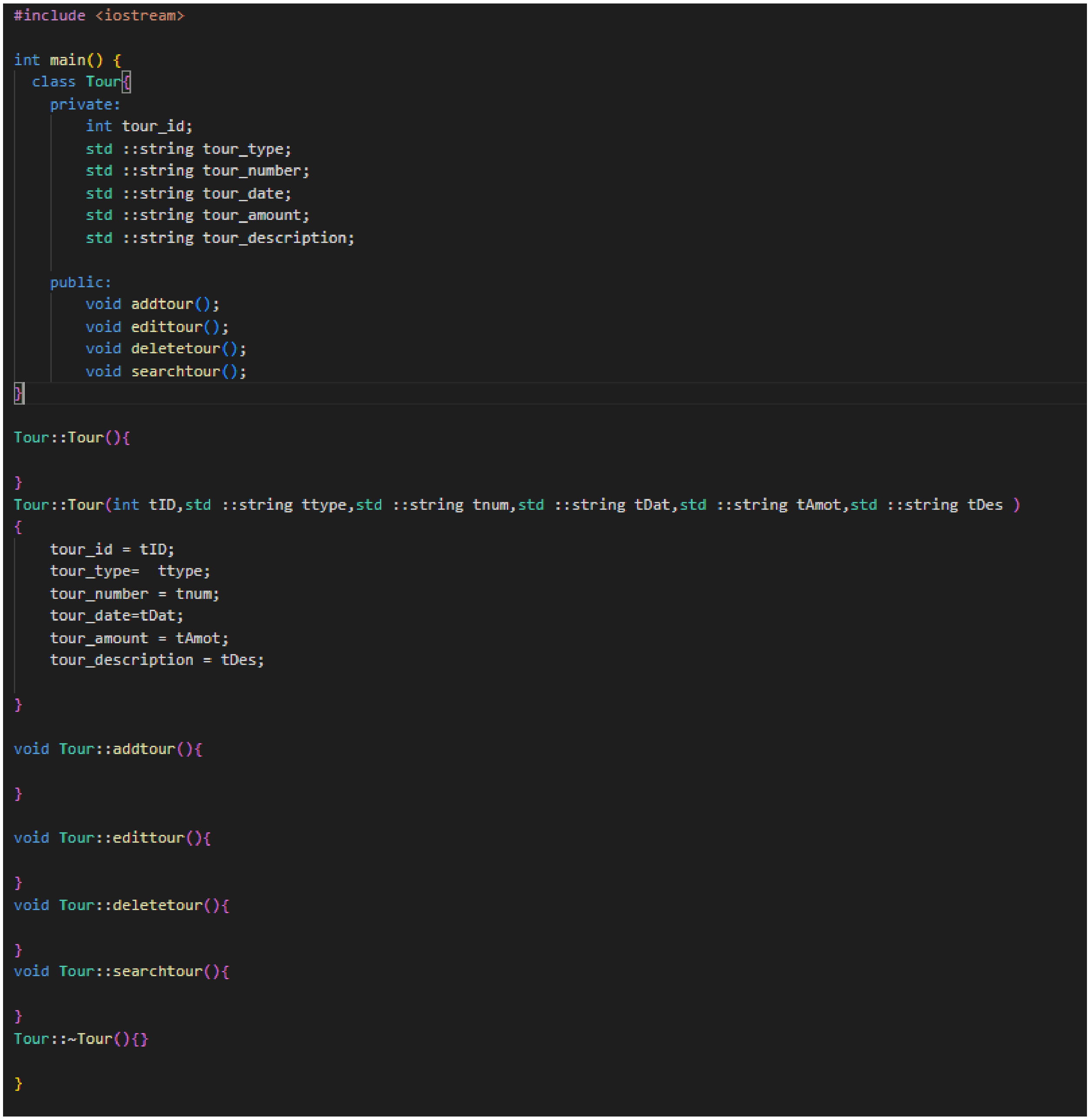
```
#include <iostream>

int main() {

class Member{
    private:
        int Member_id;
        std ::string Member_mobile;
        std ::string Member_email;
        std ::string Member_address;
        std ::string Member_password;
        std ::string Member_Username;
        std ::string Member_name;
    public:
        void add Member();
        void edit Member();
        void delete Member();
        void search Member();

} Member: Member(){
} Member: Member(int mID,std ::string mNum,std ::string mEmill,std ::string mAdsress,std ::string mPwd,std ::string mUname,std::String mName )
{
 Member_id = mID;
 Member_mobile = mNum;
 Member_email=  mEmill;
 Member_address = mAdsress;
 Member_password = mPwd;
 Member_Username = mUname;
 Member_name=mName;
}

void Member::add Member(){

}

void Member::edit Member(){

}
void Member::delete Member(){

}
void Member::search Member(){

} Member:: Member(){}

}
```

Admin :



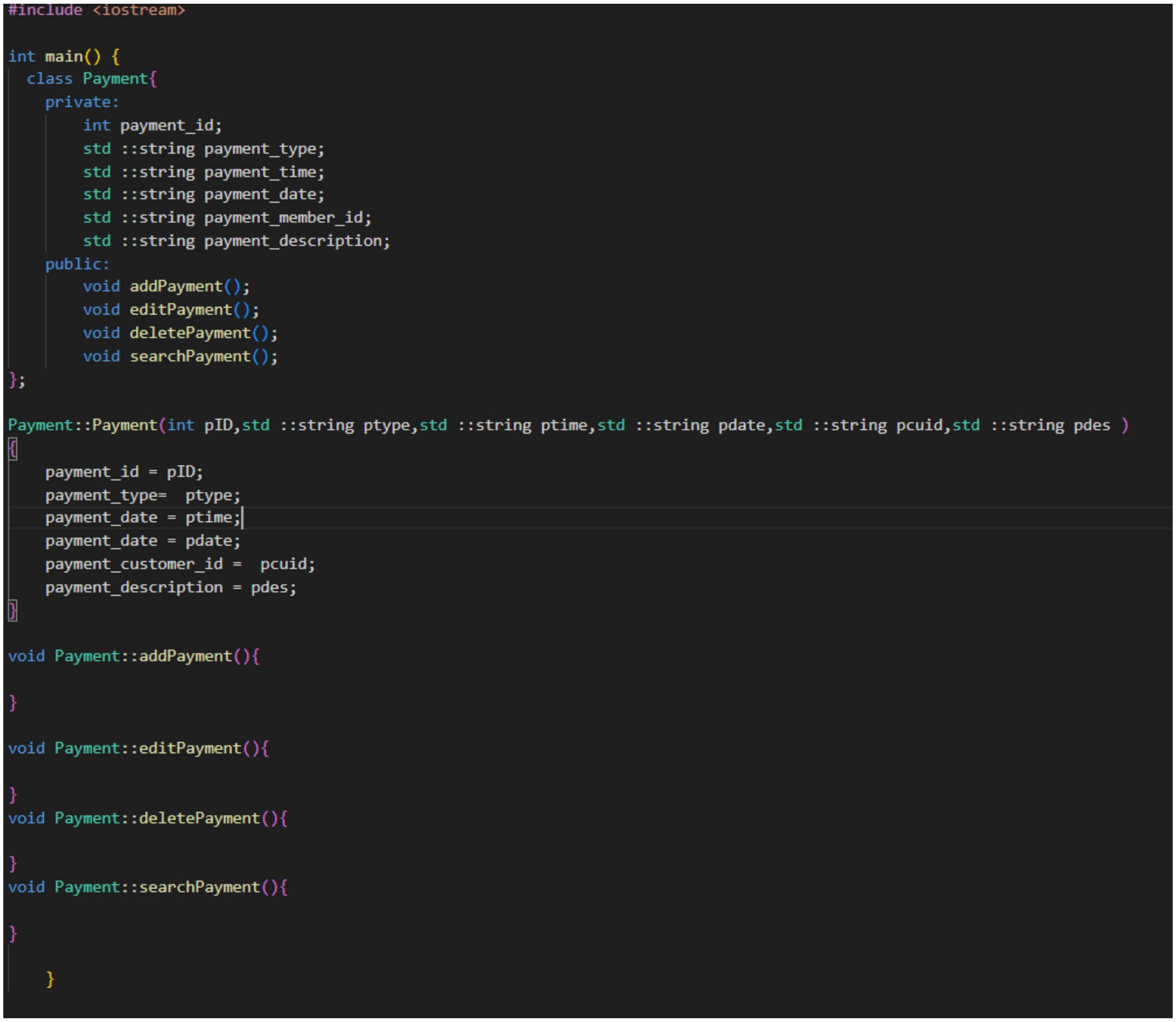
```
#include <iostream>

int main() {
  class Admin{
    private:
        int Admin_id;
        std ::string Admin_email;
        std ::string Admin_address;
        std ::string Admin_name;
        std ::string Admin_password;
        std ::string Admin_dob;


    public:
        void addAdmin();
        void editAdmin();
        void deleteAdmin();
        void searchAdmin();

}

Admin::Admin(){

}
Admin::Admin(int aID,std ::string aemail,std ::string aadres,std ::string aname,std ::string aPass,std ::string adob  )
{
  Admin_id = aID;
  Admin_email=  aemail;
  Admin_address = aadres;
  Admin_name = aname;
  Admin_password = aPass;
  Admin_dob = adob;

}

void Admin::addAdmin(){

}

void Admin::editAdmin()(){

}
void Admin::deleteAdmin(){

}
void Admin::searchAdmin(){

}
Admin::~Admin(){}

}
```

Tour :



```
#include <iostream>

int main() {
  class Tour{
    private:
        int tour_id;
        std ::string tour_type;
        std ::string tour_number;
        std ::string tour_date;
        std ::string tour_amount;
        std ::string tour_description;

    public:
        void addtour();
        void edittour();
        void deletetour();
        void searchtour();
}

Tour::Tour(){

}
Tour::Tour(int tID,std ::string ttype,std ::string tnum,std ::string tDat,std ::string tAmot,std ::string tDes )
{
    tour_id = tID;
    tour_type=  ttype;
    tour_number = tnum;
    tour_date=tDat;
    tour_amount = tAmot;
    tour_description = tDes;

}

void Tour::addtour(){

}

void Tour::edittour(){

}
void Tour::deletetour(){

}
void Tour::searchtour(){

}
Tour::~Tour(){}

}
```

Payment :



```
#include <iostream>

int main() {
  class Payment{
    private:
        int payment_id;
        std ::string payment_type;
        std ::string payment_time;
        std ::string payment_date;
        std ::string payment_member_id;
        std ::string payment_description;
    public:
        void addPayment();
        void editPayment();
        void deletePayment();
        void searchPayment();
};

Payment::Payment(int pID,std ::string ptype,std ::string ptime,std ::string pdate,std ::string pcuid,std ::string pdes )
{
    payment_id = pID;
    payment_type=  ptype;
    payment_date = ptime;
    payment_date = pdate;
    payment_customer_id =  pcuid;
    payment_description = pdes;
}

void Payment::addPayment(){

}

void Payment::editPayment(){

}
void Payment::deletePayment(){

}
void Payment::searchPayment(){

}

    }
```