

PROFILE

string username;  
string email;  
string password;  
string name;  
bool twoFactorOpted;  
bool twoFactorVerified; // authentication

void profileSignup();  
void TwoFactorAuthentication();

AGREEMENT CONTRACTS

bool isComplete;  
int contractID;  
int renterID  
int startDate;  
int endDate;

enterVehicleInfo();  
enterRenterInfo();  
getRates();  
getLocation();  
PaymentInfo :: getPaymentInfo();

Friend of Employee

VEHICLE

string Make;  
string Model;  
int year;  
bool isLuxury;  
int carID;  
string type;  
int locationID;  
int price;  
vector<int> inventory;  
bool status;

void enterCarInfo();  
bool isAvailability();  
void getVehicleInfo();

RENTER

string address;  
int birthDate;  
string paymentMethod;  
int licenseNum; // driver's license  
int phoneNum;  
int currZipCode;

void enterRenterInfo();  
void rentalStatus();  
void viewCurrContracts();  
bool enableGPS();  
void findLocations(currZipCode);  
void changeZipCode(int code);  
void getDirections(locationID);  
void makeRental();  
void paymentMethod();  
void viewHistory(); // bought cars  
PaymentInfo :: setPaymentInfo();  
PaymentInfo :: getPaymentInfo();

EMPLOYEE

string locationID;  
int employeeID;

void viewHistory(location); //sold  
void setLocationID();  
void setUpAgreementContract();  
void viewRentalStatus(Renter a);  
void enterEmployeeInfo();

PAYMENT INFO

string cardType;  
int cardNum;  
int cvv;  
int billingZip;  
string billingAddress;  
int month; // expire  
int day; // expire

void setPaymentInfo();//would include a  
large amount of catches for valid info  
void getPaymentInfo();

RATES

int driverAge;  
int numDaysRented;  
bool isWeekend;  
bool discount;  
bool isMilitary;

void getVehicleInfo();  
void getLocation();  
int calculateRate();

LOCATION

int locationID;  
int maxCapacity;  
int carAmount; //cars at location  
int carList[] //variable array of carIDs

int changeMax(int nexMax); //change  
maxCapacity  
int changeAmount(int newAmount);

SALES ASSOCIATE

int salesTarget;  
int currentSales;

void viewVehicleInfo(Vehicle a);  
void Vehicle :: setAvailability();  
Int Location :: changeAmount();

MANAGER

string department;  
int numOfSalesAssosiates;  
string shift;

void manageEmployee(Sales Associate a);  
bool changeLocation(int newLocationID);

Friend of Vehicle & Location

REGIONAL MANAGER

string region;  
int numOfManagers;  
Int numOfSalesAssociates;

void manageEmployee(Manager a);  
void manageEmployee(Sales Associate a);  
void removeVehicle(Vehicle a);  
void addVehicle();

Friend of Vehicle, Rates & Location