**ManagerList class**

//create singleton instance of List

private static List list;

public ManagerList(){

if(List is null) return new List();

else return list;

}

public bool add(Manager m){

if(m is null) return false;

else {

list.add(m);

return true;

}

return false;  
}

public bool remove(Manager m){

if(m is null) return false;

else {

list.remove(m);

return true;

}

return false;

}

public bool update(Manager m){

if(m is null) return false;

else{

check if m is in the list by m's id

if m is in the list

list.remove(m);

list.add(m);

return true;

else

return false;

}

return false;

}

public bool isEmpty(){

return (list != null && list.size() == 0)

}

public List getMemberList(){

return list;

}

public Manager search(String name){

if the member(checked by name) is in the member list

return member;

else return null;

}

**ProviderReport class**

private Provider p

private Schedule sch

private ProviderList pl

private MemberList ml

public ProviderReport(){

}

public ProviderReport(Schedule sch, Provider p){

this.sch = sch;

this.p = p;  
}

public Provider selectProvider(String name){

if the provider(checked by name or p) is in the provider list

return p;

else return null;  
}

public void runSchedule(){

//email the provider information to the corresponding client

//also, including the date and time of service

//and member list information, including member name, member number

//service code and fee to be paid

ml = new MemberList();

List list = ml.getMemberList();

//at last, calculate the total number of consultations with members

//and total fee for week(up to $99,999.99);

}

**MemberReport class**

private Member m

private Schedule sch

private ServiceList pl

private MemberList ml

public MemberReport(){

}

public MemberReport (Schedule sch, Member m){

this.sch = sch;

this.m = m;  
}

public Member selectMember(String name){

if the member(checked by name or m) is in the member list

return m;

else return null;  
}

public void runSchedule(){

//email the member information to the corresponding client

//also, including the date and time of service

//and service list information, date of service, provider name

//and service name

sl = new ServiceList();

List list = ml.getServiceList();

}

**Account Class**

private String name

private String id;

private String streeAddress;

private String state

private String city

private String zipCode

public Account()

public Account(String id, String name, String streeAddress, String state, String city, String zipCode)

public setName(String name) void

public getNam() String

public setId(int id) void

public getId() int

public setState(String state)

public getState() String

public getCity() String

public setCity(String city)

public getStreetAddress()

public setStreetAddress(String streetAddress)

**Manager class extends Account**

{

//all fields should already be defined in the account class

}