**UserInterface**

Public void ProviderMenu(){

Do{

Int Choice;

//Ask user to input the choice for menu

Switch(Choice){

//print the menu option for Provider

}

}while(Quit)

}

Public void ManagerMenu(){

Do{

Int Choice;

//Ask user to input the choice for menu

Switch(Choice){

//print the menu option for Provider

}

}while(Quit)

}

Public void AdminMenu(){

Do{

Int Choice;

//Ask user to input the choice for menu

Switch(Choice){

//print the menu option for Admin

}

}while(Quit)

}

Public void Login(){

Int inputNumber;

//Display a message and ask user to login by using their ID

If(//inputNumber == MemberID){

MemberMenu();

}

If(//inputNumber == ProviderID){

ProviderMenu();

}

If(//inputNumber == ManagerID){

ManagerMenu();

}

If(inputNumber == AdminID){

AdminMenu();

}

}

Public void Process(){

This.Login();

}

Public void GetWeeklyMemberReports(){

//Declare new list to hold the list report result

List MemberReport = new LinkList();

MemberReport = application. GetWeeklyMemberReports();

}

Public void GetWeeklyProviderReports(){

//Declare new list to hold the list report result

List ProviderReport = new LinkList();

ProviderReport = application. GetWeeklyProviderReports();

}

Public void GetWeeklyEFTReports(){

//Declare new list to hold the list report result

List EFTReport = new LinkList();

EFTReport = application. GetWeeklyEFTReports();

}

Public void GetMemberReport(){

Int memberID;

Date startDate, endData;

//Ask user to input the memberID, and startDate and endDate

application. GetMemberReport(memberID, startDate, endDate)

}

Public void GetEFTReport(){

Int providerID;

Date startDate, endData;

//Ask user to input the memberID, and startDate and endDate

application. GetMemberReport(memberID, startDate, endDate)

}

Public void UpdateMemberReportSchedule(){

Int weekday;

Time time;

//Ask user to input the weekday and time

If(application. UpdateMemberReportSchedule(weekday,time) ){

//Generate the successful message

}

Else{

//Generate the error message

}

}

Public void UpdateProviderReportSchedule() {

Int weekday;

Time time;

//Ask user to input the weekday and time

if(Application.UpdateProviderReportSchedule(weekDay, time)){

//Generate the successful message

}

else{

//Generate the error message

}

}

Public void UpdateEFTReportSchedule(){

Int weekday;

Time time;

//Ask user to input the weekday and time

If(Application.UpdateEFTReportSchedule(weekDay, time))

//Generate the successful message

}

else{

//Generate the error message

}

}

Public void VerifyWeeklyProviderReportServices() {

Int providerID;

Date startDate, endDate;

//ask user to input the data

If(Application. VerifyWeeklyProviderReportServices(providerID , startDate , endDate)){

//Generate the successful message

}

Else{

//Generate the error message

}

}

Public void VerifyWeeklyProviderReportFees(){

Int providerID;

Date startDate, endDate;

//ask user to input the data

If(Application. VerifyWeeklyProviderReportFees(providerID , startDate , endDate)){

//Generate the successful message

}

Else{

//Generate the error message

}

}

Public void AddServiceRecord() : Boolean{

ServiceRecord newServiceRecord = NULL;

//Ask user to input the information about the serviceRecord and check the input information

If(Application.AddServiceRecord(newServiceRecord)){

//generate successful message

}

Else{

//generate error message

}

}

Public void AddService(){

Service newService = NULL;

//Ask user to input the information about the service and check the input information

If(Application.AddService (newService)){

//generate successful message

}

else{

//generate error message

}

}

Public void GetAllService (){

List services = new (LinkLIst);

service = application.GetAllService ();

//output service

}

Public void UpdateService () {

Service updatedService = NULL;

//ask user to input the information about service and check the input

If(application.UpdateService (updatedService)){

//generate the successful message

}

Else{

//generate the error message

}

}

Public void DeleteService(){

Int serviceCode;

//Ask user to input the serviceCode

If(application.deleteService(serviceCode)){

//generate the successful message

}

Else{

//generate the error message

}

}

Public void GetMember(){

Member member = NULL;

Int memberID

//ask user to input the member ID

Member = application.GetMember(memberID);

//output member’s information

}

Public void GetProvider(){

Provider provider = NULL;

Int provider ID

//ask user to input the provider ID

Provider = application.GetProvider(provider ID);

//output provider ‘s information

}

Public void GetManager(){

Manager manager = NULL;

Int managerID;

//ask user to input the manager ID

Manager = application.GetManager(managerID);

//output Manager’s information

}

Public void GetAdmin(){

Admin admin= NULL;

Int adminID;

//ask user to input the member ID

Admin = application.GetAdmin(adminID);

//output admin’s information

}

Public void AddMember(){

Member member = NULL;

//ask user to input the member’s information

If(application.AddMember(member)){

//generate the successful message

}

Else{

//generate the error message

}

}

Public void AddProvider(){

Provider provider = NULL;

//ask user to input the provider‘s information

If(application.AddProvider(provider)){

//generate the successful message

}

Else{

//generate the error message

}

}

Public void AddManager(){

Manager manager = NULL;

//ask user to input the manager’s information

If(application.AddManager(manager)){

//generate the successful message

}

Else{

//generate the error message

}

}

Public void AddAdmin(){

Admin admin = NULL;

//ask user to input the admin’s information

If(application.AddAdmin(admin)){

//generate the successful message

}

Else{

//generate the error message

}

}

Public void UpdateMember(){

int memberID;

//ask user to input member ID;

Member member = application.GetMember(memberID);

if(Application.UpdateMember(member)){

//generate successful message

}

Else{

//generate error message

}

}

Public void UpdateProvider(){

int providerID;

//ask user to input provider ID;

Provider provider = application.GetProvider(providerID);

if(Application.UpdateProvider(provider)){

//generate successful message

}

Else{

//generate error message

}

}

Public void UpdateManager(){

int managerID;

//ask user to input manager ID;

Manager manager = application.GetManager(managerID);

if(Application.UpdateMember(manager)){

//generate successful message

}

Else{

//generate error message

}

}

Public void UpdateAdmin(){

int adminID;

//ask user to input admin ID;

Admin admin = application.GetAdmin(adminID);

if(Application.UpdateAdmin(admin)){

//generate successful message

}

Else{

//generate error message

}

}

Public void DeleteMember(){

int memberID;

//ask user to input member ID;

if(Application.DeleteMember(memberID)){

//generate successful message

}

Else{

//generate error message

}

}

Public void DeleteProvider(){

int providerID;

//ask user to input provider ID;

if(Application.DeleteProvider(providerID)){

//generate successful message

}

Else{

//generate error message

}

}

Public void DeleteManager(){

int managerID;

//ask user to input manager ID;

if(Application.DeteleManager(managerID)){

//generate successful message

}

Else{

//generate error message

}

}

Public void DeleteAdmin(){

int adminID;

//ask user to input admin ID;

if(Application.DeleteAdmin(adminID)){

//generate successful message

}

Else{

//generate error message

}

}

Public void ValidateMember(){

int memberID;

//ask user to input memberID

If(Application.ValidateMember(memberID ) == 1){

//output the status of the member

}

If(Application.ValidateMember(memberID ) == 2){

//output the status of the member

}

If(Application.ValidateMember(memberID ) == 3){

//output the status of the member

}

}