**CLASS: PizzaAnonymousApplication**

**Summary:**

**Elements:**

1. private ManageAccount manageAccount
2. private ManageReport manageReport
3. private ManageService manageService

**Functions:**

1. public List<MemberReport> GetWeeklyMemberReports()
2. public List<ProviderReport> GetWeeklyProviderReports()
3. public List<EFTReport> GetWeeklyEFTReports()
4. public MemberReport GetMemberReport(integer memberID, date startDate, date endDate)
5. public EFTReport GetEFTReport(integer providerID, date startDate, date endDate)
6. public boolean UpdateMemberReportSchedule(integer weekday, time time)
7. public boolean UpdateProviderReportSchedule(integer weekday, time time)
8. public boolean UpdateEFTReportSchedule(integer weekday, time time)
9. public boolean VerifyWeeklyProviderReportServices  
   (integer providerID, date startDate, date endDate)
10. public boolean VerifyWeeklyProviderReportFees  
    (integer providerID, date startDate, date endDate)
11. public boolean AddServiceRecord(ServiceRecord newServiceRecord)
12. public boolean AddService(Service newService)
13. public List<Service> GetAllServices()
14. public boolean UpdateService(Service updatedService)
15. public boolean DeleteService(integer serviceCode)
16. public Member GetMember(integer memberID)
17. public Provider GetProvider(integer providerID)
18. public Manager GetManager(integer managerID)
19. public Admin GetAdmin(integer adminID)
20. public boolean AddMember(Member member)
21. public boolean AddProvider(Provider provider)
22. public boolean AddManager(Manager manager)
23. public boolean AddAdmin(Admin admin)
24. public boolean UpdateMember(Member member)
25. public boolean UpdateProvider(Provider provider)
26. public boolean UpdateManager(Manager manager)
27. public boolean UpdateAdmin(Admin admin)
28. public boolean DeleteMember(integer memberID)
29. public boolean DeleteProvider(integer providerID)
30. public boolean DeleteManager(integer managerID)
31. public boolean DeleteAdmin(integer adminID)
32. public integer ValidateMember(integer memberID)

**Detail:**

1. **public List<MemberReport> GetWeeklyMemberReports()**

Module name : **GetWeeklyMemberReports**

Module type : Function

Return type : **List<MemberReport>**

Input arguments : None

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageReport.GetWeeklyMemberReports()

Narrative : This method returns weekly member reports.

public List<MemberReport> GetWeeklyMemberReports()

{

var reports = new List<MemberReport>();

try

{

Reports = manageReport.GetWeeklyMemberReports();

}

catch (Exception e)

{

//record exception

reports = null;

}

return reports;

}

1. **public List<ProviderReport> GetWeeklyProviderReports()**

Module name : **GetWeeklyProviderReports**

Module type : Function

Return type : **List<ProviderReport>**

Input arguments : None

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageReport.GetWeeklyProviderReports()

Narrative : This method returns weekly provider reports.

public List<ProviderReport> GetWeeklyProviderReports()

{

var reports = new List<MemberReport>();

try

{

Reports = manageReport.GetWeeklyProviderReports();

}

catch (Exception e)

{

//record exception

reports = null;

}

return reports;

}

1. **public List<EFTReport> GetWeeklyEFTReports()**

Module name : **GetWeeklyEFTReports**

Module type : Function

Return type : **List<EFTReport>**

Input arguments : None

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageReport.GetWeeklyEFTReports()

Narrative : This method returns weekly EFT reports.

public List<EFTReport> GetWeeklyEFTReports()

{

var reports = new List<EFTReport>();

try

{

Reports = manageReport.GetWeeklyEFTReports();

}

catch (Exception e)

{

//record exception

reports = null;

}

return reports;

}

1. **public MemberReport GetMemberReport  
   (integer memberID, date startDate, date endDate)**

Module name : **GetMemberReport**

Module type : Function

Return type : **MemberReport**

Input arguments : memberID : integer

startDate : date

endDate : date

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageReport.GetMemberReport

(integer memberID, date startDate, date endDate)

Narrative : This method returns a member report.

public MemberReport GetMemberReport(integer memberID,

date startDate, date endDate)

{

var report = new MemberReport();

try

{

report = manageReport.GetMemberReport(memberID, startDate, endDate);

}

catch (Exception e)

{

//record exception

report = null;

}

return report;

}

1. **public EFTReport GetEFTReport  
   (integer providerID, date startDate, date endDate)**

Module name : **GetEFTReport**

Module type : Function

Return type : **EFTReport**

Input arguments : providerID : integer

startDate : date

endDate : date

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageReport.GetEFTReport

(integer providerID, date startDate, date endDate)

Narrative : This method returns an EFT report.

public EFTReport GetEFTReport(integer providerID,

date startDate, date endDate)

{

var report = new EFTReport();

try

{

report = manageReport.GetMemberReport(providerID,

startDate, endDate);

}

catch (Exception e)

{

//record exception

report = null;

}

return report;

}

1. **public boolean UpdateMemberReportSchedule(integer weekDay, time time)**

Module name : **UpdateMemberReportSchedule**

Module type : Function

Return type : **boolean**

Input arguments : weekDay : integer

time : time

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageReport.UpdateMemberReportSchedule

(integer weekDay, time time)

Narrative : This method updates member report schedule.

public boolean UpdateMemberReportSchedule(integer weekDay, time time)

{

var success = false;

try

{

success = manageReport.UpdateMemberReportSchedule(weekday, time);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean UpdateProviderReportSchedule(integer weekDay, time time)**

Module name : **UpdateMemberProviderSchedule**

Module type : Function

Return type : **boolean**

Input arguments : weekDay : integer

time : time

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageReport.UpdateProviderReportSchedule

(integer weekDay, time time)

Narrative : This method updates provider report schedule.

public boolean UpdateProviderReportSchedule(integer weekDay, time time)

{

var success = false;

try

{

success = manageReport.UpdateProviderReportSchedule(weekday, time);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean UpdateEFTReportSchedule(integer weekDay, time time)**

Module name : **UpdateEFTReportSchedule**

Module type : Function

Return type : **boolean**

Input arguments : weekDay : integer

time : time

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageReport.UpdateEFTReportSchedule

(integer weekDay, time time)

Narrative : This method updates EFT report schedule.

public boolean UpdateEFTReportSchedule(integer weekDay, time time)

{

var success = false;

try

{

success = manageReport.UpdateEFTReportSchedule(weekday, time);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean VerifyWeeklyProviderReportServices  
   (integer providerID, date startDate, date endDate)**

Module name : **VerifyWeeklyProviderReportServices**

Module type : Function

Return type : **boolean**

Input arguments : providerID : integer

startDate : date

endDate : date

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageReport.VerifyWeeklyProviderReportServices

(integer providerID, date startDate, date endDate)

Narrative : This method verifies provider report services.

public boolean VerifyWeeklyProviderReportServices

(integer providerID, date startDate, date endDate)

{

var success = false;

try

{

success = manageReport.VerifyWeeklyProviderReportServices

(providerID, startDate, endDate);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean VerifyWeeklyProviderReportFees  
   (integer providerID, date startDate, date endDate)**

Module name : **VerifyWeeklyProviderReportFees**

Module type : Function

Return type : **boolean**

Input arguments : providerID : integer

startDate : date

endDate : date

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageReport.VerifyWeeklyProviderReportFees

(integer providerID, date startDate, date endDate)

Narrative : This method verifies provider report fees.

public boolean VerifyWeeklyProviderReportFees

(integer providerID, date startDate, date endDate)

{

var success = false;

try

{

success = manageReport.VerifyWeeklyProviderReportFees

(providerID, startDate, endDate);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean AddServiceRecord(ServiceRecord newServiceRecord)**

Module name : **AddServiceRecord**

Module type : Function

Return type : **boolean**

Input arguments : newService : Service

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.AddServiceRecord(ServiceRecord newServiceRecord)

Narrative : This method adds a new service.

public boolean AddServiceRecord(ServiceRecord newServiceRecord)

{

var success = false;

try

{

success = manageService.AddServiceRecord(newServiceRecord);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean AddService(Service newService)**

Module name : **AddService**

Module type : Function

Return type : **boolean**

Input arguments : newService : Service

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.AddService(Service newService)

Narrative : This method adds a new service.

public boolean AddService(Service newService)

{

var success = false;

try

{

success = manageService.AddService(newServiceInfo);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public List<Service> GetAllServices()**

Module name : **GetAllServices**

Module type : Function

Return type : **List<Service>**

Input arguments : None

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.GetAllServices()

Narrative : This method returns all service info.

public List<Service> GetAllServices()

{

var services = new List<Service>();

try

{

services = manageService.GetAllServices();

}

catch (Exception e)

{

//record exception

services = false;

}

return services;

}

1. **public boolean UpdateService(Service updatedService)**

Module name : **UpdateService**

Module type : Function

Return type : **boolean**

Input arguments : updatedService : Service

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.UpdateService(Service updatedService)

Narrative : This method updates an existing service info.

public boolean UpdateService(Service updatedService)

{

var success = false;

try

{

success = manageService.UpdateService(updatedService);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean DeleteService(integer serviceCode)**

Module name : **DeleteService**

Module type : Function

Return type : **boolean**

Input arguments : serviceCode : integer

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.DeleteService(integer serviceCode)

Narrative : This method updates an existing service info.

public boolean DeleteService(integer serviceCode)

{

var success = false;

try

{

success = manageService.DeleteService(serviceCode);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public Member GetMember(integer memberID)**

Module name : **GetMember**

Module type : Function

Return type : **Member**

Input arguments : memberID : integer

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageAccount.GetMember(integer memberID)

Narrative : This method returns an existing member.

public Member GetMember(integer memberID)

{

var account = new Member();

try

{

account = manageAccount.GetMember(memberID);

}

catch (Exception e)

{

//record exception

account = null;

}

return account;

}

1. **public Member GetProvider(integer providerID)**

Module name : **GetProvider**

Module type : Function

Return type : **Provider**

Input arguments : providerID : integer

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageAccount.GetProvider(integer providerID)

Narrative : This method returns an existing provider.

public Member GetProvider(integer providerID)

{

var account = new Provider();

try

{

account = manageAccount.GetProvider(providerID);

}

catch (Exception e)

{

//record exception

account = null;

}

return account;

}

1. **public Member GetManager(integer managerID)**

Module name : **GetManager**

Module type : Function

Return type : **Manager**

Input arguments : managerID : integer

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageAccount.GetManager(integer managerID)

Narrative : This method returns an existing manager.

public Member GetManager(integer managerID)

{

var account = new Manager();

try

{

account = manageAccount.GetManager(managerID);

}

catch (Exception e)

{

//record exception

account = null;

}

return account;

}

1. **public Member GetAdmin(integer adminID)**

Module name : **GetAdmin**

Module type : Function

Return type : **Admin**

Input arguments : adminID : integer

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageAccount.GetAdmin(integer adminID)

Narrative : This method returns an existing admin.

public Member GetAdmin(integer adminID)

{

var account = new Admin();

try

{

account = manageAccount.GetAdmin(adminID);

}

catch (Exception e)

{

//record exception

account = null;

}

return account;

}

1. **public boolean AddMember(Member newMember)**

Module name : **AddMember**

Module type : Function

Return type : **boolean**

Input arguments : newMember : Member

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.AddMember(Member newMember)

Narrative : This method adds a new member.

public boolean AddMember(Member newMember)

{

var success = false;

try

{

success = manageService.AddMember(newMember);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean AddProvider(Provider newProvider)**

Module name : **AddProvider**

Module type : Function

Return type : **boolean**

Input arguments : newProvider : Provider

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.AddProvider(Member newProvider)

Narrative : This method adds a new provider.

public boolean AddProvider(Member newProvider)

{

var success = false;

try

{

success = manageService.AddProvider(newProvider);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean AddManager(Member newManager)**

Module name : **AddManager**

Module type : Function

Return type : **boolean**

Input arguments : newManager : Manager

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.AddManager(Member newManager)

Narrative : This method adds a new manager.

public boolean AddManager(Member newManager)

{

var success = false;

try

{

success = manageService.AddManager(newManager);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean AddAdmin(Member newAdmin)**

Module name : **AddAdmin**

Module type : Function

Return type : **boolean**

Input arguments : newAdmin : Admin

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.AddAdmin(Member newAdmin)

Narrative : This method adds a new admin.

public boolean AddAdmin(Member newAdmin)

{

var success = false;

try

{

success = manageService.AddAdmin(newAdmin);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean UpdateMember(Member updatedMember)**

Module name : **UpdateMember**

Module type : Function

Return type : **boolean**

Input arguments : updatedMember : Member

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.UpdateMember(Member updatedMember)

Narrative : This method updates an existing member.

public boolean UpdateMember(Member updatedMember)

{

var success = false;

try

{

success = manageService.UpdateMember(updatedMember);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean UpdateProvider(Member updatedProvider)**

Module name : **UpdateProvider**

Module type : Function

Return type : **boolean**

Input arguments : updatedProvider : Provider

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.UpdateProvider(Member updatedProvider)

Narrative : This method updates an existing provider.

public boolean UpdateProvider(Member updatedProvider)

{

var success = false;

try

{

success = manageService.UpdateProvider(updatedProvider);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean UpdateManager(Member updatedManager)**

Module name : **UpdateManager**

Module type : Function

Return type : **boolean**

Input arguments : updatedManager : Manager

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.UpdateManager(Member updatedManager)

Narrative : This method updates an existing manager.

public boolean UpdateManager(Member updatedManager)

{

var success = false;

try

{

success = manageService.UpdateManager(updatedManager);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean UpdateAdmin(Member updatedAdmin)**

Module name : **UpdateAdmin**

Module type : Function

Return type : **boolean**

Input arguments : updatedAdmin : Admin

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.UpdateAdmin(Member updatedAdmin)

Narrative : This method updates an existing admin.

public boolean UpdateAdmin(Member updatedAdmin)

{

var success = false;

try

{

success = manageService.UpdateAdmin(updatedAdmin);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean DeleteMember(integer memberID)**

Module name : **DeleteMember**

Module type : Function

Return type : **boolean**

Input arguments : memberID : integer

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.DeleteMember(integer memberID)

Narrative : This method deletes an existing member.

public boolean DeleteMember(integer memberID)

{

var success = false;

try

{

success = manageService.DeleteMember(memberID);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean DeleteProvider(integer providerID)**

Module name : **DeleteProvider**

Module type : Function

Return type : **boolean**

Input arguments : providerID : integer

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.DeleteProvider(integer providerID)

Narrative : This method deletes an existing provider.

public boolean DeleteProvider(integer providerID)

{

var success = false;

try

{

success = manageService.DeleteProvider(providerID);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean DeleteManager(integer managerID)**

Module name : **DeleteManager**

Module type : Function

Return type : **boolean**

Input arguments : managerID : integer

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.DeleteManager(integer managerID)

Narrative : This method deletes an existing manager.

public boolean DeleteManager(integer managerID)

{

var success = false;

try

{

success = manageService.DeleteManager(managerID);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean DeleteAdmin(integer adminID)**

Module name : **DeleteAdmin**

Module type : Function

Return type : **boolean**

Input arguments : adminID : integer

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.DeleteAdmin(integer adminID)

Narrative : This method deletes an existing admin.

public boolean DeleteAdmin(integer adminID)

{

var success = false;

try

{

success = manageService.DeleteAdmin(adminID);

}

catch (Exception e)

{

//record exception

success = false;

}

return success;

}

1. **public boolean ValidateMember(integer memberID)**

Module name : **ValidateMember**

Module type : Function

Return type : **boolean**

Input arguments : memberID : integer

Output arguments : None

Error messages : None

Files accessed : None

Files changed : None

Modules called : ManageService.ValidateMember(integer memberID)

Narrative : This method returns status of an existing member.

public boolean ValidateMember(integer memberID)

{

var status = new int?();

try

{

status = manageService.ValidateMember(memberID);

}

catch (Exception e)

{

//record exception

status = null;

}

return status;

}