**TP AirportManagement**

**Partie 7: Service générique et Services spécifiques**

**7.1. Implémentation du service générique**

* 1. Sous le dossier **Interfaces** du projet **AM.ApplicationCore**, créer l’interface générique « **IService** ».

public interface IService<T> where T : class

{

void Add(T entity);

void Update(T entity);

void Delete(T entity);

void Delete(Expression<Func<T, bool>> where);

T GetById(params object[] keyValues);

IEnumerable<T> GetMany(Expression<Func<T, bool>> where = null);

T Get(Expression<Func<T, bool>> where);

void Commit();

}

* 1. Sous le dossier **Services** du projet **AM.ApplicationCore**, créer la classe **Service** générique qui implémente l’interface **IService**.

public class Service<T> : IService<T> where T : class

{

private IGenericRepository<T> repository;

private IUnitOfWork unitOfWork;

public Service(IUnitOfWork unitOfWork)

{

this.repository = unitOfWork.Repository<T>();

this.unitOfWork = unitOfWork;

}

public virtual void Add(T entity)

{

repository.Add(entity);

}

public virtual void Update(T entity)

{

repository.Update(entity);

}

public virtual void Delete(T entity)

{

repository.Delete(entity);

}

public virtual void Delete(Expression<Func<T, bool>> where)

{

repository.Delete(where);

}

public virtual T GetById(params object[] keyValues)

{

return repository.GetById(keyValues);

} public virtual IEnumerable<T> GetMany(Expression<Func<T, bool>> filter = null)

{

return repository.GetMany(filter);

}

public virtual T Get(Expression<Func<T, bool>> where)

{

return repository.Get(where);

}

public void Commit()

{

try

{

unitOfWork.Save();

}

catch (Exception ex)

{

throw;

}

}

}

**7.2. Implémentation du service spécifique**

* 1. Modifier toutes les interfaces de Services Spécifiques, I**ServicePlane** et I**ServicePassenger,** afin qui ils héritent de l’interface du service générique « I**Service** ».

namespace AM.ApplicationCore.Interfaces

{

public interface IServicePlane : IService<Plane>

{

}

}

namespace AM.ApplicationCore.Interfaces

{

public interface IServicePassenger:IService<Passenger>

{

}

}

* 1. Modifier toutes les classes de Services Spécifiques, **ServicePlane** et **ServicePassenger,** afin qui ils héritent de la classe du service générique « **Service** ».

public class ServicePlane :Service<Plane>, IServicePlane

{

public ServicePlane(IUnitOfWork unitOfWork) : base(unitOfWork)

{

}

}

public class ServicePassenger :Service<Passenger>, IServicePassenger

{

public ServicePassenger(IUnitOfWork unitOfWork) : base(unitOfWork)

{

}

}