```
using Meditech.App.CommonService.Dtos;
using Meditech.App.CommonService.ViewModels;
using Meditech.Core.Common.Repositories;
using Meditech.Core.Infrastructure.Database;
using Meditech.Lib.CommonShare;

namespace Meditech.App.CommonService.Services
{
    public interface IAttachmentService
    {
        List<AttachmentDto> GetByObjectId(string id, string type="", string? search="");
        ReturnCode Add(List<AttachmentAddViewModel> model, Guid userId);
        ReturnCode Update(AttachmentUpdateViewModel model, Guid userId);
        ReturnCode Delete(Guid id, Guid userId);

    }
    public class AttachmentService : IAttachmentService
    {
        private readonly IAttachmentRepository _attachmentRepository;
        private Core.Auth.Entities.IUserRepository _userRepository;
        private CommonUnitOfWork _unitOfWork;
        public AttachmentService(IAttachmentRepository attachmentRepository,
                                CommonUnitOfWork unitOfWork,
                                 Core.Auth.Entities.IUserRepository userRepository)
        {
            _attachmentRepository = attachmentRepository;
            _unitOfWork = unitOfWork;
            _userRepository = userRepository;
        }
        public List<AttachmentDto> GetByObjectId(string id, string type, string? search)
        {
            var query = (from attachment in _attachmentRepository.Table
                        // join user in _userRepository.Table on attachment.CreatedById equals user.Id
                        where attachment.ObjectId == id && attachment.IsDeleted == false &&
                        (string.IsNullOrEmpty(type) || attachment.ObjectType == type)
                        orderby attachment.CreatedTime descending
                        select new AttachmentDto
                        {
                            Id = attachment.Id,
                            Description = attachment.Description,
                            FileName = attachment.FileName,
                            FileSizeKb = attachment.FileSizeKb,
                            FileType = attachment.FileType,
                            ObjectId = attachment.ObjectId,
                            FileUrl = attachment.Url,
                            ObjectType = attachment.ObjectType,
                            // CreatedByName = user.FullName??"",
                            CreatedById = attachment.CreatedById,
                            CreatedTime = attachment.CreatedTime,
                        });

            if (!string.IsNullOrEmpty(search))
            {
                string keyword = search.Trim().ToLower();
                query = query.Where(x => (x.Description + "").ToLower().Contains(keyword) || (x.FileName + "").ToLower().Contains(keyword));
            }

            var data = query.ToList();

            var userIds = data.Select(t => t.CreatedById).ToList();
            var userData = _userRepository.GetMulti(t => userIds.Contains(t.Id)).Select(x => new { x.Id, x.FullName}).ToList();
            foreach (var item in data)
            {
                var user = userData.Find(t => t.Id == item.CreatedById);
                if(user != null)
                {
                    item.CreatedByName = user.FullName;
                }
            }
            return data;
        }

        public ReturnCode Add(List<AttachmentAddViewModel> models, Guid userId)
        {
            ReturnCode ret = new ReturnCode();


            if (models == null || models.Count < 1)
            {
                ret.Code = ErrorCode.Common_NotFound;
                ret.Description = "Không có dữ liệu";
                return ret;
            }
            foreach (var model in models)
            {
                var existed = _attachmentRepository.GetSingleByCondition(t => t.ObjectId == model.ObjectId && t.Url == model.FileUrl);
                if (existed != null)
                {
                    ret.Code = ErrorCode.Common_Existed;
                    ret.Description = "File đính kèm đã tồn tại";
                    return ret;
                }
                if (string.IsNullOrEmpty(model.FileUrl))
                {
                    ret.Code = ErrorCode.Common_Invalid;
                    ret.Description = "File URL trống, xin vui lòng upload lại";
                    return ret;
                }
                var attachment = new Meditech.Core.Common.DBModels.Attachment();
                attachment.Id = Guid.NewGuid();
                attachment.ObjectId = model.ObjectId;
                attachment.ObjectType = model.ObjectType;
                attachment.CreatedById = userId;
                attachment.CreatedTime = DateTime.UtcNow;
                attachment.Description = model.Description;
                attachment.FileName = model.FileName;
                attachment.FileSizeKb = model.FileSizeKb;
                attachment.Url = model.FileUrl;
                attachment.FileType = model.FileType;
                attachment.IsDeleted = false;
                _attachmentRepository.Insert(attachment);
            }
            _unitOfWork.Save();
            return ret;
        }
        public ReturnCode Update(AttachmentUpdateViewModel model, Guid userId)
        {
            ReturnCode ret = new ReturnCode();
            var attachment = _attachmentRepository.GetById(model.Id);
            if (attachment == null)
            {
                ret.Code = ErrorCode.Common_NotFound;
                ret.Description = "Dữ liệu đính kèm không tồn tại";
                return ret;
            }
            attachment.Description = model.Description;
            attachment.ModifiedById = userId;
            attachment.ModifiedTime = DateTime.UtcNow;
            _attachmentRepository.Update(attachment);
            _unitOfWork.Save();

            return ret;
        }
        public ReturnCode Delete(Guid id, Guid userId)
        {
            ReturnCode ret = new ReturnCode();
            var attachment = _attachmentRepository.GetById(id);
            if (attachment == null)
            {
                ret.Code = ErrorCode.Common_NotFound;
                ret.Description = "Dữ liệu đính kèm không tồn tại";
                return ret;
            }
            attachment.IsDeleted = true;
            attachment.ModifiedTime = DateTime.UtcNow;
            attachment.ModifiedById = userId;
            _attachmentRepository.Update(attachment);
            _unitOfWork.Save();
            return ret;
        }
    }
}
```