

```

using Meditech.App.Crm.Services;
using Meditech.App.Crm.ViewModels;
using Meditech.Core.Auth.Authorization;
using Meditech.Core.Auth.Entities;
using Microsoft.AspNetCore.Mvc;

namespace Meditech.App.Crm.Controllers
{
    [MediAuthorize]
    [ApiController]
    [Route("{culture:culture}/api/v{version:apiVersion}/{controller}")]
    [ApiVersion("1.0")]
    public class NotificationController : ControllerBase
    {
        private readonly ILogger<NotificationController> _logger;
        private readonly INotificationService _notificationService;
        public NotificationController(ILogger<NotificationController> logger,
            INotificationService NotificationService)
        {
            _logger = logger;
            _notificationService = NotificationService;
        }

        [HttpPost]
        [Route("search-grid")]
        public IActionResult SearchGrid(SearchGridNotificationViewModel model)
        {
            try
            {
                var requestUser = (BaseUser)Request.HttpContext.Items["User"];
                if (requestUser == null) return Forbid();
                var data = _notificationService.SearchGrid(model);
                return Ok(new { Data = data });
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, $"SearchGrid error. ");
                return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
            }
        }

        [HttpPost]
        [Route("mobile/search")]
        public IActionResult SearchMobile([FromBody] NotificationQuerySearch model)
        {
            try
            {
                var requestUser = (BaseUser)Request.HttpContext.Items["User"];
                if (requestUser == null) return Forbid();
                var data = _notificationService.SearchMobile(model, requestUser.Id);
                return Ok(new { Data = data });
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, "SearchMobile exception");
                return StatusCode(StatusCodes.Status500InternalServerError);
            }
        }

        [HttpGet]
        [Route("read-all")]
        public IActionResult ReadAll()
        {
            try
            {
                var requestUser = (BaseUser)Request.HttpContext.Items["User"];
                if (requestUser == null) return Forbid();
                _notificationService.ReadAll(requestUser.Id);
                return Ok();
            }
            catch (Exception ex)

```

```

    {
        _logger.LogError(ex, "ReadAll exception");
        return StatusCode(StatusCodes.Status500InternalServerError);
    }
}

[HttpGet]
[Route("delete-all")]
public IActionResult DeleteAll()
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();
        _notificationService.DeleteAllNo(requestUser.Id);
        return Ok();
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, "ReadAll exception");
        return StatusCode(StatusCodes.Status500InternalServerError);
    }
}

[HttpGet]
[Route("mark-read/{notiId}")]
public IActionResult MarkRead(Guid notiId)
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();
        _notificationService.ReadNotification(notiId, requestUser.Id);
        return Ok();
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, "MarkRead exception");
        return StatusCode(StatusCodes.Status500InternalServerError);
    }
}

[HttpGet]
[Route("get-by-id")]
public IActionResult GetNotificationById(Guid id)
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();
        var data = _notificationService.getNotificationById(id);
        return Ok(new { Data = data });
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"GetAllNotification error. ");
        return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
    }
}

[HttpPost]
public IActionResult AddNewNotification(NotificationViewModel model)
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();
        var data = _notificationService.AddNewNotification(model, requestUser.Id);
        if (data.IsOk) return Ok(new { Data = data });
        return StatusCode(600, data);
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"SearchGrid error. ");
    }
}

```

