

```

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

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

        [HttpPost]
        [Route("search-grid")]

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

        [HttpGet]
        [Route("get-all")]
        [MediAllowAnonymous]
        public IActionResult GetAllHospital()
        {
            try
            {
                var data = _hospitalManagementService.getAllHospital();
                return Ok(new { Data = data });
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, $"GetAllHospital error. ");
                return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
            }
        }

        [HttpGet]
        [Route("get-by-id")]
        public IActionResult GetHospitalById(Guid id)
        {
            try
            {
                var requestUser = (BaseUser)Request.HttpContext.Items["User"];
                if (requestUser == null) return Forbid();
                var data = _hospitalManagementService.getHospitalById(id);
            }
        }
    }
}

```

```

        return Ok(new { Data = data });
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"GetHospitalById error. ");
        return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
    }
}

[HttpPost]
public IActionResult AddNewHospitalManagement(HospitalManagementViewModel model)
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();
        var data = _hospitalManagementService.AddNewHospitalManagement(model, requestUser.Id);
        if (data.IsOk) return Ok(new { Data = data });
        return StatusCode(600, data);
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"AddNew error. ");
        return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
    }
}

[HttpPut]
public IActionResult UpdateHospitalManagement(UpdateHospitalManagementViewModel model)
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();
        var data = _hospitalManagementService.UpdateHospitalManagement(model);
        if (data.IsOk) return Ok(new { Data = data });
        return StatusCode(600, data);
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"Update error. ");
        return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
    }
}

[HttpDelete]
public IActionResult DeleteHospitalManagement(Guid id)
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();
        var data = _hospitalManagementService.DeleteHospitalManagement(id);
        if (data.IsOk) return Ok(new { Data = data });
        return StatusCode(600, data);
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"Delete error. ");
        return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
    }
}

[HttpPut]
[Route("change-state")]
public IActionResult StatusHospitalManagement(UpdateHospitalStatusViewModel model)
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();
        var data = _hospitalManagementService.StatusHospitalManagement(model);
        return Ok(new { Data = data });
    }
}

```

```
catch (Exception ex)
{
    _logger.LogError(ex, $"Status error. ");
    return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
}
}
}
```