

```

using Meditech.App.Crm.Services;
using Meditech.App.Crm.ViewModels;
using Meditech.Core.Auth.Authorization;
using Meditech.Core.Auth.Entities;
using Meditech.Lib.Share;
using Meditech.Lib.Share.CommonClass.Paging;
using Microsoft.AspNetCore.Mvc;

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

        [HttpPost]
        [Route("search-grid")]
        public IActionResult SearchGrid(SearchGridAppointmentViewModel model)
        {
            try
            {
                var requestUser = (BaseUser)Request.HttpContext.Items["User"];
                if (requestUser == null) return Forbid();
                var data = _appointmentService.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/get-all")]
        public IActionResult GetAllAppointment(PaginationQueryModel<MobileAppointmentSearchAQueryData> model)
        {
            try
            {
                var requestUser = (BaseUser)Request.HttpContext.Items["User"];
                if (requestUser == null) return Forbid();
                var data = _appointmentService.GetAllAppointment(model, requestUser.Id);
                return Ok(new { Data = data });
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, $"GetAllAppointment error. ");
                return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
            }
        }

        [HttpGet]
        [Route("mobile/get-by-id")]
        public IActionResult GetAppointmentById(Guid id)
        {
            try
            {
                var requestUser = (BaseUser)Request.HttpContext.Items["User"];
                if (requestUser == null) return Forbid();
                var data = _appointmentService.GetAppointmentById(id);
                if (data.IsOk) return Ok(new { Data = data });
                return StatusCode(600, data);
            }
        }
    }
}

```

```

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

// [MediAuthorize(new[] { Lib.Share.PermitValue.Read, PermitValue.ReadAll }, Lib.Share.ActionPermit.en)]
[HttpPost]
public IActionResult AddNewAppointment(AppointmentViewModel model)
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();
        var data = _appointmentService.AddNewAppointment(model, requestUser.Id);
        if (data.IsOk) return Ok(new { Data = data });
        return StatusCode(600, data);
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"AddNewAppointment error. ");
        return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
    }
}

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

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

[HttpDelete]
public IActionResult DeleteAppointment(Guid id)
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();
        var data = _appointmentService.DeleteAppointment(id);
        if (data.IsOk) return Ok(new { Data = data });
        return StatusCode(600, data);
    }
    catch (Exception ex)

```

```

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

[HttpPut]
[Route("change-state")]
public IActionResult ChangeStatusAppointment(UpdateAppointmentStateViewModel model)
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();
        var data = _appointmentService.AppointmentState(model, requestUser.Id);
        if (data.IsOk) return Ok(new { Data = data });
        return StatusCode(600, data);
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"Status error. ");
        return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
    }
}

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

[HttpPut]
[Route("rating-submit")]
public IActionResult UpdateRatingSubmit(UpdateRatingSubmitViewModel model)
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();
        var data = _appointmentService.UpdateRatingSubmit(model, requestUser.Id);
        if (data.IsOk) return Ok(new { Data = data });
        return StatusCode(600, data);
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"RatingSubmit error. ");
        return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
    }
}

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

```

```

    }
}

#region asynchis
[MediAllowAnonymous]
[HttpGet]
[Route("asynchis/get-by-id")]
public IActionResult GetAsynchHisAppointmentById(Guid id)
{
    try
    {
        var ret = _appointmentService.getAsynchHisFromAppointmentById(id);
        if (!ret.IsOk)
        {
            return StatusCode(600, ret);
        }
        return Ok(new { Data = ret.Data });
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"GetAsynchHisAppointmentById error.");
        return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
    }
}

[MediAllowAnonymous]
[HttpGet]
[Route("asynchis/get-by-customer-id")]
public IActionResult GetAsynchHisCustomerById(Guid id)
{
    try
    {
        var ret = _appointmentService.getAsynchHisCustomerById(id);
        if (!ret.IsOk)
        {
            return StatusCode(600, ret);
        }
        return Ok(new { Data = ret.Data });
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"GetAsynchHisAppointmentByCustomerId error.");
        return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
    }
}

[MediAllowAnonymous]
[HttpPut]
[Route("asynchis/change-state")]
public IActionResult AsynchHisAppointmentStatus(Guid Id, AsynchHisToAppoinmentViewModel model)
{
    try
    {
        var data = _appointmentService.AsynchHisAppointmentState(model, Id);
        if (data.IsOk) return Ok(new { Data = data });
        return StatusCode(600, data);
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"Status error. ");
        return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
    }
}

[MediAllowAnonymous]
[HttpPost]
[Route("asynchis/hen-kham-lai")]
public IActionResult AsynchHisHenKhamLaiAppointment(SynchHISHenKhamLaiViewModel model)
{
    try
    {
        var data = _appointmentService.AsynchHisHenKhamLaiAppointment(model);
        if (data.IsOk) return Ok(new { Data = data.Data });
        return StatusCode(600, data);
    }
    catch (Exception ex)
    {

```

```
        _logger.LogError(ex, $"AsyncHisHenKhamLaiAppointment error. ");
        return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
    }
}
#endregion
}
}
```