USE [DigitalServices]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CustomerHappinessSelect] Script Date: 2/12/2020 10:02:30 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Anil S>

-- Create date: <26-03-2020>

-- Description: <Gets the records for Customer Happiness dash board>

-- =============================================

ALTER PROCEDURE [dbo].[CustomerHappinessSelect] (

--===============================

--Common fileds

--===============================

@LanguageId INT

,@RoleId INT

,@UserId INT

,@intPageNumber INT

,@intPageSize INT

,@varOrderBy VARCHAR(100) = NULL

,@varOrderDirection VARCHAR(10) = NULL

--================================

--Quick Search Fields

--================================

,@SearchText VARCHAR(255) = NULL

,@ServiceId INT

,@IncludeCompletedTasks BIT

)

AS

/\*

Declare

--===============================

--Common fileds

--===============================

@LanguageId INT =1

,@RoleId INT = null

,@UserId INT =10

,@intPageNumber INT =1

,@intPageSize INT=5300

,@varOrderBy VARCHAR(100) = 'submittedOn'

,@varOrderDirection VARCHAR(10) = NULL

--================================

--Quick Search Fields

--================================

,@SearchText VARCHAR(255) = 'RF-7063'

,@ServiceId INT =NULL

,@IncludeCompletedTasks BIT=1

EXEC CustomerHappinessSelect

@LanguageId

,@RoleId

,@UserId

,@intPageNumber

,@intPageSize

,@varOrderBy

,@varOrderDirection

,@SearchText

,@ServiceId

,@IncludeCompletedTasks

\*/

BEGIN

BEGIN TRY

SET NOCOUNT ON;

SET FMTONLY OFF;

DECLARE @CardDispatcheddStatusId INT

,@ExpiredStatusId INT

,@bitValidInput BIT = 1

,@varSQLText NVARCHAR(MAX)

,@SubServiceId INT

,@LicenseIssued INT

,@CardPrinted INT

,@Resubmitted INT

,@RequestForReissue INT

,@Draft INT

,@ElectronicallyRecieved INT

SELECT @CardDispatcheddStatusId = [CardDispatchedStatusId]

,@ExpiredStatusId = [ExpiredStatusId]

,@SubServiceId = [SubProcessWeeklyNew]

,@LicenseIssued = [LicenseIssuedStatusId]

,@CardPrinted = [CardPrintedStatusId]

,@Resubmitted = [ReSubmittedStatusId]

,@RequestForReissue = [RequestForReissueStatusId]

,@Draft = [DraftStatusId]

,@ElectronicallyRecieved =[ElectronicallyRecieved]

FROM [dbo].[ConfigRecreationalFishing]

SET @SearchText = RTRIM(LTRIM(ISNULL(@SearchText, '')))

SET @varSQLText = @varSQLText + '';

IF @varOrderDirection IS NULL

SET @varOrderDirection = 'ASC'

IF @varOrderBy IS NULL

SET @varOrderBy = 'ApplicationsOrder'

SELECT @bitValidInput = [dbo].[ValidateAlphabetsInput](ISNULL(@varOrderBy, '') + ',' + ISNULL(@varOrderDirection, ''))

IF (@bitValidInput = 0)

BEGIN

RAISERROR (

'Invalid order input'

,15

,1

);

RETURN

END

SET @varSQLText =

'

SELECT NEWID() AS NewRowId

,RFA.ID AS [RFAppilcationId]

,A.ServiceId AS ServiceId

,RFA.SubServiceId AS SubServiceId

,RFA.ProfileId AS [ProfileID]

,PL.[Name] AS [ServiceName]

,SPL.[Name] AS [ApplicationType]

,RFA.ApplicationID AS [ApplicationID]

,RFA.ApplicationNumber AS [ApplicationNumber]

,UL1.FullName AS CustomerName

,UL2.FullName AS ClaimedBy

,UL3.FullName AS ActionBy

,RFA.ModifiedOn AS ActionDate

,RFA.SubmittedDate AS SubmittedOn

,SL.[Name] AS [Status]

,CAST(IIF((

RFA.ApplicationWithUserID IS NULL

AND RFA.StatusID IN (

@LicenseIssued

,@CardPrinted

,@Resubmitted

,@RequestForReissue

)

), 1, 0) AS BIT) AS Claim

,CAST(IIF((

RFA.ApplicationWithUserID != @UserId

AND RFA.StatusID IN (

@LicenseIssued

,@CardPrinted

,@Resubmitted

,@RequestForReissue

)

), 1, 0) AS BIT) AS ReClaim

,CAST(IIF((

RFA.ApplicationWithUserID = @UserId

AND RFA.StatusID IN (

@LicenseIssued

,@CardPrinted

,@Resubmitted

,@RequestForReissue

)

), 1, 0) AS BIT) AS RevokeClaim

,CASE

WHEN RFA.ApplicationWithUserID = @UserId

AND RFA.StatusID IN (

@LicenseIssued

,@CardPrinted

,@Resubmitted

,@RequestForReissue

)

THEN 1

WHEN RFA.ApplicationWithUserID != @UserId

AND RFA.StatusID IN (

@LicenseIssued

,@CardPrinted

,@Resubmitted

,@RequestForReissue

)

THEN 2

WHEN RFA.ApplicationWithUserID IS NULL

AND RFA.StatusID IN (

@LicenseIssued

,@CardPrinted

,@Resubmitted

,@RequestForReissue

)

THEN 3

ELSE 4

END AS ApplicationsOrder

,RFA.ROWVERSION AS ROWVERSION

FROM [dbo].[RecreationalFishingApplication] RFA WITH (NOLOCK)

INNER JOIN [dbo].[Application] A ON RFA.ApplicationID = A.ID

INNER JOIN [dbo].[MtServiceLanguage] PL WITH (NOLOCK) ON PL.ServiceId = A.ServiceId

AND PL.LanguageID = @LanguageId

INNER JOIN [dbo].[MtSubServiceLanguage] SPL WITH (NOLOCK) ON SPL.SubServiceId = RFA.SubServiceId

AND SPL.LanguageID = @LanguageId

INNER JOIN [dbo].[MtStatusLanguage] SL WITH (NOLOCK) ON RFA.StatusID = SL.StatusID

AND SL.LanguageID = @LanguageId

LEFT JOIN [dbo].[User] U1 WITH (NOLOCK) ON RFA.CreatedBy = U1.ID

LEFT JOIN [dbo].[UserLanguage] UL1 WITH (NOLOCK) ON U1.ID = UL1.UserId

AND UL1.LanguageId = @LanguageId

LEFT JOIN [dbo].[User] U2 WITH (NOLOCK) ON RFA.ApplicationWithUserID = U2.ID

LEFT JOIN [dbo].[UserLanguage] UL2 WITH (NOLOCK) ON U2.ID = UL2.UserId

AND UL2.LanguageId = @LanguageId

LEFT JOIN [dbo].[User] U3 WITH (NOLOCK) ON RFA.ModifiedBy = U3.ID

LEFT JOIN [dbo].[UserLanguage] UL3 WITH (NOLOCK) ON U3.ID = UL3.UserId

AND UL3.LanguageId = @LanguageId

WHERE RFA.SubServiceId != @SubServiceId

AND RFA.StatusID != @Draft

'

--- ======================

--- For Free Search

--- ======================

IF (LEN(LTRIM(RTRIM(ISNULL(@SearchText, '')))) > 0)

BEGIN

SET @SearchText = '%' + @SearchText + '%';

SET @varSQLText = @varSQLText + '

AND (

RFA.ApplicationNumber LIKE @searchText

OR UL1.FullName LIKE @searchText

OR A.LicenseNumber LIKE @searchText

)

'

END

--- Service ID

--- ======================

IF (

@ServiceId IS NOT NULL

AND @ServiceId != 0

)

SET @varSQLText = @varSQLText + ' AND

PL.ServiceId = @ServiceId

'

-- @IncludeCompletedTasks

--- ======================

IF (@IncludeCompletedTasks = 1)

SET @varSQLText = @varSQLText + ' AND

RFA.StatusID IN (@CardDispatcheddStatusId

,@ExpiredStatusId

,@LicenseIssued

,@CardPrinted

,@Resubmitted

,@RequestForReissue

,@ElectronicallyRecieved)

'

IF (@IncludeCompletedTasks = 0)

SET @varSQLText = @varSQLText + ' AND

RFA.StatusID IN (@LicenseIssued

,@CardPrinted

,@Resubmitted

,@RequestForReissue)

'

--======================================================

SET @varSQLText = 'SELECT

RFAppilcationId

,ServiceId

,SubServiceId

, ProfileID

, ServiceName

, ApplicationType

, ApplicationId

, ApplicationNumber

, CustomerName

, ClaimedBy

, ActionBy

, ActionDate

, SubmittedOn

, Status

, Claim

, ReClaim

, RevokeClaim

, ApplicationsOrder

, RowVersion

, CONVERT(int,ROW\_NUMBER() OVER (ORDER BY

NewRowId

ASC

) +

ROW\_NUMBER() OVER (ORDER BY

NewRowId

DESC

) -1) AS TotalRecords

FROM

(

' + @varSQLText + '

)

Result

ORDER BY

' + @varOrderBy + ' ' + @varOrderDirection

IF (@intPageNumber <> - 1)

SET @varSQLText = @varSQLText + '

OFFSET (@intPageNumber - 1 ) \* @intPageSize ROWS FETCH NEXT @intPageSize ROWS ONLY

'

PRINT @varSQLText

--EXEC AS USER = 'DOMOS\_DynamicSql\_User'

EXEC dbo.sp\_executesql @varSQLtExt

,N'

@LanguageId int

,@RoleId int

,@intPageNumber int

,@intPageSize int

,@varOrderBy varchar(100)

,@varOrderDirection varchar(10)

,@SearchText varchar(255)

,@ServiceId INT

,@IncludeCompletedTasks BIT

,@SubServiceId INT

,@CardDispatcheddStatusId INT

,@LicenseIssued INT

,@CardPrinted INT

,@Resubmitted INT

,@RequestForReissue INT

,@UserId INT

,@Draft INT

,@ExpiredStatusId INT

,@ElectronicallyRecieved INT

'

,@LanguageId

,@RoleId

,@intPageNumber

,@intPageSize

,@varOrderBy

,@varOrderDirection

,@SearchText

,@ServiceId

,@IncludeCompletedTasks

,@SubServiceId

,@CardDispatcheddStatusId

,@LicenseIssued

,@CardPrinted

,@Resubmitted

,@RequestForReissue

,@UserId

,@Draft

,@ExpiredStatusId

,@ElectronicallyRecieved

END TRY

BEGIN CATCH

THROW

END CATCH

END

USE [DigitalServices]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CommonApplicationsDashboardSelect] Script Date: 2/12/2020 10:06:23 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Srikanth ch>

-- Create date: <13-09-2020>

-- Description: <Gets the records for Common Application dash board>

-- =============================================

/\*

Declare

@LanguageId INT =1

,@RoleId INT = 20

,@UserId INT =10975

,@intPageNumber INT =1

,@intPageSize INT=10

,@varOrderBy VARCHAR(100) = null

,@varOrderDirection VARCHAR(10) = 'Desc'

,@SearchText NVARCHAR(255) = ''

,@ServiceId INT =3

,@IncludeCompletedTasks BIT=1

EXEC CommonApplicationsDashboardSelect

@LanguageId

,@RoleId

,@UserId

,@intPageNumber

,@intPageSize

,@varOrderBy

,@varOrderDirection

,@SearchText

,@ServiceId

,@IncludeCompletedTasks

\*/

ALTER PROCEDURE [dbo].[CommonApplicationsDashboardSelect] (

@LanguageId INT

,@RoleId INT

,@UserId INT

,@intPageNumber INT

,@intPageSize INT

,@varOrderBy VARCHAR(100) = NULL

,@varOrderDirection VARCHAR(10) = NULL

,@SearchText NVARCHAR(255) = NULL

,@ServiceId INT

,@IncludeCompletedTasks BIT = 0

)

AS

BEGIN

BEGIN TRY

SET NOCOUNT ON;

SET FMTONLY OFF;

DECLARE @bitValidInput BIT = 1

,@varSQLText NVARCHAR(MAX)

,@DefaultUserRoleId SMALLINT

IF ISNULL(@SearchText,'') != '' OR ISNULL(@ServiceId,0) !=0

BEGIN

SET @IncludeCompletedTasks = 1

END

SET @SearchText = RTRIM(LTRIM(ISNULL(@SearchText, '')))

SET @varSQLText = @varSQLText + '';

SELECT @DefaultUserRoleId = RoleID FROM dbo.CtUserRoleMapping

where UserID= @UserId AND IsDefaultRole = 1

IF @varOrderDirection IS NULL

SET @varOrderDirection = 'DESC'

SELECT @bitValidInput = [dbo].[ValidateAlphabetsInput](ISNULL(@varOrderBy, '') + ',' + ISNULL(@varOrderDirection, ''))

IF (@bitValidInput = 0)

BEGIN

RAISERROR (

'Invalid order input'

,15

,1

);

RETURN

END

IF @varOrderBy IS NULL

SET @varOrderBy = 'RevokeClaim DESC, ReClaim DESC, Claim DESC, SubmittedOn'

SET @varSQLText =

'

SELECT NEWID() AS NewRowId

,AD.SubServiceApplicationId AS [SubServiceApplicationId]

,AD.RequestId AS [RequestId]

,AD.[DataClassificationId] AS [DataClassificationTypeId]

,AD.ServiceId AS ServiceId

,ADL.[Location] AS [Location]

,SUSL.[Name]+'' - ''+ADL.[Description] AS [Description]

,AD.ApplicationNumber AS ApplicationNumber

,SL.[Name] AS [Status]

,ML.[Name] AS [ApplicationWithRoleName]

,AD.SubmittedDate AS [SubmittedOn]

,AD.DueDate AS [DueDate]

,ADL.Customer AS [CustomerName]

,AD.Id AS AppilcationId

,AD.SubServiceId AS SubServiceId

,SVL.[Name] AS [ServiceName]

,ISNULL(AD.ApplicationWithRoleId,@DefaultUserRoleId) AS RoleId

,UL2.FullName AS ClaimedBy

,AD.ApplicationWithUserId AS ClaimedByUserId

,RL.[Name] AS ApplicationRoleName

,CAST(IIF((

AD.ApplicationWithUserID IS NULL

AND IsActionPendingWithEAD = 1

AND SASRM.sTATUSiD IS NOT NULL

), 1, 0) AS BIT) AS Claim

,CAST(IIF((

AD.ApplicationWithUserID != @UserId

AND IsActionPendingWithEAD = 1

AND SASRM.sTATUSiD IS NOT NULL

), 1, 0) AS BIT) AS ReClaim

,CAST(IIF((

AD.ApplicationWithUserID = @UserId

AND IsActionPendingWithEAD = 1

AND SASRM.sTATUSiD IS NOT NULL

), 1, 0) AS BIT) AS RevokeClaim

,AD.RowVersion AS RowVersion

FROM [ApplicationDashboard] AS AD

LEFT JOIN [CtServiceApplicaitonStatusRoleMapping] AS SASRM ON SASRM.ServiceId = AD.ServiceId

AND SASRM.StatusId = AD.StatusId

AND (SASRM.SubServiceId IS NULL OR AD.SubServiceId=SASRM.SubServiceId)

LEFT JOIN [MtDataClassificationTypeLanguage] AS DCTL ON DCTL.DataClassificationTypeId = AD.DataClassificationId

AND DCTL.LanguageId = @LanguageId

LEFT JOIN [ApplicationDashboardLanguage] AS ADL ON ADL.ApplicationDashboardId = AD.ID

AND ADL.LanguageId = @LanguageId

LEFT JOIN [MtStatusLanguage] AS SL ON SL.StatusID = AD.StatusId

AND SL.LanguageId = @LanguageId

LEFT JOIN MtRoleLanguage AS ML ON ML.RoleID = AD.ApplicationWithRoleId

AND ML.LanguageId = @LanguageId

LEFT JOIN [MtServiceLanguage] AS SVL ON SVL.ServiceId = AD.ServiceId

AND SVL.LanguageId = @LanguageId

LEFT JOIN [MtSubServiceLanguage] AS SUSL ON SUSL.SubServiceID = AD.SubServiceId

AND SUSL.LanguageId = @LanguageId

LEFT JOIN [dbo].[User] U2 WITH (NOLOCK) ON AD.ApplicationWithUserID = U2.ID

LEFT JOIN [dbo].[UserLanguage] UL2 WITH (NOLOCK) ON U2.ID = UL2.UserId

AND UL2.LanguageId = @LanguageId

LEFT JOIN MtRoleLanguage AS RL ON RL.RoleID = AD.ApplicationWithRoleId

AND RL.LanguageId = @LanguageId

WHERE

(AD.ServiceId in

(SELECT RSM.ServiceId from CtRoleServiceMapping RSM INNER JOIN CtUserRoleMapping URM ON URM.UserID =@UserId AND RSM.RoleID = URM.RoleID AND URM.IsActive=1 AND RSM.IsActive=1)

OR

AD.ApplicationWithUserId = @UserId

)

'

--- ======================

--- For Free Search

--- ======================

IF (LEN(LTRIM(RTRIM(ISNULL(@SearchText, '')))) > 0)

BEGIN

SET @SearchText = '%' + @SearchText + '%';

SET @varSQLText = @varSQLText + '

AND (

DCTL.[Name] LIKE @searchText

OR ADL.[Location] LIKE @searchText

OR ADL.[Description] LIKE @searchText

OR AD.ApplicationNumber LIKE @searchText

OR SL.[Name] LIKE @searchText

OR ML.[Name] LIKE @searchText

OR ADL.Customer LIKE @searchText

)

'

END

--- Service ID

--- ======================

IF (

@ServiceId IS NOT NULL

AND @ServiceId != 0

)

SET @varSQLText = @varSQLText + ' AND

AD.ServiceId = @ServiceId

'

-- @IncludeCompletedTasks

--- ======================

--IF (@IncludeCompletedTasks = 1)

-- SET @varSQLText = @varSQLText + ' AND

-- AD.IsActionPendingWithEAD = 0

-- '

IF (@IncludeCompletedTasks = 0)

SET @varSQLText = @varSQLText + ' AND

AD.IsActionPendingWithEAD = 1

'

--======================================================

SET @varSQLText = 'SELECT

[DataClassificationTypeId]

,SubServiceApplicationId

,RequestId

,ServiceId

,SubServiceId

,[Location]

,[Description]

,ApplicationNumber

,[Status]

,[ApplicationWithRoleName]

,[SubmittedOn]

,[DueDate]

,[CustomerName]

,AppilcationId

,[ServiceName]

,ApplicationRoleName

,RoleId

,ClaimedBy

,ClaimedByUserId

,Claim

,ReClaim

,RevokeClaim

,RowVersion

, CONVERT(int,ROW\_NUMBER() OVER (ORDER BY

NewRowId

ASC

) +

ROW\_NUMBER() OVER (ORDER BY

NewRowId

DESC

) -1) AS TotalRecords

FROM

(

' + @varSQLText + '

)

Result

ORDER BY

' + @varOrderBy + ' ' + @varOrderDirection

IF (@intPageNumber <> - 1)

SET @varSQLText = @varSQLText + '

OFFSET (@intPageNumber - 1 ) \* @intPageSize ROWS FETCH NEXT @intPageSize ROWS ONLY

'

-- Declare @Pos int=4000

-- print @Pos

--PRINT SUBSTRING(@varSQLtExt,1,@Pos)

--PRINT SUBSTRING(@varSQLtExt,@pos+1,8000)

--PRINT @varSQLtExt

EXEC dbo.sp\_executesql @varSQLtExt

,N'

@LanguageId INT

,@RoleId INT

,@intPageNumber INT

,@intPageSize INT

,@varOrderBy VARCHAR(100)

,@varOrderDirection VARCHAR(10)

,@SearchText VARCHAR(255)

,@ServiceId INT

,@UserId INT

,@DefaultUserRoleId SMALLINT

'

,@LanguageId

,@RoleId

,@intPageNumber

,@intPageSize

,@varOrderBy

,@varOrderDirection

,@SearchText

,@ServiceId

,@UserId

,@DefaultUserRoleId

END TRY

BEGIN CATCH

THROW

END CATCH

END