using System;

using System.Web;

using System.Collections.Generic;

using System.Text.RegularExpressions;

using Avis.Common.Customer.Text.RegularExpression;

using Avis.Mask.Application.Business;

using Avis.Mask.Application.Business.Interfaces;

using Avis.Mask.Application.Common;

using Avis.Mask.Application.Common.Enums;

using Avis.Mask.Application.Common.Exceptions;

using Avis.Mask.Application.Common.Helpers;

using Avis.Mask.Application.Common.Messages;

using Avis.Mask.Application.Common.Interfaces;

using Avis.Mask.Application.Presentation.Services;

using Avis.Mask.Application.Presentation.Views;

using Avis.Mask.Application.Presentation.Utility;

using Avis.Mask.Application.Common;

//using Avis.Mask.Application.WebApp.Helper;

namespace Avis.Mask.Application.Presentation.Presenters

{

public class WaitPresenter : IPresenter

{

#region Members

private readonly IWaitView \_view;

private readonly ISummaryService \_service;

private readonly ISessionProvider \_session;

private readonly IProfilManagerService \_profilService;

#endregion

#region Implementation of IPresenter

public void FillModelFromView()

{

}

public void Initialize()

{

\_view.Message = MaskHelper.GetTranslation("lblWaitMessage");

}

#endregion

public WaitPresenter(IWaitView view, ISummaryService service, ISessionProvider session, IProfilManagerService profilService)

{

\_view = view;

\_service = service;

\_session = session;

\_profilService = profilService;

}

public void CreateProfile()

{

Profile profile = \_session[SessionHelper.SessionSelectedProfile] as Profile;

if (profile != null)

{

if (profile.CurrentMode == MaskMode.Success)

{

if (MaskSettings.IsInternet)

\_view.NavigateToHomeInternet();

else

\_view.NavigateToHome();

}

else

{

bool isAuthorizationCalled = false;

bool isBusinessPass = false;

ICards card = null;

try

{

if (!string.IsNullOrEmpty(profile.BirthCountryCode) && profile.BirthCountryCode.ToUpper() == "KR")

profile.BirthCountryCode = "KP";

profile.ReferenceDate = DateTime.Now;

MaskType maskeType = MaskHelper.GetMaskTypeFromAwd(profile.ProgramClubMask);

if (!MaskSettings.IsProgramme247)

{

if (profile.ActionType == MaskHelper.TypeActionRen)

profile.TransactionSource =

MaskHelper.GetRenTransactionSource(maskeType, profile.ProgramIsPreferred,

profile.ProgramIsMra);

else

profile.TransactionSource = MaskHelper.GetAdhOrModTransactionSource(maskeType,

profile.

ProgramIsPreferred,

profile.ProgramIsMra);

}

if (MaskSettings.IsAgence)

profile.SourceType = "AGENCE";

else if (MaskSettings.IsBarcelone)

profile.SourceType = "RESOFF";

else

profile.SourceType = "INTERNET";

profile.ShowPreferredPdf = false;

bool replaceCard = false;

if (!string.IsNullOrEmpty(profile.WizardNumber))

{

if (!string.IsNullOrEmpty(profile.ProgramClubMask))

{

if (profile.OldProgramClub != profile.ProgramClubMask)

replaceCard = true;

}

string newValues = profile.Title.Trim() + profile.LastName.Trim() + profile.FirstName.Trim();

string oldValues = profile.OldTitle.Trim() + profile.OldLastName.Trim() +

profile.OldFirstName.Trim();

if (newValues.ToUpper() != oldValues.ToUpper())

replaceCard = true;

if (profile.ProgramClubMask != MaskSettings.AwdPreferred

&& ((profile.ProgramIsPreferred && string.IsNullOrEmpty(profile.OldAvisExpres))

|| (profile.ProgramIsMra && string.IsNullOrEmpty(profile.OldAvisPrefer))))

{

profile.ShowPreferredPdf = true;

}

}

else if (profile.ProgramIsPreferred && profile.ProgramClubMask != MaskSettings.AwdPreferred)

profile.ShowPreferredPdf = true;

Agent agent = null;

if (\_session != null && \_session.Contains(SessionHelper.SessionAgent))

agent = \_session[SessionHelper.SessionAgent] as Agent;

//if (MaskSettings.IsInternet && MaskSettings.IsProgramme247)

//{

// bool success = \_service.SaveProfileFor247(profile, agent);

// \_session[SessionHelper.SessionSelectedProfile] = profile;

// if (!success)

// \_view.NavigateToSummaryInternet();

// else

// \_view.NavigateToSuccess();

//}

//else

//{

if (profile.IsForBusinessPass)

{

isBusinessPass = true;

//bool hasWizard = !string.IsNullOrEmpty(profile.WizardNumber);

profile.ProgramClubMask = MaskSettings.AwdBusinessPass;

if (!string.IsNullOrEmpty(profile.WizardNumber) &&

!string.IsNullOrEmpty(profile.ProgramClubMask))

{

if (profile.OldProgramClub != profile.ProgramClubMask)

replaceCard = true;

}

card = \_service.SaveProfileInCards(profile, agent);

\_service.SaveProfileInWizard(profile, agent, replaceCard);

card.Etat = "1";

if (!string.IsNullOrEmpty(profile.WizardNumber))

card.WizardNumber = profile.WizardNumber;

\_service.SaveCard(card);

profile.Pack.WizardNumber = profile.WizardNumber;

profile.ProgramClubMask = MaskSettings.AwdPreferred;

bool success = \_service.SaveBusinessPass(profile.Pack);

if (!success)

{

DisplayedMessageCollection collection = new DisplayedMessageCollection();

collection.Add(

new DisplayedMessage(MaskHelper.GetTranslation("lblErrFailBusinessPass")));

profile.ErrorCollection = collection;

if (MaskSettings.IsInternet)

\_view.NavigateToSummaryInternet();

else

\_view.NavigateToSummary();

}

else

{

\_session[SessionHelper.SessionSelectedProfile] = profile;

\_view.NavigateToSuccess();

}

}

else

{

if (profile.Is3DSecured)

{

CreditCard creditCard = profile.MaskCreditCard;

\_service.ValidateAuthenticationAndOrder(creditCard, profile);

using (Logger logger = new Logger("Checking the Response in wait presenter"))

{

logger.Write("Response Object status:" + profile.ResponseCode);

}

if (profile.ResponseCode == "00")

{

//profile creation & save after payment confirmation

//\_service.SaveProfileInWizard(profile, agent, replaceCard);

\_view.NavigateToSuccess();

}

}

else

{

card = \_service.SaveProfileInCards(profile, agent);

if (!IsCardFree(profile))

{

if (MaskSettings.PaymentActive)

{

CreditCard creditCard = profile.MaskCreditCard;

//Moto transaction

if (!MaskSettings.IsInternet)

{

\_service.CardOrder(creditCard, profile);

}

else

{

\_service.Authorize(creditCard, profile);

}

//till here

//\_service.Authorize(creditCard, profile);

isAuthorizationCalled = true;

}

}

//Ananth - Commented Save profile in Wizard

//\_service.SaveProfileInWizard(profile, agent, replaceCard);

card.Etat = "1";

card.AuthorizationId = profile.SipsAuthorizationId;

card.TransactionId = profile.SipsTransactionId;

card.TransactionDate = profile.SipsTransactionDate;

card.TransactionTime = profile.SipsTransactionTime;

if(!string.IsNullOrEmpty(profile.WizardNumber))

card.WizardNumber = profile.WizardNumber;

\_service.SaveCard(card);

#region comments 24/7

//if (!string.IsNullOrEmpty(profile.WizardNumber))

//{

// if (!string.IsNullOrEmpty(profile.RefImgPermis))

// {

// string newName = profile.WizardNumber +

// profile.RefImgPermis.Substring(profile.RefImgPermis.Length - 4, 4);

// MaskHelper.RenameFile(profile.RefImgPermis, newName,

// \_view.GetPathApplication() + MaskSettings.FolderPermis);

// profile.RefImgPermis = newName;

// }

// if (!string.IsNullOrEmpty(profile.RefImgQuittance))

// {

// string newName = profile.WizardNumber +

// profile.RefImgQuittance.Substring(

// profile.RefImgQuittance.Length - 4, 4);

// MaskHelper.RenameFile(profile.RefImgQuittance, newName,

// \_view.GetPathApplication() + MaskSettings.FolderQuittance);

// profile.RefImgQuittance = newName;

// }

//}

#endregion

profile.IsEnrolled = !string.IsNullOrWhiteSpace(profile.RedirectionUrl);

\_session[SessionHelper.SessionSelectedProfile] = profile;

#region comments 24/7

//if (MaskSettings.IsProgramme247)

//{

// Profil247 profile247 = null;

// if (\_session[SessionHelper.SessionSelectedProfile247] != null)

// {

// profile247 = \_session[SessionHelper.SessionSelectedProfile] as Profil247;

// }

// else

// {

// IList<Profil247> lstProfile247 =

// \_profilService.SearchProfiles247ByLogin(profile.Login);

// if (lstProfile247 != null && lstProfile247.Count == 1)

// profile247 = lstProfile247[0];

// }

// if (profile247 != null)

// {

// //Update Properties for 24/7

// profile247.RefImgPermis = profile.RefImgPermis;

// profile247.RefImgQuittance = profile.RefImgQuittance;

// if (!string.IsNullOrEmpty(profile.Password247))

// {

// //TODO

// }

// //Update Profil 24/7

// var pfl247 = (Profil247)\_profilService.SaveProfil247(profile247);

// if (pfl247 != null)

// \_session[SessionHelper.SessionSelectedProfile] = pfl247;

// }

//}

#endregion

// if redirectionURL is not null in response object

if (!string.IsNullOrWhiteSpace(profile.RedirectionUrl))

{

//using (Logger logger = new Logger("Before 3ds redirect"))

//{

// logger.Write("Message:" + profile.RedirectionUrl);

//}

// pay\\\\

// web

// var session = new WebSession();

// \_session.Add(SessionHelper.SessionSelectedProfile, profile);

// htt

\_session[SessionHelper.SessionSelectedProfile] = profile;

\_session[Avis.Mask.Application.Common.SessionHelper.SessionSelectedProfile] = profile;

\_view.NavigateTo3DS(profile.RedirectionUrl);

htt

HttpContext.Current.\_session[SessionHelper.SessionSelectedProfile] = profile;

\_session[Avis.Mask.Application.Common.SessionHelper.SessionSelectedProfile]= profile;

\_session.Add(SessionHelper.SessionSelectedProfile, profile);

}

else

{

//moksha MOTO change

if(!MaskSettings.IsInternet)

{

if (profile.ResponseCode == "00")

{

//profile creation & save after payment confirmation

//\_service.SaveProfileInWizard(profile, agent, replaceCard);

\_view.NavigateToSuccess();

}

}

//till here

if (profile.RedirectionStatusCode == "01")

{

//proceed to authorisation

CreditCard creditCard = profile.MaskCreditCard;

//moksha MOTO TRANSACTION

//if(creditCard.Cryptogramme != "") //till here

\_service.ValidateAuthenticationAndOrder(creditCard, profile);

if (profile.ResponseCode == "00")

{

//profile creation & save after payment confirmation

//\_service.SaveProfileInWizard(profile, agent, replaceCard);

\_view.NavigateToSuccess();

}

}

\_view.NavigateToSuccess();

// Avis.Mask.Application.Business.Profile currentProfil = \_session[Avis.Mask.Application.Common.SessionHelper.SessionSelectedProfile] as Avis.Mask.Application.Business.Profile;

}

}

}

// var Session=new webSession;

}

catch (CancelException exp)

{

using(Logger logger = new Logger("Logging The Cancel Excepttion"))

{

logger.Write("Message:" + exp.Message + "Source:" + exp.Source + "Inner Exception:" + exp.InnerException +"Exception object logged:"+ exp);

}

if (card != null)

{

card.AuthorizationId = profile.SipsAuthorizationId;

card.TransactionId = profile.SipsTransactionId;

if (exp.ErrorsCollection != null && exp.ErrorsCollection.Count > 0)

{

foreach (DisplayedMessage message in exp.ErrorsCollection)

{

if (!string.IsNullOrEmpty(card.Remarks))

card.Remarks += "\*\*\*";

card.Remarks += message.Message;

}

}

}

if (isAuthorizationCalled)

{

\_service.Cancel(profile);

if (card != null)

{

card.CancelTransactionDate = profile.SipsCancelTransactionDate;

card.CancelTransactionTime = profile.SipsCancelTransactionTime;

}

}

if (card != null)

\_service.SaveCard(card);

profile.ErrorCollection = ManageErrorMessage(exp.ErrorsCollection, isBusinessPass);

\_session[SessionHelper.SessionSelectedProfile] = profile;

if (MaskSettings.IsInternet)

\_view.NavigateToSummaryInternet();

else

\_view.NavigateToSummary();

}

catch (Exception exp)

{

using (Logger logger = new Logger("Logging The Exception"))

{

logger.Write("Message:" + exp.Message + "Source:" + exp.Source + "Inner Exception:" + exp.InnerException + "Exception object logged:"+ exp);

}

var collection = new DisplayedMessageCollection();

collection.Add(new DisplayedMessage(exp.Message));

if (card != null)

{

card.Remarks = exp.Message;

\_service.SaveCard(card);

}

profile.ErrorCollection = ManageErrorMessage(collection, isBusinessPass);

\_session[SessionHelper.SessionSelectedProfile] = profile;

if (MaskSettings.IsInternet)

\_view.NavigateToSummaryInternet();

else

\_view.NavigateToSummary();

}

}

}

}

private static DisplayedMessageCollection ManageErrorMessage(DisplayedMessageCollection collection, bool isBusinessPass)

{

var messageCollection = new DisplayedMessageCollection();

if (collection != null && collection.Count > 0)

{

foreach (DisplayedMessage message in collection)

{

messageCollection.Add(new DisplayedMessage(ManageErrorMessage(message.Message, isBusinessPass)));

}

}

return messageCollection;

}

private static string ManageErrorMessage(string message, bool isBusinessPass)

{

Match match = CustomerRegularExpression.CreditIdInUse.Match(message);

Match matchUpdate = CustomerRegularExpression.CreditIdInUseUpdate.Match(message);

if (match != null && !string.IsNullOrEmpty(match.Value))

{

string tmp = match.Value.Trim();

string wiz = string.Empty;

Match wizardNumber = CustomerRegularExpression.WizardNumber2.Match(tmp);

if (wizardNumber != null && !string.IsNullOrEmpty(wizardNumber.Value))

wiz = "(N° " + wizardNumber.Value + ") ";

if (isBusinessPass)

message = string.Format(MaskHelper.GetTranslation("lblErrCreditIdInUsePass"), wiz);

else

message = string.Format(MaskHelper.GetTranslation(MaskSettings.IsInternet

? "lblErrCreditIdInUseInternet"

: "lblErrCreditIdInUse"), wiz);

}

else if (matchUpdate != null && !string.IsNullOrEmpty(matchUpdate.Value))

{

string tmp = matchUpdate.Value.Trim();

string wiz = string.Empty;

Match wizardNumber = CustomerRegularExpression.WizardNumber2.Match(tmp);

if (wizardNumber != null && !string.IsNullOrEmpty(wizardNumber.Value))

wiz = "(N° " + wizardNumber.Value + ") ";

if (isBusinessPass)

message = string.Format(MaskHelper.GetTranslation("lblErrCreditIdInUsePass"), wiz);

else

message = string.Format(MaskHelper.GetTranslation(MaskSettings.IsInternet

? "lblErrCreditIdInUseInternet"

: "lblErrCreditIdInUse"), wiz);

}

else

{

if (MaskSettings.IsInternet)

{

message = MaskHelper.GetTranslation(isBusinessPass ? "lblErrExceptionPass" : "lblErrException");

}

}

return message;

}

private static bool IsCardFree(Profile profile)

{

return (profile.ActionType == MaskHelper.TypeActionMod

|| profile.ProgramClubMask == MaskSettings.AwdPreferred

|| profile.ProgramClubMask == MaskSettings.AwdBusinessPass);

}

}

}