using System;

using System.Collections.Generic;

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.Helpers;

using Avis.Mask.Application.Presentation.Models;

using Avis.Mask.Application.Common.Interfaces;

using Avis.Mask.Application.Presentation.Services;

using Avis.Mask.Application.Presentation.Views;

namespace Avis.Mask.Application.Presentation.Presenters

{

public class SummaryPresenter : IPresenter

{

#region Members

private readonly ISummaryView \_view;

private readonly ISessionProvider \_session;

private readonly SummaryModel \_model;

private readonly ISummaryService \_service;

private readonly IProfilManagerService \_service247;

#endregion

#region Properties

public bool IsFrom3DS;

#endregion

#region Constructors

public SummaryPresenter(ISummaryView view, SummaryModel model, ISummaryService service, ISessionProvider session, IProfilManagerService service247)

{

\_view = view;

\_model = model;

\_service = service;

\_session = session;

\_service247 = service247;

if (\_model != null)

\_model.PropertyChanged += ModelPropertyChanged;

}

void ModelPropertyChanged(object sender, System.ComponentModel.PropertyChangedEventArgs e)

{

//throw new System.Exception("The method or operation is not implemented.");

}

#endregion

#region Public Methods

public void Initialize()

{

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

{

\_view.NavigateToHome();

}

else

{

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

if (\_session == null)

{

}

var x = \_session;

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

{

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

if (profile != null)

{

\_view.ShowPrefSummary(true);

\_view.ShowProfSummary(true);

\_view.ShowCryptogramme = MaskSettings.IsInternet;

\_view.SummaryTitle = GetCardDescription(profile.ProgramClubMask);

if (profile.ProgramClubMask == MaskSettings.AwdMouv)

{

\_view.ShowPrefSummary(false);

\_view.ShowProfSummary(false);

}

if (profile.CurrentMode == MaskMode.Edit)

{

if (MaskSettings.IsInternet)

\_view.NavigateToInternetEditForm();

else

\_view.NavigateToEditForm();

}

string urlrefer = null;

if (\_session.Contains("URI\_REFER"))

{

urlrefer = \_session["URI\_REFER"].ToString();

}

if (!string.IsNullOrEmpty(urlrefer) && urlrefer.Contains(MaskSettings.Formulaire247))

{

\_view.ShowPayControl(false);

\_view.ShowConfirmButton(true);

\_view.ShowModifyButton(true);

}

else

{

if (IsCardFree(profile))

{

\_view.ShowPayControl(false);

\_view.ShowConfirmButton(true);

\_view.ShowModifyButton(true);

}

else

{

\_view.ShowPayControl(true);

\_view.ShowConfirmButton(false);

\_view.ShowModifyButton(true);

}

}

FillViewFromProfile(profile);

if(profile.ErrorCollection != null && profile.ErrorCollection.Count>0)

{

\_view.ShowMessages(profile.ErrorCollection);

}

}

}

}

}

private static string GetCardDescription(string awdNumber)

{

string description = MaskHelper.GetTranslation("lblTypeCartePreferred");

if (awdNumber == MaskSettings.AwdAzur)

description = MaskHelper.GetTranslation("lblTypeCarteAzur");

else if (awdNumber == MaskSettings.AwdBusiness)

description = MaskHelper.GetTranslation("lblTypeCarteBusiness");

else if (awdNumber == MaskSettings.AwdMouv)

description = MaskHelper.GetTranslation("lblTypeCarteMouv");

else if (awdNumber == MaskSettings.AwdSenior)

description = MaskHelper.GetTranslation("lblTypeCarteSenior");

return string.Format(MaskHelper.GetTranslation("lblSummaryTitle"), description);

}

private static bool IsCardFree(Profile profile)

{

return (profile.ActionType == MaskHelper.TypeActionMod

|| profile.ProgramClubMask == MaskSettings.AwdPreferred

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

}

public void CreateProfile()

{

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

if (IsFrom3DS)

{

//Added this condition since after 3ds redirection creditcardtype is gettng as visa by default.

if (profile.MaskCreditCard.Type == "MASTERCARD")

{

this.\_view.CreditCardType = "CM";

}

else if (profile.MaskCreditCard.Type == "VISA")

{

this.\_view.CreditCardType = "CX";

}

}

FillModelFromView();

//moksha commented this and added above condition.

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

if (profile != null)

{

if (!\_model.Validate() && !IsCardFree(profile))

\_view.ShowMessages(\_model.DisplayedMessages);

if (!\_model.ValidateCredit() && !IsCardFree(profile))

\_view.ShowMessages(\_model.DisplayedMessages);

else

{

CreditCard creditCard = \_model.CurrentCreditCard;

if (MaskSettings.IsInternet)

creditCard.Cryptogramme = \_model.CreditCardCryptogramme;

profile.MaskCreditCard = creditCard;

profile.ErrorCollection = null;

if (IsFrom3DS)

{

profile.Is3DSecured = true;

IsFrom3DS = false;

}

\_session[SessionHelper.SessionSelectedProfile] = profile;

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

\_view.NavigateToWait();

}

}

else

\_view.NavigateToHome();

}

//public void CretaProfil247()

//{

// try

// {

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

// if (profile != null)

// {

// //save the profil 24/7 in DB

// IProfil247 profil247 = new Profil247();

// profil247.Login = profile.PrivEmail;

// profil247.Password = profile.Password247;

// profil247.PinCode = profile.PinCode;

// profil247.RefImgPermis = profile.RefImgPermis;

// profil247.RefImgQuittance = profile.RefImgQuittance;

// profil247.RefSmartCard = profile.RefSmartCard;

// profil247.StatusAccount = !string.IsNullOrEmpty(profile.StatusAccount) ? profile.StatusAccount : Profile247AccountStatus.Inactif.ToString();

// //profil247.StatusRequest = !string.IsNullOrEmpty(profile.StatusRequest) ? profile.StatusRequest : Profile247RequestStatus.Nouveau.ToString();

// profil247.StatusRequest = Profile247RequestStatus.Nouveau.ToString();

// profil247.TransactionDate = profile.TransactionDate;

// profil247.TransactionExpiryDate = profile.TransactionExpiryDate;

// profil247.TransactionSource = profile.TransactionSource;

// profil247.WizardNumber = profile.WizardNumber;

// profil247.PackProtectionPlus = profile.PackProtectionPlus247;

// \_service247.SaveProfil247(profil247);

// if (MaskSettings.IsInternet)

// {

// \_service247.SendEmailOnDemandConfirmationNewAdhesionFromInternet(profil247.Login);

// \_service.Log(MaskSettings.ApplicationName,

// "SendEmailOnDemandConfirmationNewAdhesionFromInternet 247",

// Avis.Common.Log.LogLevel.Info.ToString());

// }

// if (MaskSettings.IsAgence)

// {

// \_service247.SendEmailOnDemandConfirmationNewAdhesionFromAgency(profil247.Login);

// \_service.Log(MaskSettings.ApplicationName,

// "SendEmailOnDemandConfirmationNewAdhesionFromAgency 247",

// Avis.Common.Log.LogLevel.Info.ToString());

// }

// //TODO BARCELONE

// \_view.ShowMessageConfirmation();

// }

// else

// \_view.NavigateToHome();

// }

// catch (Exception ex)

// {

// \_service.Log(MaskSettings.ApplicationName,"CretaProfil247(): "+ex.Message,Avis.Common.Log.LogLevel.Error.ToString());

// }

//}

public void NavigateBack()

{

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

if (profile != null)

{

profile.CurrentMode = MaskMode.Edit;

if (MaskSettings.IsProgramme247)

profile.CurrentMode = MaskMode.ShowPrivate;

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

}

if (MaskSettings.IsInternet)

\_view.NavigateToInternetEditForm();

else

\_view.NavigateToEditForm();

}

#endregion

#region Private Methods

private string GetCountryName(string countryCode)

{

string name = string.Empty;

try

{

IList<Country> countries = \_service.GetCountries();

List<Country> list = (List<Country>)countries;

Country country = list.Find(delegate(Country x) { return x.Code == countryCode; });

if (country != null)

name = country.CaptionFr;

}

catch (Exception exp)

{

}

return name;

}

private void FillViewFromProfile(Profile profile)

{

\_view.AcceptElectronicReceipts = profile.AcceptEbill ? "Oui" : "Non";

\_view.AcceptEmailOffers = profile.AcceptNewsLetter ? "Oui" : "Non";

\_view.BirthCountry = GetCountryName(profile.BirthCountryCode);

\_view.BirthDate = (profile.BirthDate.HasValue)

? profile.BirthDate.Value.ToString(MaskSettings.BirthDateFormat)

: string.Empty;

\_view.BirthPlace = profile.BirthPlace;

\_view.DriverLicenseCountry = GetCountryName(profile.DriverLicenseCountryCode);

\_view.DriverLicenseYear = (profile.DriverLicenseDate.HasValue)

? profile.DriverLicenseDate.Value.ToString(MaskSettings.LicenseDateFormat)

: string.Empty;

\_view.DriverLicenseNumber = profile.DriverLicenseNumber;

\_view.FrequentTravellerCode = profile.FrequentTravellerCode;

\_view.FrequentTravellerNumber = profile.FrequentTravellerNumber;

\_view.FullName = profile.Title + " " + profile.LastName + " " + profile.FirstName;

\_view.MobilePhoneNumber = profile.MobilePhoneNumber;

\_view.PrivateAddressCity = profile.PrivAddressCity;

\_view.PrivateAddressCountry = GetCountryName(profile.PrivAddressCountryCode);

\_view.PrivateAddressLine1 = profile.PrivAddressLine1;

\_view.PrivateAddressLine2 = profile.PrivAddressLine2;

\_view.PrivateAddressZipCode = profile.PrivAddressZipCode;

\_view.PrivatePhoneNumber = profile.PrivatePhoneNumber;

\_view.PrivateEmail = profile.PrivEmail;

\_view.ProfAddressCity = profile.ProfAddressCity;

\_view.ProfAddressCountry = GetCountryName(profile.ProfAddressCountryCode);

\_view.ProfAddressLine1 = profile.ProfAddressLine1;

\_view.ProfAddressLine2 = profile.ProfAddressLine2;

\_view.ProfAddressZipCode = profile.ProfAddressZipCode;

\_view.ProfCompanyName = profile.ProfCompanyName;

\_view.ProfFunction = profile.ProfFunction;

\_view.ProfPhoneNumber = profile.ProfPhoneNumber;

\_view.ProgramIsPreferred = profile.ProgramIsPreferred ? "Oui" : "Non";

if (profile.ProgramIsPreferred)

{

\_view.ProgramIsMra = profile.ProgramIsMra ? "Non" : "Oui"; //Modification benjamin 15-12-2011 il y avait une inversion entre IsMra et Signature de chacun des contrat

\_view.ProgramPayDateExp = (profile.ProgramPayDateExp.HasValue)

? profile.ProgramPayDateExp.Value.ToString(

MaskSettings.ExpirationDateFormat)

: string.Empty;

\_view.ProgramPayNumber = profile.ProgramPayNumber;

\_view.ProgramPayType = profile.ProgramPayType;

\_view.ProgramPrefCarGroup = profile.ProgramPrefCarGroup;

//if (profile.PackProtectionPlus247.HasValue)

//{

// \_view.ProgramHasPackProtectionPlus = ((bool) profile.PackProtectionPlus247) ? "Oui" : "Non";

//}

//else

//{

// \_view.ProgramHasPackProtectionPlus = "Non";

//}

}

}

#endregion

#region IPresenter Members

public void FillModelFromView()

{

\_model.CreditCardCryptogramme = \_view.CreditCardCryptogramme;

\_model.CreditCardExpiration = \_view.CreditCardExpiration;

\_model.CreditCardNumber = \_view.CreditCardNumber;

\_model.CreditCardType = \_view.CreditCardType;

}

#endregion

}

}