Socket = new ConnectSocket();

Socket.WIO.OnConnected += WIO\_OnConnected;

Socket.WIO.On("Login", response =>

{

int userId = response.GetValue<int>(0);

if (userId != userCurrent.ID)

{

this.Dispatcher.Invoke(new Action(() =>

{

int index = ChatList.FindIndex(con => con.UserContact.ID == userId && con.isGroup == 0);

if (index != -1)

{

ChatList[index].UserContact.isOnline = 1;

ChatList[index].IsOnline = 1;

if (ChatList[index].IsFavorite == 1)

{

ListViewMenuChatFavorite.Items.SortDescriptions.Add(new SortDescription("TimeLastMessage", ListSortDirection.Descending));

ListViewMenuChatFavorite.Items.Refresh();

}

else

{

ListViewMenuChat.Items.SortDescriptions.Add(new SortDescription("TimeLastMessage", ListSortDirection.Descending));

ListViewMenuChat.Items.Refresh();

}

}

int indexContact = contactList.FindIndex(contact => contact.UserContact.ID == userId);

if (indexContact != -1)

{

contactList[indexContact].UserContact.IsOnline = 1;

contactList[indexContact].HaveActive = true;

}

if (UserCurrent.CompanyId != 0)

{

int indexContactCompany = contactCompany.FindIndex(contact => contact.UserContact.ID == userId);

if (indexContactCompany != -1)

{

contactCompany[indexContactCompany].UserContact.IsOnline = 1;

contactCompany[indexContactCompany].HaveActive = true;

}

else

{

using (WebClient httpClient = new WebClient())

{

try

{

httpClient.QueryString.Clear();

httpClient.QueryString.Add("ID", userId.ToString());

httpClient.UploadValuesCompleted += (sender, e) =>

{

APIUser receiveInfoAPI = JsonConvert.DeserializeObject<APIUser>(UnicodeEncoding.UTF8.GetString(e.Result));

if (receiveInfoAPI.data != null)

{

UserFromServer userMessageFromSerVer = receiveInfoAPI.data.user\_info;

User userContact = new User(userMessageFromSerVer.ID, userMessageFromSerVer.ID365, userMessageFromSerVer.Type365, userMessageFromSerVer.Email, userMessageFromSerVer.Password, userMessageFromSerVer.Phone, userMessageFromSerVer.UserName, userMessageFromSerVer.AvatarUser, userMessageFromSerVer.Status, userMessageFromSerVer.StatusEmotion, userMessageFromSerVer.LastActive, userMessageFromSerVer.Active, userMessageFromSerVer.isOnline, userMessageFromSerVer.Looker, userMessageFromSerVer.CompanyId, userMessageFromSerVer.CompanyName);

if (userContact.CompanyId == UserCurrent.CompanyId)

{

contactCompany.Add(new MenuContact(userContact, 0));

}

}

};

httpClient.UploadValuesAsync(new Uri(URL + "User/GetInfoUser"), "POST", httpClient.QueryString);

}

catch (Exception ex)

{

Socket\_NoInternet();

}

}

}

}

if (SelectionConversation != null && SelectionConversation.UserContact.ID == userId)

{

int indexConversation = ChatList.FindIndex(con => con.UserContact.ID == userId && con.isGroup == 0);

if (indexConversation != -1)

{

SelectionConversation = ChatList[indexConversation];

if (ChatList[indexConversation].UserContact.ID == userId)

{

inforConversation.DataContext = null;

infoChat.DataContext = null;

inforConversation.DataContext = SelectionConversation;

infoChat.DataContext = SelectionConversation;

}

}

}

DisplayContact();

}));

}

});

Socket.WIO.On("Logout", response =>

{

int userId = response.GetValue<int>(0);

if (userId != userCurrent.ID)

{

this.Dispatcher.Invoke(new Action(() =>

{

int index = ChatList.FindIndex(con => con.UserContact.ID == userId && con.isGroup == 0);

if (index != -1)

{

ChatList[index].UserContact.isOnline = 0;

ChatList[index].IsOnline = 0;

if (ChatList[index].IsFavorite == 1)

{

ListViewMenuChatFavorite.Items.SortDescriptions.Add(new SortDescription("TimeLastMessage", ListSortDirection.Descending));

ListViewMenuChatFavorite.Items.Refresh();

}

else

{

ListViewMenuChat.Items.SortDescriptions.Add(new SortDescription("TimeLastMessage", ListSortDirection.Descending));

ListViewMenuChat.Items.Refresh();

}

}

int indexContact = contactList.FindIndex(contact => contact.UserContact.ID == userId);

if (indexContact != -1)

{

contactList[indexContact].UserContact.IsOnline = 0;

contactList[indexContact].HaveActive = false;

}

if (UserCurrent.CompanyId != 0)

{

int indexContactCompany = contactCompany.FindIndex(contact => contact.UserContact.ID == userId);

if (indexContactCompany != -1)

{

contactCompany[indexContactCompany].UserContact.IsOnline = 0;

contactCompany[indexContactCompany].HaveActive = false;

}

}

if (SelectionConversation != null && SelectionConversation.UserContact.ID == userId)

{

int indexConversation = ChatList.FindIndex(con => con.UserContact.ID == userId && con.isGroup == 0);

if (indexConversation != -1)

{

SelectionConversation = ChatList[indexConversation];

if (ChatList[indexConversation].UserContact.ID == userId)

{

inforConversation.DataContext = null;

infoChat.DataContext = null;

inforConversation.DataContext = SelectionConversation;

infoChat.DataContext = SelectionConversation;

}

}

}

DisplayContact();

}));

}

});

Socket.WIO.On("changeName", response =>

{

int userId = response.GetValue<int>(0);

string nameUser = response.GetValue<string>(1);

if (userId == userCurrent.ID)

{

User userNewAvatar = UserCurrent;

userNewAvatar.UserName = nameUser;

UserCurrent = userNewAvatar;

}

else

{

int index = ChatList.FindIndex(con => con.UserContact.ID == userId && con.isGroup == 0);

if (index != -1)

{

ChatList[index].UserContact.UserName = nameUser;

ChatList[index].Name = nameUser;

this.Dispatcher.Invoke(new Action(() =>

{

if (ChatList[index].IsFavorite == 1)

{

ListViewMenuChatFavorite.Items.Refresh();

}

else

{

ListViewMenuChat.Items.Refresh();

}

}));

}

int indexContact = contactList.FindIndex(contact => contact.UserContact.ID == userId);

if (indexContact != -1)

{

contactList[indexContact].UserContact.UserName = nameUser;

}

if (UserCurrent.CompanyId != 0)

{

int indexContactCompany = contactCompany.FindIndex(contact => contact.UserContact.ID == userId);

if (indexContactCompany != -1)

{

contactCompany[indexContactCompany].UserContact.UserName = nameUser;

}

}

if (SelectionConversation != null && SelectionConversation.UserContact.ID == userId)

{

int indexConversation = ChatList.FindIndex(con => con.UserContact.ID == userId && con.isGroup == 0);

if (indexConversation != -1)

{

SelectionConversation = ChatList[indexConversation];

if (ChatList[indexConversation].UserContact.ID == userId)

{

inforConversation.DataContext = null;

infoChat.DataContext = null;

inforConversation.DataContext = SelectionConversation;

infoChat.DataContext = SelectionConversation;

}

}

}

DisplayContact();

}

});

Socket.WIO.On("changeActive", response =>

{

int userId = response.GetValue<int>(0);

int activeUser = response.GetValue<int>(1);

if (userId != userCurrent.ID)

{

this.Dispatcher.Invoke(new Action(() =>

{

int index = ChatList.FindIndex(con => con.UserContact.ID == userId && con.isGroup == 0);

if (index != -1)

{

ChatList[index].UserContact.Active = activeUser;

ChatList[index].Active = activeUser;

if (ChatList[index].IsFavorite == 1)

{

ListViewMenuChatFavorite.Items.SortDescriptions.Add(new SortDescription("TimeLastMessage", ListSortDirection.Descending));

ListViewMenuChatFavorite.Items.Refresh();

}

else

{

ListViewMenuChat.Items.SortDescriptions.Add(new SortDescription("TimeLastMessage", ListSortDirection.Descending));

ListViewMenuChat.Items.Refresh();

}

}

int indexContact = contactList.FindIndex(contact => contact.UserContact.ID == userId);

if (indexContact != -1)

{

contactList[indexContact].UserContact.Active = activeUser;

}

if (UserCurrent.CompanyId != 0)

{

int indexContactCompany = contactCompany.FindIndex(contact => contact.UserContact.ID == userId);

if (indexContactCompany != -1)

{

contactCompany[indexContactCompany].UserContact.Active = activeUser;

}

}

if (SelectionConversation != null && SelectionConversation.isGroup == 0 && SelectionConversation.UserContact.ID == userId)

{

int indexConversation = ChatList.FindIndex(con => con.UserContact.ID == userId && con.isGroup == 0);

if (indexConversation != -1)

{

SelectionConversation = ChatList[indexConversation];

if (ChatList[indexConversation].UserContact.ID == userId)

{

inforConversation.DataContext = null;

infoChat.DataContext = null;

inforConversation.DataContext = SelectionConversation;

infoChat.DataContext = SelectionConversation;

}

}

}

DisplayContact();

}));

}

});

Socket.WIO.On("changeAvatarUser", response =>

{

int userId = response.GetValue<int>(0);

string avatarUser = response.GetValue<string>(1);

this.Dispatcher.Invoke(new Action(() =>

{

if (userId == userCurrent.ID)

{

User userNewAvatar = UserCurrent;

userNewAvatar.AvatarUser = avatarUser;

UserCurrent = userNewAvatar;

}

else

{

int index = ChatList.FindIndex(con => con.UserContact.ID == userId && con.isGroup == 0);

if (index != -1)

{

ChatList[index].UserContact.AvatarUser = avatarUser;

ChatList[index].AvatarUser = avatarUser;

if (ChatList[index].IsFavorite == 1)

{

ListViewMenuChatFavorite.Items.SortDescriptions.Add(new SortDescription("TimeLastMessage", ListSortDirection.Descending));

ListViewMenuChatFavorite.Items.Refresh();

}

else

{

ListViewMenuChat.Items.SortDescriptions.Add(new SortDescription("TimeLastMessage", ListSortDirection.Descending));

ListViewMenuChat.Items.Refresh();

}

}

int indexContact = contactList.FindIndex(contact => contact.UserContact.ID == userId);

if (indexContact != -1)

{

contactList[indexContact].UserContact.AvatarUser = avatarUser;

}

if (UserCurrent.CompanyId != 0)

{

int indexContactCompany = contactCompany.FindIndex(contact => contact.UserContact.ID == userId);

if (indexContactCompany != -1)

{

contactCompany[indexContactCompany].UserContact.AvatarUser = avatarUser;

}

}

if (SelectionConversation != null && SelectionConversation.UserContact.ID == userId)

{

int indexConversation = ChatList.FindIndex(con => con.UserContact.ID == userId && con.isGroup == 0);

if (indexConversation != -1)

{

SelectionConversation = ChatList[indexConversation];

if (ChatList[indexConversation].UserContact.ID == userId)

{

inforConversation.DataContext = null;

infoChat.DataContext = null;

inforConversation.DataContext = SelectionConversation;

infoChat.DataContext = SelectionConversation;

}

}

}

}

DisplayContact();

}));

});

Socket.WIO.On("UpdateStatus", response =>

{

int userId = response.GetValue<int>(0);

string statusUser = response.GetValue<string>(1);

if (userId == userCurrent.ID)

{

User userNewName = UserCurrent;

userNewName.Status = statusUser;

UserCurrent = userNewName;

}

else

{

this.Dispatcher.Invoke(new Action(() =>

{

int index = ChatList.FindIndex(con => con.UserContact.ID == userId && con.isGroup == 0);

if (index != -1)

{

ChatList[index].UserContact.Status = statusUser;

ChatList[index].Status = statusUser;

if (ChatList[index].IsFavorite == 1)

{

ListViewMenuChatFavorite.Items.SortDescriptions.Add(new SortDescription("TimeLastMessage", ListSortDirection.Descending));

ListViewMenuChatFavorite.Items.Refresh();

}

else

{

ListViewMenuChat.Items.SortDescriptions.Add(new SortDescription("TimeLastMessage", ListSortDirection.Descending));

ListViewMenuChat.Items.Refresh();

}

}

int indexContact = contactList.FindIndex(contact => contact.UserContact.ID == userId);

if (indexContact != -1)

{

contactList[indexContact].UserContact.Status = statusUser;

}

if (UserCurrent.CompanyId != 0)

{

int indexContactCompany = contactCompany.FindIndex(contact => contact.UserContact.ID == userId);

if (indexContactCompany != -1)

{

contactCompany[indexContactCompany].UserContact.Status = statusUser;

}

}

if (SelectionConversation != null && SelectionConversation.UserContact.ID == userId)

{

int indexConversation = ChatList.FindIndex(con => con.UserContact.ID == userId && con.isGroup == 0);

if (indexConversation != -1)

{

SelectionConversation = ChatList[indexConversation];

if (ChatList[indexConversation].UserContact.ID == userId)

{

inforConversation.DataContext = null;

infoChat.DataContext = null;

inforConversation.DataContext = SelectionConversation;

infoChat.DataContext = SelectionConversation;

}

}

}

}));

}

});

Socket.WIO.On("DeleteMessage", response =>

{

int conversationId = response.GetValue<int>(0);

string messageId = response.GetValue<string>(1);

int index = ChatList.FindIndex(con => con.ConversationID == conversationId);

if (index != -1)

{

if (ChatList[index].ListMess != null && ChatList[index].ListMess.Count != 0)

{

int indexMessage = ChatList[index].ListMess.FindIndex(mess => mess.MessageId != null && mess.MessageId == messageId);

if (indexMessage == ChatList[index].ListMess.Count - 2 || indexMessage == ChatList[index].ListMess.Count - 1)

{

if (SelectionConversation != null && SelectionConversation.ConversationID == conversationId)

{

if (indexMessage == SelectionConversation.ListMess.Count - 2)

{

if (ChatList[index].ListMess[indexMessage].MessageId == ChatList[index].ListMess[indexMessage + 1].MessageId)

{

ChatList[index].LastChat = ChatList[index].ListMess[indexMessage - 1].Message;

ChatList[index].TimeLastMessage = ChatList[index].ListMess[indexMessage - 1].CreateAt;

ChatList[index].setLastChat(ChatList[index].ListMess[indexMessage - 1].MessageType, ChatList[index].ListMess[indexMessage - 1].Message);

if (ChatList[index].isGroup == 1)

{

ChatList[index].AvatarLastSender = ChatList[index].ListMess[indexMessage - 1].UserSender.ImageAvatar;

}

ChatList[index].Time = ChatList[index].ListMess[indexMessage - 1].CreateAt.ToString("h:mm tt");

}

int indexMessage2 = SelectionConversation.ListMess.FindIndex(mess => mess.MessageId != null && mess.MessageId == messageId);

if (indexMessage2 != -1)

{

this.Dispatcher.Invoke(new Action(() =>

{

DisplayMessage.Items.Remove(ChatList[index].ListMess[indexMessage2]);

DisplayMessage.Items.Refresh();

}));

SelectionConversation.ListMess.Remove(SelectionConversation.ListMess[indexMessage2]);

}

indexMessage2 = SelectionConversation.ListMess.FindIndex(mess => mess.MessageId != null && mess.MessageId == messageId);

if (indexMessage2 != -1)

{

this.Dispatcher.Invoke(new Action(() =>

{

DisplayMessage.Items.Remove(ChatList[index].ListMess[indexMessage2]);

DisplayMessage.Items.Refresh();

}));

SelectionConversation.ListMess.Remove(SelectionConversation.ListMess[indexMessage2]);

}

}

else

{

ChatList[index].LastChat = ChatList[index].ListMess[indexMessage - 1].Message;

ChatList[index].TimeLastMessage = ChatList[index].ListMess[indexMessage - 1].CreateAt;

ChatList[index].setLastChat(ChatList[index].ListMess[indexMessage - 1].MessageType, ChatList[index].ListMess[indexMessage - 1].Message);

if (ChatList[index].isGroup == 1)

{

ChatList[index].AvatarLastSender = ChatList[index].ListMess[indexMessage - 1].UserSender.ImageAvatar;

}

ChatList[index].Time = ChatList[index].ListMess[indexMessage - 1].CreateAt.ToString("h:mm tt");

int indexMessage3 = SelectionConversation.ListMess.FindIndex(mess => mess.MessageId != null && mess.MessageId == messageId);

if (indexMessage3 != -1)

{

this.Dispatcher.Invoke(new Action(() =>

{

DisplayMessage.Items.Remove(ChatList[index].ListMess[indexMessage3]);

DisplayMessage.Items.Refresh();

}));

if (SelectionConversation.ListMess.Contains(SelectionConversation.ListMess[indexMessage3]))

{

SelectionConversation.ListMess.Remove(SelectionConversation.ListMess[indexMessage3]);

}

}

}

}

}

else

{

if (indexMessage != -1)

{

if (ChatList[index].ListMess[indexMessage + 1].MessageId.Equals(messageId))

{

++indexMessage;

}

if (SelectionConversation != null && SelectionConversation.ConversationID == conversationId)

{

this.Dispatcher.Invoke(new Action(() =>

{

DisplayMessage.Items.Remove(ChatList[index].ListMess[indexMessage]);

DisplayMessage.Items.Refresh();

}));

SelectionConversation.ListMess.Remove(SelectionConversation.ListMess[indexMessage]);

}

ChatList[index].ListMess.Remove(ChatList[index].ListMess[indexMessage]);

}

}

}

this.Dispatcher.Invoke(new Action(() =>

{

ListViewMenuChat.Items.Refresh();

ListViewMenuChat.Items.SortDescriptions.Add(new SortDescription("TimeLastMessage", ListSortDirection.Descending));

ListViewMenuChatFavorite.Items.Refresh();

ListViewMenuChatFavorite.Items.SortDescriptions.Add(new SortDescription("TimeLastMessage", ListSortDirection.Descending));

}));

}

});

Socket.WIO.On("EditMessage", response =>

{

int conversationId = response.GetValue<int>(0);

string messageId = response.GetValue<string>(1);

string message = response.GetValue<string>(1);

int index = ChatList.FindIndex(con => con.ConversationID == conversationId);

if (index != -1)

{

if (ChatList[index].ListMess != null && ChatList[index].ListMess.Count != 0)

{

int indexMessage = ChatList[index].ListMess.FindIndex(mess => mess.MessageId != null && mess.MessageId == messageId);

if (indexMessage == ChatList[index].ListMess.Count - 1)

{

ChatList[index].LastChat = message;

}

else

{

if (ChatList[index].ListMess[indexMessage + 1].MessageId.Equals(messageId))

{

++indexMessage;

}

}

ChatList[index].ListMess[indexMessage].Message = message;

if (SelectionConversation != null && SelectionConversation.ConversationID == conversationId)

{

SelectionConversation.ListMess[indexMessage].Message = message;

}

this.Dispatcher.Invoke(new Action(() =>

{

DisplayMessage.Items.Refresh();

ListViewMenuChat.Items.SortDescriptions.Add(new SortDescription("TimeLastMessage", ListSortDirection.Descending));

ListViewMenuChat.Items.Refresh();

}));

}

}

});

Socket.WIO.On("SendMessage", response =>

{

Messages messageInfo = response.GetValue<Messages>(0);

if (ChatList != null && ChatList.Count != 0)

{

int index = ChatList.FindIndex(con => con.ConversationID == messageInfo.ConversationID);

if (index != -1)

{

if (ChatList[index].IsHidden == 1)

{

this.Dispatcher.Invoke(new Action(() =>

{

ChatList[index].IsHidden = 0;

HiddenConversation(ChatList[index].ConversationID, 0, UserCurrent.ID);

}));

}

index = ChatList.FindIndex(con => con.ConversationID == messageInfo.ConversationID);

ChatList[index].LastChat = messageInfo.Message;

ChatList[index].setLastChat(messageInfo.MessageType, messageInfo.Message);

ChatList[index].TimeLastMessage = messageInfo.CreateAt;

ChatList[index].Time = messageInfo.CreateAt.ToString("h:mm tt");

int indexMemember = ChatList[index].ListMember.FindIndex(member => member.ID == messageInfo.SenderID);

if (indexMemember != -1)

{

if (ChatList[index].isGroup == 1)

{

ChatList[index].AvatarLastSender = ChatList[index].ListMember[indexMemember].ImageAvatar;

}

if (Properties.Settings.Default.Notification == true)

{

try

{

this.Dispatcher.Invoke(new Action(() =>

{

if (!this.IsActive)

{

if (messageInfo.SenderID != UserCurrent.ID)

{

if (messageInfo.MessageType.Equals("sendFile") || messageInfo.MessageType.Equals("sendPhoto"))

{

string fileName = "";

if (!String.IsNullOrEmpty(messageInfo.NameFile))

{

for (int j = 0; j < messageInfo.NameFile.Length; j++)

{

if (messageInfo.NameFile[j] == '-')

{

fileName = messageInfo.NameFile.Substring(j + 1);

break;

}

}

}

notification = new ToastContentBuilder().AddText(ChatList[index].Name)

.AddText(ChatList[index].ListMember[indexMemember].UserName + ": " + fileName)

.AddAppLogoOverride(new Uri(ChatList[index].ListMember[indexMemember].AvatarUser), ToastGenericAppLogoCrop.Circle);

}

else

{

notification = new ToastContentBuilder().AddText(ChatList[index].Name)

.AddText(ChatList[index].ListMember[indexMemember].UserName + ": " + messageInfo.Message)

.AddAppLogoOverride(new Uri(ChatList[index].ListMember[indexMemember].AvatarUser), ToastGenericAppLogoCrop.Circle);

//.AddInputTextBox("tbReply", placeHolderContent: "Nhập để trả lời tin nhắn").AddButton(new ToastButton().SetContent("Reply").AddArgument("action", "reply").SetBackgroundActivation())

}

notification.Show();

}

}

}));

}

catch (Exception ex)

{

}

}

}

if (ChatList[index].ListMess != null && ChatList[index].ListMess.FindIndex(mess => mess.MessageId != null && mess.MessageId.Equals(messageInfo.MessageID)) == -1)

{

DisplayNewMessage(messageInfo);

}

if (ChatList[index].ListMess == null)

{

index = ChatList.FindIndex(chat => chat.ConversationID == messageInfo.ConversationID);

if (index != -1)

{

ChatList[index].CountMessage++;

if (UserCurrent.ID != messageInfo.SenderID)

{

++ChatList[index].Count;

CountConversationUnreader = ChatList.Where(con => con.HaveUnReader == "True").Count();

}

}

}

int index2 = ChatList.FindIndex(chat => chat.ConversationID == messageInfo.ConversationID);

if (ChatList[index2].IsFavorite == 1)

{

this.Dispatcher.Invoke(new Action(() =>

{

ListViewMenuChatFavorite.Items.Refresh();

ListViewMenuChatFavorite.Items.SortDescriptions.Add(new SortDescription("TimeLastMessage", ListSortDirection.Descending));

ChatList.Sort(new CompareByTimeLastMessage());

}));

}

else

{

this.Dispatcher.Invoke(new Action(() =>

{

ListViewMenuChat.Items.Refresh();

ListViewMenuChat.Items.SortDescriptions.Add(new SortDescription("TimeLastMessage", ListSortDirection.Descending));

ChatList.Sort(new CompareByTimeLastMessage());

}));

}

}

else

{

this.Dispatcher.Invoke(new Action(() =>

{

DisplayNewConversation(messageInfo.ConversationID);

}));

}

}

else

{

this.Dispatcher.Invoke(new Action(() =>

{

DisplayNewConversation(messageInfo.ConversationID);

}));

}

});

Socket.WIO.On("OutTyping", response =>

{

int userId = response.GetValue<int>(0);

int conversationId = response.GetValue<int>(1);

if (ChatList != null && ChatList.Count != 0)

{

int index = ChatList.FindIndex(con => con.ConversationID == conversationId);

if (index != -1)

{

ChatList[index].UserNameTyping = "";

int indexUser = ChatList[index].ListMember.FindIndex(contact => contact.ID == userId);

if (indexUser != -1)

{

if (SelectionConversation != null && SelectionConversation.ConversationID == conversationId)

{

this.Dispatcher.Invoke(new Action(() =>

{

if (SelectionConversation.ListMember[indexUser].UserName.Equals(NameUserTyping.Text))

{

NameUserTyping.Text = "";

Bordertyping.Visibility = Visibility.Collapsed;

}

}));

}

}

}

}

});

Socket.WIO.On("Typing", response =>

{

int userId = response.GetValue<int>(0);

int conversationId = response.GetValue<int>(1);

if (ChatList != null && ChatList.Count != 0)

{

int index = ChatList.FindIndex(con => con.ConversationID == conversationId);

if (index != -1)

{

int indexUser = ChatList[index].ListMember.FindIndex(contact => contact.ID == userId);

if (indexUser != -1)

{

ChatList[index].UserNameTyping = ChatList[index].ListMember[indexUser].UserName;

if (SelectionConversation != null && SelectionConversation.ConversationID == conversationId)

{

this.Dispatcher.Invoke(new Action(() =>

{

NameUserTyping.Text = ChatList[index].ListMember[indexUser].UserName;

Bordertyping.Visibility = Visibility.Visible;

}));

}

}

}

}

});

Socket.WIO.On("AddFriend", response =>

{

int userId = response.GetValue<int>(0);

int contactId = response.GetValue<int>(1);

this.Dispatcher.Invoke(new Action(() =>

{

if (contactId == UserCurrent.ID)

{

if (requestContactList.Count == 0)

{

requestContactList.Add(new RequestContact(UserCurrent.ID, userId, "request"));

}

else

{

for (int i = 0; i < requestContactList.Count; i++)

{

if (requestContactList[i].contactId == userId)

{

requestContactList[i].status = "request";

break;

}

if (i == requestContactList.Count - 1)

{

requestContactList.Add(new RequestContact(UserCurrent.ID, userId, "request"));

}

}

}

if (SelectionConversation != null && SelectionConversation.isGroup == 0 && SelectionConversation.UserContact.ID == userId)

{

using (WebClient httpClient = new WebClient())

{

try

{

httpClient.QueryString.Clear();

httpClient.QueryString.Add("ID", userId.ToString());

httpClient.UploadValuesCompleted += (sender, e) =>

{

APIUser receiveInfoAPI = JsonConvert.DeserializeObject<APIUser>(UnicodeEncoding.UTF8.GetString(e.Result

));

if (receiveInfoAPI.data != null)

{

UserFromServer userMessageFromSerVer = receiveInfoAPI.data.user\_info;

User userContact = new User(userMessageFromSerVer.ID, userMessageFromSerVer.ID365, userMessageFromSerVer.Type365, userMessageFromSerVer.Email, userMessageFromSerVer.Password, userMessageFromSerVer.Phone, userMessageFromSerVer.UserName, userMessageFromSerVer.AvatarUser, userMessageFromSerVer.Status, userMessageFromSerVer.StatusEmotion, userMessageFromSerVer.LastActive, userMessageFromSerVer.Active, userMessageFromSerVer.isOnline, userMessageFromSerVer.Looker, userMessageFromSerVer.CompanyId, userMessageFromSerVer.CompanyName);

ClickUser(userContact);

}

};

httpClient.UploadValuesAsync(new Uri(URL + "User/GetInfoUser"), "POST", httpClient.QueryString);

}

catch (Exception ex)

{

Socket\_NoInternet();

}

}

}

}

else

{

if (requestContactList.Count == 0)

{

requestContactList.Add(new RequestContact(UserCurrent.ID, contactId, "send"));

}

else

{

for (int i = 0; i < requestContactList.Count; i++)

{

if (requestContactList[i].contactId == contactId)

{

requestContactList[i].status = "send";

break;

}

if (i == requestContactList.Count - 1)

{

requestContactList.Add(new RequestContact(UserCurrent.ID, contactId, "send"));

break;

}

}

}

if (SelectionConversation != null && SelectionConversation.isGroup == 0 && SelectionConversation.UserContact.ID == userId)

{

using (WebClient httpClient = new WebClient())

{

try

{

httpClient.QueryString.Clear();

httpClient.QueryString.Add("ID", contactId.ToString());

httpClient.UploadValuesCompleted += (sender, e) =>

{

APIUser receiveInfoAPI = JsonConvert.DeserializeObject<APIUser>(UnicodeEncoding.UTF8.GetString(e.Result));

if (receiveInfoAPI.data != null)

{

UserFromServer userMessageFromSerVer = receiveInfoAPI.data.user\_info;

User userContact = new User(userMessageFromSerVer.ID, userMessageFromSerVer.ID365, userMessageFromSerVer.Type365, userMessageFromSerVer.Email, userMessageFromSerVer.Password, userMessageFromSerVer.Phone, userMessageFromSerVer.UserName, userMessageFromSerVer.AvatarUser, userMessageFromSerVer.Status, userMessageFromSerVer.StatusEmotion, userMessageFromSerVer.LastActive, userMessageFromSerVer.Active, userMessageFromSerVer.isOnline, userMessageFromSerVer.Looker, userMessageFromSerVer.CompanyId, userMessageFromSerVer.CompanyName);

ClickUser(userContact);

}

};

httpClient.UploadValuesAsync(new Uri(URL + "User/GetInfoUser"), "POST", httpClient.QueryString);

}

catch (Exception ex)

{

Socket\_NoInternet();

}

}

}

}

}));

});

Socket.WIO.On("DecilineRequestAddFriend", response =>

{

int userId = response.GetValue<int>(0);

int contactId = response.GetValue<int>(1);

this.Dispatcher.Invoke(new Action(() =>

{

if (contactId == UserCurrent.ID)

{

for (int i = 0; i < requestContactList.Count; i++)

{

if (userId == requestContactList[i].contactId)

{

requestContactList[i].status = "deciline";

break;

}

}

if (SelectionConversation != null && SelectionConversation.isGroup == 0 && SelectionConversation.UserContact.ID == userId)

{

using (WebClient httpClient = new WebClient())

{

try

{

httpClient.QueryString.Clear();

httpClient.QueryString.Add("ID", userId.ToString());

httpClient.UploadValuesCompleted += (sender, e) =>

{

APIUser receiveInfoAPI = JsonConvert.DeserializeObject<APIUser>(UnicodeEncoding.UTF8.GetString(e.Result));

if (receiveInfoAPI.data != null)

{

UserFromServer userMessageFromSerVer = receiveInfoAPI.data.user\_info;

User userContact = new User(userMessageFromSerVer.ID, userMessageFromSerVer.ID365, userMessageFromSerVer.Type365, userMessageFromSerVer.Email, userMessageFromSerVer.Password, userMessageFromSerVer.Phone, userMessageFromSerVer.UserName, userMessageFromSerVer.AvatarUser, userMessageFromSerVer.Status, userMessageFromSerVer.StatusEmotion, userMessageFromSerVer.LastActive, userMessageFromSerVer.Active, userMessageFromSerVer.isOnline, userMessageFromSerVer.Looker, userMessageFromSerVer.CompanyId, userMessageFromSerVer.CompanyName);

ClickUser(userContact);

}

};

httpClient.UploadValuesAsync(new Uri(URL + "User/GetInfoUser"), "POST", httpClient.QueryString);

}

catch (Exception ex)

{

Socket\_NoInternet();

}

}

}

}

else

{

for (int i = 0; i < requestContactList.Count; i++)

{

if (requestContactList[i].contactId == contactId)

{

requestContactList.RemoveAt(i);

break;

}

if (SelectionConversation != null && SelectionConversation.isGroup == 0 && SelectionConversation.UserContact.ID == userId)

{

using (WebClient httpClient = new WebClient())

{

try

{

httpClient.QueryString.Clear();

httpClient.QueryString.Add("ID", userId.ToString());

httpClient.UploadValuesCompleted += (sender, e) =>

{

APIUser receiveInfoAPI = JsonConvert.DeserializeObject<APIUser>(UnicodeEncoding.UTF8.GetString(e.Result));

if (receiveInfoAPI.data != null)

{

UserFromServer userMessageFromSerVer = receiveInfoAPI.data.user\_info;

User userContact = new User(userMessageFromSerVer.ID, userMessageFromSerVer.ID365, userMessageFromSerVer.Type365, userMessageFromSerVer.Email, userMessageFromSerVer.Password, userMessageFromSerVer.Phone, userMessageFromSerVer.UserName, userMessageFromSerVer.AvatarUser, userMessageFromSerVer.Status, userMessageFromSerVer.StatusEmotion, userMessageFromSerVer.LastActive, userMessageFromSerVer.Active, userMessageFromSerVer.isOnline, userMessageFromSerVer.Looker, userMessageFromSerVer.CompanyId, userMessageFromSerVer.CompanyName);

ClickUser(userContact);

}

};

httpClient.UploadValuesAsync(new Uri(URL + "User/GetInfoUser"), "POST", httpClient.QueryString);

}

catch (Exception ex)

{

Socket\_NoInternet();

}

}

}

}

}

}));

});

Socket.WIO.On("AcceptRequestAddFriend", response =>

{

int userId = response.GetValue<int>(0);

int contactId = response.GetValue<int>(1);

this.Dispatcher.Invoke(new Action(() =>

{

if (contactId == UserCurrent.ID)

{

for (int i = 0; i < requestContactList.Count; i++)

{

if (userId == requestContactList[i].contactId)

{

requestContactList[i].status = "accept";

break;

}

}

using (WebClient httpClient = new WebClient())

{

try

{

httpClient.QueryString.Clear();

httpClient.QueryString.Add("ID", userId.ToString());

httpClient.UploadValuesCompleted += (sender, e) =>

{

APIUser receiveInfoAPI = JsonConvert.DeserializeObject<APIUser>(UnicodeEncoding.UTF8.GetString(e.Result));

if (receiveInfoAPI.data != null)

{

UserFromServer userMessageFromSerVer = receiveInfoAPI.data.user\_info;

User userContact = new User(userMessageFromSerVer.ID, userMessageFromSerVer.ID365, userMessageFromSerVer.Type365, userMessageFromSerVer.Email, userMessageFromSerVer.Password, userMessageFromSerVer.Phone, userMessageFromSerVer.UserName, userMessageFromSerVer.AvatarUser, userMessageFromSerVer.Status, userMessageFromSerVer.StatusEmotion, userMessageFromSerVer.LastActive, userMessageFromSerVer.Active, userMessageFromSerVer.isOnline, userMessageFromSerVer.Looker, userMessageFromSerVer.CompanyId, userMessageFromSerVer.CompanyName);

contactList.Add(new MenuContact(userContact, 0));

if (lbSelectionTypeContact.Text.Equals(App.Current.Resources["TextMyContact"].ToString()))

{

DisplayContact();

}

if (SelectionConversation != null && SelectionConversation.isGroup == 0 && SelectionConversation.UserContact.ID == userId)

{

ClickUser(userContact);

}

}

};

httpClient.UploadValuesAsync(new Uri(URL + "User/GetInfoUser"), "POST", httpClient.QueryString);

}

catch (Exception ex)

{

Socket\_NoInternet();

}

}

}

else

{

for (int i = 0; i < requestContactList.Count; i++)

{

if (requestContactList[i].contactId == contactId)

{

requestContactList.RemoveAt(i);

break;

}

}

using (WebClient httpClient = new WebClient())

{

try

{

httpClient.QueryString.Clear();

httpClient.QueryString.Add("ID", contactId.ToString());

httpClient.UploadValuesCompleted += (sender, e) =>

{

APIUser receiveInfoAPI = JsonConvert.DeserializeObject<APIUser>(UnicodeEncoding.UTF8.GetString(e.Result));

if (receiveInfoAPI.data != null)

{

UserFromServer userMessageFromSerVer = receiveInfoAPI.data.user\_info;

User userContact = new User(userMessageFromSerVer.ID, userMessageFromSerVer.ID365, userMessageFromSerVer.Type365, userMessageFromSerVer.Email, userMessageFromSerVer.Password, userMessageFromSerVer.Phone, userMessageFromSerVer.UserName, userMessageFromSerVer.AvatarUser, userMessageFromSerVer.Status, userMessageFromSerVer.StatusEmotion, userMessageFromSerVer.LastActive, userMessageFromSerVer.Active, userMessageFromSerVer.isOnline, userMessageFromSerVer.Looker, userMessageFromSerVer.CompanyId, userMessageFromSerVer.CompanyName);

contactList.Add(new MenuContact(userContact, 0));

if (lbSelectionTypeContact.Text.Equals(App.Current.Resources["TextMyContact"].ToString()))

{

DisplayContact();

}

ClickUser(userContact);

}

};

httpClient.UploadValuesAsync(new Uri(URL + "User/GetInfoUser"), "POST", httpClient.QueryString);

}

catch (Exception ex)

{

Socket\_NoInternet();

}

}

}

}));

});

Socket.WIO.On("ReadMessage", response =>

{

int userId = response.GetValue<int>(0);

int conversationId = response.GetValue<int>(1);

int index = ChatList.FindIndex(con => con.ConversationID == conversationId);

if (index != -1)

{

if (UserCurrent.ID == userId)

{

ChatList[index].Count = 0;

if (ChatList[index].IsFavorite == 1)

{

this.Dispatcher.Invoke(new Action(() =>

{

ListViewMenuChatFavorite.Items.Refresh();

CountConversationUnreader = ChatList.Where(con => con.HaveUnReader == "True").Count();

}));

}

else

{

this.Dispatcher.Invoke(new Action(() =>

{

ListViewMenuChat.Items.Refresh();

CountConversationUnreader = ChatList.Where(con => con.HaveUnReader == "True").Count();

}));

}

}

else

{

if (ChatList[index].ListMember.Count <= 5)

{

int indexUser = ChatList[index].ListMember.FindIndex(contact => contact.ID == userId);

if (indexUser != -1)

{

index = ChatList.FindIndex(con => con.ConversationID == conversationId);

if (index != -1)

{

ChatList[index].ListMember[indexUser].UnReader = 0;

}

index = ChatList.FindIndex(con => con.ConversationID == conversationId);

if (index != -1 && ChatList[index].ListMess != null && ChatList[index].ListMess.Count != 0)

{

bool isSelection = false;

if (SelectionConversation != null && SelectionConversation.ConversationID == conversationId)

{

isSelection = true;

}

int indexMessage = ChatList[index].ListMess.FindIndex(mess => mess.ListMemberSeen != null && mess.ListMemberSeen.Contains(ChatList[index].ListMember[indexUser]));

if (indexMessage != -1)

{

if (ChatList[index].ListMess[indexMessage].ListMemberSeen.Count == 1)

{

ChatList[index].ListMess.RemoveAt(indexMessage);

if (isSelection)

{

this.Dispatcher.Invoke(new Action(() =>

{

DisplayMessage.Items.RemoveAt(indexMessage);

}));

}

}

else

{

ChatList[index].ListMess[indexMessage].ListMemberSeen.Remove(ChatList[index].ListMember[indexUser]);

if (isSelection)

{

this.Dispatcher.Invoke(new Action(() =>

{

DisplayMessage.Items[indexMessage] = ChatList[index].ListMess[indexMessage];

}));

}

}

if (ChatList[index].ListMess[ChatList[index].ListMess.Count - 1].ListMemberSeen != null)

{

ChatList[index].ListMess[ChatList[index].ListMess.Count - 1].ListMemberSeen.Add(ChatList[index].ListMember[indexUser]);

if (isSelection)

{

this.Dispatcher.Invoke(new Action(() =>

{

DisplayMessage.Items[ChatList[index].ListMess.Count - 1] = ChatList[index].ListMess[ChatList[index].ListMess.Count - 1];

}));

}

}

else

{

List<MemberConversation> memberSenner = new List<MemberConversation>();

memberSenner.Add(ChatList[index].ListMember[indexUser]);

ChatList[index].ListMess.Add(new DisplayMessage(memberSenner));

if (isSelection)

{

this.Dispatcher.Invoke(new Action(() =>

{

DisplayMessage.Items.Add(new DisplayMessage(memberSenner));

}));

}

}

}

}

}

}

}

}

});

Socket.WIO.On("MarkUnreader", response =>

{

int conversationId = response.GetValue<int>(0);

int index = ChatList.FindIndex(con => con.ConversationID == conversationId);

if (index != -1)

{

ChatList[index].Count = 1;

if (ChatList[index].IsFavorite == 1)

{

this.Dispatcher.Invoke(new Action(() =>

{

ListViewMenuChatFavorite.Items.Refresh();

CountConversationUnreader = ChatList.Where(con => con.HaveUnReader == "True").Count();

}));

}

else

{

this.Dispatcher.Invoke(new Action(() =>

{

ListViewMenuChat.Items.Refresh();

CountConversationUnreader = ChatList.Where(con => con.HaveUnReader == "True").Count();

}));

}

}

});

Socket.WIO.On("DeleteContact", response =>

{

int userId = response.GetValue<int>(0);

int contactId = response.GetValue<int>(1);

this.Dispatcher.Invoke(new Action(() =>

{

for (int i = 0; i < contactList.Count; i++)

{

if (contactList[i].UserContact.ID == contactId || contactList[i].UserContact.ID == userId)

{

User contact = contactList[i].UserContact;

contactList.Remove(contactList[i]);

ClickUser(contact);

}

}

}));

});

Socket.WIO.On("OutGroup", response =>

{

int conversationId = response.GetValue<int>(0);

int userId = response.GetValue<int>(1);

int adminId = response.GetValue<int>(2);

this.Dispatcher.Invoke(new Action(() =>

{

int index = ChatList.FindIndex(con => con.ConversationID == conversationId);

if (index != -1)

{

MenuChat conversation = ChatList[index];

if (UserCurrent.ID == userId)

{

ListViewMenuChat.Items.Remove(ChatList[index]);

ListViewMenuChat.Items.Refresh();

ChatList.Remove(conversation);

for (int j = 0; j < callList.Count; j++)

{

if (callList[j].ConversationID == conversation.ConversationID)

{

callList.Remove(callList[j]);

break;

}

}

if (SelectionConversation != null && SelectionConversation.ConversationID == conversationId)

{

chatPage.Visibility = Visibility.Hidden;

homePage.Visibility = Visibility.Visible;

}

}

else

{

int index2 = conversation.ListMember.FindIndex(member => member.ID == userId);

for (int k = 0; k < index2; k++)

{

if (conversation.ListMember[k].ID == UserCurrent.ID)

{

index2--;

break;

}

}

conversation.ListMember.RemoveAt(index2 + 1);

conversation.setNameConversation(conversation.GroupName, conversation.ListMember, userCurrent.ID);

conversation.Status = conversation.ListMember.Count + "";

if (UserCurrent.ID == adminId)

{

conversation.IsAdmin = true;

}

conversation.AdminId = adminId;

index = ChatList.FindIndex(con => con.ConversationID == conversation.ConversationID);

ChatList[index] = conversation;

ListViewMenuChat.Items.Refresh();

if (SelectionConversation != null && SelectionConversation.ConversationID == conversationId)

{

this.Dispatcher.Invoke(new Action(() =>

{

SelectionConversation = conversation;

inforConversation.DataContext = null;

infoChat.DataContext = null;

inforConversation.DataContext = SelectionConversation;

infoChat.DataContext = SelectionConversation;

ListViewInfoOfGroup.Items.RemoveAt(index2);

}));

}

}

}

}));

});

Socket.WIO.On("changeNameGroup", response =>

{

int conversationId = response.GetValue<int>(0);

string nameGroup = response.GetValue<string>(1);

int index = ChatList.FindIndex(con => con.ConversationID == conversationId);

if (index != -1)

{

MenuChat conversation = ChatList[index];

conversation.GroupName = nameGroup;

conversation.setNameConversation(conversation.GroupName, conversation.ListMember, UserCurrent.ID);

this.Dispatcher.Invoke(new Action(() =>

{

index = ChatList.FindIndex(con => con.ConversationID == conversation.ConversationID);

ChatList[index] = conversation;

ListViewMenuChat.Items.Refresh();

}));

if (SelectionConversation != null && SelectionConversation.ConversationID == conversation.ConversationID)

{

this.Dispatcher.Invoke(new Action(() =>

{

SelectionConversation = conversation;

inforConversation.DataContext = null;

infoChat.DataContext = null;

inforConversation.DataContext = SelectionConversation;

infoChat.DataContext = SelectionConversation;

}));

}

}

});

Socket.WIO.On("changeAvatarGroup", response =>

{

int conversationId = response.GetValue<int>(0);

string avararGroup = response.GetValue<string>(1);

this.Dispatcher.Invoke(new Action(() =>

{

int index = ChatList.FindIndex(con => con.ConversationID == conversationId);

if (index != -1)

{

ChatList[index].AvatarUser = avararGroup;

ListViewMenuChat.Items.Refresh();

if (SelectionConversation != null && SelectionConversation.ConversationID == ChatList[index].ConversationID)

{

SelectionConversation = ChatList[index];

inforConversation.DataContext = null;

infoChat.DataContext = null;

inforConversation.DataContext = SelectionConversation;

infoChat.DataContext = SelectionConversation;

}

}

}));

});

Socket.WIO.On("shareGroupFromLink", response =>

{

int conversationId = response.GetValue<int>(0);

int shareGroupFromLink = response.GetValue<int>(1);

int index = ChatList.FindIndex(con => con.ConversationID == conversationId);

if (index != -1)

{

MenuChat conversation = ChatList[index];

conversation.ShareGroupFromLink = shareGroupFromLink;

this.Dispatcher.Invoke(new Action(() =>

{

index = ChatList.FindIndex(con => con.ConversationID == conversation.ConversationID);

ChatList[index] = conversation;

ListViewMenuChat.Items.Refresh();

}));

if (SelectionConversation != null && SelectionConversation.ConversationID == conversation.ConversationID)

{

this.Dispatcher.Invoke(new Action(() =>

{

SelectionConversation = conversation;

inforConversation.DataContext = null;

infoChat.DataContext = null;

inforConversation.DataContext = SelectionConversation;

infoChat.DataContext = SelectionConversation;

}));

}

}

});

Socket.WIO.On("changeBowserMemberGroup", response =>

{

int conversationId = response.GetValue<int>(0);

int BowserMemberGroup = response.GetValue<int>(1);

int index = ChatList.FindIndex(con => con.ConversationID == conversationId);

if (index != -1)

{

MenuChat conversation = ChatList[index];

conversation.BrowseMember = BowserMemberGroup;

this.Dispatcher.Invoke(new Action(() =>

{

index = ChatList.FindIndex(con => con.ConversationID == conversation.ConversationID);

ChatList[index] = conversation;

ListViewMenuChat.Items.Refresh();

}));

if (SelectionConversation != null && SelectionConversation.ConversationID == conversation.ConversationID)

{

this.Dispatcher.Invoke(new Action(() =>

{

SelectionConversation = conversation;

inforConversation.DataContext = null;

infoChat.DataContext = null;

inforConversation.DataContext = SelectionConversation;

infoChat.DataContext = SelectionConversation;

}));

}

}

});

Socket.WIO.On("AddNewMemberToGroup", response =>

{

int conversationId = response.GetValue<int>(0);

int[] listMember = response.GetValue<int[]>(1);

this.Dispatcher.Invoke(new Action(() =>

{

int index = ChatList.FindIndex(conn => conn.ConversationID == conversationId);

if (index != -1)

{

foreach (int j in listMember)

{

using (WebClient httpClient = new WebClient())

{

try

{

httpClient.QueryString.Clear();

httpClient.QueryString.Add("ID", j.ToString());

httpClient.UploadValuesCompleted += (sender, e) =>

{

APIUser receiveInfoAPI = JsonConvert.DeserializeObject<APIUser>(UnicodeEncoding.UTF8.GetString(e.Result));

if (receiveInfoAPI != null)

{

UserFromServer userSenderFromSerVer = receiveInfoAPI.data.user\_info;

ChatList[index].ListMember.Add(new MemberConversation(userSenderFromSerVer.ID, userSenderFromSerVer.UserName, userSenderFromSerVer.AvatarUser, userSenderFromSerVer.Status, userSenderFromSerVer.StatusEmotion, userSenderFromSerVer.LastActive, userSenderFromSerVer.Active, userSenderFromSerVer.isOnline, 0));

}

};

httpClient.UploadValuesAsync(new Uri(URL + "User/GetInfoUser"), "POST", httpClient.QueryString);

}

catch (Exception ex)

{

Socket\_NoInternet();

}

}

}

ChatList[index].setNameConversation(ChatList[index].GroupName, ChatList[index].ListMember, userCurrent.ID);

ChatList[index].Status = ChatList[index].ListMember.Count + "";

ListViewMenuChat.Items.Refresh();

if (SelectionConversation != null && SelectionConversation.ConversationID == conversationId)

{

ChatList[index].Count = 0;

SelectionConversation = ChatList[index];

inforConversation.DataContext = null;

infoChat.DataContext = null;

inforConversation.DataContext = SelectionConversation;

infoChat.DataContext = SelectionConversation;

ListViewInfoOfGroup.Items.Add(new MenuContact(new User(ChatList[index].ListMember[ChatList[index].ListMember.Count - 1].ID, ChatList[index].ListMember[ChatList[index].ListMember.Count - 1].UserName, ChatList[index].ListMember[ChatList[index].ListMember.Count - 1].AvatarUser, ChatList[index].ListMember[ChatList[index].ListMember.Count - 1].Status, ChatList[index].ListMember[ChatList[index].ListMember.Count - 1].StatusEmotion, ChatList[index].ListMember[ChatList[index].ListMember.Count - 1].LastActive, ChatList[index].ListMember[ChatList[index].ListMember.Count - 1].Active, ChatList[index].ListMember[ChatList[index].ListMember.Count - 1].isOnline), 0));

}

}

}));

});

Socket.WIO.On("AddBrowseMember", response =>

{

int conversationId = response.GetValue<int>(0);

int userId = response.GetValue<int>(1);

int[] listBrowse = response.GetValue<int[]>(2);

this.Dispatcher.Invoke(new Action(() =>

{

int index = ChatList.FindIndex(conn => conn.ConversationID == conversationId);

if (index != -1)

{

for (int j = 0; j < listBrowse.Length; j++)

{

using (WebClient httpClient = new WebClient())

{

try

{

httpClient.QueryString.Clear();

httpClient.QueryString.Add("ID", listBrowse[j].ToString());

httpClient.UploadValuesCompleted += (sender, e) =>

{

APIUser receiveInfoAPI = JsonConvert.DeserializeObject<APIUser>(UnicodeEncoding.UTF8.GetString(e.Result));

if (receiveInfoAPI.data != null)

{

UserFromServer userSenderFromSerVer = receiveInfoAPI.data.user\_info;

if (ChatList[index].ListBrowerMember == null)

{

ChatList[index].ListBrowerMember = new List<BrowseMember>();

}

ChatList[index].ListBrowerMember.Add(new BrowseMember(new User(userSenderFromSerVer.ID, userSenderFromSerVer.ID365, userSenderFromSerVer.Type365, userSenderFromSerVer.Email, userSenderFromSerVer.Password, userSenderFromSerVer.Phone, userSenderFromSerVer.UserName, userSenderFromSerVer.AvatarUser, userSenderFromSerVer.Status, userSenderFromSerVer.StatusEmotion, userSenderFromSerVer.LastActive, userSenderFromSerVer.Active, userSenderFromSerVer.isOnline, userSenderFromSerVer.Looker, userSenderFromSerVer.CompanyId, userSenderFromSerVer.CompanyName), userId));

}

};

httpClient.UploadValuesAsync(new Uri(URL + "User/GetInfoUser"), "POST", httpClient.QueryString);

}

catch (Exception ex)

{

Socket\_NoInternet();

}

}

ListViewMenuChat.Items.Refresh();

if (SelectionConversation != null && SelectionConversation.ConversationID == conversationId)

{

SelectionConversation = ChatList[index];

inforConversation.DataContext = null;

infoChat.DataContext = null;

inforConversation.DataContext = SelectionConversation;

infoChat.DataContext = SelectionConversation;

}

}

}

}));

});

Socket.WIO.On("AddNewConversation", response =>

{

int conversationId = response.GetValue<int>(0);

this.Dispatcher.Invoke(new Action(() =>

{

DisplayNewConversation(conversationId);

CountConversation++;

}));

});

Socket.WIO.On("DeleteBrowse", response =>

{

int conversationId = response.GetValue<int>(0);

int[] listBrowse = response.GetValue<int[]>(1);

this.Dispatcher.Invoke(new Action(() =>

{

int index = ChatList.FindIndex(conn => conn.ConversationID == conversationId);

if (index != -1)

{

if (ChatList[index].ListBrowerMember != null)

{

for (int i = 0; i < listBrowse.Length; i++)

{

for (int j = 0; j < ChatList[index].ListBrowerMember.Count; j++)

{

if (ChatList[index].ListBrowerMember[j].UserMember.ID == listBrowse[i])

{

ChatList[index].ListBrowerMember.Remove(ChatList[index].ListBrowerMember[j]);

break;

}

}

}

SelectionConversation = ChatList[index];

inforConversation.DataContext = null;

infoChat.DataContext = null;

inforConversation.DataContext = SelectionConversation;

infoChat.DataContext = SelectionConversation;

}

}

}));

});