using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using NLog;

using System.IO;

using System.Data.SqlClient;

using System.Data.Entity;

namespace TGTicketingAppEF

{

class Program

{

private static NLog.Logger logger = NLog.LogManager.GetCurrentClassLogger();

string MyValue { get; set;}

static void Main(string[] args)

{

DisplayMenu(true);

string MSelection = Console.ReadLine().Trim();

string LSelection = "";

// connect to DB

logger.Debug("Starting application");

do

{

// User search removed to Notepad for easier look at Ticket

if (MSelection == "2") // Tickets

{

TicketMenu();

LSelection = Console.ReadLine().Trim();

do

{

if (LSelection == "1") // Display

{

Console.WriteLine("You chose Display Tickets");

int tCount = 0;

Console.WriteLine("Enter ticket summary information to search");

string entry = Console.ReadLine();

using (var dbContext = new TicketContext())

{

var results = dbContext.Tickets

.Include(x => x.WatchingUsers.Select(u => u.User))

.Include(t => t.TicketType)

.Where(d => d.Summary.Contains(entry))

.ToList();

foreach (var record in results)

{

tCount++;

if (tCount % 20 == 0)

{

Console.WriteLine("Display more records? y/n");

string continueDisp = Console.ReadLine();

if (continueDisp.ToUpper() == "N")

{

break;

}

}

record.DisplayTickets();

}

logger.Info("{0} was search, {1} returned", entry, tCount);

Console.WriteLine();

Console.WriteLine("-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*");

Console.WriteLine();

break;

}

}

else if (LSelection == "2") // Add

{

Console.WriteLine("Enter a summary");

var sum = Console.ReadLine();

Console.WriteLine("Enter priority level");

var priority = Console.ReadLine();

using (var dbContext = new TicketContext())

{

var record = new Ticket();

record.Summary = sum;

record.Priority = priority;

TicketType typeCheck = null;

do

{

Console.WriteLine("Enter the ticket type");

var ttype = Console.ReadLine();

typeCheck = dbContext.TicketTypes.Where(t => t.Description == ttype).SingleOrDefault();

} while (typeCheck == null);

record.TicketType = typeCheck;

string addUser = "";

do

{

Console.WriteLine("Enter a user ID - when finished enter 'end'");

addUser = Console.ReadLine();

if (addUser != "end")

{

var aUser = Convert.ToInt32(addUser);

var user = dbContext.Users.Where(u => u.UserID == aUser).FirstOrDefault();

if (user != null)

{

User u = new User();

u.UserID = aUser;

record.Assigned.Add(new WatchingUser { User = aUser }); // this should add an assigned user, creating a new WatchingUser item

}

}

} while (addUser != "end");

dbContext.Tickets.Add(record);

logger.Info("Ticket {0} was created", record.TicketID);

dbContext.SaveChanges();

}

}

else if (LSelection == "3") // Update

{

Console.WriteLine("Enter the TicketID you would like to update");

string tID = Console.ReadLine();

int tIDint;

bool success = false;

try

{

tIDint = Convert.ToInt32(tID);

success = true;

}

catch (Exception e)

{

Console.WriteLine("Invalid Data Type");

logger.Error(e.Message);

}

if (success)

{

// update ticket

tIDint = Convert.ToInt32(tID);

Console.WriteLine("Enter new status of ticket");

Console.WriteLine("1) Open");

Console.WriteLine("2) Closed");

string newStat = Console.ReadLine();

if (newStat.Trim() == "1")

{

newStat = "Open";

}

else if (newStat.Trim() == "2")

{

newStat = "Closed";

}

else

{

Console.WriteLine("Bad input");

break;

}

ticket.UpdateTicket(tIDint, newStat);

Console.WriteLine("You updated {0}", tID);

logger.Trace("Ticket {0} was updated", tID);

}

break;

}

else if (LSelection == "4") // Exit

{

return;

}

else

{

Console.WriteLine("Bad Input");

break;

}

} while (LSelection != "4");

}

else if (MSelection == "3") // Exit

{

return;

}

else

{

Console.WriteLine("Bad Input");

break;

}

DisplayMenu(false);

MSelection = Console.ReadLine().Trim();

} while (MSelection != "3");

return;

}

public static void DisplayMenu(bool First)

{

if (First)

{

Console.WriteLine("Welcome to the User and Ticketing System");

}

Console.WriteLine("Select what you would like to do");

Console.WriteLine("1) Users");

Console.WriteLine("2) Tickets");

Console.WriteLine("3) Exit Application");

}

public static void UserMenu()

{

Console.WriteLine("1) Display User Information");

Console.WriteLine("2) Add New User");

Console.WriteLine("3) Update User Information");

Console.WriteLine("4) Exit Application");

}

public static void TicketMenu()

{

Console.WriteLine("1) Display Tickets");

Console.WriteLine("2) Add New Ticket");

Console.WriteLine("3) Update Existing Ticket");

Console.WriteLine("4) Exit Application");

}

}

}