ASP.NET Project – Helpdesk

## Overview

Web базирана backend система за заявки за обслужване на компютри и потребителски проблеми с приложения и техника - Хелпдеск. Включва Инвентарна система за наличната и разпределена по потребители и дирекции компютърна техника - Инвентар.

**Потребителите**, трябва да се регистрират, за да ползват услугите на Хелпдеск и Инвентар.

## Хелдеск система

Потребителите могат да :

* подават **Заявки** / Request
* да виждат само собствените си заявки и техния статус
* да редактират заявката ( да слагат мнение/ уточнение в коментар или да я откажат)

Заявките имат **Статус**:

* Open
* Assigned
* Close
* Waiting customer

**Администраторите** могат да виждат всички заявки и да редактират само статуса. При Close на заявката задължително да пишат коментар, как е отстранен проблема

## Инвентар система

В Инвентар системата въвеждат и редактират Администраторите. Потребителите виждат само заведената им техника.

Техниката има Статус:

* Заведена на Потребител или на Склад
* Подготвена за бракуване
* Бракувана
* Дарена
* Подготвена за дарение

## Identity Requirements

Използва се **default** **IdentityUser с**ъс следните изисквания за парола:

The password requirements for the IdentityUser are the following:

* Require confirmed account: **false**
* Require digits: **false**
* Require non-alphanumeric characters: **false**
* Required uppercase letters: **false**

Това са изисквания за средата за разработка, след това ще се променят.

## Database Requirements

The **Database** of **HelpdeskDB**:

**ASPNETUsers –added fields - ApplicationUser**

* Has **Id –** a unique string  **, Primary Key**
* Has **FirstName – string required** min 3 max length **30 (required)**
* Has **LastName – string required** min 3 max length **50 (required)**
* Has **Address – string required** min 3 max length **80 (required)**
* Has **Phone – string (required)** min 6 /max length **20(required)**
* Has **DirectorateId – int required (foreign key)**
* Has **DirectoratesUnit– DirectoratesUnit**
* Has **Position – string (optional) min 5/ max 50**
* **Has Reguests – A Collection of Request**

**Operator – with Admin Role**

* Has **Id** – a unique Guid, Primary Key
* Has **UserId** – string (required)(foreignKey)
* Has IT**User** – IdentityUser/ApplicationUser
* Has AssignReguests - Collection of Request
* Has **IsActive – bool – default false**

**Request**

* Has **Id** – a unique Guid, Primary Key
* Has **UserId** – string (required)(foreignKey) потребителя подал заявката
* Has **UserMI** – IdentityUser/ApplicationUser
* Has **Description –** string (required) min 5/ max 100
* Has **CategoryId-** **int (required) (foreign key)**
* Has **Category**– Category**(required)**
* Has **StartDate** – **DateTime** with format "**dd/MM/yyyy**" (**required**)
* Has **EndDate** – **DateTime** with format "**dd/MM/yyyy**"
* Has **RequestStateId-** **int (required) (foreign key)**
* Has **RequestState**– AgrreementState**(required)**
* Has **OperatorId**
* Has **Operator Operator**
* Has Comment – string (optional) min 5/ max 50
* **IsActive**

**RequestState**:

* Has **Id –** a unique  **integer, Primary Key**
* **Has Name string** with min length **3** and max length **20(required)**
  + Open
  + Assigned
  + Close
  + Waiting customer

**Category**:

* Has **Id –** a unique  **integer, Primary Key**
* **Has Name string** with min length **5** and max length **100(required)**
  + Компютърна техника
  + Интернет/Интранет страница
  + Нов служител
  + За ФУС
  + Други

**DirectoratesUnit**

* Has **Id –** a unique  **integer, Primary Key**
* **Has Name - string** with min length **3** and max length **120(required)**
  + Правна
  + ФУС
  + ИКОУКС
  + Политически кабинет
  + Инспекторат
  + Други - Извън щата
* Has **IsActive – bool – default = false могат да се изтриват само ако няма потребители**

**Agreement**

* Has **Id –** a unique  **int, Primary Key (по-добре е да е Giud, но за по-бързо го правя с int)**
* Has **GoodName** – **string** with min length **3** and max length **50(required)**
* **Has Description – string (optional)**
* Has **Price – decimal (required) min 20 – max 1000 (required)**
* Has **ReturnPrice – decimal (required) min 20 – max 1000 (required)**
* Has Duration **– integer (required)**
* Has **Ainterest** – **decimal/ not required**
* Has **StartDate** – **DateTime** with format "**dd/MM/yyyy**" (**required**)
* Has **EndDate** – **DateTime** with format "**dd/MM/yyyy**" (**required**)
* Has **UserId** - **string**
* Has **ApplicationUser** Account
* **Has IsDeleted – bool – default = false**
* **Has AgrreementStateId-** **int (required) (foreign key)**
* Has **AgrreementState**– AgrreementState**(required)**
* **Has Collection Interests, type of Interest**

**Interest**

* Has **Id –** a unique  **integer, Primary Key**
* Has **AgreementId – int (required) foreign key (required)**
* **Has Agreement - Agreement**
* Has **ValueInterest – decimal (required) min 1 – max 1000 (required)**
* Has **DateInterest** – **DateTime** with format "**dd/MM/yyyy**" (**required**)
* Has **UserId** - **string**
* Has **ApplicationUser** Account
* **Has IsDeleted – bool – default = false**

**Shop**

* Has **Id –** a unique  **integer, Primary Key**
* Has **Name** – **string**
* **Has Description – string (optional)**
* Has **AgreementId – int (required) foreign key (required)**
* **Has Agreement - Agreement**
* Has **SellPrice – decimal (required) min 1 – max 100000**
* Has **SoldDate** – **DateTime** with format "**dd/MM/yyyy**" optional
* **Has IsDeleted – bool – default = false**

**Connection string default:**

"Server=.\\SQLEXPRESS; atabase=PawnShop;Trusted\_Connection=True;TrustServerCertificate=True;"

## SEED DATA

private void SeedUsers()

{

var hasher = new PasswordHasher<IdentityUser>();

AdminUser = new ApplicationUser()

{

Id = "b97f29e7-1edd-4666-a8d8-8882858d7ccf",

UserName = "admin@abv.bg",

NormalizedUserName = "ADMIN@ABV.BG",

Email = "admin@abv.bg",

NormalizedEmail = "ADMIN@ABV.BG",

FirstName = "Admin",

LastName = "Boss"

};

AdminUser.PasswordHash = hasher.HashPassword(AdminUser, "123456");

UserMsef = new ApplicationUser()

{

Id = "5cf194c7-b26e-42ed-8efd-d98c7980373b",

UserName = "msef@abv.bg",

NormalizedUserName = "MSEF@ABV.BG",

Email = "msef@abv.bg",

NormalizedEmail = "MSEF@ABV.BG",

FirstName = "Mary",

LastName = "Seferov"

};

UserMsef.PasswordHash = hasher.HashPassword(UserMsef, "123456");

UserKsef = new ApplicationUser()

{

Id = "ffae7662-4ff3-4698-8f36-c4e4f392da18",

UserName = "ksef@abv.bg",

NormalizedUserName = "KSEF@ABV.BG",

Email = "ksef@abv.bg",

NormalizedEmail = "KSEF@ABV.BG",

FirstName = "Kalin",

LastName = "Sarafov"

};

UserKsef.PasswordHash = hasher.HashPassword(UserKsef, "123456");

GuestUser = new ApplicationUser()

{

Id = "70e39283-bd42-4d0a-aa2e-46d2a31c4f87",

UserName = "guest.abv.bg",

NormalizedUserName = "GUEST@ABV.BG",

Email = "guest@abv.bg",

NormalizedEmail = "GUEST@ABV.BG",

FirstName = "Guest",

LastName = "Guestov"

};

GuestUser.PasswordHash = hasher.HashPassword(GuestUser, "123456");

}

private void SeedClient()

{

ClientMsef = new Client

{

Id = 1,

PhoneNumber = "1234567890",

Address = "Sofiq,Dianabad",

UserId = UserMsef.Id,

IsDeleted = false

};

ClientKsef = new Client

{

Id = 2,

PhoneNumber = "1234599999",

Address = "Plovdiv,Izgrev",

UserId = UserKsef.Id,

IsDeleted = false

};

AdminClient = new Client

{

Id = 3,

PhoneNumber = "8888899999",

Address = "Varna,Center",

UserId = AdminUser.Id,

IsDeleted = false

};

}

private void SeedState()

{

AwaitingApproval = new AgreementState

{

Id = 1,

Name = "Аwaiting approval"

};

Approved = new AgreementState

{

Id = 2,

Name = "Approved (Active)"

};

Finished = new AgreementState

{

Id = 3,

Name = "Finished"

};

Late = new AgreementState

{

Id = 4,

Name = "Late"

};

ForShop = new AgreementState

{

Id = 5,

Name = "For а Shop"

};

}

private void SeedAgreement()

{

Laptop = new Agreement

{

Id = 1,

GoodName = "лаптоп Acer",

Description = "4GB RAM, 120SSD",

Price = 200,

ReturnPrice = 201,

Duration = 10,

StartDate = new DateTime(2024, 11, 25, 0, 0, 0),

EndDate = new DateTime(2024, 12, 5, 0, 0, 0),

UserId = UserKsef.Id,

AgrreementStateId = 1,

Ainterest = 1

};

TV = new Agreement

{

Id = 2,

GoodName = "SONY TV",

Description ="used - produced 2012",

Price = 100,

ReturnPrice = 103,

Duration = 30,

StartDate = new DateTime(2024, 11, 26, 0, 0, 0),

EndDate = new DateTime(2024, 12, 26, 0, 0, 0),

UserId = UserMsef.Id,

AgrreementStateId = 2,

Ainterest = 3

};

Bike = new Agreement

{

Id = 3,

GoodName = "Bike",

Description = "model Balkan",

Price = 30,

ReturnPrice = 33,

Duration = 30,

StartDate = new DateTime(2024, 11, 27, 0, 0, 0),

EndDate = new DateTime(2024, 12, 27, 0, 0, 0),

UserId = UserMsef.Id,

AgrreementStateId =5

};

}

private void SeedShop()

{

BikeForShop = new Shop

{

Id = 1,

Name = Bike.GoodName,

AgreementId = Bike.Id,

SellPrice = Bike.Price

};

}

private void SeedInterest()

{

TVInterest = new Interest

{

Id = 1,

ValueInterest = TV.Ainterest,

DateInterest = DateTime.Now,

AgreementId = TV.Id,

UserId = TV.UserId,

};