

**Курсов проект по ППС#**

Тема: Туристическа Агенция

Изготвен от

Илия Веселинов Чакъров, СИ, 1 курс магистър

Ф.н.: 24651225

1. Задание на проекта, описание

**Изисквания:**

Таблица Екскурзии

* Каталожен номер на екскурзията– числово поле, поле ключ;
* Дестинация /държава/ – текстово поле;
* Дестинация /град/– текстово поле;
* Период /начална дата/ – тип дата;
* Период /крайна дата/ – тип дата;
* Цена /транспорт/;
* Цена на екскурзия /стая+храна/;
* Цена за пакет от допълнителни опции /посещение на: музей, театър, опера/

Таблица Клиенти

* Клиентски номер – числово поле, поле ключ;
* ЕГН – числово поле;
* Три имена – текстово поле;
* Държава – текстовото поле;
* Пол –текстово поле;
* Дата на раждане – тип дата;
* Застраховка номер – тип дата;

Таблица Поръчки

* Номер на поръчка –числово поле, поле ключ;
* Каталожен номер на екскурзията – числово поле;
* Клиентски номер – числово поле;
* Дата на изготвеният документ – тип дата.

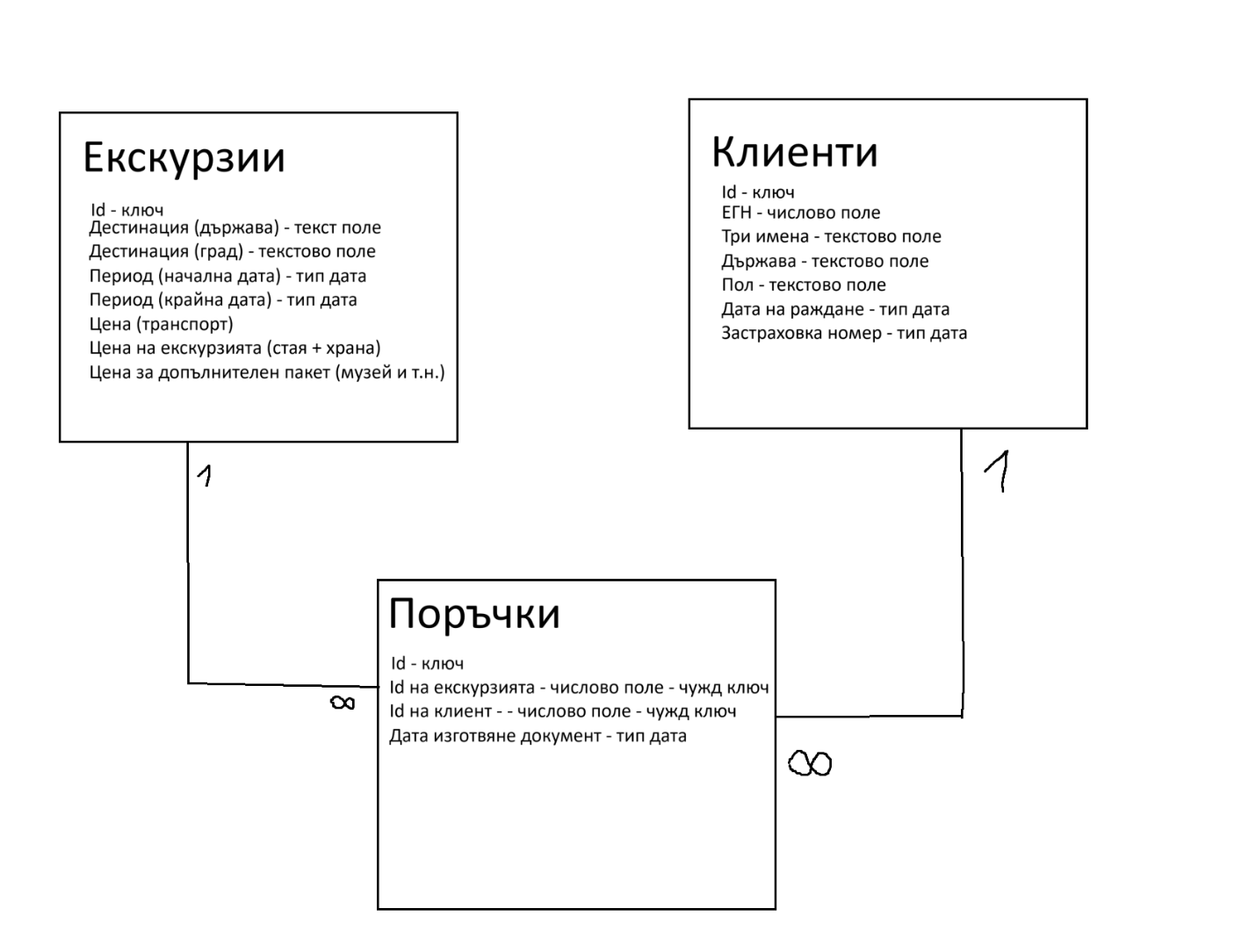
**Форми:** Да се създадат форми за попълване на данни в съответните таблици (въвеждане, актуализиране и изтриване на данни);

**Изисквания за справки:**

* да се изведе списък с всички клиенти по каталожен номер, три имена и ЕГН, по дестинация за период зададен чрез параметри - начална и крайна дата;
* да се изведе списък с дестинациите посетени от конкретен клиент, цената която е заплатил за всяка услуга, номера на застраховката му
* да се изчертае графика Дестинации - Клиенти, която визуализира, най-често посещаваните дестинации според държавата от която е клиента
* извеждане на всички поръчки по зададени:
  + ЕГН на клиент;
  + застраховка;
  + по датата на издаване на поръчката.

1. Анотация

Зададената структура на проекта е както следва на рисунката по-долу. Съдържа 3 таблици – Екскурзии, Клиенти и Поръчки. Един клиент може да има множество поръчки и много поръчки могат да бъдат направени от един клиент. Следователно те ще бъдат свързани с връзка от тип едно-към-много. Същото важи и за екскурзиите. По-долу са посочени връзките на таблиците и са описани полетата им. Има числови, текстови и полета тип дата.



В общи линии крайната цел на проекта е той да бъде система за резервиране и заплащане на екскурзии. Има набор от екскурзии с посочени дати, клиент се регистрира и след като си избере дадена екскурзия поръчката бива направена.

Това приложение най-вероятно ще бъде за вътрешно ползване – тоест клиентът няма да може сам да разглежда дестинациите, те са предоставени в това приложение по-скоро за персонала на екскурзионната компания и по изисквания на клиента те добавят поръчка с нужните екстри. Тази система служи за улеснение на персонала, за да може да бъдат филтрирани, намирани, добавяни, изчислявани поръчките.

1. Увод

Актуалността на такъв тип приложение е относителна. Тя зависи от различни видове критерии. Най-вече от използваната технология. Избрана е технологията WinForms със C# като достатъчна за изпълнение на всички изисквания. Нейната актуалност е оценена по следните критерии:

* 1. **Изисквания за модернен UI/UX -** WinForms не е подходяща за модерни, уеб-базирани или мобилно насочени приложения. Ако клиентът или потребителите очакват съвременен интерфейс, други технологии като WPF, Blazor или уеб-базирани решения (например ASP.NET Core) са по-подходящи
  2. **Приложението е за вътрешна употреба -** Ако приложението ще се използва в организация, където основният приоритет е простотата и бързата разработка, WinForms все още е подходящ избор. Това включва и първия критерий също. Ако е за вътрешна употреба, тогава изискванията към потреибтелския интерфейс са по-ниски
  3. **Бързо изпълнение с налични технологии** - Ако разработчиците вече имат опит с WinForms и няма изискване за дългосрочна поддръжка или сложна интеграция, приложението може да бъде реализирано бързо и с минимални разходи
  4. **Целта е обучение -** Ако проектът е предназначен за учебни цели, за демонстрация на умения в проектирането на бази данни, обработка на данни, работа с Windows Forms и връзка с база данни чрез Entity Framework, това е отличен пример

Използването на C# WinForms за този проект има следните **недостаъци**:

* Ограничена поддръжка за модерни интерфейси и адаптивност.
* По-малко подходящо за мащабируеми решения.
* Потенциални затруднения при интеграция с облачни или мобилни услуги.

Но въпреки това е избрана тази технология. Целта въпреки недостатъците да се продължи с тази технология е, че **предимствата** накланят везните драстично в своя полза:

* Лесна интеграция с база данни.
* Подходящо за бърза реализация:
  + WinForms предлага **визуален дизайнер** в Visual Studio, който позволява създаването на форми и интерфейси с лесно поставяне на контроли.
  + Контролите като **DataGridView**, **ComboBox**, **TextBox** и **Button** са лесни за конфигуриране и използване.
* Добра поддръжка на функционалности като DataGridView, Chart контрол и обработка на събития.
* Подходящо за малки и средни приложения:
  + WinForms е идеално за проекти, които не изискват сложна архитектура или разпределена обработка.
  + Подходящо за вътрешни приложения в организации, където мащабируемостта и модулността не са основни изисквания.
* Ниска зависимост от външни ресурси
  + WinForms приложенията обикновено работят **локално** и не зависят от интернет връзка, което ги прави стабилни за автономна употреба.
  + Липса на сложни конфигурации за хостинг или разполагане.

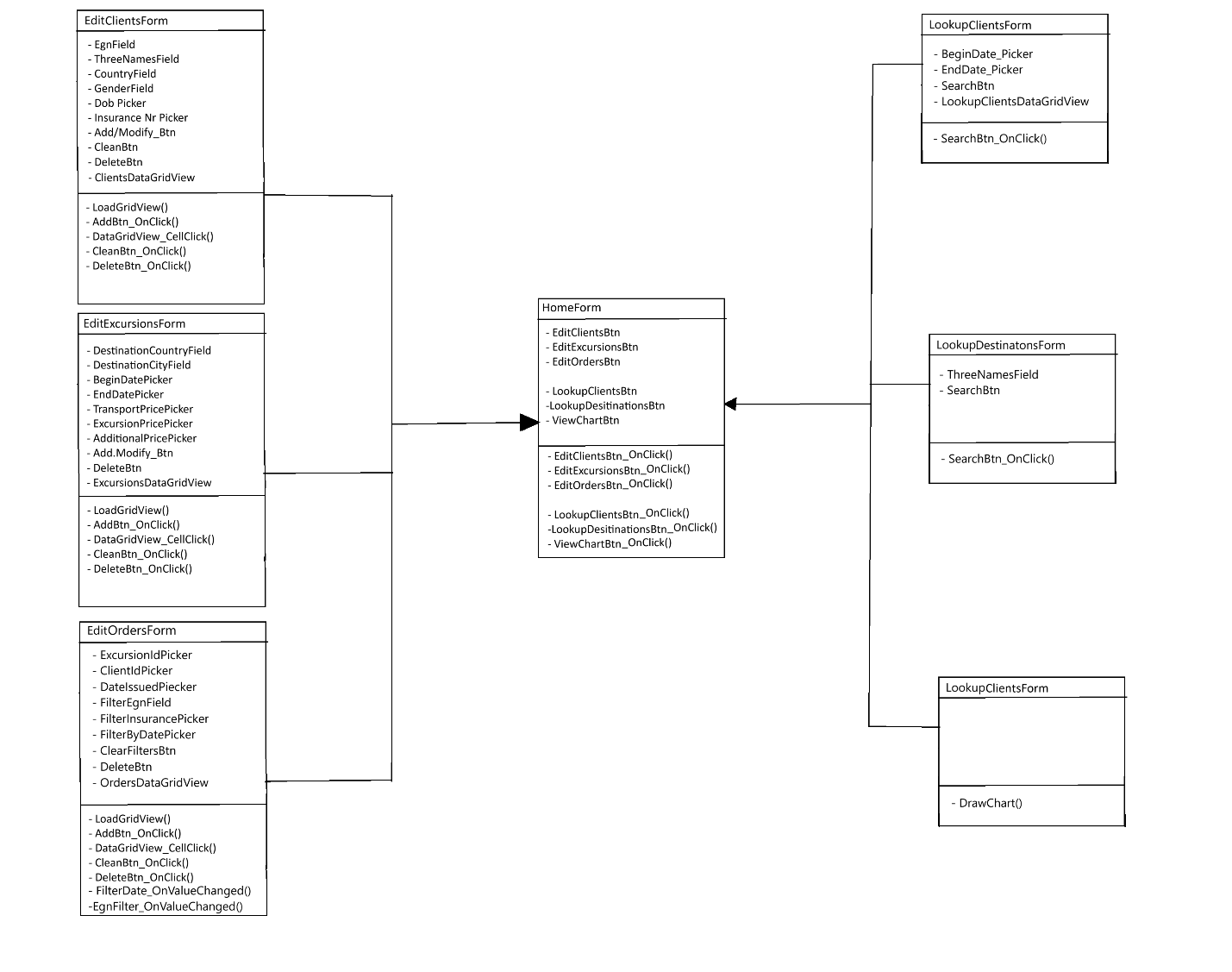
1. Обзор на съществуващите решения. Изводи. Цел и задачи.
   1. **Обзор на съществуващите решения**: Съществуват множество решения за управление на данни за екскурзии, клиенти и поръчки, като най-често срещаните включват уеб-базирани приложения, мобилни приложения и настолни системи. Много от тях са разработени с помощта на съвременни технологии като **ASP.NET Core**, **Blazor**, **React**, докато настолните решения често използват **WinForms** или **JavaFX**. (Източник StackOverflow: https://stackoverflow.com/beta/discussions/78018971/winforms-vs-electron-vs-javafx-for-tray-application)
   2. **Изводи**: Настолните приложения, разработени с **WinForms**, са подходящи за специфични нужди като локално управление на данни в малки организации или за учебни и прототипни проекти. Те се отличават с лесно внедряване, ниски разходи и бързина на разработката. Въпреки това, ограниченията в дизайна и липсата на уеб функционалности ги правят по-малко конкурентоспособни в сравнение с модерните уеб решения.
   3. **Цел**: Разработване на настолно приложение за управление на екскурзии, клиенти и поръчки, което да предостави лесен и удобен интерфейс за въвеждане, актуализиране и справка на данни, като същевременно отговори на нуждите на малки и средни организации или учебни цели.
   4. **Задачи**:

* Проектиране на база данни, която да съдържа всички необходими таблици за екскурзии, клиенти и поръчки.
* Създаване на форми за:
  + Въвеждане на данни.
  + Актуализиране и изтриване на съществуващи записи.
* Разработване на справки, включително:
  + Списък на клиенти по дестинация и период.
  + Дестинации, посетени от конкретен клиент, и свързаните разходи.
  + Графика за популярността на дестинациите по държави.
* Осигуряване на удобен и интуитивен потребителски интерфейс с помощта на WinForms.
* Използване на SQL заявки или LINQ за извличане на данни и генериране на отчети.

С реализацията на тези задачи ще бъде постигната целта за разработване на функционално и практично приложение за управление на данни в контекста на екскурзии и свързаните с тях услуги.

1. Проектиране и описание на предлаганото решение
   1. **Изисквания към програмната система**
      1. **Функционални изисквания:**

* Въвеждане, редактиране и изтриване на данни за:
  + Екскурзии (дестинации, цени, периоди и опции).
  + Клиенти (лична информация, застраховки).
  + Поръчки (свързване на екскурзии и клиенти).
* Генериране на справки:
  + Списък на клиенти по зададена дестинация и период.
  + Дестинации, посетени от даден клиент, с разходи и застраховка.
  + Графика на най-често посещаваните дестинации.
  + Списък на поръчки по зададени ЕГН, застраховка или дата.
* Автоматично извършване на формален и логически контрол на въведените данни.
  + 1. **Нефункционални изисквания:**
* Потребителски интерфейс, който да бъде интуитивен и лесен за използване.
* Поддръжка на минимални системни изисквания за работа на настолен компютър.
* Локална база данни (MSSQL), без необходимост от интернет връзка.
* Възможност за лесно архивиране и възстановяване на данни.
  1. **Логически модел на програмната система**

****

* 1. **Архитектура на системата**

Системата следва трислойна архитектура:

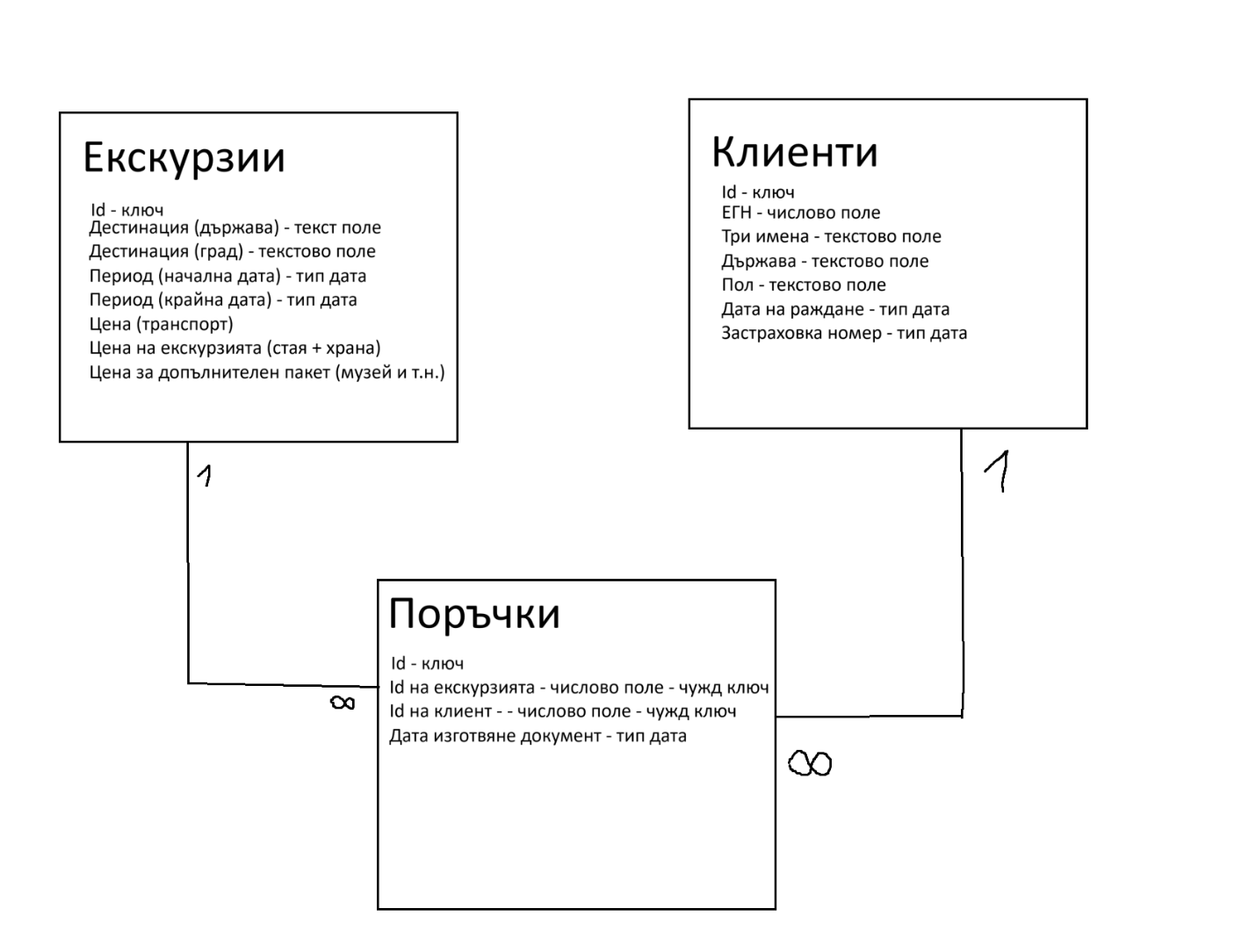
* Презентационен слой:
  + Потребителският интерфейс (форми и контроли в WinForms).
* Логически слой:
  + Модули за валидиране, обработка и бизнес логика.
* Слой за данни:
  + Взаимодействие с базата данни чрез Entity Framework, Repository Pattern и Unit Of Work Pattern
  1. **Организация на данните, концептуален модел на БД:**
  2. **Избор на език и среда за програмиране**
* Език за програмиране: C#.
* Среда за разработка: Visual Studio.
* Причини за избора:
  + Лесна интеграция с WinForms за настолни приложения.
  + Добра поддръжка за работа с бази данни - Entity Framework.
  + Интегрирани инструменти за визуално проектиране.
  1. **Реализация на програмната система**
     1. **Структура на данните**

Таблица Екскурзии

* Каталожен номер на екскурзията– числово поле, поле ключ;
* Дестинация /държава/ – текстово поле;
* Дестинация /град/– текстово поле;
* Период /начална дата/ – тип дата;
* Период /крайна дата/ – тип дата;
* Цена /транспорт/;
* Цена на екскурзия /стая+храна/;
* Цена за пакет от допълнителни опции /посещение на: музей, театър, опера/

Таблица Клиенти

* Клиентски номер – числово поле, поле ключ;
* ЕГН – числово поле;
* Три имена – текстово поле;
* Държава – текстовото поле;
* Пол –текстово поле;
* Дата на раждане – тип дата;
* Застраховка номер – тип дата;

Таблица Поръчки

* Номер на поръчка –числово поле, поле ключ;
* Каталожен номер на екскурзията – числово поле;
* Клиентски номер – числово поле;
* Дата на изготвеният документ – тип дата.
  + 1. **Описание на програмните модули**

От заглавната форма може да се стигне до всички форми чрез бутони:

private void open\_orders\_btn\_Click(object sender, EventArgs e)

{

var ordersForm = \_serviceProvider.GetService<OrdersForm>();

this.Hide();

ordersForm.ShowDialog();

this.Show();

}

private void open\_excursions\_btn\_Click(object sender, EventArgs e)

{

var excursionsForm = \_serviceProvider.GetService<ExcursionsForm>();

this.Hide();

excursionsForm.ShowDialog();

this.Show();

}

Във всяка форма се съдържа OnClick метод, който обработва данните:

private async void button\_excursionsDelete\_Click(object sender, EventArgs e)

{

if (selectedExcursionId == null) return;

var confirmResult = MessageBox.Show(

"Are you sure you want to delete this excursion?",

"Confirm Delete",

MessageBoxButtons.YesNo,

MessageBoxIcon.Warning);

if (confirmResult == DialogResult.No) return;

var toDelete = await \_unitOfWork.Excursions.GetAsync(e => e.ExcursionId == selectedExcursionId);

\_unitOfWork.Excursions.Delete(toDelete);

await \_unitOfWork.SaveAsync();

CleanFields();

LoadGridView();

}

private void CleanFields()

{

textBox\_excursionDestinationCountry.Text = "";

textBox\_excursionDestinationCity.Text = "";

dateTimePicker\_excursionBeginDate.Value = DateTime.Today;

dateTimePicker\_\_excursionEndDate.Value = DateTime.Today;

numericUpDown\_excursionPriceTransport.Value = 0m;

numericUpDown\_excursionPriceExcursion.Value = 0m;

numericUpDown\_excursionAdditionalPrice.Value = 0m;

selectedExcursionId = null;

}

* + 1. **Структура и организация на потребителския интерфейс**
* Основно меню с достъп до модулите за екскурзии, клиенти и поръчки.
* Формуляри с TextBox и DataGridView за въвеждане и преглед на данни.
  1. **Формат на входните документи**
* Входни полета с валидиране (например контрол на дата, допустими стойности за цени).
* Логически контрол (например стартовата дата да е преди крайната).
  1. **Формат на извежданите справки**
* Табличен вид в DataGridView.
* Графики за визуализация.
  1. **Инструкции и изисквания за работа с програмната система:**
     1. **Ръководство на потребителя**

Програмата има просто използване. Бутоните правят описаното

* При въвеждане на данни с избиране на бутон за добавяне, се добавя запис.
* Таблицата с информация позволява избиране на ред, след което избрания ред може да се редактира или изтрива.
  + 1. **Инструкции за инсталиране**
* Инсталиране на изпълнимия файл и необходимите библиотеки.
* Изисквания за .NET Framework или .NET Runtime.
  + 1. **Инструкции за поддържане**
* Създаване на резервни копия на базата данни
  + 1. **Изисквания към апаратното осигуряване**
* Минимални изисквания:
  + Операционна система: Windows 10 или по-нова.
  + RAM: 4 GB.
  + Свободно дисково пространство: 500 MB

1. **Използвана литература**

StackOverflow: <https://stackoverflow.com/questions/69841320/uml-diagram-in-winforms>

WinForms Documentation: <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/?view=netdesktop-9.0>

Помагала от бакалавърска степен по Софтуерни и интернет технологии по предмети: Проектиране на потребителски интерфейси, Софтуерни изисквания и спецификации, Технологии на софтуерното производство