

# Technická dokumentace

## Přehled

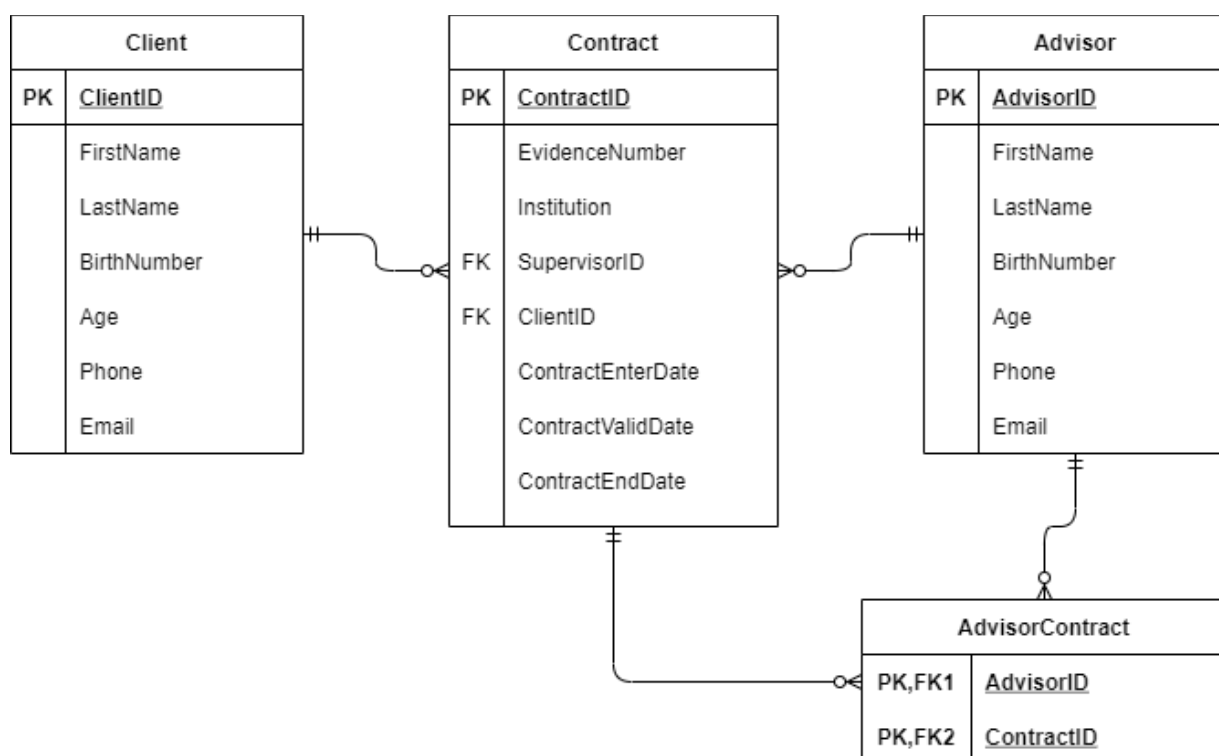
Cílem práce bylo vytvořit aplikaci pro správu smluv. V systému vystupují entity, jako jsou klienti, poradci a smlouvy, přičemž mají tyto entity určité vlastnosti a omezení. Aplikace má být vytvořena jako klasická aplikace s využitím rámce ASP.NET Core, návrhového vzoru Model-View-Controller a značkovacího jazyka Razor, který umožňuje vkládat kód serverové části do webových stránek.

## Použité technologie

Aplikace byla implementována ve vývojovém prostředí Visual Studio 2019 s použitím webového rámce ASP.NET Core verze 5.0. Jako databázový server byl zvolen MS SQL Server Express, přičemž mapování tabulek databáze a jejich záznamů na instance tříd reprezentující entity tohoto systému zajišťuje **Entity Framework Core**, rámec pro objektově relační mapování.

## Datová vrstva (Model)

V systému je zapotřebí pracovat s entitami, kterými jsou klienti (**Client**), poradci (**Advisor**) a smlouvy (**Contract**). Pro jednotlivé entity jsou zadány vlastnosti a omezení, které musí splňovat. U každé smlouvy musí být uvedeno evidenční číslo, název instituce, klient, správce smlouvy, datum uzavření, datum platnosti a datum ukončení. Každá smlouva může víc více účastníků (poradců), přičemž jeden účastník je povinný a ten je zároveň správcem smlouvy. Pro zajištění tohoto vztahu **many-to-many**, byla vytvořena vazební entita **AdvisorContract**, která reprezentuje vztah mezi entitami poradce a kontrakt, kde mohou být poradci dalšími účastníky smlouvy. Dále je u každého klienta a poradce nutné evidovat jméno, příjmení, e-mail, telefon, rodné číslo a věk. Klienti a poradci dědí vlastnosti z entity **UserBase**, která obsahuje definici jejich společných vlastností. Pro reprezentaci těchto entit a jejich vzájemných vztahů byl vytvořen entity-relationship diagram, který je zobrazen na následujícím obrázku. Tyto entity mají své implementace ve složce **/Models**.



## Aplikační vrstva (Controller)

Pro práci s modelem (stavem aplikace, který je dán jeho daty) a generování adekvátní pohledů (views) klientům jako odpovědi na jejich požadavky (requests), slouží řadiče (controllers). Řadiče vytváří odpovědi na klientské požadavky, nemusí se jednat jen o zasílání pohledů, mohou to být i přesměrování na jiné řadiče atp.

V tomto řešení jsou implementovány řadiče **AdvisorsController**, **ClientsController**, **ContractsController** a **HomeController**. O zasílání pohledu na domovskou stránku se stará HomeController. Ostatní řadiče implementují akce (jedná se o metody), které zpracovávají uživatelské požadavky, jako jsou editace, exportování dat do CSV, vytváření nových modelů, zasílání pohledů konkrétních modelů na základě filtrovacích pravidel a starají se o výběr a zasílání příslušných pohledů. Tyto řadiče jsou implementovány ve složce **/Controllers**.

## Prezentační vrstva (View)

V podsložkách složky **/Views** se nachází soubory .cshtml, které obsahují definice jednotlivých pohledů v značkovacím jazyce **Razor**. Kontrolery se starají o výběr těchto pohledů, jejich zpracování a zaslání klientům. Responzivní design webových stránek je zajištěn CSS knihovnou Bootstrap verze 4.3.1., která je použita pro definování stylů HTML elementů, které jsou součástí syntaxe Razor.

## Dodatečné informace

Pro vytvoření databáze je zapotřebí ve vývojovém prostředí Visual Studio v konzoli pro správu balíčků zadat příkaz **update-database**. Po spuštění aplikace je vytvořen databázový kontext a metoda Initialize() třídy DbInitializer zjistí, zda databáze obsahuje nějaká data, pokud ne, tak je databáze naplněna ukázkovými daty.