



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Aplikace pro přehlednou správu inzerátů nabídek pracovních pozic
Student:	Hynek Bulíř
Vedoucí:	Ing. Pavel Štěpán
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce zimního semestru 2019/20

Pokyny pro vypracování

Cílem práce je aplikace pro přehlednou správu inzerátů nabídek pracovních pozic, kde bude možno přehledně pracovat s nabídkami a exportovat výsledky s možností filtrace. Aplikace bude vyvinuta pro operační systém Windows, na platformě .NET. Součástí práce bude grafický design aplikace a uživatelský návod k použití.

Postupujte dle následujících kroků:

1. Analyzujte požadavky cílové skupiny uživatelů.
2. Navrhněte aplikaci vyhovující požadavkům.
3. Implementujte aplikaci a řádně ji otestujte .
4. Zhodnoťte výsledek a diskutujte další možná vylepšení.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 28. února 2018



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

Bakalářská práce

Aplikace pro přehlednou správu inzerátů nabídek pracovních pozic

Hynek Bulíř

Katedra softwarového inženýrství
Vedoucí práce: Ing. Pavel Štěpán

10. ledna 2019

Poděkování

Chtěl bych poděkovat svému vedoucímu bakalářské práce Ing. Pavlu Štěpánovi za velice vstřícný a trpělivý přístup. Také bych rád poděkoval své mamince, že mě podporovala i v nelehkých chvílích při studiu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 10. ledna 2019

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2019 Hynek Bulíř. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Bulíř, Hynek. *Aplikace pro přehlednou správu inzerátů nabídek pracovních pozic*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.

Abstrakt

Tato bakalářská práce spočívá ve vytvoření aplikace pro správu pracovních nabídek. Aplikace je určena především pro personální oddělení těch firem, které se starají o nábor svých zaměstnanců samostatně, tedy bez spolupráce s personální agenturou. Cílem aplikace je pomoci udržet pořádek a přehled ve vlastních nabídkách firmy na inzertních webových stránkách. Efektivní zachování pořádku v evidenci nabídek je možné především díky intuitivnímu ovládání aplikace. K řešení práce jsou použity technologie ASP.NET, Javascript, Entity Framework, aplikace je tedy určena pro platformu Windows s .NET knihovnou verzí 4.6. Aplikace nahrazuje papírovou evidenci a zjednodušuje tak výrazně manipulaci s nabídkami.

Klíčová slova ASP.NET MVC, JavaScript, informační systém, lidské zdroje (HR), personalistika, pracovní nabídky, webová aplikace, .NET

Abstract

This bachelor thesis deals with creation of a web application for easier management of job announcements. The application aims to Human Resources departments of those companies that recruit new employees independently, i. e. without cooperation with any personal agency. Purpose of the application is to help keep transparency and order in their own job offers which are being advertised at different websites. Application's intuitive control enables effectively keeping order in job offers. Technologies used in this application are ASP.NET, JavaScript, Entity Framework. The application is designed for Windows platform with .NET 4.6 version. The application replaces paper records of job announcements and makes their management significantly easier.

Keywords ASP.NET MVC, JavaScript, information system, human resources (HR), job announcements, web application, .NET

Obsah

Úvod	1
1 Cíl práce	3
2 Analýza problému	5
2.1 Rešerše existujících řešení	5
2.2 Analýza požadavků	6
3 Návrh aplikace a zvolené technologie	9
3.1 MVC	9
3.2 ASP.NET MVC	9
3.3 C#	10
3.4 HTML	11
3.5 Razor	11
3.6 CSS	11
3.7 JavaScript	12
3.8 Knockoutjs	12
3.9 AJAX	13
3.10 Bootstrap	13
3.11 Entity framework	14
4 Implementace aplikace	15
4.1 Vývoj	15
4.2 Ukázky kódu	16
4.3 Uživatelské rozhraní	17
5 Testování a výsledky	19

5.1 Průběžné testování během vývoje	21
5.2 Uživatelské testování	21
Závěr	23
Literatura	25
A Uživatelský návod k použití	27
B Seznam použitých zkratk	31
C Obsah přiloženého CD	33

Seznam obrázků

3.1	MVC Diagram	10
3.2	MVVM Diagram + jazyky jednotlivých vrstev	13
3.3	EF - Code First přístup	14
4.1	Notifikace úspěšného uložení	17
4.2	Vyskakovací modal	18
A.1	Seznam pracovních nabídek	28
A.2	Přidání nového záznamu	29
A.3	Notifikace uložení záznamu	29
A.4	Notifikace smazání záznamu	29

Úvod

Každá větší firma má personální oddělení pro nábor nových zaměstnanců a toto oddělení má na starosti inzerci nabídek volných pozic. Nabídky je však nutné nějakým způsobem sledovat a spravovat, ideálně získávat data o jednotlivých nabídkách na různých inzerentních webech a také o jejich úspěšnosti. Oddělení pak má přehled, na kterém webu je inzerce prodělečná a na kterém naopak, a může tak přizpůsobit svoji strategii, aby dosáhla co největších zisků za co nejmenší vynaložené náklady. Udržovat tyto informace v „papírové podobě“ je v dnešní době zbytečné, jakékoli vyhledávání a udržování statistik by bylo příliš složité.

S využitím programů pro zpracování těchto dat, jako například Microsoft Excel, LibreOffice Calc, by to již mohlo být snazší, ale stále nejsou uzpůsobeny přímo pro toto použití. Má aplikace se bude touto problematikou zabývat, a ušetří tak spousty zbytečné práce pracovníkům personálního oddělení. Toto téma jsem si zvolil na základě vypořádání nedostatků ve firmě, kde pracuji, zároveň předpokládám, že tato firma nebude jediná, co se potýká s takovouto problematikou.

Motivace

Za dobu mého působení ve firmě Commerz Systems GmbH¹, od roku 2015, jsem se dostal k mnoha příležitostem, kdy jsem byl členem vývojářského týmu určité aplikace. Jednou z těchto příležitostí bylo vytvořit vlastní aplikaci pro personální oddělení, která by měla za úkol spravovat nábor nových zaměstnanců do firmy. Návrh samotné aplikace personální oddělení značně ocenilo, protože by jim velmi ulehčil správu nabídek, která byla do té doby

¹Dnes již sjednoceno s Commerzbank AG

vedena pouze v programu Microsoft Excel.

Nasbírali jsme tedy potřebné informace k vytvoření této aplikace a provedli analýzu požadavků pracovníků tohoto oddělení. Ačkoliv by aplikace byla personalizovaná a šitá tedy na míru, firma se nakonec rozhodla objednat jinou, již hotovou aplikaci, u jiného poskytovatele. Důvodem byly pravděpodobně nižší pořizovací náklady, ačkoliv o tvorbu naší aplikace se měl starat pouze tříčlenný tým studentů. Nicméně mě dané téma oslovilo natolik, že jsem se jím chtěl dále zabývat ve své bakalářské práci, jelikož i v budoucnu bych se rád zabýval tvorbou webových aplikací. Rozhodl jsem se proto aplikaci samostatně vytvořit, rozšířit si tak své obzory a vyzkoušet, co tvorba webové aplikace obnáší od úplného začátku.

Cíl práce

Cílem této práce je vytvoření aplikace, která nám v první řadě umožní jednoduše udržovat informace jako je: popis pozice, datum expirace u jednotlivých pracovních nabídek vložených na inzertní weby, konkrétně jobs.cz, práce.cz a další. Dále bude udržovat statistiky o úspěšnosti inzerátů na jednotlivých inzertních webech, díky tomu bude možné upřednostňovat webové stránky s vyšší úspěšností před těmi ostatními pro vyšší efektivitu. Personální oddělení tak jednoduše získá přehled, na které stránky se soustředit, co dále zlepšit, atp.

Samotné implementaci aplikace předchází analýza, která zjistí, co přesně vyhovuje pracovníkům, aby aplikace byla udělaná na míru pro potřeby personálního oddělení. Následuje popis a výběr technologií, užitých při vývoji aplikace. Na závěr pak otestování a shrnutí výsledku.

Analýza problému

Personální oddělení menší až střední firmy potřebuje udělat nábor nových kandidátů na otevřené pracovní pozice a zároveň si nějakým způsobem udržovat přehled o již nabídnutých pozicích. Zároveň existuje mnoho způsobů, kam nabídku inzerovat, ať už papírovou formou, například na nástěnce ve škole, nebo elektronickou formou, (kterou se zde budeme hlavně zabývat), na inzerentních portálech, jako jsou jobs.cz², prace.cz³ a mnoho dalších. K jednotlivým nabídkám je dobré si udržovat i drobnou statistiku úspěšnosti, aby se tak firma vyhnula inzerování na místa, kde je úspěšnost příliš malá. To ušetří firmě peníze a personálnímu oddělení práci a čas.

2.1 Rešerše existujících řešení

2.1.1 Papírová forma

Za asi nejhorší možné řešení považuji psaní evidence nabízených inzerátů na papír. Ve firmě, která je opravdu malá, by toto řešení pravděpodobně obstálo, ale u firmy nabízející současně několik pracovních pozic by bylo toto řešení opravdu složité na údržbu a zároveň by bylo velice nepřehledné.

2.1.2 Kancelářský software

V Kancelářských balících (Microsoft Office, LibreOffice a další) se již dá práce značně usnadnit a zajistit lepší přehlednost. Sdílení těchto záznamů je také podstatně snazší, díky elektronické poště, nebo ukládáním záznamů na místa přístupná v síti. Avšak stále u těchto nástrojů chybí možnost

²Dostupné na: <https://www.jobs.cz>

³Dostupné na: <https://www.prace.cz>

přístupovat do záznamů více uživatelí najednou. Tuto možnost nám dovolí například online nástroj Google Sheets ⁴, který obsahuje více, pro tento účel, užitečných funkcí, jako chat mezi uživateli, komentáře, atd.

2.1.3 Specializovaná webová aplikace

Webová aplikace vytvořena na míru potřebám personálního oddělení přináší mnoho výhod. Je přístupná odkudkoliv, je přehledná tím, že má jasně danou strukturu záznamů. Nestane se tak, že někdo špatným zásahem poškodí formátování záznamů. Je možno ji stále vylepšovat a upravovat podle potřeb uživatelů.

2.2 Analýza požadavků

Vycházel jsem zde ze zkušeností ze svého zaměstnání ve firmě Commerz Systems GmbH, kde jsem působil v týmu programátorů, který měl za úkol vytvořit podobnou aplikaci pro personální oddělení, ale bohužel k vývoji této aplikace nakonec nikdy nedošlo.

Požadavky byly velice jednoznačné a to ukládání záznamů jednotlivých nabídek s popiskem, kde jednotlivé záznamy obsahují datum od kdy a do kdy jsou nabídky inzerovány, dále na jakém portálu jsou umístěny a mít možnost u nich uchovávat informace, kolik lidí na nabídku zareagovalo, kolik jich bylo pozváno k pohovoru a kolik jich ve výsledku bylo zaměstnáno. Díky tomu se personální oddělení bude moci soustředit spíše na internetové stránky, na kterých mají větší úspěšnost s náborem pracovníků. V neposlední řadě, ač to může znít legračně, je důležitý vzhled aplikace. V dnešní době si každý oblíbí raději aplikaci s efektními animacemi a barevným designem, než obyčejnou formulářovou aplikaci se strohým vzhledem. Nesmí to být na úkor přehlednosti aplikace, ale měla by mít poutavý vzhled.

Aplikace by měla být lokalizovaná pro konkrétní zemi a aby bylo zamezeno pozdějšímu složitějšímu refaktorování a dohledávání zobrazujících se výrazů v daném jazyce, budou soustředěny do jednoho lokalizačního souboru, který stačí pouze vyměnit za jiný, a celá aplikace bude v jiném jazyce. Toto snažení je z důvodu případného budoucího rozšiřování, aby bylo pro vývojáře co nejsnazší. Díky soustředění výrazů do jednoho souboru[1], znamená překlad kompletní aplikace jen několik desítek minut práce.

A v poslední řadě k aplikaci se bude přístupovat přes tzv. tenkého klienta, kterého bude představovat webový prohlížeč. Tím se zjednoduší

⁴Dostupné na: <https://www.google.com/intl/cs/sheets/about/>

přenositelnost a nebude potřebná jakákoliv instalace na straně klienta, jelikož aplikace poběží na jednom serveru.

2.2.1 Funkční a nefunkční požadavky

Tato sekce se věnuje funkčním a nefunkčním požadavkům a popisuje, jak by se měl výsledný produkt chovat a jaké procesy užití má splňovat.

Funkční požadavky jsou:

- Vytvoření pracovní nabídky
- Úprava nabídky
- Smazání nabídky
- Filtrování
- Export filtrovaných nabídek
- Jazyková lokalizace

Aplikace musí zaznamenávat a ukládat následující:

- Název nabídky
- Popis
- Portál, na kterém se nachází
- Datum platnosti od/do
- Počet zájemců
- Počet zájemců na pohovoru
- Počet zaměstnaných zájemců
- Datum vytvoření a poslední úpravy nabídky

Nefunkční požadavky, tedy omezující požadavky, které musí systém, na kterém aplikace poběží, splňovat:

- Webová aplikace
- Tenký klient
- .NET verze 4.6.1

2. ANALÝZA PROBLÉMU

- Knockoutjs verze 3.4.2
- Bootstrap verze 3.4.0
- jQuery verze 3.3.1

Návrh aplikace a zvolené technologie

Tato kapitola popisuje, které technologie byly zvoleny k vypracování této práce.

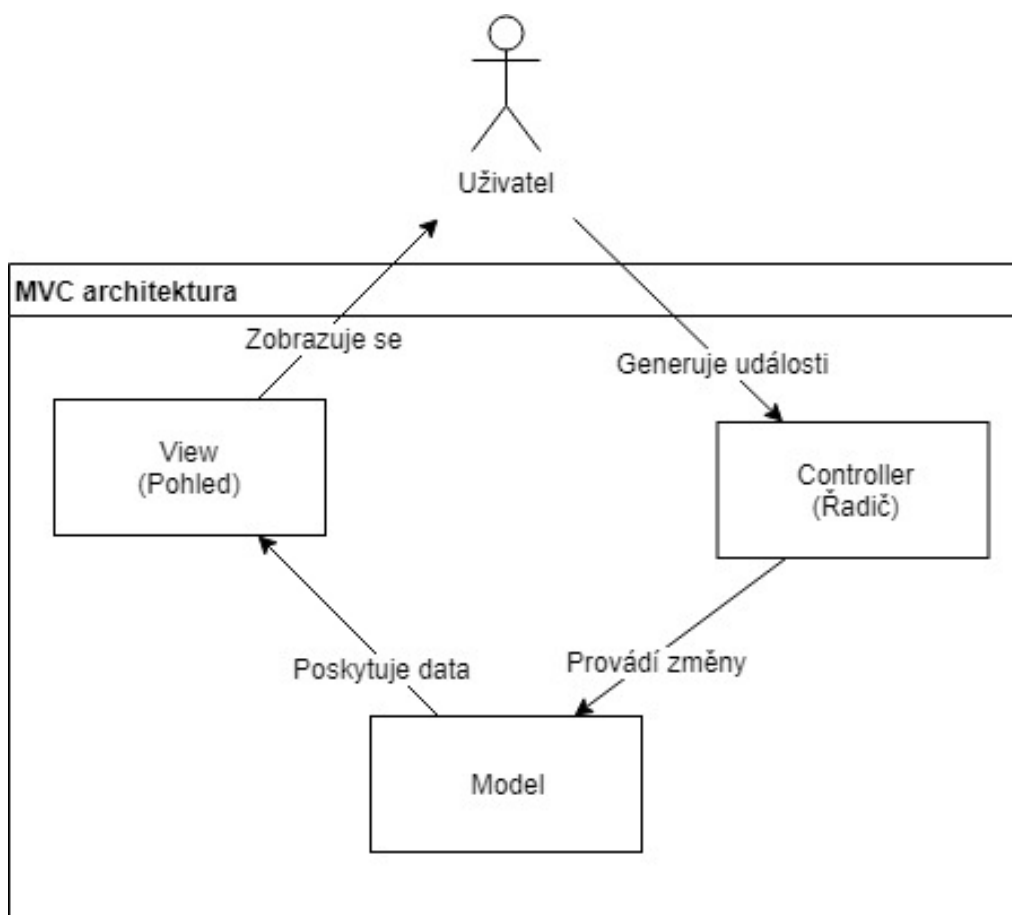
3.1 MVC

Byla zvolena architektura MVC, která je dnes běžná pro tvorbu uživatelského rozhraní. Rozděluje aplikaci do tří částí a odděluje tak interní data od toho, co uživatel ve výsledku vidí. Jednotlivé komponenty jsou:

- Model - který reprezentuje vnitřní data, přímo s nimi nakládá a stará se o jejich uložení do databáze
- View(Pohled) - zajišťuje prezentaci dat uživateli, data obdrží od modelu
- Controller(Řadič) - „naslouchá“ událostem od uživatele a zpracovává je dál pro model

3.2 ASP.NET MVC

ASP.NET MVC je nástroj pro tvorbu webových aplikací. Je součástí platformy .NET, který běží na straně serveru a je určen pro tvorbu dynamických webových aplikací a služeb. Jedná se o sadu knihoven, díky kterým se webová aplikace vyvíjí v jazyce C#.



Obrázek 3.1: MVC Diagram

3.3 C#

C# [2] je objektově orientovaný programovací jazyk vyvinutý firmou Microsoft, umožňující vývoj aplikací běžících na platformě .NET. Je založen na jazycích C, C++ a Java a může být využit k tvorbě webových aplikací, servisních služeb, formulářových aplikací a mnoha dalších. Jazyk C# spadá do kategorie programovacích jazyků s virtuálním strojem, jako například Java. Znamená to, že je napsaný kód nejdříve přeložen do tzv. *mezikódu*, neboli CIL, a tento kód je dále interpretovatelný virtuálním strojem, u .NET je to CLR a výsledkem je již strojový kód. Tento proces vypadá na první pohled složitě, ale ve skutečnosti přináší mnoho výhod. Umožňuje například použití více jazyků, jako C++, Visual Basic, C#, F# a tyto jazyky jsou následně přeloženy do CIL. Dále má výhodu zabezpečení. Aplikace se dále může šířit jen jako kód v CIL, který není snadno čitelný. Virtuální stroj

CLR, na kterém běží tyto CIL kódy, pak dokáže sám spravovat paměť a dokáže rozpoznat nebezpečný kód před jeho vykonáním a neprovést ho.

3.4 HTML

HTML[3] je základní stavební kámen téměř každé webové stránky. Jedná se o značkovací jazyk, protože používá značky k vytváření a formátování obsahu webových stránek. Tento kód je následně zobrazen webovým prohlížečem. Dnes se jazyk HTML používá spíše pouze k definování struktury webové stránky. Dříve se používal i k formátování vzhledu, ale k tomu se dnes používají spíše kaskádové styly, viz dále.

3.5 Razor

Razor je nástroj, který je v ASP.NET MVC od verze 3, umožňuje použít jazyk C# přímo v HTML kódu webové stránky.

3.6 CSS

Kaskádové styly[4] popisují, jak mají jednotlivé elementy HTML vypadat. Bud' můžou být styly popsány v odděleném souboru, typicky *style.css*, který se pak ale musí nechat v hlavičce HTML stránky naimportovat dvěma způsoby:

```
<head>
<link rel="stylesheet" type="text/css"
      href="style.css">
</head>
```

Což lze v ASP.NET MVC 3+ nahradit za volání s využitím Razor:

```
<head>
@Styles.Render("style.css")
</head>
```

Nebo rovnou zadávat parametry stylu k HTML elementu:

```
<h2 style="margin-top: 20px;">...</h2>
```

3.7 JavaScript

JavaScript[5] je, jak už jméno napovídá, skriptovací jazyk pro tvorbu a ovládání dynamického webového obsahu. Tím se myslí pohyblivé věci, animace obrázků, aktualizace, aniž by se musela aktualizovat celá webová stránka. K webové HTML stránce se přidá podobně jako kaskádové styly:

```
<script type="text/javascript"
  src="/path/to/javascript"></script>
```

Lze odkazovat přímo na online verzi:

```
<script type="text/javascript"
  src="https://code.jquery.com/jquery-3.2.1.min.js">
</script>
```

Nebo lze skript připsat rovnou do HTML kódu, což je vhodné využít pouze pokud je skript opravdu stručný:

```
<script type="text/javascript">
alert("Page_loaded");
</script>
```

3.8 Knockoutjs

Tato technologie je v práci zásadní, ale je uvedena na konci, aby již byly definované ostatní technologie, které jsou pro tuto technologii nutné. Knockout[6] je čistě JavaScriptová knihovna, která pomáhá vytvořit uživatelské rozhraní. Konkrétně stará se o svázání dat uživatelského rozhraní s datovým modelem a dokáže aktualizovat uživatelské rozhraní, na základě změny datového modelu. Oproti například Angularu [7] je knockoutjs lehčí a pro aplikace, které jsou méně složité na práci s daty bohatě dostačuje. Podporuje všechny nejrozšířenější internetové prohlížeče.

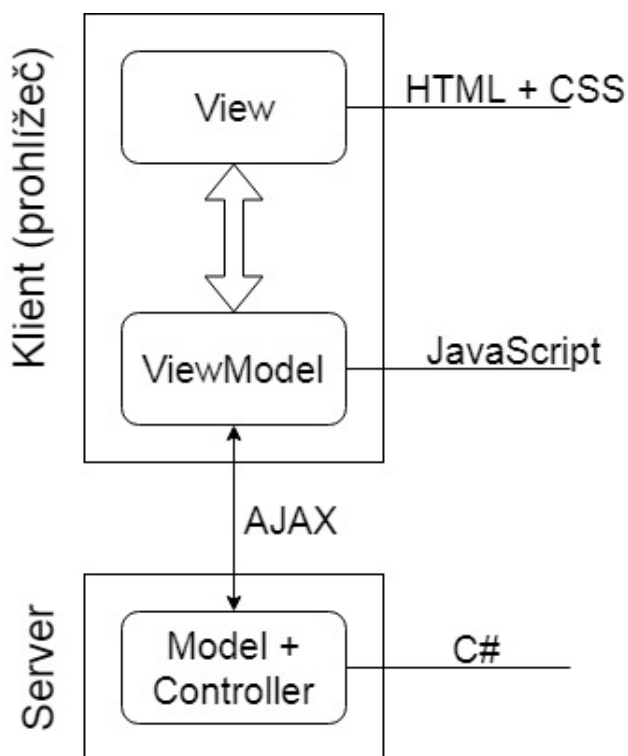
Využívá se zde architektura MVVM, kde hlavní rozdíl mezi ním a MVC architekturou je, že MVVM má *binder* (svázání dat), které umožňuje automatickou komunikaci mezi View a ViewModelem. Jakákoliv uživatelem vyvolaná událost se tak projeví ve ViewModelu a obráceně, pokud se změní nějaká data ve ViewModelu, zobrazí se i uživateli ve View. Na obrázku MVVM Diagram je zobrazeno, jak implementace MVVM (tedy Model-View-ViewModel) vypadá.

3.9 AJAX

Neboli asynchronní JavaScript a XML je využíván při tvorbě interaktivních webových aplikací, kde mění obsah View bez nutnosti celkové aktualizace stránky.

3.10 Bootstrap

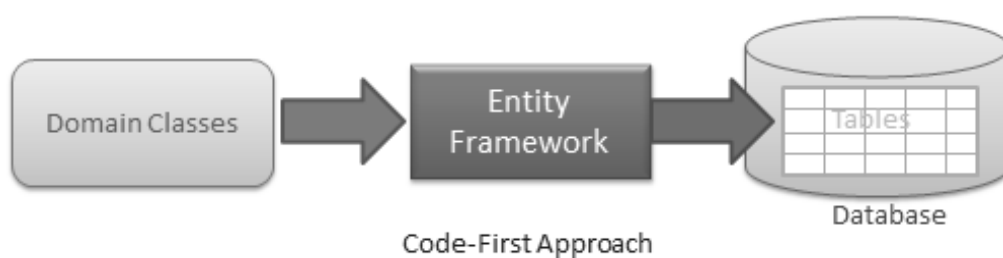
Bootstrap je dnes nejrozšířenější [8] volně dostupný nástroj pro tvorbu responzivních webů. Tento framework vznikl v roce 2010 ve firmě Twitter jako *Twitter Blueprint*. Je to framework obsahující šablony a styly, které může vývojář při tvorbě webu použít a jednoduše tak vytvořit responzivní vzhled. Má rozsáhlou, propracovanou dokumentaci, která je dostupná na oficiálních stránkách [9]. Bootstrap je aktuálně nejnovější ve verzi 4.2.1, já jsem se rozhodl použít verzi 3.4.0, jelikož s touto verzí jsem již v minulosti pracoval a mám s ní vlastní zkušenosti.



Obrázek 3.2: MVVM Diagram + jazyky jednotlivých vrstev

3.11 Entity framework

Entity Framework[10] je určen pro interakci mezi .NET aplikací a relační databází, je to takzvaný ORM nástroj, který má usnadnit mapování mezi prvky relační databáze (tabulky, sloupce) a objektů v aplikaci. Je více způsobů použití tohoto nástroje, první je tzv. *Database first*, kdy se nejdříve vytvoří databátový model. Pro tuto práci bude využit tzv. *Code first*, kde se nejdříve vytvoří model v jazyce C# a z něj se následně vytvoří a namapuje databáze. Nespornou výhodou tohoto způsobu vytvoření je čitelnost kódu.



Obrázek 3.3: EF - Code First přístup

Implementace aplikace

V této kapitole popisuji, jak jsem samotnou aplikaci realizoval. K vývoji jsem použil Visual Studio 2017 Community Edition[11]

4.1 Vývoj

Nejprve jsem zprovoznil jednoduchý projekt s architekturou MVC[12], nevyužívající žádná další rozšíření, a na něj jsem postupně přidával jednotlivou funkcionalitu. Začal jsem pracovat na serverové části, tedy vytvoření tříd potřebných k ukládání dat. Tady jsem použil Entity framework, kde se nastaví kontext databáze a EF se postará o databázi samotnou na základě zadaných modelů. Následně jsem se přesunul k části prezentační, tedy na View. Tady jsem se snažil vytvořit co nejpřehlednější, jednoduchý a zároveň hezky vypadající vzhled aplikace, jelikož to je to jediné, s čím samotný uživatel přijde do kontaktu. Po nespočetném množství přečtených článků a návodů o uživatelských rozhraních, po pročtení spousty rad a tipů ohledně vzhledů webových stránek a nástroje Bootstrap jsem byl s vzhledem spokojen. Přišlo mi, že už je skoro hotovo, a přesunul jsem se k poslední věci a to k propojení samotného View s Modelem. To byl pro mě kámen úrazu a několik probdělých nocí. Nasazení komponenty ViewModel považuji za nejnáročnější část celé práce. S JavaScriptem jsem si moc nerozuměl, tak jsem se mu chtěl vyhnout a nasadit ViewModel pomocí knihovny AutoMapper, kterou se namapují Modely a jim příslušné prvky View. Bohužel jsem čerpal z literatury, která byla zhruba tři roky stará, a má verze AutoMapper byla, jak jsem o dost později zjistil, novější. Po prohraném boji s Automapperem, jelikož jsem aktuální verzi nebyl schopný rozchodit, jsem se rozhodl dát na zkušenosti programátorů na fórech a napsat ViewModel v JavaScriptu, kterému jsem se tolik vyhýbal, a plánoval jsem v něm psát jen

funkce pro výměnu dat s Controllerem. Jelikož jsem v JavaScriptu doposud pracoval jen okrajově, bylo to spoustu četby dokumentace a návodů, ale nakonec lituji, že jsem v něm nepracoval hned od začátku.

4.2 Ukázky kódu

Následující kód ve ViewModelu se stará o načtení uživatelem zadaných dat z View, převede je do formátu JSON a posílá je Controlleru

```
var submitData = {
    ID : jobAnnouncementID ,
    Title : jobAnnouncementTitle ,
    Descr ... //zkraceni zapisu
};

$.ajax({
    type: 'PUT' ,
    url: '/api/JobAnnouncementsAPI/'+jobAnnouncementID ,
    contentType: 'application/json' ,
    data: ko.toJSON(submitData) ,
    success: function (data) {
        localStorage.setItem("Saved", true);
        window.location.href = urlPath + '/';
    }
})
```

Zde je metoda Controlleru, která přijme data a zpracuje si je. Pokud jsou v pořádku, uloží je do kontextu databáze.

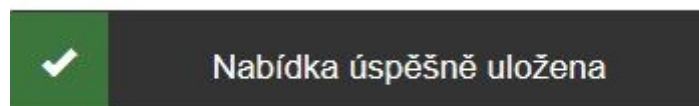
```
// PUT: api/JobAnnouncementsAPI/5
[ResponseType(typeof(void))]
public IHttpActionResult PutJobAnnouncement(int id ,
    JobAnnouncement jobAnnouncement)
{
    if (!ModelState.IsValid)
    {
        return BadRequest(ModelState);
    }
    if (id != jobAnnouncement.ID)
    {
        return BadRequest();
    }
}
```

```
db.Entry(jobAnnouncement).State =
    EntityState.Modified;

try
{
    db.SaveChanges();
}
catch (DbUpdateConcurrencyException)
{
    if (!JobAnnouncementExists(id))
    {
        return NotFound();
    }
    else
    {
        throw;
    }
}
return StatusCode(HttpStatusCode.NoContent);
}
```

4.3 Uživatelské rozhraní

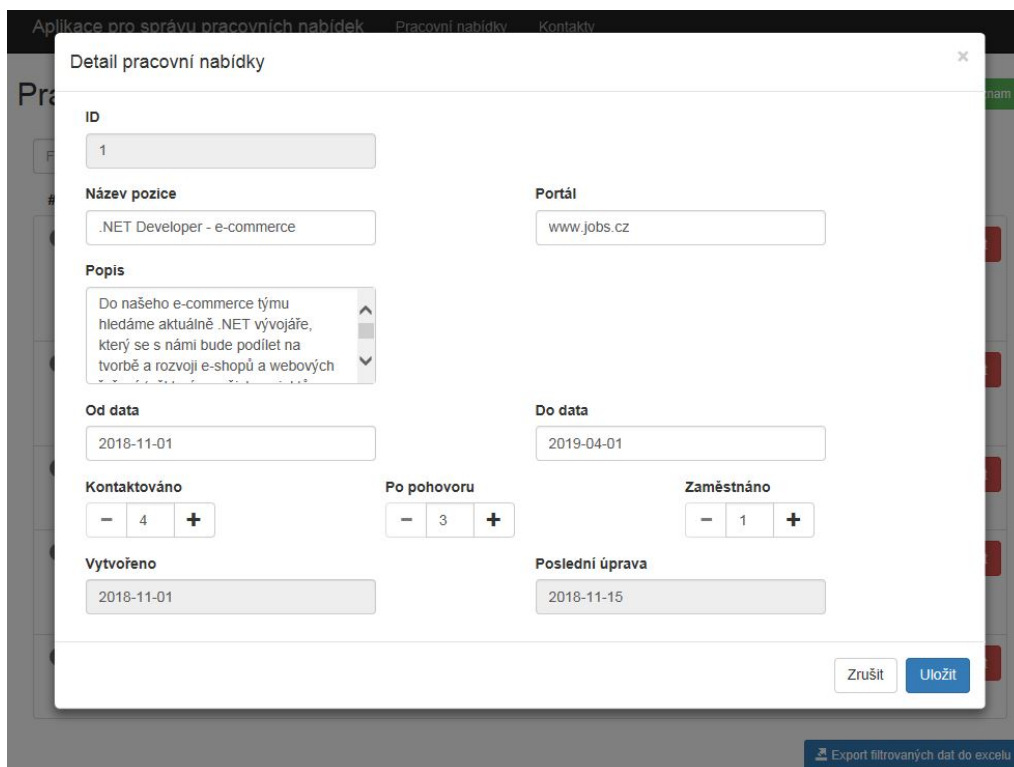
Snažil jsem se vytvořit přehledné a dobře vypadající prostředí. Tomu značně napomohl nástroj bootstrap se spoustou šablon. Aby se uživatel vyhnul přepínání a odkazování na jiné stránky, rozhodl jsem se vytvořit formulář na přidání a editaci záznamu přímo na stránce pomocí vyskakovacího modalu. Díky tomu se uživatel nesetká s žádnou bílou obrazovkou a čekáním na znovunačtení celé webové stránky. Při ukládání a načítání je použito asynchronní volání, po kterém se aktualizuje pouze určitá část stránky a uživatele to tak nijak nerozptyluje. Dále jsou v aplikaci použity elegantní animované notifikace, informující o úspěšnosti uživatelem provedené akce.



Obrázek 4.1: Notifikace úspěšného uložení

4. IMPLEMENTACE APLIKACE

Uživatelské rozhraní má texty a popisky v českém jazyce, využívající lokalizační soubor s těmito texty, umožňující jednoduchý překlad celé aplikace změnou jednoho souboru.



Obrázek 4.2: Vyskakovací modal

Testování a výsledky

Na testování aplikace byl použit nástroj Selenium [13], který dokáže simulovat interakci uživatele s webovou aplikací. Pomocí identifikátorů jednotlivých ovládacích prvků ve webové aplikaci, na nich dokáže tento nástroj vyvolat události, jako je kliknutí, nebo vyplnění textového vstupu. Byl použit spolu s nástrojem NUnit 3[14], který slouží pro tvorbu unit testů. Tento nástroj NUnit je určen pro .NET a je portován z JUnit, který je určen pro jazyk Java.

Bylo zvoleno celkem 6 scénářů a to následovně:

1. Přidání nové nabídky
2. Kontrola data vytvoření nabídky
3. Úprava nabídky
4. Kontrola data poslední úpravy nabídky
5. Smazání nabídky
6. Funkce filtrování

Následující kód představuje jeden z testů a to přidání nového záznamu pracovní nabídky. Návěští každý NUnit test obsahuje části [SetUp], kde se vše nainicializuje a připraví k testu. Zde se nastaví ovladač pro webový prohlížeč Google Chrome⁵. Další návěští [Test] metody značí, že se jedná o samotný test, kde se provedou operace simulující chování uživatele. Nakonec se spustí metoda s návěstím [TearDown], kde probíhá případné vyčištění po testu a uzavření spojení s ovladačem.

⁵Dostupný na: <https://www.google.com/chrome/>

```
public class NUnitTest1
{
    IWebDriver driver;
    string testTitle = "Testovací_nazev";
    [SetUp]
    public void Initialize ()
    {
        driver = new ChromeDriver ();
    }
    [Test]
    public void AddNewJATest ()
    {
        driver.Url = ".../JobAnnouncement";
        driver.FindElement(By.Id("addJA")).Click();
        Thread.Sleep(1500); //nez se modal zobrazi
        //prepnuti na modal, který se zobrazil
        driver.SwitchTo().ActiveElement();
        //nasleduje vyplneny udaju
        driver.FindElement(By.Id("TitleInput"))
            .SendKeys(testTitle);
        driver.FindElement(By.Id("PortalInput"))
            .SendKeys("www.test.com");
        //... zkraceni zapisu...
        driver.FindElement(By.Id("saveBtn")).Click();
        //vyckani na knockout aktualizaci dat
        Thread.Sleep(5000);
        //kontrola jestli se zaznam nachazi v seznamu
        IReadOnlyCollection<IWebElement> list = driver
            .FindElements(By.XPath
                ("//*[contains(text(),' "
                    + testTitle + "')]"));
        Assert.True(list.Count()>0,"Job_announcement_"
            + testTitle + "_not_found!");
    }
    [TearDown]
    public void EndTest ()
    {
        driver.Close ();
    }
}
```

5.1 Průběžné testování během vývoje

Aplikace byla průběžně spouštěna a testována na lokálním IIS Express, které je součástí operačního systému MS Windows. Vývojářský nástroj MS Visual Studio podporuje ladění přímo v prohlížeči Internet Explorer od stejné firmy. Tento ladící nástroj umožňuje krokování a sledování hodnot proměnných, což poměrně usnadní hledání chyb. Ladící nástroj nabízí sice i ladění skriptů, které ale bohužel nechtělo moc spolupracovat, a tak muselo stačit vypisování logů do konzole prohlížeče, což lehce znepříjemňovalo práci. Jako databáze bylo využito MSSQLLocalDB[15], která je také součástí MS Visual Studia.

5.2 Uživatelské testování

Testování uživateli bylo prováděno již při vývoji aplikace. Několika osobám byl zadán úkol v aplikaci něco zaznamenat, nasimulovat práci personálního oddělení. Instrukce pro testery byly pouze, že se jedná o aplikaci pro personální oddělení a že zaznamenává pracovní nabídky, zbytek už byl na nich. Následně byli dotázáni, zda je pro ně vše dostatečně intuitivní, na první pohled jednoduché a co by případně změnili. Toto testování proběhlo celkem úspěšně, na jeho základě bylo jen upraveno pár popisků ve formuláři pro zadávání nové nabídky, aby na první pohled bylo jasné, co je do konkrétního pole očekáváno za vstup.

Závěr

Tato webová aplikace je první webová .NET aplikace, kterou jsem vytvořil od úplného začátku. Doposud jsem na webových aplikacích pracoval, ale pouze jsem je rozšiřoval, či opravoval existující řešení. Nebyl to tedy pro mě snadný úkol. Začátek byl nakonec těžší, než jsem předpokládal, jelikož než jsem se pustil do samotného programování, musel jsem si neprve rozšířit znalosti okolo ASP.NET MVC a zvolit si technologie, které použiji. U programování takové aplikace totiž existuje opravdu nespočetně mnoho možností, jak jakoukoli věc implementovat, takže i po dlouhém předchozím studiu různých technologií jsem se dostal do mnoha „slepých uliček“, kdy jsem se i po několika hodinové práci musel vrátit zpět na začátek. Ze začátku mi nejvíce pomohla kniha[16] o ASP.NET MVC 5, kterou jsem měl možnost si zapůjčit v zaměstnání. Je sice z roku 2015, takže některé součásti již nebyly úplně aktuální, ale alespoň pro základy, na kterých bych mohl stavět, mi dobře posloužila.

Cílem práce bylo navrhnout a implementovat webovou aplikaci pro jednoduchou správu inzerátů tak, aby byla pro uživatele přehledná a měla intuitivní ovládání. Výsledkem je funkční webová aplikace zahrnující všechny požadavky, které byly v analýze stanoveny. Aplikace má českou lokalizaci, pro jednoduché a srozumitelné ovládání, která může být ale kdykoliv a jednoduše přepsána na jakoukoli jinou lokalizaci. Aplikace má pro uživatele atraktivní vzhled, zmínil bych například animované notifikace o úspěšnosti při uložení či smazání nabídek.

Další možná vylepšení

Aplikaci by bylo možné doplnit o podporu evidence kandidátů, správu jejich životopisů, mezd a jejich schvalování příslušnými vedoucími pracovníky

ZÁVĚR

oddělení, kam se ucházejí o místo. K tomuto účelu by jistě bylo nutné zavést autorizaci uživatelů a přidělit jim jednotlivé role, aby každý měl pravomoc upravovat jen to, co spadá do jeho zaměření. Bylo by to jistě ještě hodně pracné, ale pak by již aplikace mohla plnohodnotně zastoupit nynější nástroje pro personální oddělení.

Literatura

- [1] Microsoft: *Localize strings in your UI and app package manifest [online]*. [cit. 2018-12-19]. Dostupné z: <https://docs.microsoft.com/en-us/windows/uwp/app-resources/localize-strings-ui-manifest>
- [2] Microsoft: *Introduction to the C# Language and the .NET Framework [online]*. 2015 [cit. 2018-12-30]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>
- [3] Shannon, R.: *What is HTML? [online]*. 2012 [cit. 2018-12-30]. Dostupné z: <https://www.yourhtmlsource.com/starthere/whatishtml.html>
- [4] Morris, S.: *Tech 101: The Ultimate Guide to CSS [online]*. 2012 [cit. 2018-12-30]. Dostupné z: <https://skillcrush.com/2012/04/03/css/>
- [5] Morris, S.: *Tech 101: What is JavaScript? [online]*. 2012 [cit. 2018-12-30]. Dostupné z: <https://skillcrush.com/2012/04/05/javascript/>
- [6] Google: *Introduction - Knockoutjs [online]*. © 1996 [cit. 2018-11-20]. Dostupné z: <https://knockoutjs.com/documentation/introduction.html>
- [7] Tomas Kirda: *Angular vs. Knockout: Similarities and Fundamental Differences [online]*. 2014 [cit. 2018-11-20]. Dostupné z: <https://www.devbridge.com/articles/angular-vs-knockout-similarities-and-fundamental-differences/>

LITERATURA

- [8] Arsenault, C. *Top 10 Front-End Frameworks of 2018 [online]*, září 2018, [cit. 2018-11-20]. Dostupné z: <https://www.keycdn.com/blog/front-end-frameworks>
- [9] *Bootstrap - The most popular HTML, CSS, and JS library in the world.. [online]*. © 1996 [cit. 2018-11-20]. Dostupné z: <https://getbootstrap.com/docs/4.2/getting-started/introduction/>
- [10] *Entity Framework - Overview [online]*. 2018 [cit. 2018-12-30]. Dostupné z: https://www.tutorialspoint.com/entity_framework/entity_framework_overview.htm
- [11] Microsoft: *Visual Studio 2017 version 15.9.0 [online]*. 2018 [cit. 2018-11-15]. Dostupné z: <https://docs.microsoft.com/en-us/visualstudio/releases/notes/vs2017-relnotes>
- [12] Freeman, A.: *Pro ASP.NET MVC 5*. New York: Apress, 2013, ISBN 978-1430265290.
- [13] *Selenium - Web Browser Automation [online]*. 2010 [cit. 2018-12-30]. Dostupné z: <https://www.seleniumhq.org/>
- [14] *NUnit - unit-testing framework for all .Net languages [online]*. 2017 [cit. 2018-12-30]. Dostupné z: <https://nunit.org/>
- [15] Microsoft: *Microsoft SQL Server 2016 Express LocalDB [online]*. 2016 [cit. 2018-12-30]. Dostupné z: <https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/sql-server-2016-express-localdb?view=sql-server-2017>
- [16] Munro, J.: *ASP.NET MVC 5 with Bootstrap and Knockout.js*. O'Reilly Media, Inc., 2015, ISBN 9781491914373.

Uživatelský návod k použití

Aplikace pro správu pracovních nabídek

Návod k použití

Po vstoupení na adresu, na které je webová aplikace hostována, přistoupíme na panel „Pracovní nabídky“ a zobrazí se nám seznam veškerých pracovních nabídek, viz obrázek Seznam pracovních nabídek.

Chceme-li filtrovat uložené pracovní nabídky, napíšeme hledaný výraz do pole „Filtrovat“. Námi zadaný výraz se bude vyhledávat v polích „Název pozice“ a „Popis“.

Tlačítko „Export filtrovaných dat do excelu“ nám umožní uložení pracovních nabídek, bude na ně aplikován výše zmíněný filtr, a tak uložíme pouze ty nabídky, které chceme. Export je do souboru s příponou *.xlsx*, podporovanou programy Kancelářských balíků, jako jsou Microsoft Excel, LibreOffice a další.

Pro vložení nové pracovní nabídky stiskneme tlačítko „Přidat nový záznam“. Zobrazí se modální okno s textovými poli pro nový záznam pracovní nabídky, viz obrázek 'Nová pracovní nabídka'. V tomto okně vyplníme požadovaná pole (pole pro název pozice je povinné) a stiskneme tlačítko „Uložit“ pro zaznamenání nabídky. V opačném případě pak tlačítko „Zrušit“, nebo „X“ v pravém horním rohu pro zahazení změn.

Po uložení záznamu se zespodu okna vysune notifikace o úspěšnosti této operace. Stejně tak při smazání záznamu nabídky nás aplikace upozorní, zda byla nabídka úspěšně smazána.

A. UŽIVATELSKÝ NÁVOD K POUŽITÍ

Aplicace pro správu pracovních nabídek Pracovní nabídky Kontakty

Pracovní nabídky 5 Počet záznamů + Přidat nový záznam

Filtrovat Pole pro filtraci záznamů

#	Název pozice	Popis	Do data	Možnosti
1	.NET Developer - e-commerce	Do našeho e-commerce týmu hledáme aktuálně .NET vývojáře, který se s námi bude podílet na tvorbě a rozvoji e-shopů a webových řešení (některé z našich projektů a řešení je možné najít na: http://webomat.sprinx.com/Reference).	2019-04-01	Upravit Smazat
2	Front-end Developer - ReactJS (možnost HO režimu)	* HTML, CSS, sass, less * JavaScript React, NodeJS, Flow, Neutrino, react scripts	2019-04-01	Upravit Smazat
3	Tester: SoapUI + SQL + pozitivní přístup k vývojářům	Chybí nám zkušený tester, který obsáhne SoapUI, SQL, i nějaký kus automatizace testů (skriptování postačí). Velký důraz klademe zejména na pokročilou znalost SoapUI.	2019-03-01	Upravit Smazat
4	C / C++ Programátor	Vývoj nové verze informačního systému v C++ Vývoj probíhá v prostředí OS Windows nebo Linux (možnost specializace) Spolupráce při analýze a testování funkčnosti vyvíjených aplikací	2019-01-10	Upravit Smazat
5	.NET ASP.NET Programátor / .NET Senior Developer	Do nově otevřeného pražského vývojového centra zahraniční SW společnosti hledáme juniorské i zkušené ASP.NET programátory.	2019-10-10	Upravit Smazat

[ID záznamu](#) Export filtrovaných dat do excelu

Obrázek A.1: Seznam pracovních nabídek

Applikace pro správu pracovních nabídek Pracovní nabídky Kontakty

Detail pracovní nabídky

ID
Zde bude po vytvoření ID

Název pozice
Název

Popis
Popis

Portál
Portál

Od data
Platnost od

Do data
Platnost do

Kontaktováno
- +

Po pohovoru
- +

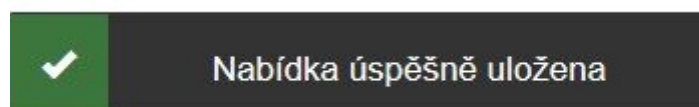
Zaměstnáno
- +

Vytvořeno
Datum přidání

Poslední úprava
Datum poslední úpravy

Zrušit Uložit

Obrázek A.2: Přidání nového záznamu



Obrázek A.3: Notifikace uložení záznamu



Obrázek A.4: Notifikace smazání záznamu

Seznam použitých zkratk

AJAX Asynchronous JavaScript and XML

CIL Common Intermediate Language

CSS Cascading Style Sheets

EF Entity Framework

HR Human Resources

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

IIS Internet Information Services

JS JavaScript

JSON JavaScript Object Notation

MVC Model-View-Controller

MVVM Model-View-ViewModel

ORM Object-Relational Mapping

SQL Structured Query Language

URL Uniform Resource Locator

XML Extensible Markup Language

Obsah přiloženého CD

	readme.txt	stručný popis obsahu CD
	src		
		impl Visual Studio řešení, zdrojové kódy
		thesis zdrojová forma práce ve formátu L ^A T _E X
	text	text práce
		thesis.pdf text práce ve formátu PDF