

Katedra informatiky  
Přírodovědecká fakulta  
Univerzita Palackého v Olomouci

# BAKALÁŘSKÁ PRÁCE

Webový portál pro sportovní klub



2020

Vedoucí práce: Mgr. Jiří Zaccpal,  
Ph.D.

Martin Urbanec

Studijní obor: Aplikovaná informatika,  
prezenční forma

## **Bibliografické údaje**

Autor: Martin Urbanec  
Název práce: Webový portál pro sportovní klub  
Typ práce: bakalářská práce  
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci  
Rok obhajoby: 2020  
Studijní obor: Aplikovaná informatika, prezenční forma  
Vedoucí práce: Mgr. Jiří Zaccpal, Ph.D.  
Počet stran: 51  
Přílohy: 1 CD/DVD  
Jazyk práce: český

## **Bibliographic info**

Author: Martin Urbanec  
Title: Web portal for a sport club  
Thesis type: bachelor thesis  
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc  
Year of defense: 2020  
Study field: Applied Computer Science, full-time form  
Supervisor: Mgr. Jiří Zaccpal, Ph.D.  
Page count: 51  
Supplements: 1 CD/DVD  
Thesis language: Czech

## **Anotace**

*Cílem práce je vytvořit webový portál pro sportovní klub. Výsledná webová stránka bude sestávat z grafického rozhraní v desktopové verzi a bude optimalizovaná pro mobilní zařízení.*

## **Synopsis**

*The aim of the thesis is to create web portal for a sport club. Web application will consist of a graphical interface for the desktop devices and will be optimized for the mobile devices.*

**Klíčová slova:** portál; webový portál; webová stránka; C#; ASP.NET Core

**Keywords:** portal; web portal; web page; C#; ASP.NET Core

Chtěl bych poděkovat mému vedoucímu práce Mgr. Jiřímu Zaccalovi, PhD. za jeho vstřícný přístup, cenné připomínky a odborné rady, kterými přispěl k vypracování mé bakalářské práce. Mé poděkování rovněž patří Šárce Navrátilíkové, která mi pomáhala s návrhem některých částí uživatelského rozhraní. Dále děkuji rodině za podporu a vytvoření podmínek k sepsání této práce.

*Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.*

datum odevzdání práce

podpis autora

# Obsah

<b>1</b>	<b>Úvod</b>	<b>9</b>
1.1	Zadání práce . . . . .	9
1.1.1	Zaznamenávání docházky . . . . .	10
1.1.2	Kalendář soutěží a školení . . . . .	10
1.1.3	E-mailové notifikace . . . . .	10
1.1.4	Optimalizace zobrazení pro mobilní zařízení . . . . .	11
1.1.5	Případ použití . . . . .	11
1.2	Vhodné technologie . . . . .	11
1.2.1	Webový portál . . . . .	11
1.2.2	Uložení dat . . . . .	11
<b>2</b>	<b>Podobné aplikace</b>	<b>12</b>
2.1	Bakaláři . . . . .	12
2.2	Uplikace . . . . .	13
<b>3</b>	<b>Použité technologie</b>	<b>15</b>
3.1	Jazyk C# . . . . .	15
3.2	ASP.NET Core . . . . .	15
3.2.1	Razor Pages . . . . .	15
3.2.2	Entity Framework Core . . . . .	16
3.2.3	Identity . . . . .	17
3.2.4	CSV Helper . . . . .	18
3.2.5	MimeKit/MailKit . . . . .	18
3.3	Microsoft SQL Server . . . . .	19
3.4	Diagram databáze . . . . .	20
<b>4</b>	<b>Programátorská příručka</b>	<b>21</b>
4.1	Identity . . . . .	21
4.2	Odesílání e-mailů . . . . .	22
4.3	Popis tříd (modelů) . . . . .	23
4.3.1	Třída Attendance . . . . .	23
4.3.2	Třída CalendarData . . . . .	23
4.3.3	Třída DefaultPasswordCheck . . . . .	24
4.3.4	Třída Event . . . . .	24
4.3.5	Třída Group . . . . .	24
4.3.6	Třída Request . . . . .	25
4.3.7	Třída ScheduledEvent . . . . .	25
4.3.8	Třída UploadedFile . . . . .	25
4.3.9	Třída UserGroup . . . . .	26
4.3.10	Třída UserName . . . . .	26
4.3.11	Třída Warning . . . . .	26
4.4	Úlohy na pozadí . . . . .	28
4.4.1	EmailService . . . . .	28

4.5	Napojení na databázi . . . . .	29
4.6	Poznámky . . . . .	29
4.6.1	CSS . . . . .	29
4.6.2	Ikony . . . . .	29
<b>5</b>	<b>Uživatelská příručka</b>	<b>30</b>
5.1	Uživatel (User) . . . . .	30
5.1.1	Správa uživatelského účtu . . . . .	30
5.1.2	Rozvrh a docházka . . . . .	31
5.1.3	Termínový kalendář . . . . .	33
5.1.4	Žádosti . . . . .	33
5.2	Trenér (Coach) . . . . .	33
5.2.1	Vytvoření skupiny . . . . .	33
5.2.2	Přidání a úprava rozvrhových událostí . . . . .	34
5.2.3	Zápis docházky . . . . .	35
5.2.4	Statistika docházky pro trenéry . . . . .	35
5.3	Administrátor (Administrator) . . . . .	36
5.3.1	Vytvoření termínového kalendáře . . . . .	36
5.3.2	Registrace uživatelů . . . . .	37
5.3.3	Správa uživatelů . . . . .	37
5.4	Obnovení hesla . . . . .	38
	<b>Závěr</b>	<b>39</b>
	<b>Conclusions</b>	<b>40</b>
<b>A</b>	<b>Přehled tříd a jejich vlastností</b>	<b>41</b>
A.1	Třída Attendance . . . . .	41
A.2	Třída CalendarData . . . . .	41
A.3	Třída CalendarFilter . . . . .	42
A.4	Třída CSV . . . . .	42
A.5	Třída DefaultPasswordCheck . . . . .	43
A.6	Třída Event . . . . .	43
A.7	Třída EventChange . . . . .	44
A.8	Třída Group . . . . .	44
A.9	Třída MyEvent . . . . .	44
A.10	Třída MyUser . . . . .	45
A.11	Třída Request . . . . .	46
A.12	Třída ScheduledEvent . . . . .	46
A.13	Třída UploadedFile . . . . .	46
A.14	Třída UserFilter . . . . .	47
A.15	Třída UserGroup . . . . .	47
A.16	Třída UserName . . . . .	47
A.17	Třída Warning . . . . .	48

B Obsah přiloženého CD	49
Seznam zkratk	50
Bibliografie	51

## Seznam obrázků

1	Případ použití portálu . . . . .	11
2	Mobilní aplikace Bakaláři . . . . .	13
3	Mobilní aplikace UPlikace (starý design) . . . . .	14
4	Mobilní aplikace UPlikace (nový design) . . . . .	14
5	Diagram databáze pro webový portál . . . . .	20
6	Nastavení účtu s úspěšně ověřenou e-mailovou adresou . . . . .	31
7	Ukázka výpisu rozvrhových událostí . . . . .	32
8	Náhled detailů události . . . . .	34
9	Statistika docházky na rozvrhové události . . . . .	35

## Seznam tabulek



# 1 Úvod

Cílem práce je vytvoření webového portálu pro sportovní klub T.J. Sokol Zlín-Prštne, speciálně pro oddíl sálové cyklistiky (krasojízda, kolová). Portál bude přístupný pouze pro členy klubu, kteří budou mít různá oprávnění – administrátor, trenéři a uživatelé.

Tento portál, který jsem se rozhodl vytvořit pro klub, jehož jsem zároveň i členem, bude umožňovat zápis docházky závodníků na tréninky, zadávání žádostí k vyřízení (např. proplacení cesty vlastním vozidlem na turnaj nebo žádost o náhradní díly na kolo) s možností sledovat stav podaných žádostí (zda byla přijata nebo vyřízena) a e-mailové notifikace na plánované závody, schůze, školení nebo obecné události. Občas se totiž stane, že je nutné členy oddílu neustále upomínat na důležité události, schůze, či jim připomínat věci, které mají k vyřízení.

Není to zcela častý jev, ale někteří členové někdy nepřijdou na pravidelný trénink, aniž by předtím svou účast omluvili. Proto součástí portálu bude již zmíněné zaznamenávání absence jednotlivých členů a při nadměrné absenci by dostali upozornění a stejně tak jejich trenér, který by následně mohl zvážit udělení trestu.

V posledních letech je program soutěží aktuálního roku v tištěné podobě dostupný pouze pro závody v kolové, nikoliv však pro krasojízdu. Případně je zde ještě disciplína *SC* – sálová cyklistika, tj. obě disciplíny dohromady na jedné soutěži. Pro závodníky krasojízdy je to určitá nevýhoda, protože si nemohou zjistit v jakých termínech se jejich soutěže konají. Pro tento případ bude portál obsahovat i termínový kalendář jak pro závody v kolové, tak i v krasojízdě a nebudou chybět ani společné soutěže, tedy disciplína *SC*.

Registrace a editace uživatelů (jejich oprávnění, uživatelské jméno, jméno a příjmení) bude pouze v kompetenci administrátora. Speciální uživatelský účet s názvem „admin“ nebude možné smazat ani editovat práva nebo uživatelské jméno. Administrátor bude rovněž řešit vytvoření rozvrhu, rozdělení uživatelů do rozvrhových skupin a v neposlední řadě i upozornění danému uživateli na doposud nepotvrzenou e-mailovou adresu. Změnu hesla nebo telefonního čísla budou moci uživatelé provádět sami. Stejně tak i změnu e-mailové adresy, nicméně poté bude nutné zažádat o zaslání odkazu pro potvrzení na nově zvolenou adresu.

## 1.1 Zadání práce

- Portál by měl umožňovat zaznamenávat docházku jednotlivých členů oddílu.
- Součástí portálu bude kalendář soutěží a školení s možností filtrování.
- Portál bude umožňovat zasílat upozornění na plánované akce pomocí e-mailu.
- Web bude optimalizován pro zobrazení na mobilním zařízení nebo bude součástí práce mobilní aplikace.

### 1.1.1 Zaznamenávání docházky

Pro vytvoření rozvrhu budou uživatelé nejdříve rozděleni do skupin, kdy jeden uživatel může být členem více skupin, a ke každé skupině budou vytvořeny odpovídající rozvrhové události. Každá rozvrhová událost však může být přidělena pouze jedné skupině. Ke každé události budou poté vygenerovány její termíny (datum, čas začátku a předpokládaný čas konce).

Samotná docházka bude zaznamenávána na jednotlivé termíny dané rozvrhové události. Zde bude u každého termínu počet záznamů roven počtu uživatelů, kteří tento termín navštěvují (mají jej v rozvrhu). Výchozí hodnota docházky bude „N“ (tj. nezapsáno). Uživatel s oprávněním *Trenér* bude moci provést zápis absence na daný termín a to zda uživatel:

- Byl přítomen na termínu (bude označeno hodnotou „S“)
- Nebyl přítomen na termínu (bude označeno hodnotou „A“)
- Přišel pozdě na termín (bude označeno hodnotou „P“)
- Odešel dříve z termínu (bude označeno hodnotou „O“)
- Se omluvil (bude označeno hodnotou „OA“)

Kompletní přehled o docházce členů na jednotlivé události si poté bude moci zobrazit ve speciální trenérské statistice.

*Trenér* provede zápis docházky pouze na rozvrhových událostech, ve kterých je zapsán. Zápis docházky všech událostí bude umožněn pouze administrátorovi portálu.

### 1.1.2 Kalendář soutěží a školení

Kalendář soutěží je každoročně vytvářen v programu *Microsoft Excel*, tudíž není problémem jeho převod do formátu *.csv*. Výsledný *CSV* soubor bude poté zpracován a data z něj budou nahrána do databáze. Soubor bude rovněž uložen na serveru a administrátor portálu jen vybere, ze kterého souboru se mají data pro kalendář zobrazovat. Pokud dojde ke změně, data v databázi budou smazána a tabulka bude naplněna daty z nové verze kalendáře.

Uživatel bude moci při vyhledávání v kalendáři využít filtrování a tím rychle a pohodlně nalézt soutěž či školení, které hledá, nebo se jen podívat, které soutěže jej čekají v daném měsíci apod.

### 1.1.3 E-mailové notifikace

Pro zaslání zpráv pomocí e-mailu může být použit některý z balíčků galerie *NuGet*<sup>1</sup>.

---

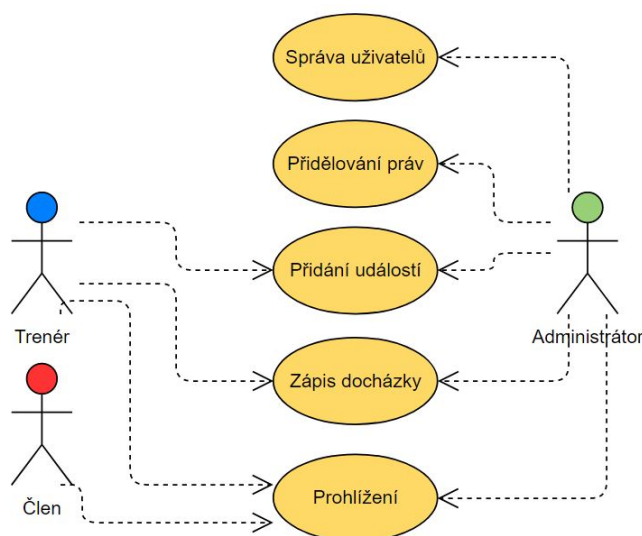
<sup>1</sup>NuGet je správce balíčků pro .NET a obsahuje balíčky používané všemi autory a spotřebiteli

E-mailová zpráva bude odeslána například pro ověření účtu, při změně e-mailové adresy, v případě podávání žádostí nebo před blížící se událostí v rozvrhu (platí pro školení, soutěže nebo obecné jednorázové události, u tréninků notifikace odesílány nebudou).

#### 1.1.4 Optimalizace zobrazení pro mobilní zařízení

Součástí portálu bude knihovna CSS pravidel *Bootstrap*, která zajistí kompatibilitu uživatelského rozhraní i pro mobilní zařízení.

#### 1.1.5 Příklad použití



Obrázek 1: Příklad použití portálu

## 1.2 Vhodné technologie

### 1.2.1 Webový portál

Vhodnou technologií pro implementaci portálu by mohla být platforma ASP.NET Core, jež nám umožňuje tvorbu webových aplikací v jazyce C#. Výhodou může být velké množství funkcí, které .NET Core nabízí jako je například podpora *LINQ* dotazů nebo spousta rozšiřujících knihoven, které nám ulehčují práci. Můžeme zmínit i větší bezpečnost než například u PHP.

### 1.2.2 Uložení dat

Pro uložení dat při použití ASP.NET Core je možné použít databázi SQL Server od Microsoftu. ASP.NET Core obsahuje knihovnu Entity Framework Core, která byla vyvíjena právě pro MSSQL.

## 2 Podobné aplikace

Podobných aplikací existuje celá řada, větší firmy mají zcela jistě své vlastní portály pro zaměstnance. V dnešní době už se také používá na mnoha školách elektronická žákovská knížka, kde mj. žáci vidí svou absenci nebo jim mohou učitelé zasílat hromadné zprávy, či zadávat domácí úkoly. Hlavní účel těchto aplikací je v tomto případě pro analýzu podobných aplikací největší nevýhodou – k celému systému mají přístup pouze zaměstnanci firmy, případně studenti dané školy. Na základě mé zkušenosti jsem vybral systém *Bakaláři*, se kterým jsem měl možnost se osobně setkat při studiu na střední škole a mobilní aplikaci *UPlikace*, kterou využívám při studiu na vysoké škole.

### 2.1 Bakaláři

*Bakaláři* je placený systém pro správu školy určený především pro základní a střední školy. Nahrazuje klasické papírové verze „žakovských knížek“ a umožňuje rodičům žáků kdykoliv nahlédnout, zda je jejich dítě ve škole, jaké má známky a jestli správně plní své povinnosti. Kromě zaznamenávání absence a vkládání známek umožňuje systém učitelům komunikaci s rodiči, vkládání domácích úkolů, či vyplnění aktuálně vyučovaného tématu.

Administrátor systému je oprávněn spravovat školní matriku (přidávání nových studentů, úprava osobních údajů, . . .), tvorba rozvrhu pomocí generátoru rozvrhů implementovaného přímo v systému, zadávání suplování, vytváření plánů akcí nebo rozpisu maturit. Žákům slouží ke kontrole svých povinností internetová žákovská knížka, tedy desktopová verze aplikace i mobilní aplikace dostupná na systémech *Android* a *iOS*<sup>2</sup>.

#### Výhody/Inspirace:

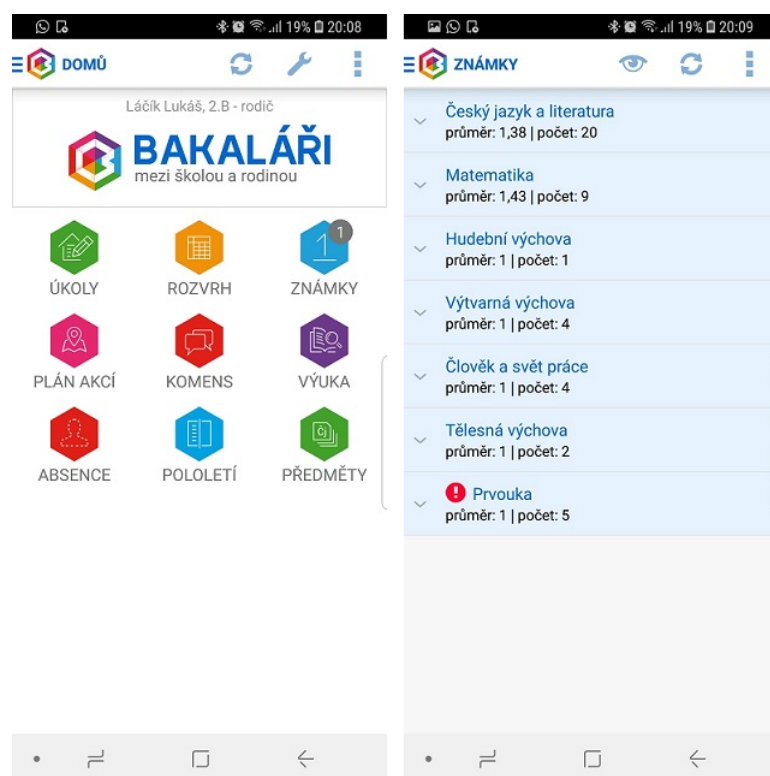
- Možnost uložení velkého množství informací o žácích.
- Poměrně přehledné uživatelské rozhraní pro uživatele jak webové, tak mobilní verze.

#### Nevýhody:

- Notifikace nově přidané známky se zobrazuje konstantní časovou dobu, tj. pokud se do aplikace přihlásím a známku si zobrazím, stále mám upozornění o nové známce. Toto upozornění se zobrazovalo cca. pět dnů.

---

<sup>2</sup>Přiložený obrázek vložen se souhlasem paní Ing. Martiny Láčíkové.



Obrázek 2: Mobilní aplikace Bakaláři

## 2.2 UPlikace

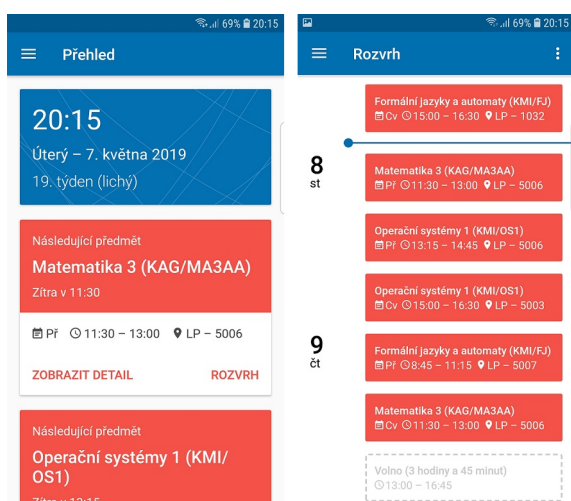
Mobilní aplikace studijní agendy pro studenty Univerzity Palackého. Aplikace je zdarma a je velmi přehledná, jednoduchá a intuitivní. Nabízí možnost zobrazit aktuální rozvrh včetně akcí mimo pravidelný plán (např. zkoušky), zobrazit výčet aktuálně studovaných předmětů, či přepnutí na předchozí akademické roky. Umožňuje také registraci na zkoušky a nechybí ani upozornění na splnění předmětu nebo vypsání termínu zkoušky, popřípadě zápočtu. Během tvorby této bakalářské práce navíc proběhla podstatná změna designu aplikace. Proto zde uvádím srovnání jak staré, tak i nové verze *UPlikace*.

### Výhody/Inspirace:

- Přehledné a intuitivní uživatelské rozhraní
- Možnost zápisu na zkoušky
- Odkazy na související aplikace (např. Menzu UPOL)

### Nevýhody:

- Nemožnost zobrazit si osobní údaje (např. studijní číslo, adresu, ...)



Obrázek 3: Mobilní aplikace Uplikace (starý design)



Obrázek 4: Mobilní aplikace Uplikace (nový design)

## 3 Použité technologie

### 3.1 Jazyk C#

C# je objektově orientovaný programovací jazyk a je úzce zpjat s platformou .NET, případně .NET Core a stejně jako obě zmíněné architektury je vyvíjen společností Microsoft[1]. Důvod, proč využívám pro vytváření mého projektu právě tento jazyk je, že zároveň využívám architekturu ASP.NET Core.

### 3.2 ASP.NET Core

ASP.NET Core je open-source architektura sloužící k vytváření webových aplikací. Narozdíl od frameworku .NET, který je určen výhradně pro Windows, je její bezesporu největší výhodou platformová nezávislost. Nezáleží tedy zda výsledná aplikace vytvořená v ASP.NET Core poběží na systému Windows, Linux nebo macOS. Další výhodou je nepřehorné množství doplňujících knihoven (např. [CSV Helper](#) nebo [MailKit](#)), které nám usnadňují práci a většina z nich je navíc dostupná k volnému používání.

Aktuální verze ASP.NET Core je verze 3.0, nicméně pro vytvoření portálu používám ještě verzi 2.1, neboť v době, kdy jsem začínal pracovat na projektu, nebyla ještě verze 3.0 dostupná.

ASP.NET Core nám umožňuje vytvářet dynamické webové stránky. V praxi to znamená, že pokud uživatel přistupuje na daný web, vyšle se požadavek na server, na kterém běží ASP.NET Core a ten následně vygeneruje stránku v jazyce HTML. Uživatel tedy vidí pouze výslednou HTML stránku a nemusí mít ani ponětí, co se děje na pozadí při vykonávání požadavku. Při vytváření výsledné stránky se vykonává kód jazyka C# (např. požadavek na získání dat z databáze, odeslání e-mailu, různé výpočty. . .). Narozdíl od PHP, kde píšeme kód do stejného souboru jako HTML a rovněž ve výsledné stránce po něm není ani památky, v ASP.NET Core máme kód v C# oddělen od HTML kódu. Tímto je celý kód přehlednější, což je velmi výhodné pro vytváření větších projektů. Neznamená to však, že bychom do HTML nemohli napsat ani kousek kódu v C#. Občas je nezbytné provést cyklus nebo podmínku. Vložení C# kódu do HTML můžeme provést pomocí znaku @.

#### 3.2.1 Razor Pages

Ve frameworku .NET je nejčastější technologie MVC (Model, View, Controller). .NET Core nám však přináší novinku – Razor pages.

Stránka Razor obsahuje HTML kód, který může být doplněn i kódem v C# jak bylo zmíněno výše. Výsledné HTML (ovšem už bez C# kódu) je pak zobrazeno uživateli. Součástí stránky Razor je i tzv. „controller“ třídy PageModel a každá stránka Razor má svůj soubor této třídy. Tento soubor obsahuje pouze kód v jazyce C#. Po vytvoření obsahuje metodu OnGet, která se zavolá při příchodu

na stránku. Pokud je součástí stránky i formulář, pak tento „controller“ obsahuje i metodu `OnPost`, která nám zajistí zpracování dat z formuláře. Soubor `PageMode` třídy může také obsahovat další uživatelem definované pomocné metody. Stránky Razor poznáme podle přípony `.cshtml` a soubory `PageModel` třídy pak podle `.cshtml.cs`. Názvy obou souborů jsou dle konvence shodné.

Oproti MVC, kdy jsme si museli vytvořit samostatně `Model`, `View` i `Controller`, je to výrazné zjednodušení, neboť při vytvoření Razor stránky `.cshtml` se nám automaticky vygeneruje i soubor třídy `PageModel`.

### 3.2.2 Entity Framework Core

Entity Framework Core je určen pro použití na platformě ASP.NET Core. Jedná se o objektově relační mapovač a umožňuje nám pracovat s daty objektově orientovaným způsobem.[2]

Třídou EF Core je `DbContext` a třída `ApplicationDbContext` z této třídy dědí (pokud tomu není jinak, jako v případě použití `Identity`). `DbContext` nám zajistí připojení k databázi a můžeme provádět přidávání, úpravy nebo mazání dat. S výsledky z databáze pracuje jako se seznamem (tudíž máme přístupné všechny seznamové operace). Pro uložení všech změn slouží metoda `SaveChanges`, která všechny změny, které chceme provést, zabalí do transakce a díky tomu se provedou jako atomická operace.

Pro práci s určitou tabulkou v databázi je potřeba vytvořit si `Model`, tj. třídu, jejíž atributy budou odpovídat atributům dané tabulky v databázi. A tuto třídu poté přidat pomocí vlastnosti `DbSet` ve třídě `ApplicationDbContext` jak je uvedeno v následujícím ukázkovém kódu.

```
1 public class ApplicationDbContext : IdentityDbContext
2 {
3     public ApplicationDbContext(DbContextOptions<ApplicationDbContext>
4         options) : base(options)
5     {
6     }
7     public DbSet<UserName> UserNames { get; set; }
8     public DbSet<Request> Requests { get; set; }
9 }
```

Zdrojový kód 1: Nastavení třídy `ApplicationDbContext` pro práci s databází za pomoci EF Core



A nyní ještě jak by mohl vypadat zmíněný Model.

```
1 public class UserName
2 {
3     public string UserId {get; set;}
4     public string Name {get; set;}
5 }
```

Zdrojový kód 2: Model

Nyní tedy můžeme z databáze získat díky Modelu UserName z tabulky UserNames jména a ID uživatelů.

Velkou výhodou EF Core je, že při používání databáze nepotřebujeme psát dotazy v jazyce SQL (v případě použití SQL databázi), ale používáme LINQ, tj. integrovaný dotazovací jazyk .NET Core.

Například dotaz pro zobrazení jména daného uživatele (jehož ID máme k dispozici) vypadá následovně:

```
1 //promenna _context je tridy ApplicationDbContext
2
3 var result = _context.UserNames.Where(p => p.UserId = id).Select(p
    => p.Name).FirstOrDefault();
```

Zdrojový kód 3: Dotaz v jazyce LINQ

### 3.2.3 Identity

ASP.NET Core Identity je systém ověřování, díky kterému můžeme vytvořit v naší aplikaci běžící na platformě .NET Core části, které jsou dostupné pouze přihlášeným uživatelům a těmto uživatelům přiřadit i role a filtrovat tak obsah a funkce, ke kterým budou mít přístup.

Při vytváření projektu je nutné zvolit si možnost, že chceme v aplikaci používat systém ověřování. Po vytvoření projektu se nám taktéž vytvoří v databázi několik tabulek, kde budou uchovávána data o uživatelích a jejich rolích. Pokud si vytvoříme databázi později, stačí zadat příkaz `update-database` do **Konzole správce balíčků**. Poté budou vytvořeny všechny tabulky pro správné fungování ASP.NET Core Identity. Dále je pro správné fungování Identity nutné zajistit, aby třída `ApplicationDbContext` dědila ze třídy `IdentityDbContext` (která dědí z výše zmíněné třídy `DbContext` knihovny EF Core).

Identity sestává z uživatelského rozhraní, kde je vytvořeno přihlášení a registrace uživatelů, úprava jejich údajů jako je heslo, telefonní číslo nebo e-mail a uživatelé mají rovněž možnost obnovit si zapomenuté heslo. Mimo jiné můžeme přidat i dvoufaktorové ověřování nebo externí přihlášení (tyto možnosti nejsou v mém portálu dostupné).

Identity nám dále nabízí výše zmíněné vytvoření rolí a přidělení jednotlivým uživatelům. Máme tedy zajištěno, že uživatel bude moci přistoupit pouze na stránky, na které mu to dovolí jeho úroveň oprávnění (role) a v případě přístupu na stránku, na kterou by neměl mít přístup, dojde k přesměrování na stránku s informací, že mu byl přístup odepřen.

Identity se rovněž stará o to, zda má uživatel ověřenou e-mailovou adresu. Pokud tak doposud neučinil, může si nechat zaslat nový ověřovací e-mail a svou adresu tak potvrdit.

### 3.2.4 CSV Helper

Knihovna dostupná z galerie *NuGet* pod licencí *MPL*<sup>3</sup>. Umožňuje rychlé a pohodlné čtení a zápis dat v CSV souborech. Pro čtení ze souboru je třeba vytvořit instanci třídy `StreamReader` obsahující cestu k danému souboru a třídu obsahující atributy jako jsou v hlavičce CSV souboru, ze kterého budeme načítat data. Samotné čtení ze souboru poté provedou metody implementované v knihovně `CSV Helper`<sup>4</sup>.

```
1 using (var reader = new StreamReader(path))
2 using (var csvReader = new CsvReader(reader))
3 {
4     var records = csvReader.GetRecords<CSV>();
5 }
```

Zdrojový kód 4: Čtení dat z CSV souboru

Proměnná `records` bude seznamem obsahující jednotlivé řádky CSV souboru jako položky seznamu.

### 3.2.5 MimeKit/MailKit

Knihovna dostupná z galerie *NuGet* pod licencí *MIT*. Díky této knihovně můžeme implementovat odesílání e-mailových zpráv do webového portálu. Knihovna disponuje funkcemi pro vytvoření textové zprávy, kterou následně přeformátuje (např. do jazyka HTML, který je běžně pro e-mailové zprávy používán). Nejdůležitějšími parametry zprávy jsou e-mailová adresa odesílatele a příjemce, předmět zprávy, ale především tělo zprávy – tedy samotný text. Dále nám *MailKit* zajistí připojení k SMTP serveru, kdy je zadána adresa serveru, port a zda server používá autentifikaci, popřípadě jakou (SSL nebo STARTTLS). Po úspěšném připojení může být e-mailová zpráva odeslána a poté dochází k odpojení od serveru<sup>5</sup>.

---

<sup>3</sup>Microsoft Public License

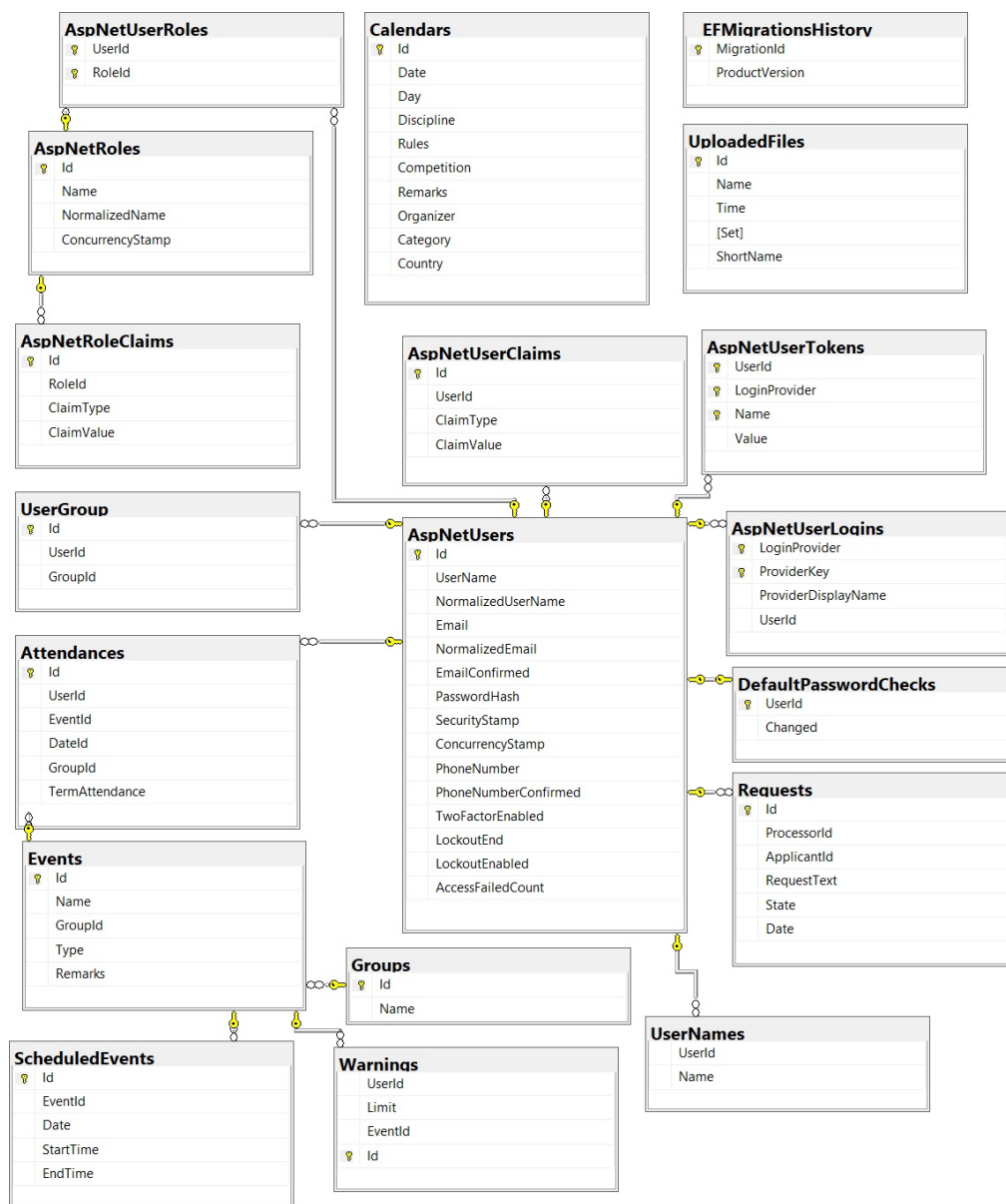
<sup>4</sup>Verze knihovny použitá v mé práci je 12.2.3.

<sup>5</sup>Verze knihovny použitá v mé práci je 2.5.2.

### 3.3 Microsoft SQL Server

Microsoft SQL Server je relační databázový systém. Jeho nejčastější využití je u internetových obchodů a větších firem. Narozdíl od systémů *MySQL*, *MariaDB* nebo *Postgre SQL*, které jsou zdarma, je MSSQL placeným systémem. Lze však používat verzi *Express*, která je zdarma ke stažení a následnému použití. Verze *Express* poskytuje základní funkce databázového serveru, které postačí pro můj webový portál. Plná verze se všemi funkcemi je vhodná především u již zmíněných e-shopů a velkých firem.

### 3.4 Diagram databáze



Obrázek 5: Diagram databáze pro webový portál

## 4 Programátorská příručka

### 4.1 Identity

**Identity** je rozhraní, pro pohodlnou správu uživatelů. Implementuje velké množství metod pro vytváření, mazání a úpravu uživatelů, které jsou přístupné přes třídu `userManager<IdentityUser>`. Rovněž jsou k dispozici metody pro správu rolí, které je možné použít přes třídu `RoleManager<IdentityRole>`.

Základem celého rozhraní je databáze, kde jsou ukládána data o uživateli. Po vytvoření projektu dojde i k vytvoření všech potřebných tabulek (v [diagramu databáze](#) se jedná o tabulky s předponou v názvu „AspNet“). Součástí je i uživatelské rozhraní, které je možné si přizpůsobit dle vlastní libosti.

```
1  var random = new Random();
2  char[] ch = new char[8];
3  string str1 = "abcdefghijklmnopqrstuvwxyz";
4  string str2 = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
5  string str3 = "0123456789";
6  string str4 = "!@*";
7  int l1 = str1.Length;
8  int l3 = str3.Length;
9  int l4 = str4.Length;
10 for (int i = 0; i < 8; i++)
11 {
12     if (i == 0)
13     {
14         ch[i] = str2[random.Next(l1)];
15     }
16
17     if (i > 0 && i <= 4)
18     {
19         ch[i] = str1[random.Next(l1)];
20     }
21
22     if (i > 4 && i <= 6)
23     {
24         ch[i] = str3[random.Next(l3)];
25     }
26
27     if (i == 7)
28     {
29         ch[i] = str4[random.Next(l4)];
30     }
31 }
32 string password = new string(ch);
```

Zdrojový kód 5: Vygenerování výchozího hesla při registraci

V mém projektu byly největší změny provedeny v uživatelském rozhraní – překlad textů do českého jazyka (výchozí jazyk všech stránek je angličtina) a poté

úprava registrace, kdy jsem přidal registraci více uživatelů pomocí `.csv` souboru a přidání automatického vygenerování hesla při registraci jednoho uživatele.

## 4.2 Odesílání e-mailů

Odesílání e-mailových zpráv zajišťuje knihovna *MailKit/MimeKit*. Nejdříve je nutné vytvoření zprávy. V jejích parametrech uvedeme odesílací e-mailovou adresu, jméno, které bude zobrazeno příjemci a předmět zprávy. Poté můžeme připojit i tělo zprávy, které obsahuje samotný text. U textu je důležité uvést, v jakém bude formátu. V mé práci používám u všech e-mailových notifikací formát *HTML*.

```
1 var message = new MimeMessage
2 {
3     Sender = new MailboxAddress("Portál Sokol-Prštné", "martin.
         urbanec01@upol.cz"),
4     Subject = "Potvrzení e-mailové adresy"
5 };
6
7 string text = "<h2>E-mailová adresa není potvrzena</h2>" +
8     "<p>Vaše e-mailová adresa nebyla doposud potvrzena.
         Potvrdit ji můžete v nastavení účtu, kde si zažádáte
           o zaslání potvrzovací zprávy. " +
9     "Učiňte tak prosím co nejdříve.</p>" +
10    " <a href=\"https://portal.sczlin.cz/\">portal.sczlin.cz
        </a><br>" +
11    "<p>Jedná se o beta verzi bakalářské práce. Odesílání e-
        mailů probíhá v rámci testování.</p>";
12
13 message.Body = new TextPart(TextFormat.Html) { Text = text };
14 message.To.Add(new MailboxAddress(email));
```

Zdrojový kód 6: Vytvoření zprávy

Aby mohla být zpráva odeslána, musíme mít připojení k SMTP serveru. Podle nastavení tohoto serveru zadáme jeho adresu, port a typ ověřování. Pokud je ověřování vyžadováno (tj. uvedený typ ověřování je *SSL* nebo *STARTTLS*), uvedeme přihlašovací jméno a heslo k tomuto serveru<sup>6</sup>.

---

<sup>6</sup>SMTP server Univerzity Palackého je používán bez ověřování, tudíž zde nebylo potřeba uvádět přihlášení a tato část kódu je pouze zakomentovaná

```

1 try
2 {
3     using (var smtp = new MailKit.Net.Smtp.SmtpClient())
4     {
5         smtp.MessageSent += (sender, args) => { };
6         smtp.ServerCertificateValidationCallback = (s, c, h, e) => true;
7         smtp.AuthenticationMechanisms.Remove("XOAUTH2");
8
9         await smtp.ConnectAsync("smtps.upol.cz", 25, SecureSocketOptions.
                None).ConfigureAwait(false);
10        //await smtp.AuthenticateAsync("", "").ConfigureAwait(false);
11        await smtp.SendAsync(message).ConfigureAwait(false);
12        await smtp.DisconnectAsync(true).ConfigureAwait(false);
13    }
14 }
15 catch (Exception e)
16 {
17     string m = e.Message;
18 }

```

Zdrojový kód 7: Připojení k SMTP serveru

## 4.3 Popis tříd (modelů)

### 4.3.1 Třída Attendance

Třída umožňující zápis docházky členů na jednotlivé termíny rozvrhových událostí. Pro každý termín rozvrhové události má každý člen svůj záznam.

Nejdůležitější vlastnosti třídy:

- string UserId – identifikátor uživatele
- int EventId – identifikátor rozvrhové události
- int DateId – identifikátor termínu rozvrhové události
- int GroupId – identifikátor skupiny, v níž je daný uživatel členem a ke které je rozvrhová událost navázána
- string TermAttendance – údaj o docházce

### 4.3.2 Třída CalendarData

Třída umožňující výpis dat z databáze pro termínový kalendář.

Nejdůležitější vlastnosti třídy:

- DateTime Date – datum konání soutěže/školení
- string Day – den v týdnu, který odpovídá datu konání soutěže/školení

- `string Discipline` – disciplína, pod níž spadá soutěž/školení
- `string Rules` – pravidla, podle kterých se koná soutěž/školení
- `string Competition` – název soutěže/školení
- `string Remarks` – doplňující informace k soutěži/školení
- `string Category` – kategorie závodníků, pro něž je soutěž určena
- `string Country` – pořadatelská země

### 4.3.3 Třída `DefaultPasswordCheck`

Třída sloužící pro dočasné uložení informace o výchozím hesle nově registrovaného uživatele v databázi. V případě, že zde uživatel má záznam, zobrazuje se mu na úvodní stránce varování o používání výchozího hesla a je doporučena jeho změna. Po změně hesla je záznam odstraněn.

Nejdůležitější vlastnosti třídy:

- `string UserId` – identifikátor uživatele
- `int Changed` – udává stav změny hesla

### 4.3.4 Třída `Event`

Třída sloužící pro ukládání informací o rozvrhových událostech do databáze.

Nejdůležitější vlastnosti třídy:

- `string Name` – název rozvrhové události
- `int GroupId` – identifikátor skupiny, ke které je rozvrhová událost navázána
- `string Type` – typ rozvrhové události
- `string Remarks` – doplňující poznámky k rozvrhové události

### 4.3.5 Třída `Group`

Třída sloužící pro ukládání informací o skupinách do databáze.

Nejdůležitější vlastnosti třídy:

- `int Id` – slouží jako identifikátor skupiny, je přidělován implicitně na databázovém serveru
- `string Name` – název skupiny



### 4.3.6 Třída Request

Třída sloužící pro ukládání a výpis žádostí z databáze.

Nejdůležitější vlastnosti třídy:

- `int Id` – identifikátor žádosti, je přidělován implicitně na databázovém serveru
- `string ProcessorId` – identifikátor uživatele, na kterého je žádost podávána
- `string ApplicantId` – identifikátor uživatele, který žádost podal
- `string RequestText` – text žádosti

### 4.3.7 Třída ScheduledEvent

Třída umožňující ukládání informací o jednotlivých termínech rozvrhových událostí.

Nejdůležitější vlastnosti třídy:

- `int Id` – identifikátor termínu, je přidělován implicitně na databázovém serveru
- `int EventId` – identifikátor rozvrhové události
- `DateTime Date` – datum termínu
- `string StartTime` – čas začátku termínu
- `string EndTime` – předpokládaný čas ukončení

### 4.3.8 Třída UploadedFile

Třída sloužící pro uložení informací o souborech uložených na serveru do databáze. Používána při vytváření termínového kalendáře, kdy jsou `.csv` soubory uloženy na serveru a data z nich mohou být kdykoliv použita pro nahrání do databáze a vytvoření termínového kalendáře.

Nejdůležitější vlastnosti třídy:

- `string Name` – název souboru
- `string ShortName` – zkrácený název souboru (bez vygenerované předpony, která zajišťuje, že nebudou existovat dva soubory se stejným názvem)
- `string Set` – informace, zda jsou data ze souboru momentálně nahrána v databázi termínového kalendáře

### 4.3.9 Třída `UserGroup`

Třída umožňující uložení informace do databáze, kteří uživatelé jsou členy dané skupiny.

Nejdůležitější vlastnosti třídy:

- `string` `UserId` – identifikátor uživatele
- `int` `GroupId` – identifikátor skupiny, v níž je uživatel členem

### 4.3.10 Třída `UserName`

Třída umožňující uložení jména a příjmení uživatelů do databáze, jelikož jméno a příjmení není ukládáno pomocí `Identity`.

Vlastnosti třídy:

- `string` `UserId` – identifikátor uživatele
- `string` `Name` – jméno a příjmení uživatele

### 4.3.11 Třída `Warning`

Třída pro uložení informace o nadměrné absenci uživatele v dané rozvrhové události do databáze. Zajišťuje, aby e-mailová notifikace o nadměrné absenci v dané události byla odeslána pouze jednou.

Nejdůležitější vlastnosti třídy:

- `string` `UserId` – identifikátor uživatele
- `int` `Limit` – hodnota limitu absence, který byl právě překročen (může nabývat hodnot 20 nebo 50)
- `int` `EventId` – identifikátor rozvrhové události, v níž byl překročen limit

Upozornění uživatele a trenéra dané události při dosažení 20% absence znázorňuje následující kód:

```
1 int count1 = await _context.Attendances.Where(p => p.UserId == s.  
    UserId && p.EventId == eventId && p.TermAttendance == "A").  
    CountAsync();  
2 int count2 = await _context.Attendances.Where(p => p.EventId ==  
    eventId && p.UserId == s.UserId).CountAsync();  
3 double result = ((double)count1 / (double)count2) * 100;  
4 result = Math.Round(result, 2);
```

Zdrojový kód 8: Výpočet absence v procentech

```

1  if (result >= 20 && result < 50)
2  {
3    if (!_context.Warnings.Where(p => p.UserId == s.UserId && p.Limit
4      == 20).Any())
5    {
6      await SendEmail(s.UserId, name, 20, 1); //odešle uživateli
7      Warning warning = new Warning
8      {
9        UserId = s.UserId,
10       Limit = 20,
11       EventId = eventId
12     };
13     await _context.Warnings.AddAsync(warning);
14     await _context.SaveChangesAsync();
15     var coaches = from roles in _context.UserRoles
16                   join roleNames in _context.Roles on roles.RoleId
17                   equals roleNames.Id
18                   where roleNames.Name == "Coach"
19                   select new { roleNames.Name, roles.UserId };
20     var users = from c in coaches join sa in SettingAttendances on c.
21                UserId equals sa.UserId select sa;
22     foreach (var v~in users)
23     {
24       await SendEmail(v.UserId, name, 20, 2);
25     }
26   }
27 }

```

Zdrojový kód 9: Příklad, kdy došlo k překročení 20 % absence

## 4.4 Úlohy na pozadí

### 4.4.1 EmailService

Tato služba na pozadí se stará o odesílání e-mailových notifikací na blížící se rozvrhové události, které jsou typu „Školení, Soutěž“ nebo „Obecná událost.“

Po prvním spuštění aplikace na serveru dojde nejdříve k uspání vlákna do nejbližší celé hodiny. Poté již služba každou hodinu projde databází a vyhledá události, které odpovídají zmíněným typům. V případě, že se některá událost bude konat za čtyři dny a hodina jejího startovacího času je rovna aktuálně hodině, dojde k vyhledání všech členů skupiny, ke které je tato událost navázána, a následnému odeslání e-mailové notifikace o blížícím se termínu. Poté je vlákno uspáno na hodinu a poté se celý proces opakuje<sup>7</sup>.

```
1 while (!stoppingToken.IsCancellationRequested)
2 {
3     _logger
4     .LogDebug(
5     "Email service is now running in background."
6     );
7
8     DateTime today = DateTime.Today;
9
10    await CheckScheduledEvents(today);
11
12    await Task.Delay(3600000, stoppingToken);
13 }
```

Zdrojový kód 10: Opakující se činnost služby EmailService

```
1 var events = from se in _context.ScheduledEvents
2               join e in _context.Events on se.EventId equals e.Id
3               where se.Date == today.AddDays(4) && (e.Type == "Školení"
4               || e.Type == "Soutěž" || e.Type == "Obecná událost")
5               select new { se, e};
```

Zdrojový kód 11: Vyhledání událostí konajících se za čtyři dny

---

<sup>7</sup>Pro správnou funkčnost je nutné, aby webový server (např. IIS) běžící aplikaci neukončoval a nechal ji stále spuštěnou.

## 4.5 Napojení na databázi

Následující zdrojový kód ukazuje připojení jednotlivých modelů (tříd, entit) popsaných v předchozí sekci na databázi pomocí rozšíření *Entity Framework Core*.

```
1 public class ApplicationDbContext : IdentityDbContext
2 {
3     public ApplicationDbContext(DbContextOptions<ApplicationDbContext>
4         options)
5         : base(options)
6     {
7
8     }
9     public DbSet<UserName> UserNames { get; set; }
10    public DbSet<Request> Requests { get; set; }
11    public DbSet<UploadedFile> UploadedFiles { get; set; }
12    public DbSet<CalendarData> Calendars { get; set; }
13    public DbSet<MyUser> User { get; set; }
14    public DbSet<UserGroup> UserGroup { get; set; }
15    public DbSet<Group> Groups { get; set; }
16    public DbSet<Event> Events { get; set; }
17    public DbSet<ScheduledEvent> ScheduledEvents { get; set; }
18    public DbSet<Attendance> Attendances { get; set; }
19    public DbSet<DefaultPasswordCheck> DefaultPasswordChecks { get;
20        set; }
21    public DbSet<Warning> Warnings { get; set; }
22 }
```

Zdrojový kód 12: Entity Framework Core DbContext

## 4.6 Poznámky

### 4.6.1 CSS

Pro vytváření vlastních stylů elementů typu checkbox a radio button jsem využil výukových lekcí na webu *W3Schools*. Při optimalizaci vzhledu portálu pro mobilní zařízení mi pomohl framework *Bootstrap* verze 3, jež ale obsahoval jednu chybu u elementu nav třídy *navbar*, který se při určité šířce obrazovky nezobrazil korektně. Zde jsem proto využil řešení ze stránky *Stack Overflow*. Oba zmíněné zdroje jsou včetně odkazů uvedeny v bibliografii a v souboru kaskádových stylů *.css* v projektu jsou tyto části označeny.

### 4.6.2 Ikony

Veškeré ikony použité ve webové aplikaci byly převzaty z webu *Flaticon*. Odkaz včetně popisu je taktéž uveden v bibliografii.

## 5 Uživatelská příručka

Pro správu uživatelů používám výše zmíněný systém ověřování *Identity* a každý uživatel má přiřazeno oprávnění, které mu umožňuje přistupovat pouze k některým funkcím portálu. Uživatelskou příručku jsem se proto rozhodl popsat podle jednotlivých oprávnění a to od role uživatele (nejnižší stupeň oprávnění) až po administrátora (nejvyšší stupeň oprávnění).

Pro využívání jakýchkoliv funkcí portálu je nutné mít vytvořenou registraci. Neregistrovaní uživatelé nemají přístup k žádným funkcím portálu.

### 5.1 Uživatel (User)

#### 5.1.1 Správa uživatelského účtu

- Profil – V této části může uživatel provést změnu e-mailové adresy nebo telefonního čísla. Tato část obsahuje také kolonky s uživatelským jménem a jménem a příjmením, které však není možné upravovat a v případě změny je nutné kontaktovat administrátora<sup>8</sup>.
- Změna hesla – Tato část umožňuje uživateli provést změnu hesla. Toto je doporučeno především po prvním přihlášení, kdy nově registrovaný uživatel používá výchozí heslo vytvořené při registraci. Tato skutečnost je mu připomínána i na úvodní stránce, kdy se mu zobrazuje upozornění na výchozí heslo dokud neprovede jeho změnu. Požadavky na heslo jsou následující – heslo by mělo obsahovat alespoň jedno velké písmeno, alespoň jedno číslo a alespoň jeden speciální znak. Délka hesla musí být nejméně šest znaků.

Jelikož portál zasílá i e-mailové notifikace, je potřeba mít e-mailovou adresu ověřenou. V případě, že uživatel e-mail ověřený nemá, ztrácí možnost být informován např. o nových žádostech či nově přidaných událostech<sup>9</sup>. Výjimku zde tvoří pouze odeslání odkazu pro obnovení zapomenutého hesla, v tomto případě uživatel zprávu obdrží i přesto, že doposud svou e-mailovou adresu nepotvrdil. Pokud uživatel provede změnu e-mailové adresy, je potřeba si nechat opětovně zaslat e-mail s odkazem pro ověření adresy.

Informace o ověřené e-mailové adrese je zobrazena přímo u kolonky „Email“. Pokud bylo ověření úspěšné, je zde zobrazen symbol zelené fajfky. V opačném případě se pod tímto políčkem nachází odkaz „Poslat ověřovací e-mail“.

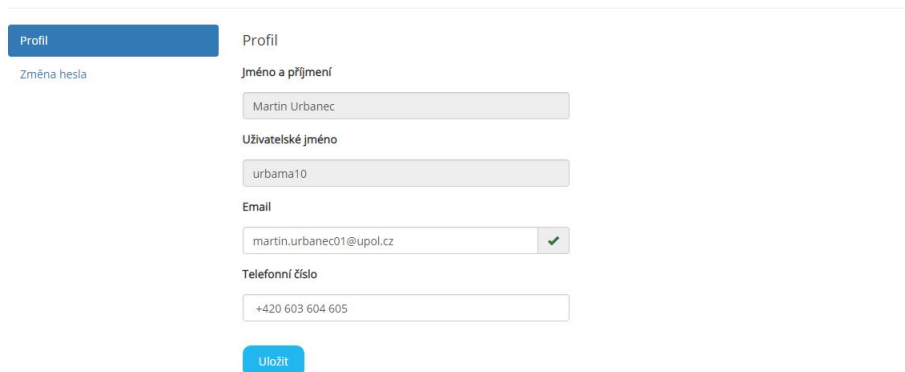
---

<sup>8</sup>Změna jména a příjmení není možná z důvodu, že by si někteří uživatelé nastavili místo jména a příjmení přezdívkou nebo si zvolili zcela jiné jméno. Mohlo by to totiž vést k určitým nedorozuměním.

<sup>9</sup>Zde se opět jedná o určité bezpečnostní opatření, aby portál zbytečně neodesílal notifikace na e-mailové adresy, které nepatří žádnému z členů oddílu a nejednalo se tak o obtěžující e-maily (SPAM).

## SPRÁVA NASTAVENÍ ÚČTU

Změnit nastavení účtu



The screenshot shows a web interface for account settings. On the left, there is a navigation menu with two items: 'Profil' (highlighted in blue) and 'Změna hesla'. The main content area is titled 'Profil' and contains several input fields: 'Jméno a příjmení' with the value 'Martin Urbanec', 'Uživatelské jméno' with 'urbama10', 'Email' with 'martin.urbanec01@upol.cz' (marked with a green checkmark), and 'Telefonní číslo' with '+420 603 604 605'. At the bottom of the form is a blue 'Uložit' button.

Obrázek 6: Nastavení účtu s úspěšně ověřenou e-mailovou adresou

### 5.1.2 Rozvrh a docházka

Rozvrh obsahuje akce, které jsou interní a týkají se pouze oddílu T.J. Sokol Zlín - Prštné (narozdíl od [Termínového kalendáře](#), který obsahuje soutěže a školení naplánované sálovou komisí *Českého svazu cyklistiky*). Každému uživateli je zobrazen jeho individuální rozvrh (samozřejmě mohou existovat uživatelé, kteří mají v rozvrhu společné události) a jednotlivé události jsou děleny podle čtyř typů:

- Trénink
- Soutěž
- Školení
- Obecná událost

Výpis rozvrhových akcí je vytvořen formou odkazů, lze tudíž každou z událostí rozkliknout a prohlédnout si její detaily – poznámky k dané události, uživatele, kteří mají tuto událost rovněž v rozvrhu, nebo zjistit stav docházky na aktuálním termínu. Pokud doposud nebyla trenérem docházka daného termínu zapsána, je možné se na tento termín omluvit (jestliže uživatel provede omluvu omylem, musí požádat trenéra, který má na starost docházku dané události, nebo administrátora, aby stav docházky změnil).

Pokud uživatel potřebuje nahlédnout do minulosti, je možné prohlédnout si i termíny, které již proběhly a to v „Archivu událostí.“

Na úvodní stránce rozvrhu si uživatel může rovněž zvolit statistiku docházky zvolením možnosti „Moje docházka“ a prohlédnout si, jaká je jeho docházka ve všech rozvrhových událostech z jeho rozvrhu bez ohledu na to, zda termíny těchto událostí již proběhly, právě probíhají nebo teprve proběhnou. V této statistice je v přehledné tabulce uvedeno u každé události počet termínů podle jednotlivých stavů docházky:

- Nezapsáno
- Přítomen
- Nepřítomen
- Pozdní příchod
- Dřívější odchod
- Omluven
- Absence v procentech – Absence v procentech je vypočítávána z celkového počtu termínů dané události a z počtu nepřítomných a omluvených termínů. Pokud přesáhne hodnotu 20 %, obdrží uživatel e-mailové upozornění o nadměrné absenci a stejně tak i trenér z téže události. Druhé e-mailové upozornění se odesílá pokud absence přesáhne hodnotu 50 %<sup>10</sup>

Týden 16, Duben 2020

Čt	<b>Test</b> Trénink 10:10 — 11:10 16.04.2020	<b>Trénink Karanténa</b> Trénink 15:00 — 16:00 16.04.2020	
Pá	<b>Testovací událost</b> Školení 12:00 — 12:01 17.04.2020		
So	<b>Testovací událost</b> Školení 12:00 — 12:01 18.04.2020	<b>Test mailu 2</b> Obecná událost 12:00 — 12:01 18.04.2020	<b>Test mailu</b> Obecná událost 16:00 — 17:00 18.04.2020
Ne	<b>Test mailu 3</b> Školení 11:00 — 10:00 19.04.2020	<b>Testovací událost</b> Školení 12:00 — 12:01 19.04.2020	

Obrázek 7: Ukázka výpisu rozvrhových událostí

<sup>10</sup>V případě, že absence přesáhne rovnou hranici 50 %, odesílá se pouze jedno upozornění s informací o přesažení 50% absence.



### 5.1.3 Termínový kalendář

Termínový kalendář obsahuje všechny soutěže a školení z kalendáře vydávaného sálovou komisí *Českého svazu cyklistiky*. Narozdíl od Rozvrhu, kde měl každý uživatel de facto svůj vlastní rozvrh, je Termínový kalendář společný pro všechny uživatele. Ti zde navíc mohou využít možnosti filtrování např. podle jednotlivých disciplín, kategorií, států apod. Soutěže a školení mohou vyhledávat i pomocí názvu<sup>11</sup>.

### 5.1.4 Žádosti

Portál rovněž obsahuje systém na podávání žádostí (např. o proplacení cestovného za jízdu na soutěž vlastním autem, žádost o nový náhradní díl ke kolu, . . .). Žádosti jsou děleny na uživatelem podané a žádosti podané jiným uživatelem na aktuálního uživatele. Pro vytvoření žádosti stačí pouze zvolit uživatele, na kterého bude žádost směřována, a vyplnit text žádosti. O každé nové žádosti je uživatel informován pomocí e-mailové notifikace a stejně tak i o změně stavu žádosti. Žádost se může nacházet ve třech stavech:

- Nepotvrzeno
- Přijato
- Vyřízeno

Pokud je žádost ve stavu „Vyřízeno“, je umožněno uživateli, který je autorem této žádosti, ji smazat.

## 5.2 Trenér (Coach)

Uživatelé s oprávněním „Coach“ mohou využívat veškeré funkce, které jsou přístupné uživatelům s oprávněním „User.“

### 5.2.1 Vytvoření skupiny

Aby bylo možné vytvořit rozvrhovou událost, je nutné nejprve vytvořit skupinu uživatelů, pro které bude tato událost určena. Skupinu je i po vytvoření možné nadále upravovat – přidávat či odebírat členy nebo měnit název (nemohou však existovat dvě skupiny se stejným názvem).

Pokud jsou ke skupině navázány rozvrhové události a dojde k jejímu smazání, pak se automaticky smažou i všechny rozvrhové události navázané na tuto skupinu. Stejně tak pokud je odebrán ze skupiny nějaký člen, pak dojde k jeho odstranění ze všech rozvrhových událostí navázaných na danou skupinu. Opačným případem je přidání člena do skupiny, v tomto případě je člen přidán na

---

<sup>11</sup>Vzhledem k tomu, že tento kalendář obsahuje spoustu soutěží, kterých se neúčastní žádný člen oddílu nebo jsou nominováni pouze někteří členové, je tato sekce oddělena od Rozvrhu na přání vedení klubu T.J. Sokol Zlín - Prštné.

všechny termíny rozvrhových událostí, které doposud neproběhly. Nedojde tedy k případu, že by nově přidaný člen měl nezapsanou docházku na termínu, který již proběhl.

### 5.2.2 Přidání a úprava rozvrhových událostí

Jak bylo řečeno výše, pro vytvoření nové rozvrhové události musí být vytvořena skupina (uživatelů), na kterou bude tato událost navázána. Jedna rozvrhová událost může být navázána pouze na jednu skupinu.

Při vytváření nové události je dobré mít na paměti, že tzv. „koncové datum“, tedy datum posledního termínu nemůže být starší než datum prvního termínu události. Dále název události může být zvolen libovolně a je dovoleno vytvořit více událostí se stejným názvem i pokud sdílejí stejnou skupinu. Rovněž je možné nastavit interval opakování jednotlivých termínů (jednorázový, denní, týdenní a měsíční), ty jsou poté generovány podle zvoleného intervalu od počátečního data až po koncové datum. Pokud by poslední termín přesáhl koncové datum, není tento již zařazen do rozvrhu. Po vytvoření události není možné změnit skupinu.

Data i časy jednotlivých termínů je možné změnit každý zvlášť v náhledu detailu události, zde platí pouze jedno omezení a sice, že nové datum termínu nemůže být starší než dnešní datum (platí pro den, kdy je prováděna editace). Případně je změna data i času možná přímo u náhledu detailů jednotlivých termínů (dostupné z Rozvrhu). Všechny další údaje (vyjma skupiny) lze změnit v editační stránce.

#### DETAILY UDÁLOSTI

##### TRÉNINK V KARANTÉNĚ

Název	Trénink v karanténě
Poznámky	tréninky se z důvodu koronavirové nákazy nebudou konat prezenčně
Skupina	Test mailu
Typ	Trénink

##### PŘEHLED TERMÍNŮ UDÁLOSTI

Datum	Akce
27.04.2020	Změnit datum
04.05.2020	Změnit datum
11.05.2020	Změnit datum
18.05.2020	Změnit datum
25.05.2020	Změnit datum



Obrázek 8: Náhled detailů události

### 5.2.3 Zápis docházky

Zápis docházky je prováděn na jednotlivé termíny rozvrhových událostí. Uživatelé s oprávněním „Coach“, kteří mají daný termín v rozvrhu, mohou provádět tento zápis (administrátor portálu může provádět zápis docházky na libovolný termín jakékoliv rozvrhové události). Docházku je možné zapsat v několika stavech:

- Přítomen
- Nepřítomen
- Pozdní příchod
- Dřívější odchod
- Omluven

Docházku je možné zapsat i částečně a dokončit ji později. Pokud však docházka nebyla na termínu zcela dokončena, zůstává tento termín v rozvrhu i přesto, že již proběhl. Termíny s úspěšně zapsanou docházkou se po proběhnutí přesouvají do „Archivu událostí“, kde je možné docházku i nadále upravovat.

### 5.2.4 Statistika docházky pro trenéry

Statistika docházky pro trenéry obsahuje přehledné tabulky docházky členů na jednotlivých termínech rozvrhových událostí. Aby trenér viděl statistiku pro danou událost, musí mít tuto událost v rozvrhu. Zde poté vidí přehled všech členů, kteří mají rovněž tuto událost v rozvrhu, a jejich docházku, tedy počet nepřítomných termínů, počet pozdních příchodů, atd. Samozřejmě nechybí ani procentuální výpočet absence.

**PŘEDMĚT: TRÉNINK**  
Počet termínů: 8

Jméno a příjmení	Nezapsáno	Přítomen	Nepřítomen	Pozdní příchod	Dřívější odchod	Omluven	Absence
Tereza Buršíková	7	1	0	0	0	0	0 %
Martin Urbanec	7	0	0	1	0	0	0 %
Miloslav Urbanec	7	1	0	0	0	0	0 %
Natálie Vrbová	7	0	1	0	0	0	12,5 %

Obrázek 9: Statistika docházky na rozvrhové události

V případě, kdy je uživatel přidán do skupiny později a termíny některé rozvrhové události přiřazené k této skupině proběhly, tj. tento uživatel je vůbec neměl v rozvrhu, pak není zobrazen u dané rozvrhové události ve statistice docházky.

## 5.3 Administrátor (Administrator)

Uživatel s oprávněním „Administrator“ může využívat veškerých funkcí, které portál nabízí. Při prvotním nastavení dojde k vytvoření uživatelského účtu s názvem „admin“, který má práva administrátora. Tento účet není možné měnit ani odstranit.

### 5.3.1 Vytvoření termínového kalendáře

Termínový kalendář je vytvářen jako soubor *.xlsx*, tudíž je jednoduchý jeho převod do formátu *.csv*. Soubor formátu *.csv* potom může být nahrán na server a jako termínový kalendář nastaven později<sup>12</sup>. Změnu na jinou verzi (ať už starší nebo novější) je možné kdykoliv provést v nastavení termínového kalendáře. U každého souboru je navíc uvedeno i datum jeho nahrání na server a informace, zda je aktuálně nastaven jako termínový kalendář. Pokud není potřeba nadále tento soubor ukládat, může být bez problému smazán.

Důležité u tohoto *.csv* souboru je, že musí mít uvedené záhlaví, které obsahuje následující sloupce a tuto strukturu je nutné zachovat:

- Date
- Day
- Discipline
- Rules
- Competition
- Remarks
- Organizer
- Category
- Country

Pro správné zobrazení výsledků v kalendáři je navíc nutné tento *.csv* soubor vytvořit v kódování *UTF-8*, aby nedocházelo k problémům se zobrazením některých znaků české abecedy.

---

<sup>12</sup>Zkraje roku dochází ještě ke změnám v kalendáři a čeká se na jeho definitivní potvrzení. Proto pokud je vydána nová verze, u které ještě není jisté, zda bude finální, je možné ji pouze nahrát na server a v momentě, kdy bude tato verze potvrzena, můžou být data nahrána do databáze.

### 5.3.2 Registrace uživatelů

Registrovat nové uživatele do portálu má možnost pouze uživatel s oprávněním „Administrator.“ Registraci je možné provést buď přidáním jednoho uživatele nebo využít registraci více uživatelů z *.csv* souboru (zde je opět nutné vytvořit tento soubor v kódování *UTF-8*).

Při registraci nového uživatele je třeba dbát na to, že každý uživatel musí mít své uživatelské jméno a e-mailovou adresu (nemohou tedy existovat dva uživatelé se stejným uživatelským jménem nebo e-mailovou adresou). V případě, že by se tak stalo, vyskočí chybová hláška s příslušnou informací a registrace uživatele neproběhne. Pokud by taková situace nastala při registraci z *.csv* souboru, pak uživatelé, u nichž by došlo k chybě, budou přeskočeni a do portálu registrování nebudou.

Pro registraci z *.csv* je opět potřeba mít v tomto souboru záhlaví, které obsahuje následující sloupce a tuto strukturu je nutné zachovat:

- UserName
- Name
- Email
- Role

Po úspěšné registraci obdrží nově registrovaný uživatel e-mailovou notifikaci na uvedenou e-mailovou adresu, kde bude mít informace k prvnímu přihlášení – uživatelské jméno a heslo vygenerované při registraci – a odkaz pro potvrzení e-mailové adresy.

### 5.3.3 Správa uživatelů

Samotní uživatelé mohou měnit pouze e-mailovou adresu, telefonní číslo a samozřejmě heslo. Změnu uživatelského jména, jména a příjmení nebo oprávnění může provést pouze administrátor portálu. Rovněž může provést i změnu e-mailové adresy i telefonního čísla. Ke správě hesla jednotlivých uživatelů nemá administrátor přístup (o [obnovení zapomenutého hesla](#) bude řeč později).

Změny lze provést přístupem přes detaily jednotlivých uživatelů. Administrátor má rovněž možnost každého uživatele smazat<sup>13</sup>, v tomto případě dojde k odebrání všech jeho žádostí i k odstranění z rozvrhových událostí. Navíc pokud nemá uživatel doposud ověřenou e-mailovou adresu, může mu administrátor zaslat upozornění, aby tak učinil co nejdříve.

---

<sup>13</sup>Zde opět upozorňuji na výjimku uživatelského účtu „admin“, který nemůže být smazán ani editován.

## 5.4 Obnovení hesla

V případě, že uživatel zapomene své heslo, může si jej obnovit. Po zadání e-mailové adresy propojené s jeho uživatelským účtem obdrží na tuto adresu odkaz, který jej přesměruje na vytvoření nového hesla, a poté se již může přihlásit ke svému účtu pod novým heslem.

## Závěr

Úspěšně jsem vytvořil webový portál pro klub T.J. Sokol Zlín – Prštne. Portál umožňuje sledování docházky jednotlivých členů oddílu, vytváření rozvrhových událostí a to jak tréninků, školení, soutěží, tak i obecných událostí (jako je například příprava hrací plochy na nadcházející soutěž, apod.). Rovněž si každý člen může prohlédnout termínový kalendář pro aktuální rok a případně si vyfiltrovat výsledky podle jeho preferencí. Součástí portálu je i vkládání žádostí, kdy si uživatel zvolí na koho je žádost podávána a co po dané osobě požaduje. Web je optimalizován i pro zobrazení na mobilních zařízeních, tudíž všichni registrovaní členové mohou využívat funkci portálu pohodlně kdykoliv a kdekoliv se jim bude chtít.

Funkce portálu mohou být i nadále rozšiřovány dle požadavků vedení klubu. Výhodou této webové aplikace je, že může být použita i pro jiné oddíly v naší republice podobné tomu zlínskému. Na samotný závěr uvádím vyjádření dvou členů vedení klubu T.J Sokol Zlín – Prštne – pana Ing. Jindřicha Mudříka a pana Ing. Miloslava Urbance.

**Ing. Jindřich Mudřík:** „Je zřejmé, že portál byl zpracován se znalostí potřeb, či lépe řečeno, možného využití pro evidenci činností týkajících se života našeho oddílu. Portál, ač původně zamýšlený pro informování členů a evidenci jejich aktivit, je díky podrobnému zpracování a širokému rozčlenění, obecně využitelný i pro jiné účely či činnosti. Po získání větších zkušeností s jeho využitím, by bylo možno s tímto systémem pracovat jako s obecnou platformou pro koordinaci a evidenci činností celé sálové cyklistiky.“

**Ing. Miloslav Urbanec:** „Portál by měl zjednodušit administrativu spojenou s organizací sportovní činnosti v rámci Tělocvičné jednoty Sokol Zlín-Prštne. Díky portálu by mohli trenéři operativněji organizovat tréninkový proces závodníků, jejich účast na všech soutěžích. V současné době se informace (např. rozpis k soutěžím, termínový kalendář) k závodníkům a zákonným zástupcům nezletilých závodníků dostávají především prostřednictvím e-mailové komunikace, která se postupem času stává pro všechny nepřehlednou a při vyhledávání informací časově náročnou.“

## Conclusions

I have successfully created a web portal for T.J. Sokol Zlin – Prstne Club. This portal records attendance of individuals on a team and creates schedule events for training, workshops, competitions and general events (such as preparing the training hall for an upcoming competition etc.). Every team member can also view a term calendar for the current year and filter results according to their preferences. A part of the web portal is also a submission of personal requests for other members of the Club. This website is also optimized for mobile display, so all registered members can use the portal easily whenever and wherever they want.

Several portal functions can be extended according to the needs of the club leaders. The great advantage of this web portal is that it can be used by different teams of Sokol in the Czech Republic. At the very end, I present an opinion of two leaders of T.J. Sokol Zlin – Prstne Club – Mr. Ing. Jindrich Mudrik and Mr. Ing. Miloslav Urbanec.

**Ing. Jindrich Mudrik:** “It is evident that the portal was made with the knowledge of the needs or rather with the knowledge of a possible utilisation for an evidence of activities relating to the life of our club. Although originally, the portal was intended for informing members and recording their activities, it is exploitable in general also for different purposes or functions because of a detailed processing and a wide division. After gaining significant experience with its employment, it would be possible to work with this system as with the general platform for coordination and records of activities of the whole indoor cycling.”

**Ing. Miloslav Urbanec:** “Web portal should make easier the administrative connection with the sport organization of team members from Sokol Zlin-Prstne. Thanks to this portal, trainers can organize the training process of athletes, supervise their attendance at competitions. It would provide and immediate feedback about sport training of athletes for trainers. Currently, the information is sent to athletes and legal guardians of under-aged athletes via e-mail communication, which becomes more and more chaotic and time consuming.”



## A Přehled tříd a jejich vlastností

### A.1 Třída Attendance

Třída umožňující zápis docházky členů na jednotlivé termíny rozvrhových událostí. Pro každý termín rozvrhové události má každý člen svůj záznam.

Vlastnosti třídy:

- `int Id` – slouží jako identifikátor záznamu, je přidělován implicitně na databázovém serveru
- `string UserId` – identifikátor uživatele
- `int EventId` – identifikátor rozvrhové události
- `int DateId` – identifikátor termínu rozvrhové události
- `int GroupId` – identifikátor skupiny, v níž je daný uživatel členem a ke které je rozvrhová událost navázána
- `string TermAttendance` – údaj o docházce

### A.2 Třída CalendarData

Třída umožňující výpis dat z termínového kalendáře.

Vlastnosti třídy:

- `int Id` – slouží jako identifikátor záznamu, je přidělován implicitně na databázovém serveru
- `DateTime Date` – datum konání soutěže/školení
- `string Day` – den v týdnu, který odpovídá datu konání soutěže/školení
- `string Discipline` – disciplína, pod níž spadá soutěž/školení
- `string Rules` – pravidla, podle kterých se koná soutěž/školení
- `string Competition` – název soutěže/školení
- `string Remarks` – doplňující informace k soutěži/školení
- `string Category` – kategorie závodníků, pro něž je soutěž určena
- `string Country` – pořadatelská země

### A.3 Třída `CalendarFilter`

Pomocná třída sloužící pro filtrování dat v termínovém kalendáři.

Vlastnosti třídy:

- `int Day` – den v měsíci konání soutěže/školení
- `int Month` – měsíc konání soutěže/školení
- `string Discipline` – volba disciplíny
- `string Rules` – volba pravidel
- `string Category` – volba kategorie závodníků
- `string Country` – pořadatelská země
- `string Name` – vyhledání podle názvu soutěže/školení

### A.4 Třída `CSV`

Pomocná třída sloužící pro načtení dat z `.csv` souboru při vytváření termínového kalendáře. Při použití knihovny `CSV Helper` je důležité mít na paměti, že názvy jednotlivých vlastností se musejí shodovat s názvy jednotlivých sloupců v `.csv` souboru.

Vlastnosti třídy:

- `public string Date` – datum konání soutěže/školení
- `public string Day` – kalendářní den podle data konání soutěže/školení
- `string Discipline` – disciplína, pod níž spadá soutěž/školení
- `string Rules` – pravidla, podle kterých se koná soutěž/školení
- `string Competition` – název soutěže/školení
- `string Remarks` – doplňující informace k soutěži/školení
- `string Category` – kategorie závodníků, pro něž je soutěž určena
- `string Country` – pořadatelská země

Načtení do databáze za pomoci třídy `CalendarData` ukazuje následující zdrojový kód:

```

1  CalendarData calendar;
2  using (var reader = new StreamReader(path))
3  using (var csvReader = new CsvReader(reader))
4  {
5      var records = csvReader.GetRecords<CSV>();
6
7      _context.UploadedFiles.ToList().ForEach(p => p.Set = "false");
8      _context.UploadedFiles.Where(p => p.Name == name).ToList().ForEach
9          (p => p.Set = "true");
10     _context.SaveChanges();
11
12     foreach (CSV rec in records)
13     {
14         if (rec.Date == "" && rec.Competition == "") continue;
15
16         calendar = new CalendarData
17         {
18             Date = DateTime.Parse(rec.Date),
19             Day = rec.Day,
20             Discipline = rec.Discipline,
21             Rules = rec.Rules,
22             Competition = rec.Competition,
23             Remarks = rec.Remarks,
24             Organizer = rec.Organizer,
25             Category = rec.Category,
26             Country = rec.Country
27         };
28         _context.Calendars.Add(calendar);
29     }
30     _context.SaveChanges();
31 }

```

Zdrojový kód 13: Načtení dat z CSV souboru do databáze

## A.5 Třída DefaultPasswordCheck

Třída sloužící pro dočasné uložení informace o výchozím hesle nově registrovaného uživatele v databázi. V případě, že zde uživatel má záznam, zobrazuje se mu na úvodní stránce varování o používání výchozího hesla, a je doporučena jeho změna. Po změně hesla je záznam odstraněn.

Vlastnosti třídy:

- `string UserId` – identifikátor uživatele
- `int Changed` – udává stav změny hesla

## A.6 Třída Event

Třída sloužící pro ukládání informací o rozvrhových událostech do databáze.

Vlastnosti třídy:

- `int Id` – slouží jako identifikátor události, je přidělován implicitně na databázovém serveru
- `string Name` – název rozvrhové události
- `int GroupId` – identifikátor skupiny, ke které je rozvrhová událost navázána
- `string Type` – typ rozvrhové události
- `string Remarks` – doplňující poznámky k rozvrhové události

## A.7 Třída `EventChange`

Pomocná třída pro uchování vlastností při změně data a času u jednotlivých termínů událostí.

Vlastnosti třídy:

- `DateTime Date` – datum termínu rozvrhové události
- `string StartTime` – začátek termínu rozvrhové události
- `string EndTime` – předpokládaný konec termínu rozvrhové události

## A.8 Třída `Group`

Třída sloužící pro ukládání informací o skupinách do databáze.

Vlastnosti třídy:

- `int Id` – slouží jako identifikátor skupiny, je přidělován implicitně na databázovém serveru
- `string Name` – název skupiny

## A.9 Třída `MyEvent`

Pomocná třída pro výpis informací o rozvrhových událostech. Jsou používány dvě varianty nezávislé na sobě:

1. Seznam rozvrhových událostí. Její vlastnosti jsou:
  - `int Id` – identifikátor rozvrhové události
  - `string Name` – název rozvrhové události
  - `string GroupName` – název skupiny, k níž je událost navázána
  - `string Type` – typ rozvrhové události

- `DateTime StartDate` – datum prvního termínu události
- `DateTime EndDate` – datum posledního termínu události

2. Výpis termínů v Rozvrhu. Vlastnosti v tomto případě jsou:

- `int DateId` – identifikátor termínu
- `DateTime Date` – datum konání termínu
- `string StartTime` – čas začátku termínu
- `string EndTime` – předpokládaný čas ukončení
- `string EventName` – název rozvrhové události
- `bool Completed` – udává, zda byla docházka zapsána všem členům, kteří mají daný termín v rozvrhu
- `string Remarks` – poznámky k rozvrhové události

## A.10 Třída `MyUser`

Pomocná třída sloužící k výpisu informací o uživateli.

Vlastnosti třídy:

- `string Id` – identifikátor uživatele
- `string DefaultUserName` – uživatelské uživatelské jméno (používané pouze při změně uživatelského jména, dochází k porovnání hodnot ve vlastnosti `UserName` a `DefaultUserName`, jinak je hodnota shodná s hodnotou vlastnosti `UserName`)
- `string Name` – jméno a příjmení uživatele
- `string UserName` – uživatelské jméno
- `string Email` – e-mail uživatele
- `string Phone` – telefonní číslo uživatele
- `string Role` – oprávnění uživatele
- `string EmailConfirmed` – informace, zda uživatel potvrdil svou e-mailovou adresu

## A.11 Třída Request

Třída sloužící pro ukládání a výpis žádostí z databáze.

Vlastnosti třídy:

- `int Id` – identifikátor žádosti, je přidělován implicitně na databázovém serveru
- `string ProcessorId` – identifikátor uživatele, na kterého je žádost podávána
- `string ApplicantId` – identifikátor uživatele, který žádost podal
- `string RequestText` – text žádosti
- `int State` – stav žádosti
- `string Date` – datum podání žádosti

## A.12 Třída ScheduledEvent

Třída umožňující ukládání informací o jednotlivých termínech rozvrhových událostí.

Vlastnosti třídy:

- `int Id` – identifikátor termínu, je přidělován implicitně na databázovém serveru
- `int EventId` – identifikátor rozvrhové události
- `DateTime Date` – datum termínu
- `string StartTime` – čas začátku termínu
- `string EndTime` – předpokládaný čas ukončení

## A.13 Třída UploadedFile

Třída sloužící pro uložení informací o souborech uložených na serveru do databáze. Používaná při vytváření termínového kalendáře, kdy jsou `.csv` soubory uloženy na serveru a data z nich mohou být kdykoliv použita pro nahrání do databáze a vytvoření termínového kalendáře.

Vlastnosti třídy:

- `int Id` – identifikátor souboru, je přidělován implicitně na databázovém serveru
- `string Name` – název souboru

- `string ShortName` – zkrácený název souboru (bez vygenerované předpony, která zajišťuje, že nebudou existovat dva soubory se stejným názvem)
- `DateTime Time` – datum nahrání souboru na server
- `string Set` – informace, zda jsou data ze souboru momentálně nahrána v databázi termínového kalendáře

## A.14 Třída `UserFilter`

Pomocná třída sloužící k vyhledávání uživatelů podle jména a jejich řazení podle abecedy v seznamu registrovaných uživatelů.

Vlastnosti třídy:

- `string Name` – jméno uživatele
- `string Order` – udává, zda bude seznam uživatelů řazen podle abecedy od A do Z nebo od Z do A

## A.15 Třída `UserGroup`

Třída umožňující uložení informace do databáze, kteří uživatelé jsou členy dané skupiny.

Vlastnosti třídy:

- `int Id` – identifikátor záznamu, je přidělován implicitně na databázovém serveru
- `string UserId` – identifikátor uživatele
- `int GroupId` – identifikátor skupiny, v níž je uživatel členem

## A.16 Třída `UserName`

Třída umožňující uložení jména a příjmení uživatelů do databáze, jelikož jméno a příjmení není ukládáno pomocí `Identity`.

Vlastnosti třídy:

- `string UserId` – identifikátor uživatele
- `string Name` – jméno a příjmení uživatele

## A.17 Třída Warning

Třída pro uložení informace o nadměrné absenci uživatele v dané rozvrhové události do databáze. Zajišťuje, aby e-mailová notifikace o nadměrné absenci v dané události byla odeslána pouze jednou.

Vlastnosti třídy:

- `int Id` – identifikátor záznamu, je přidělován implicitně na databázovém serveru
- `string UserId` – identifikátor uživatele
- `int Limit` – hodnota limitu absence, který byl právě překročen (může nabývat hodnot 20 nebo 50)
- `int EventId` – identifikátor rozvrhové události, v níž byl překročen limit



## B Obsah příloženého CD

Zde je uveden popis struktury obsahu příloženého CD.

### **bin/**

Kompletní adresářová struktura webové aplikace PORTÁL SOKOL-PRŠTNÉ (v ZIP archivu) pro zkopírování na webový server.

### **doc/**

Text práce ve formátu PDF vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce včetně všech příloh a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

### **src/**

Kompletní zdrojové texty webové aplikace PORTÁL SOKOL-PRŠTNÉ se všemi potřebnými (příp. převzatými) zdrojovými texty, knihovny a dalšími soubory potřebnými pro bezproblémové vytvoření adresářové struktury pro zkopírování na webový server nebo pro spuštění webové aplikace na lokálním PC.

### **readme.txt**

Instrukce pro nasazení webové aplikace PORTÁL SOKOL-PRŠTNÉ na webový server, včetně všech požadavků pro její bezproblémový provoz, a webová adresa na které je aplikace nasazena pro účel testování při tvorbě práce a pro účel obhajoby práce.

Navíc CD obsahuje:

### **data/**

Ukázková data použitá v práci a pro potřeby testování práce při tvorbě posudků a obhajoby práce.

### **install/**

Instalátory aplikací, runtime knihoven a jiných souborů potřebných pro provoz webové aplikace PORTÁL SOKOL-PRŠTNÉ, které nejsou standardní součástí operačního systému určeného pro provoz webové aplikace.

U veškerých cizích převzatých materiálů obsažených na CD jejich zahrnutí dovolují podmínky pro jejich šíření nebo příložený souhlas držitele copyrightu. Pro všechny použité (a citované) materiály, u kterých toto není splněno a nejsou tak obsaženy na CD, je uveden jejich zdroj (např. webová adresa) v bibliografii nebo textu práce nebo v souboru `readme.txt`.



## Bibliografie

- [1] BAKALÁŘI, Co všechno umíme?. Bakaláři – mezi školou a rodinou [online]. [cit. 2020-05-11]. Dostupné z: <https://www.bakalari.cz/Static/WhatWeKnow>
- [2] WIKIPEDIA, C#. Wikipedia [online]. 2019-06-22. [cit. 2019-12-23]. Dostupné z: [https://cs.wikipedia.org/wiki/C\\_Sharp](https://cs.wikipedia.org/wiki/C_Sharp)
- [3] ENTITY FRAMEWORK CORE, Entity Framework Core Documentation and Tutorials. Learn Entity Framework Core [online]. [cit. 2019-12-23]. Dostupné z: <https://www.learnentityframeworkcore.com/>
- [4] STACK OVERFLOW, Bootstrap 3 Navbar Collapse. Stack Overflow [online]. 2013-11-27. [cit. 2020-05-11]. Dostupné z: <https://stackoverflow.com/questions/18192082/bootstrap-3-navbar-collapse>
- [5] W3SCHOOLS.COM, How To Create a Custom Checkbox and Radio Buttons. W3Schools [online]. [cit. 2020-05-11]. Dostupné z: [https://www.w3schools.com/howto/howto\\_css\\_custom\\_checkbox.asp](https://www.w3schools.com/howto/howto_css_custom_checkbox.asp)
- [6] FLATICON, Icon made by Pixel perfect from www.flaticon.com. Flaticon [online]. [cit. 2020-05-11]. Dostupné z: <https://www.flaticon.com/>
- [7] FLATICON, Icon made by Freepik from www.flaticon.com. Flaticon [online]. [cit. 2020-05-11]. Dostupné z: <https://www.flaticon.com/>