

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačního inženýrství**



**Bakalářská práce**

**Návrh a implementace webové aplikace pro skautský  
oddíl**

**Alžběta Pokorná**

**© 2020 ČZU v Praze**

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Alžběta Pokorná

Systémové inženýrství a informatika  
Informatika

Název práce

**Návrh a implementace webové aplikace pro skautský oddíl**

Název anglicky

**Design and implementation of web application for scout troop**

---

### Cíle práce

Cílem práce je analýza a návrh webové aplikace pro skautský oddíl a následně její implementace. Tato webová aplikace bude sloužit jako informační portál pro rodiče členů oddílu a širší veřejnost. Zároveň bude využívána pro přehled o členech oddílu a jako prostředek pro snazší organizování jeho akcí.

### Metodika

Práce se sestává z teoretické a praktické části.

Metodika zpracování teoretické části práce spočívá ve studiu odborných informačních zdrojů. Na základě zjištěných poznatků budou formulována teoretická východiska pro zpracování praktické části práce.

Praktická část práce bude zahrnovat analýzu požadavků, návrh a implementaci webové aplikace. Při analýze a návrhu bude použito standardních metod a nástrojů softwarového inženýrství. Implementace bude provedena v ASP.NET s využitím vývojového prostředí MS Visual Studio. Aplikace bude prakticky nasazena a otestována, budou shrnuty poznatky získané při její tvorbě a navrženy možnosti jejího případného budoucího rozvoje.

## Doporučený rozsah práce

30-40 stran

## Klíčová slova

skautský oddíl, webová aplikace, C#, Entity Framework, PostgreSQL

---

## Doporučené zdroje informací

ALBAHARI, J. – ALBAHARI, B. C# 7.0 in a Nutshell. Sebastopol: O'Reilly Media, 2017. ISBN 9781491987650.

Dokumentace jazyka C#, dostupné z:

<https://docs.microsoft.com/cs-cz/dotnet/csharp/language-reference/>

Dokumentace k ASP.NET, dostupné z: <https://docs.microsoft.com/cs-cz/aspnet/>

LERMAN, J. Programming Entity Framework. Sebastopol: O'Reilly Media, 2009. ISBN 059652028x.

---

## Předběžný termín obhajoby

2019/20 LS – PEF

## Vedoucí práce

Ing. Jiří Brožek, Ph.D.

## Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 19. 2. 2020

**Ing. Martin Pelikán, Ph.D.**

Vedoucí katedry

Elektronicky schváleno dne 19. 2. 2020

**Ing. Martin Pelikán, Ph.D.**

Děkan

V Praze dne 18. 03. 2020

### **Čestné prohlášení**

Prohlašuji, že svou bakalářskou práci "Návrh a implementace webové aplikace pro skautský oddíl" jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autorka uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušila autorská práva třetích osob.

V Praze dne 18. 3. 2020

---

### **Poděkování**

Ráda bych touto cestou poděkovala Ing. Jiřímu Brožkovi, Ph.D. za vstřícnost a ochotu při konzultacích práce a za poskytnuté cenné rady.

# Návrh a implementace webové aplikace pro skautský oddíl

## Abstrakt

Tato bakalářská práce se zabývá analýzou, návrhem a implementací webové aplikace vyvíjené s využitím ASP.NET.

Teoretické část práce obsahuje popis fungování webových aplikací, vymezení a přiblížení termínu MVC architektura a dalších návrhových vzorů, které s touto architekturou souvisejí. Dále jsou popsány frameworky ASP.NET a ASP.NET Core, Entity Framework, databáze PostgreSQL, a další technické prostředky pro implementaci webových aplikací.

V praktické části je uvedena analýza požadavků, analýza existujících řešení a návrh databáze. Samotná webová aplikace je navržena s pomocí UI specifikace a je popsána její následná implementace. Aplikace je dále nasazena, otestována a zhodnocena.

**Klíčová slova:** skautský oddíl, webová aplikace, C#, Entity Framework, PostgreSQL

# **Design and implementation of web application for scout troop**

## **Abstract**

The main goal of this bachelor thesis is to analyse, propose and implement a web application by using ASP.NET technology.

The theoretical part describes how web applications work, unfolds the MVC architecture and other design patterns somehow related to the former. In addition, other frameworks such as ASP.NET and ASP.NET Core, Entity Framework and PostgreSQL database are equally involved in attention, as well as other technologies of web application development.

The implementation part focuses on analysis of the requirements, analysis of already existing solutions and a design of a database model. A design of the application by using UI methodology follows, as well as the implementation. Finally, the application is deployed, tested and put into evaluation.

**Keywords:** scout troop, web application, C#, Entity Framework, PostgreSQL

# Obsah

<b>1 Úvod</b> .....	<b>14</b>
<b>2 Cíl práce a metodika</b> .....	<b>15</b>
2.1 Cíle práce .....	15
2.2 Metodika .....	15
<b>3 Teoretická východiska</b> .....	<b>16</b>
3.1 Webová aplikace .....	16
3.1.1 Webový server .....	16
3.1.2 Aplikační server .....	16
3.1.3 Statické a dynamické stránky .....	16
3.1.4 Zpracování dynamických stránek .....	17
3.2 MVC architektura.....	17
3.2.1 Způsob fungování MVC architektury .....	17
3.2.2 ViewModel .....	18
3.2.3 Repository pattern.....	18
3.2.3.1 Unit of Work pattern .....	19
3.2.4 Dependency Injection .....	19
3.3 ASP.NET.....	19
3.3.1 ASP.NET Core.....	19
3.3.1.1 ASP.NET Core Identity .....	20
3.3.2 NuGet.....	20
3.3.3 Scaffolding.....	20
3.3.4 ASP.NET MVC .....	20
3.3.4.1 Konvence .....	21
3.3.5 .NET 5.....	21
3.4 PostgreSQL .....	21
3.4.1 Dědičnost .....	21
3.4.2 MVCC.....	21
3.4.2.1 MVCC v PostgreSQL.....	22
3.5 Entity Framework.....	22
3.5.1 Objektově relační mapování .....	22
3.5.2 Entity Framework Core .....	22
3.5.3 Entity Data Model.....	23
3.5.4 Možnosti přístupu .....	23
3.5.4.1 Migrace.....	25
3.5.4.2 Vytvoření modelu v závislosti na přístupu .....	25



3.5.5	DbContext.....	25
3.5.5.1	Metody nad DbContext .....	25
3.5.6	DbSet .....	26
3.5.7	Práce s databázemi v ASP.NET Core MVC.....	26
3.6	C#.....	26
3.6.1	Datové typy.....	26
3.6.2	Atributy .....	27
3.6.3	Validace v C#.....	27
3.6.4	LINQ.....	27
3.6.4.1	LINQ to Objects .....	27
3.6.4.2	LINQ to Entities .....	28
3.6.4.3	Syntaxové metody .....	28
3.7	Razor .....	28
3.7.1	HTML .....	29
3.7.2	Práce s Razorem s použitím C# .....	29
3.7.2.1	Direktivy Razoru .....	29
3.7.2.2	Razor Helpers .....	30
3.8	Hosting webových aplikací.....	30
3.8.1	Typy webového hostingu.....	30
3.8.2	Platform as a Service .....	31
3.8.2.1	Amazon Web Services .....	31
3.8.2.2	Azure App Service .....	31
3.8.2.3	Heroku .....	32
3.8.2.4	Google Cloud Platform.....	32
<b>4</b>	<b>Vlastní práce.....</b>	<b>33</b>
4.1	Analýza požadavků .....	33
4.1.1	Funkční požadavky .....	33
4.1.1.1	F1 – Uživatelé.....	33
4.1.1.2	F2 – Přihlášení .....	33
4.1.1.3	F3 – Odhlášení.....	34
4.1.1.4	F4 – Registrace .....	34
4.1.1.5	F5 – Přidávání členů do správy rodičům.....	34
4.1.1.6	F6 – Změna hesla.....	34
4.1.1.7	F7 – Obnova hesla .....	34
4.1.1.8	F8 – Informační portál.....	34

4.1.1.9	F9 – Aktuality .....	35
4.1.1.10	F10 – Družiny .....	35
4.1.1.11	F11 – Seznam akcí.....	35
4.1.1.12	F12 – Přihlašování na akce.....	35
4.1.1.13	F13 – Členové a uživatelé, kteří jedou na akci.....	36
4.1.1.14	F14 – Zobrazení uživatelského profilu.....	36
4.1.1.15	F15 – Archiv akcí .....	36
4.1.1.16	F17 – Informace o uživateli a členech.....	36
4.1.2	Nefunkční požadavky .....	36
4.1.2.1	N1 – Implementace.....	36
4.1.2.2	N2 – Vzhled.....	36
4.1.2.3	N3 – Cena .....	36
4.2	Analýza dostupných řešení .....	37
4.2.1	Skautský oddíl Krnovská Trojka .....	37
4.2.2	Jeleni – 175. skautský oddíl.....	38
4.2.3	7. dívčí skautský oddíl, středisko Pětilisté Růže, Jindřichův Hradec .....	38
4.2.4	Závěr analýzy.....	39
4.3	Technické prostředky .....	39
4.4	Návrh databáze.....	40
4.4.1	Datové typy.....	41
4.4.2	Tabulky a vazby mezi nimi.....	41
4.5	UI specifikace.....	42
4.5.1	Use Case Diagram .....	42
4.5.2	Tabulka pokrytí funkčních požadavků .....	47
4.5.3	Seznam akcí .....	48
4.5.3.1	Use Case .....	48
4.5.3.2	Scénář .....	48
4.5.3.3	Wireframe.....	50
4.5.4	Moje družina .....	50
4.5.4.1	Use Case .....	50
4.5.4.2	Scénář .....	50
4.5.4.3	Wireframe.....	51
4.6	Implementace .....	51
4.6.1	Seznam akcí .....	52
4.6.1.1	Model.....	52
4.6.1.2	Repozitář.....	53

4.6.1.3	ViewModel .....	54
4.6.1.4	Controller.....	54
4.6.1.5	View .....	56
4.6.1.6	Výsledná obrazovka .....	59
4.6.2	Moje družina .....	60
4.6.2.1	Model.....	60
4.6.2.2	Repozitář.....	63
4.6.2.3	ViewModel .....	64
4.6.2.4	Controller.....	65
4.6.2.5	View .....	68
4.6.2.6	Výsledná obrazovka .....	70
<b>5</b>	<b>Výsledky a diskuse .....</b>	<b>72</b>
5.1	Testování webové aplikace .....	72
5.2	Zhodnocení ze strany autora .....	72
5.2.1	Návrhy na vylepšení .....	73
<b>6</b>	<b>Závěr.....</b>	<b>74</b>
<b>7</b>	<b>Seznam použitých zdrojů .....</b>	<b>75</b>
<b>8</b>	<b>Přílohy .....</b>	<b>83</b>

## Seznam obrázků

Obrázek 1:	Způsob fungování MVC architektury .....	18
Obrázek 2:	EF – Database First přístup .....	23
Obrázek 3:	EF – Model First přístup .....	24
Obrázek 4:	EF – Code First přístup .....	24
Obrázek 5:	Webové stránky oddílu Krnovská Trojka .....	37
Obrázek 6:	Webové stránky oddílu Jeleni .....	38
Obrázek 7:	Webové stránky 7. dívčího skautského oddílu.....	39
Obrázek 8:	Návrh databáze.....	41
Obrázek 9:	Use Case Diagram část 1.....	43
Obrázek 10:	Use Case Diagram část 2.....	44
Obrázek 11:	Use Case Diagram část 3.....	45
Obrázek 12:	Use Case Diagram část 4.....	46
Obrázek 13:	Use Case Diagram část 5.....	47

Obrázek 14: Výsledná obrazovka seznamu akcí .....	60
Obrázek 15: Výsledná obrazovka informací o družině .....	71

## Seznam ukázek kódu

Kód 1: Možnosti zápisu LINQ.....	28
Kód 2: Ukázka Razoru.....	30
Kód 3: Model „Akce“ .....	53
Kód 4: Repozitář „AkceRepository“ – Metoda „VsechnyAkce“ .....	54
Kód 5: ViewModel „AkceViewModel“ .....	54
Kód 6: Metoda „SeznamAkci“ .....	55
Kód 7: Metoda „SeznamAkci“ POST.....	56
Kód 8: View „SeznamAkci“ – Úvodní deklarace.....	57
Kód 9: View „SeznamAkci“ – Vyhledávací formulář.....	58
Kód 10: View „SeznamAkci“ – Odkaz pro přidání akce .....	58
Kód 11: View „SeznamAkci“ – Jednotlivé akce .....	59
Kód 12: View „SeznamAkci“ – JavaScript .....	59
Kód 13: Model „Druzina“ .....	61
Kód 14: Model „Uzivatel“ .....	62
Kód 15: Model „Clen“ .....	63
Kód 16: Repozitář „AkceRepository“ – Metoda „AkcePodleId“ .....	64
Kód 17: Repozitář „ClenRepository“ .....	64
Kód 18: ViewModel „MojeDruzinaViewModel“ .....	65
Kód 19: Metoda „DetailDruziny“ – Ošetření existence družiny .....	66
Kód 20: Metoda „DetailDruziny“ – Zjištění konkrétní družiny .....	66
Kód 21: Metoda „DetailDruziny“ – Naplnění listů.....	67
Kód 22: Metoda „DetailAkce“ POST .....	68
Kód 23: Metoda „EmailVedouci“ .....	68
Kód 24: View „MojeDruzina“ – Popis družiny.....	69
Kód 25: View „MojeDruzina“ – Seznam akcí pro družinu .....	69
Kód 26: View MojeDruzina – Odeslání e-mailu .....	70

## Seznam tabulek

Tabulka 1: Pokrytí funkčních požadavků .....	48
--	----

## Seznam Wireframů

Wireframe 1: Seznam akcí.....	50
Wireframe 2: Moje družina.....	51

## Seznam použitých zkratk

AWS	Amazon Web Services
DI	Dependency Injection
EDM	Entity Data Model
EF	Entity Framework
GCP	Google Cloud Platform
MVC	Model-View-Controller
MVCC	Multi Version Concurrency Control
ORM	Object Relational Mapping
PaaS	Platform as a Service

# 1 Úvod

Skauting je velmi oblíbenou možností trávení volného času, jak pro děti, které se takto mohou dostávat častěji do příroda, získávat nové zkušenosti a potkávat další vrstevníky, tak i pro vedoucí jednotlivých oddílů. Právě ti se této aktivitě věnují ve svém volném čase, nejčastěji při studiu, a kromě zábavy jim přináší i velké množství administrativní práce, komunikace s rodiči a organizování. Vedoucí, ač ze stejného oddílu, nemusí ani pobývat ve stejných částech republiky, což celkovou komunikaci ještě více ztěžuje.

Pro potřeby nabírání nových členů a informování veřejnosti o aktivitách oddílu, je nutné, aby každý oddíl vlastnil webové stránky.

Žijeme v době masivního používání informačních technologií, které nám můžou napomáhat v každodenních záležitostech. Tato bakalářská práce má za cíl právě díky informačním technologiím zlepšit fungování jednoho konkrétního skautského oddílu, a to pomocí webové aplikace, která bude pro tento oddíl sloužit jako komunikační, organizační a informační nástroj.

## **2 Cíl práce a metodika**

### **2.1 Cíle práce**

Hlavním cílem této bakalářské práce je analýza, návrh a implementace webové aplikace s využitím ASP.NET, která byla zadána konkrétním skautským oddílem. Aplikace bude sloužit jako informační portál pro veřejnost a případné zájemce o členství v oddílu. Dále bude fungovat jako komunikační kanál, organizační nástroj sloužící k evidenci členů oddílu, akcí, které oddíl pořádá, a také ke sledování účasti jednotlivých členů na těchto akcích.

Analýza se bude skládat z rozboru požadavků oddílu na konkrétní funkcionality aplikace, a také z prozkoumání již existujících webových aplikací jiných skautských oddílů.

Webová aplikace bude nasazena, otestována, zhodnocena a budou vytvořeny návrhy na její další vylepšení a rozšiřování.

Dílčím cílem práce je získání potřebných teoretických znalostí pro samotnou implementaci aplikace.

### **2.2 Metodika**

Teoretická část práce se bude zabývat jednotlivými technologiemi potřebnými k vytvoření zadané webové aplikace a obecným popisem fungování webových aplikací.

Praktická část bude zahrnovat analýzu požadavků zadaných klientem, rozbor dostupných řešení, návrh aplikace a následně samotnou implementaci, která bude popsána na konkrétních funkcích aplikace pomocí ukázek kódu a popisu. Implementace bude provedena za pomoci znalostí získaných z teoretické části práce.

V praktické části bude provedeno i následné nasazení aplikace, její otestování a budou sepsány získané poznatky a návrhy na rozšíření této webové aplikace.

## **3 Teoretická východiska**

### **3.1 Webová aplikace**

Webová aplikace je softwarový program, který je provozován pomocí webového serveru a spouštěn prostřednictvím webového prohlížeče. Toto řešení umožňuje práci s aplikací bez nutnosti instalace. Oproti běžným desktopovým programům má tu výhodu, že je multiplatformní, tedy spustitelná v kterémkoliv operačním systému, bez nutnosti vývojářů vytvářet více verzí téhož programu. (1) Pro práci s webovými aplikacemi je požadován přístup k internetu. (2)

#### **3.1.1 Webový server**

Webový server můžeme chápat jako hardware, software, nebo kombinaci obou. Z hlediska webového serveru jakožto hardwaru se jedná o počítač, kde jsou ukládána data webových stránek. Software webového serveru pak slouží k určování přístupu uživatelů k datům uloženým na serveru.

Pokud uživatel chce pracovat s webovou stránkou, je na server pomocí internetového prohlížeče poslána žádost o její zobrazení. (3) Stránka ze serveru je pak buď odeslána v neměnné formě, nebo je nejdříve vygenerována pomocí spuštění skriptů a získáním dat z databáze. (4)

Samotný webový server je ale určen pouze pro práci se statickými stránkami. (5)

#### **3.1.2 Aplikační server**

Aplikační server je software, který umožňuje pracovat i s dynamickým obsahem stránek. (5) Díky němu je také možné pracovat ve webových aplikacích s databázemi. Pokud stránka požaduje zobrazení dat z databáze, aplikační server požádá o tato data databázový ovladač, který je ze samotné databáze získá. Aplikační server pak tato data vloží do stránky. (6)

#### **3.1.3 Statické a dynamické stránky**

Statická webová stránka je taková stránka, která nemění svůj obsah. Zobrazuje se tedy všem klientům stejně. Kvůli své stálé formě v nich nelze pracovat například s daty z databáze či formulářů. Využívá se tedy spíše k jednodušším webovým prezentacím. (7) Webový server, který pracuje s takovými stránkami se nazývá statický webový server. (3)



Oproti tomu dynamické webové stránky umožňují měnit svůj obsah a jsou dotvořeny až v závislosti na konkrétních požadavcích klienta na server.

Webová aplikace je pak množinou statických a dynamických stránek. (6)

### 3.1.4 Zpracování dynamických stránek

Ve chvíli, kdy klient požádá o zobrazení dynamické stránky, je potřeba, aby byla převedena do podoby zobrazitelné klientem.

Požadavek klienta zachytí webový server. Ten předá tento požadavek aplikačnímu serveru pro zpracování. Aplikační server ho poté vyhodnotí a převede dynamickou stránku do statické podoby, kterou již klient zvládne zobrazit. (6)

## 3.2 MVC architektura

Model-View-Controller (MVC) architektura je softwarový návrhový vzor používaný při vývoji webových aplikací. (8) Cílem této architektury je rozdělit kód aplikace do tří samostatných logických částí. To umožňuje úpravu některé z částí aplikace s minimálním dopadem na zbylé logické celky. Tyto tři logické části se nazývají Model, View a Controller. (9)

- **Model** je část aplikace, která je určena ke správě dat. (8) Slouží k popisu samotných dat, ale i k popisu pravidel, jak s těmito daty pracovat. (10)
- **View** je určen pro reprezentaci dat uživateli v určitém formátu. (8)
- **Controller** má na starosti zpracování požadavků uživatele. (8)

MVC je součástí mnoha frameworků pro tvorbu webových aplikací, jako jsou frameworky pro PHP (Nette, Zend), Ruby on Rails pro jazyk Ruby, či ASP .NET MVC pro ASP .Net (11), který je více popsán v následujících kapitolách.

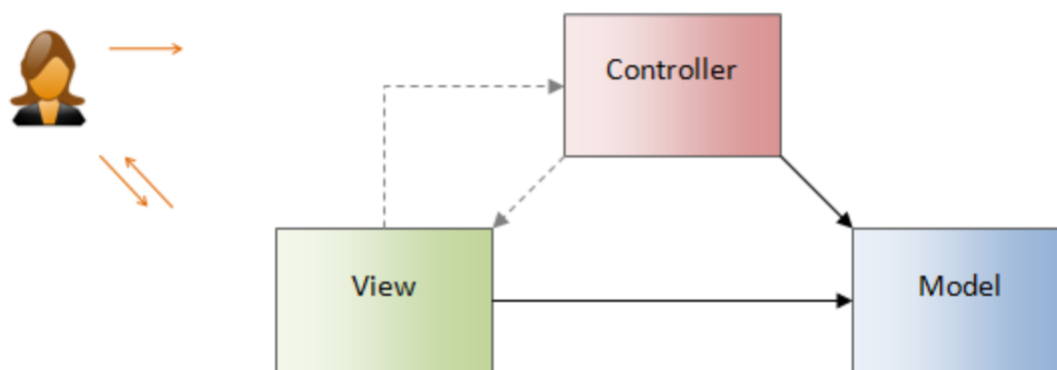
MVC je využíván především kvůli umožnění rychlejšího vývoje webových aplikací, snadnější aktualizaci kódu těchto aplikací, jednoduššímu ladění (12) a již zmíněným možnostem úprav jednotlivých logických celků bez narušení funkčnosti aplikace. (9)

### 3.2.1 Způsob fungování MVC architektury

Významnou roli ve fungování MVC hraje router. Ten je schopný ze zadané adresy webové aplikace a přiložených parametrů určit, kterého Controlleru se zpracování tohoto požadavku týká. Tento Controller pak zavolá potřebný Model, který obsahuje metody ke

získání dat z databáze a práci s nimi. Nakonec Controller vykreslí View, který obsahuje takto získaná data, a je zobrazován uživateli. (11) Interakce uživatele a celková funkcionality MVC je znázorněna na obrázku (Obrázek 1).

Obrázek 1: Způsob fungování MVC architektury



Zdroj: (9)

### 3.2.2 ViewModel

ViewModel v MVC je třída velmi podobná Modelu. Zatímco Model obsahuje nejen data zobrazitelná ve View, ale i logiku pro práci s nimi a s databází, ViewModel slouží pouze pro vykreslování View. Používá se ve chvíli, kdy je potřeba vykreslovat data z více tabulek a kdy není potřeba využít všechny atributy Modelu, ale naopak by byl jejich přenos do View zbytečnou datovou zátěží. (13)

### 3.2.3 Repository pattern

Repository pattern je návrhový vzor oddělující vrstvu přístupu k datům a vrstvu zabývající se logikou, která s těmito daty pracuje. Logika přístupu k datům je zde řešena pomocí tzv. repozitářů. (14) Repozitář je třída, obsahující metody pro práci s jednou konkrétní databázovou tabulkou. Dalším možným přístupem je práce s jedním generickým repozitářem. (15)

Při využití Repository pattern Controllery ve chvíli, kdy potřebují získat data pro vytvoření View, pracují pouze se samotnými repozitáři. (14) Tímto způsobem je s daty pracováno nepřímě, což zamezuje vzniku duplicitního kódu přistupujícího k databázím a ulehčuje jeho testování. Přístup k databázím je navíc centrálně spravován. (16)

### 3.2.3.1 Unit of Work pattern

Změny nad databází se provádějí v transakcích. Jedna akce (například registrace nového uživatele) může zapříčinit změnu na více tabulkách. Kdyby se pro změnu v každé tabulce vytvořila nová transakce, a následně pokus o změnu dat v jedné tabulce selhal a v druhé proběhl správně, vedlo by to k nekonzistenci dat. Proto je potřeba, aby všechny tyto změny byly zahrnuty pouze v jedné transakci. K tomu slouží třída Unit of Work. Ta vytváří další vrstvu mezi Controllery a repozitáři a je volána při potřebě uložení změn na databázi. Protože uložení provádí pouze tato třída, a ne každý repozitář zvlášť, je vytvořena pouze jedna transakce. (15)

### 3.2.4 Dependency Injection

Dependency Injection (DI) je návrhový vzor pro objektově orientované jazyky. Umožňuje používání závislostí (tj. objektů, které potřebují pro práci objekty jiné (17)). (18) Díky Dependency Injection v sobě třídy nemusí obsahovat kód pro získání cizích objektů, ale závislosti jsou jim předávány v abstraktní formě například v konstruktoru třídy (19), nebo pomocí vlastností. Největší výhodou DI je zajištění nízké provázanosti objektů v systému (18).

Využití tohoto návrhového vzoru může být například v Controlleru, kde při potřebě přístupu k databázi je nezbytné pracovat s repozitáři. Pomocí DI je možné získat dané repozitáře bez ručního vytváření. (20)

## 3.3 ASP.NET

ASP.NET je framework vytvořený firmou Microsoft, který slouží k tvorbě webových aplikací. Je založený na platformě .NET, která poskytuje nejrůznější nástroje, knihovny a programovací jazyky pro vývoj. ASP.NET tuto platformu rozšiřuje například o autentizační systém, syntaxi pro tvorbu dynamického webu s pomocí C#, označovanou jako Razor, a obecně o funkce pro práci s webem. Backend kód webové aplikace je v ASP.NET psaný v jazyce C#. (21)

### 3.3.1 ASP.NET Core

V současnosti je pro vývoj nových aplikací doporučováno firmou Microsoft využívat open-source multiplatformní verzi frameworku ASP.NET Core. Inovace pro framework vznikají převážně již jen pro tuto verzi. (21)

### 3.3.1.1 ASP.NET Core Identity

Pro řešení problematiky přihlašování uživatelů do webové aplikace a správy uživatelských účtů slouží API s názvem ASP.NET Core Identity. Uživatelé si díky ní mohou vytvářet účty buď přímo v aplikaci, nebo využívat již existující účty externích zprostředkovatelů, jako je například Facebook. Identita potřebné informace uchovává v databázi či na jiném trvalém úložišti. (22)

### 3.3.2 NuGet

Balíčkovacím systémem pro platformu .NET je NuGet. Slouží vývojářům pro vytváření a využívání knihoven vytvořených pro tuto platformu. (23) Může být využíván i pro sdílení soukromých balíčků kódu v organizaci.

Důležitou funkcí NuGet je správa závislostí mezi jednotlivými balíčky. Pokud některý z používaných balíčků používá balíček jiný, NuGet se postará o správné nastavení závislostí. (24)

Aby se při práci na projektu na jiném zařízení nemusely vždy všechny balíčky znovu instalovat a zároveň nedocházelo k nadměrnému využívání paměti, neumisťuje NuGet binární sestavy balíčků přímo v projektu, ale vytváří si seznam odkazů na balíčky, který projekt využívá. (24)

### 3.3.3 Scaffolding

Při práci s ASP.NET (ASP.NET Core) je možné využít tzv. Scaffolding. Jedná se o framework sloužící ke generování kódu pro tvorbu webových aplikací. Díky tomu lze mimo jiné i vygenerovat základní projekt používající MVC, ze kterého je možné vycházet v další práci. (25)

### 3.3.4 ASP.NET MVC

ASP.NET MVC je framework pro vytváření webových aplikací s využitím MVC architektury. Model je zde reprezentován vrstvou sloužící k přístupu k datům, často využívající nástroje typu Entity Framework. Pro tvorbu View se využívají šablony umožňující dynamické generování HTML a Controllery jsou speciální třídy propojující Model a View. (10)

#### 3.3.4.1 Konvence

Dodržování konvence pro zápis v ASP.NET MVC umožňuje vývojářům pracovat, bez nutnosti přílišných konfigurací a specifikací. Při dodržování správných pojmenování zvládne framework velké množství věcí sám odvodit a nastavit. Konkrétní Pohledy tak například defaultně vždy hledá ve složce s názvem „Views“ a u tříd Controllerů předpokládá název se sufixem „controller“ (10)

#### 3.3.5 .NET 5

Mimo .NET Framework a .NET Core existuje ještě jedna platforma .NETu: Mono. V budoucnosti se plánuje sjednocení všech těchto odnoží do jedné jediné verze s názvem .NET 5. (26)

### 3.4 PostgreSQL

PostgreSQL je objektově relační databázový systém vyvinut původně pro UNIX-like operační systémy. Byl však navržen tak, aby se dal využívat i na jiných platformách. PostgreSQL je open source software a pod PostgreSQL licencí lze jeho kód používat, upravovat i distribuovat (27) a to jak nekomerčně, tak i komerčně. Je vyvíjen skupinou nezávislých vývojářů a uživatelů. (28)

Umožňuje použití většiny datových typů používaných v SQL, včetně datového typu pro zápis data, časového razítka či boolean. Mimo to lze ale pracovat i s daty ve formátu JSON nebo XML. Podporuje použití cizích klíčů, pohledů nad tabulkami a práci s procedurami. (28)

Jednou z hlavních předností tohoto databázového systému je jeho rozšiřitelnost. V případě potřeby je například možné definovat vlastní datové typy. (28)

#### 3.4.1 Dědičnost

PostgreSQL umožňuje vytvářet nové tabulky pomocí dědění atributů z jedné, nebo více jiných tabulek. Při práci výhradně s rodičovskou tabulkou je pak nutné používat příkaz ONLY pro nezahrnutí tabulek, které z ní byly vytvořeny pomocí dědění. (29)

#### 3.4.2 MVCC

Při současné práci více uživatelů s databází je nutné zajistit konzistenci dat izolováním jednotlivých transakcí. K tomu slouží MVCC (Multi Version Concurrency

Control) architektura, která zajišťuje, aby změny provedené uživatelem nebyly viditelné pro ostatní až do chvíle potvrzení transakce. (30) Tím se vytvoří více verzí dat a zabrání se vzájemnému blokování práce mezi uživateli. (31)

#### 3.4.2.1 MVCC v PostgreSQL

Databázové systémy využívají rozdílných přístupů k MVCC architektuře. V případě PostgreSQL se při aktualizaci řádku uloží nová verze do tabulky. (31) Tento řádek může v jednu chvíli existovat ve více verzích, každá transakce vidí ale pouze jednu z aktuálních verzí řádku. (30) Jak nové, tak i předchozí verze tohoto řádku zůstávají uloženy v databázi, ale ty starší jsou označeny jako neplatné a zároveň slouží jako ukazatelé na verzi novou. Staré verze zůstávají v databázi do té doby, než přestanou být potřebné a jsou smazány.

I pokud transakce není dokončena, uloží se. Je ovšem označena jako přerušená. Proto tabulky v systému PostgreSQL mohou obsahovat data z přerušovaných transakcí. (31)

To, zda je verze konkrétního řádku viditelná transakcí, je dáno hodnotami stavových proměnných xmin a xmax, které obsahuje každá verze. Hodnota xmin označuje transakci, díky které daná verze vznikla a xmax tu, kvůli které naopak zanikla. Pokud hodnota xmin označuje aktuální nebo potvrzenou transakci, a hodnota xmax nikoliv, je tato verze viditelná. (30)

## 3.5 Entity Framework

### 3.5.1 Objektově relační mapování

Objektově relační mapování (ORM) je technika převodu dat mezi relační databází a objektovým modelem. Díky tomuto postupu je možné s daty z databázových tabulek pracovat pomocí objektů, které je reprezentují, zcela bez příkazů SQL. (32) Výhodami této techniky je možnost práce s databázemi bez znalosti dotazovacího jazyka a také eventuální změna relačního databázového systému za jiný v kterékoliv fázi vývoje. (33) Jedním z dostupných ORM variant je Entity Framework (EF) od firmy Microsoft. (34)

### 3.5.2 Entity Framework Core

Entity Framework Core je multiplatformní open source verze EF. (35) Pomocí plug-in knihoven umožňuje práci s desítkami různých databázových systémů. Tyto knihovny se označují jako poskytovatelé databází (database providers). Neplacenou verzi

poskytovatele pro PostgreSQL spravuje Npgsql Development Team a je dostupná přes balíčkovací systém NuGet Package. (36)

V současné době EF Core nepodporuje NoSQL databáze. (37)

### 3.5.3 Entity Data Model

Pro všechny operace s databází je v Entity Framework používán Entity Data Model (EDM). Jedná se o model, který popisuje entity a jejich vzájemné vazby. (38) EDM je tvořen třemi částmi: Storage Schema model popisující schéma databáze, Conceptual model je model entit, nad kterými se pak provádějí dotazy a Mapping model, který provádí mapování mezi Storage Schema modelem a Conceptual modelem. (39)

Entity Framework poskytuje .NET API pro možnost psaní .NET aplikací s využitím Entity Data Modelu. (40)

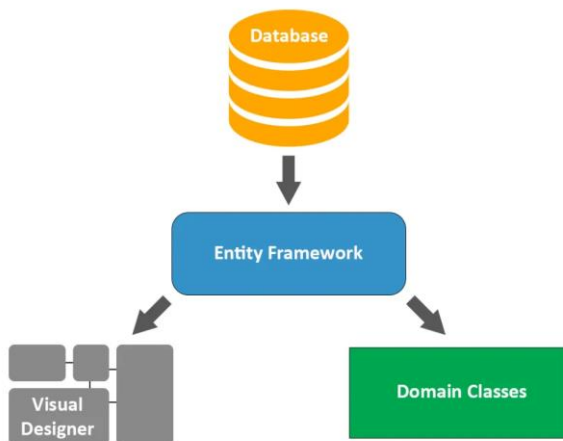
### 3.5.4 Možnosti přístupu

Existují tři možnosti, jak práci s Entity Framework pojmout:

- Database First
- Model First
- Code First

První způsob (Database First) vyžaduje nejdříve vytvořit databázi, na základě které pak EF vytvoří model (EDM), ze kterého se poté vygeneruje kód odpovídající schématu databáze, jak je znázorněno na obrázku (Obrázek 2). (40) Tento přístup se hodí využít v případě již existující databáze. (41)

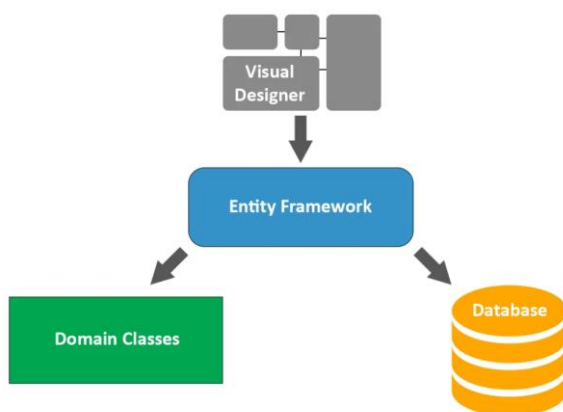
Obrázek 2: EF – Database First přístup



Zdroj: (41)

Model First varianta pak předpokládá existující model, ze kterého následně vygeneruje databázi (Obrázek 3). (40) Pro tento přístup je potřeba využít některého z nástrojů pro modelování databází. (41)

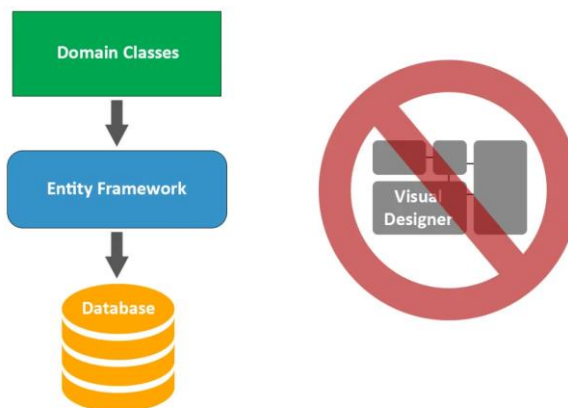
Obrázek 3: EF – Model First přístup



Zdroj: (41)

Při použití Code First varianty stačí k nadefinování modelových objektů pouze klasické třídy objektově orientovaného programovacího jazyka. Díky tomu je možné vytvořit databázi zcela bez jakýchkoliv diagramů, modelovacích nástrojů, anebo znalosti databáze viz obrázek (Obrázek 4). (41)

Obrázek 4: EF – Code First přístup



Zdroj: (41)



#### 3.5.4.1 Migrace

Během vývoje je potřeba přidávat do projektu nové entity a popřípadě ty současné upravovat. Tím se mění datový model. Pokud by se při každé změně v případě Code First přístupu musela vždy generovat nová databáze, zapříčinilo by to komplikovanou manipulaci s daty, aby nedošlo k jejich ztrátě. V EF proto existuje funkce migrace pro postupné aktualizace databázového schématu. Ta vygeneruje kód, který synchronizuje datový model s databází bez ztráty dat. Pomocí migrace se také vytváří i první verze databáze. (42)

#### 3.5.4.2 Vytvoření modelu v závislosti na přístupu

Od vybraného přístupu k práci s EF se odvíjí i způsob vytváření EDM. Při použití Code First generuje EF model za běhu, z tříd vytvořených vývojářem. Jiný postup vyžaduje vytvoření modelu přímo vývojářem. (43)

### 3.5.5 DbContext

Hlavní komunikační třídou využívanou při práci s EF je DbContext. (44) Slouží ke správě připojení k databázi, ukládání dat do databáze, řízení transakcí a detekci změn v databázi. (45)

Díky této třídě je možné provádět dotazy nad konkrétními entitami v databázi. Z výsledků těchto dotazů jsou vytvářeny objekty, u kterých DbContext sleduje prováděné změny, a ty poté převádí zpět do databáze. (44)

DbContext sám o sobě je pouze interface a pro reálnou práci s databází je nutné v projektu vytvořit jeho instanci – kontextovou třídu. (45)

#### 3.5.5.1 Metody nad DbContext

Pro provádění jednotlivých akcí nad databází slouží několik metod. Například metoda *Add* umožňuje vkládání nových entit do databáze, *Update* tyto entity měnit a *Remove* je mazat. Všechny tyto změny jsou provedeny až ve chvíli, kdy je nad DbContext zavolána metoda *SaveChanges*. (45)

### 3.5.6 DbSet

Třída DbSet reprezentuje jednotlivé entity databáze. Pro vytvoření databázových tabulek z konkrétních tříd v aplikaci je nutné v kontextové třídě mít pro tyto třídy definovány vlastnosti datového typu DbSet. (46)

### 3.5.7 Práce s databázemi v ASP.NET Core MVC

Při práci s kontextovou třídou v ASP.NET Core MVC se využívá Dependency Injection. Konkrétní instanci třídy DbContext je tedy potřeba zaznamenat v metodě ConfigureServices. Pro připojení k databázi slouží ConnectionString zapsaný například v souboru appSettings.json. (47)

## 3.6 C#

C# je objektově orientovaný programovací jazyk vyvíjen společností Microsoft. (48) Dříve sloužil výhradně pro vývoj pro platformy Windows. V současné době se jedná o jazyk, který není závislý na konkrétní platformě. Protože je C# založen na objektově-orientovaném paradigmatu, obsahuje zapouzdření, dědění i polymorfismus. (49) Zároveň umožňuje využívat již existující softwarové komponenty, díky čemuž podporuje komponentově orientované programování. (48) Syntaxe C# je velmi podobná syntaxím jazyka C a C++. (49)

### 3.6.1 Datové typy

C# má jednotný typový systém. Obsahuje jeden základní typ, který vznikl zapouzdřením potřebných funkcí a dat a z něhož následně ostatní datové typy vychází. Proto lze na každý objekt či proměnou s některým z primitivních datových typů zavolat stejné funkce, jako například funkci *ToString*. (49)

Instance jednotlivých typů spolu mohou komunikovat pouze jimi definovanými protokoly. To zajišťuje konzistentnost typů a typovou bezpečnost jazyka C#. (49) Díky typové bezpečnosti také není možné například číst neinicializované proměnné či pracovat za hranicemi definované paměti pro pole. (48)

C# je také tzv. „strongly typed language“ a proto je práce s datovými typy velmi striktně hlídaná. Není tedy mimo jiné možné například do proměnné celočíselného typu vkládat jinak definovaná čísla. (49)

### 3.6.2 Atributy

Atributy v jazyce C# slouží k přiřazování metadat a deklarativních informací k částem kódu. (50) Přístup k těmto atributům pak umožňuje tzv. reflexe, což je funkce, která vrací objekty popisující sestavy, moduly a typy. (51) Atributy se do kódu vkládají pomocí hranatých závorek a názvu daného atributu. Defaultně jsou určeny pro kód, který za nimi následuje, to však může být změněno pomocí klíčového slova *target*. (50) Příkladem využití atributů může být například zápis s [HttpPost], který umožňuje spuštění následující metody pouze po zaslání požadavku http typu post. (52)

### 3.6.3 Validace v C#

Pro přidání validačních funkcí do projektu psaného v C# se využívá namespace DataAnnotations, který je součástí .NET. Obsahuje validační atributy pro třídy a vlastnosti. Například atribut [Required] označující danou vlastnost jako povinnou či atribut [Range] určující povolený rozsah hodnot. Při používání EF přístupu Code First je tímto zajištěno, že se data neuloží do databáze, nejsou-li validní. (53)

### 3.6.4 LINQ

LINQ (Language Integrated Query) je množina funkcí sloužící k psaní strukturovaných dotazů nad kolekcemi objektů a vzdálenými datovými zdroji. (49) Jedná se o dotazovací technologii, která je přímo obsažena v syntaxi jazyka C#. LINQ je svou funkcionalitou velmi podobný dotazovacímu jazyku SQL. (54)

Pro LINQ existuje více různých typů poskytovatelů (providerů) a je možné si vytvořit i vlastní provider. (54) Tyto typy obsahují různé implementace metod pro práci s LINQ, odvíjející se od použité platformy. Provider LINQ to Objects umožňuje dotazovat se nad kolekcemi a LINQ to SQL obsahuje ORM, čímž umožňuje mapovat LINQ příkazy na dotazy SQL. (55) LINQ to Entities pak funguje na stejném principu, jako LINQ to SQL, pouze s využitím Entity Framework. (56)

#### 3.6.4.1 LINQ to Objects

Základní jednotky, se kterými se v LINQ to Objects pracuje, jsou elementy a sekvence. Sekvencemi se rozumí kolekce, které implementují IEnumerable<T>. Elementy jsou pak jednotlivými položkami těchto sekvencí. (49) V přístupu ke kolekcím může LINQ

to Objects nahrazovat používání cyklu *foreach*, oproti kterému jsou LINQ příkazy stručnější a přenositelné mezi různými datovými zdroji. (57)

#### 3.6.4.2 LINQ to Entities

LINQ to Entities umožňuje psaní dotazů nad Conceptual Modelem EDM Entity Frameworku. Výsledky takovýchto dotazů jsou pak objekty, se kterými se následně může pracovat jak v Entity Framework, tak v LINQ. (56) Funkcionalitu pro vyhodnocování dotazů poskytuje rozhraní `IQueryable<T>`, které implementuje rozhraní `IEnumerable<T>`. (58) Jako zdroj dat pro dotazy slouží instance třídy, která `IQueryable<T>` rozhraní implementuje. (56)

#### 3.6.4.3 Syntaxové metody

Pro psaní příkazů v LINQ existují dva typy syntaxe. Dotazovací syntaxe svým zápisem více připomíná jazyk SQL, syntaxe využívající funkce pak naopak jazyk C#. V prvním typu jsou k psaní dotazů používána klíčová slova jako *from*, *where* a *select*, v druhém typu syntaxe pak tzv. Lambda výrazy. Rozdíl je znázorněn na ukázkovém kódu (Kód 1) (59)

Kód 1: Možnosti zápisu LINQ

```
var jmena = new List<string>()
{
    "Anna",
    "Josef",
    "Karel"
};

var dlouhaJmena = from jmeno in jmena where jmeno.Length >= 5 select jmeno;

var dlouhaJmena2 = jmena.Where(jmeno => jmeno.Length >= 5);
```

Zdroj: Vlastní zpracování podle (59)

## 3.7 Razor

Razor je značkovácí syntaxe využíváná ve frameworku ASP.NET. (60) Od verze ASP.NET MVC 3 se jedná o defaultní View engine, využíváný pro vytváření View v ASP.NET. (10) Slouží k vkládání kódu zpracovávaného na serveru do webových stránek. Syntaxe Razoru obsahuje HTML značky, značky Razoru a kód vyššího programovacího jazyka. (60) Dá se využívat s jazykem C# nebo s Visual Basic. (61)

Pro interpretaci kódu psaného v Razoru je používáno speciální API, které ho přetvoří do kódu spustitelného frameworkem ASP.NET. Na začátku celého procesu převodu tzv. Razor Parser analyzuje text psaný v Razoru a rozdělí ho na bloky, které mají hierarchickou strukturu. Tím vznikne dokument celý psaný ve značkovacím jazyce. Následně je tento dokument přeložen do .NET kódu a vytvoří se z něj jednoduchá třída, která obsahuje především metadata. Z ní už pak Razor API vytvoří spustitelnou třídu. (62)

Stránky se syntaxí Razor využívající C# se poznají podle přípony .cshtml. (61)

### 3.7.1 HTML

Značkovací jazyk HTML (HyperText Markup Language) slouží pro tvorbu webových stránek a definici jejich struktury. Skládá se z jednotlivých elementů, které internetovému prohlížeči popisují, jak má zobrazovat jejich obsah. Jednotlivé prvky jsou reprezentovány množinou předem definovaných značek. Aktuální verzí je HTML5. (63)

### 3.7.2 Práce s Razorem s použitím C#

Hlavní složkou syntaxe Razoru je znak zavináč. Ten indikuje přechod ze značkovacího jazyka do psaní kódu. (62) Stejný znak se použije i pro označení klíčových slov vyhrazených pro Razor.

C# může být použit pro psaní výrazů vkládaných do značek HTML. Takové výrazy jsou pak i vykreslovány v prohlížeči. Pro vložení větší sekvence kódu se využívá složených závorek umístěných za znakem zavináče. Tyto bloky kódu slouží jen pro pomocné operace a uživateli zobrazovány nejsou.

Za klíčovými slovy označujícími řídicí struktury v jazyce C# není nutné po složených závorkách již uvádět znovu znak zavináče detekující C# kód.

Razor umožňuje psaní komentářů podle syntaxe HTML a v blocích kódu i podle syntaxe používaného programovacího jazyka. (60)

#### 3.7.2.1 Direktivy Razoru

Direktivy jsou psány pomocí znaku zavináče a klíčových slov vyhrazených pro Razor. Obvykle slouží ke změně způsobu zobrazení obsahu či k použití speciálních funkcí. Například direktiva `@injection` slouží pro práci s Dependency injection, `@model` specifikuje typ Modelu (popř. ViewModelu), který je používán v dané View stránce a `@functions` umožňuje vkládání polí, metod a vlastností. (60)

### 3.7.2.2 Razor Helpers

Razor Helper je funkce, která umožňuje znovupoužití konkrétního kódu napsaného v Razoru. Funguje na podobném principu jako globální funkce. Slouží především k práci s menšími sekcemi kódu. (62)

Použití syntaxe Razor je znázorněno na ukázkovém zdrojovém kódu (Kód 2).

Kód 2: Ukázka Razoru

```
@model Uzivatel
@{
    bool promenna = true;
    string napis = "Hello world";
}

<h1>View</h1>
<h2>@Model.KrestniJmeno @Model.Prijmeni</h2>
@if (promenna)
{
    <p>@napis</p>
}
```

Zdroj: (Autor)

## 3.8 Hosting webových aplikací

Pro zpřístupnění webové aplikace či webových stránek koncovému uživateli se využívá webového hostingu. Tato služba je umožněna díky poskytovatelům, kteří vlastní servery, na kterých umožňují webové stránky hostovat. Tyto servery fungují na principu, který byl popsán v první kapitole. (64)

### 3.8.1 Typy webového hostingu

Existuje více typů pojetí webového hostingu. Nejpoužívanějším z nich je sdílený webový hosting, při kterém se na jednom serveru nachází více stránek patřící různým uživatelům. Toto řešení je levné, neumožňuje však mimo jiné přístup ke konfiguracím serveru daného serveru.

Další možností webového hostingu je využití dedikovaného serveru. Takový server je určen pouze jednomu uživateli. Je využíván hlavně při potřebě uživatele plně kontrolovat server, je to také dražší řešení. (64) Cena za dedikovaný server se pohybuje od 1700 Kč měsíčně. (65) Na podobném principu funguje i tzv. Managed hosting, zde ale uživatel není odpovědný za správu serveru (64)

Střední cestou mezi dedikovaným serverem a sdíleným hostingem je virtuální privátní server. Zde má každá webová stránka svůj vlastní prostor na serveru, který je ale sdílený mezi více uživateli.

Novějším typem webového hostingu je Cloud webhosting. Princip tohoto hostingu je využití klastrovaných serverů, kdy jednu stránku má na starosti více serverů, v případě selhání jednoho. Cloud webhosting automaticky zálohuje data a je snadno škálovatelný v případě zvýšené návštěvnosti. (64)

Protože je ASP.NET Core multiplatformní, je možné hostovat aplikace psané v tomto frameworku jak na serverech s operačním systémem Windows, tak i na Linuxových serverech. (66)

### **3.8.2 Platform as a Service**

Platform as a Service (PaaS) je prostředí pro vývoj a běh aplikací v cloudu. PaaS zahrnuje servery, uložení, síť, middleware i systém pro správu databází a další prostředky potřebné k chodu aplikací. (67) PaaS nabízí mnoho poskytovatelů, v následujících podkapitolách jsou vybrány čtyři umožňující běh ASP.NET Core webových aplikací.

#### **3.8.2.1 Amazon Web Services**

Amazon Web Services podporuje hosting webových aplikací psané mimo jiné v .NET na serverech Apache, Nginx, Passenger nebo ISS. Při práci s Amazon Web Services je využíváno nástroje AWS Elastic Beanstalk, který se stará o automatické nasazení aplikace a její škálování. (68) AWS neumožňuje hosting webové aplikace psané v .NET Core zdarma. (69)

#### **3.8.2.2 Azure App Service**

Další platformou, kterou lze využít k hostování webových aplikací je Azure App Service. Umožňuje automaticky nasazovat aplikace například pomocí Githubu či Azure DevOps, (70) tedy nástroje, který mimo jiné poskytuje i Git repozitář. (71) Je zde možnost hostování webové aplikace zdarma. (72)

PaaS Azure obsahuje nástroj SendGrid sloužící k odesílání e-mailů z webové aplikace. Tato služba je poskytována zdarma na 25 000 e-mailů měsíčně. (73)

### 3.8.2.3 Heroku

Heroku je PaaS, která umožňuje běh webových aplikací pomocí kontejnerů. (74)  
Možností je hostování webové aplikace zdarma. (75)

Heroku neobsahuje oficiální podporu .NET Core frameworku. Pro nasazení takovýchto aplikací je možné využít nainstalováním balíčků poskytovaných třetí stranou. Další možností je využití Dockeru. (76) Nástroje, umožňujícího vývoj a nasazování webových aplikací pomocí kontejnerů, který umožňují běh aplikace i na serverech, které by jinak použité framework a jazyk nepodporovaly. (77)

### 3.8.2.4 Google Cloud Platform

Google Cloud Platform (GCP) umožňuje hostovat statické webové stránky i webové aplikace v Cloudu. (78) Pro hosting webové aplikace psané v ASP.NET Core se využívá Docker. Jednou z možností nasazení aplikace na GCP je využití rozšíření pro Visual Studio, které se zároveň postará i o vytvoření potřebných Docker souborů. (79) GCP neumožňuje hosting webových aplikací zdarma. (80)



## 4 Vlastní práce

### 4.1 Analýza požadavků

Po konzultaci se skautským oddílem ohledně potřebných funkcionalit vyvíjené webové aplikace, byly sestaveny následující požadavky. Ty jsou podle zásad softwarového inženýrství děleny na požadavky funkční a nefunkční. (81)

#### 4.1.1 Funkční požadavky

Funkční požadavky jsou takové požadavky, které říkají, co by navrhovaný systém měl umět. (81) Pro přehlednost a následnou práci s nimi, jsou označeny identifikátory. Písmenem F a pořadím daného požadavku.

##### 4.1.1.1 F1 – Uživatelé

Aplikace bude obsahovat databázi uživatelů. Ti budou vystupovat v jedné ze čtyř rolí. Admin, vedoucí, člen a rodič. Admin bude sloužit jako administrátor aplikace a bude mít dostupné všechny její funkcionality.

Vedoucí bude aplikaci spravovat s omezenějšími možnostmi než samotný admin. Na rozdíl od něj nebude moct upravovat registrované uživatele a upravovat všechny akce a aktuality

Uživatel v roli rodič bude mít na starosti zajišťování přihlašování na jednotlivé akce těch členů oddílu, jejichž je zákonným zástupcem a které ještě nejsou ve věku, aby tyto akce mohli provádět sami.

Uživatel v roli člen je takový uživatel, kterému je již více jak třináct let a již s aplikací pracuje sám. Zároveň však není vedoucím oddílu a nevystupuje tedy v roli vedoucí. Pro označení rozdílu mezi členem, který je registrován v aplikaci a mezi členem, který je pouze ve správě rodiče, bude tato role označována vždy jako „uživatel v roli člen“ a člen ve správě pouze jako člen.

Jednotlivé role se nemohou překrývat.

##### 4.1.1.2 F2 – Přihlášení

V aplikaci bude možnost přihlašování jednotlivých uživatelů, kteří mají v aplikaci založeny své účty. K přihlašování bude využíván e-mail a heslo. Nepřihlášený uživatel se

nebude moci přihlašovat na pořádané akce a případně (dle své role) upravovat jednotlivé příspěvky či účty uživatelů.

#### 4.1.1.3 F3 – Odhlášení

Přihlášení členové se budou moct kdykoliv odhlásit a aplikaci používat dále jako nepřihlášení uživatelé.

#### 4.1.1.4 F4 – Registrace

Webová aplikace skautského oddílu je částečně uzavřena pro veřejnost. Aplikace bude umožňovat přidávání nových účtů pouze přihlášenými uživateli v roli admin nebo vedoucí. Při registraci tito uživatelé zadají e-mail nového uživatele, jeho jméno a roli, ve které bude v aplikaci vystupovat. Uživatelé v roli vedoucí mohou přidávat pouze uživatele v roli rodič či člen. Po registraci aplikace odešle na zadaný e-mail nově registrovaného uživatele vygenerované heslo, které si nově registrovaný uživatel může následně změnit.

#### 4.1.1.5 F5 – Přidávání členů do správy rodičům

Aplikace bude umožňovat přidávat uživatelům v roli rodič do správy jednotlivé děti. To bude umožněno přihlášeným uživatelům v roli vedoucí či admin. Takto přiřazení členové budou pouze abstraktní, k takovýmto účtům nebude možné se přihlašovat. V aplikaci budou označovány jako členové. Tyto členy může admin a vedoucí upravovat a mazat.

#### 4.1.1.6 F6 – Změna hesla

Přihlášený uživatel má možnost změnit si své přihlašovací heslo. Tuto akci aplikace bude umožňovat v neomezeném množství.

#### 4.1.1.7 F7 – Obnova hesla

V případě zapomenutí přihlašovacího hesla uživatelem aplikace bude umožňovat toto heslo obnovit pomocí zadaného e-mailu.

#### 4.1.1.8 F8 – Informační portál

Pro přihlášené i nepřihlášené uživatele bude aplikace fungovat jako informační portál. Zobrazovány budou informace o samotném oddíle a možnostech kontaktování

daného oddílu. Dále také informace o dalších projektech tohoto skautského oddílu. Webová aplikace také bude obsahovat odkaz k veřejnému fotoalbu oddílu.

#### 4.1.1.9 F9 – Aktuality

Aplikace bude zobrazovat příspěvky o aktuálním dění oddílu, které budou přístupné i nepřihlášeným uživatelům a budou přidávány přihlášenými uživateli v roli vedoucí či admin. Tyto aktuality pak mohou být upravovány či mazány jejich autory. Pro uživatele v roli admin je možnost upravit a smazat jakoukoliv aktualitu, bez ohledu na autorství. Aktuality budou řazeny dle data a času přidání od nejnovější po nejstarší.

#### 4.1.1.10 F10 – Družiny

Členové skautského oddílu jsou rozděleny do jednotlivých družin podle jejich věku.

Aplikace bude umožňovat jednotlivým uživatelům v roli člen mít přiřazenou jednu z družin. Družiny budou přiřazovány i členům, které mají ve správě někteří z rodičů. Vedoucím a adminům se družina nepřirazuje a člen nemusí patřit do žádné družiny. Vedoucí, admin či uživatel v roli člen bude moci být vedoucím některé z družin.

Členové si budou moci zobrazit informace o své družině.

#### 4.1.1.11 F11 – Seznam akcí

Aplikace bude obsahovat databázi oddílem pořádaných akcí, které budou přidávat přihlášení uživatelé v roli admin či vedoucí. Na stránkách se budou zobrazovat pouze ty akce, které se budou konat v budoucnu od příštího dne včetně. Zobrazené akce budou obsahovat popis, datum konání a družiny, pro které je tato akce určena.

Aplikace umožní úpravu a mazání akcí těmi uživateli, kteří jsou autory těchto akcí. Uživatel v roli admin může upravovat a mazat všechny akce.

#### 4.1.1.12 F12 – Přihlašování na akce

Na jednotlivé akce zobrazované aplikací, budou mít přihlášení uživatelé možnost se přihlásit. Uživatelé v roli člen pouze mají-li přiřazenou družinu a pokud bude akce určena pro jejich družiny.

Uživatelé v roli rodič budou moci členy ve své správě přihlásit k jednotlivým akcím pouze mají-li tyto členové přiřazenou družinu a bude-li akce určena pro jejich družiny.

Admini a vedoucí se budou moci přihlásit na akci bez jakéhokoliv omezení.

#### 4.1.1.13 F13 – Členové a uživatelé, kteří jedou na akci

Aplikace bude umožňovat zobrazení přihlášených členů a uživatelů, které se dané akce zúčastní. Tuto informaci si budou moci zobrazovat pouze vedoucí a admini.

#### 4.1.1.14 F14 – Zobrazení uživatelského profilu

Přihlášení uživatelé si budou moci zobrazovat informace o svém profilu na dané webové aplikaci. Zobrazován bude i seznam akcí, na které je konkrétní uživatel přihlášen. Uživatelé v roli rodič si budou moci zobrazit seznam jim přiřazených členů a zobrazit si, na které akce jsou tito členové přihlášení a informace o nich.

#### 4.1.1.15 F15 – Archiv akcí

Aplikace bude obsahovat informace o již proběhlých akcích. Tyto informace, včetně uživatelů a členů, kteří se jich zúčastnili, budou zobrazovány adminům a vedoucím.

#### 4.1.1.16 F17 – Informace o uživatelích a členech

Aplikace bude obsahovat funkcionalitu pro zobrazení informací o členech a uživatelích pro adminy a vedoucí.

### **4.1.2 Nefunkční požadavky**

Nefunkční požadavky vyjadřují, co by systém měl splňovat a jaký by měl být. (81)  
Zde budou identifikovány písmenem N a pořadím požadavku.

#### 4.1.2.1 N1 – Implementace

Webová aplikace bude vyvíjena pomocí frameworku ASP.NET Core.

#### 4.1.2.2 N2 – Vzhled

Webová aplikace bude odpovídat grafickým normám skautské organizace podle dostupného skautského grafického manuálu.

#### 4.1.2.3 N3 – Cena

Protože je oddíl neziskovou organizací, hosting aplikace by měl být zdarma.

## 4.2 Analýza dostupných řešení

Vzhledem k velkému počtu skautských oddílů v České republice, je na internetu dostupné obdobně velké množství webových stránek a webových aplikací těchto oddílů. Valná většina z těchto webových prezentací jsou však pouze statické stránky, které neobsahují žádnou z funkcionalit popsaných v analýze požadavků, kromě informačního portálu.

Bylo tedy vybráno jen několik webových prezentací, které využívají obdobné mechanismy jako vyvíjená webové aplikace a u kterých se nejedná pouze o statické webové stránky.

### 4.2.1 Skautský oddíl Krnovská Trojka

Skautský oddíl Krnovská Trojka má webové stránky s možností přihlášení na jednotlivé akce, které jsou i přehledně vyznačeny v grafických kalendářích. Je zde vedena detailní evidence členů, která je volně dostupná pro veřejnost. A to i například evidence plateb. Možností je odesílání zpráv jednotlivým členům, což je také možné dělat i bez přihlášení. Mimo odesílání soukromých zpráv se zde nachází i návštěvní kniha, kde uživatelé mohou publikovat své příspěvky. Vzhled webu si uživatel může sám upravovat nastavováním čtyř různých barev. (82)

Stránky tohoto oddílu jsou ukázány na titulní stránce webu (Obrázek 5).

Obrázek 5: Webové stránky oddílu Krnovská Trojka



Zdroj: (82)

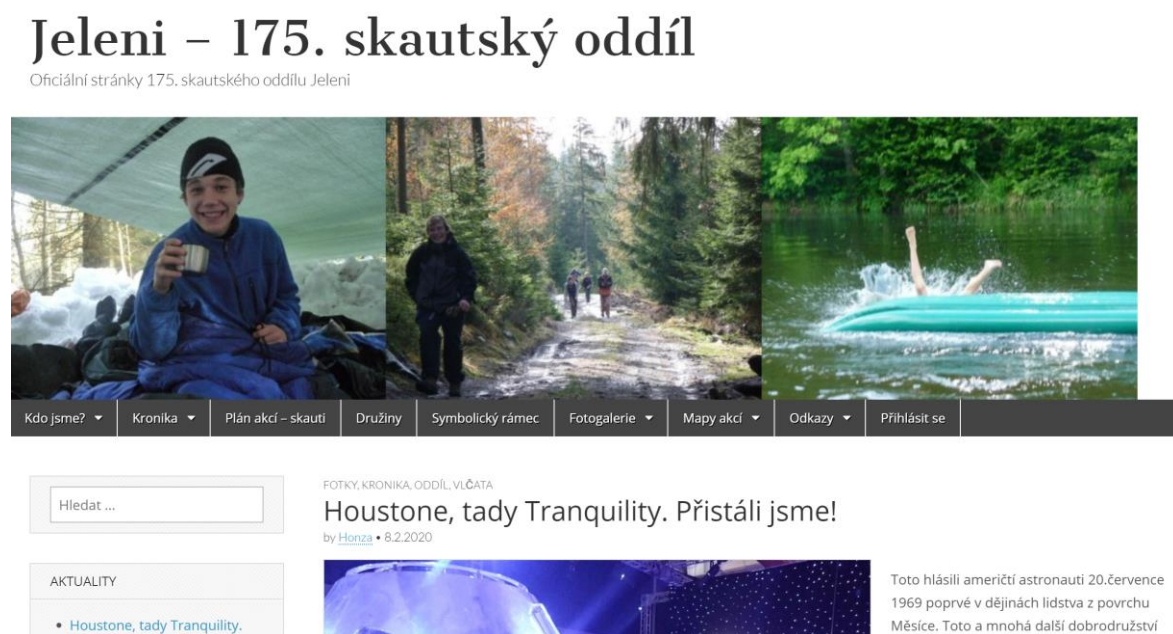
#### 4.2.2 Jeleni – 175. skautský oddíl

Webová aplikace tohoto oddílu obsahuje možnost přihlášení uživatelů na akce externě pomocí Google kalendáře. Dále je zde také záložka mapy akcí, která je ovšem prázdná a archiv akcí s poslední aktualizací v roce 2014. Tedy tento mechanismus již zjevně není nadále využíván. Záložka „Plán akcí – skauti“ obsahuje výčet jednotlivých akcí bez bližšího popisu.

Pro veřejnost jsou zobrazitelné prakticky veškeré funkcionality aplikace, které jsou čistě informativního rázu. Obsahují zprávy o aktuální činnosti oddílu, informace obecně o skautské organizaci, či archivní zprávy o akcích oddílu. Přihlášení uživatelé navíc mohou vkládat pod příspěvkům komentáře. (83)

Stránky tohoto oddílu jsou ukázány na titulní stránce webu (Obrázek 6).

Obrázek 6: Webové stránky oddílu Jeleni



Zdroj: (83)

#### 4.2.3 7. dívčí skautský oddíl, středisko Pětিলisté Růže, Jindřichův Hradec

Tyto stránky, jako jediné ze zde zmíněných, umožňují registraci i veřejnosti. Ze stránek však není patrné, co registrace a následné přihlášení umožní. Jedna ze záložek hlavního menu odkazuje na „Vzkazník“, který ovšem v tuto chvíli není funkční. Stránky slouží tedy spíše pouze jako informační portál bez dalších funkcionalit. (84)

Stránky tohoto oddílu jsou ukázány na titulní stránce webu (Obrázek 7).

Obrázek 7: Webové stránky 7. dívčího skautského oddílu



Zdroj: (84)

#### 4.2.4 Závěr analýzy

Jen velmi malé procento skautských oddílů využívá ve své činnosti webové aplikace. Z desítek navštívených webových stránek také pouze jedna obsahovala design dělaný podle skautského grafického manuálu. Z výše popsaných webových aplikací pak takovouto design neměla žádná.

Funkcionalitou, která není uvedena v analýze požadavků na aplikaci, a která se ve výše popsaných webových aplikacích vyskytovala, je možnost přidávání příspěvků všemi uživateli. A to nejen příspěvků týkajících se chodu oddílu, ale i v obecném diskuzním fóru. Tato funkcionality však nebyla zadána v požadavcích na vyvíjenou aplikaci.

Autoři webových aplikací se zjevně příliš nezaměřovali na ochranu soukromí a bezpečnost, neboť většina interních informací o členech a oddílu je zcela dostupná veřejnosti a lehce zneužitelná.

### 4.3 Technické prostředky

Pro implementaci zadané webové aplikace budou využity následující technické prostředky.

Aplikace bude napsaná pomocí ASP.NET Core, jak je uvedeno v nefunkčních požadavcích. Bude použita MVC architektura, která bude rozšířena o Repository pattern. Aplikace bude tedy napsána v jazyce C# a jednotlivé View pak v Razoru. Design stránek bude vytvořen pomocí kaskádových stylů a frameworku Bootstrap. Některé funkcionality ve View budou zajištěny pomocí JavaScriptu.

Jako databázový systém využívaný v aplikaci byl zvolen PostgreSQL, a to z důvodů, že se jedná o Open source databázový systém, který je navíc multiplatformní stejně jako ASP.NET Core. Přístupovat se k němu bude pomocí Entity Framework Core. Zvoleným postupem pro realizaci databáze je Code First.

K nasazení samotné aplikace bude použita PaaS Azure App Service pomocí Azure DevOps. Tato PaaS byla vybrána, protože je zde umožněn hosting webové aplikace zdarma a konkrétně aplikace psané v ASP.NET Core je navíc možné nasadit i bez použití kontejnerů. Dále je zde umožněno využití e-mailového klienta zdarma.

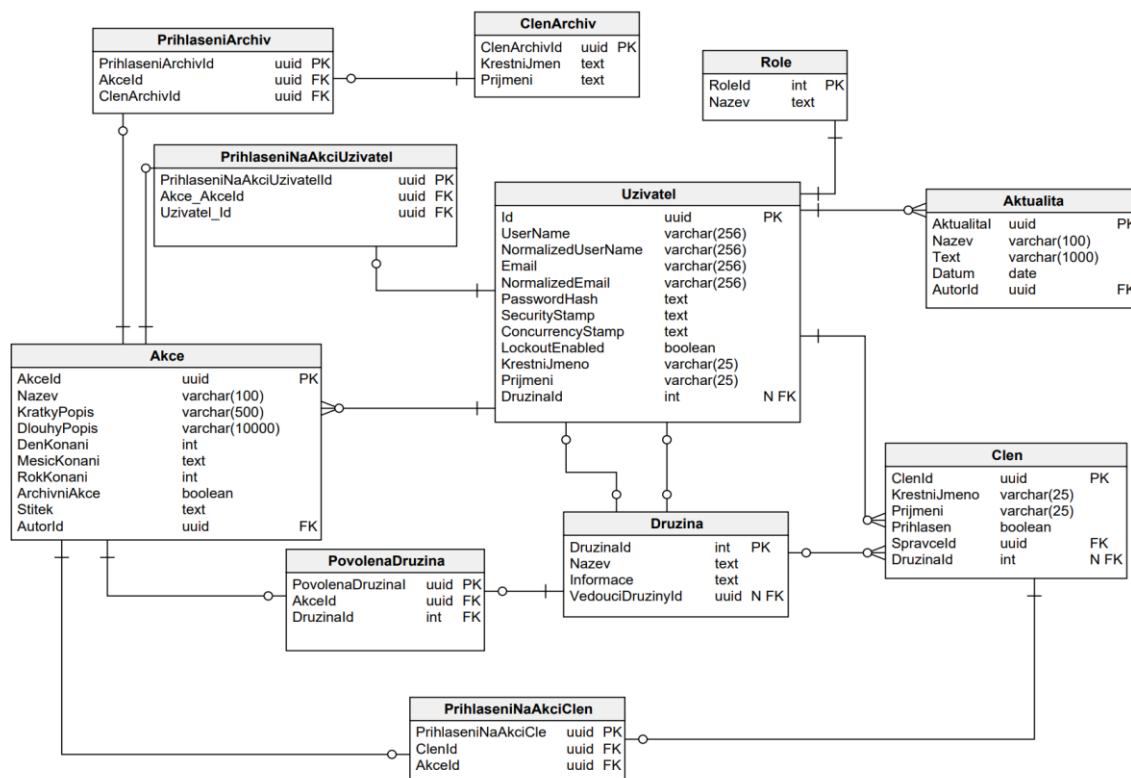
#### **4.4 Návrh databáze**

Databáze používaná pro vyvíjenou webovou aplikací má celkem jedenáct tabulek. Tabulky a atributy byly pojmenovány v jednotném čísle a v českém jazyce bez diakritiky. Primární klíče jsou identifikovány podle vzoru *Název tabulky* + Id. Výjimku tvoří tabulka „Uzivatel“, která je částečně generována pomocí ASP.NET Core Identity a tedy názvy některých z jejích atributů jsou pevně stanovené tímto frameworkem.

Návrh databáze je znázorněn na databázovém modelu (Obrázek 8).



Obrázek 8: Návrh databáze



Zdroj: (Autor)

#### 4.4.1 Datové typy

Jako databázový systém byl pro vývoj vybrán PostgreSQL, tedy datové typy byly zvoleny tak, aby byly podporované tímto systémem. Primární klíče jsou téměř u všech tabulek typu *Guid* (v PostgreSQL nazýván *uuid*). U tabulek „Role“ a „Druzina“ byl zvolen datový typ primárního klíče *int* z toho důvodu, je se v těchto tabulkách bude nacházet pouze minimální počet entit, které navíc budou pevně dané.

Další datové typy použité v tomto návrhu jsou *varchar* pro textové atributy s daným omezením délky, *text* pro textové atributy bez omezení a dále *date* a *boolean*.

#### 4.4.2 Tabulky a vazby mezi nimi

K evidenci registrovaných uživatelů slouží tabulka „Uzivatel“. Každý uživatel musí mít právě jednu přiřazenou roli a maximálně jednu družinu. Mezi tabulkami „Uzivatel“ a „Druzina“ je navíc ještě jedna nepovinná 1:1 vazba vyjadřující, že každá družina může mít maximálně jednoho vedoucího družiny. Uživatelé mohou být autoři více aktualit či akcí.

V tabulce „Clen“ jsou evidováni členové, které mají uživatelé v roli rodič ve své správě.

Pro vyznačení přihlášení uživatele, respektive člena na akci, slouží tabulky „PrihlaseniNaAkciUzivatel“ a „PrihlaseniNaAkciClen“. Pokud má konkrétní akce nastavena hodnotu atributu „ArchivniAkce“ na *true*, jedná se o akci již proběhlou. Přihlášení na archivní akce je nadále evidováno pomocí tabulky „PrihlaseniArchiv“ a tabulky „ClenArchiv“, která slouží k uchování informací o všech uživateli a členech, které se kdy v databázi vyskytovali bez ohledu na to, jestli jejich účty stále existují.

Pro uložení dat o družinách, které se akce směly zúčastnit je v databázi tabulka „PovolenaDruzina“.

## 4.5 UI specifikace

Jako podklad pro vývoj webové aplikace byla vytvořena UI specifikace, která je umístěna v příloze této bakalářské práce. Zde je vybrána pro ilustraci část z této specifikace, která pokrývá požadavky, jejichž implementace bude názorně popsána v následující kapitole. Je zde navíc přidán Use Case Diagram a tabulka pokrytí funkčních požadavků. Oba tyto prvky vycházejí z kompletní UI specifikace.

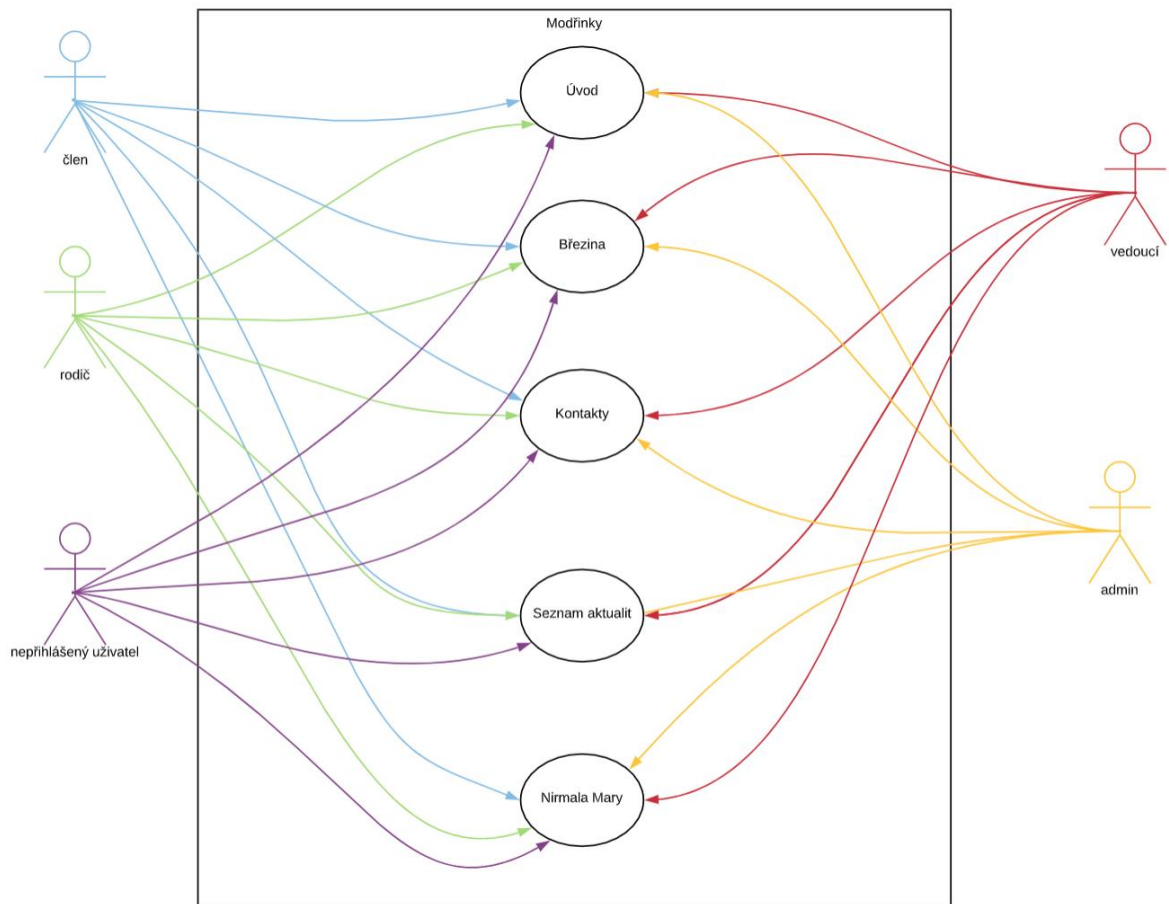
### 4.5.1 Use Case Diagram

Z důvodu velkého množství dat, a z toho vyplývající nepřehlednosti, byl Use Case Diagram rozdělen do pěti částí (Obrázek 9, 10, 11, 12, 13) a jednotliví aktéři a jejich vazby byly označeny barvami.

Aktérů je v diagramu pět. Nepřihlášený uživatel znázorňuje neregistrovaného nebo registrovaného nepřihlášeného uživatele systému. Je označen fialovou barvou.

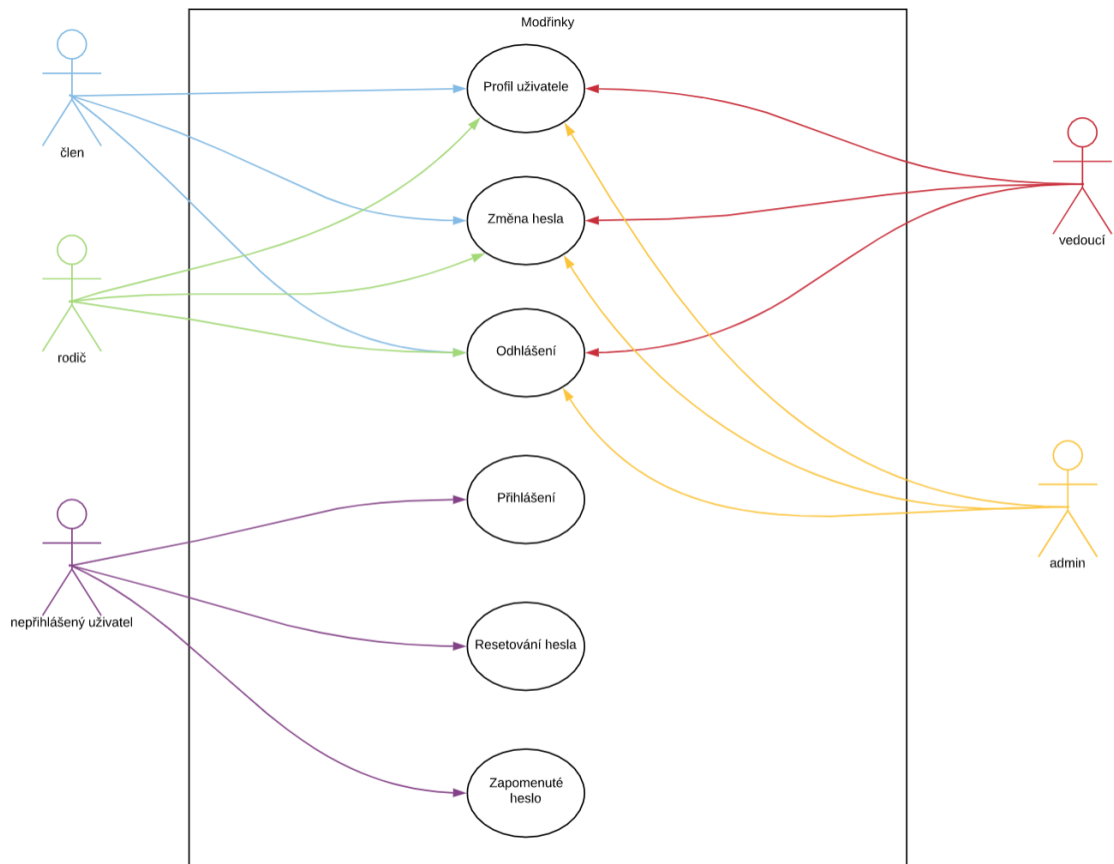
Další čtyři aktéři znázorňují již přihlášené uživatele. Modře je označen uživatel v roli člen, zeleně uživatel v roli rodič, červeně uživatel v roli vedoucí a žlutě uživatel v roli admin.

Obrázek 9: Use Case Diagram část 1



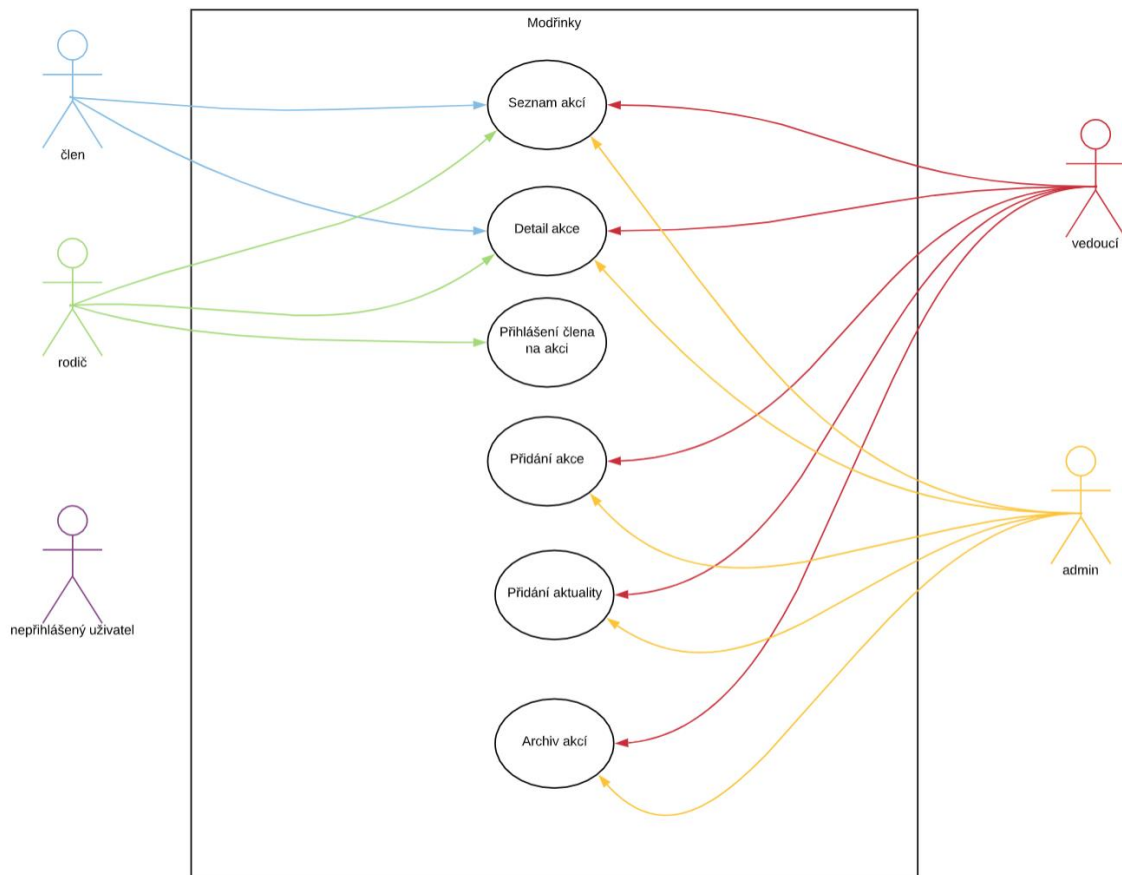
Zdroj: (Autor)

Obrázek 10: Use Case Diagram část 2



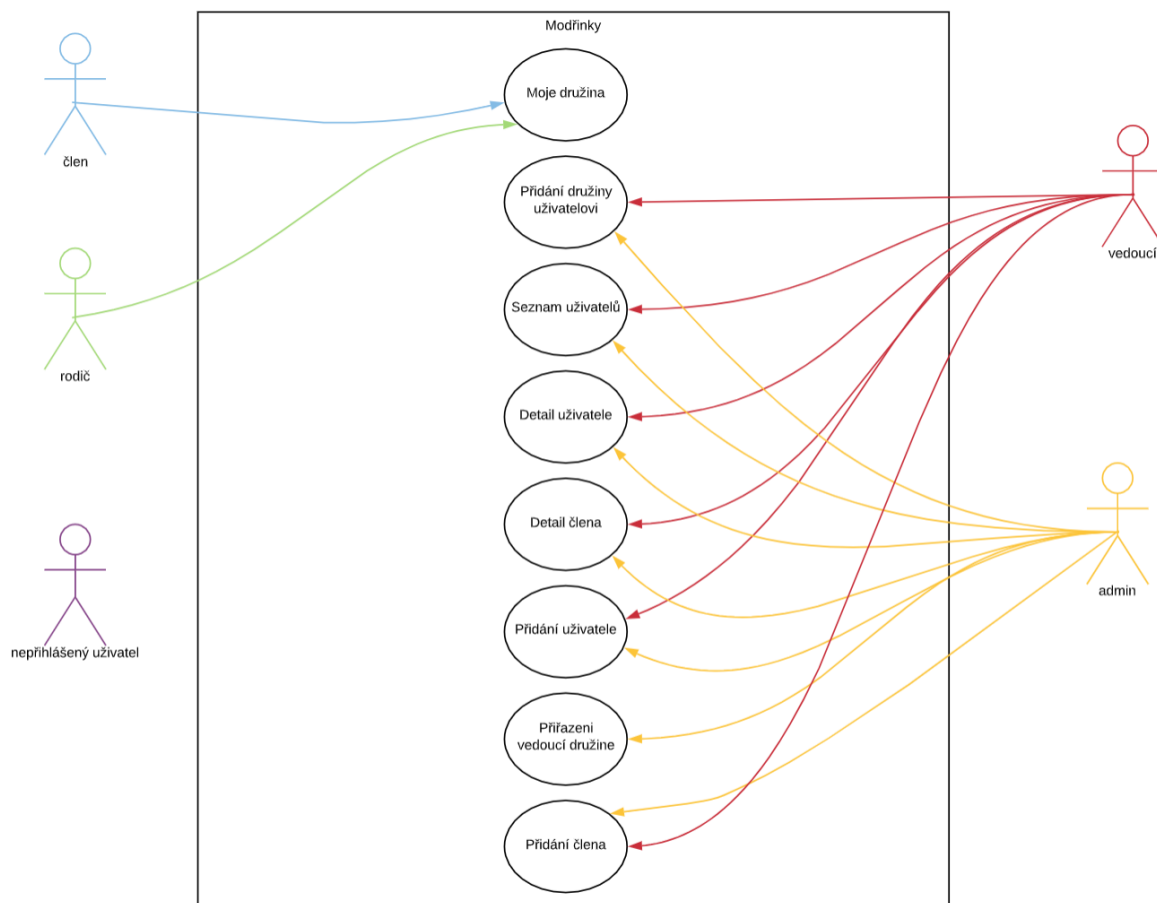
Zdroj: (Autor)

Obrázek 11: Use Case Diagram část 3



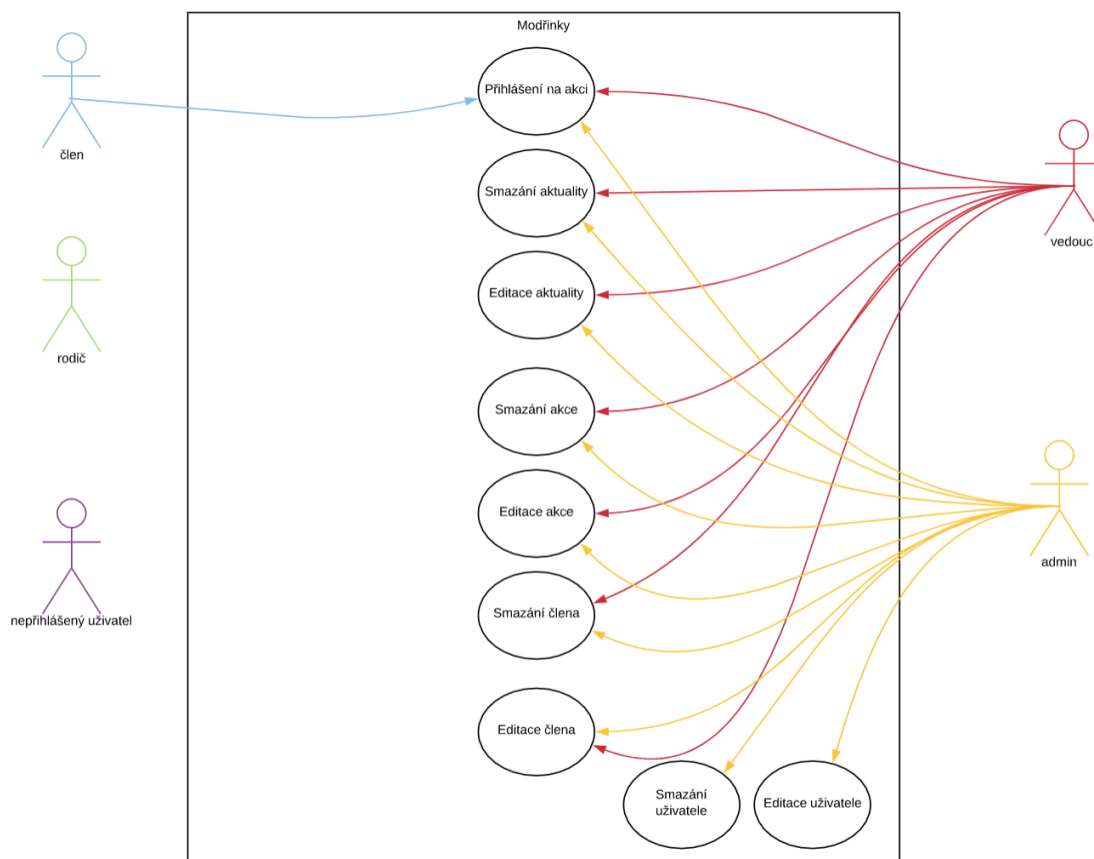
Zdroj: (Autor)

Obrázek 12: Use Case Diagram část 4



Zdroj: (Autor)

Obrázek 13: Use Case Diagram část 5



Zdroj: (Autor)

#### 4.5.2 Tabulka pokrytí funkčních požadavků

Následující tabulka (Tabulka 1) znázorňuje, jak jednotlivé Use Casy pokrývají zadané funkční požadavky.

Tabulka 1: Pokrytí funkčních požadavků

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17
Úvod																	
Kontakty																	
Nirmala Mary																	
Březina																	
Seznam aktualit																	
Profil uživatele																	
Změna hesla																	
Odhlášení																	
Přihlášení																	
Resetování hesla																	
Zapomenuté heslo																	
Seznam akcí																	
Detail akce																	
Přihlášení člena na akci																	
Přidání akce																	
Přidání aktuality																	
Archiv akcí																	
Moje družina																	
Přidání družiny uživatelovi																	
Seznam uživatelů																	
Detail uživatele																	
Detail člena																	
Přidání uživatele																	
Přidání vedoucí družině																	
Přihlášení na akci																	
Smazání aktuality																	
Editace aktuality																	
Smazání akce																	
Editace akce																	
Smazání člena																	
Editace člena																	
Smazání uživatele																	
Editace uživatele																	
Přidání člena																	

Zdroj: (autor)

### 4.5.3 Seznam akcí

#### 4.5.3.1 Use Case



Uživatel očekává:

- Možnost zobrazit si chystané akce, je-li přihlášený uživatel;
- Možnost vyfiltrvat si jednotlivé akce podle zadaných parametrů;
- Možnost dozvědět se o akci více informací;
- Možnost přidat novou akci, je-li uživatel v roli admin či vedoucí;
- Možnost editovat či smazat akci, je-li uživatel jejím autorem nebo je-li uživatel v roli admin.



#### 4.5.3.2 Scénář

System zobrazí:



- Seznam akcí, seřazené podle nejdříve konané, každá akce má název, počáteční den konání, název, štítek dle typu akce a krátký popis;
- Textové pole pro zadání názvu akce k vyfiltrování;
- Pole výběru pro zadání měsíce konání akce k vyfiltrování;
- Pole výběru pro zadání roku konání akce k vyfiltrování;
- Tlačítko „Hledat“ pro zahájení filtrování;
- Tlačítko „Reset“ pro resetování vyhledávacích formulářů;
- Je-li uživatel v roli vedoucí či admin, odkaz „Přidat akci“ pro přidání nové akce;
- Je-li uživatel v roli vedoucí či admin, odkaz „Kdo jede?“ pro zobrazení účastníků akce;
- Je-li uživatel v roli admin či je-li uživatel autorem aktuality, ikony  a .

Po kliknutí na:

- Tlačítko „Hledat“ systém zobrazí pouze ty akce, které odpovídají zadaným parametrům;
- Tlačítko „Reset“ systém vymaže zadané vstupy ve vyhledávacích formulářích;
- Odkaz „Přidat akci“ systém zobrazí stránku „Přidání akce“;
- Ikonu  systém zobrazí stránku „Editace akce“;
- Ikonu  systém zobrazí stránku „Smazání akce“;
- Odkaz „Kdo jede?“ systém zobrazí seznam přihlášených účastníků;
- Název akce systém zobrazí detail akce.

### 4.5.3.3 Wireframe

#### Wireframe 1: Seznam akcí

Úvod Kontakty Aktuality Akce Nirmala Mary Fotky Březina Pro Budiš Alžběta Pokorná Odhlásit se

## Akce

Název akce Měsíc Rok Hledat Reset

[Přidat akci](#)

1. leden 2000 Oddílová akce	
<a href="#">Název akce</a>	Popis
<a href="#">Kdo jede?</a>	

Zdroj: (Autor)

## 4.5.4 Moje družina

### 4.5.4.1 Use Case

Uživatel očekává:

- Možnost zobrazit si informace o družině, ve které je členem, je-li tento uživatel v roli člen, nebo ve které je členem člen v jeho správě;
- Možnost kontaktovat vedoucího své družiny, respektive vedoucího družiny člena, kterého má ve správě;
- Možnost zobrazit si akce určené pro tuto družinu.

### 4.5.4.2 Scénář

System zobrazí:

- Detailní informace o dané družině;
- Textové pole pro vložení zprávy pro vedoucí družiny, má-li družina přiřazeného vedoucího;
- Seznam akcí, kterých se členové dané družiny mohou zúčastnit;
- Tlačítko „Odeslat“.

Po kliknutí na:

- Tlačítko „Odeslat“ systém odešle e-mail vedoucímu družiny s textem vyplněným v textovém poli;
- Odkaz s názvem akce systém zobrazí detailní informace o dané akci.

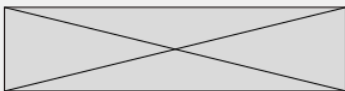
#### 4.5.4.3 Wireframe

Wireframe 2: Moje družina

Úvod Kontakty Aktuality Akce Nirmala Mary Fotky Březina Pro členy ▾ Marie Dvořáková Odhlásit se

### Družina

Popis



Vedoucí družiny: xx

Seznam členů:  
Marie Dvořáková

Seznam akcí pro družinu:  
[Akce1](#)

Napsat vedoucímu družiny:

Odeslat

Zdroj: (Autor)

## 4.6 Implementace

Kód projektu je rozdělen do sedmi složek: Controllers, kde se nachází veškeré Controllery aplikace, Models pro Modely, Views pro View s podsložkou Shared, kde je obsažen například soubor s hlavním rozložením stránky, ViewModels pro ViewModely, Areas pro RazorPages vygenerované pomocí ASP.NET Core Identity a následně upravené pro potřeby aplikace, wwwroot pro statické objekty a Migrations pro migrace Entity Framework.

V projektu je dále obsažena například třída Program se spouštěcí metodou Main, třída Startup s konfiguracemi aplikace, nastavení politiky cookies, servicami jednotlivých repozitářů a připojení k databázi. Samotný ConnectionString pro připojení k databázi je umístěn v souboru appsettings.json

Pro ukázkou implementace byly vybrány dvě části aplikace: seznam pořádaných akcí a stránka s informacemi o družině.

#### **4.6.1 Seznam akcí**

Tato část popisuje implementaci seznamu akcí pomocí konkrétních částí kódu, které pro ni byly použity.

##### 4.6.1.1 Model

Primárním Modelem použitým při implementaci seznamu akcí je Model „Akce“, který odpovídá v databázi tabulce „Akce“. Model (Kód 3) obsahuje jednotlivé atributy akce a dále také omezení hodnot textových atributů, jenž uživatel sám vyplňuje, a které jsou opatřené chybovými hláškami. Přidání atributu [ForeignKey] nad objekt typu „Uživatel“ slouží pro označení cizího klíče do příslušné tabulky pro Entity Framework.

Každá akce má název, krátký popis, který je zobrazen v seznamu akcí, dlouhý popis, který se zobrazuje v následném detailu akce, rozepsané datum konání akce, štítek, který určuje typ akce (oddílová, středisková nebo celorepubliková), autora dané akce a označení, zda se jedná o akci aktuální, nebo archivní.

### Kód 3: Model „Akce“

```
public class Akce
{
    75 references
    public Guid AkceId { get; set; }

    [Required(ErrorMessage = "Akce musí mít název")]
    [StringLength(100)]
    25 references
    public string Nazev { get; set; }

    [Required(ErrorMessage = "Akce musí mít základní popis")]
    [StringLength(500)]
    8 references
    public string KratkyPopis { get; set; }

    [Required(ErrorMessage = "Akce musí mít popis")]
    [StringLength(10000)]
    9 references
    public string DlouhyPopis { get; set; }

    [Required(ErrorMessage = "Zadejte den konání akce")]
    20 references
    public int DenKonani { get; set; }

    [Required(ErrorMessage = "Zadejte měsíc konání akce")]
    22 references
    public string MesicKonani { get; set; }

    [Required(ErrorMessage = "Zadejte rok konání akce")]
    25 references
    public int RokKonani { get; set; }
    52 references
    public string Stitek { get; set; }

    [ForeignKey(nameof(Uzivatel))]
    5 references
    public Guid AutorId { get; set; }
    1 reference
    public Uzivatel Uzivatel { get; set; }
    6 references
    public List<PovolenaDruzina> PovoleneDruziny { get; set; }
    7 references
    public bool ArchivniAkce { get; set; }
}
```

Zdroj: (Autor)

#### 4.6.1.2 Repozitář

Pro práci s daty jednotlivých akcí v Controlleru byl využit repozitář „AkceRepository“, který implementuje rozhraní „IAkceRepository“, v němž jsou definovány metody tohoto repozitáře. Pro práci s databází v „AkceRepository“ byla pomocí Dependency Injection přidána kontextová třída databáze, která se v projektu nazývá „ModrinkyDbContext“. Na tomto principu fungují i všechny ostatní repozitáře aplikace.

Pro účely funkcionality seznamu akcí je využita metoda „VsechnyAkce“, která vrací list aktuálních akcí (Kód 4).

Kód 4: Repozitář „AkceRepository“ – Metoda „VsechnyAkce“

```
public class AkceRepository : IAkceRepository
{
    private readonly ModrinkyDbContext dbContext;

    1 reference
    public AkceRepository(ModrinkyDbContext dbContext)
    {
        this.dbContext = dbContext;
    }
    11 references
    public List<Akce> VsechnyAkce()
    {
        return dbContext.Akce.Where(x => x.ArchivniAkce == false).ToList();
    }
}
```

Zdroj: (Autor)

#### 4.6.1.3 ViewModel

Protože ve View seznamu akcí je potřeba zobrazovat i jiná data, než pouze obsah tabulky „Akce“, bylo využito při implementaci stránky „Seznam akcí“ ViewModelu „AkceViewModel“ (Kód 5). Ten obsahuje list zobrazovaných akcí a pomocné vlastnosti typu *string*, které slouží k uchování hodnot zadaných do vyhledávacího formuláře.

Kód 5: ViewModel „AkceViewModel“

```
6 references
public class AkceViewModel
{
    5 references
    public List<Akce> Akce { get; set; }
    3 references
    public string NazevAkceHledani { get; set; }
    3 references
    public string MesickHledani { get; set; }
    3 references
    public string RokKHledani { get; set; }
}
```

Zdroj: (Autor)

#### 4.6.1.4 Controller

Controller propojující Model „Akce“ a View, který samotný seznam akcí vykresluje uživateli, se nazývá „AkceController“. Protože seznam akcí není přístupný nepřihlášeným uživatelům, je celá třída „AkceController“ označena atributem [Authorize], který zajišťuje nutnost přihlášení.

Repozitáře jsou do Controlleru přidány vlastnostmi pomocí Dependency Injection přes rozhraní IUnitOfWork, které obsahuje deklaraci jednotlivých repozitářů v projektu.

Metoda pro vykreslení samotného View seznamu akcí se nazývá „SeznamAkci“ a je v Controlleru dvakrát. První metoda (Kód 6), která nepřijímá žádný parametr je spuštěna při HTTP metodě GET. Obsahuje deklaraci instance AkceViewModelu a funkci pro spuštění archivace, která veškeré akce, které již nejsou aktuální označí jako archivní a zamezí tak jejich zobrazení v seznamu akcí. Akce k zobrazení jsou uloženy do listu „Akce“ obsaženého ve ViewModelu a jsou pomocí jazyka LINQ seřazeny podle data konání. Pro řazení pomocí měsíce konání je využita statická třída „DatumPrevody“, která obsahuje metodu pro převod měsíce zadaného slovem na jeho číselné označení.

Pokud se v databázi nenacházejí žádné akce, je uživatel o této skutečnosti informován pomocí dočasné zprávy ve View.

Metoda vrací View se shodným názvem, jako je název akce, a s instancí ViewModelu.

Kód 6: Metoda „SeznamAkci“

```
0 references
public IActionResult SeznamAkci()
{
    var akceViewModel = new AkceViewModel();

    Archivace();

    akceViewModel.Akce = akceRepository.VsechnyAkce().
        OrderBy(ak => ak.RokKonani)
        .ThenBy(ak => int.Parse(DatumPrevody.MesicCiselne(ak.MesicKonani)))
        .ThenBy(ak => ak.DenKonani)
        .ToList();

    if(akceViewModel.Akce.Count() == 0)
    {
        TempData["Zprava"] = "Žádné akce k zobrazení.";
    }

    return View(akceViewModel);
}
```

Zdroj: (Autor)

Druhá metoda „SeznamAkci“ (Kód 7) se zavolá při HTTP metodě POST, tedy při žádosti o odeslání dat na server. To nastane, využije-li uživatel vyhledávacího formuláře. Tato metoda navíc přijímá parametr typu „AkceViewModel“.

Nejdříve je otestováno, zda obsah jednotlivých polí vyhledávacího formuláře není nulový (v ukázkovém kódu zobrazeno pouze pro rok konání akce). Následně jsou pomocí LINQ vybrány ty akce, které odpovídají zadaným parametrům ve vyhledávacím formuláři. Parametry jsou převedeny na velká písmena, aby vyhledávání nebylo case-sensitive. Seřazení akcí je řešeno stejně jako v předchozí metodě. Metoda vrací stejnojmenné View a odesílá mu instanci ViewModelu „AkceViewModel“. Pokud vyhledávacím parametrům neodpovídají žádné akce, ve View se o této skutečnosti zobrazí dočasná zpráva.

#### Kód 7: Metoda „SeznamAkci“ POST

```
if (akceViewModelPost.RokKHledani != null)
{
    rokKHledani = akceViewModelPost.RokKHledani.ToUpper();
}
else
{
    rokKHledani = null;
}

var vsechnyAkce = akceRepository.VsechnyAkce();
IEnumerable<Akce> akce = from a in vsechnyAkce
                        where
                            (
                                (string.IsNullOrEmpty(nazevAkceHledani) || a.Nazev.ToUpper().Contains(nazevAkceHledani))
                                &&
                                (a.MesicKonani.ToUpper() == mesicKHledani || string.IsNullOrEmpty(mesicKHledani))
                                &&
                                (a.RokKonani.ToString() == rokKHledani || string.IsNullOrEmpty(rokKHledani))
                            )
                        orderby a.Nazev
                        select a;

var akceViewModel = new AkceViewModel();
akceViewModel.Akce = akce.OrderBy(ak => ak.RokKonani)
                        .ThenBy(ak => int.Parse(DatumPrevody.MesicCiselo(ak.MesicKonani)))
                        .ThenBy(ak => ak.DenKonani).ToList();

if (akceViewModel.Akce.Count == 0)
{
    TempData["Zprava"] = "Žádné akce k zobrazení.";
}

return View(akceViewModel);
```

Zdroj: (Autor)

#### 4.6.1.5 View

Aby ASP.NET rozpoznal, který View má po provedení metody v Controlleru spustit, je View pro seznam akcí pojmenován „SeznamAkci“, tedy stejně jako daná metoda v Controlleru.

V první části View „SeznamAkci“ (Kód 8) je pomocí @model určeno, že se zde bude pracovat s modelem typu „AkceViewModel“. Na tento model je dále odkazováno pomocí klíčového slova Model. Direktiva @using přidává knihovnu pro práci s ASP.NET Core Identity a pomocí @inject jsou připojeny potřebné repositáře a objekty pro práci



s ASP.NET Core Identity. Pro další práci ve View je určena role přihlášeného uživatele, aktuální rok a další dva roky po něm následující.

Kód 8: View „SeznamAkci“ – Úvodní deklarace

```
@model AkceViewModel

@using Microsoft.AspNetCore.Identity
@inject SignInManager<Uzivatel> SignInManager
@inject UserManager<Uzivatel> UserManager
@inject IRepository roleRepository
@inject IUzivatelRepository uzivatelRepository

@{
    string prihlasenyRole = "";
    if (SignInManager.IsSignedIn(User))
    {
        Guid prihlasenyUserId = Guid.Parse(UserManager.GetUserId(User));
        int prihlasenyRoleId = uzivatelRepository.UzivatelPodleId(prihlasenyUserId).RoleId;
        prihlasenyRole = roleRepository.RolePodleId(prihlasenyRoleId).Nazev;
    }

    int rok1 = System.DateTime.Now.Year;
    int rok2 = rok1 + 1;
    int rok3 = rok2 + 1;
}

<h1 class="text-center">Seznam akcí</h1>
<br />
```

Zdroj: (Autor)

Po nadpisu stránky je vymezen prostor pro případnou dočasnou zprávu. Formulář pro vyhledávání mezi akcemi je vymezen HTML značkou <form></form> a označen metodou POST. Jednotlivá pole formuláře slouží pro filtraci pomocí názvu akce, měsíce hledání (výběrové pole s výběrem jednotlivých měsíců, pro účel ukázky je výčet měsíců skryt) a roku hledání, kde je na výběr mezi aktuálním rokem a dvěma roky po něm následujícími. Všechna tři formulářová pole obsahují defaultní text přidáný pomocí atributu *placeholder*. Pro spuštění filtrace slouží tlačítko „Hledat“ a pro resetování vstupů tlačítko „Reset“. Grafické prvky v tomto View obsahují třídy frameworku Bootstrap, který slouží ke stylizování a který je používán i dále v projektu (Kód 9).

### Kód 9: View „SeznamAkci“ – Vyhledávací formulář

```
@if (TempData["Zprava"] != null)
{
    <div class="alert alert-info">@TempData["Zprava"]</div>
}

<form id="vyhledavani" method="post" class="form-inline justify-content-center">
    <span class="form-group mx-sm-3 mb-2">
        <input id="navez" type="search" class="form-control" asp-for="@Model.NazevAkceHledani" placeholder="Název akce" />
    </span>
    <span class="form-group mx-sm-3 mb-2">
        <select id="mesic" class="form-control">...</select>
    </span>
    <span class="form-group mx-sm-3 mb-2">
        <select id="rok" class="form-control" asp-for="@Model.RokKHledani" placeholder="Měsíc konání">
            <option selected disabled>Rok</option>
            <option>@rok1</option>
            <option>@rok2</option>
            <option>@rok3</option>
        </select>
    </span>
    <button type="submit" class="btn btn-primary mb-2">Hledat</button>
    <button class="btn btn-default mx-sm-3 mb-2" id="reset">Reset</button>
</form>
```

Zdroj: (Autor)

Odkaz pro přidání akce je zobrazen pouze je-li přihlášený uživatel v roli admin nebo vedoucí. V jiném případě tento odkaz není vůbec vykreslen. Odkaz uživatele přesměruje na akci v Controlleru „AkceController“, která obsluhuje funkcionalitu přidávání akce (Kód 10).

### Kód 10: View „SeznamAkci“ – Odkaz pro přidání akce

```
@if (SignInManager.IsSignedIn(User) && prihlasenyRole == "vedouci" || prihlasenyRole == "admin")
{
    <span class="pridani"><i class="fas fa-plus"></i> <a asp-controller="Akce" asp-action="PridaniAkce" role="button">Přidat akci</a></span>
    <br />
    <br />
}
}
```

Zdroj: (Autor)

Samotný výčet jednotlivých akcí je zajištěn pomocí cyklu *foreach*. Ten iteruje mezi jednotlivými akcemi v listu „Akce“ získaného z instance ViewModelu „AkceViewModel“. Pro každou akci je nejprve vypsáno datum přidání a štítek neboli označení typu akce.

Podle tohoto štítku je pak také pomocí *if* a *else if* vybrána barva názvu dané akce. Název akce slouží jako odkaz vedoucí na metodu „DetailAkce“ v Controlleru „AkceController“ (Kód 11).

## Kód 11: View „SeznamAkci“ – Jednotlivé akce

```
@foreach (var a in Model.Akce)
{
    <div class="card border border-dark rounded-lg">
    <div class="card-header">
        <b> @a.Denkonani.ToString(). @a.Mesickonani @a.Rokkonani.ToString() </b>@a.Stitek
        <br />
        <br />
        @if (a.Stitek == "Oddílová akce")
        {
            <h4><a style="color:cornflowerblue" asp-controller="Akce" asp-action="DetailAkce" asp-route-id="@a.AkceId" role="button">@a.Nazev</a></h4>
        }
        else if (a.Stitek == "Středisková akce")
        {
            <h4><a style="color:firebrick" asp-controller="Akce" asp-action="DetailAkce" asp-route-id="@a.AkceId" role="button">@a.Nazev</a></h4>
        }
        else if (a.Stitek == "Celorepubliková akce")
        {
            <h4><a style="color:darkcyan" asp-controller="Akce" asp-action="DetailAkce" asp-route-id="@a.AkceId" role="button">@a.Nazev</a></h4>
        }
    </div>
    <div class="card-body">
        <p>
            @a.KratkyPopis
            <br />
            <br />
            @if (SignInManager.IsSignedIn(User) && (a.AutorId == Guid.Parse(UserManager.GetUserId(User)) || prihlasenyRole == "admin"))
            {
                <a asp-controller="Akce" asp-action="EditaceAkce" asp-route-id="@a.AkceId" role="button"><i class="fas fa-edit"></i></a>
                <a asp-controller="Akce" asp-action="OdebraniAkce" asp-route-id="@a.AkceId" role="button"><i class="fas fa-trash-alt"></i></a>
                <a asp-controller="Akce" asp-action="KdoJedeNaAkci" asp-route-id="@a.AkceId" role="button">Kdo jede?</a>
            }
        </p>
    </div>
</div>
<br />
}
```

Zdroj: (Autor)

Funkcionalita tlačítka „Reset“ je zajištěna skriptem napsaným v JavaScriptu (Kód 12), který navrátí hodnoty polí vyhledávacího formuláře na prázdné a defaultní hodnoty.

## Kód 12: View „SeznamAkci“ – JavaScript

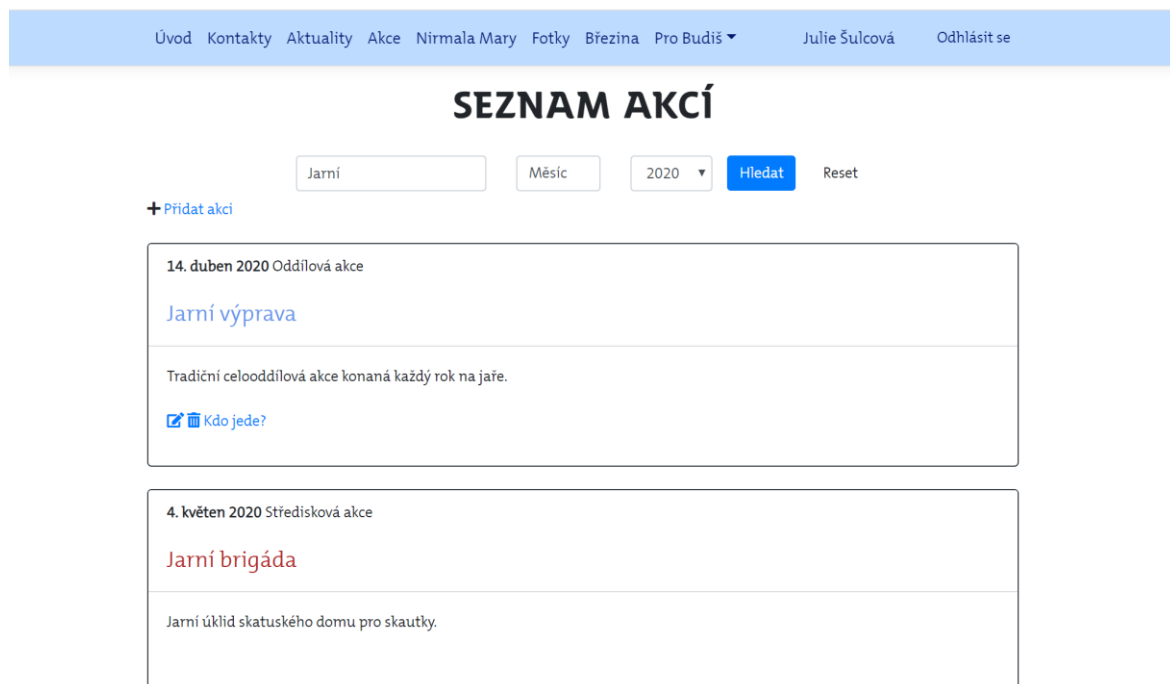
```
<script type="text/javascript">
    $(document).ready(function () {
        $("#reset").on
        ("click",function()
        {
            $("#mesic").val($("#mesic option:first").val());
            $("#rok").val($("#rok option:first").val());
            $("#nazev").val($("#nazev :input[placeholder]").val());
        }
    });
</script>
```

Zdroj: (Autor)

### 4.6.1.6 Výsledná obrazovka

Výstup View „SeznamAkci“ byl pomocí souboru kaskádových stylů určených pro celý projekt dotvořen do výsledné podoby dle skautského grafického manuálu (85). V ukázce (Obrázek 14) je stránka „Seznam akcí“ po provedení filtrace akcí přihlášeným fiktivním uživatelem v roli vedoucí, který je autorem jedné z akcí.

Obrázek 14: Výsledná obrazovka seznamu akcí



Úvod Kontakty Aktuality Akce Nirmala Mary Fotky Březina Pro Budiš Julie Šulcová Odhlásit se

## SEZNAM AKCÍ

Jarní Měsíc 2020 Hledat Reset

+ Přidat akci

14. duben 2020 Oddílová akce

Jarní výprava

Tradiční celooddílová akce konaná každý rok na jaře.

[Kdo jede?](#)

4. květen 2020 Středisková akce

Jarní brigáda

Jarní úklid skatuského domu pro skautky.

Zdroj: (Autor)

### 4.6.2 Moje družina

Následující část popisu implementace stránky „Moje družina“, která je zobrazitelná pro uživatele v roli člen a pro rodiče členů, které jsou součástí konkrétní družiny. Slouží k informování o akcích pro družinu a k možnosti kontaktování případného vedoucího dané družiny.

#### 4.6.2.1 Model

Primárním modelem využívaným v této části je model „Družina“ (Kód 13). Obsahuje jednoznačný identifikátor celočíselného typu, název, krátkou informaci o dané družině a cizí klíč určující vedoucího družiny. Družina nemusí mít povinně vedoucího přiřazeného, proto má tento cizí klíč datový typ nepovinného *Guid*.

Kód 13: Model „Druzina“

```
public class Druzina
{
    19 references
    public int DruzinaId { get; set; }
    5 references
    public string Nazev { get; set; }
    1 reference
    public string Informace { get; set; }

    [ForeignKey(nameof(Uzivatel))]
    11 references
    public Guid? VedouciDruzinyId { get; set; }
    1 reference
    public Uzivatel Uzivatel { get; set; }
}
```

Zdroj: (Autor)

Dalšími Modely využitými v této části jsou Modely „Uzivatel“ (Kód 14) a „Clen“ (Kód 15). Ty obsahují některé stejné vlastnosti, konkrétně jméno a cizí klíč k tabulce „Druzina“.

„Clen“ má navíc informaci o uživatelském účtu, který ho spravuje a pomocnou vlastnost „Prihlasen“ používanou při přihlašování na akce. „Uzivatel“ dále obsahuje informaci o roli, pod kterou v systému vystupuje.

Samotná tabulka „Uzivatel“ obsahuje ještě další atributy, které nejsou v Modelu specifikovány, protože je generována pomocí ASP.NET Core Identity.

Kód 14: Model „Uzivatel”

```
public class Uzivatel : IdentityUser<Guid>
{
    [Required(ErrorMessage = "Vyplnění jména je vyžadováno")]
    [StringLength(25)]
    29 references
    public string KrestniJmeno { get; set; }

    [Required(ErrorMessage = "Vyplnění příjmení je vyžadováno")]
    [StringLength(25)]
    29 references
    public string Prijmeni { get; set; }

    [ForeignKey(nameof(Role))]
    48 references
    public int RoleId { get; set; }
    1 reference
    public Role Role { get; set; }

    [ForeignKey(nameof(Druzina))]
    30 references
    public int? DruzinaId { get; set; }
    1 reference
    public Druzina Druzina { get; set; }
}
```

Zdroj: (Autor)

Kód 15: Model „Clen“

```
public class Clen
{
    43 references
    public Guid ClenId { get; set; }

    [Required(ErrorMessage = "Člen musí mít jméno")]
    [StringLength(25)]
    23 references
    public string KrestniJmeno { get; set; }

    [Required(ErrorMessage = "Člen musí mít příjmení")]
    [StringLength(25)]
    22 references
    public string Prijmeni { get; set; }

    [ForeignKey(nameof(Uzivatel))]
    23 references
    public Guid SpravceId { get; set; }
    1 reference
    public Uzivatel Uzivatel { get; set; }
    3 references
    public bool Prihlasen { get; set; } = false;

    [ForeignKey(nameof(Druzina))]
    32 references
    public int? DruzinaId { get; set; }
    1 reference
    public Druzina Druzina { get; set; }
}
```

Zdroj: (Autor)

Stránka „Moje družina“ využívá dále již dříve popsany Model „Akce“.

#### 4.6.2.2 Repozitář

V této webové aplikaci je pro každý Model vytvořen i repozitář a interface, který repozitář implementuje. Repozitáře jsou vytvořeny tedy i pro výše uvedené Modely. Z nich tato část aplikace využívá několik metod, které jsou znázorněny v následujících ukázkách kódu.

V „AkceRepository“ je to metoda „AkcePodleId“ (Kód 16), která vrací konkrétní akci nalezenou podle přijatého parametru identifikátoru akce. Stejným principem jsou implementovány i metody „UzivatelPodleId“ a „ClenPodleId“ v repozitářích „UzivatelRepository“ a „ClenRepository“.

Kód 16: Repozitář „AkceRepository“ – Metoda „AkcePodleId“

```
public Akce AkcePodleId(Guid akceId)
{
    return dbContext.Akce.Include(x => x.PovoleneDruziny).FirstOrDefault(a => a.AkceId == akceId);
}
```

Zdroj: (Autor)

„ClenRepository“ (Kód 17) dále obsahuje mimo jiné metody „VsichniClenove“ a „ClenoveUzivatele“. První z nich vrací kompletní list členů v databázi a druhá list členů, které spravuje zadaný uživatel. „UzivatelRepository“ a „DruzinaRepository“ pak obsahují metody „VsichniUzivatele“ a „VsechnyDruziny“ založené na stejném principu jako metoda „VsichniClenove“.

Kód 17: Repozitář „ClenRepository“

```
11 references
public List<Clen> VsichniClenove()
{
    return dbContext.Clen.ToList();
}

7 references
public List<Clen> ClenoveUzivatele(Guid uzivatelId)
{
    return dbContext.Clen.Where(x => x.SpravceId == uzivatelId).ToList();
}
```

Zdroj: (Autor)

#### 4.6.2.3 ViewModel

ViewModel využít k implementaci části „Moje družina“ se nazývá „MojeDruzinaViewModel“. Obsahuje vlastnost typu boolean určující, zda má být uživateli zobrazena informace o družině, nebo o tom, že uživatel či člen se zatím v žádné družině nenacházejí. Dále ViewModel obsahuje konkrétní družinu, jejíž informace budou vypsány, vedoucího této družiny, pokud existuje, list uživatelů a členů, kteří jsou v této družině, akce, kterých se členové družiny mohou zúčastnit a vlastnost pro text e-mailu, který se dá pak v samotném View uživatelem vyplnit.

Pokud je přihlášený uživatel v roli rodič, je potřebná také informace o členovi, přes kterého je tato stránka zobrazována.



Kód 18: ViewModel „MojeDruzinaViewModel“

```
public class MojeDruzinaViewModel
{
    3 references
    public bool MaDruzinu { get; set; }
    8 references
    public Druzina Druzina { get; set; }
    7 references
    public Uzivatel VedouciDruziny { get; set; }
    3 references
    public List<Uzivatel> UzivateleVDruzine { get; set; }
    4 references
    public Clen Clen { get; set; }
    3 references
    public List<Clen> ClenoveVDruzine { get; set; }
    4 references
    public List<Akce> AkceDruziny { get; set; }

    [Required(ErrorMessage = "Email musí mít text")]
    [StringLength(1000)]
    3 references
    public string TextEmailu { get; set; }
}
```

Zdroj: (Autor)

#### 4.6.2.4 Controller

Controller, obsahující metodu, která zobrazuje samotné View pro stránku „Moje družina“ se nazývá „MojeDruzinaController“. Zde jsou opět pomocí Dependency Injection přidány jednotlivé repozitáře a třída „UnitOfWork“. Obsah stránky „Moje družina“ je přístupný pouze přihlášeným uživatelům, tedy je Controller označen atributem [Authorize].

Metoda pro zobrazení stránky „Moje družina“ je pojmenována „DetailDruziny“ a opět je v Controlleru obsažena dvakrát. Jednou pro metodu GET a podruhé pro metodu POST.

V první části první metody „DetailDruziny“ (Kód 19) je nejdříve zjištěno id družiny, ke které je přihlášený uživatel přiřazen. Následně je zajištěno, aby vedoucím a adminům bylo zamítnuto zobrazení stránky „Moje družina“ a uživatelům bez družiny a rodičům bez členů byla zobrazena stránka „Moje družina“ s informací o neexistující družině.

Přijímaný parametr id slouží k identifikaci člena, přes kterého je stránka zobrazována, je-li přihlášený uživatel v roli rodič.

### Kód 19: Metoda „DetailDruziny” – Ošetření existence družiny

```
public IActionResult DetailDruziny(Guid id)
{
    Guid prihlasenyUserId = Guid.Parse(userManager.GetUserId(User));
    Uzivatel prihlaseny = uzivatelRepository.UzivatelPodleId(prihlasenyUserId);
    int? druzinaId = prihlaseny.DruzinaId;

    MojeDruzinaViewModel mojeDruzina = new MojeDruzinaViewModel();

    if(VratRoli() == "vedoucí" || VratRoli() == "admin")
    {
        return View("PristupZakazan");
    }

    if(
        (VratRoli() == "člen" && druzinaId == null)
        ||
        (VratRoli() == "rodič" && !clenRepository.ClenoveUzivatele(prihlasenyUserId).Any())
    )
    {
        mojeDruzina.MaDruzinu = false;
        return View(mojeDruzina);
    }
}
```

Zdroj: (Autor)

Poté je určena konkrétní družina, která bude předmětem stránky „Moje družina“ a je ošetřeno, že družina zobrazovaná přes člena rodiče existuje. Podle zjištěné družiny je určen vedoucí této družiny. (Kód 20)

### Kód 20: Metoda „DetailDruziny” – Zjištění konkrétní družiny

```
else
{
    mojeDruzina.MaDruzinu = true;
    mojeDruzina.UzivateleVDruzine = new List<Uzivatel>();
    mojeDruzina.ClenoveVDruzine = new List<Clen>();
    mojeDruzina.AkceDruziny = new List<Akce>();

    if(VratRoli() == "rodič")
    {
        mojeDruzina.Clen = clenRepository.ClenPodleId(id);
        mojeDruzina.Druzina = druzinaRepository.VsechnyDruziny()
            .FirstOrDefault(x => x.DruzinaId == mojeDruzina.Clen.DruzinaId);

        if(mojeDruzina.Druzina == null)
        {
            mojeDruzina.MaDruzinu = false;
            return View(mojeDruzina);
        }
    }
    else if(VratRoli() == "člen")
    {
        mojeDruzina.Druzina = druzinaRepository.VsechnyDruziny().FirstOrDefault(x => x.DruzinaId == druzinaId);
    }

    mojeDruzina.VedouciDruziny = uzivatelRepository.UzivatelPodleId(mojeDruzina.Druzina.VedouciDruzinyId);
}
```

Zdroj: (Autor)

Listy akcí, uživatelů a členů jsou vytvořeny pomocí tří *foreach* cyklů, kde je ošetřeno, zda daný uživatel nebo člen jsou přiřazeny k určené družině, a zda je akce pro tuto družinu určena. U akce je dále zkontrolováno, zda se nejedná o již archivní akci. (Kód 21)

Kód 21: Metoda "DetailDruziny" – Naplnění listů

```
        foreach(var u in uzivatelRepository.VsichniUzivatele())
        {
            if(u.DruzinaId == mojeDruzina.Druzina.DruzinaId)
            {
                mojeDruzina.UzivateleVDruzine.Add(u);
            }
        }

        foreach(var c in clenRepository.VsichniClenove())
        {
            if(c.DruzinaId == mojeDruzina.Druzina.DruzinaId)
            {
                mojeDruzina.ClenoveVDruzine.Add(c);
            }
        }

        foreach(var p in povolenaDruzinaRepository.VsechnyPovoleneDruziny())
        {
            if(p.DruzinaId == mojeDruzina.Druzina.DruzinaId)
            {
                Akce akce = akceRepository.AkcePodleId(p.AkceId);
                if (!akce.ArchivniAkce)
                {
                    mojeDruzina.AkceDruziny.Add(akce);
                }
            }
        }
    }
    return View(mojeDruzina);
}
```

Zdroj: (Autor)

Druhá metoda „DetailDruziny“ (Kód 22), která je volána při metodě POST, obsahuje především funkcionalitu pro odeslání e-mailu uživatelem vedoucímu družiny. Nejdříve je zjištěna informace o přihlášeném uživateli a vedoucím družiny a následně je asynchronně zavolána metoda „EmailVedouci“ přijímající jako parametry přihlášeného uživatele, vedoucího družiny a text e-mailu, který byl ve View zadán uživatelem.

#### Kód 22: Metoda „DetailAkce“ POST

```
[HttpPost]
0 references
public IActionResult DetailDruziny(MojeDruzinaViewModel mojeDruzina)
{
    Guid prihlasenyUserId = Guid.Parse(userManager.GetUserId(User));
    Uzivatel prihlaseny = uzivatelRepository.UzivatelPodleId(prihlasenyUserId);
    Uzivatel vedouciDruziny = uzivatelRepository.UzivatelPodleId(mojeDruzina.VedouciDruziny.Id);

    EmailVedouci(prihlaseny, vedouciDruziny, mojeDruzina.TextEmailu).Wait();

    TempData["Zprava"] = "Zpráva úspěšně odeslána";

    return RedirectToAction("DetailDruziny", mojeDruzina.Clen.ClenId);
}
```

Zdroj: (Autor)

Metoda „EmailVedouci“ (Kód 23) vytvoří nového klienta pomocí apiKey získaného ze stránky Azure. Samotný e-mail se skládá z adresáta, příjemce, předmětu a textu. Posledním parametrem přijímaným metodou „CreateSingleEmail“ je text v HTML formátu, který zde není využit.

#### Kód 23: Metoda „EmailVedouci“

```
1 reference
static async Task EmailVedouci(Uzivatel odesilatel, Uzivatel vedouciDruziny, string text)
{
    var klient = new SendGridClient(ApiKey);
    var od = new EmailAddress(odesilatel.Email);
    var pro = new EmailAddress(vedouciDruziny.Email);
    var predmet = "Zpráva ze stránek 9. oddílu Modřinky";

    var msg = MailHelper.CreateSingleEmail
    (
        od, pro, predmet, text, null
    );

    await klient.SendEmailAsync(msg);
}
```

Zdroj: (Autor)

#### 4.6.2.5 View

V první části View „DetailDruzinyView“ (Kód 24) je určeno, že bude využíváno „MojeDruzinaViewModel“. Pokud má vlastnost tohoto ViewModelu „MaDruzinu“ hodnotu *true*, je zobrazován obsah o družině. Jméno vedoucího družiny je zobrazeno pouze pokud je nějaký vedoucí přiřazen.

## Kód 24: View „MojeDruzina“ – Popis družiny

```
@model MojeDruzinaViewModel
@{
    ViewData["Title"] = "MojeDruzina";
}

@if (Model.MaDruzinu)
{
    <h1 class="text-center">@Model.Druzina.Nazev</h1>

    @if (TempData["Zprava"] != null)
    {
        <div class="alert alert-info">@TempData["Zprava"]</div>
    }

    <p><b>Popis:</b> @Model.Druzina.Informace</p>
    if (Model.VedouciDruziny != null)
    {
        <p><i class="fas fa-compass"></i> <b>Vedoucí družiny:</b> @Model.VedouciDruziny.KrestniJmeno @Model.VedouciDruziny.Prijmeni</p>
    }
    <p><i class="fas fa-child"></i> <b>Seznam členů:</b></p>
    <ul>
        @foreach (var u in Model.UzivateleVDruzine)
        {
            <li>@u.KrestniJmeno @u.Prijmeni</li>
        }
        @foreach (var c in Model.ClenoveVDruzine)
        {
            <li>@c.KrestniJmeno @c.Prijmeni</li>
        }
    </ul>
}
```

Zdroj: (Autor)

Družiny určené pro akci slouží zároveň jako odkazy na detaily těchto akcí, které si uživatel může rozkliknout (Kód 25).

## Kód 25: View „MojeDruzina“ – Seznam akcí pro družinu

```
@if (Model.AkceDruziny.Any())
{
    <p><i class="fas fa-campground"></i> <b>Seznam akcí určených pro družinu:</b></p>
    <ul>
        @foreach (var a in Model.AkceDruziny)
        {
            <li>
                <a style="color:cornflowerblue" asp-controller="Akce" asp-action="DetailAkce" asp-route-id="@a.AkceId" role="button">@a.Nazev</a>
            </li>
        }
    </ul>
}
```

Zdroj: (Autor)

V poslední části „DetailDruzinyView“ (Kód 26) je řešena problematika odesílání e-mailu vedoucímu pomocí formuláře s textovým polem pro obsah e-mailu a s tlačítkem „Odeslat“. Při chybném vyplnění formuláře je vypsána validační hláška definována v „MojeDruzinaViewModel“. Formulář dále obsahuje neviditelná pole, která slouží pro přenos informace o id vedoucího družiny a id člena, přes kterého je stránka uživateli v roli rodič zobrazována.

Pokud „MaDruzinu“ má hodnotu *false*, obsah o družině není zobrazen.

#### Kód 26: View MojeDruzina – Odesílání e-mailu

```
3  @if (Model.VedouciDruziny != null)
    {
3      <p>
        <i class="fas fa-pen"></i> <b>Napsat vedoucí družiny:</b>
3      </p>
        <form asp-action="DetailDruziny" method="post" role="form">
3          <div class="form-group">
                <label>Text emailu</label>
                <textarea asp-for="@Model.TextEmailu" class="form-control" rows="3"></textarea>
                <span asp-validation-for="@Model.TextEmailu" class="text-danger"></span>
            </div>
            <input type="hidden" asp-for="@Model.VedouciDruziny.Id" />
            <input type="hidden" asp-for="@Model.Clen.ClenId" />
3          <button type="submit" class="btn btn-primary">
                Odeslat
            </button>
        </form>
    }
}
else
{
    <h1 class="text-center">Žádný obsah k zobrazení</h1>
}
```

Zdroj: (Autor)

#### 4.6.2.6 Výsledná obrazovka

Výsledek implementace části „Moje družina“ je znázorněn na následujícím obrázku (Obrázek 15).

Obrázek 15: Výsledná obrazovka informací o družině

Úvod Kontakty Aktuality Akce Nirmala Mary Fotky Březina Pro členy ▾ Elena Fidlerová Odhlásit se

# MODRÁSCI

Popis: Družina pro děti od 7 do 9 let

🔗 Vedoucí družiny: Alžběta Pokorná

👤 Seznam členů:

- Eliška Kavková
- Elena Fidlerová
- Tamara Jirková
- Jitka Čevelová

📅 Seznam akcí určených pro družinu:

- [Nirmala akce](#)
- [Jarní výprava](#)

✍️ Napsat vedoucí družiny:

Text emailu

Odeslat

Zdroj: (Autor)

## 5 Výsledky a diskuse

### 5.1 Testování webové aplikace

Před zahájením testování byla dokončená webová aplikace nasazena a je dostupná na webové adrese <https://modrinkytabor.azurewebsites.net/>. Testování se zúčastnilo šest subjektů, a to jak přímo z oddílu, pro který byla webová aplikace vyvíjena, tak i z veřejnosti, aby bylo zjištěno, jak moc je aplikace intuitivní i pro případné nové členy, kteří se nevyznají ve zvyklostech oddílu. Dvěma testerům byl založen uživatelský účet v roli admin, dalším dvěma v roli vedoucí, jednomu jako rodič a poslednímu v roli člen. Registrovaní uživatelé si zároveň vyzkoušeli i pohyb po aplikaci bez přihlášení. Každý ze subjektů dostal za úkol vyzkoušet všechny prvky aplikace bez bližšího navádění.

Při testování se aplikace ukázala jako velmi intuitivní, nikdo ze subjektů neměl problém zadané úkoly zvládnout. Během procesu bylo objeveno několik chyb v podobě překlepů, které byly následně opraveny.

Ze strany uživatelů v roli člen a rodič přišly návrhy na umožnění některých funkcionalit, které mají zpřístupněny pouze vedoucí a admini, jako je zobrazení si účastníků akcí či možnost přiřadit sám sobě nebo svým členům družinu. Ocenili by také souhrnný kalendář, který by přesně vyznačoval, ve které dny se akce konají.

Uživatelé v roli vedoucí navrhovali přidat funkci, která by umožnila vyloučit člena z přihlášené akce, možnost upravit již proběhlé akce a přidávat fotografie a obrázky do aktualit oddílu.

Uživatelé v roli admin se shodli na tom, že jim chybí možnost vkládání obrázků a dalších dokumentů do zakládaných akcí a možnost kontaktovat každého vedoucího přímo ze stránek oddílu, ne pouze vedoucí jednotlivých družin.

Design aplikace byl všemi subjekty přijat pozitivně a bylo oceněno minimalistické pojetí a přehlednost.

Celkově byla aplikace zhodnocena jako produkt, který významně zjednoduší organizaci oddílu, evidenci členů a komunikaci mezi členy a vedoucími.

### 5.2 Zhodnocení ze strany autora

Vývoj webové aplikace pomocí ASP.NET Core nabízí rozsáhlé množství funkcionalit. Při dodržování konvencí framework pracuje v mnoha ohledech samostatně a sám například zvládá dohledat potřebné soubory i bez uvedení umístění. Díky ASP.NET Core Identity je



velmi snadno řešena problematika přihlašování uživatelů, je ale složitější, pokud vývojář chce upravit vzhled těchto funkcionalit.

ASP.NET Core je mezi vývojáři velmi rozšířen, neboť je dohledatelné velké množství zdrojů zabývajících se touto problematikou. I přes to, že je tento framework multiplatformní, není snadné najít možnost bezplatného hostingu bez použití kontejnerů, která by navíc podporovala i PostgreSQL databázi.

### **5.2.1 Návrhy na vylepšení**

Protože bude aplikace reálně využívána, budou po dohodě s vedením oddílu implementovány všechny návrhy na vylepšení ze strany testerů, kromě možnosti editace již archivovaných akcí, neboť tato funkcionality by se rozcházela s celkovým účelem archivu akcí.

Z poznatků získaných z analýzy dostupných řešení a z rozhovorů s testery autorka navrhuje přidání možnosti vedení evidence přijatých plateb, která by byla přístupná pouze vedoucím oddílu a adminům stránek. Dále také navrhuje vyšší využití schopnosti aplikace odesílat e-mailové zprávy a přidání hromadné obsílky informací o založení nové akce těm uživatelům a členům, kteří se této akce mohou na základě svých práv zúčastnit.

## 6 Závěr

Cílem této práce byla analýza, návrh a implementace webové aplikace s využitím ASP.NET. Dále měly být analyzovány požadavky na aplikaci a existující řešení a aplikace měla být nasazena, otestována a zhodnocena.

V teoretické části práce byla vymezena teoretická východiska potřebná pro následnou praktickou práci na zadané webové aplikaci.

V praktické části byly nejdříve splněny cíle, jako je analýza požadavků a řešení a následně byl za pomoci metod softwarového inženýrství proveden návrh samotné aplikace, který byl také součástí hlavního cíle práce. V kapitole „Implementace“ byla demonstrována implementace pomocí ukázek zdrojového kódu vytvořené aplikace.

Otestování a zhodnocení aplikace bylo provedeno v části „Výsledky a diskuse“, kde byla aplikace zhodnocena uživateli a testery jako přehledná a intuitivní a splňující cíl o usnadnění komunikace a organizace skautského oddílu. Aplikace také funguje jako informační portál pro veřejnost. Následně byly navrženy rozšiřující funkcionality, jako je přidání přehledného kalendáře, povolení části funkcí vedoucích i uživatelům v roli člen a uživatelům v roli rodič a vyšší využití možnosti e-mailové obsílky. Toto rozšíření bude provedeno v další etapě vývoje aplikace a není součástí této práce.

## 7 Seznam použitých zdrojů

1. CHRISTENSSON, Per. Web Application. *TechTerms* [online]. Sharpened Productions, [cit. 2020-01-13]. Dostupné z: [https://techterms.com/definition/web\\_application](https://techterms.com/definition/web_application)
2. Webová aplikace (Web Application). *Managemant Mania* [online]. [cit. 2020-01-13]. Dostupné z: <https://managementmania.com/cs/webova-aplikace-web-application>
3. What is a web server? *MDN Web Docs* [online]. [cit. 2020-01-13]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Learn/Common\\_questions/What\\_is\\_a\\_web\\_server](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server)
4. Web Server. *Tutorialspoint* [online]. Hyderabad: Tutorials Point, [cit. 2020-01-13]. Dostupné z: [https://www.tutorialspoint.com/internet\\_technologies/web\\_servers.htm](https://www.tutorialspoint.com/internet_technologies/web_servers.htm)
5. Application server vs. web server: differences and similarities. *Colocation America* [online]. [cit. 2020-01-13]. Dostupné z: <https://www.colocationamerica.com/blog/application-server-vs-web-server>
6. Informace o webových aplikacích. *Adobe* [online]. Adobe, [cit. 2020-01-13]. Dostupné z: <https://helpx.adobe.com/cz/dreamweaver/using/web-applications.html>
7. Tvorba www stránek – Statické stránky *Savvy* [online]. Brno: Savvy, [cit. 2020-01-13]. Dostupné z: <https://www.savvy.cz/webdesign/web-staticke-stranky>
8. Basic MVC Architecture. *Tutorialspoint* [online]. [cit. 2020-01-15]. Dostupné z: [https://www.tutorialspoint.com/struts\\_2/basic\\_mvc\\_architecture.htm](https://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm)
9. BERNARD, Borek. Úvod do architektury MVC. *Zdroják* [online]. [cit. 2020-01-15]. Dostupné z: <https://www.zdrojak.cz/clanky/uvod-do-architektury-mvc/>
10. GALLOWAY, Jon, Brad WILSON, K. Scott ALLEN a David MATSON. *Professional ASP.NET MVC 5*. Indianapolis, IN: Wrox, a Wiley brand, [2014]. Wrox professional guides. ISBN 978-1118794753.
11. ČÁPKA, David. MVC architektura. *ITnetwork* [online]. [cit. 2020-01-15]. Dostupné z: <https://www.itnetwork.cz/navrh/mvc-architektura-navrhovy-vzor>
12. What is MVC? Advantages and Disadvantages of MVC. *InterServer* [online]. [cit. 2020-01-15]. Dostupné z: <https://www.interserver.net/tips/kb/mvc-advantages-disadvantages-mvc/>

13. VATSA, Abhimanyu K. What is Model and ViewModel in MVC Pattern? *C# Corner* [online]. [cit. 2020-01-15]. Dostupné z: <https://www.c-sharpcorner.com/UploadFile/abhikumarvatsa/what-is-model-and-viewmodel-in-mvc-pattern/>
14. SHEKHAWAT, Sandeep Singh. CRUD using the Repository Pattern in MVC. *C# Corner* [online]. [cit. 2020-01-15]. Dostupné z: <https://www.c-sharpcorner.com/UploadFile/3d39b4/crud-using-the-repository-pattern-in-mvc/>
15. JASMINDER, Singh. Unit of Work in Repository Pattern. *C# Corner* [online]. [cit. 2020-01-15]. Dostupné z: <https://www.c-sharpcorner.com/UploadFile/b1df45/unit-of-work-in-repository-pattern/>
16. DHANANJAY, Kumar. How to Implement the Repository Pattern in ASP.NET MVC Application. *C# Corner* [online]. [cit. 2020-01-15]. Dostupné z: [https://www.infragistics.com/community/blogs/b/dhananjay\\_kumar/posts/how-to-implement-the-repository-pattern-in-asp-net-mvc-application](https://www.infragistics.com/community/blogs/b/dhananjay_kumar/posts/how-to-implement-the-repository-pattern-in-asp-net-mvc-application)
17. Dependency injection in ASP.NET Core. *Microsoft* [online]. [cit. 2020-01-15]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-3.1>
18. OJO, Femi. C# With Dependency Injection. *Codementor Community* [online]. [cit. 2020-01-15]. Dostupné z: <https://www.codementor.io/@mrfojo/c-with-dependency-injection-k2qfxbb8q>
19. Co je Dependency Injection? *PHP Fashion* [online]. [cit. 2020-01-15]. Dostupné z: <https://phpfashion.com/co-je-dependency-injection>
20. PRIBEAGU, Alin. Dependency Injection in ASP.NET MVC Tutorial. *Assist* [online]. [cit. 2020-01-15]. Dostupné z: <https://assist-software.net/blog/dependency-injection-aspnet-mvc-tutorial>
21. What is ASP.NET? *Microsoft* [online]. [cit. 2020-01-20]. Dostupné z: <https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet>
22. Introduction to Identity on ASP.NET Core. *Microsoft* [online]. [cit. 2020-01-20]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-3.1&tabs=visual-studio>
23. NuGet documentation. *Microsoft* [online]. [cit. 2020-01-21]. Dostupné z: <https://docs.microsoft.com/en-us/nuget/>

24. An introduction to NuGet. *Microsoft* [online]. [cit. 2020-01-21]. Dostupné z: <https://docs.microsoft.com/en-us/nuget/what-is-nuget>
25. ASP.NET MVC – Scaffolding. *Tutorialspoint* [online]. [cit. 2020-01-21]. Dostupné z: [https://www.tutorialspoint.com/asp.net\\_mvc/asp.net\\_mvc\\_scaffolding.htm](https://www.tutorialspoint.com/asp.net_mvc/asp.net_mvc_scaffolding.htm)
26. HERCEG, Tomáš. Build, .net core a na co se můžeme těšit: .NET 5.0? VÁŽNĚ? *DotNetPortal* [online]. 7. 5. 2019 [cit. 2020-01-21]. Dostupné z: <https://www.dotnetportal.cz/blogy/3/Tomas-Herceg/8571/Build-NET-Core-a-na-co-se-muzeme-tesit>
27. What is PostgreSQL? *PostgreSQL Tutorial* [online]. [cit. 2020-01-23]. Dostupné z: <https://www.postgresqltutorial.com/what-is-postgresql/>
28. *PostgreSQL* [online]. [cit. 2020-01-23]. Dostupné z: <https://postgres.cz/wiki/PostgreSQL>
29. Inherence. *PostgreSQL* [online]. [cit. 2020-01-24]. Dostupné z: <https://www.postgresql.org/docs/10/tutorial-inheritance.html>
30. Slovník. *PostgreSQL* [online]. [cit. 2020-01-24]. Dostupné z: <https://postgres.cz/wiki/Slovn%C3%ADk#MVCC>
31. KAPILA, Amit. Well-known Databases Use Different Approaches for MVCC. *The EDB Blog* [online]. March 18, 2015 [cit. 2020-01-24]. Dostupné z: <https://www.enterprisedb.com/blog/well-known-databases-use-different-approaches-mvcc>
32. What is Object Relational Mapping (ORM)? *Visual Paradigm Community Circle* [online]. [cit. 2020-01-24]. Dostupné z: <https://circle.visual-paradigm.com/docs/database-design-engineering/programmers-guide/what-is-object-relational-mapping-orm/>
33. HOYOS, Mario. What is an ORM and Why You Should Use it: An introduction to Object-Relational-Mappers. *Bits and Pieces* [online]. Dec 24, 2018 [cit. 2020-01-25]. Dostupné z: <https://blog.bitsrc.io/what-is-an-orm-and-why-you-should-use-it-b2b6f75f5e2a>
34. Entity Framework Documentation. *Microsoft* [online]. [cit. 2020-01-25]. Dostupné z: <https://docs.microsoft.com/en-us/ef/#pivot=entityfmwk>
35. Entity Framework Core. *Microsoft* [online]. [cit. 2020-01-25]. Dostupné z: <https://docs.microsoft.com/en-us/ef/core/>

36. Database Providers. *Microsoft* [online]. [cit. 2020-01-25]. Dostupné z:  
<https://docs.microsoft.com/en-us/ef/core/providers/index?tabs=dotnet-core-cli>
37. NoSQL with Entity Framework Core. *Entity Framework Core* [online]. [cit. 2020-01-25]. Dostupné z: <https://entityframeworkcore.com/knowledge-base/37570098/nosql-with-entity-framework-core>
38. Creating an Entity Data Model. *Entity Framework Tutorial* [online]. [cit. 2020-01-26]. Dostupné z: <https://www.entityframeworktutorial.net/entityframework6/create-entity-data-model.aspx>
39. Entity Framework – Data Model. *Tutorialspoint* [online]. [cit. 2020-01-26]. Dostupné z:  
[https://www.tutorialspoint.com/entity\\_framework/entity\\_framework\\_data\\_model.htm](https://www.tutorialspoint.com/entity_framework/entity_framework_data_model.htm)
40. LERMAN, Julia. *Programming entity framework*. Sebastopol, CA: O'Reilly, c2009. ISBN 059652028x.
41. ASP.NET / CODING / SQL Code-First vs Model-First vs Database-First: Pros and Cons: A comparison of three different ORM Data Modeling Approaches: Code-First, Model-First and Database-First. *Ryadel* [online]. June 17, 2018 [cit. 2020-01-26]. Dostupné z: <https://www.ryadel.com/en/code-first-model-first-database-first-vs-comparison-orm-asp-net-core-entity-framework-ef-data/>
42. Migrations. *Microsoft* [online]. [cit. 2020-01-26]. Dostupné z:  
<https://docs.microsoft.com/en-us/ef/core/managing-schemas/migrations/index?tabs=dotnet-core-cli>
43. Creating a Model. *Microsoft* [online]. [cit. 2020-01-26]. Dostupné z:  
<https://docs.microsoft.com/en-us/ef/ef6/modeling/index>
44. Working with DbContext. *Microsoft* [online]. [cit. 2020-01-26]. Dostupné z:  
<https://docs.microsoft.com/en-us/ef/ef6/fundamentals/working-with-dbcontext>
45. Entity Framework Core: DbContext. *Entity Framework Tutorial* [online]. [cit. 2020-01-26]. Dostupné z: <https://www.entityframeworktutorial.net/efcore/entity-framework-core-dbcontext.aspx>
46. DbSet in Entity Framework 6. *Entity Framework Tutorial* [online]. [cit. 2020-01-26]. Dostupné z:  
<https://www.entityframeworktutorial.net/entityframework6/dbset.aspx>

47. Work with SQL in ASP.NET Core. *Microsoft* [online]. [cit. 2020-01-26]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/working-with-sql?view=aspnetcore-3.1&tabs=visual-studio>
48. A tour of the C# language. *Microsoft* [online]. [cit. 2020-01-26]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/index>
49. ALBAHARI, Joseph a Ben ALBAHARI. *C# 7.0 in a nutshell*. 7th edition. Sebastopol: O'Reilly, 2018. ISBN 978-1-491-98765-0.
50. Attributes (C#). *Microsoft* [online]. [cit. 2020-01-26]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/attributes/>
51. Reflection (C#). *Microsoft* [online]. [cit. 2020-01-26]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/reflection>
52. Routing to controller actions in ASP.NET Core. *Microsoft* [online]. [cit. 2020-01-26]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/core/mvc/controllers/routing?view=aspnetcore-3.1>
53. IServiceProvider Interface. *Microsoft* [online]. [cit. 2020-01-26]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/api/system.iserviceprovider?view=netframework-4.8>
54. ČÁPKA, David. Lekce 7 - LINQ v C# – revoluce v dotazování. *ITnetwork* [online]. [cit. 2020-01-26]. Dostupné z: <https://www.itnetwork.cz/csharp/kolekce-a-linq/c-sharp-tutorial-linq-dotazy>
55. ČÁPKA, David. Lekce 8 - LINQ provideři, anonymní typy, řazení a seskupování. *ITnetwork* [online]. [cit. 2020-01-26]. Dostupné z: <https://www.itnetwork.cz/csharp/kolekce-a-linq/c-sharp-tutorial-linq-provideri-anonymni-typy-razeni-grupovani>
56. LINQ to Entities. *Microsoft* [online]. [cit. 2020-01-26]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/framework/data/adonet/ef/language-reference/linq-to-entities>
57. LINQ to Objects. *Microsoft* [online]. [cit. 2020-01-26]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/csharp/programming-guide/concepts/linq/linq-to-objects>

58. IQueryable<T> Interface. *Microsoft* [online]. [cit. 2020-01-26]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.iqueryable-1?view=netframework-4.8>
59. LINQ: Query Syntax vs. Method syntax. *C# Tutorial* [online]. [cit. 2020-01-26]. Dostupné z: <https://csharp.net-tutorials.com/linq/linq-query-syntax-vs-method-syntax/>
60. Razor syntax reference for ASP.NET Core. *Microsoft* [online]. [cit. 2020-01-28]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/core/mvc/views/razor?view=aspnetcore-3.1>
61. ASP.NET Razor – C# and VB Code Syntax. *W3Schools* [online]. [cit. 2020-01-28]. Dostupné z: [https://www.w3schools.com/asp/razor\\_syntax.asp](https://www.w3schools.com/asp/razor_syntax.asp)
62. JESS, Chadwick. *Programming Razor*. USA: O'Reilly Media, 2011. ISBN 9781449306762.
63. HTML Introduction. *W3Schools* [online]. [cit. 2020-01-28]. Dostupné z: [https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)
64. TSYMBALYUK, Aliona. How to Choose the Proper Web Hosting Service for Your Website: Types and Tips. *Yalantis* [online]. [cit. 2020-01-28]. Dostupné z: <https://yalantis.com/blog/types-of-hosting-solutions/>
65. WOOD, Kevin. How Much Does Dedicated Hosting Cost? *HostGator* [online]. December 21, 2018 [cit. 2020-02-01]. Dostupné z: <https://www.hostgator.com/blog/dedicated-hosting-cost/>
66. Publishing and running your ASP.NET Core project on Linux. *To .NET* [online]. July 16, 2017 [cit. 2020-02-01]. Dostupné z: <https://blog.todotnet.com/2017/07/publishing-and-running-your-asp-net-core-project-on-linux/>
67. Co je to PaaS?: Platforma jako služba. *Microsoft Azure* [online]. [cit. 2020-02-01]. Dostupné z: <https://azure.microsoft.com/cs-cz/overview/what-is-paas/>
68. AWS Elastic Beanstalk: Easy to begin, Impossible to outgrow. *AWS* [online]. [cit. 2020-02-01]. Dostupné z: <https://aws.amazon.com/elasticbeanstalk/>
69. Host a .NET Application: Build and deploy a .NET application using AWS Elastic Beanstalk. *AWS* [online]. [cit. 2020-02-01]. Dostupné z: <https://aws.amazon.com/getting-started/projects/host-net-web-app/>



70. App Service Documentation. *Microsoft Azure* [online]. [cit. 2020-02-01]. Dostupné z: <https://docs.microsoft.com/en-us/azure/app-service/>
71. What is Azure DevOps? Making Azure DevOps Work for You. *Bmc* [online]. [cit. 2020-02-01]. Dostupné z: <https://www.bmc.com/blogs/azure-devops/>
72. App Service pricing: Create web and mobile apps for any platform. *Microsoft Azure* [online]. [cit. 2020-02-01]. Dostupné z: <https://azure.microsoft.com/en-us/pricing/details/app-service/linux/>
73. How to Send Email Using SendGrid with Azure. *Microsoft Azure* [online]. [cit. 2020-02-01]. Dostupné z: <https://docs.microsoft.com/en-us/azure/sendgrid-dotnet-how-to-send-email>
74. The Heroku Platform. *Heroku* [online]. [cit. 2020-02-01]. Dostupné z: <https://www.heroku.com/platform>
75. Simple, flexible pricing: Plans for the needs of every app. *Heroku* [online]. [cit. 2020-02-01]. Dostupné z: <https://www.heroku.com/pricing>
76. Does Heroku support .NET applications? *Heroku* [online]. [cit. 2020-02-01]. Dostupné z: <https://help.heroku.com/PAT3YEDU/does-heroku-support-net-applications>
77. What is Docker? *Opensource* [online]. [cit. 2020-02-01]. Dostupné z: <https://opensource.com/resources/what-docker>
78. Website & web apps solutions: Deliver your web experience on Cloud Platform's fast, scalable VMs. *Google Cloud* [online]. [cit. 2020-02-01]. Dostupné z: <https://cloud.google.com/solutions/websites/>
79. 4 ways you can deploy an ASP.NET Core app to GCP. *Google Cloud* [online]. [cit. 2020-02-01]. Dostupné z: <https://cloud.google.com/blog/products/gcp/4-ways-you-can-deploy-an-aspnet-core-app-to-gcp>
80. App Engine Pricing. *Google Cloud* [online]. [cit. 2020-02-01]. Dostupné z: <https://cloud.google.com/appengine/pricing/>
81. Analytická specifikace a její zpracování. *Jihočeská univerzita v Českých Budějovicích* [online]. [cit. 2020-02-14]. Dostupné z: [http://ecom.ef.jcu.cz/web2/download/teorie/04\\_analyticka\\_specifikace\\_a\\_jeji\\_zpracovani.pdf](http://ecom.ef.jcu.cz/web2/download/teorie/04_analyticka_specifikace_a_jeji_zpracovani.pdf)
82. *Skautský oddíl Krnovská Trojka* [online]. [cit. 2020-02-15]. Dostupné z: <https://krnov.svazskautu.cz/>

83. *Jeleni - 175. skautský oddíl* [online]. [cit. 2020-02-15]. Dostupné z:  
<https://www.jeleni.org/>
84. *7. dívčí skautský oddíl* [online]. [cit. 2020-02-15]. Dostupné z:  
<http://sedmicka.aspone.cz/default.aspx>
85. Rozcestník Skautského vizuálního stylu. *Logo skaut* [online]. [cit. 2020-02-27].  
Dostupné z: <http://logo.skauting.cz/>

## **8 Přílohy**

Příloha A      CD se zdrojovým kódem aplikace

Příloha B      Kompletní UI specifikace

## **Příloha A – CD se zdrojovým kódem aplikace**

ModrinkyWebApp.zip

## Příloha B – Kompletní UI specifikace

### UI specifikace

#### Motivace a cíle

Tato UI specifikace vznikla jako výchozí bod pro implementaci webové aplikace pro dívčí skautský oddíl Modřinky. Jsou zde popsány jednotlivé funkcionality připravované aplikace a navržnuty wireframy, znázorňující, jak budou jednotlivé prvky stránek webové aplikace rozmístěny.

Hlavním cílem je implementovat webovou aplikaci sloužící skautskému oddílu ke zlepšení komunikace, zvýšení přehlednosti v přihlašování na akce a jako informační portál.

Vedlejšími cíli je udělat aplikaci co nejvíce uživatelsky přívětivou a jednoduchou, aby v ní bylo pohodlné se pohybovat i pro uživatele, kteří nejsou zvyklí pracovat často s informačními technologiemi.

#### Persony

##### Primární persona

**Jméno:** Ráchel Krejčová

**Pohlaví:** Žena

**Věk:** 14 let

**Koníčky:** Skauting, zpěv

**Historie:** Ráchel žije od narození v Táboře. Už od mala ráda trávila čas s přáteli v přírodě. V jedenácti letech ji rodiče přihlásili do místního skautského oddílu, kde je členkou dodnes. Chodí na základní školu a chystá se na gymnázium.

**Typický den:** Ráchel vstane v šest hodin ráno a společně se dvěma mladšími sourozenci vyrážejí do školy. Přestávky většinou tráví na mobilu. Po skončení vyučování buď dochází na skautské schůzky, do křesťanského sboru, nebo je s přáteli ve městě. Večer tráví u počítače procházením sociálních sítí a sledováním seriálů. O víkendech jezdí na skautské akce.

**Cíl:** Ráchel by chtěla mít přehled o svém skautském oddíle a plánovaných akcí stále u sebe a v jiné formě než pomocí nepřehledné e-mailové komunikace.

##### Sekundární persona

**Jméno:** Terezie Krchová

**Pohlaví:** Žena

**Věk:** 43 let

**Koníčky:** četba, zahradničení

**Historie:** Terezie pochází z malé vesnice poblíž Tábora. Má 4 sourozence a oba její rodiče jsou zahradníci. Vydala se stejnou cestou a vystudovala zemědělskou školu – obor zahradnictví. V tomto oboru pokračovala i na vysoké škole v Brně a následně se vrátila do rodné vesnice, kde postupně převzala rodinnou zahradnickou firmu. Je vdaná a má tři dcery, které umístila do skautského oddílu, protože sama také v mládí byla skautkou.

**Typický den:** Terezie vstává v 5 hodin ráno a připravuje dětem svačiny do školy. Poté se sama nasnídá a jde pracovat do svého zahradnictví. Odpoledne se ještě zastaví na místní faře, kde se věnuje dobrovolnické činnosti. Večer se společně s rodinou nají a před spaním kontroluje, zda mají děti vše potřebné do školy

**Cíl:** Terezii velmi záleží, aby její děti trávili čas smysluplně. Chce mít přehled, čemu se ve skautském oddíle věnují, zda chodí dostatečně do přírody a pokud ne, tak případně kontaktovat oddíl a požádat o nápravu. Samozřejmě si také ráda prohlédne fotky z akcí, aby viděla, jak si jí její ratolesti užily.

## Obecná pravidla

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
Úvod	Kontakty	Aktuality	Akce	Nirmala Mary	Fotky	Březina	Přihlášení

10 9

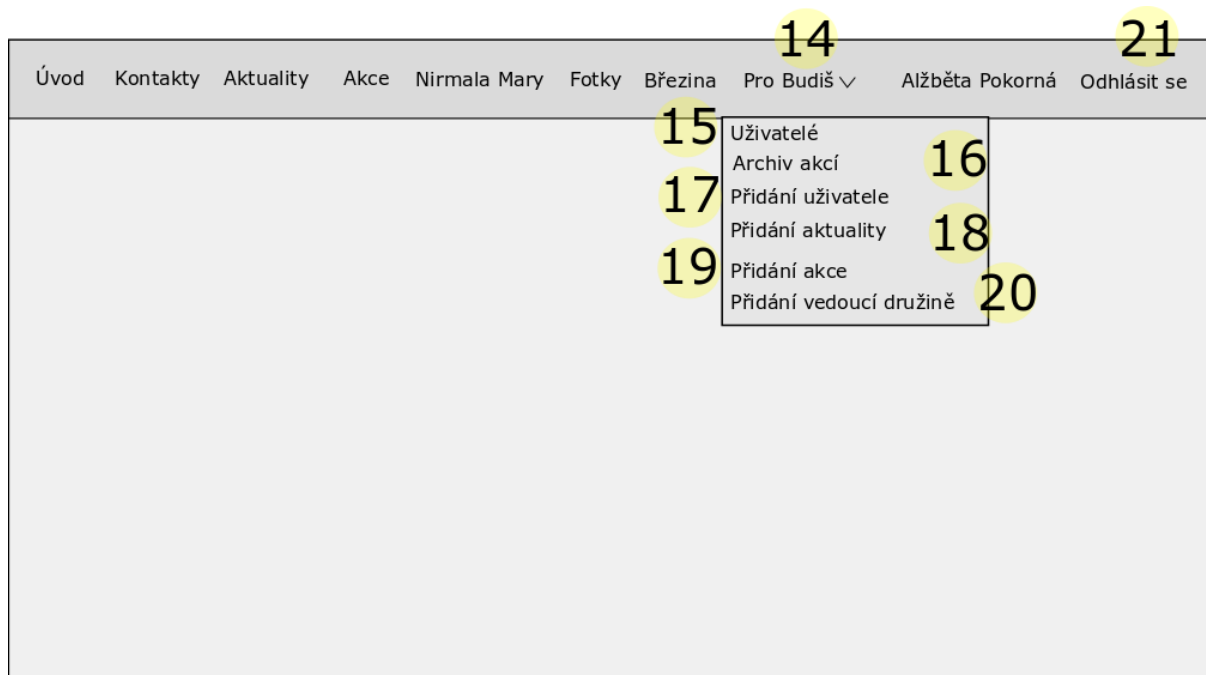
Úvod Kontakty Aktuality Akce Nirmala Mary Fotky Březina Pro členy ▾ Marie Dvořáková Odhlásit se

11 Moje družina

12

Úvod Kontakty Aktuality Akce Nirmala Mary Fotky Březina Pro rodiče ▾ Jana Nováková Odhlásit se

13 Jméno dítěte  
Jméno dítěte2



1. Úvod – záložka v menu vedoucí na základní informace o činnosti oddílu
2. Kontakty – záložka v menu vedoucí ke kontaktům na nejdůležitější vedoucí oddílu, adrese klubovny a střediska
3. Aktuality – záložka menu vedoucí na seznam jednotlivých aktualit skautského oddílu
4. Akce – záložka menu vedoucí na seznam chystaných akcí, na které se mohou přihlášení členové přihlásit, nebo na které rodiče mohou přihlásit své dítě
5. Nirmala Mary – záložka v menu vedoucí k informacím o projektu „Nirmala Mary“, neboli o dívce z Indie, kterou oddíl adoptoval na dálku
6. Foto – záložka menu sloužící jako odkaz na fotoalbum oddílu, umístěného na jiných webových stránkách
7. Březina – záložka v menu vedoucí k informacím o skautské základně „Březina“, kterou oddíl poskytuje k pronájmu
8. Přihlásit se – záložka v menu vedoucí ke přihlašovacímu formuláři
9. Jméno uživatele – záložka v menu, pojmenována podle celého jména aktuálně přihlášeného uživatele, vedoucí k zobrazení profilu uživatele a jeho editaci
10. Pro členy – „Drop down menu“ zobrazené, pokud je přihlášený uživatel v roli člen
11. Moje družina – záložka v menu vedoucí na informace o družině, ke které je člen přiřazen. Popřípadě vede na informaci, že člen žádnou družinu nemá.
12. Pro rodiče – „Drop down menu“ zobrazené, pokud je přihlášený uživatel v roli rodič a pokud má ve své správě alespoň jednoho člena.



13. „Jméno dítěte“ – záložka v menu nesoucí jméno konkrétního člena ve správě vedoucí na detailní informace o daném členovi.
14. Pro Budiš - „Drop down menu“ zobrazené, pokud je uživatel přihlášen a v roli „vedoucí“ nebo „admin“.
15. Uživatelé – záložka v menu vedoucí k seznamu všech uživatelů
16. Archiv akcí – záložka v menu vedoucí k seznamu proběhlých akcí a informacím o nich
17. Přidání uživatele – záložka v menu určená jako zkratka k přidání uživatele
18. Přidání aktuality – záložka v menu určená jako zkratka k přidání aktuality
19. Přidání akce – záložka v menu určená jako zkratka k přidání akce
20. Přidání vedoucí družině – záložka v menu zobrazená pouze je-li uživatel v roli admin. Odkazuje na formulář sloužící k přiřazování vedoucích k družinám.
21. Odhlásit se – záložka v menu pro odhlášení uživatele

## Úvod

### Use Case

Uživatel očekává:

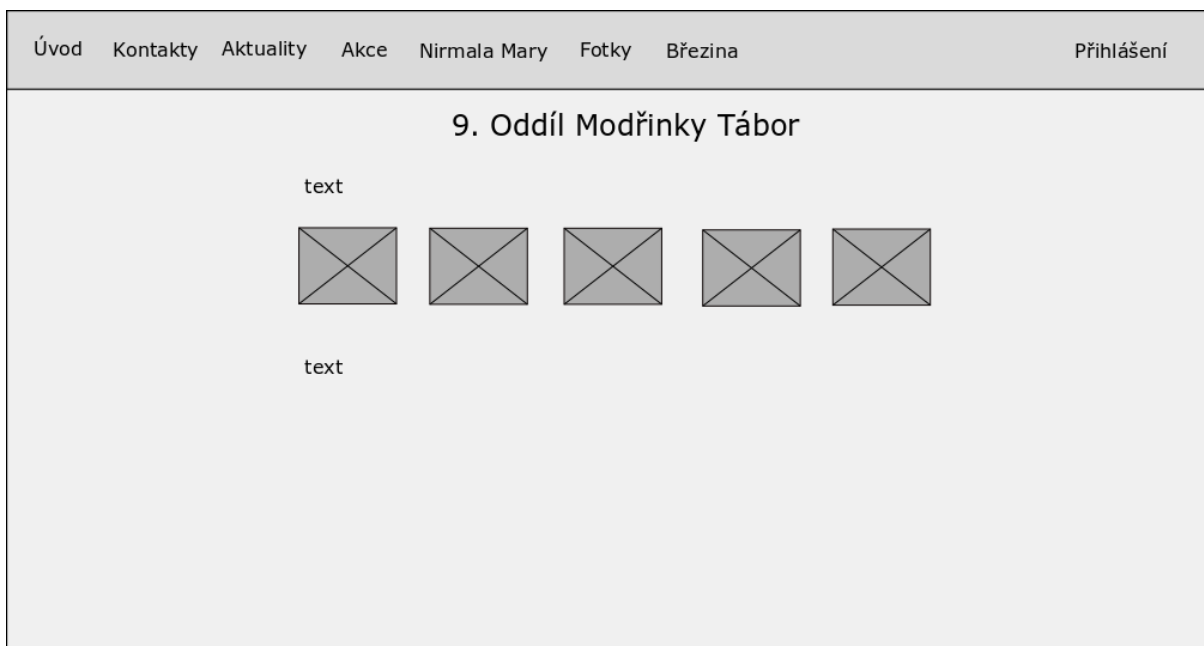
- Zobrazení úvodních informací o skautském oddíle a krátký popis o jeho činnosti.

### Scénář

Systém zobrazí:

- Základní informace o oddíle;
- Ilustrační obrázky k jednotlivým družinám.

### 8.1.1.1 Logický design



## Kontakty

### Use Case

Uživatel očekává:

- Nalezení adresy skautské klubovny;
- Zobrazení adresy skautské klubovny na mapě;
- Nalezení informace o vůdkyni oddílu a kontaktu na ni;
- Nalezení informace o zástupkyni vůdkyně oddílu a kontaktu na ni;
- Nalezení adresy sídliště střediska oddílu;
- Zobrazení adresy sídliště střediska oddílu na mapě;
- Nalezení kontaktu na správce stránek;
- Možnost získat více informací o oddílu z jiných zdrojů;
- Možnost získat více informací o středisku z jiných zdrojů.



### Scénář

System zobrazí:

- Adresu skautské klubovny;
- Rozklikávací mapu s přednastavenou adresou skautské klubovny;
- Kontakty na vedoucí oddílu;

- Kontakty na zástupkyni vedoucí oddílu;
- Adresu střediska;
- Rozklikávací mapu s přednastavenou adresou střediska;
- Kontakt na správce stránek;
- Odkaz na Facebook oddílu;
- Odkaz na stránky střediska.

## Logický design

Úvod	Kontakty	Aktuality	Akce	Nirmala Mary	Fotky	Březina	Přihlášení
<b>Kontakty</b>							
Klubovna	adresa						
Vůdkyně oddílu	jméno, email						
Zástupce vůdkyně oddílu	jméno, email						
Středisko	adresa						
Technické kontakty	jméno, email						
<a href="#">Facebook oddílu</a> <a href="#">Stránky střediska</a>							

## Nirmala Mary

### Use Case

Uživatel očekává:

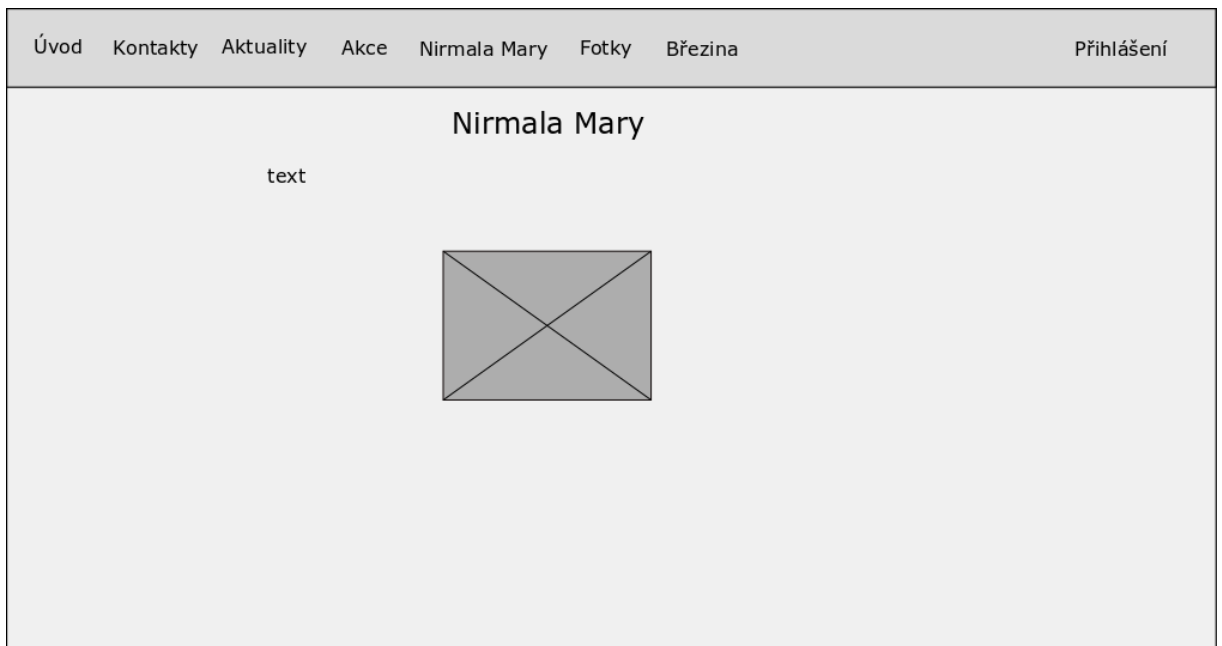
- Nalezení informací o projektu „Nirmala Mary“.

### Scénář

System zobrazí:

- Informace o projektu „Nirmala Mary“;
- Fotografie adoptované dívky.

## Logický design



### **Březina**

#### Use Case

Uživatel očekává:

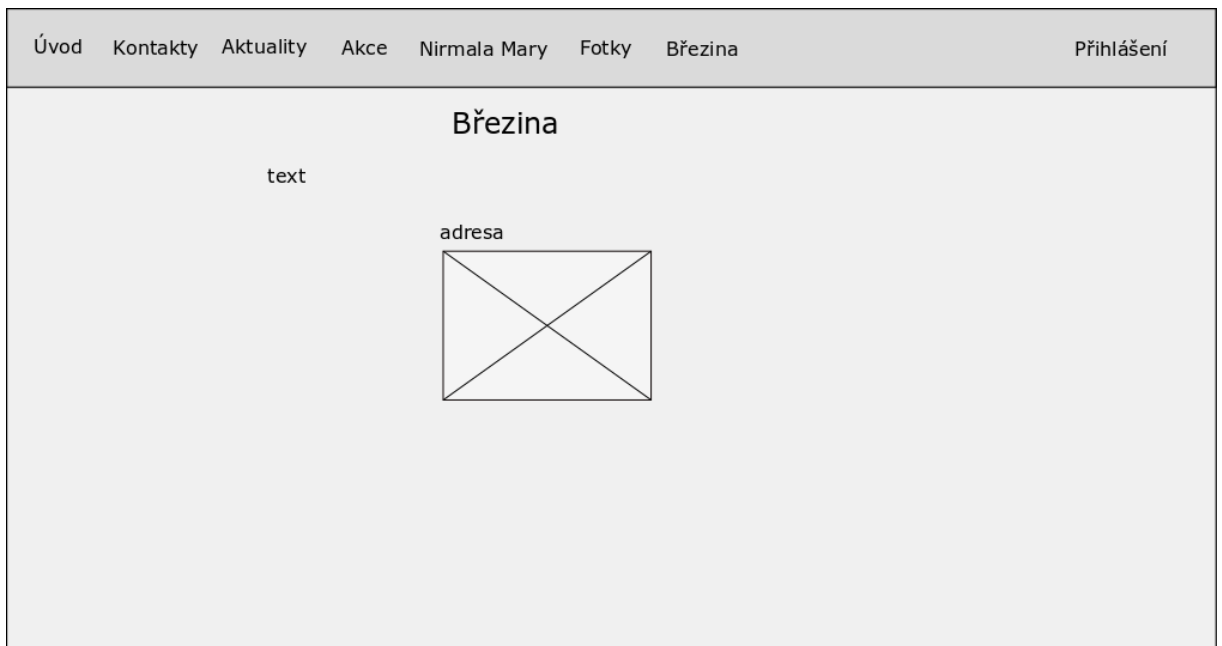
- Nalezení informací o skautské základně Březina;
- Označení adresy Březiny na mapě.

#### Scénář

System zobrazí:

- Informace o skautské základně Březina;
- Rozklikávací mapu s přednastavenou adresou základny.

## Logický design



## Přihlášení

### Use Case

Uživatel očekává:

- Možnost přihlásit se pomocí svých přihlašovacích údajů;
- Možnost obnovy zapomenutého hesla.

### Scénář

System zobrazí:

- Textové pole pro vyplnění e-mailu;
- Textové pole pro vyplnění hesla, znaky zde budou nahrazeny znaky „\*“;
- Tlačítko „Přihlásit se“ pro provedení přihlášení;
- V případě špatně zadaných údajů chyby, kvůli kterým nebylo možné přihlášení provést;
- Odkaz pro uživatele, kteří zapomněli své heslo.

Po kliknutí na:

- Tlačítko „Přihlásit se“ systém provede přihlášení uživatel, jsou-li jeho přihlašovací údaje vyplněné a správné;

- Odkaz „Zapomenuté heslo“ převede systém uživatele na formulář stránky „Zapomenuté heslo“

## Logický design

Úvod Kontakty Aktuality Akce Nirmala Mary Fotky Březina Přihlášení

### Přihlášení

Email

Heslo

[Zapomenuté heslo](#)

## Zapomenuté heslo

### Use Case

Uživatel očekává:

- Možnost zadat e-mail, který používá k přihlašování, k zaslání nového hesla;
- Možnost nechat si zaslat nové heslo.

### Scénář

Systém zobrazí:

- Textové pole pro zadání e-mailu;
- Tlačítko „Odeslat“.

Po kliknutí na:

- Tlačítko „Odeslat“
  - V případě nezadání e-mailu zobrazí validační chyby;
  - Z bezpečnostních důvodů v případě zadání e-mailu, který není využíván žádným registrovaným uživatelem, systém pouze zobrazí hlášku o odeslání e-mailu, e-mail však není odeslán;

- V případě správného vyplnění e-mailu systém zašle na zadaný e-mail odkaz pro resetování hesla a zobrazí hlášku o odeslaném e-mailu;
- Odkaz „Zapomenuté heslo“ je zobrazena stránka „Zapomenuté heslo“.

## Logický design

## Resetování hesla

### Use Case

Uživatel očekává:

- Možnost zadat svůj přihlašovací e-mail;
- Možnost nastavit si nové heslo k účtu.

### Scénář

Systém zobrazí:

- Textové pole pro zadání přihlašovacího e-mailu;
- Textové pole pro zadání nového hesla, znaky zde budou nahrazeny znaky „\*“;
- Textové pole pro potvrzení nově zadaného hesla, znaky zde budou nahrazeny znaky „\*“;
- Tlačítko „Resetovat“ pro nastavení nového hesla.

Po kliknutí na:

- Tlačítko „Resetovat“
  - Systém zobrazí validační hlášky, byly-li údaje vyplněné chybně

- V případě správně vyplněných údajů systém resetuje uživatelské heslo a zobrazí hlášku o resetování hesla s odkazem k přihlášení.

## Logický design

Úvod Kontakty Aktuality Akce Nirmala Mary Fotky Březina Přihlášení

### Resetování hesla

Email

Heslo

Potvrzení hesla

## Seznam aktualit

### Use Case

Uživatel očekává:


- Možnost přečíst si zprávy o aktuálním dění v oddíle;
- Možnost dostat se i k aktualitám, které se nevejdou na první stránku;
- Možnost přidat aktualitu, je-li v roli vedoucí či admin;
- Možnost editovat aktualitu, je-li jejím autorem;
- Možnost smazat aktualitu, je-li jejím autorem;
- Možnost editovat a smazat všechny aktuality, je-li uživatel v roli admin.

### Scénář



Systém zobrazí:

- Seznam jednotlivých aktuálních zpráv o oddíle, seřazených od nejpozději přidané, každá aktualita má nadpis, datum přidání a krátký text;
- Číselné vyjádření celkového počtu stránek, které aktuality zabírají;
- Je-li uživatel v roli vedoucí či admin, odkaz „Přidat aktualitu“ pro přidání nové aktuality

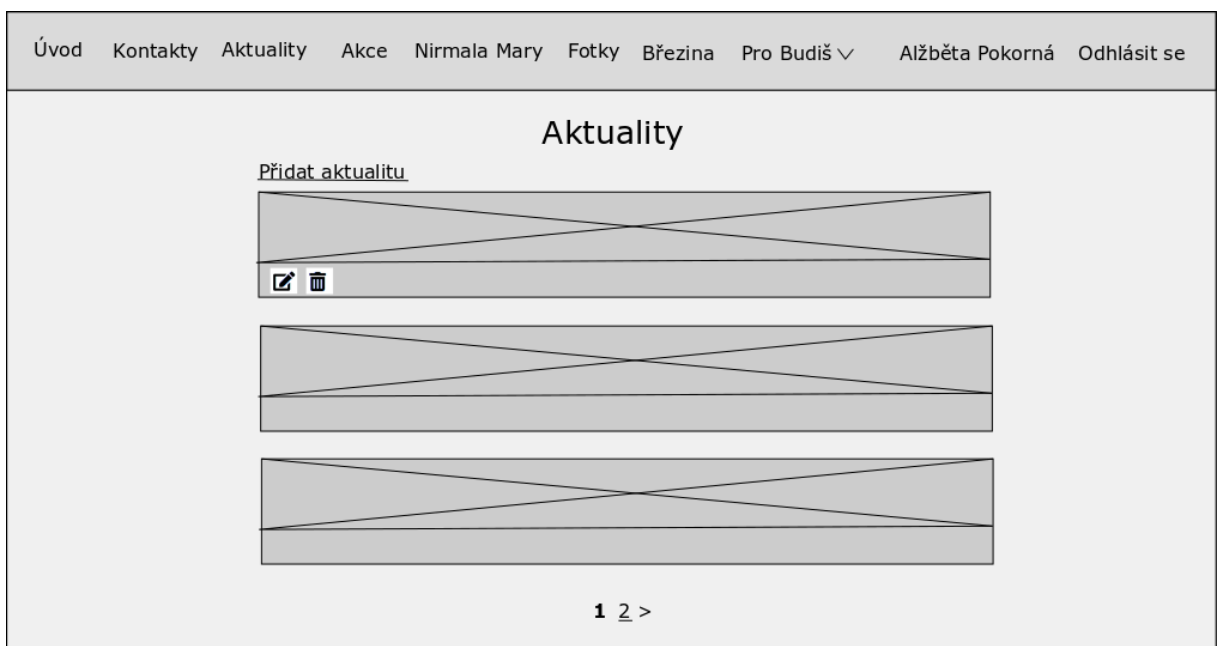


- Je-li uživatel v roli admin či je-li uživatel autorem aktuality, ikony  a  .

Po kliknutí na:

- Odkaz „Přidat aktualitu“ systém zobrazí stránku „Přidání aktuality“
- Ikonu  systém zobrazí stránku „Editace aktuality“;
- Ikonu  systém zobrazí stránku „Smazání aktuality“;
- Číslo zastupující jednotlivé stránky systém zobrazí stránku odpovídající stisknutému číslu.

## Logický design



## Přidání aktuality

### Use Case

Uživatel očekává:

- Možnost zadání názvu aktuality;
- Možnost zadání textu aktuality;
- Možnost novou aktualitu uložit;
- Možnost zrušit akci.

### Scénář

Systém zobrazí:

- Textové pole pro vyplnění názvu aktuality;

- Textové pole pro vyplnění textu aktuality;
- Tlačítko „Uložit“ pro uložení nové aktuality;
- Tlačítko „Zpět“ pro zrušení akce.

Po kliknutí na:

- Tlačítko „Uložit“
  - Pokud je vyplnění validní, systém aktualitu uloží a vrátí uživatele zpět na seznam aktualit;
  - Pokud vyplnění není validní, zobrazí chybové hlášky;
- Tlačítko „Zpět“ systém vrátí uživatele na seznam aktualit.

Logický design

## Editace aktuality

Use Case

Uživatel očekává:

- Možnost upravit již existující aktualitu;
- Možnost zrušit editaci bez uložení změn.

Scénář

Systém zobrazí:

- Formulář stejný jako při přidávání aktualit, pouze s předvyplněnými údaji zde dané aktuality;

- Tlačítko „Uložit“;
- Tlačítko „Zpět“.

Po kliknutí na:

- Tlačítko „Uložit“
  - Je-li formulář vyplněn chybně, systém zobrazí chybové hlášky;
  - Je-li formulář vyplněn správně, systém změny uloží a přesměruje uživatele na stránku „Seznam aktualit“;
- Tlačítko „Zpět“ systém přesměruje uživatele na stránku „Seznam aktualit“.

## **Smazání aktuality**

Use Case

Uživatel očekává:

- Možnost smazat aktualitu.

Scénář

Systém zobrazí:

- Hlášku o potvrzení smazání aktuality a následně danou aktualitu smaže.

## **Seznam akcí**

Use Case



Uživatel očekává:

- Možnost zobrazit si chystané akce, je-li přihlášený uživatel;
- Možnost vyfiltrvat si jednotlivé akce podle zadaných parametrů;
- Možnost dozvědět se o akci více informací;
- Možnost přidat novou akci, je-li uživatel v roli admin či vedoucí;
- Možnost editovat či smazat akci, je-li uživatel jejím autorem nebo je-li uživatel v roli admin.



Scénář

Systém zobrazí:

- Seznam akcí, seřazené podle nejdříve konané, každá akce má název, počáteční den konání, název, štítek dle typu akce a krátký popis;
- Textové pole pro zadání názvu akce k vyfiltrování;

- Pole výběru pro zadání měsíce konání akce k vyfiltrování;
- Pole výběru pro zadání roku konání akce k vyfiltrování;
- Tlačítko „Hledat“ pro zahájení filtrování;
- Tlačítko „Reset“ pro resetování vyhledávacích formulářů;
- Je-li uživatel v roli vedoucí či admin, odkaz „Přidat akci“ pro přidání nové akce;
- Je-li uživatel v roli vedoucí či admin, odkaz „Kdo jede?“ pro zobrazení účastníků akce;
- Je-li uživatel v roli admin či je-li uživatel autorem aktuality, ikony  a .

Po kliknutí na:

- Tlačítko „Hledat“ systém zobrazí pouze ty akce, které odpovídají zadaným parametrům;
- Tlačítko „Reset“ systém vymaže zadané vstupy v hledacích formulářích;
- Odkaz „Přidat akci“ systém zobrazí stránku „Přidání akce“;
- Ikonu  systém zobrazí stránku „Editace akce“;
- Ikonu  systém zobrazí stránku „Smazání akce“;
- Odkaz „Kdo jede?“ systém zobrazí seznam přihlášených účastníků;
- Název akce systém zobrazí detail akce.

## Logický design

Úvod Kontakty Aktuality Akce Nirmala Mary Fotky Březina Pro Budiš ▾ Alžběta Pokorná Odhlásit se

### Akce

[Přidat akci](#)

1. leden 2000 Oddílová akce

Název akce	Popis
<div style="display: flex; align-items: center; gap: 10px;"> <span></span> <span></span> <span><a href="#">Kdo jede?</a></span> </div>	

## Detail akce



### Use Case

Uživatel očekává:



- Možnost přečíst si bližší informace o akci;
- Možnost akci editovat či smazat, je-li autorem akce nebo je-li v roli admin;
- Možnost zobrazit si účastníky akce, je-li v roli vedoucí, nebo admin;
- Možnost přihlásit se na akci, je-li v roli admin, vedoucí, nebo člen;
- Možnost přihlásit členy ve své správě, je-li v roli rodič;
- Možnost odhlásit se z akce;
- Možnost vrátit se zpět na seznam akcí.

### Scénář

System zobrazí:

- Detailní popis akce;
- Seznam družin, pro které je akce určená;
- Je-li uživatel v roli vedoucí či admin, odkaz „Kdo jede?“ pro zobrazení účastníků akce;
- Je-li uživatel v roli admin či je-li uživatel autorem aktuality, ikony  a  ;
- Tlačítko „Přihlásit se na akci“ je-li uživatel v roli admin, vedoucí, nebo člen;
- Tlačítko „Odhlásit se z akce“ je-li uživatel v roli admin, vedoucí, nebo člen a je-li na danou akci již přihlášen;
- Tlačítko „Přihlásit člena na akci“ je-li uživatel v roli rodič;
- Tlačítko „Zpět na seznam akcí“.

Po kliknutí na:

- Ikonu  systém zobrazí stránku „Editace akce“;
- Ikonu  systém zobrazí stránku „Smazání akce“;
- Odkaz „Kdo jede?“ systém zobrazí seznam přihlášených účastníků;
- Tlačítko „Zpět na seznam akcí“ systém zobrazí seznam akcí;
- Tlačítko „Přihlásit se na akci“
  - Je-li uživatel v roli člen a nemá přiřazenou družinu, zobrazí se informace o tom, že se přihlásit nemůže;

- Je-li uživatel v roli člen a jeho družina není v seznamu družin povolených pro akci, zobrazí se informace o tom, že se přihlásit nemůže;
- Je-li uživatel v roli člen a se správnou družinou, pro kterou je akce určena, zobrazí se stránka „Přihlášení na akci“;
- Je-li uživatel v roli vedoucí, nebo admin, zobrazí se potvrzovací formulář pro přihlášení a uživatel je následně na akci přihlášen;
- Tlačítko „Odhlásit se z akce“ systém zobrazí potvrzovací formulář a následně je uživatel z akce odhlášen;
- Tlačítko „Přihlásit člena na akci“ systém zobrazí stránku „Přihlášení člena na akci“.

## Logický design

Úvod Kontakty Aktuality Akce Nirmala Mary Fotky Březina Pro Budiš ▾ Alžběta Pokorná Odhlásit se

### Název akce

Začátek akce: 1. leden 2000

  [Kdo jede?](#)

Bližší popis akce:

Pro koho je akce určená: Fialky  
Modrásci

## Přihlášení na akci

### Use Case

Uživatel očekává:

- Možnost potvrdit přihlášení na akci.

### Scénář

Systém zobrazí:

- Hlášku o potvrzení přihlášení a následně uživatele na akci přihlásí.

## Přihlášení člena na akci

### Use Case

Uživatel očekává:

- Možnost přihlásit členy ve své správě na akci;
- Možnost odhlásit členy ve své správě z akce;
- Možnost vrátit se zpět bez uložení změn.

### Scénář

Systém zobrazí:

- Seznam členů ve správě uživatele, pro které je akce určena;
- Checkboxy pro označení uživatelů k přihlášení, které jsou primárně označené;
- Informaci o již přihlášených členech;
- Odkaz „Odhlásit člena z akce“ u členů, kteří jsou již přihlášení;
- Tlačítko „Přihlásit vybrané členy“;
- Tlačítko „Zpět“.

Po kliknutí na:

- Checkbox u jednotlivých členů se změní, zda bude daný člen přihlášen po stisknutí tlačítka „Přihlásit vybrané členy“;
- Odkaz „Odhlásit vybrané členy“ systém zobrazí formulář pro potvrzení odhlášení a daný člen je následně odhlášen;
- Tlačítko „Zpět“ je uživatel přesměrován na detail akce;
- Tlačítko „Přihlásit vybrané členy“ jsou vybraní členové přihlášení na danou akci.

Úvod   Kontakty   Aktuality   Akce   Nirmala Mary   Fotky   Březina   Pro rodiče ▾   Jana Nováková   Odhlásit se

### Přihlášení na akci: Akce1

Jméno Dítěte je již přihlášen [Odhlásit člena z akce](#)

Jméno Dítěte2

## Přidání akce

### Use Case

Uživatel očekává:

- Možnost vyplnění potřebných textových údajů k nové akci;
- Možnost určit, pro které družiny je akce určena;
- Možnost určit, jakého typu akce je;
- Možnost uložit novou akci;
- Možnost odejít z formuláře bez uložení změn.

### Scénář

Systém zobrazí:

- Textové pole pro vyplnění názvu nové akce;
- Textové pole pro vyplnění úvodního popisu akce;
- Textové pole pro vyplnění detailního popisu akce;
- Pole výběru pro zadání dne konání akce;
- Pole výběru pro zadání měsíce konání akce;
- Pole výběru pro zadání roku konání akce;
- Checkboxy pro zaškrtnutí povolených družin;



- Tlačítka typu radiobutton pro zaškrtnutí typu akce;
- Tlačítko „Uložit“;
- Tlačítko „Zpět“.

Po kliknutí na:

- Tlačítka „Uložit“
  - Jsou-li data zadána správně, je akce uložena a uživatel přesměrován na stránku „Seznam akcí“;
  - Jsou-li data uložena chybně, systém zobrazí chybové hlášky;
  - Je-li zadáno dnešní, či již proběhlé datum, akce není uložena a objeví se varovná hláška;
- Tlačítka „Zpět“ je uživatel přesměrován na stránku „Seznam akcí“.

## Logický design

Úvod
Kontakty
Aktuality
Akce
Nirmala Mary
Fotky
Březina
Pro Budiš ▾
Alžběta Pokorná
Odhlásit se

### Přidání akce

Název nové akce

Úvodní popis nové akce

Detailní popis nové akce

Povolené družiny

Modrásci

Fialky

Delfíni

Chobotnice

Měsíc konání akce

Den konání akce

Rok konání akce

Oddílová akce

Středisková akce

Celorepubliková akce

## Editace akce

### Use Case

Uživatel očekává:

- Možnost editace již existující akce;
- Možnost přerušit editaci bez uložení změn.

### Scénář

Systém zobrazí:

- Stejný formulář jako při přidávání akce, který je předvyplněn údaji editované akce;
- Tlačítko „Uložit“;
- Tlačítko „Zpět“;

Po kliknutí na:

- Tlačítko „Uložit“
  - Jsou-li údaje zadané chybně, systém zobrazí chybové hlášky;
  - Jsou-li údaje zadané správně, systém změny uloží, pokud zároveň dojde k odebrání dříve povolených družin, odhlásí členy s odebranou družinou z akce a oznámí jim tuto skutečnost e-mailem. Následně uživatele přesměruje na seznam akcí;
- Tlačítko „Zpět“ systém uživatele přesměruje na seznam akcí.

## Logický design

Úvod Kontakty Aktuality Akce Nirmala Mary Fotky Březina Pro Budiš ▾ Alžběta Pokorná Odhlásit se

### Editace akce

Název nové akce

Úvodní popis nové akce

Detailní popis nové akce

Povolené družiny

- Modrásci
- Fialky
- Delfíni
- Chobotnice

Měsíc konání akce

Den konání akce

Rok konání akce

- Oddílová akce
- Středisková akce
- Celorepubliková akce

### Smazání akce

#### Use Case

Uživatel očekává:

- Možnost smazat existující akci.

Scénář:

System zobrazí:

- Hlášku o potvrzení smazání akce a následně akci smaže.

### Seznam uživatelů

#### Use Case

Uživatel očekává:

- Možnost zobrazit si seznam uživatelů, je-li v roli admin nebo vedoucí;
- Možnost přidat nového uživatele;

- Možnost zobrazení detailních informací o jednotlivých uživateli;
- Možnost vyfiltrovat si seznam uživatelů s omezujícími parametry;

## Scénář

System zobrazí:

- Seznam uživatelů seřazený podle abecedy;
- Textové pole pro zadání křestního jména;
- Textové pole pro zadání příjmení;
- Tlačítko „Hledat“;
- Tlačítko „Reset“
- Odkaz „Přidat uživatele“.

Po kliknutí na:

- Tlačítko „Hledat“ je zobrazen seznam uživatelů dle zadaných parametrů;
- Tlačítko „Reset“ je smazán obsah vyhledávacího formuláře;
- Jméno uživatele jsou zobrazeny detailní informace o tomto uživateli;
- Odkaz „Přidat uživatele“ je uživatel přesměrován na stránku „Přidat uživatele“.

## Logický design

Úvod	Kontakty	Aktuality	Akce	Nirmala Mary	Fotky	Březina	Pro Budiš ▾	Alžběta Pokorná	Odhlásit se
------	----------	-----------	------	--------------	-------	---------	-------------	-----------------	-------------

### Uživatelé

[Přidat uživatele](#)

Dvořáková Marie  
 Nováková Jana  
 Pokorná Alžběta

## Přidání uživatele

### Use Case

Uživatel očekává:

- Možnost vyplnit potřebné informace o novém uživateli;
- Možnost uložit nového uživatele;

### Scénář

Systém zobrazí:

- Textové pole pro zadání křestního jména uživatele
- Textové pole pro zadání příjmení uživatele;
- Textové pole pro zadání e-mailu uživatele;
- Pole výběru pro zvolení role uživatele, je-li uživatel v roli vedoucí, na výběr jsou pouze role „rodič“ a „člen“;
- Tlačítko „Přidat uživatele“.

Po kliknutí na:

- Tlačítko „Přidat uživatele“
  - Je-li formulář vyplněný chybně, objeví se chybové výpisy;
  - Je-li formulář vyplněn správně a byla-li role nového uživatele určena jako člen, systém uživatele uloží, odešle mu e-mail s vygenerovaným heslem a přesměruje uživatele na stránku „Přidání družiny uživatelovi“;
  - Je-li formulář vyplněn správně a byla-li role nového uživatele určena jiná než jako člen, systém uživatele uloží a odešle mu e-mail s vygenerovaným heslem.

## Logický design

Úvod Kontakty Aktuality Akce Nirmala Mary Fotky Březina Pro Budiš ▾ Alžběta Pokorná Odhlásit se

### Přidat uživatele

Email

Jméno

Příjmení

Role

## Editace uživatele

### Use Case

Uživatel očekává:

- Možnost editace již existujícího uživatele;
- Možnost zrušit editaci bez uložení změn.

### Scénář

Systém zobrazí:

- Formulář stejný jako při přidávání uživatele, ale bez možnosti změny e-mailu a s předvyplněnými údaji dle editovaného uživatele;
- Tlačítko „Uložit“;
- Tlačítko „Zpět“.

Po kliknutí na:

- Tlačítko „Uložit“
  - Jsou-li údaje vyplněné chybně, systém zobrazí chybové hlášky;
  - Jsou-li údaje vyplněné správně, systém uloží změny a přesměruje uživatele na seznam uživatelů;
- Tlačítko „Zpět“ systém přesměruje uživatele na seznam uživatelů.

## **Smazání uživatele**

### Use Case

Uživatel očekává:

- Možnost smazat již existujícího uživatele.

### Scénář

Systém zobrazí:

- Hlášku o potvrzení smazání uživatele a následně uživatele smaže.

## **Přidání družiny uživateli**

### Use Case

Uživatel očekává:

- Možnost přiřadit uživateli novou družinu;
- Možnost odejít z formuláře bez uložení změn.

### Scénář

Systém zobrazí:

- Pole výběru pro přiřazení nové družiny, které může zůstat i nevyplněné;
- Tlačítko „Uložit“;
- Tlačítko „Zpět“.

Po kliknutí na:

- Tlačítko „Uložit“ je nová družina přidána, při nevyplnění pole naopak odebrána;
- Tlačítko „Zpět“ systém přesměruje uživatele na detail uživatele, kterému byla družina přiřazována.

## Logický design

The screenshot shows a web application interface. At the top, there is a navigation bar with the following items: Úvod, Kontakty, Aktuality, Akce, Nirmala Mary, Fotky, Březina, Pro Budiš (with a dropdown arrow), Alžběta Pokorná, and Odhlásit se. Below the navigation bar, the main content area features a form with a label 'Družina' above a dropdown menu. Below the dropdown menu are two buttons: 'Uložit' and 'Zpět'.

### Detail uživatele

#### Use Case

Uživatel očekává:



- Je-li v roli admin nebo vedoucí, možnost zobrazit si detailní informace o daném uživateli;
- Je-li v roli admin, uživatele smazat či editovat;
- Je-li sledovaný uživatel v roli člen, možnost změnit mu družinu;
- Je-li sledovaný uživatel v roli rodič, možnost přidat mu nové členy do správy a prohlédnout si detailní informace o členech, které již ve správě má;
- Možnost vrátit se zpět na seznam uživatelů.

#### Scénář



System zobrazí:

- E-mail uživatele a roli uživatele;
- Je-li sledovaný uživatel v roli člen, družinu tohoto uživatele;
- Je-li sledovaný uživatel v roli rodič, seznam členu, které má daný uživatel ve své správě;
- Je-li sledovaný uživatel v roli člen, odkaz „Změnit družinu uživatele“;
- Je-li sledovaný uživatel v roli rodič, odkaz „Přidat členu“;

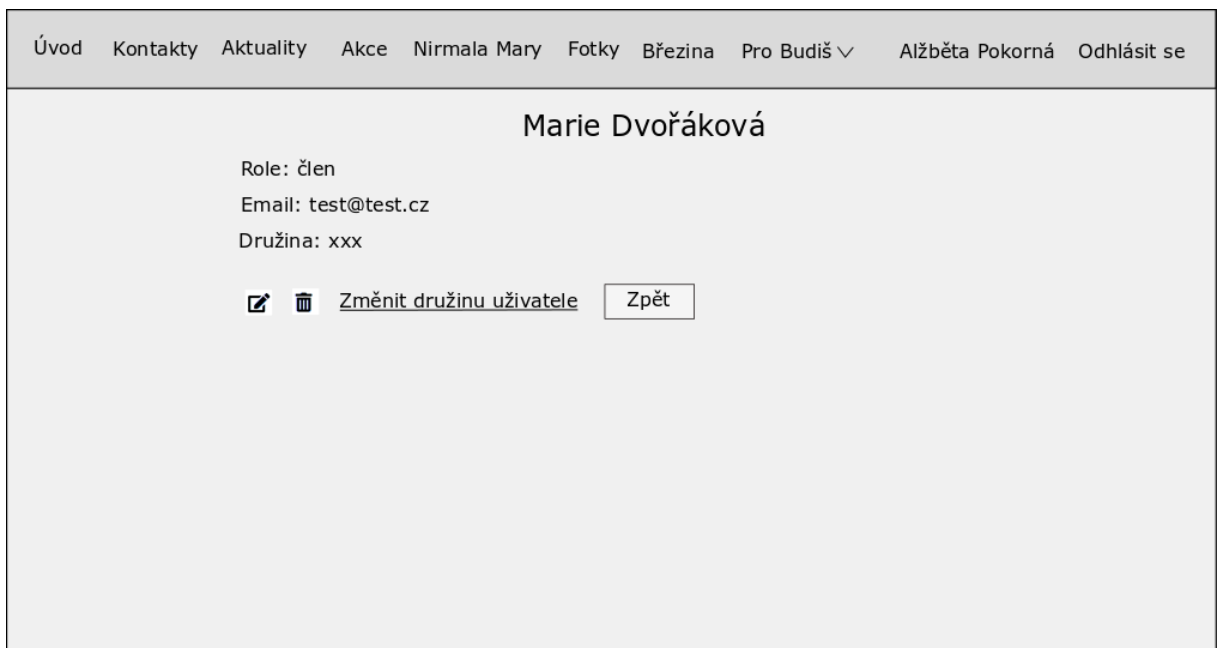


- Je-li přihlášený uživatel v roli admin, ikony  a  ;
- Tlačítko „Zpět“

Po kliknutí na:

- Ikonu  systém zobrazí stránku „Editace uživatele“;
- Ikonu  systém zobrazí stránku „Smazání uživatele“;
- Odkaz „Změnit družinu uživatele“ je uživatel přesměrován na stránku „Přidání družiny uživateli“;
- Odkaz „Přidat člena“ je uživatel přesměrován na stránku „Přidat člena“;
- Jméno člena ve správě, jsou zobrazeny detailní informace o daném členovi;
- Tlačítko „Zpět“ je uživatel přesměrován na seznam uživatelů.

## Logický design



## Přidat člena

### Use Case

Uživatel očekává:

- Možnost přidat člena do správy uživateli;
- Možnost zrušit přidávání člena bez uložení změn.

### Scénář

Systém zobrazí:

- Pole na vyplnění jména a družiny nového člena;
- Tlačítko „Uložit“;
- Tlačítko „Zpět“.

Po kliknutí na:

- Tlačítko „Uložit“
  - Jsou-li data zadána chybně, systém zobrazí chybové hlášky;
  - Jsou-li data zadaná správně, systém nového člena uloží a uživatele přesměruje na detail uživatele, kterému byl člen přidáván;
- Tlačítko „Zpět“, systém uživatele přesměruje na detail uživatele, kterému byl člen přidáván.

## Detail člena



### Use Case

Uživatel očekává:



- Možnost zobrazit si detailní informace o členovi, kterého má ve správě některý z uživatelů v roli rodič, je-li přihlášený uživatel v roli admin nebo vedoucí;
- Možnost editovat či smazat takového člena, je-li přihlášený uživatel v roli admin nebo vedoucí;
- Možnost vrátit se zpět na detailní informace o uživatelovi, který má tohoto člena ve správě.

### Scénář

Systém zobrazí:

- Detailní informace o daném členovi;
- Je-li přihlášený uživatel v roli admin nebo vedoucí, ikony  a  ;
- Tlačítko „Zpět“

Po kliknutí na:

- Ikonu  systém zobrazí stránku „Editace člena“;
- Ikonu  systém zobrazí stránku „Smazání člena“;
- Tlačítko „Zpět“ systém zobrazí stránku „Detail uživatele“ pro uživatele, který je správcem daného člena.

## Logický design

The screenshot shows a web application interface. At the top, there is a navigation bar with the following items: Úvod, Kontakty, Aktuality, Akce, Nirmala Mary, Fotky, Březina, Pro Budiš (with a dropdown arrow), Alžběta Pokorná, and Odhlásit se. Below the navigation bar, the main content area has a title "Jméno Dítěte" centered at the top. Underneath the title, it says "Družina: xxx". Below that, there are two small icons: a pencil and a trash can. To the right of these icons is a button labeled "Zpět".

### Editace člena

#### Use Case

Uživatel očekává:

- Možnost editace již existujícího člena;
- Možnost přerušit editaci bez uložení změn.

#### Scénář

System zobrazí:

- Formulář stejný jako při přidávání člena, ale s předvyplněnými údaji dle editovaného člena;
- Tlačítko „Uložit“;
- Tlačítko „Zpět“.

Po kliknutí na:

- Tlačítko „Uložit“
  - Jsou-li údaje vyplněny chybně, systém zobrazí chybové hlášky;
  - Jsou-li údaje vyplněny správně, systém změny uloží a přesměruje uživatele na detail uživatele, jehož člen byl editován;
- Tlačítko „Zpět“ systém přesměruje uživatele na detail uživatele, jehož člen byl editován.

## **Smazání člena**

### Use Case

Uživatel očekává:

- Možnost smazat již existujícího člena;

### Scénář

System zobrazí:

- Hlášku s potvrzením smazání člena a následně člena smaže.

## **Archiv akcí**

### Use Case

Uživatel očekává:

- Možnost zobrazit si seznam již proběhlých akcí, je-li v roli admin nebo vedoucí;
- Možnost zobrazit si detailní informace o těchto akcích.

### Scénář

System zobrazí:

- Seznam proběhlých akcí, identifikovaných podle roku konání a názvu, seřazených podle nejpozději proběhlé;

Po kliknutí na:

- Odkaz s názvem akce systém zobrazí seznam, kdo se této akce zúčastnil a detailní popis akce.



## Profil uživatele

### Use Case

Uživatel očekává:

- Možnost zobrazit si informace o svém uživatelském účtu;
- Možnost změnit si heslo;
- Možnost zobrazit si akce, na které je přihlášen, není-li v roli rodič;
- Možnost zobrazit si detailní informace o členovi, které má ve správě, je-li v roli rodič
- Možnost zobrazit si informace o své družině, je-li v roli člen.

### Scénář

System zobrazí:

- Informace o uživatelském profilu;
- Odkaz „Změnit heslo“;
- Seznam akcí, na které je uživatel přihlášen;
- Seznam členů, které má uživatel ve své správě;
- Odkaz s názvem družiny;

Po kliknutí na:

- Odkaz „Změnit heslo“ systém zobrazí stránku „Změna hesla“;

- Odkaz s názvem družiny systém zobrazí stránku „Moje družina“;
- Odkaz s názvem člena systém zobrazí detailní informace o členovi včetně odkazu na jeho družinu;
- Odkaz s názvem akce systém zobrazí detailní popis této akce.

## Logický design

<p>Úvod   Kontakty   Aktuality   Akce   Nirmala Mary   Fotky   Březina   Pro členy ▾   Marie Dvořáková   Odhlásit se</p>
<p><b>Marie Dvořáková</b></p>
<p>Role: člen</p> <p>Email: test@test.cz</p> <p>Družina: <a href="#">Modrásci</a></p> <p>Seznam akcí, na které jsem přihlášena:</p> <p style="text-align: right;"><a href="#">Akce1</a></p> <p><a href="#">Změna hesla</a></p>

## Změna hesla

### Use Case

Uživatel očekává:

- Možnost vyplnění nového hesla;
- Možnost zrušit akci bez uložení změn.

### Scénář

Systém zobrazí:

- Textové pole pro vložení původního hesla;
- Textové pole pro vložení nového hesla;
- Textové pole pro potvrzení nového hesla;
- Tlačítko „Změnit heslo“;
- Tlačítko „Zpět“.

Po kliknutí na:

- Tlačítko „Změnit heslo“
  - Je-li formulář vyplněn chybně, systém zobrazí chybové hlášky;
  - Je-li formulář vyplněn správně, systém změní heslo;
- Tlačítka „Zpět“ systém zobrazí stránku „Profil uživatele“.

Logický design

Úvod Kontakty Aktuality Akce Nirmala Mary Fotky Březina Pro rodiče ▾ Jana Nováková Odhlásit se

### Změna hesla

Současné heslo

Nové heslo

Potvrzení nového hesla

Změnit heslo Zpět

## Moje družina

Use Case

Uživatel očekává:

- Možnost zobrazit si informace o družině, ve které je členem, nebo ve které je člen v jeho správě;
- Možnost kontaktovat vedoucího své družiny;
- Možnost zobrazit si akce určené pro jeho družinu.

Scénář

Systém zobrazí:

- Detailní informace o dané družině;
- Textové pole pro vložení správy pro vedoucí družiny, má-li družina přiřazenou vedoucí;
- Seznam akcí, kterých se členové dané družiny mohou zúčastnit;

- Tlačítko „Odeslat“.

Po kliknutí na:

- Tlačítko „Odeslat“ systém odešle e-mail vedoucímu družiny s textem vyplněným v textovém poli;
- Odkaz s názvem akce systém zobrazí detailní informace o dané akci.

## Logický design

The screenshot shows a web application interface with a navigation bar at the top containing links: Úvod, Kontakty, Aktuality, Akce, Nirmala Mary, Fotky, Březina, Pro členy (with a dropdown arrow), Marie Dvořáková, and Odhlásit se. The main content area is titled "Družina". Under the heading "Popis", there is a placeholder image represented by a rectangle with a diagonal cross. Below this, the text reads "Vedoucí družiny: xx". The "Seznam členů:" section lists "Marie Dvořáková". The "Seznam akcí pro družinu:" section contains a link labeled "Akce1". At the bottom, there is a text input field labeled "Napsat vedoucímu družiny:" and a button labeled "Odeslat".

## Přiřazení vedoucí družině

### Use Case

Uživatel očekává:

- Možnost přiřazení nebo odebrání vedoucí jednotlivým družinám, je-li v roli admin;
- Možnost uložit tyto změny.

### Scénář

Systém zobrazí:

- Pole výběru pro vybraní vedoucí jednotlivým družinám s možností nechat pole prázdné;
- Tlačítko „Uložit“.

Po kliknutí na:



- Tlačítko „Uložit“ jsou dané změny uloženy.

## Logický design

The screenshot shows a web application interface with a navigation bar at the top containing the following items: Úvod, Kontakty, Aktuality, Akce, Nirmala Mary, Fotky, Březina, Pro Budiš (with a dropdown arrow), Alžběta Pokorná, and Odhlásit se. The main content area is titled "Přidání vedoucí družině" and contains four dropdown menus labeled "Modrásci", "Fialky", "Delfíni", and "Chobotnice". Each dropdown menu is currently empty and has a small downward-pointing arrow on the right side. Below the dropdown menus is a button labeled "Uložit".

## Odhlášení

### Use Case

Uživatel očekává:

- Možnost odhlásit se, je-li přihlášen.

### Scénář

Po kliknutí na:

- Záložku menu „Odhlásit se“ systém zobrazí úvodní stranu a uživatele odhlásí.