

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

Návrh webové stránky pro členy včelařského spolku

Antonín Jelínek

© 2024 ČZU v Praze

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Antonín Jelínek

Informatika

Název práce

Návrh webové stránky pro členy včelařského spolku

Název anglicky

Design and Implementation of Web Application for Local Beekeepers Association

Cíle práce

Cílem práce je návrh webové aplikace typu client-server pro členy včelařského spolku. Aplikace bude postavena na frameworku ASP.NET a napsána v programovacím jazyce C#. Aplikace bude rozdělena na veřejnou a neveřejnou část. Veřejná část bude dostupná všem uživatelům z internetové sítě a bude obsahovat informace o historii spolku, jeho činnosti a o poskytovaných službách veřejnosti. Neveřejná část bude přístupna po přihlášení uživatelů přes session a jejich ověření. V neveřejné části budou pro členy spolku informace o zprávách z zasedání spolku či informace pro členy. Dále budou moci členové vést vlastní evidenci o včelstev, která bude obsahovat informace o počtu včelstev, jejich umístění v rámci úlu, hmotnosti, množství medu pole jednotlivých včelstev, či stav úlové váhy.

Všechny informace by se ukládaly do znormalizované MySQL databáze, která by byla propojena s aplikací pomocí Entity Frameworku.

Metodika

1. Na základě studia odborných zdrojů popište problémovou doménu formou literární rešerše
2. Provedte analýzu uživatelských požadavků na webovou aplikaci a popište je vhodnými nástroji (Use Case)
3. Provedte analýzu a zhodnocení existujících webových rezervačních systémů
4. V souladu s doporučenými postupy softwarového inženýrství vytvořte návrh vlastní webové aplikace (konceptuální model)
5. Návrh aplikace implementujte ve vhodném prostředí a řádně otestujte

Doporučený rozsah práce

30 – 40 stran

Klíčová slova

včelařský spolek, C#, ASP.NET, Entity Framework, MySQL, webová aplikace

Doporučené zdroje informací

ARLOW, J., NEUSTADT, I. UML 2 a unifikovaný proces vývoje aplikací : objektově orientovaná analýza a návrh prakticky. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.

CASTRO, Elizabeth a Bruce HYSLOP. HTML5 a CSS3: názorný průvodce tvorbou WWW stránek. 1. vyd. Brno: Computer Press, 2012. ISBN 9788025137338.

HOGAN, Brian P. HTML5 a CSS3: výukový kurz webového vývojáře. Brno: Computer Press, 2011. ISBN 978-80-251-3576-1.

VIRIUS, Miroslav. Programování v C#: od základů k profesionálnímu použití. Praha: Grada Publishing, 2020. ISBN 978-80-271-1216-6.

Předběžný termín obhajoby

2023/24 LS – PEF

Vedoucí práce

Ing. David Buchtela, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 4. 9. 2023

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 3. 11. 2023

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 15. 03. 2024

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Návrh webové stránky pro členy včelařského spolku" jsem vypracoval 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 autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15.03.2024

Poděkování

Rád bych touto cestou poděkoval vedoucímu mé bakalářské práce, panu Ing. Davidu Buchtelovi, Ph.D., mé rodině za podporu při tvorbě bakalářské práce a Pavlu Tomanovi za cenné informace ohledně Centrálního informačního systému, které jsem ve své práci využil.

Návrh webové stránky pro členy včelařského spolku

Abstrakt

Bakalářské práce se zabývá návrhem webové aplikace pro členy včelařského spolku. Ve veřejné části, která bude přístupna bez autentifikace, budou uvedeny informace o spolku, jeho činnosti a službách pro veřejnost. V neveřejné části této aplikace bude uživatel schopný zadávat informace o svých včelstev, jejich váze, jejich umístění v rámci úlu, hmotnosti, množství medu podle jednotlivých včelstev, či stav úlové váhy. Dále jako člen organizace bude moci diskutovat s ostatními členy spolku ve fóru.

V teoretické části autor popisuje danou problematiku formou literární rešerše. V teoretických východiscích popisuje jednotlivé technologie, ze kterých bude vycházet samostatná realizace vlastní práce. Dále autor porovnává současnou nabídku programů umožňující uchovávání informací o včelstev. Využívá znalosti, které autor čerpá z doporučených zdrojů informací.

Vlastní práce bude aplikací postavenou na frameworku ASP.NET s využitím návrhového vzoru MVC a napsána v jazyce C#. Všechny údaje budou uchovávány v databázi MySQL, která bude normalizována podle platných pravidel, včetně referenčních integrit. Dále bude použito značkovacích jazyků HTML pro zobrazení textu na stránce a CSS pro stylování HTML elementů. Dále bude použit JavaScript pro tvorbu dynamických prvků v aplikaci.

Klíčová slova: včelařský spolek, C#, ASP.NET, Entity Framework, MVC, MySQL, webová aplikace

Design and Implementation of Web Application for Local Beekeepers Association

Abstract

The bachelor thesis deals with the design of a web application for the members of a beekeeping association. In the public part, which will be accessible without authentication, information about the association, activities and services for the public will be presented. In the non-public part of application accessible after authentication by login, the user will be able to enter information about his beehives, their weight, their location within the hive, weight, amount of honey produced by beehives, or the status of the hive scale. Also, as a member of the organization, user will be able to discuss with other members of the association in the forum.

In the theoretical part, the author describes the subject in the form of a literature search. In the theoretical background, individual technologies, which will be the basis for the independent implementation of the work itself, will be described. In the next section, author compares the current range of programs that enable the storage of information about beehives. The knowledge from the recommended sources of information will be used.

The actual work will be an application built on the ASP.NET framework using the MVC design pattern and written in C# programming language. All data will be stored in a MySQL database that will be normalized according to current rules, including referential integrity. In addition, HTML markup languages will be used to display text on the page and CSS will be used to style HTML elements. In addition, JavaScript will be used to create dynamic elements in the application.

Keywords: Beekeepers Association, C#, ASP.NET, Entity Framework, MySQL, Web Application, MVC

Obsah

1 Úvod	11
2 Cíl práce a metodika	12
2.1 Cíl práce	12
2.2 Metodika	12
Teoretická východiska	14
Základní pojmy a technologie	14
Včelařský spolek	14
2.2.1 Webová aplikace	14
2.2.2 Programovací jazyk C#.....	15
2.2.3 .NET.....	16
2.2.4 ASP.NET	17
2.2.5 Model-View-Controller (MVC)	17
2.2.6 Entity Framework	18
2.2.7 MySQL	19
2.2.8 Relační databáze	19
2.2.9 Structured Query Language (SQL)	19
2.2.10 phpMyAdmin.....	20
2.2.11 XAMPP.....	21
2.2.12 HTML	21
2.2.13 CSS (Cascading Style Sheets)	22
2.2.14 JavaScript.....	23
2.3 Popis a porovnání systémů pro včelaře.....	23
2.3.1 Centrální informační systém Českého svazu včelařů	23
2.3.2 Portál včelaře	25
2.3.3 sVčela.....	27
2.3.4 Chytrý včelař.....	27
3 Vlastní práce	30
3.1 Vzorové persony	30
3.1.1 Persona A.....	31
3.1.2 Persona B	31
3.1.3 Persona C	32
3.2 Stanovení uživatelských cílů.....	33
3.3 Pojmy specifikace uživatelského rozhraní	33
3.3.1 Případy užití.....	33
3.3.2 Scénář.....	34
3.3.3 Drátěný model.....	34

3.3.4	Diagram model případu užití	34
3.4	Tvorba specifikace uživatelského rozhraní	35
3.4.1	Úvodní stránka aplikace (HomePage)	36
3.4.1.1	Případy užití (Use Case)	36
3.4.1.2	Scénář	36
3.4.1.3	Drátový model (Wireframe)	37
3.4.2	Přihlášení do neveřejné části (log-in)	37
3.4.2.1	Případy užití (Use Case)	37
3.4.2.2	Scénář	38
3.4.2.3	Wireframe	38
3.4.3	Úvodní stránka přihlášeného uživatele (rozcestník)	38
3.4.3.1	Případy užití (Use Case)	38
3.4.3.2	Scénář	39
3.4.3.3	Wireframe	39
3.4.4	Správa stanoviště	39
3.4.4.1	Případy užití (Use Case)	39
3.4.4.2	Scénář	40
3.4.4.3	Wireframe	41
3.4.5	Správa informací o úlu	41
3.4.5.1	Případy užití (Use Case)	41
3.4.5.2	Scénář	42
3.4.5.3	Wireframe	43
43		
3.4.6	Diskuzní chat	44
3.4.6.1	Případy užití (Use Case)	44
3.4.6.2	Scénář	44
3.4.6.3	Wireframe	44
3.5	Implementace webové aplikace	45
3.5.1	Návrh databáze	45
3.5.2	Propojení aplikace s databází	48
3.5.3	Registrace a přihlášení uživatelů	49
3.5.4	Diskusní chat	50
3.5.5	Zaznamenávání informací o včelách ze strany uživatele	50
3.5.6	Grafy a export do souborů	52
3.5.7	Přidávání článků	55

3.5.8	Přidávání souborů	55
3.5.9	Správa uživatelů.....	56
4	Výsledky a diskuse	57
5	Závěr	58
6	Seznam použitých zdrojů	59
6.1	Seznam obrázků	62
6.2	Seznam použitých zkratk.....	62

1 Úvod

Včelařství patří mezi jedno z nejstarších řemesel na světě. První zmínky o něm máme z období paleolitu. Jako každé řemeslo, postupem času bylo potřeba nějaké spolupráce mezi včelaři. Z období středověku máme zprávy o existenci včelařských cechů. V roce 1872 vznikl Včelařský spolek pro Čechy, ze kterého vznikl dnešní Český svaz včelařů (ČSV), který sdružuje většinu včelařů v České republice.

Pro úspěšné včelaření by si měl každý včelař nějakým způsobem ukládat data o svých včelstev v nějaké podobě. Pokud budeme tato data zpracovávat od všech včelařů v rámci spolku v určité formě, získáme skupinové informace o včelstvech, které můžeme dále zpracovávat a agregovat.

Včelařské spolky spadající pod ČSV tvoří základní organizace (ZO). Tyhle ZO pracují s Centrálním informačním systémem (CIS). K tomuto systému nemají samostatní včelaři přístup, a slouží jen ke komunikaci základní organizace s ČSV. Od září 2023 sice pro samotné členy existuje tzv. Portál včelaře, ten ale zatím neumožňuje plnohodnotnou komunikaci. Pokud se některý včelařský spolek rozhodne být nezávislý na ČSV, nemá oproti členským organizacím žádnou plnohodnotnou platformu. Existují sice různé programy a aplikace, které ale slouží k potřebám samostatných včelařů, bez možnosti odesílat data včelařské organizaci.

Proto se autor této bakalářské práce rozhodl vytvořit aplikaci, která bude sloužit nejen samostatným včelařům, ale celému včelařskému spolku.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem práce bude webová aplikace, která bude sloužit nejen pro potřeby samotného včelaře, ale i pro potřeby celé organizace. Včelař bude moci evidovat informace o svých včelstev, např. počet včelstev, jejich umístění, počet úlů apod. Některé z těchto dat si bude včelař evidovat pro svou potřebu, některá se budou používat pro potřeby spolku, který tak získá informace, které může různě zpracovávat, ať do sestav, nebo do grafů.

Aplikace nebude jen o zpracování vstupních dat od včelařů. Aplikace bude taky obsahovat diskusní fórum, které bude platformou pro vzájemnou komunikaci včelařů v rámci spolku. Pro vnitřní potřeby spolku bude možné ukládat soubory, např. záznamy ze schůzí spolku. To bude umožněno pouze přihlášeným uživatelům. Nepřihlášení uživatelé budou moci přistupovat jen na veřejně přístupnou webovou stránku spolku, kde najdou informace o činnosti spolku, zaměření a službách pro veřejnost.

2.2 Metodika

Metodika bakalářské práce bude rozdělena do dvou částí – teoretické a praktické.

V teoretické části budou popsány technologie, které budou využity při tvorbě praktické části práce. Dále bude provedena analýza a porovnání vybraných aplikací a programů pro potřeby včelařů. Informace o jednotlivých technologiích budou zpracovány formou literární rešerše a budou vycházet z doporučených zdrojů.

V rámci praktické části bakalářské práce bude prvně vytvořen drátěný model (Wireframe) vytvořený v aplikaci Figma. Dále bude z tohoto modelu vycházet vzhled samostatné aplikace, která bude napsána v programovacím jazyce C# ve frameworku ASP.NET, který je určen pro tvorbu webových aplikací. Databáze bude postavena na systému řízení báze dat MySQL. Propojení aplikace a databáze bude zajištěno pomocí objektově-relačního frameworku Entity Framework. Text se bude vykreslovat na stránku pomocí značkovacího jazyka HTML za podpory CSS, což je značkovací jazyk, který slouží

pro stylování jednotlivých HTML elementů. Pro tvorbu grafů a dalších dynamických prvků bude využit programovací jazyk JavaScript.

Teoretická východiska

Základní pojmy a technologie

Včelařský spolek

Vůbec první zmínky o včelařství pochází z období paleolitu (1). Zprávy o včelařství na dnešním území České republiky se nacházejí v kronikách od 5. století. Počátky toho, co dnes nazýváme včelařskými spolky, můžeme najít v 10. století, kdy se včely chovaly při klášterech. (2) Středověk byl ve znamení včelařských cechů, které měly svá práva. Například v roce 1350 udělil král Karel IV. práva včelařům tzv. Diplomem norimberským (3).

Dnes se většina včelařů sdružuje v Českém svazu včelařů (ČSV). Jeho vznik se datuje do roku 1872, kdy byl založen Včelařský spolek pro Čechy. Postupem času docházelo pod vlivem různých událostí k několika změnám názvu, až v roce 1971 vzniká dnešní ČSV (4).

Zatímco v roce 1872 měl svaz při svém založení 600 členů, dnes má více než 54 000 členů a 203 včelařských spolků mládeže, což představuje 98 procent všech včelařů v České republice. Svaz se dělí na krajské koordinační výbory, okresní organizace a základní organizace (5).

2.2.1 Webová aplikace

Webová aplikace je pojem, který označuje kolekci statických a dynamických webových stránek. Jedná se o aplikaci typu klient-server, kdy klient z webového prohlížeče zašle webovému serveru požadavek, na který server klientovi následně odpoví. Tyhle webové aplikace nemají předem přesně napsaný obsah, ten se zobrazuje až podle požadavků uživatele. Díky technologiím HTML5 a CSS3 dochází v poslední době k velkému rozvoji webových aplikací, které v mnoha případech nahrazují nutnost instalovat software na počítač uživatele, a to díky jednoduché aktualizaci a možnosti rozšiřování aplikací. Další jejich výhodou je multiplatformost, kdy k používání webové aplikace stačí jen internetový prohlížeč, a není nutno vytvářet software pro různé operační systémy. (6) V prostředí mobilních operačních systémů lze už vytvořenou webovou aplikaci velmi snadno upravit přímo do podoby mobilní aplikace bez nutnosti tvořit kód zcela od nuly. Takto vytvořená

aplikace postavená na webových technologiích se nazývají Progressive Web App (PWA) (7).

Samotnou webovou aplikaci lze rozdělit na front-end a back-end. Front-end se dá nazvat prezenční částí, se kterou pracuje uživatel. Má na starost rozmístění prvků ve stránce, se kterou pracuje uživatel. Back-end je část, která má na starost práci s daty, která jsou uložena v nějaké formě, často ve formě databáze. Zpracovaná data pak posílá k zobrazení do front-endu. Pro front-end se obvykle vedle značkovacích jazyků HTML a CSS používá programovací jazyk JavaScript, který umožňuje tvorbu dynamických a interaktivních prvků na stránce. Pro back-endové prostředí se většinou používá nějaký z hojně rozšířených programovacích jazyků, např. C#, PHP, Python, Java, Pearl a jiné, či nějaký framework postavený na jednom z těchto jazyků. Pro webové aplikace postavené na jazyce PHP se používá vývojářský balíček XAMPP, pro aplikace na jazyce C# se používá framework ASP.NET.

2.2.2 Programovací jazyk C#

C# (angl. Výslovnost: „C Sharp“) je objektově-orientovaný programovací jazyk vytvořený společností Microsoft a postavený na platformě .NET.

Jedná se o tzv. vysokoúrovňový jazyk, což znamená, že programátor nepracuje ve strojovém kódu na úrovni procesoru, a je proto nutné kód přeložit do univerzálního pomocného jazyka Common Intermediate Language (CIL), což je společné s ostatními programovacími jazyky postavenými na frameworku .NET (8).

Datové typy se v C# dělí na 2 typy: hodnotové a referenční:

- Hodnotové datové typy obsahují určitou hodnotu a dělí se na celočíselné datové typy, reálné datové typy, a na logické datové typy. Příkladem celočíselného typu je např. Integer (int), který má velikost 4 bytů a rozsah od -2,147,483,648 do 2,147,483,647. Příkladem reálného datového typu je

double, který má velikost 8 bytů a rozsah od $\pm 5.0 \times 10^{-324}$ do $\pm 1.7 \times 10^{308}$ s přesností 15-17 desetinných míst. Logický datový typ je jen jeden, a to boolean (bool), který umožňuje uchovávat 2 hodnoty: true a false. Zvláštním hodnotovým datovým typem je Character (char), který umožňuje zapsat jeden specifický znak, a to buď ve formě literátů, nebo ve formě kódu hexadecimální (šestnáctkové) soustavy (9).

- Referenční datové typy se dají nazvat odkazem (referencí) na konkrétní hodnoty. Sem patří např. řetězce, pole a kolekce. Řetězec je sekvencí znaků (datový typ char), pole a kolekce jsou seskupením stejných datových typů či tříd. Rozdíl mezi polem a kolekcí je dán přesnou velikostí pole, kdy při nutnosti změny pole je nutné například cyklem přenést data ze starého pole o velikosti X do nového pole o velikosti Y, přičemž je nutné, aby platilo, že velikost pole Y je větší, než velikost pole X. U kolekce tento problém není, protože kolekce nemá předem danou velikost. Do ní se dají přidávat a odebírat jednotlivé prvky podle potřeby (9).

Syntaxe tohoto jazyka vychází z programovacích jazyků C++ a Java, zároveň se ale v některých částech liší. Asi největším rozdílem od C++ je to, že C# je čistý objektový jazyk, všechny třídy dědí ze společného předka, čímž je třída Object. Dalším rozdílem od obou programovacích jazyků je podpora přetěžování operátorů (konstruktorů). Oproti těmto programovacím jazykům má prostředí .NET tzv. Garbage Collector, to znamená, že nepoužívané objekty budou z paměti automaticky odstraněny (8).

2.2.3 .NET

.NET (dříve .NET Core) je open-source multiplatformní framework od společnosti Microsoft, který podporuje programování v programovacích jazycích C#, F# a Visual Basic (dříve VB.NET). Jedná se o nástupce původního .NET Frameworku, který na rozdíl od současného .NET neumožňoval tvorbu projektů pro platformy mimo Windows. Tento framework lze díky jeho knihovnám využít např. na tvorbu konzolových aplikací, desktopových formulářových aplikací pomocí Windows Forms či novějšího WPF,

mobilních aplikací postavených na platformě Xamarin, či webové aplikace na frameworku ASP.NET (10).

Díky společnému běhovému prostředí Common Language Runtime (CLR) a společnému mezijazyku Common Intermediate Language (CIL) lze knihovny používat pro všechny 3 jazyky, což garantuje vzájemnou kompatibilitu knihoven a umožňuje mít projekt napsaný v několika podporovaných programovacích jazycích. Jejich vzájemná spolupráce je dána standardem Common Type System (CTS), který definuje reprezentaci a popis datových typů v paměti, jaké operace mohou datové typy vykonávat a jak mají být spravovány v CLR. Z CIL se kód přeloží do samotného strojového kódu pomocí překladače Just in Time (JiT). Kód, který běží pod CLR, se nazývá kódem řízeným. To znamená, že využívá služeb CLR a je pod jeho kontrolou (8).

2.2.4 ASP.NET

ASP.NET (dříve ASP.NET Core) je frameworková nadstavba pro tvorbu webových aplikací a souvisejícího API postavená nad základním frameworkem .NET (11).

Základem fungování ASP.NET jsou tzv. Razor Pages, které umožňují vkládat .NET kód do HTML v rámci souborů s koncovkou .cshtml, a to ve spolupráci s tzv. Tag Helpers, což jsou speciální vlastnosti HTML elementů začínající na předponu „asp-“ a umožňují např. zpracovávání dat z formulářů a jejich následné vykreslení právě pomocí Razorů. Vložený kód se následně zobrazí klientovi v převedené formě do klasického HTML (12).

V praxi se používá při tvorbě webových aplikací návrhového vzoru Model-View-Controller, zkráceně MVC.

2.2.5 Model-View-Controller (MVC)

Model-View-Controller (zkráceně MVC) je návrhový vzor využívány při tvorbě aplikací a programů s grafickým vizuálním prostředím (GUI). Tento návrhový vzor se používá pro rozdělení kódu podle jeho funkce a vymezení toho, k čemu má jaká část sloužit.

Tím se zabrání tomu, aby v rámci jedné komponenty nebyl definován model a jeho logika společně s konečným výstupem. Kód je rozdělen do 3 částí:

- Model představuje datovou základnu aplikace a samotnou logiku. Rozumí se tím samotná definice objektu a jeho vlastnosti. Pokud se v rámci projektu používá databáze, měl by být model definován podle jednotlivých atributů v databázi.
- View (pohled) se dá nazvat finálním výsledkem, který se zobrazí uživateli v prohlížeči. Jedná se o prezenční vrstvu, kde uživatel může zadávat data, která se mají zpracovat, nebo naopak zobrazit již zpracovaná data.
- Controller (kontrolér) představuje prostřední část, která zajišťuje komunikaci mezi modelem a pohledem.

Celý princip tohoto vzoru funguje následovně: Na základě adresy a parametrů zachytí požadavek Router, který ho podle parametrů v adrese přesměruje na správný Kontrolér. Následně, je-li požadavek na získání dat (ať už výchozím nastavením Kontroléru, nebo na základě zasílaných parametrů), vyžádá si Kontrolér z Modelu příslušná data, která si uchovává v proměnných. (V případě ASP.NET se tyto proměnné označují ViewBag). Po zpracování dat Kontrolér pošle výsledek na Pohled, který se zobrazí uživateli, a se kterým může dále pracovat (13).

2.2.6 Entity Framework

Entity Framework je podle definice Microsoftu „*objektově-relační mapování, které umožňuje pracovat s relačními daty přes doménově-specifikované objekty*“. Laicky řečeno můžeme říci, že se jedná o framework, který umožňuje vzájemný přenos dat mezi vnější databází a projekty aplikací, které jsou postavené na platformě .NET. To zahrnuje základní operace s daty známé pod označením CRUD. Framework podporuje nejenom databázové systémy od společnosti Microsoft, ale i vybrané databáze třetích stran. Entity Framework umožňuje práci s daty pomocí vlastních metod, nebo pomocí dotazovacího jazyka LINQ (14).

2.2.7 MySQL

MySQL je označení pro open-source multiplatformní relační systém řízení báze dat (DBMS) vyvíjený původně švédskou společností MySQL AB, dnes společností Oracle. Podle webu db-engines.com se jedná o druhý nejpoužívanější SŘBD, hned po databázi Oracle od stejné společnosti (15). Pro práci s databází se používá jazyk SQL. MySQL ukládá stejně jako ostatní relační SŘBD svá data do tabulek, podporuje vyhledávání v nich pomocí dotazů, příslušné CRUD operace, tvorbu triggerů (spouštěčů) a pohledů (16).

2.2.8 Relační databáze

Relační databázi se rozumí uspořádaná množina dat v podobě relace (tabulky), která se skládá z prvků (řádků) a atributů (sloupců). Celá databáze funguje podle teorie relačně databázového modelu. Každý atribut v relaci má svůj název a typ, který musí prvek splňovat. Každý prvek by měl obsahovat jednoznačně identifikovatelný údaj. Ten se nazývá primární klíč a zajišťuje jednoznačnou identifikaci prvku. Pokud nějaký údaj odkazuje na primární klíč uložený v jiné relaci, nazývá se tento údaj cizím klíčem.

Relační databáze by měla splňovat určité charakteristiky. Údaje v relacích by měly být atomické, neboli dále nedělitelné. Dále by měly být údaje sklární, což znamená, že mají pouze jeden rozměr. Každý uložený údaj by měl být dostupný kombinací názvu relace (tabulky), názvu atributu (sloupce) a hodnoty primárního klíče pro určitý prvek (řádek). To bývá ve většině SŘBD postavených na jazyku SQL implementováno v podobě příkazu SELECT. Systémy řízení báze dat (SŘBD) nazýváme prostředím propojující aplikační prostředí s daty uloženými v databázi. Operace s uloženými daty v databázi lze provádět právě přes SŘBD. Mezi známé SŘBD patří např. Microsoft SQL Server, PostgreSQL, SQLite či IBM Db2 (17).

2.2.9 Structured Query Language (SQL)

Structured Query Language (SQL) je deklarativní dotazovací jazyk vytvořený společností IBM v 70. letech minulého století, který umožňuje přes SŘBD pracovat s daty

uloženými v databázi. Jedná se o jazyk deklarativní, což znamená, že se deklaruje cíl, nikoliv postup, jak provést operaci. Slovo dotazovací v jeho definici označuje, že se jedná o jazyk pracující s daty nad určitou databází, nelze se na něj dívat jako na běžný programovací jazyk. V rámci standardizace by se měly psát příkazy velkými písmeny. Jazyk SQL se v základu dělí na 4 části:

- Příkazy pro definici dat (DDL) slouží pro práci se strukturou databáze, tzn. s databází, tabulkami, indexy, pohledy, sekvencemi, procedurami a triggerem. Základ tvoří příkazy CREATE, ALTER, DROP a DELETE.
- Příkazy pro manipulaci s daty (DML) tvoří základ fungování jazyka SQL a slouží k provádění základních operací CRUD. Základ tvoří příkazy SELECT, INSERT, UPDATE a DELETE.
- Příkazy pro řízení dat (DCL) umožňují přidělování práv jednotlivým uživatelům v rámci databáze. Data je možné přidělovat jednotlivým uživatelům, nebo hromadně v rámci určitých skupin. Základ tvoří příkazy GRANT a REVOKE.
- Příkazy pro kontrolu transakcí (TCL) umožňují provádět transakce. Základ tvoří příkazy COMMIT a ROLLBACK (17).

2.2.10 phpMyAdmin

phpMyAdmin je bezplatný softwarový nástroj, který slouží ke správě SŘBD MySQL přes webové rozhraní. Kromě MySQL podporuje i MariaDB, což je fork vzniklý z MySQL. phpMyAdmin prostřednictvím SŘBD podporuje časté operace prováděné v databázi, jako správa tabulek, sloupců, relací, uživatelů atd. Kromě toho nabízí i export tabulek z databáze do několika formátů mimo SQL, včetně nejznámějších v podobě CSV či XML. phpMyAdmin běží na modulu Apache, který je součástí vývojářského prostředí XAMPP (18).

2.2.11 XAMPP

XAMPP je multiplatformní vývojářské prostředí napsané v jazyce PHP. Obsahuje modul Apache, který může sloužit jako webový server pro aplikace napsané v jazyce PHP, nebo jako webový server pro prostředí phpMyAdmin, které umožňuje práci s SŘBD MySQL. Ten může fungovat v rámci XAMPPu v podobě modulu MySQL, tak i mimo tento modul. Apache kromě toho nabízí prostředí pro jazyk Pearl. Další z funkcí, které XAMPP nabízí, je např. modul pro instalaci FileZilla serveru, který slouží jako úložiště dat, která mohou být uživateli stažena z koncových stanic pomocí protokolu FTP (File Transfer Protocol) (19).

2.2.12 HTML

HTML (HyperText Markup Language) je značkovací jazyk určený pro vykreslování textu na webovou stránku. Za vývojem jazyka stojí konsorcium W3C, ale nejnovější verzi jazyka, jmenovitě HTML5, vyvíjí pracovní skupina WHATWG. Výsledky práce této skupiny zapojuje W3C do své specifikace.

Základní syntaxe jazyka je tvořena tzv. elementy, které tvoří základní prvek jazyka HTML. Elementy jsou vymezeny hranatými závorkami („<“, „>“), které se nazývají tagy. Mezi tagy lze uvést text, který se bude zobrazovat na webové stránce. Element je tedy tvořen tagy a textem mezi ním. Některé tagy se píšou jen jednou a nevyžadují uzavírací část, příkladem je tag
, kdy text za tímto tagem zalomí na další řádek. V rámci tagu se dají přidávat atributy, které mají hodnotu. Tyhle atributy se používají např. pro vložení adresy stránky, na kterou se odkazuje, nebo popisují prvek v rámci formuláře. Velkou výhodou jazyka HTML je možnost vnořování elementů, jeden element tak může v sobě obsahovat více elementů a kód tak hierarchicky členit. Ve verzi HTML5 lze tohle vnořování elementů využít k popsání struktury pomocí sémantických elementů.

Elementy neslouží jenom k zobrazování textu. Elementy mohou vykreslovat např. i obrázky, či sloužit jako přehrávač videa či audia. Zvláštními elementy jsou prvky formulářů, např. input field, radiobox, nebo checkbox.

Formuláře představují kolekci různých elementů, kam může uživatel zadávat údaje. Formuláře se dají zpracovávat dvěma metodami, GET a POST. Metoda GET umístí vložené hodnoty ze strany uživatele do adresního řádku, čehož je většinou využíváno např. při vyhledávání ve vyhledávači. Z principu se tento přenos dat přes tuto metodu nedá považovat za bezpečný. Pro přenos zabezpečených dat se používá metoda POST, která vložené hodnoty skryje do těla HTTP požadavku.

Nejnovější verze jazyka ve verzi HTML5, jehož finální verze byla vydána v roce 2014, usnadňuje práci s videem. Největší předností nové verze je ale možnost strukturálního členění stránky. Toho je docíleno novými sémantickými elementy header, main, footer, aside, nav, article a section, doplněnými o figure a figcaption (20).

2.2.13 CSS (Cascading Style Sheets)

CSS (Cascading Style Sheets) je značkovací jazyk určený pro stylování elementů a textů napsaných v jazyce HTML. Nejnovějším standardem tohoto jazyka je CSS3. Jazyk je vyvíjen konsorciem W3C.

Existují 3 způsoby, jak aplikovat stylování CSS na HTML. Zaprvé jako atribut „style“ v tagu elementu, zadruhé jako element v souboru HTML, nebo samostatně v rámci samostatného CSS souboru, který bude propojen přes tag <link> v HTML souboru.

CSS se skládá ze selektoru, vlastnosti a hodnoty. Selektor určuje HTML prvek, na který se bude CSS vztahovat. Jako selektor můžeme označit element, třídu, nebo ID. To se dá kombinovat, můžeme tak jako selektor použít všechny elementy ve určité třídě. Vlastnost označuje, co se bude v rámci selektoru měnit. Hodnota v rámci vlastnosti určuje, jak konkrétně se změní vlastnost v rámci selektoru. Například v rámci selektoru, který může být v podobě elementu „h1“ vlastnost „color“ vyjadřuje změnu barvy a hodnota „red“ určuje, že se barva elementu změní na červenou.

Kromě základních aplikací CSS, jako např. změna barvy textu, nastavení obrázku v pozadí stránky, nebo odstranění podtržení odkazu nabízí CSS možnosti flex nebo grid jako hodnoty v rámci vlastnosti display. Hodnoty flex a grid určují rozmístění jednoho či více prvků v rámci určeného elementu. Rozmístění v rámci flexu může mít podobu řádky nebo sloupce. Hodnota grid umožňuje rozmístění prvku nebo prvků v rámci pole, čili řádků a sloupců najednou. Flex může být tak použit např. pro rozmístění prvků pod sebou (jako

sloupec) v rámci menu, zatímco grid řeší rozmístění prvků v rámci celé stránky. Jelikož HTML umožňuje vnořování více elementů v rámci jednoho elementu, lze flex a grid vzájemně kombinovat (20).

2.2.14 JavaScript

JavaScript je vysokoúrovňový programovací jazyk, který slouží pro tvorbu dynamického obsahu na webové stránce na straně klienta. Název Java v názvu jazyka je uveden čistě z marketingových důvodů, nejedná se o nadstavbu jazyka Java, jde jen o jazyk, který se Javou jenom částečně inspiroval. Jazyk ale umožňuje i tvorbu komplexních her, nebo různých miniaplikací. Hodně populární jsou různé javascriptové knihovny a frameworky, jako např. jQuery či React (21).

2.3 Popis a porovnání systémů pro včelaře

2.3.1 Centrální informační systém Českého svazu včelařů

Centrální informační systém Českého svazu včelařů (CIS ČSV) slouží zástupcům základních a okresních organizací, které spadají pod ČSV. Na jeho vývoji spolupracoval ČSV se společností FormSoft. Tento CIS umožňuje evidenci a zpracování osobních údajů pro potřeby všech složek ČSV. Přístup do systému má pouze pověřený člen ZO či OO, nikoliv všichni členové (22).

Pro potřeby základních organizací je CIS rozdělen do několika níže uvedených nabídek:

- Nabídka Nástěnka slouží jako hlavní stránka celého uživatelského systému. Najdeme zde aktuality od administrátora CIS, základní číselné statistiky o základní organizaci a životní jubilea.
- V nabídce Uživatelé najdeme uživatele, kteří mají přístup k CIS. Příslušný admin ZO zde může jednotlivým uživatelům přidělovat práva, která budou mít v systému.

- Nabídka Členové obsahuje seznam všech členů (včelařů), kteří jsou členy příslušné ZO. Jednotlivé členy lze filtrovat podle několika kritérií, jako např. podle rodného čísla, bydliště, či funkce. Seznam členů lze exportovat do několika tiskopisů, např. pro potřeby prezenční listiny na schůzích ZO či souhlas se zpracováním osobních údajů. Dále je možné určit vedoucího kroužku mládeže, který bude vést včelařský kroužek přidružený k organizaci. V rámci seznamu lze zobrazit detail jednotlivých členů (včelařů) na tzv. Kartě včelaře, kde lze o nich evidovat základní osobní údaje, seznam stanovišť se včelstvy včetně informací o konkrétním stanovišti, roční statistiky včelaře, a seznam čerpaných dotací. Mezi další vedlejší možnosti v Kartě včelaře patří objednávka svazového časopisu Včelařství, funkce a ocenění včelaře, vkládání dokumentů o včelaři, evidování jeho dosaženého odborného vzdělávání pro potřeby ČSV, či možnost ukončit členství, např. v důsledku úmrtí.
- Nabídka Dotace a příspěvky obsahuje přehled dotací a příspěvků. Všichni členové ZO jsou povinni každoročně platit členský poplatek. V této nabídce se zobrazuje u každého člena, zda je poplatek zaplacen, či nikoliv. Dále nabídka poskytuje přehled žádostí o dotace, o které žádala ZO.
- Nabídka Vzdělávací akce obsahuje přehled jednotlivých školení pro potřeby ZO. Každá ZO totiž může pro své členy uspořádat školení v určité oblasti. K dispozici je seznam přednášejících, kteří nabízejí ZO školení v různých oblastech pro potřeby členů ZO.
- Nabídka Nastavení ZO slouží pro potřeby základní organizace. Slouží např. pro nastavení základních údajů o základní organizaci, k určení funkcionářů základní organizace, k určování výše členského příspěvku, svolávání členských schůzí, či k zjišťování ochranných pásem při nákazách včel.
- Nabídka Statistika nabízí výběr statistických dat zpracovaných ve formě tabulky, která je řazena podle roku. Dále nabídka obsahuje souhrnné statistiky včelařů ze všech ZO ČSV včetně filtrace podle několika parametrů. Dále je k dispozici mapa úhynů včelstev podle okresů.

- Nabídka Úložiště obsahuje přehled nahraných materiálů ze strany správce CIS.

Co se týče hodnocení CIS ze strany autora práce, tak kladně hodnotí intuitivní prostředí, což zahrnuje rozmístění ovládacích prvků a položek v menu. Kladně také hodnotí generování a následný export různých dokumentů do několika formátů. Méně kladně hodnotím nemožnost vkládat do CIS některé údaje ze strany řadových členů. Pro vložení těchto informací je kontaktovat příslušného člena ZO s přístupem do CIS, který požadované údaje vloží. Dále nabídka Úložiště neumožňuje nahrávat soubory jen pro potřeby vybrané ZO. Jinak práci v CIS hodnotí autor kladně (22) (23).

2.3.2 Portál včelaře

Portál včelaře při ČSV je novou webovou aplikací pro včelaře spadající pod ČSV. Pilotní verze byla spuštěna v září 2023 a plný provoz bude zahájen 1. listopadu 2023. Cílem nového Portálu včelaře je zjednodušit informovanost členské základny. Zároveň dochází k sjednocení elektronické komunikace pro samotné členy ČSV. Oproti CIS ČSV je Portál přístupný všem, nikoliv jen pověřením pracovníkům.

Základ aplikace tvoří menu, které tvoří nabídky Moje údaje, Základní organizace, Důvěrníci, Ankety, Příspěvky, Požadavek na léčiva, Výsledky vyšetření, Veřejný registr půdy, IZR a Časopis:

- Nabídka Moje údaje obsahuje podnabídky obsahující osobní údaje o včelaři, přehled jeho stanovišť, roční statistiku, přehled vyplacených dotací, příspěvky k úhradě a jejich historie, či výpis z Integrovaného zemědělského registru.
- Nabídka Základní organizace obsahuje základní informace o ZO, seznam funkcionářů, kontakty na ně a přehled katastrálních území, která spadají pod příslušnou ZO.
- Nabídka Důvěrníci obsahuje seznam důvěrníků příslušné organizace. Každý včelař spadá pod určitého důvěrníka, pokud je ZO má.

- Nabídka Ankety není před nasazením do plného provozu zprovozněna. V budoucnu by měla dávat členům ČSV možnost hlasovat a vyjadřovat se ohledně dění v ČSV.
- Nabídka Příspěvky je jen odkazem do nabídky Moje údaje do stejnojmenné podnabídky.
- Nabídka Požadavek na léčiva není před nasazením do plného provozu zprovozněna. V budoucnu by měla obsahovat možnost objednat léčiva pro včelstva.
- Nabídka Výsledky vyšetření obsahuje, jak název vypovídá, výsledky vyšetření včelstev na různé nemoci, jako je např. mor včelího plodu, nebo varoáza.
- Nabídka Veřejný registr půdy je odkazem do geografického informačního systému LPIS, který je spravován Ministerstvem zemědělství ČR.
- Nabídka Časopis odkazuje na webovou stránku časopisu Včelařství, který nabízí pro své předplatitele množnost prohlížení v elektronické verzi.

Co se týče hodnocení Portálu ze strany autora bakalářské práce, tak oceňuje snahu do budoucna zjednodušit elektronickou komunikaci pro všechny členy ZO, čemuž právě Portál včelaře dává základy. Autor oceňuje intuitivní uživatelské prostředí včetně responzivního přizpůsobení podle zařízení a možnost zobrazit si informace o svých včelstvech a výsledcích vyšetření. Bohužel celý Portál v době před spuštěním ostré verze neobsahuje přímo možnosti komunikace či vkládání různých údajů tak, jako to nabízí CIS. Autor tak v tomto směru doufá ve zlepšení a přidání dalších funkcí, které se podle ČSV do budoucna přidají do portálu, jako je např. mapa prodejců medu, požadavky na změnu osobních údajů, či bonusový portál partnerů ČSV (24).

2.3.3 sVčela

sVčela je název aplikace umožňující uchovávat informace o včelstev ze strany včelaře. Aplikace je k dispozici jak na mobilní telefony s operačním systémem Android, tak i pro počítače s operačním systémem Windows. Za aplikací stojí Petr Janda, který vystupuje pod obchodním názvem PJ Software. Aplikaci má bezplatnou verzi bez licence a placenou s licenci. Rozdíl mezi nimi je v tom, že bezplatná verze umožňuje spravovat jen jednoho včelaře, 1 stanoviště a maximálně 5 úlů (25).

V desktopové verzi je rozdělena aplikace do dvou menu. První menu slouží k nastavení vlastností, jako je např. včelař, jeho stanoviště a úly, plemenáčky a oplodňáčky, či informace o krmivu, léčení a způsobu ošetřování včelstev. V rámci tohoto menu lze zadat do jednoho stanoviště více úlů nebo uchovávat informace o odběratelích medu. Druhé menu slouží k zobrazení dat v tabulce, jako záznamy o stanovištích, chovu včel, medu, či krmení včelstev. Informace v tomhle menu se dělí na záznamy, chov, med, krmení, léčení a finance (26).

Co se týče hodnocení autorem práce, tak kladně hodnotí možnost včelaře si poznamenat informace o svých včelách a možnost zálohy dat na úložiště DropBox. Jako velkou nevýhodu ovšem vidí neintuitivní a matoucí prostředí, kdy např. oplodňáčky nejdou nastavit v nabídce Stanoviště v prvním menu, ale až v druhém menu v nabídce Chov v podnabídce Oplodňáčky. Při zavření dialogového okna tlačítkem storno často dochází k zobrazení chyby, resp. neošetřené výjimky. Export dat je možný pouze ve formě dialogu pro tisk, nelze si zvolit jiný formát.

2.3.4 Chytrý včelař

Chytrý včelař (anglicky Smart Beekeeper) je název aplikace pro mobilní operační systémy Android a iOS, existuje také jako webová aplikace pro zobrazení v prohlížeči. Stejně jako ostatní aplikace tohoto druhu, umožňuje uchovávat informace o včelstev ze strany včelaře. Za vývojem této aplikace stojí tým vývojářů PunkHive pod vedením Víta Zaorala.

Uživatel této aplikace má na výběr, zda pro její použití založí účet v aplikaci, nebo zda bude pracovat bez účtu. Standardní uživatel může zadávat údaje jen v mobilní aplikaci. Pokud by je chtěl zadávat i v prostředí webového prohlížeče, musí si zakoupit Prémiový účet. Aplikace má 3 hlavních nabídky a 5 doplňkových nabídek.

- V nabídce Moje včely si uživatel zadává stanoviště, pod která spadají jednotlivé úly. U úlů lze evidovat jejich typ, míru rámků, datum založení rámků, typ včelstva a počet nástavků. Dále lze evidovat, zda včelstvo v úlu uhynulo, zda se v něm nachází mateří mřížka, krmivo, výkluz, česnová zábrana, zaklíckovaná matka, gabonový pásek či odpařování. Každý úl má svoji matku, u té lze zadávat informace o datumu vylíhnutí a přidání do úlu, původu či linii. Dále lze evidovat, kdy došlo ke krmení včelstva v úlu.
- S nabídkou Moje záznamy bude uživatel pracovat nejčastěji. Bude zde uchovávat záznamy o úlech, které si definoval v nabídce Moje včely. U stanoviště si může zadat manuálně počasí, nebo počasí nastavit automaticky podle zadané polohy. U jednotlivých úlů má na výběr možnost popsat stav česna, podmetu a podložky na základě předpřipravených možností. Dále zaznamenávat informace o stavu včelstva, evidovat stav plodu, sílu včelstva a jejich náladu, množství zásobních plátů či různé nemoci. V podnabídce Zásahy lze zaznamenat různé zásahy do úlu, jako např. odstranění mateří mřížky, zaklíckování matky, či zadání počtu rámků, nástavků nebo mezistěn. V podnabídce Léčení lze uvést druh a množství podaného léčiva. Obdobně lze evidovat druh a množství krmení ve stejnojmenné podnabídce. Jako doplněk je možné ke každému úlu vložit jeho fotku.
- V nabídce Medobraní může uživatel evidovat údaje o vytočeném medu. Eviduje se zde, na jakém stanovišti byl vytočen med, o jaký druh medu se jedná, hmotnost vytočeného medu, vlhkost medu v procentech a z jakých úlů pochází med.
- V nabídce Moje žihadla, jak už sám název vypovídá o sobě, eviduje počet žihadel od včel.

- V nabídce Spad roztočů může uživatel nahrát fotku podložky z úlu. Algoritmus následně rozpozná na fotografii roztoče a orientačně odhadne jejich počet

Další nabídky – Kalkulačka, Kalendář matek a Nahlásit sršně – vedou na stažení příslušné aplikace v Apple App Storu, je-li aplikace nainstalována na iOS, nebo na Google Play, je-li aplikace nainstalována na Androidu.

Co se týče hodnocení, tak autor chválí intuitivní uživatelské prostředí, aplikace používá UI Kit příslušného operačního systému, na kterém je nainstalována. Dále chválí způsob, jak lze evidovat údaje o včelstvech v jednotlivých nabídkách, včetně filtrace záznamů podle parametrů. Autor by ocenil možnost evidovat detailnější informace z hlediska financí, neboli možnost zaznamenávat finanční údaje o prodeji medu či nákupu krmiva. Kromě toho by ocenil v Premium verzi možnost exportu do více formátů, než PDF (27).

3 Vlastní práce

V teoretické části bakalářské práce byla provedena rešerše ze seznamu doporučené literatury. Nyní dochází k tvorbě vlastní práce. V rámci zásad tvorby uživatelského prostředí budou vytvořeny a popsány vzorové osoby uživatelů. Na základě jejich charakteristik budou definovány uživatelské cíle (User Goals). Dále bude vytvořen drátěný model aplikace (wireframe), z kterého bude vycházet uživatelské prostředí celé aplikace. Drátěný model bude spojen s scénáři a případy použití (Use Case). Bude vytvořen UML diagram, který bude popisovat strukturu a vztahy databáze, která bude s aplikací spojena. Dále bude vytvořen i UML diagram aplikace, podle kterého dojde k jejímu naprogramování podle návrhového vzoru MVC ve frameworku ASP.NET a programovacím jazyce C#.

3.1 Vzorové osoby

Personami se nejen v oblasti návrhu softwaru rozumí fiktivní postavy, které mají vlastnosti typických uživatelů tvořeného softwaru. Tyto vzorové osoby často vznikají během rozhovorů s uživateli, nebo jsou výsledkem statistického zjišťování. Pomáhají tak vývojářům uzpůsobit program, stanovit uživatelské cíle a případy užití tak, aby co nejlépe vyhovovaly potřebám budoucích uživatelů. Osoby se dělí na 3 typy.

První typ osoby představuje typického uživatele. Bude využívat všechny funkce budovaného programu, který by měl být vyvíjen podle jeho potřeb. To zahrnuje grafické rozhraní aplikace a případy užití.

Druhým typem osoby je příležitostný uživatel. Podle tohoto druhu uživatele netvoříme grafické rozhraní, ale měl by být schopný dosáhnout svých cílů v rámci programu, neboli případech užití.

Třetím typem osoby je záporná osoba, někdy nazývána též jako antipersona. Tenhle typ osoby se nikdy neseťká s naším softwarem, proto se její požadavky neberou v úvahu (28).

3.1.1 Persona A

Jméno a příjmení	Jiří Novák
Věk	43
Pohlaví	Muž
Zaměstnání	Obsluha CNC frézky
Koníčky a zájmy	Včelaření, lyžování, cyklistika
Popis	Pan Jiří pracuje v pracovní dny jako obsluha CNC frézy v místní továrně. Přes víkend se věnuje své rodině a chovu svých včelstev na své velké zahradě u svého domu. Je členem včelařského spolku, který nemá svojí agendu nijak zdigitalizovanou. Z tohoto důvodu hledá nějaký způsob jak komunikovat se spolkem přes internet a zároveň si vést záznamy o svých včelstvech v přehledné podobě.

3.1.2 Persona B

Jméno a příjmení	Bedřich Novotný
Věk	67
Pohlaví	Muž
Zaměstnání	Důchodce
Koníčky a zájmy	Včelaření, sledování fotbalových zápasů v TV, diskuze s kamarády v hospodě
Popis	Pan Bedřich je důchodce, který svůj volný čas věnuje svým včelám. Je členem včelařského spolku a vlastní maringotku s několika včelstvy u nedalekého lesa. Informace o svých včelách nijak zvlášť neviduje. Myslí si, že jeho paměť je natolik dobrá, že nepotřebuje psát různé údaje na papír, natož někam do internetové aplikace.

3.1.3 Persona C

Jméno a příjmení	Patricie Tománková
Věk	29
Pohlaví	Muž
Zaměstnání	Programátorka v začínajícím start-upu
Koníčky a zájmy	Včelaření, zahradničení, programování
Popis	Patricie pracuje během pracovního týdne jako programátorka v začínajícím start-upu. Svou zálibu ohledně včelaření zdědila po svém otci včelaři. Patricie je členkou místního včelařského spolku a má jedno včelstvo na pronajatém pozemku, které chce ale postupem času rozšířit. Ráda by informace o svých včelách evidovala někde v internetové aplikaci.

3.2 Stanovení uživatelských cílů

Po vytvoření vzorových person bude na základě jejich společných požadavků vytvořeny uživatelské cíle. Daný cíl by měl uživatel splnit co nejrychleji a zároveň co nejsnadněji. (28).

Uživatel by měl být schopný splnit následující cíle:

- Zjistit základní informace o spolku a jeho aktivitách
- Přihlásit se přes formulář do neveřejné části
- Komunikovat s ostatními členy spolku v diskusním chatu
- Najít informace vedení spolku určené členům
- Mít přehled o členských poplatcích včelařského spolku
- Evidovat informace o umístění svých včelstev v rámci stanovišť
- Zaznamenávat informace o krmení a léčbě svých včelstev
- Dále zpracovávat agregované údaje v několika formátech

3.3 Pojmy specifikace uživatelského rozhraní

V rámci tvorby specifikace uživatelského rozhraní budou na základě uživatelských cílů vytvořeny případy užití a scénáře, které budou sloužit pro tvorbu drátěného modelu aplikace. Podrobnější popis případů užití pro jednotlivé stránky bude realizován slovně, model případů užití v rámci celé aplikace bude realizován pomocí diagramu realizovaném v jazyce UML.

3.3.1 Případy užití

V oblasti nejen softwarového inženýrství se případem užití (Use Case) rozumí seznam akcí nebo kroků událostí zachycující interakci mezi rolí (v rámci jazyka UML aktérem) a systémem za účelem dosažení vytyčeného cíle. Z pohledu HCI a interakčního designu se jedná o popis potřeb uživatele. Případy užití jsou vždy iniciovány aktérem a zároveň vždy psány z jeho pohledu. Případ užití lze popsat slovně, nebo graficky, např. v jazyce UML. Před tvorbou samotných případů užití je nutné definovat hranice systému a aktéry.

Hranice systému (v rámci jazyka UML subjekt) zachycuje systém, s kterým bude z vnějšku provádět interakci aktér. Špatná definice hranic může způsobit nemožnost znázornění některých interakcí. V případě této práce budeme za subjekt považovat samotnou webovou aplikaci pro členy včelařského spolku.

Aktérem se rozumí člověk, nebo vnější systém, který přichází z vně systému a chce dosáhnout svých cílů. V rámci slovního popisu případu užití rozlišujeme, zda aktér očekává, nebo požaduje své cíle. V rámci této práce budeme pracovat s třemi aktéry – nepřihlášeným uživatelem, přihlášeným uživatelem a aktérem (28) (29).

3.3.2 Scénář

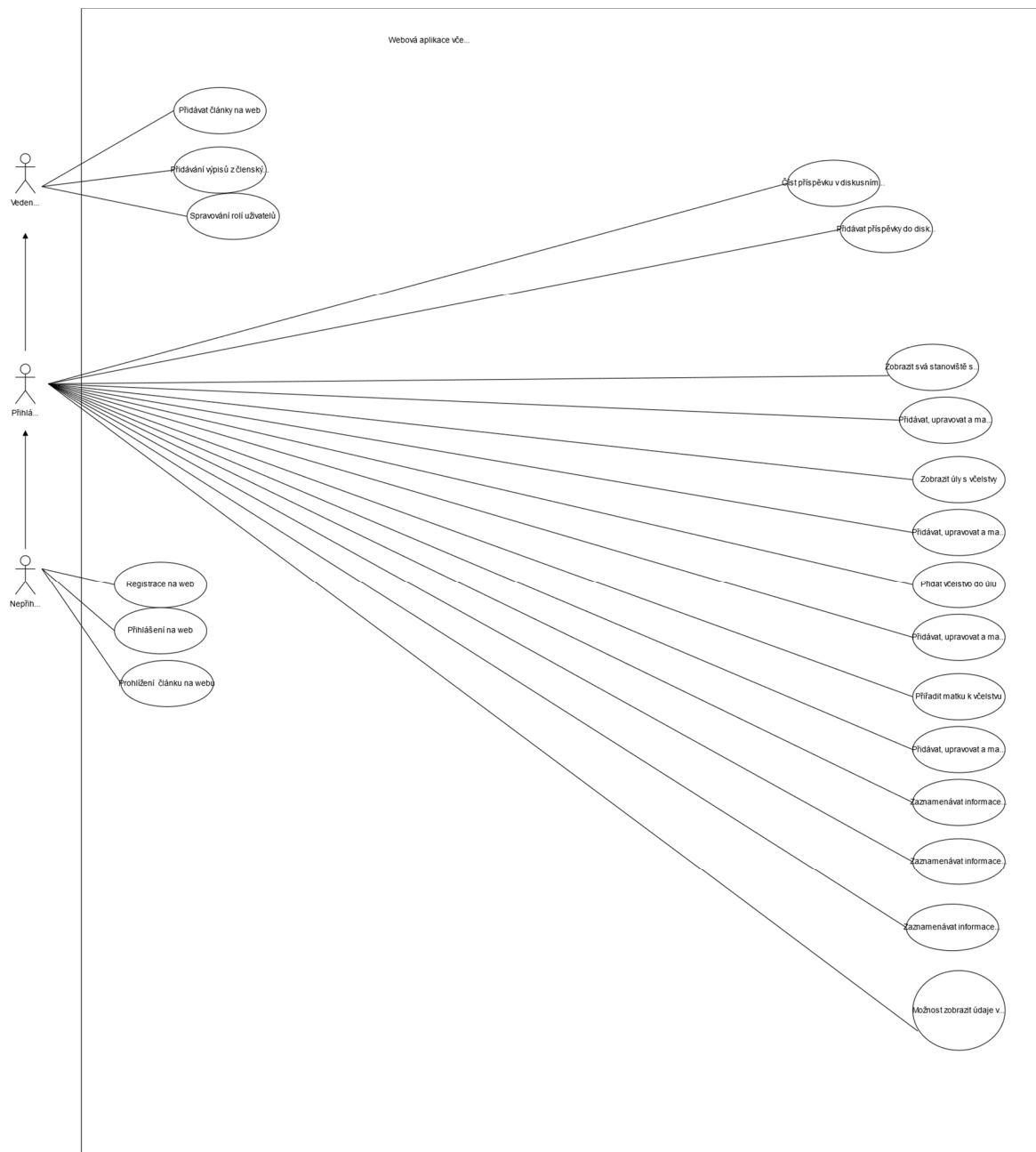
Scénář (Scenario) popisuje, jak bude systém reagovat na cíle uživatele uvedené v případech užití. Z pohledu interaktivního designu se doporučuje, aby byl scénář napsán z pohledu systému. Scénář se obvykle skládá z několika krátkých vět (28).

3.3.3 Drátěný model

Drátěný model (Wireframe) představuje kostru webu. Určuje umístění jednotlivých prvků na webu tak, aby co nejlépe plnily daný cíl. Drátěné modely neobsahují barvy, typ zvoleného písma ani grafiku. Model popisuje, jaké prvky bude stránka obsahovat a kde budou umístěny. Po tvorbě modelu se již tvoří samotný design stránku a vzhled prvků v ní (28).

3.3.4 Diagram model případu užití

Diagram modelu případu užití (Use Case Diagram) využívá prvky jazyka UML ke grafické reprezentaci daných případu užití v rámci aplikace. Zobrazuje vymezení hranic systému, v tomto případě webová aplikaci, dále zobrazuje aktéry, přičemž ke každému přiřazuje use-case, pomocí kterého pracuje s aplikací. Jak je vidět na diagramu, nejvíce use-case bude mít přihlášený uživatel, který má mnoho požadavků a očekávání, co by měl systém umožňovat, aby uspokojil jeho potřeby (29).



Obrázek 1 - Diagram případu užití pro webovou aplikaci včelařského spolku.

3.4 Tvorba specifikace uživatelského rozhraní

Tvorba specifikace uživatelského rozhraní bude popsána na vybraných stránkách webové aplikace. Pro každou stránku budou použity případy užití, scénáře, a z nich vycházející drátový model. Ten bude vytvořen v aplikaci Figma.

3.4.1 Úvodní stránka aplikace (HomePage)

3.4.1.1 Případy užití (Use Case)

Na úvodní stránce aktér očekává:

- Nalezení základních informací o spolku
- Přehled poskytovaných služeb veřejnosti (např. prodej medu)
- Přehled článků od členů spolku
- Kontaktní údaje na vedení spolku

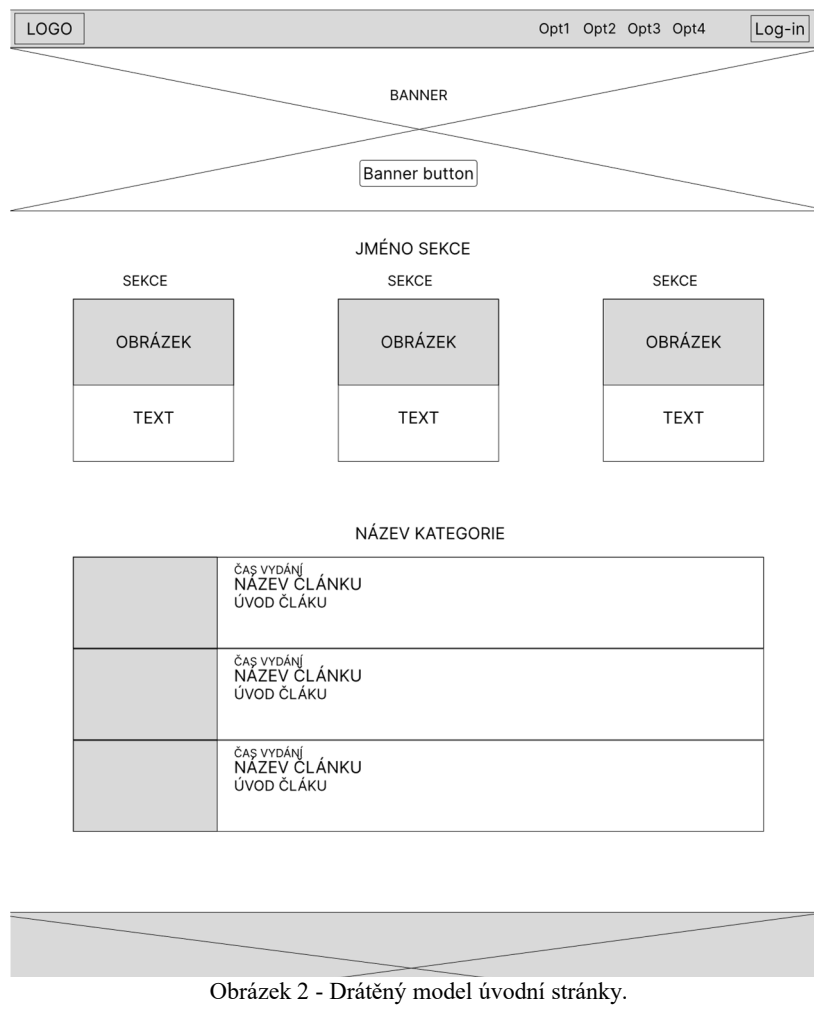
Dále aktér požaduje:

- Tlačítko pro přihlášení do neveřejné sekce
- Možnost prohlížet si obsah článků

3.4.1.2 Scénář

Po příchodu aktéra na úvodní stránku systém zobrazí pod horním menu banner informující aktéra o základním účelu stránky. Pod tímto bannerem se nachází sekce s několika podsekcemi informující o činnostech spolku. Po kliknutí na jednu z těchto podsekcí dojde k přesměrování aktéra na stránku věnující se příslušné podsekcí. Následně následuje sekce obsahující přehled článků. Po kliknutí na některý z nich se zobrazí článek v celé délce.

3.4.1.3 Drátový model (Wireframe)



3.4.2 Přihlášení do neveřejné části (log-in)

3.4.2.1 Případy užití (Use Case)

Uživatel očekává:

- Možnost se přihlásit do neveřejné části pro členy spolku
- V případě, že není členem, možnost registrace

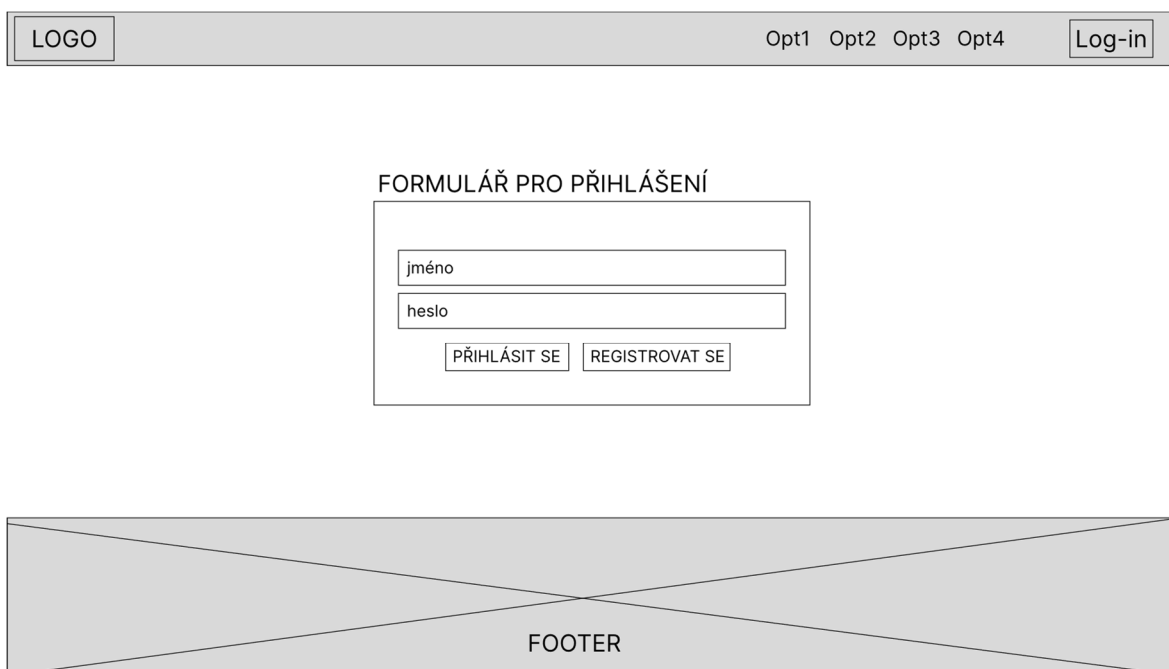
Uživatel požaduje:

- Zobrazení formuláře, do kterého zadá své přihlašovací údaje

3.4.2.2 Scénář

System zobrazí formulář, kde je nutné pro přihlášení zadat e-mail člena a jeho heslo. Pod formulářem se nachází dvě tlačítka. První tlačítko zajistí přihlášení uživatele do systému. V případě, že aktér nemá účet, resp. ještě není členem spolku, kliknutím na druhé tlačítko se zaregistruje.

3.4.2.3 Wireframe



Obrázek 3 - Drátěný model přihlašovací stránky (Log-in)

3.4.3 Úvodní stránka přihlášeného uživatele (rozcestník)

3.4.3.1 Případy užití (Use Case)

Uživatel očekává:

- Možnost dostat se na příslušnou stránku poskytující informace o jeho včelstvech
- Možnost dostat se do diskusního chatu
- Možnost dostat se do nastavení svého účtu

- Možnost dostat se k agregovaným datum

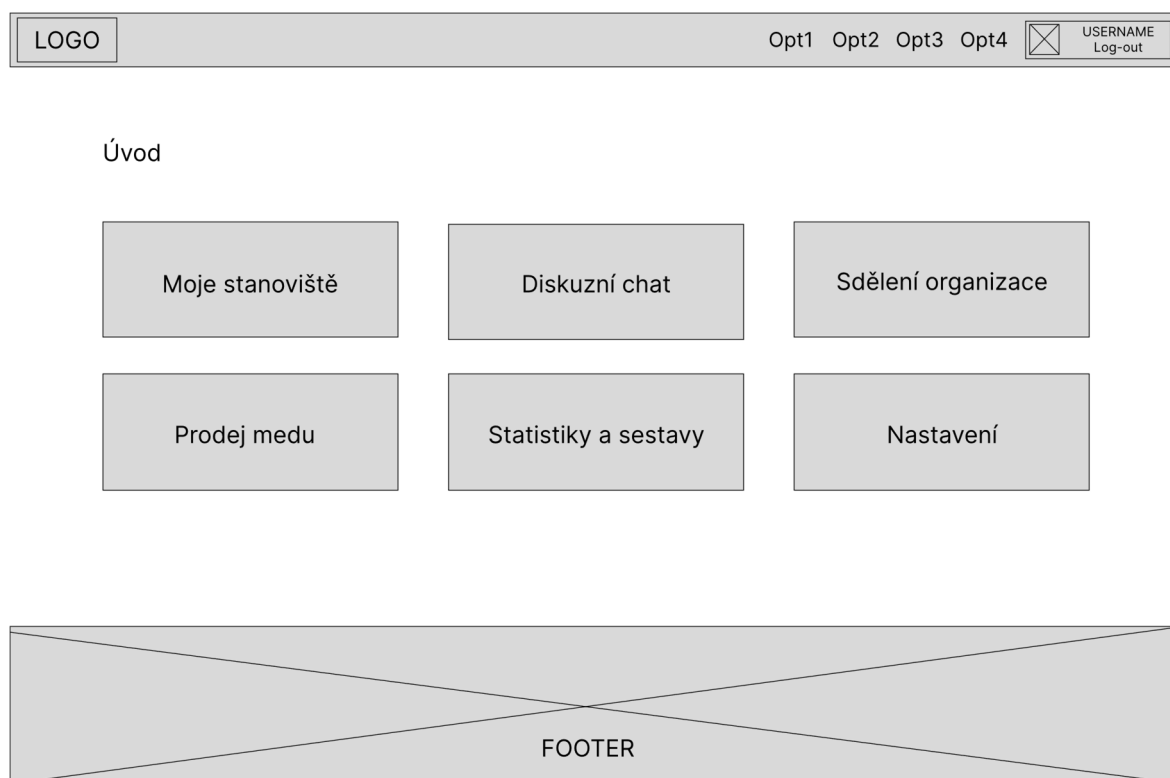
Uživatel požaduje:

- Zorientovat se v menu
- Využít menu k splnění jeho potřeb

3.4.3.2 Scénář

Po úspěšném přihlášení uživatele se zobrazí menu, ve kterém kliknutím na jednu z možností se dostane na stránku s příslušnou akcí.

3.4.3.3 Wireframe



Obrázek 4 - Drátěný model úvodní stránky přihlášeného uživatele.

3.4.4 Správa stanoviště

3.4.4.1 Případy užití (Use Case)

Uživatel očekává:

- Zobrazení informací o jeho vybraném stanovišti
- Seznam úlů na jeho stanovišti

- Krátký popis jednotlivých úlů na stanovišti

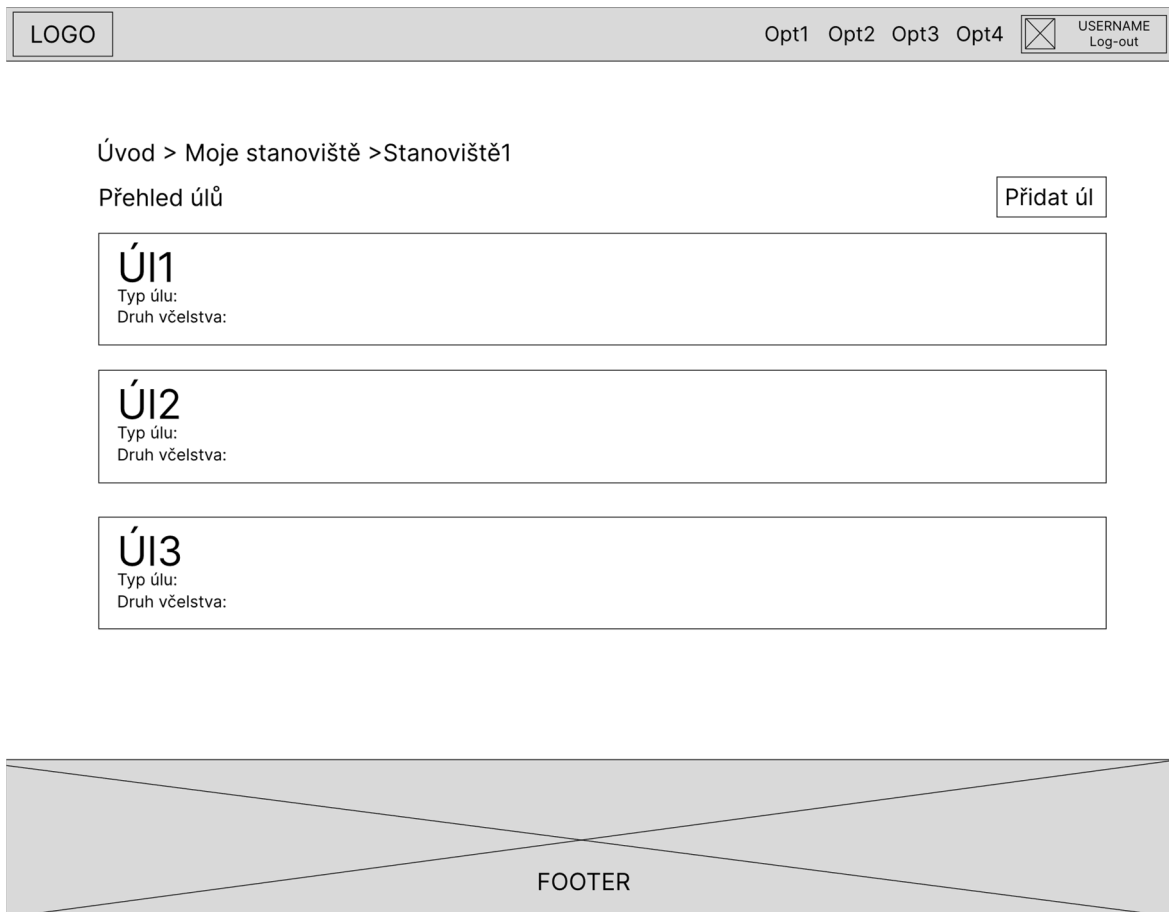
Uživatel požaduje:

- Možnost přidání úlu na stanoviště
- Zobrazit detail jednotlivých úlů

3.4.4.2 Scénář

Po příchodu uživatele na stránku zobrazí systém seznam jednotlivých úlů. U každého úlu je jeho krátký popis, o jaký druh úlu a typ včelstva se jedná. Po kliknutí na jeden z úlů se zobrazí detailnější informace o úlu. Po kliknutí na tlačítko přidat stanoviště se uživateli zobrazí formulář, ve kterém uvede základní informace o něm, např. o jaký typ úlu se jedná, kde se nachází apod.

3.4.4.3 Wireframe



Obrázek 5 - Drátěný model stránky pro správu stanoviště.

3.4.5 Správa informací o úlu

3.4.5.1 Případy užití (Use Case)

Uživatel očekává:

- Možnost získat informací o úlu a včelstvu v něm
- Možnost získat informace o matce včelstva
- Možnost zobrazit si záznamy o krmení včelstva

- Možnost zobrazit si záznamy o léčení včelstva
- Možnost zobrazit si záznamy o vyprodukovaném medu

Uživatel požaduje:

- Zapisovat chronologicky údaje o krmení včelstva
- Zapisovat chronologicky údaje o léčení včelstva
- Zapisovat chronologicky údaje o vyprodukovaném medu

3.4.5.2 Scénář

Po zobrazení stránky se zobrazí základní údaje o úlu, včelstvu a matce včelstva. Po přiřazení matky ke včelstvu má uživatel možnost zadat údaje o matce. Dále má možnost evidovat chronologicky různé údaje o včelstvu v tabulkách, např. jejich krmení, léčení množství vyprodukovaného medu. Do každé z této tabulek může přidávat údaje prostřednictvím stejnojmenného tlačítka, které zobrazí formulář pro zadání údajů. Dále může evidovat různé chronologické poznámky, např. vizuální stav úlu, náladu včelstva, úpravy úlu, či stav zásob.

3.4.5.3 Wireframe

LOGOOpt1 Opt2 Opt3 Opt4USERNAME
Log-out

Úl1

Upravit údaje

Základní informace o úlu

Typ úlu:
Míra rámku:
Počet vložených rámků:
Počet nástavků:

Základní informace o včelstvu Přidat údaje

Datum přidání do úlu
Původ matky:
Linie matky:

Základní informace o matce Přidat údaje

Datum přidání:
Původ matky:
Linie matky:

Krmení Přidat údaje

Datum krmení	Krmená látka	Množství	Jednotky	

Léčba Přidat údaje

Datum léčby	Léčebná látka	Druh látky	Množství	Jednotky	

Med Přidat údaje

Datum vytáčení	Druh medu	Množství	Nástavek	

Dodatečné informace Přidat údaje

Datum	Poznámka	

FOOTER

Obrázek 6 Drátěný model přehledu úlu a včelstva v něm.

3.4.6 Diskuzní chat

3.4.6.1 Případy užití (Use Case)

Uživatel očekává:

- Možnost komunikovat s ostatními členy spolku v chatu

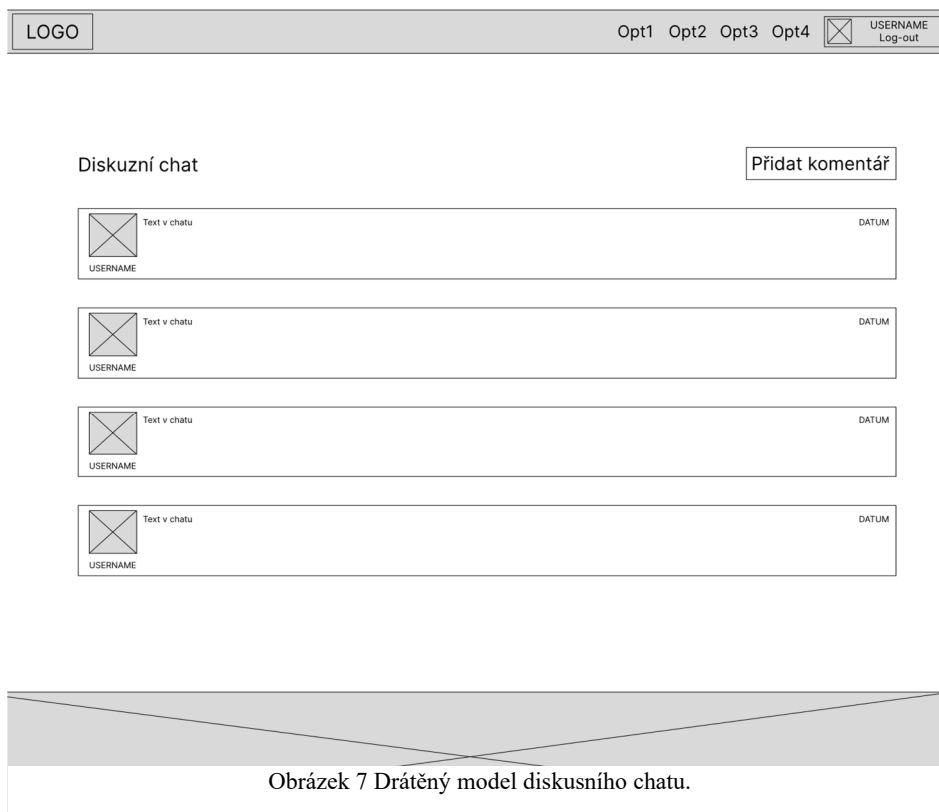
Uživatel požaduje:

- Přidávat komentáře do chatu

3.4.6.2 Scénář

Po zobrazení stránky uživatel vidí komentáře všech ostatních členů v chatu. Po kliknutí na tlačítko přidat komentář v pravém horním rohu se zobrazí formulář, do kterého napíše zprávu, která se následně zobrazí v chatu.

3.4.6.3 Wireframe



3.5 Implementace webové aplikace

3.5.1 Návrh databáze

Pro popis databáze, se kterou bude pracovat webová aplikace, byl vytvořen diagram v jazyce UML. Vzhledem k rozsahu problematiky, kterou se tato práce zabývá a v rámci pravidel normalizace se v databázi nachází celkově 13 tabulek. Nejdůležitější tabulkami jsou vcelari a vcelstva.

V tabulce vcelari jsou evidovány základní informace o včelaři, jeho jméno, adresa, a určení toho, zda je ověřeným členem a zda je adminem, který má právo přidávat články na web a ověřovat členy. Zápis této role je zajištěn pomocí datového typu bit umožňující zachovávat 2 hodnoty – true a false.

Pokud má včelař práva administrátora, může publikovat články na webu ve veřejné části. Veškeré informace o článcích se ukládají do tabulky clanky. Zde je uváděn název článku, jeho prefix, obsah, autor a datum publikování.

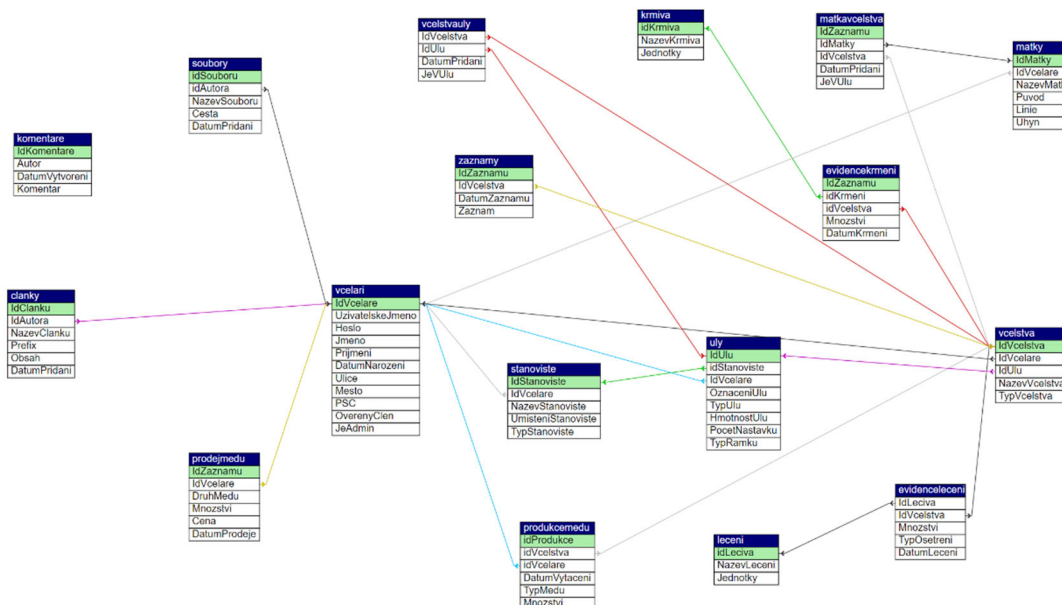
Další důležitou tabulkou je tabulka vcelstva. V rámci této tabulky se název a typ včelstva.

Tabulka uly, tak jak vypovídá její název, obsahuje informace o úlu. Jmenovitě se zde eviduje označení úlu, jeho typ, typ rámků a jejich počet.

Jednotlivé úly se nachází na stanovištích, které jsou uvedeny v tabulce stanoviste. Zde se eviduje název stanoviště, umístění a jeho typ.

Každé včelstvo má svou matku. Informace o ní se evidují v rámci tabulky matky. V té se eviduje její název, původ, linie a její úhyn v podobě stejnojmenného atributu, který má přiřazen datový typ bool. Matka se přiřazuje ke včelstvu prostřednictvím tabulky matkavcelstva.

Dále musí být včelstvo léčeno různými léčivy. Léčiva jsou zaznamenána v tabulce lecení, kde je uveden název léčebné látky a jednotka, ve které je léčivo podáváno. Záznam o podání



Obrázek 8 diagram databáze.

léčiva včelstvu je zaznamenán v příslušné tabulce evidenceleceni, kde je uvedeno množství podané léčebné látky, jakým způsobem bylo včelstvo ošetřeno a datum podání látky.

Produkce medu je zaznamenávána v tabulce produkcemedu. V té je evidován typ vyprodukovaného medu, množství a datum vytáčení.

Veškeré další informace týkající se včelstva se evidují v tabulce zaznamy. Zde se eviduje datum zadání záznamu a jeho obsah.

Obdobně je důležité myslet na to, že i včely se musí krmit. Seznam krmných látek je k dispozici v tabulce krmiva, kde primární klíč je přiřazen k atributu krmiva. Dále je evidován název krmiva a jednotka podaného krmiva. Samotný záznam o podání krmiva je evidován v tabulce evidencekrmeni. Je zde uvedeno množství podané krmné látky a datum krmení.

Vyprodukovaný med lze i prodávat. Od toho slouží tabulka prodejmedu. V té je uveden druh medu, množství, cena a datum prodeje.

V rámci webové aplikace mohou členové komunikovat v chatu. Jejich příspěvky se ukládají do tabulky komentare. Zde se eviduje autor komentáře, co napsal a kdy to napsal.

Obdobně slouží i tabulka clanky pro clanky publikované na webu. Zde se eviduje název článku, jeho úvod (prefix), obsah a datum přidání.

Všichni uživatelé mohou ukládat v rámci aplikace soubory. Od toho je tabulka soubory, kde se eviduje název souboru, cesta k němu na serveru a datum přidání.

3.5.2 Propojení aplikace s databází

Důležité pro fungování celé aplikace je důležité zajistit funkční zajištění jejího propojení s databází. Toho je docíleno pomocí objektově-relačního mapování EntityFramework, které bylo přidáno do prostředí jako balíček přes Správu balíčků NuGet (NuGet Package Manager). Základ tvoří balíček Microsoft.EntityFrameworkCore, který z důvodu zajištění různých ostatních funkcí pro práci s databází doplňují následující balíčky:

- Microsoft.EntityFrameworkCore.Design
- Microsoft.EntityFrameworkCore.Proxies

Samotné propojení s databází je realizováno ve třídě Context, kde se nachází metoda OnConfiguring obsahující řetězec umožňující přístup k databázi. Řetězec ze skládá z adresy serveru, názvu databáze na serveru, jméno pro připojení k databázi a heslo.

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
{
    optionsBuilder.UseMySQL("server=127.0.0.1;database=dbspolek;user=root;password=root");
}
```

Obrázek 9 Metoda zajišťující propojení aplikace s databází

Dále, aby bylo zajištěna konzistence dat při vzájemném přenosu, včetně správných datových typů, došlo v aplikaci k vytvoření tříd, jejíž názvy odpovídají názvům tabulek v databázi. V těchto třídách jsou nadeklarovány proměnné v příslušných tabulkách tak, aby jejich název a datový typ odpovídal stejnojmenným sloupcům v příslušné tabulce. Tyto třídy se nachází ve složce Models tak, aby byl dodržován princip fungování vzoru MVC. Tyto třídy s proměnnými jsou následně vloženy do třídy Context jako kolekce DbSet. Díky této kolekci lze tak konzistentně pracovat s hodnotami v databázových tabulkách.

3.5.3 Registrace a přihlášení uživatelů

Aby mohli uživatelé plnohodnotně používat aplikaci, musí se registrovat a následně se mohou do aplikace přihlásit. Celý proces registrace a přihlášení probíhá na kontroléru HomeController.

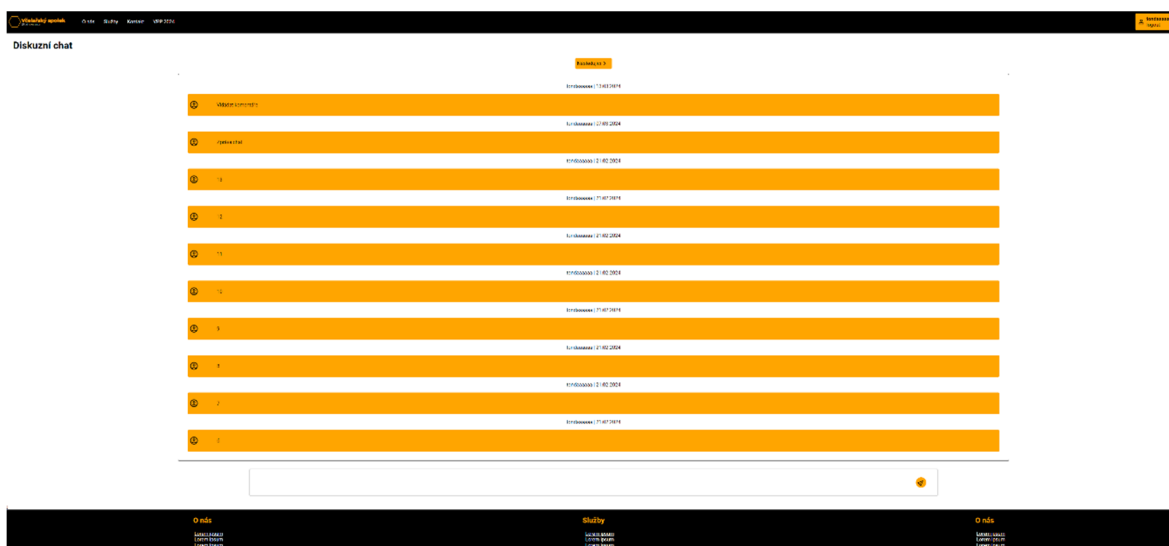
Chce-li se uživatel registrovat, klikne v pravém horním rohu záhlaví (header), následně se mu zobrazí menu, kde vybere možnost „Registrovat se“. Následně bude přesměrován na formulář, kde zadá uživatel uživatelské jméno, pod kterým bude uživatel viděn v systému a heslo. Následně v další části formuláře uživatel zadává svoje Jméno, příjmení, datum narození a kontaktní údaje. Po kliknutí na tlačítko „Registrovat se“ budou údaje zadány do databáze. Před samotným odesláním uživatelských dat se provede validace, zda byly všechny položky vyplněny. Pokud ne, bude uživatel vyzván k doplnění chybějících údajů.

Aby nebylo z databáze možné vyčíst heslo uživatele v čitelné podobě, bude heslo zašifrováno pomocí hashovací funkce SHA-2 ve verzi SHA-256. To je možné díky naimportování direktivy System.Security.Cryptography do kontroléru Home, kde probíhá veškeré zpracování dat z formuláře.

Pokud se chce uživatel přihlásit, obdobně jako při registraci klikne do záhlaví na možnost „Registrovat se“. Následně se mu zobrazí formulář, kde zadá uživatelské jméno a heslo. Následně se provede na kontroléru dojde k zavolání dotazu z databáze, zda existuje uživatel v databázi. Pokud ano, zkontroluje se následně šifra právě zadaného hesla pomocí funkce SHA-2 s šifrou v databázi. Pokud jsou šifry shodné, nastaví se do session řetězec obsahující název „ID“ s hodnotou ID uživatele získanou při volání dotazu. Pokud se chce následně uživatel odhlásit, klikne v záhlaví na možnost „Odhlásit se“, čímž bude vyčištěna session od relace uživatele.

3.5.4 Diskusní chat

Pro komunikaci mezi členy spolku je zřízen diskusní chat. Do chatu se dostane po kliknutí na možnost „Chat“ po úspěšném přihlášení v menu. Komentáře do chatu může uživatel přidávat pomocí formuláře umístěném v dolní části stránky. Na stránce se zobrazuje 10 posledních příspěvků, mezi ostatními příspěvky se může uživatel pohybovat pomocí tlačítek umístěnými nad chatem. Příspěvky aktuálně přihlášeného uživatele se zobrazují oranžově, příspěvky ostatních uživatelů jsou bílé. V případě, že se autor komentáře rozhodne opustit spolek a dojde k smazání jeho účtu, jeho komentáře budou zachovány. Proto v databázi není vztah mezi tabulkou obsahující komentáře a samotným včelařem.



Obrázek 10 Ukázka chatu

3.5.5 Zaznamenávání informací o včelách ze strany uživatele

Největší část celé aplikace tvoří část věnovaná možnosti evidovat informace o včelstvech uživatele. K té se uživatel dostane po úspěšném přihlášení do neveřejné části – pouze pro včelaře v rámci spolku. Tato část je rozdělena do několika stránek. Všechny níže uvedené stránky mají společný technický základ, který spočívá v zavolání dotazu v příslušné funkci v kontroléru, který vrátí hodnoty, které se následně vypíší do pohledu prostřednictvím hodnoty ViewBag. Stránky zároveň obsahují formuláře, které umožňují přidávat či aktualizovat zadané údaje uživatelem do příslušných tabulek v databázi. Děje se tak prostřednictvím funkcí, na které odesílá formulář data prostřednictvím atributu „asp-for“.

Základ tvoří menu „Moje včely“, které slouží jako rozcestník na zbylé stránky. Jedná se o stránky:

- Moje stanoviště
- Prodej medu
- Matky
- Správa léčiv
- Správa krmiv

Stránky „Správa léčiv“ a „Správa krmiva“ jak už název napovídá, slouží uživatelům jako seznam dostupných léčiv a krmiv. Každý uživatel si může přidat svoje vlastní léčivo nebo krmivo do seznamu, přičemž každé bude v seznamu dostupné i pro ostatní uživatele. U obou položek se zaznamenává jejich název a jednotka, ve které budou položky podávány. Léčivo či krmivo pak může použít na příslušné stránce k přidání do úlu.

Další stránkou je stránka „Správa matek“, která slouží pro evidenci dostupných matek. Na rozdíl od dvou předchozích stránek uživatel vidí a může pracovat pouze s matkami, které si přidal. U matek uživatel eviduje její název, původ, linii a to, zda uhynula. Stránka umožňuje samozřejmě kromě přidávání matek i jejich úpravu. Pokud uživatel při úpravě zaškrtně checkbox označující uhynulou matku, matka se stane uzamčenou pro budoucí úpravy a zároveň dojde k jejímu odstranění z úlu, kam byla přidána.

Na stránce „Prodej medu“ si uživatel vede informace o prodaném medu. Je zde uvedeno, kdy došlo k prodeji medu, o jaký druh medu se jednalo, jaké bylo jeho množství a za kolik se prodal.

Hlavní stránkou, resp. podstránkami v rámci této části je „Moje stanoviště“. V této stránce se nachází přehled všech stanovišť, které má uživatel k dispozici. Na každém stanovišti je vždy několik úlů. Po kliknutí na příslušné stanoviště uvidí uživatel všechna svoje včelstva. Po dalším kliknutí na příslušný úl může už uživatel evidovat přímo informace o příslušném včelstvu. K dispozici má několik tabulek. Např v tabulce o včelstvu eviduje název včelstva, typ včelstva a datum přidání do úlu. Dále se zde nachází tabulka o matce. Po kliknutí na příslušné tlačítko se zobrazí pomocí JavaScriptu formulář, ve kterém vybere si může vybrat

dostupné matky, které se ještě nenacházejí v jiném úlu a které nejsou uhynulé. Dále se zde nacházejí dvě tabulky – krmiva a léčiva, umožňující chronologický záznam o podaném léčivu. Další tabulka se týká medu – zde může uživatel evidovat, kdy, jaký druh medu a v jakém množství byl získán z příslušného včelstva. Poslední tabulka obsahuje záznamy včeláře umožňující zadat k příslušnému datu poznámku. Tímto způsobem si může zaznamenat, např. stav úlu, náladu včelstva, provedenou fumigaci apod.

```
[Route("HoneyAdd/{idVcelstva:int}/{id:int}")]
[HttpPost]
public IActionResult HoneyAdd(HiveDetailModel hdm, int idVcelstva, int id)
{
    vcelstva vcelstvo = this.context.vcelstva.Where(x => x.IdUlu ==
id).FirstOrDefault();
    hdm.produkcedu.idVcelstva = vcelstvo.IdVcelstva;
    hdm.produkcedu.idVcelare =
Convert.ToInt32(this.HttpContext.Session.GetString("ID"));
    this.context.produkcedu.Add(hdm.produkcedu);
    this.context.SaveChanges();
    return RedirectToAction("HiveDetail", new { id });
}
```

Obrázek 11 Ukázka kódu pro přidání získaného medu ze včelstva

3.5.6 Grafy a export do souborů

Uživatel má v rámci aplikace také možnost zobrazit množství vytočeného medu v rámci úlu. K tomu slouží po přihlášení možnost „Statistiky a sestavy“. Po kliknutí na tuto možnost se zobrazí formulář, kde vybere svoje včelstvo a období, z kterého chce zobrazit data. Po kliknutí na tlačítko „Nastavit“ dojde k provedení dotazu do databáze. Jelikož probíhá dotazování na získání dat z více tabulek, nelze tak použít běžnou metodu přes context, který umožňuje práci jen s jednou tabulkou. Proto bylo přistoupeno k řešení tohoto dotazu pomocí integrovaného jazyka pro dotazování LINQ. Následně předají data do grafu, který je realizován pomocí JavaScriptového API CanvasJS od společnosti Fenopix.

Kromě výsledků v grafu, který lze stáhnout jako obrázek, má uživatel pod ním tlačítka na stažení výsledků ve formátech CSV a JSON. Po kliknutí na příslušné tlačítko odpovídající formátu se soubor vygeneruje a uloží na server. Aby nebylo disk serveru zahlcen těmito soubory, po každém stažení se soubor ze serveru vymaže.

```

public IActionResult Stats(DateTime first, DateTime last, vcelstva vc)
{
    List<vcelstva> Vcelstva = this.context.vcelstva.Where(x => x.IdVcelare ==
    Convert.ToInt32(this.HttpContext.Session.GetString("ID"))).ToList();
    this.ViewBag.Vcelstva = new SelectList(Vcelstva, "IdVcelstva", "NazevVcelstva");
    List<DataPoint> dataPoints = new List<DataPoint>();
    List<ExportList> ep = new List<ExportList>();

    using (var db = new Contex())
    {
        var select = (from pm in db.produkcedu
        from v in db.vcelstva
        where pm.idVcelstva == v.IdVcelstva
        where v.IdVcelstva == vc.IdVcelstva &&
        (pm.DatumVytaceni >= first && pm.DatumVytaceni <= last)
        select new
        {
            Mnozstvi = pm.Mnozstvi,
            DatumFiltr = pm.DatumVytaceni,
            Datum = Convert.ToString(pm.DatumVytaceni.Day) + ". " +
            Convert.ToString(pm.DatumVytaceni.Month) + ". " +
            Convert.ToString(pm.DatumVytaceni.Year),
            Vcelstvo = v.NazevVcelstva
        }
        ).OrderBy(d => d.DatumFiltr);

        foreach (var result in select)
        {
            dataPoints.Add(new DataPoint(Convert.ToString(result.Datum), result.Mnozstvi));
            ep.Add(new ExportList { Datum = Convert.ToDateTime(result.Datum), Mnozstvi =
            result.Mnozstvi });
        }
    }

    string test = JsonConvert.SerializeObject(dataPoints);
    ViewBag.DataPoints = JsonConvert.SerializeObject(dataPoints);
    this.HttpContext.Session.SetString("ExportList", JsonConvert.SerializeObject(ep));
    ViewBag.popisek = JsonConvert.SerializeObject("Graf vytočeného medu podle zadaného
    včelstva");

    return View();
}

```

Obrázek 12 Princip fungování grafu na straně serveru

```

public async Task<IActionResult> Export(string type)
{
    string pathname = "";
    int counter = 0;
    List<ExportList> export_list = new List<ExportList>();
    if (type == "csv")
    {
        export_list =
JsonConvert.DeserializeObject<List<ExportList>>(this.HttpContext.Session.GetString("ExportList"));
        pathname = "file.csv";
        using (StreamWriter writer = new
StreamWriter(@"wwwroot\file\temp\file.csv"))
        {
            foreach (ExportList zaznam in export_list)
            {
                if (counter == 0)
                {
                    writer.WriteLine("{0};{1}", nameof(zaznam.Mnozstvi),
nameof(zaznam.Datum));
                    writer.WriteLine("{0};{1}", zaznam.Mnozstvi,
zaznam.Datum);
                    counter = counter + 1;
                }
                else
                {
                    writer.WriteLine("{0};{1}", zaznam.Mnozstvi,
zaznam.Datum);
                }
            }
        }
    }
    else if (type == "json")
    {
        pathname = "file.json";

        using (StreamWriter writer = new
StreamWriter(@"wwwroot\file\temp\file.json"))
        {

            writer.Write(this.HttpContext.Session.GetString("ExportList"));

        }
        var filepath = Path.Combine(Directory.GetCurrentDirectory(),
@"wwwroot\file\temp", pathname);
        var provider = new FileExtensionContentTypeProvider();
        if (!provider.TryGetContentType(filepath, out var
contenttype))
        {
            contenttype = "application/octet-stream";
        }
        var bytes = await
System.IO.File.ReadAllBytesAsync(filepath);
        System.IO.File.Delete(filepath);
        return File(bytes, contenttype, Path.GetFileName(filepath));
    }
}

```

Obrázek 13 Ukázka kódu zajišťujícího export do souborů

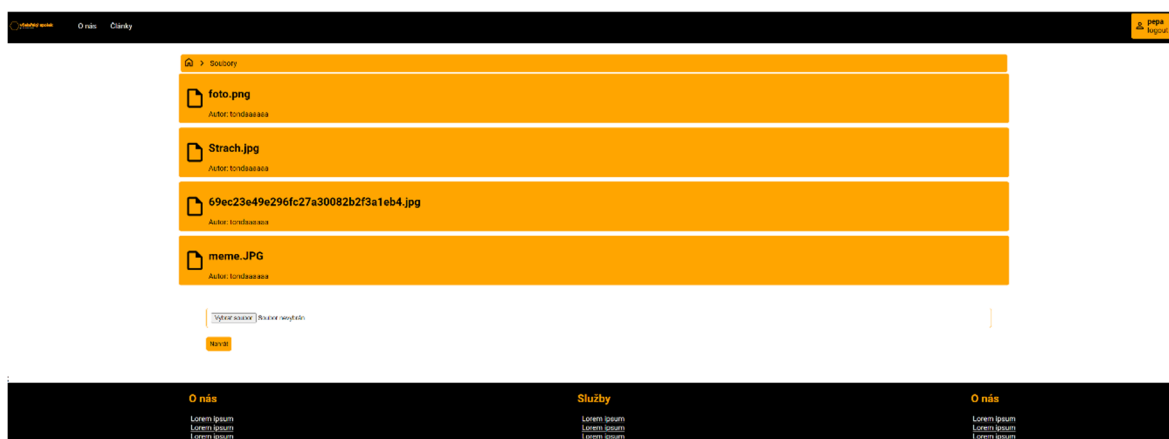
3.5.7 Přidávání článků

Členové spolku, kteří byli administrátorem ověřeni, mohou přidávat články, které budou viditelné i nepřihlášeným členům. Články se přidávají buď tlačítkem na stránce s výpisem článků, nebo tlačítkem v záložce Administrace po přihlášení.

U článků je možné zadávat název článku, perex (úvod) a obsah. Perex a obsah se na stránce zadává do textového editoru TinyMCE, který umožňuje základní formátování textu tak, jako ostatní běžné textové editory, např. změna fontu a jeho velikosti, barvy, zarovnání, řádkování apod. Kromě těchto třech údajů se do databáze zaznamenává datum přidání článku a jeho autor. Obdobně jako v případě chatu se na stránku vypisuje 10 posledních článků a mezi ostatními lze pohybovat pomocí tlačítek v horní části stránky.

3.5.8 Přidávání souborů

Všichni uživatelé mohou přidávat také nahrávat soubory, které mohou ostatní uživatelé stáhnout. Na stránku se soubory se mohou přihlášení dostat po přihlášení po kliknutí v menu na možnost „Soubory“. Na té mohou vidět název souboru a jméno uživatele, který soubor



Obrázek 14 Stránka se soubory pro stahování

přidal. Úložiště dokumentů se hodí např. pro ukládání organizačních záležitostí, usnesení apod.

3.5.9 Správa uživatelů

Po registraci má uživatel status tzv. základního uživatele. To znamená, že nemůže přidávat články na web nebo spravovat ostatní uživatele. Aby toho byl schopen, musí obdržet uživatelská práva od administrátora. Administrátor může uživateli nastavit dva typy práv – ověřený uživatel a administrátor. Ověřený uživatel může přidávat články na web a administrátor spravovat ostatní uživatele. Tohle nastavuje administrátor na příslušné stránce, kde vidí přehled uživatelů, údaje o nich a jejich práva.

4 Výsledky a diskuse

Bakalářská práce měla za cíl vytvořit webovou aplikaci pro členy webového spolku, kde si včelaři mohou zaznamenávat své informace o včelstvech a zároveň komunikovat mezi sebou.

V teoretické části práce byl uveden základní úvod celé problematiky. Dále zde došlo k popsání technologií, které se budou v práci vyskytovat, a to jak z pohledu front-endu, jmenovitě HTML, CSS a JavaScript tak i z pohledu back-endu, jmenovitě dotazovací jazyk SQL, prostředí ASP.NET ve spojení s návrhovým vzorem MVC a samotný databázový server (SRŘBD), ke kterému se přistupuje pomocí phpmyadmin klienta. Došlo k porovnání různých obdobných aplikací pro potřeby včelařů, a to jak aplikace pro mobilní telefony, tak i desktopového programu a webových aplikací.

V praktické části došlo nejprve k definování vzorových uživatelů v podobě person. Následně na základě jejich požadavků došlo k vytvoření drátěného modelu (wireframe) v programu Figma. Po vytvoření této hrubé verze uživatelského rozhraní byla celá problematika aplikace řešena v prostředí databáze, kde bylo vytvořeno několik tabulek tak, aby splňovaly pravidla pro normalizaci databáze. Následně po jejím vytvoření došlo k vytvoření dvou kontrolérů – jeden pro přihlášené uživatele, druhý pro nepřihlášené, v rámci kterých už vznikaly jednotlivé stránky v podobě pohledů (View). Pro spojení s databází vznikla třída context umožňující přenos dat mezi aplikací a databází. V pohledech (stránkách) byl použit programovací jazyk JavaScript pro zobrazování a validaci vstupních dat, a pro zobrazování grafu pomocí API CanvasJS.

Uživatelé tak mohou komunikovat v chatu, evidovat včely, zobrazovat si v grafu informace, včetně exportu do souborů CSV a JSON, a v neposlední řadě ukládat soubory pro potřeby členů spolku.

práci nebyl použit žádný CSS framework (např. Bootstrap, Tailwind). Styl webové aplikace – barvy, font a velikost ohraničení jednotlivých prvků byl vytvořen autorem práce

5 Závěr

Z pohledu autora práce má tahle práce v některých pohledech uspět v konkurenci ostatních podobě zaměřených aplikací. Český svaz včelařů sice v poslední době pokročil v oblasti digitalizace agendy v podobě Centrálního informačního systému pro vedení spolků a v Portálech včelaře pro běžné členy, problém nastává v případě, kdy se nějaký spolek rozhodne stát na Svazu nezávislým. V takovém případě nemá přístup do žádného z dvou výše uvedených systémů. Právě na tyto případy se snažila tahle aplikace reagovat a v některých částech inovovat některé prvky oproti systémům ČSV.

Aplikace má podle autora ambice k dalšímu rozvíjení – jmenovitě více možností, co zobrazit v grafech, rozšířit možnosti exportu do dalších formátů, implementace AJAXu pro odstranění nutnosti aktualizovat ručně stránku, či rozšířit možnosti poznámek u jednotlivých včelstev.

6 Seznam použitých zdrojů

1. KNÖDLOVÁ, Marie. Tak šel čas: Sborník k 100. výročí založení místeckého včelařského spolku. Frýdek-Místek: Český svaz včelařů, 2011. ISBN 978-80-260-0244-4
2. Český svaz včelařů, o. s., Včelařství v Česku. Praha 2012
3. Vlastimil DLAB. Historie včelařství. Z historie včelařství v Čechách, na Moravě a ve Slezsku. Okresní organizace ČSV Jičín. [online]. 2016 [cit. 2023-10-28]. Dostupné z: <http://www.oocsvjicin.cz/odborne-clanky/historie-vcelarstvi?showall=1>
4. 4. JARMILA, Machová. Historie buď je, a nebo není. Český svaz včelařů slaví 150 let od založení. Online. MAFRA. 2022. Dostupné z: https://www.lidovky.cz/relax/zajimavosti/vcely-cesky-vcelarsky-svaz-historie-uly-vcelari.A220519_140957_in-zajimavosti_ape. [cit. 2023-10-28].
5. ČESKÝ SVAZ VČELAŘŮ. Český svaz včelařů - informace. Online. C2010-2023. Dostupné z: <https://www.vcelarstvi.cz/cesky-svaz-vcelaru-informace/>. [cit. 2023-10-28].
6. ADOBE. Co to jsou webové aplikace a dynamické webové stránky? Online. C2019. Dostupné z: <https://helpx.adobe.com/cz/dreamweaver/using/web-applications.html>. [cit. 2023-10-28].
7. KOŘOUSKOVÁ, Barbora. CO JSOU PROGRESIVNÍ WEBOVÉ APLIKACE (PWA) A JAKÉ MAJÍ VÝHODY. Online. 2023. Dostupné z: <https://www.rascasone.com/cs/blog/progresivni-webova-aplikace-vyhody>. [cit. 2023-10-28].
8. VIRIUS, Miroslav. Programování v C#: od základů k profesionálnímu použití. Myslíme v... Grada Publishing, 2021. ISBN 978-80-271-1216-6.
9. MICROSOFT. Built-in types - C# reference - C# | Microsoft Learn. Online. 2023. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/built-in-types/built-in-types>. [cit. 2023-10-28].
10. MICROSOFT. .NET | Build. Test. Deploy. Online. C2023. Dostupné z: <https://dotnet.microsoft.com/en-us/>. [cit. 2023-10-28].
11. MICROSOFT. ASP.NET | Open-source web framework for .NET. Online. C2023. Dostupné z: <https://dotnet.microsoft.com/en-us/apps/aspnet>. [cit. 2023-10-28].
12. ANDERSON, Rick; BROCK, Dave a LARKIN, Kirk. Introduction to Razor Pages in ASP.NET Core. Online. 2023. Dostupné z: <https://learn.microsoft.com/en->

- [us/aspnet/core/razor-pages/?view=aspnetcore-7.0&tabs=visual-studio](https://aspnet/core/razor-pages/?view=aspnetcore-7.0&tabs=visual-studio). [cit. 2023-10-28].
13. ČÁPKA HARTINGER, David. MVC architektura. Online. C2023. Dostupné z: <https://www.itnetwork.cz/navrh/mvc-architektura-navrhovy-vzor>. [cit. 2023-10-28].
 14. MICROSOFT. Entity Framework documentation hub | Microsoft Learn. Online. C2023. Dostupné z: <https://learn.microsoft.com/en-us/ef/>. [cit. 2023-10-28].
 15. SOLID IT GMBH. DB-Engines Ranking - popularity ranking of database management systems. Online. C2023. Dostupné z: <https://db-engines.com/en/ranking>. [cit. 2023-10-28].
 16. ORACLE. What Is MySQL | Oracle. Online. C2023. Dostupné z: <https://www.oracle.com/mysql/what-is-mysql/>. [cit. 2023-10-28].
 17. VOSTROVSKÝ, Václav. Vytváření databází v ORACLE. Praha: Provozně ekonomická fakulta České zemědělské univerzity v Praze, c2004. ISBN 978-80-213-1191-6.
 18. THE PHPMYADMIN PROJECT. PhpMyAdmin. Online. C2003-2023. Dostupné z: <https://www.phpmyadmin.net/>. [cit. 2023-10-28].
 19. APACHE FRIENDS. XAMPP Installers and Downloads for Apache Friends. Online. C2023. Dostupné z: <https://www.apachefriends.org/>. [cit. 2023-10-28].
 20. CASTRO, Elizabeth a HYSLOP, Bruce. HTML5 a CSS3: Názorný průvodce tvorbou WWW stránek. 2. vyd. Albatros Media, 2022. ISBN 978-80-251-5045-0.
 21. ČÁPKA HARTINGER, David. Lekce 1 - Úvod do JavaScriptu. Online. C2023. Dostupné z: <https://www.itnetwork.cz/javascript/zaklady/javascript-tutorial-uvod-do-javascriptu-nepochopeny-jazyk>. [cit. 2023-10-28].
 22. ČESKÝ SVAZ VČELAŘŮ. CIS - Český svaz včelařů. Online. Dostupné z: <https://cis.vcelarstvi.cz/>. [obsah dostupný po přihlášení] [cit. 2023-10-28].
 23. ČESKÝ SVAZ VČELAŘŮ. Manuál k systému CIS Českého svazu včelařů | Manual systému CIS a PV - ČSV. Online. C2023. Dostupné z: <https://manual.vcelarstvi.cz/CIS-centralni-informacni-system-c90>. [cit. 2023-10-28].
 24. ČESKÝ SVAZ VČELAŘŮ. Portál včelaře. Online. Dostupné z: <https://portalvcelarstvi.cz/login>. [obsah dostupný po přihlášení] [cit. 2023-10-28].
 25. PETR JANDA - PJ SOFTWARE. SVčela - PJ Software. Online. C2023. Dostupné z: <https://pjsoftware.cz/index.php/svcela/>. [cit. 2023-10-28].

26. PETR JANDA – PJ SOFTWARE. sVčela [software]. [cit. 2023-10-28]. Dostupné z: https://pjsoftware.cz/loadfile/setup_svcela.exe. Požadavky na systém: Win XP, Win Vista, Win 7, Win 8, Win 8.1, Win 10, Win 11; velikost 5,17 MB.
27. PETR ZAORAL - PUNK HIVE. Chytrý včelař [software]. [cit. 2023-10-28]. Dostupné z <https://www.chytry-vcelar.cz/>. Požadavky na systém: Android, iOS; velikost 30,8 MB.
28. PAVLÍČEK, Josef. The Cookbook for Interaction Design and Human Computer Interaction. Online. Dostupné z: https://docs.google.com/presentation/d/1nbLjgEX5mS6kl_cRx6CeKuhd-fzzkyYn_j03vMLkH4/edit#slide=id.g9a94b8846e_0_7. [cit. 2024-02-02].
29. ARLOW, Jim a Ila NEUSTADT. UML 2 a unifikovaný proces vývoje aplikací: Objektově orientovaná analýza a návrh prakticky. 2., aktualizované a doplněné. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.

Seznam obrázků, tabulek, grafů a zkratek

6.1 Seznam obrázků

Obrázek 1 - Diagram případu užití pro webovou aplikaci včelařského spolku. [30]	35
Obrázek 2 - Drátěný model úvodní stránky.....	37
Obrázek 3 - Drátěný model přihlašovací stránky (Log-in).....	38
Obrázek 4 - Drátěný model úvodní stránky přihlášeného uživatele.....	39
Obrázek 5 - Drátěný model stránky pro správu stanoviště.....	41
Obrázek 6 Drátěný model přehledu úlu a včelstva v něm.	43
Obrázek 7 Drátěný model diskusního chatu.	44
Obrázek 8 Diagram databáze.....	46
Obrázek 9 Metoda zajišťující propojení aplikace s databází.....	48
Obrázek 10 Ukázka chatu.....	50
Obrázek 11 Ukázka kódu pro přidání získaného medu ze včelstva.....	52
Obrázek 12 Princip fungování grafu na straně serveru.....	53
Obrázek 13 Ukázka kódu zajišťujícího export do souborů.....	54
Obrázek 14 Stránka se soubory pro stahování.....	55

6.2 Seznam použitých zkratek

ČSV – Český svaz včelařů

ZO – Základní organizace

PWA – Progressive Web App

CIL - Common Intermediate Language

CLR - Common Language Runtime

CTS - Common Type System

JiT – Just in Time

Int – Integer

Bool – Boolean

Char - Character

MVC – Model-View-Controller

SQL – Structured Query Language

CRUD – Create-Read-Update-Delete

DDL – Data Definition Language (Příkazy pro definici dat)

DML – Data Manipulation Language (Příkazy pro manipulaci s daty)

DCL – Data Controll Language (Příkazy pro kontrolu dat)

TCL – Transact Controll Language (Příkazy pro kontrolu transakcí)

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

CIS – Centrální informační systém

Přílohy

PŘÍLOHA A Archiv s webovou aplikací

PŘÍLOHA B SQL soubor databáze

PŘÍLOHA C Výstupní soubor z prostředí programu FIGMA.