

Aplikace pro podporu malovinařů

Radek Havelka

Bakalářská práce
2022



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Radek Havelka**
Osobní číslo: **A19031**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Softwarové inženýrství**
Forma studia: **Prezenční**
Téma práce: **Aplikace pro podporu malovinařů**
Téma práce anglicky: **Application to Support Small Wine Producers**

Zásady pro vypracování

1. Analyzujte problematiku a vypracujte literární rešerši na dané téma.
2. Proveďte analýzu požadavků pro podporu malovinařů.
3. Popište technologie, které budou v práci použité.
4. Navrhněte a realizujte databázi a aplikaci.
5. Proveďte otestování aplikace a zajistěte zabezpečení aplikace.
6. Zhodnotte provedené řešení a navrhněte případné vylepšení.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. MALL, Rajib. *Fundamentals of software engineering*. PHI Learning Pvt. Ltd., 2018. ISBN 978-93-88028-03-5.
2. RAJLICH, Vaclav. *Software engineering: The current practice*. Chapman and Hall/CRC, 2019. ISBN 978-1-4665-1035-7.
3. DATE, Chris J. *Database design and relational theory: normal forms and all that jazz*. Apress, 2019. ISBN 978-1-4842-5540-7.
4. DAVIDSON, Louis a Jessica M. MOSS. *Pro SQL Server relational database design and implementation* [online]. 2016. United States: APress, 2016 [cit. 2021-12-01]. ISBN 978-1-4842-19737.
5. FREEMAN, Adam. *Pro ASP.NET Core 3 : Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages* [online]. 8th ed. 2020. Berkeley, CA: APress, 2020 [cit. 2021-12-01]. ISBN 978-1-484254400.

Vedoucí bakalářské práce: **doc. Ing. Zdenka Prokopová, CSc.**
Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce: **3. prosince 2021**

Termín odevzdání bakalářské práce: **23. května 2022**



doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan

prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 24. ledna 2022

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 16. 5. 2022

Radek Havelka v. r.
podpis studenta

ABSTRAKT

Tato bakalářská práce se zabývá návrhem a implementací webové aplikace pro podporu malovinařů. Teoretická část se věnuje již dostupným řešením na trhu, popisuje jejich výhody a nevýhody, jsou popsány technologie, které byly pro vývoj použity a jsou zde rozebrány nejčastější útoky na webové aplikace, závěrem je sběr požadavků. Praktická část se věnuje návrhem a vývojem systému. Aplikace je vytvořena pomocí frameworku ASP.NET Core s využitím architektonického vzoru MVC a Entity Framework Core. Je zde využita databáze MySQL Server.

Klíčová slova: Aplikace pro podporu malovinařů, Webová aplikace, .NET 5, C#, MVC, Entity Framework Core

ABSTRACT

This bachelor thesis deals with the design and implementation of a web application to support small-wine producers. The theoretical part is devoted to the solutions already available on the market, describes their advantages and disadvantages, the technologies that were used for the development are described and the most common attacks on web applications are discussed, concluding with the collection of requirements. The practical part deals with the design and development of the system. The application is developed using the ASP.NET Core framework using the MVC architectural pattern and the Entity Framework Core. MySQL Server database is used.

Keywords: Application to support small wine producers, web application, .NET 5, C#, MVC, Entity Framework Core

Chtěl bych poděkovat své vedoucí práce paní doc. Ing. Zdence Prokopové, CSc. za odborné vedení práce a za všechny rady a konzultace při vypracovávání této bakalářské práce.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 MALOVINAŘSTVÍ	10
2 DOSTUPNÉ ŘEŠENÍ	12
2.1 AGROWIN	12
2.2 VINEINFO	14
2.3 VINTRACE	15
2.4 EVINEYARD.....	16
2.5 PROCESS2WINE.....	17
2.6 ZHODNOCENÍ DOSTUPNÝCH ŘEŠENÍ.....	18
3 SBĚR A ANALÝZA POŽADAVKŮ	19
3.1 FUNKČNÍ POŽADAVKY	19
3.2 NEFUNKČNÍ POŽADAVKY	20
4 ÚTOKY NA WEBOVÉ APLIKACE	21
4.1 SQL INJECTION	21
4.2 CROSS-SITE SCRIPTING ÚTOKY – XSS.....	22
4.3 CROSS-SITE REQUEST FORGERY ÚTOKY – CSRF	22
5 POUŽITÉ TECHNOLOGIE	24
5.1 ASP.NET CORE	24
5.2 RAZOR PAGES	24
5.3 MYSQL WORKBENCH	24
5.4 ENTITY FRAMEWORK CORE	25
5.5 CSS A BOOTSTRAP	25
5.6 JAVASCRIPT	26
II PRAKTICKÁ ČÁST	27
6 NÁVRH APLIKACE	28
6.1 DIAGRAM PŘÍPADŮ UŽITÍ.....	28
6.2 SCÉNÁŘE	29
6.3 ERD A DATABÁZOVÝ MODEL	40
6.3.1 ER Diagram.....	40
6.3.2 Databázový model.....	41
6.4 DRÁTĚNÉ MODELY	41
7 IMPLEMENTACE	46
7.1 PROPOJENÍ DATABÁZE A KÓDU	46
7.2 AUTORIZACE A AUTENTIFIKACE	48
7.3 FUNKCE.....	49
7.4 ZABEZPEČENÍ PROTI ÚTOKŮM	50
7.4.1 SQL Injection	50
7.4.2 CSRF útok.....	51
8 PRŮVODCE APLIKACÍ	52

8.1	REGISTRACE A PŘIHLÁŠENÍ	52
8.2	NAVIGAČNÍ LIŠTA A ROLE	53
8.3	HLAVNÍ STRÁNKA	54
8.4	PRACOVNÍ ÚKONY	55
8.5	SKLIZEŇ	56
8.6	POČASÍ	56
8.7	SPRÁVA APLIKACE	57
8.8	ZAMĚSTNANCI A PANEL UŽIVATELE	57
9	TESTOVÁNÍ	59
10	ZHODNOCENÍ A PŘÍPADNÉ VYLEPŠENÍ.....	62
	ZÁVĚR	63
	SEZNAM POUŽITÉ LITERATURY.....	65
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	68
	SEZNAM OBRÁZKŮ	69
	SEZNAM TABULEK.....	70
	SEZNAM PŘÍLOH.....	71

ÚVOD

V dnešní době je popularita vína po celém světě daleko větší, než tomu bylo před pár lety, s tím je úzce spjatá vyšší kvalita vín a používané technologie pro výrobu. V tomhle ohledu je potřeba brát daleko větší zřetel na administrativní část a správu celého chodu vinařství. Pěstování vinné révy a posléze výroba je nejvíce populární na jižní Moravě, kde sídlí největší firmy v tomhle oboru, ale mezi nejpočetnější skupinu výrobců a pěstitelů vinné révy patří tzv. malovinaři. Ti vyrábí víno spíše pro vlastní potřebu, ale mnozí z nich po čase přechází k výrobě vína pro obchodní využití.

Cílem teoretické části bakalářské práce bude analyzovat požadavky pro malovinaře a usnadnit jim správu svého vinařství. Po sběru a následné analýze budu porovnávat již dostupné řešení a následně je vyhodnocovat. V práci uvedu a popíšu nejrozšířenější typy kybernetických útoků, které jsou spjaté s webovými aplikacemi. V poslední části rozeberu použité technologie pro vytvoření webové aplikace.

Praktická část se bude skládat z návrhu a implementace výsledné aplikace, která vychází ze sběru a analýzy požadavků. Návrh aplikace se skládá z vytvoření případů užití a popsaní hlavních a alternativních scénářů. Následuje vytvoření entitně relační diagram, kde jsou vytvořené jednotlivé třídy, které budou implementovány a vztahy mezi nimi. V druhé části návrhu byl vytvořen databázový model, který vychází z entitně relačního diagramu. Závěrem návrhu bude vytvoření drátěných modelů, ze kterých budu po grafické stránce vycházet při následné implementaci. Po provedení implementace bude následovat popsání systému a jeho otestování. Závěrem práce bude zhodnocení vytvořeného řešení a navrnutí případných vylepšení.

I. TEORETICKÁ ČÁST

1 MALOVINAŘSTVÍ

Mezi hlavní kritéria pro výrobu vína je samostatná kvalita hroznů, která se odvíjí od několika faktorů jako je kvalita půdy, podnebí, ošetřování samotné vinice a také pečlivě naplánovaná sklizeň hroznů. Během sklizně a při přepravě hroznů k dalšímu zpracování může být kvalita finálního produktu negativně ovlivněna několika faktory. Mezi ně patří houbové choroby révy vinné, kvasinky a bakterie. U větších podniků se tyto problémy dají minimalizovat, ale u malovinařů je preferovaná metoda ruční sklizně, která sice přináší úsporu nákladů a lepší selekci přímo při sklizni, ale na druhé straně jsou hrozny vystaveny daleko horším podmínkám, které mohou napomáhat k tvorbě již zmíněných negativních jevů.

Jako další v pořadí je technologie a proces zpracování vína, kde se výhradně dbá na udržování čistoty a základních hygienických pravidel ve vinařském sklepě. Technologické vybavení sklepa zahrnuje mlýnky, mlýnkoodzrňovače, lisy na hrozny, nádoby na víno, čerpadla, hadice apod [4].

Nesmíme zapomenout na samotný postup výroby vína, který se skládá z:

- Odzrnění – je to proces kdy se po sklizni, nejlépe ihned odstraní třapina od bobule, je důležité, aby se oddělení provádělo šetrně, aby se nepoškodila pecka, která pak může způsobit hořkost, takto oddělená bobule se nazývá „rmut“
- Lisování – proces při kterém se oddělí mošt a vylisované slupky, po této fázi se mošt odkaluje, aby se odebrali nežádoucí látky a u některých typu vín může následovat zvýšení cukernatosti
- Kvašení – další fázi je proces kvašení, který začne sám od sebe za přítomnosti kvasinek, které na hroznech už vznikají nebo se použijí speciální selektované kmeny kvasinek, tento proces se laicky dá vysvětlit jako přeměna cukru na alkohol za vzniku kyslíčnicku uhličitého a tepla, dále se dělí podle typu vína, kdy u bílého vína se kontroluje teplota kvasícího moštu, která by neměla překročit cca. 20° C, tento mošt se nazývá „burčák“. U červeného vína se po kvašení ještě provádí jablečno-mléčna fermentace, jak už z názvu vyplývá zjednodušeně probíhá enzymatická přeměna kyseliny jablečné na jemnější a hladší kyselinu mléčnou, tento proces zlepšuje celkovou chuť a plnost vína
- Školení vína – poslední a také nejtěžší fáze celého koloběhu, provádí se hned několik operací jako je přidání oxidu siřičitého, který působí jako konzervant, oddělení vína od usazených kvasnic a filtrace

Jak už bylo zmíněno rozdíly ve výrobě a zpracování vína jsou podle odrůdy, kde [5]:

- Bílé víno, které je v České republice nejrozšířenější ze všech, se po odzrnění nechává volně lisovat, aby vznikl mošt, který se odkalí, poté dochází ke kvašení
- Červené víno, které je v České republice zastoupením zhruba 25 %, podléhá fázi odzrnění, kdy dochází k rozdrčení a hrubému pomletí bobulí, aby se porušila slupka hroznů, která dodá potřebné barvivo, pak se víno nechává několik dní kvasit, dalším krokem je lisování, kde rozdíl oproti bílému vínu je, že u bílého se lisuje mošt, u červeného mladé víno
- Růžové víno má stejný proces výroby, jak víno bílé s tím rozdílem, že se používají vybrané odrůdy vinné révy

2 DOSTUPNÉ ŘEŠENÍ

Na trhu existuje nespočet řešení pro danou problematiku z dostupných řešení bylo vybranych 5 řešení, které jsou podrobně zhodnocené a analyzované:

- AgroWin software pro zemědělce a vinaře
- VineInfo
- Vintrace
- eVineyard
- Process2Wine

Při hodnocení byl dáván důraz na použitelnost a univerzálnost systému a také celkovou cenu, která by měla být co nejpříjemnější.

2.1 AgroWin

System AgroWin [6] je rozdělen na dva podsystemy Sklepmistr a Software pro evidenci spotřeby hnojiv a ochranných přípravků. První zmíněný vznikl sloučením dvou systémů, softwaru Vinařské evidence a Gastrowin, kombinací vzniknul víceúčelový systém, jak pro správu samotného vinařství, tak i pro maloobchodní prodej.

Mezi hlavní moduly náleží vinařská evidence, která je určena pro všechny vinařské firmy, zahrnuje informace o výrobě a zpracování vína, přehled o pracovních operacích a zásazích ve vinohradu, přehled výkupních lístků pro pozdější dohledatelnost a evidenční knihu pro evidenci sklizně, prodeje, zásob apod. Mezi výhody řadím velkou škálu možností a komplexnost na úkor přehlednosti aplikace. Modul kasa je určen pro maloobchodní prodej, do výhod tohoto modulu řadím použitelnost jak na dotykových zařízeních, tak pro klávesnici a myš, je zde i možnost přímého propojení s platebním terminálem.

Další moduly jsou:

- Skladové hospodářství – obsahuje přehled o zásobách skladů, kde pro hledání můžeme specifikovat údaje jako je cenová hladina, uživatelské kategorie apod.
- Vydané faktury – tato část je spjatá se skladovým hospodářstvím, kde můžeme skladové zásoby odepisovat přímo při fakturaci
- Potisk štítků – umožňuje vyrobit identifikační štítky do kterých lze zahrnout čárové kódy typu EAN13 a QR kód

- Přístupová práva – tento modul zajišťuje přidělování pracovních úkonů a povolení podle typu pracovního zařazení např.: administrátor, účetní, provozní, management atd.

Na výběr máme tři verze produktů Mini, Standard, Profi. Mini verze je určená pro zpracovávanou výměru do 5 hektarů a produkci do 50 000 litrů. Standard verze je do 50 hektarů a 250 000 litrů a Profi verze je neomezená. S jednotlivými verzemi systému se rozšiřuje sada modulů. Pro přístup k systému Vinařská evidence je potřeba zaplatit poplatek uvedený v tabulce 1. Pro přístup k systému Vinařský sešit je potřeba zaplatit poplatek uvedený v tabulce 2.

Tabulka 1. Poplatky za systém AgroWin

<i>Typ licence</i>	<i>Základní cena</i>	<i>Roční aktualizace</i>
Vinařská evidence 2021 MINI	4 100,- Kč	2 200,- Kč
Vinařská evidence 2021 STANDARD	12 500,- Kč	5 500,- Kč
Vinařská evidence 2021 PROFESIONAL	18 850,- Kč	7 900,- Kč

Tabulka 2. Poplatky za systém AgroWin

<i>Typ licence</i>	<i>Základní cena</i>	<i>Roční aktualizace</i>
Vinařský sešit 2021 MINI	2 900,- Kč	1 500,- Kč
Vinařský sešit 2021 STANDARD	7 900,- Kč	3 940,- Kč
Vinařský sešit 2021 PROFESIONAL	11 850,- Kč	6 200,- Kč

Hlavní výhody při použití systému vidím ve velké komplexnosti, propojení s ostatními viničemi za pomoci evidenčního listu vinic a v samotném použití, kde je využitelnost v plánování jednotlivých fází výroby, případný prodej nebo nákup materiálu od jiných vinic, řízení skladu. Při prodeji je možnost využití zabudovaného terminálu, který podporuje EET, v případě výpadku internetového spojení lze pracovat v režimu offline po dobu 48 hodin, po uplynutí lhůty jsou doklady hromadně odeslaný na EET portál. Zajímavou částí je i evidence směn zaměstnanců a plánování. Mezi nevýhody řadím vyšší základní cenu, která je pro potřeby malovinařů vysoká, na druhou stranu při výběru základního systému se dostaneme na

částku 4 200,- Kč. Jelikož je systém komplexní jako další nevýhodu bych zařadil přehlednost a větší nároky pro samotné uživatele.

2.2 VineInfo

VineInfo [7] je internetový systém který pracuje v reálném čase a poskytuje efektivní nástroj pro správu a přehled všech důležitých informací o vinohradu. Systém je provozován přes internetový prohlížeč, a proto není třeba investovat do nového hardwaru nebo softwaru, stačí internetové připojení a počítač. Jelikož je to aplikace třetí strany, tak data jsou umístěna na server, který je umístěn ve specializovaném zařízení s nepřetržitým zabezpečením a denním zálohováním. Je zde možnost školení pro nové uživatele a emailová podpora. Systém nabízí i online komunikaci s ostatními vinaři pro zodpovězení složitějších dotazů.

Hlavní funkce tohoto systému je správa a plánování zavlažování, pracovní příkazy k hnojivům, krycím plodinám a půdním doplňkům a také obsahuje dostupné informace o předpovědi úrody. Všechny informace se elektronicky ukládají pro další roky a je možnost je stáhnout do Excel formátu nebo Adobe PDF, vytisknout a poslat e-mailem. Poplatky se odvíjí od velikosti vinohradu a od potřeby hodin k zaškolení. Ceny jsou uvedené v tabulce níže.

Tabulka 3. Poplatky za systém VineInfo

<i>Velikost vinohradu</i>	<i>Základní cena</i>	<i>Roční aktualizace</i>
Do 8 hektaru	400 \$	168 \$
Do 16 hektaru	600 \$	108 \$
Do 40 hektaru	800 \$	72 \$
Do 80 hektaru	1 200 \$	48 \$
Do 160 hektaru	1 500 \$	36 \$
Do 325 hektaru	1 800 \$	27 \$
Do 647 hektaru	2 000 \$	18 \$

Mezi výhody toho systému řadím:

- možnost konzultace s ostatními vinaři
- plánování budoucích prací na vinici
- shromažďování informací z více zdrojů na jednom místě
- informace o počasí v reálném času z několika stanic zdarma
- velká cenová škálovatelnost

Nevýhody:

- nemožnost přidávat vlastní poznámky k plánovaným akcím
- nemožnost úpravy prostředí a celkový zastaralý vzhled
- pouze v anglickém jazyce
- systém je spíše uzpůsoben na informační bázi než na reálné použitelnosti do prostředí

2.3 Vintrace

Vintrace [8] je cloudový systém od společnosti VINX2 WINERY SOFTWARE, INC, která je dceřinou společností Vintrace Holdings Pty Ltd vyvíjený od roku 2007. Systém je využíván po celém světě, ale největší rozmach má v Spojených státech amerických. Jelikož je využíváné cloudové úložiště, je možné aplikaci používat jak na desktopových zařízeních, tak i na mobilních telefon. Samotná firma nabízí bezplatnou klientskou službu, kde můžeme zjistit více informací o aplikaci, případně při zájmu pro používání je možnost celý systém poupravit a nakonfigurovat.

Systém nabízí správu vinohradu, kde je možnost zapisovat si poznámky k jednotlivým odrůdám a poté je analyzovat pro určení nejvhodnějšího termínu pro sklizeň, správu laboratoře a vinného sklepa, kde je možné vidět jaké akce jsou potřeba vykonat nebo jaké právě probíhají, samotná aplikace je vybavená QR kód čtečkou, která nám umožňuje okamžitě vyhledat informace pro právě zvolený sud. Mezi další možnosti tohoto systému spadá kontrola skladu, tento modul zjednodušuje nákup a prodej surovin potřebných pro výrobu a údržbu vinného sklepa. Aplikace nabízí i zpravodajskou službu, kde je možnost komunikaci s jinými společnostmi, případně k vytvoření prezentace o cenách, složení, expozici v sudech, laboratorních pracích a degustačních poznámkách. Společnost nabízí 5 smluvních licencí, které se odvíjí od produkce vína za rok. Nejvyšší Enterprise licence zahrnuje prioritní podporu včetně helpdesku, která funguje 24 hodin 7 dní v týdnu a podporu vývojářů pro přístup k otevřenému API. Nejnižší licence je nabízená pro malé vinařství a zahrnuje zkušební verze zdarma.

Reálné ceny licencí nejsou uvedeny. Mezi zajímavost náleží případné připojení k software development teamu na vícero pozic jako je analytik, systémový inženýr a zástupce zákaznické podpory.

Výhody:

- bezplatná zkušební licence
- zákaznická podpora
- mobilní aplikace
- možnost použití po celém světě
- QR čtečka pro rychlejší vyhledávání záznamů

Nevýhody:

- data jsou uložena na cloudu – možná ztráta kontroly nad citlivými údaji
- není možnost přiřazovat směny

2.4 eVineyard

eVineyard [9] je systém pro správu vinařství s rozšířením po celém světě, podporuje jak ty nejmenší podniky od 1 hektaru až po ty největší. Toto řešení spravuje a vyvíjí společnost ELMIBIT, d.o.o, která se zabývá návrhy počítačových systémů a souvisejících služeb. Tato služba splňuje požadavky pro správu celého vinařství jako řešení výše. Na webových stránkách je možné dohledat měsíční poplatky pro používání, které jsou rozdělené do 4 kategorií. Měsíční poplatky se účtují za velikost vinohradu v hektarech. Nejnižší verze LITE nabízí měsíční zkušební dobu zdarma. Pro jednodušší porovnání cen byla vytvořena tabulka číslo 4. V cenách není uvedena cena se zahraniční DPH.

Tabulka 4. Poplatky za systém eVineyard

<i>Typ licence</i>	<i>Velikost vinohradu</i>	<i>Základní cena za měsíc</i>
LITE	Není uvedeno	19 \$
PROFESSIONAL	Do 40 hektaru	54 \$
ESTATE	Do 500 hektaru	575 \$
ENTERPRISE	Nad 500 hektaru	Není uvedeno

Výhody:

- GPS tracker pro jednotlivé práce
- jednoduchý tisk dokumentů přímo do PDF
- automatická analýza dat
- přívětivé moderní prostředí aplikace

Nevýhody:

- podpora není tak flexibilní a rychlá
- nemožnost úpravy prostředí na míru
- nemožnost používat systém na mobilních zařízeních

Mezi zajímavou část toho systému řadím automatickou analýzu dat za pomoci senzorů umístěných jak na samotné vinici, tak externě v okolí jako například meteorologické stanice. Data se porovnají, najdou průměry a rozhodne o nejlepších postupech na základě pozorování, jak mikroklima v určitém bloku vinice ovlivňuje sezónní růst hroznů. Funkce nabízí i přesné údaje o tom kolik závlahy by mělo být na bloku vinice aplikováno na základě počasí. Další funkce jsou modely vývoje škůdců a chorob, analýza a plánování geograficky náročných činností a podpora satelitního pozorování země a snímkování z dronů. Data jsou vyhodnoceny graficky a systém automaticky navrhne nejlepší řešení pro nejlepší kvalitu hroznů a celkových výnosů.

2.5 Process2Wine

Společnost Ertus Group nabízí cloudový systém pro správu vinohradu a vinných sklepů se jménem Process2Wine [10], firma vznikla sloučením dvou firem ERTUS Consulting, která se specializovala na vinařskou poradnu a D2E SSII, která vyvíjela podnikové aplikace. K dnešnímu dni aplikaci využívá více jak 950 klientů. Systém je rozdělen na dvě části, které je možné kombinovat, první je správa vinohradu a druhá je správa vinného sklepa. Mezi základní funkcionalitu spadá sledování pracovníků na pracovišti a odpracované hodiny, udržování časového harmonogramu pro zásahy ve vinohradu nebo sklepech, detailní zprávy o právě vykonávaných pracích atd. Řešení je distribuováno ve vícero variantách, podle toho, jak je velký vinohrad a kolik funkcí chceme využívat. Základní verze aplikace nabízí společnost po registraci zcela zdarma, tohle řešení pokrývá řízení operací na vinici pomocí mapovacího nástroje, správu všech denních příkazů a kompletní historii každého bloku vinice

v reálném čase. Při potřebě využívat další funkce se dá aplikace lehce aktualizovat na lepší verzi, která se pak ale stane placená. Reálné ceny nejsou na webových stránkách uvedeny.

Výhody:

- možnost využívat jen konkrétní část systému
- aplikace je dostupná na mobilních zařízeních
- bezplatná verze pro základní chod vinice
- graficky přívětivé prostředí

Nevýhody:

- pouze v anglickém jazyce
- data jsou uložena na cloudu – možná ztráta kontroly nad citlivými údaji

2.6 Zhodnocení dostupných řešení

Po prozkoumání jednotlivých dostupných řešení, porovnání výhod a nevýhod systému a reálné uplatitelnosti pro naše potřeby bylo vyhodnoceno stanovisko. Každý systém si přináší sebou určité výhody a nevýhody. Základní funkcionalita, která byla nastavená splňují všechny řešení kromě systému VineInfo, který slouží spíše jako informační nástroj nežli nástroj pro správu vinného sklepa a vinohradu. Další možnou nevýhodou systému je, že všechny řešení kromě systému AgroWin nemají plnou podporu českého jazyka. Z pohledu ceny nebylo možné vyhodnotit nejlepší řešení, protože některé ceny nebyly přímo uvedeny, ale vyzdvihnul bych systém Process2Wine, který nabízí pro ty nejmenší podniky svojí aplikaci se základní funkcionalitou zcela zdarma.

Některá z vybraných řešení uvádí, že je možné systém nakonfigurovat podle představ klienta, tak aby mohl naplnit všechny potřebné požadavky. Nicméně doba, za kterou by byl systém překonfigurován a nasazen do reálného provozu by mohla trvat i několik týdnů a zvýšili by se celkové náklady na údržbu systému. Mezi nejlepší řešení, co se týče flexibility již nabízených řešení byli zařazeny systémy Process2Wine a eVineyard, oba systémy nabízí velké množství funkcionalit, které je možné do aplikace implementovat, a naopak je možné některé prvky odstranit pro větší přehlednost.

3 SBĚR A ANALÝZA POŽADAVKŮ

Před návrhem a implementací samotné aplikace je potřeba nasbírat informace o tom jaké funkcionality má systém plnit. Pro sběr požadavků byly použité již existující řešení a dokumentace a rozhovory s lidmi v oboru. Analýza požadavků se řídí unifikovaným procesem vývoje aplikací [16].

3.1 Funkční požadavky

Funkční požadavky nám popisují to, co by měl systém umět dělat a vykonávat. Požadavky se mění v závislosti na získaných informacích a v každém projektu jsou rozdílné.

Zaměstnanec

- Zaměstnanec se bude moct přihlásit pomocí emailu a hesla, který bude vložen do systému po registraci
- Zaměstnanec se může podívat na právě vykonávané práce ve vinném sklepě nebo vinohradu
- Zaměstnanec může přidat záznam o sklizni
- Zaměstnanec se může podívat na evidenci skladu
- Zaměstnanec se může podívat na záznamy o kvalitě hroznů z minulých let
- Zaměstnanec si může vyfiltrovat záznamy na základě svých požadavků
- Zaměstnanec se může podívat na předpověď počasí
- Zaměstnanec může vytvořit komentář k pracovnímu úkonu nebo jej smazat

Manažer

- Manažer může vykonávat všechny operace jako zaměstnanec
- Manažer si může nechat poslat PDF záznam do mailu nebo si jí stáhnout do počítače
- Manažer může vytvořit nový sklep a vinohrad, nebo přidat odrůdy používané na vinohradě
- Manažer může vytvořit nový název pracovního úkonu a přiřadit jej k pracovnímu typu

Admin

- Admin si bude moct nechat vyjet seznam všech registrovaných uživatelů do systému
- Admin může vykonávat všechny operace jako zaměstnanec nebo manažer
- Admin bude moct upravovat uživatele v systému

3.2 Nefunkční požadavky

Nefunkční požadavky jsou požadavky, které nám určují vlastnosti softwaru, ve zkratce nám popisují to, jak by měl systém fungovat.

- Aplikace bude implementovaná jako webová aplikace využívající .NET Core framework
- Aplikace bude pro databázi využívat relační databázový systém Microsoft SQL Server
- Aplikace bude využívat architektonický vzor MVC pro lepší udržitelnost systému
- Aplikace bude využívat autorizaci a autentizaci pro zvýšení bezpečnosti
- Aplikace bude v českém jazyce
- Aplikace bude podporovat Full-HD rozlišení.
- Aplikace bude využívat Single-Page pro rychlejší načítání
- Aplikace bude mít podporu prohlížečů Mozilla Firefox, Google Chrome, Opera, Microsoft Edge

4 ÚTOKY NA WEBOVÉ APLIKACE

Množství hrozeb a typů kybernetických útoků v dnešní době rychle přibývá, může za to velký rozmach institucí, které využívají nebo přecházejí na elektronické weby, evidence, bankovníctví atd. Jelikož typů útoků je mnoho, tak byli vybrány ty nejznámější a také nejnebezpečnější útoky.

4.1 SQL Injection

Je typ útoků, který umožňuje útočnickovi zasahovat přímo do dotazů, které aplikace provádí do databáze. Pokud je útok úspěšný, útočnickovi je umožněno měnit logiku SQL dotazu a tím získat data ke kterým by za normálních okolností neměl přístup. Může se jednat o hesla a data patřícím jiným uživatelům nebo jakákoliv data ke kterým má aplikace přístup. V některých případech může útočník eskalací útoku shodit server na kterém aplikace běží nebo jinou back-end infrastrukturu [11].

Způsob vykonání útoku

Jak už bylo řečeno, SQL útok se provádí nad databází a nejčastěji za využití formulářů na webových stránkách. V případě, že obsah, který je zadán do formuláře přímo vytváří SQL příkaz bez další kontroly na zakázané znaky, může útočník přidat závadnou část kódu dovnitř SQL příkazu a tím získat přístup k informacím nad cílovou databází.

Příklad útoku na získání dat z jiných databázových tabulek:

```
SELECT jmeno, informace FROM produkty WHERE category = 'Hracky'
```

Za pomocí tzv. UNION znaku je možné do dotazu výše přidat další SELECT dotaz

```
' UNION SELECT jmeno, heslo from uzivatele—
```

Zavolání takového SQL příkazu bude mít za následek spuštění příkazu nad ostatními tabulkami v systému a získá jména a hesla uživatelů společně s jmény a informacemi o produktu.

Samozřejmě, že podobných způsobů, jak se dostat k informacím ke kterým nemáme mít přístup je nespočet. Za zmínku stojí, že mezi nejvyužívanější dotazy pro útok jsou příkazy SELECT a WHERE.

4.2 Cross-Site scripting útoky – XSS

Cross-Site scripting je typ webového útoku, kdy dochází k spuštění JavaScript kódu uvnitř nějakého HTML kódu. Na rozdíl od jiných útoků, je tento prováděn přímo na straně uživatele, nikoliv na straně aplikace na serveru. V minulosti tento typ útok byl bagatelizován, protože neumožňoval přímou kontrolu nad citlivými daty a údaji, ale kontrolu nad tím, co uživatel vidí, co provádí za akce apod. V dnešní době, ale tento typ útoku umožňuje útočníkovi vydávat se za oběť a tím provádět veškeré akce na straně klienta, které je uživatel schopen provést a dostat se tak k datům uživatele. Pokud má oběť v rámci aplikace privilegovaný přístup k datům, může být útočník schopen získat plnou kontrolu nad všemi funkcemi a daty. Mezi hlavní typy XSS útoků patří Reflected XSS, Stored XSS a DOM-based XSS, poslední jmenované si ukážeme na příkladu níže [12].

Příklad DOM-based cross-site scripting útoku:

```
var hledej = document.getElementById('hledej').value;

var vysledek = document.getElementById('vysledek');

vysledek.innerHTML = 'Hledal jsi: ' + hledej;
```

Za pomoci příkazu výše je možné za pomoci JavaScriptu číst hodnoty ze vstupního pole a zapisovat prvky do HTML kódu. Když může útočník přistupovat k hodnotě vstupního pole, tak může vytvořit škodlivou část kódu, která způsobí spuštění jeho vlastního skriptu.

```
Hledal jsi: <img src=1 onerror='/* Něco si hledal ... */>
```

Jelikož variací, jakým způsobem je možné vytvořit skripty je mnoho, je možné, že pokud je útok dobře propracovaný a naplánovaný, oběť si vůbec nemusí všimnout, že byla napadena.

4.3 Cross-Site request forgery útoky – CSRF

CSRF je typ útoku, kdy útočník přiměje oběť k otevření stránky se škodlivým kódem, za pomoci kterého získá oprávnění pro manipulaci s daty oběti. Útoky se zaměřují na funkce, které způsobují změnu stavu serveru, například změna e-mailové adresy nebo nákupu. Například přihlašovací CSRF, kdy útočník donutí oběť kliknout na stránku, kde se přihlásí a vyplní osobní údaje, jako třeba informace o kreditní kartě, ale netuší, že se přihlásil na účet, který má pod kontrolou útočník. Útočník se pak znovu přihlásí a získá potřebné informace [14].

Pro úspěch útoku musí být splnění tři podmínky. Akce, která bude provedená musí obsahovat nějaký HTTP požadavek, typicky GET požadavek a je založená na cookies pro autentizaci uživatele. Akce, při které dojde ke změně nějakých údajů, typicky změna e-mailu, hesla. Akce, která neobsahuje nepředvídatelné parametry požadavků, například při vyvolání změny hesla, kdy útočník potřebuje znát hodnotu stávajícího hesla [13].

Příklad Cross-Site Request Forgery útoku:

```
POST /email/change HTTP/1.1
Host: stranka-ktera-neni-zabezpecena.cz
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Cookie: session=tkfkgnnksPo4WekgNmMkrAsw
```

```
email=PavelNovak@seznam.cz
```

Jak si lze povšimnout, příklad splňuje všechny potřebné záležitosti, je vyvolaná změna e-mailu uživatele za pomoci POST požadavku, stránka využívá cookies pro autentizaci uživatele a útočník může jednoduše zjistit hodnoty parametrů požadavků, které jsou potřebné pro provedení útoku.

```
<html>
  <body>
    <form action="https://stranka-ktera-neni-zabezpecena.cz
      /email/change" method="POST">
      <input type="hidden" name="email" value="Utocnik@seznam.cz" />
    </form>
    <script>
      document.forms[0].submit();
    </script>
  </body>
</html>
```

Za pomoci html kódu, který útočník na závadné stránce implementuje, se dostane k uživatelskému účtu.

5 POUŽITÉ TECHNOLOGIE

5.1 ASP.NET Core

ASP.NET Core [1] je výkonná open source platforma pro vývoj moderních aplikací spojených s cloudovými aplikacemi. Mezi hlavní výhody patří, že od verze ASP.NET Core 3.x je platforma využitelná na všech operačních systémech (Windows, macOS, Linux) a nabízí vyšší výkonnost a škálovatelnost. Architektura je uzpůsobená pro vysokou testovatelnost. Další výhody jsou v množství použitelných knihoven, které nám pomůžou při vývoji.

Architektonický vzor MVC [20], tedy Model-View-Controller nám rozděluje aplikaci do tří skupin kde každá skupina zastává jednotlivou funkci pro chod aplikace. Model obsahuje logiku aplikace a vše co do ní spadá, model naplní data pro View. View obsahuje všechny komponenty které může uživatel vidět. Používá razor engine pro převod kódu na html, přijímá data z kontroléru a renderuje UI. Controller je prostředník mezi Modelem a View který komunikuje mezi těmito komponenty, reaguje na vstupní data od uživatele, manipuluje s daty z modelu, volí vhodné View a aktualizuje zobrazovaná data [27].

Mezi hlavní výhody MVC patří rychlejší vývojový proces, kde na více komponentech můžou programátoři pracovat paralelně a posléze je spojit, vylepšeních jednotlivých zobrazení nemá efekt na data, podpora asynchronních metod, které umožní programátorům vytvářet velmi rychlou aplikaci [17].

5.2 Razor Pages

Razor pages [2] nám představují naše View. Syntaxe je navržena tak aby byla snadno implementovatelná. Skládá se z Razor kódu, C# kódu a pomocných HTML tagů. Základní direktiva je znak `@{}`, uvnitř složených závorek můžeme napsat C# kód, který se po spuštění převede na HTML kód. Každá stránka se skládá z několika podpůrných značek, které nám pomáhají u přístupu k datům, např.: `asp-controller` – který nám říká z jakého kontroléru data brát, `asp-area` – název pro oblast kódu z které brát, `asp-action` – název metody, která se má spustit a podobně [21].

5.3 MySQL Workbench

MySQL Workbench [15] je multiplatformní grafický nástroj pro MySQL databázi. Obsahuje vše potřebné pro modelování dat, vývoj SQL, nástroje pro konfiguraci serveru, vytváření ER modelů, forward a reverse engineering a také klíčové funkce pro správu dokumentace, která

by normálně zabrala velké množství úsilí a času. Nástroj umožňuje kompletní a snadné řešení pro migraci tabulek, objektů a dat ze systému Microsoft SQL Server, Microsoft Access apod. do MySQL. Migrace také podporuje přechod ze starších verzí MySQL na novější.

5.4 Entity Framework Core

Entity Framework Core [3] je technologie obsažená v NuGet knihovně, která nám zajišťuje přístup k databázi za pomoci objektů .NET. Využívá ORM – objektově relační mapování. Pro dotazy nepoužíváme SQL dotazy, ale LINQ výrazy, které jsou dostupné od .NET 3,5 frameworku. Podporuje mnoho databázových providerů (Sql Server, MySql, Oracle DB, PostgreSQL, Azure Cosmos DB ...). Pro migraci dat používáme dva přístupy: Database First a Code First. První zmíněný přístup vytvoří rozhraní a třídy na základě již vytvořené databáze pomocí příkazů obsažených v EF Core. Nevýhoda při použití toho přístupu je, že EF Core nepodporuje vizualizaci při vytváření. Druhý zmíněný přístup vytvoří rozhraní a tabulky na základě již vytvořených entitních tříd. Při použití toho přístupu je možno aktualizovat databázi za použití migrací [19].

5.5 CSS a BootStrap

Kaskádové styly neboli CSS [25] je jazyk, pomocí kterého můžeme stylizovat stránky a upravovat formy v jaké se uživateli prezentují. Dokument je obvykle zapsán pomocí značkovacího jazyka, mezi nejběžnější patří HTML, ale existují i další jako například SVG nebo XML, bez CSS by stylizace stránky byla velmi jednoduchá. Pomocí CSS můžeme měnit barvu písma, styl, velikost oddílů, rozdělovat elementy na řádky a sloupce apod. Pro zapisování stylů se většinou používá externí soubor, který se pak importuje k přiložené stránce, v externím souboru můžeme definovat jednotlivé styly elementů, které se pak použijí na celé zobrazení. Styly můžeme zapisovat i přímo k jednotlivým značkám a tím je stylizovat, při použití přímého zápisu má styl větší prioritu než styl použitý z externího souboru.

Bootstrap [26] je obrovská sbírka kousků kódů, které obsahují HTML, CSS a JavaScript. Jedná se o frontendový vývojový framework, který umožňuje programátorům a také designérům vytvářet plně responzivní webové stránky. Mezi hlavní výhody při použití bootstrapu patří vysoká responzivita, už vytvořené nové komponenty jako je navigační lišta, rozbalovací menu a miniatury. Další z řady výhod je, že se jedná o open-source projekt, tím pádem se do vytváření nových komponent a kódu může zapojit i okolní komunita. Bootstrap nabízí

i několik stovek již připravených šablon, které jsou buď dostupné zdarma nebo za menší poplatek.

5.6 JavaScript

JavaScript je dynamický programovací jazyk, který se nejvíce používá při tvorbě webových stránek, aplikací, her a mnoho dalších. Tento programovací jazyk slouží pro vytvoření dynamických funkcí na webových stránkách, které by se za pomoci HTML a CSS nedali udělat. Jelikož CSS nabízí pouze animace, 2D a 3D transformace, webová stránka by bez použití JavaScriptu byla až z 90% statická. V dnešní době se jedná o 3. nejpopulárnější programovací jazyk.

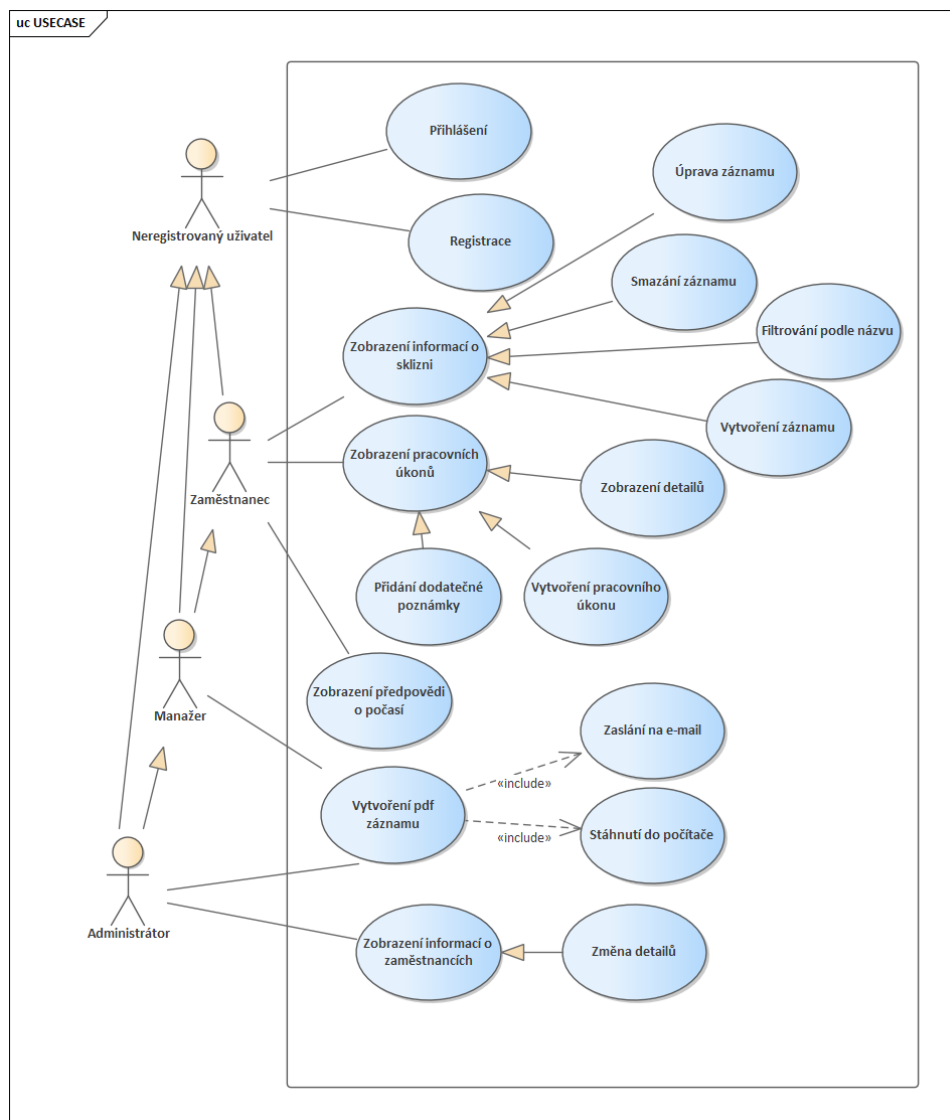
II. PRAKTICKÁ ČÁST

6 NÁVRH APLIKACE

System je navrhnutý na základě funkčních a nefunkčních požadavků. Na základě funkčních požadavků bude vytvořen diagram případů užití pro jednotlivé aktéry v systému. Diagram případů užití má za úkol vymezit hranice systému, vyhledat aktéry, k nim přiřadit případy užití a najít relaci mezi jednotlivými aktéry. K diagramu užití budou vypsány scénáře, které se dělí na úspěšné a alternativní [16].

6.1 Diagram případů užití

Diagram případů užití ukazuje to, jak by se měl systém chovat. V diagramu jsou zachyceni aktéři, kteří přímo komunikují se systémem a samotné případy užití, které nám říkají, co může, který aktér dělat.



Obrázek 1 Diagram případů užití

Na obrázku (Obrázek 1) figurují 4 aktéři, kde každý bude mít jiná práva a povolené akce, které může v systému vykonávat: Neregistrovaný uživatel, Zaměstnanec, Manažer a Administrátor.

Neregistrovaný uživatel představuje aktéra, který ještě není zaregistrován nebo přihlášen.

Zaměstnanec představuje aktéra, který si může zobrazit informaci o sklizni, případně jednotlivý záznam editovat nebo smazat anebo vytvořit záznam. Zobrazit informaci o provedených pracovních úkonech a zobrazit si detail kdo ho přidal. Zobrazit předpověď počasí.

Manažer představuje aktéra, který může vytvářet PDF záznamy a nechat si je zaslat na e-mail nebo stáhnout lokálně do počítače. Dědí všechny funkcionality od Zaměstnance.

Administrátor představuje aktéra, který může spravovat jednotlivé zaměstnance, měnit jim detaily anebo smazat ze systému. Dědí všechny funkcionality od Zaměstnance a Manažera.

6.2 Scénáře

Základní případy užití, které byli představené v minulé kapitole budou v této kapitole blíže představeny a popsány. Scénáře představují seznám kroků mezi aktérem a systémem.

Tabulka 5. Scénář případu užití Registrace

ID: UC001		
Případ užití: Registrace		
Aktéři: Neregistrovaný uživatel		
Vstupní podmínka: /		
Úspěšný scénář		
Kroky	Aktér/Systém	Popis
1.	Aktér	Aktér klikne v navigační liště na registraci.
2.	Systém	Systém zobrazí registrační formulář, který obsahuje pole pro: Jméno, Příjmení, E-mail, Telefonní číslo, Přihlašovací jméno a Heslo.
3.	Aktér	Aktér vyplní formulář a stiskne tlačítko registrovat.
4.	Systém	Systém zobrazí hlášku o úspěšné registraci.

Alternativní scénář		
1.1	Aktér	Aktér vyplní některý z údajů neplatně.
2.2	System	System označí chybné textové pole, které bylo špatně vyplněno.

Tabulka 6. Scénář případu užití Přihlášení

ID: UC002		
Případ užití: Přihlášení		
Aktéři: Neregistrovaný uživatel		
Vstupní podmínka: /		
Úspěšný scénář		
Kroky	Aktér/System	Popis
1.	Aktér	Aktér klikne v navigační liště na přihlášení.
2.	System	System zobrazí formulář pro přihlášení.
3.	Aktér	Aktér vyplní pole pro přihlašovací jméno a heslo.
4.	System	System po autentifikaci přihlásí aktéra a přesměruje na hlavní stránku.
Alternativní scénář		
1.1	Aktér	Aktér vyplní některý z údajů neplatně.
2.2	System	System zobrazí hlášku, že uživatel nebyl nalezen.

Tabulka 7. Scénář případu užití Zobrazení informací o sklizni

ID: UC003		
Případ užití: Zobrazení informací o sklizni		
Aktéři: Admin, Manažer, Zaměstnanec		
Vstupní podmínka: Aktér je přihlášen		
Úspěšný scénář		
Kroky	Aktér/Systém	Popis
1.	Aktér	Aktér klikne v navigační liště na tlačítko „Záznam“.
2.	Systém	Systém zobrazí stránku se všemi dostupnými záznamy.

Tabulka 8. Scénáře případu užití Vytvoření záznamu

ID: UC004		
Případ užití: Vytvoření záznamu		
Aktéři: Admin, Manažer, Zaměstnanec		
Vstupní podmínka: Aktér je přihlášen, Aktér má zobrazené informace o sklizni (ID: UC003)		
Úspěšný scénář		
Kroky	Aktér/Systém	Popis
1.	Aktér	Aktér klikne na tlačítko „Vytvořit záznam“.
2.	Systém	Systém zobrazí stránku pro vytvoření záznamu.
3.	Aktér	Aktér vyplní všechna povinná pole.
4.	Aktér	Aktér klikne na tlačítko „Vytvořit“.
5.	Systém	Systém zavře stránku, zobrazí informaci o úspěšné vytvoření a projeví změny v záznamu.
Alternativní scénář		

1.1	Aktér	Aktér vyplní neplatná data.
2.1	Aktér	Aktér klikne na tlačítko „Vytvořit“.
3.1	System	System označí chybné textové pole, které bylo špatně vyplněno.

Tabulka 9. Scénář případu užití Úprava záznamu

ID: UC005		
Případ užití: Úprava záznamu		
Aktéři: Admin, Manažer, Zaměstnanec		
Vstupní podmínka: Aktér je přihlášen, Aktér má zobrazené informace o sklizni (ID: UC003)		
Úspěšný scénář		
Kroky	Aktér/System	Popis
1.	Aktér	Aktér klikne na tlačítko upravit u daného záznamu.
2.	System	System zobrazí stránku pro úpravu daného záznamu.
3.	Aktér	Aktér upraví požadované pole záznamu.
4.	Aktér	Aktér klikne na tlačítko „Uložit“.
5.	System	System zavře stránku, zobrazí informaci o úspěšné editaci a projeví změny v záznamu.
Alternativní scénář		
1.1	Aktér	Aktér vyplní neplatná data.
2.1	Aktér	Aktér klikne na tlačítko „Uložit“.
3.1	System	System označí chybné textové pole, které bylo špatně vyplněno.

Tabulka 10. Scénář případu užití Smazání záznamu

ID: UC006		
Případ užití: Smazání záznamu		
Aktéři: Admin, Manažer, Zaměstnanec		
Vstupní podmínka: Aktér je přihlášen, Aktér má zobrazené informace o sklizni (ID: UC003)		
Úspěšný scénář		
Kroky	Aktér/System	Popis
1.	Aktér	Aktér stiskne tlačítko „Smazat“ u daného záznamu.
2.	System	System smaže daný záznam ze seznamu.

Tabulka 11. Scénář případu užití Filtrování dle záznamu

ID: UC007		
Případ užití: Filtrování dle zvoleného parametru		
Aktéři: Admin, Manažer, Zaměstnanec		
Vstupní podmínka: Aktér je přihlášen, Aktér má zobrazené informace o sklizni (ID: UC003)		
Úspěšný scénář		
Kroky	Aktér/System	Popis
1.	Aktér	Aktér navolí dané parametry ve vyhledávacím poli.
2.	System	System zobrazí relativní data dle obsahu vyhledávacího pole.

Tabulka 12. Scénář případu užití Zobrazení pracovních úkonů

ID: UC008		
Případ užití: Zobrazení pracovních úkonů		
Aktéři: Admin, Manažer, Zaměstnanec		
Vstupní podmínka: Aktér je přihlášen		
Úspěšný scénář		
Kroky	Aktér/Systém	Popis
1.	Aktér	Aktér klikne v navigační liště na tlačítko „Pracovní úkony“.
2.	Systém	Systém zobrazí stránku se všemi záznamy o pracovních úkonech.

Tabulka 13. Scénář případu užití Vytvoření pracovního úkonu

ID: UC009		
Případ užití: Vytvoření pracovního úkonu		
Aktéři: Admin, Manažer, Zaměstnanec		
Vstupní podmínka: Aktér je přihlášen, Aktér má zobrazené záznamy o pracovních úkonech (ID: UC008)		
Úspěšný scénář		
Kroky	Aktér/Systém	Popis
1.	Aktér	Aktér klikne na tlačítko „Vytvořit pracovní úkon“.
2.	Systém	Systém zobrazí stránku s formulářem pro vytvoření pracovního úkonu.
3.	Aktér	Aktér vyplní všechna povinná pole.
4.	Aktér	Aktér klikne na tlačítko „Vytvořit“.
5.	Systém	Systém zavře stránku, zobrazí informaci o úspěšné vytvoření a projeví změny v záznamu.

Tabulka 14. Scénář případu užití Zobrazení detailu

ID: UC010		
Případ užití: Zobrazení detailu		
Aktéři: Admin, Manažer, Zaměstnanec		
Vstupní podmínka: Aktér je přihlášen, Aktér má zobrazené záznamy o pracovních úkonech (ID: UC008)		
Úspěšný scénář		
Kroky	Aktér/Systém	Popis
1.	Aktér	Aktér si vybere daný záznam a klikne na tlačítko „Detail“.
2.	Systém	Systém zobrazí stránku s detaily daného záznamu.

Tabulka 15. Scénář případu užití Smazání pracovního úkonu

ID: UC011		
Případ užití: Smazání pracovního úkonu		
Aktéři: Admin, Manažer, Zaměstnanec		
Vstupní podmínka: Aktér je přihlášen, Aktér má zobrazené záznamy o pracovních úkonech (ID: UC008)		
Úspěšný scénář		
Kroky	Aktér/Systém	Popis
1.	Aktér	Aktér kline na tlačítko „Smazat“ u vybraného záznamu.
2.	Systém	Systém smaže daný záznam ze seznamu.

Tabulka 16. Scénář případu užití Přidání komentáře

ID: UC012		
Případ užití: Přidání komentáře		
Aktéři: Admin, Manažer, Zaměstnanec		
Vstupní podmínka: Aktér je přihlášen, Aktér má zobrazené záznamy o pracovních úkonech (ID: UC008)		
Úspěšný scénář		
Kroky	Aktér/Systém	Popis
1.	Aktér	Aktér vyplní pole pro komentář u daného záznamu.
2.	Aktér	Aktér klikne na ikonu pro přidání.
3.	Systém	Systém zaktualizuje stránku a zobrazí komentář u daného záznamu.
Alternativní scénář		
1.1	Aktér	Aktér vyplní pole pro komentář moc dlouhým řetězcem.
2.1	Systém	Systém zobrazí chybovou hlášku s popisem.

Tabulka 17. Scénář případu užití Zobrazení předpovědi o počasí

ID: UC013		
Případ užití: Zobrazení předpovědi o počasí		
Aktéři: Admin, Manažer, Zaměstnanec		
Vstupní podmínka: Aktér je přihlášen		
Úspěšný scénář		
Kroky	Aktér/Systém	Popis
1.	Aktér	Aktér klikne na tlačítko „Zobrazit předpověď“ v navigační liště.
2.	Systém	Systém zobrazí stránku s předpovědí počasí.

3.	Aktér	Aktér zadá do pole pro vyhledávání místo pro předpověď.
4.	Aktér	Aktér klikne na ikonu Hledat nebo zmáčkne „Enter“.
5.	System	System zobrazí předpověď počasí pro danou oblast.

Tabulka 18. Scénáře případu užití Vytvoření PDF záznamu

ID: UC014		
Případ užití: Vytvoření PDF záznamu		
Aktéři: Admin, Manažer		
Vstupní podmínka: Aktér je přihlášen, Aktér má zobrazené informace o sklizni (ID: UC003)		
Úspěšný scénář		
Kroky	Aktér/System	Popis
1.	Aktér	Aktér klikne na tlačítko „Vytvořit PDF“
2.	System	System vygeneruje PDF záznam v paměti.

Tabulka 19. Scénář případu užití Zaslání na e-mail

ID: UC015		
Případ užití: Zaslání na e-mail		
Aktéři: Admin, Manažer		
Vstupní podmínka: Aktér je přihlášen, Aktér má zobrazené informace o sklizni (ID: UC003), Aktér má vytvořený PDF záznam (ID: UC014)		
Úspěšný scénář		
Kroky	Aktér/System	Popis
1.	Aktér	Aktér klikne na tlačítko „Zaslat na email“.
2.	System	System zašle PDF záznam do emailové stránky.

Tabulka 20. Scénář případu užití Stáhnutí do počítače

ID: UC016		
Případ užití: Stáhnutí do počítače		
Aktéři: Admin, Manažer		
Vstupní podmínka: Aktér je přihlášen, Aktér má zobrazené informace o sklizni (ID: UC003), Aktér má vytvořený PDF záznam (ID: UC014)		
Úspěšný scénář		
Kroky	Aktér/System	Popis
1.	Aktér	Aktér klikne na tlačítko „Stáhnout lokálně“.
2.	System	System stáhne požadovaný záznam do počítače.

Tabulka 21. Scénář případu užití Zobrazení informací o zaměstnancích

ID: UC017		
Případ užití: Zobrazení informací o zaměstnancích		
Aktéři: Admin		
Vstupní podmínka: Aktér je přihlášen		
Úspěšný scénář		
Kroky	Aktér/System	Popis
1.	Aktér	Aktér klikne na tlačítko „Zaměstnanci“ v navigační liště.
2.	System	System zobrazí stránku se všemi zaměstnanci, kteří jsou zaregistrovaní v systému aplikace.

Tabulka 22. Scénář případu užití Změna detailů

ID: UC018		
Případ užití: Změna detailů		
Aktéři: Admin		
Vstupní podmínka: Aktér je přihlášen, Aktér má zobrazené záznamy o zaměstnancích (ID: UC017)		
Úspěšný scénář		
Kroky	Aktér/Systém	Popis
1.	Aktér	Aktér klikne na tlačítko „Detail“ u daného zaměstnance.
2.	Systém	Systém zobrazí stránku pro úpravu daného zaměstnance
3.	Aktér	Aktér změní požadované pole.
4.	Aktér	Aktér klikne na tlačítko „Uložit“.
5.	Systém	Systém zaktualizuje stránku se zaměstnanci a projeví změny.
Alternativní scénář		
1.1	Aktér	Aktér změní některé z polí nevalidními daty.
2.1	Systém	Systém označí chybné textové pole, které bylo špatně vyplněno.

6.3 ERD a Databázový model

6.3.1 ER Diagram

Návrh ER diagramu vychází ze sběru a analýzy požadavků. ER diagram je typ strukturálního diagramu, který se používá pro návrh databáze. Zajišťuje nám dvě důležité informace o samotném systému. Třídy neboli entity, které se budou v systému objevovat a vztah mezi nimi. U ER Diagramu také uvádíme atributy, které bude daná třída obsahovat a definujeme primární klíč pro danou třídu, primární klíč slouží jako unikátní identifikátor.

Na základě funkčních požadavků byli vytvořené tyto entity, tedy třídy, s kterými bude systém pracovat:

- Uživatel – představuje třídu se zaměstnanci v systému, obsahují informace o zaměstnanci
- AspUživatelRole – třída pro zajištění vztahu mezi rolemi a zaměstnanci
- AspRole – třída vytvořená nástrojem ASP.NET Core Identity, která v sobě nese název jednotlivých rolí
- PracovníÚkon – tabulka která uchovává všechny záznamy o pracovních úkonech
- Sklizeň – třídy se záznamy o sklizni
- Sklep – třída pro informace o všech sklepech
- Odrůda – třída pro všechny odrůdy pro tabulku Sklizeň
- TypPracovníhoÚkonu – tato třída představuje typ pracovního úkonu
- NázevPracovníhoÚkonu – třída obsahující název daného pracovního úkonu, který je odvozený od typu
- Vinohrad – třída pro všechny informace o vinohradu
- Komentář – třída pro přidání komentáře nebo nějakého dodatku k již vytvořeným pracovním úkonům

ER diagram obsažen v příloze viz. P I: ERD

6.3.2 Databázový model

Z ER diagramu byl následně vytvořen databázový model, který nám blíže specifikuje vztahy mezi jednotlivými třídami a vizualizuje nám celý systém [22].

Databázový model se dělí na tři skupiny

- Konceptuální datový model – jedná se o první fázi při návrhu systému, označuje se též jako doménový model, jsou zde zastížené jednotlivé entity a vztahy mezi jinými entitami, tento zápis je obvykle jednoduchý
- Logický datový model – tento model je méně abstraktní a poskytuje více informací a podrobností o konceptech a vztazích mezi entitami v systému, jsou zde i vyobrazené cizí a primární klíče mezi tabulkami, tento datový model je ukázán níže pro náš systém
- Fyzický datový model – nejméně abstraktní model ze všech, poskytuje informace o tom, jak budou data fyzicky uložena v databázi, jedná se o finální návrh

Databázový model obsažen v příloze viz. P II. ERDDataModeling

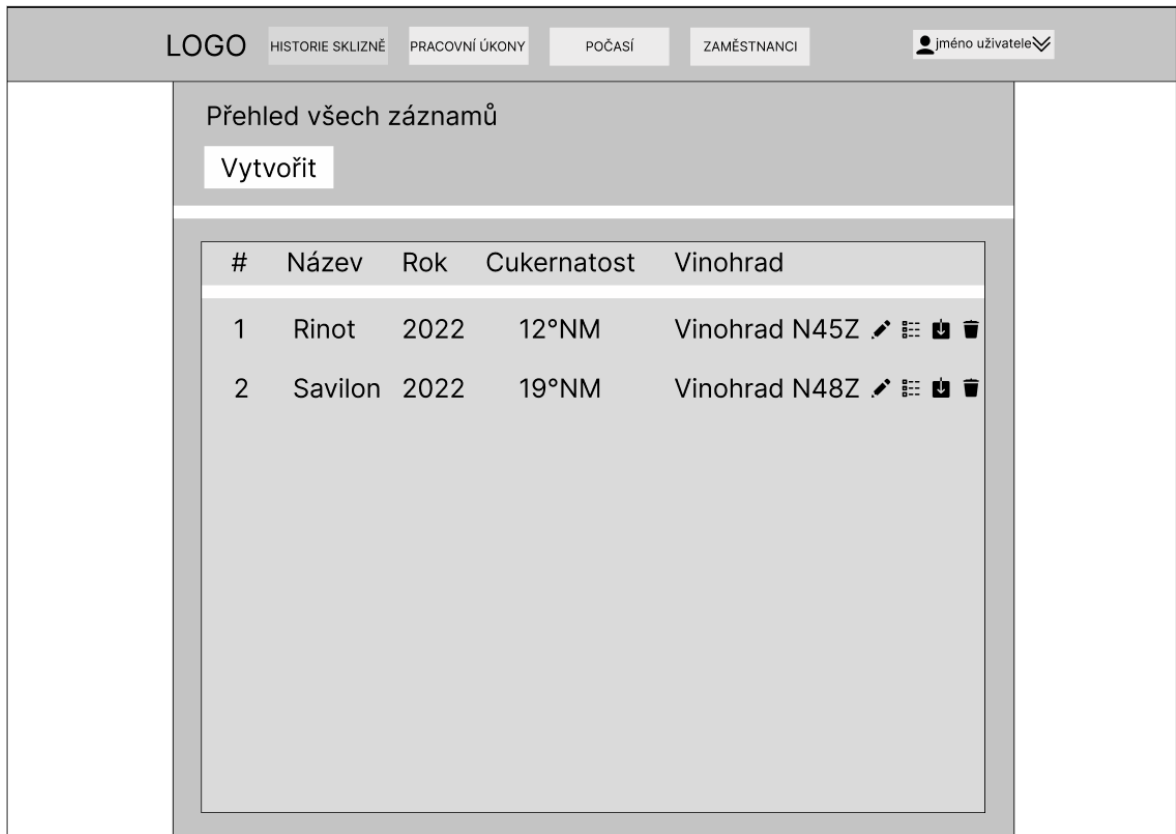
6.4 Drátěné modely

Drátěné modely neboli též „Wireframy“ se používají pro vytvoření prvního náhledu na stránky předtím, než se přidá vizuální design. Během této fáze se můžou odhalit strukturální a funkční chyby se kterými bychom se mohli při vývoji setkat. Mezi výhody vytváření drátěných modelů při vývoji jsou, že poskytnou vizuální strukturu pro klienta a ten si tak dokáže představit, jak by stránka mohla zhruba vypadat. Z praktického hlediska drátěné modely zajišťují správné umístění obsahu s daty a funkcí na stránce na základě potřeb uživatelů a firem [18]. Pro ukázkou bylo vytvořených pár drátěných modelů níže.

The image shows a wireframe model of a registration page. At the top, there is a header box with the text "Vinařská evidence". Below this, there is a central form area with several input fields and buttons. The fields are labeled: "Přihlašovací jméno:", "Jméno:", "Příjmení:", "Heslo:", and "Telefonní číslo:". Below the "Heslo:" field, there is a "Registrace" button. At the bottom of the form area, there is a "Přihlášení" button. The entire form is set against a light gray grid background.

Obrázek 2 Drátěný model registrace

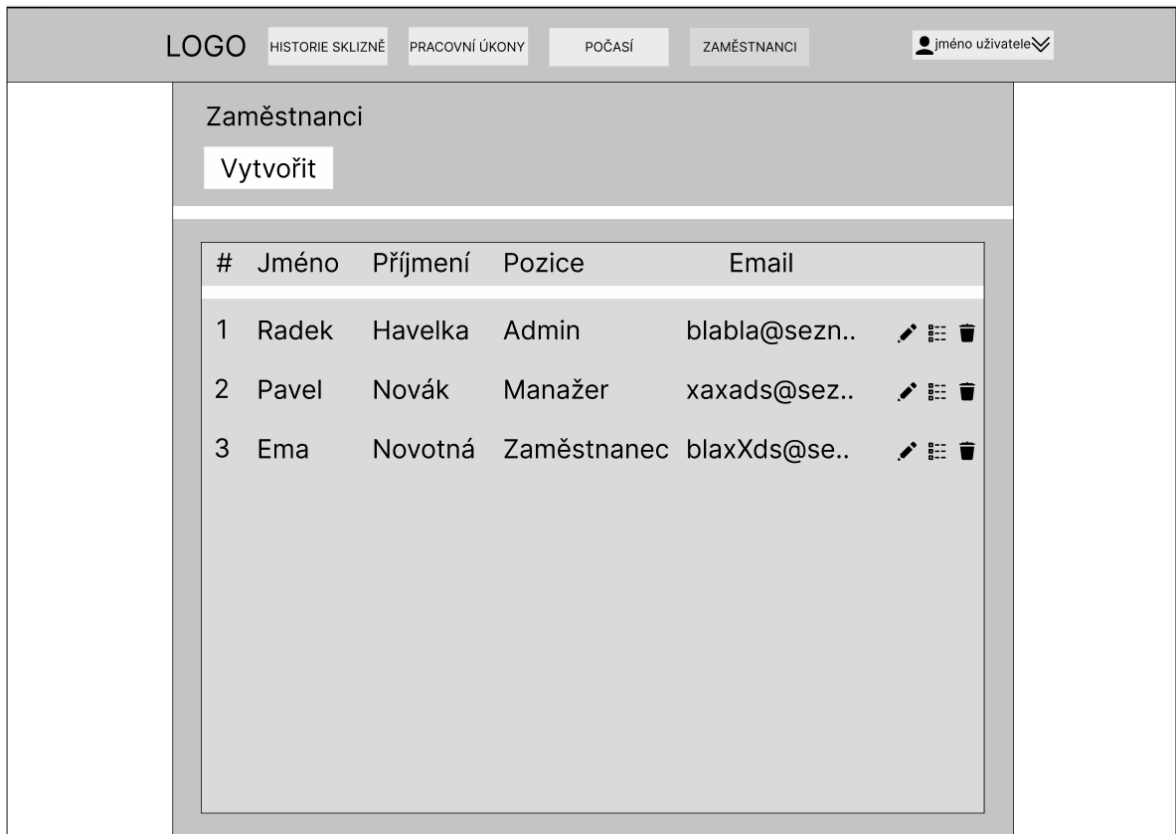
Zobrazení stránky s Registrací (Obrázek 2), kde můžeme vidět pole pro přihlašovací jméno/email které bude sloužit jako přístupové jméno do systému, jméno a příjmení uživatele, heslo a telefonní číslo. Tlačítko na registraci a tlačítko pro přihlášení na přímé přesměrování na stránku pro přihlášení. Stránka pro přihlášení je podobná jako pro registraci s tím, že se zadává jen přihlašovací jméno/email a heslo. Po úspěšné registraci je uživatel přesměrován zpět na hlavní stránku a po následném přihlášení je uživatel přesměrován do aplikace.



#	Název	Rok	Cukernatost	Vinohrad
1	Rinot	2022	12°NM	Vinohrad N45Z
2	Savilon	2022	19°NM	Vinohrad N48Z

Obrázek 3 Drátěný model záznamů

Zobrazení stránky pro Historii sklizní (Obrázek 3), kde můžeme vidět přehled všech přidávaných záznamů, každý záznam má svoje vlastní tlačítko na úpravu daného záznamu, na zobrazení detailů, tlačítko pro stáhnutí do počítače a pro odstranění záznamu. Po kliknutí na záznam se zobrazí další stránka s detaily o daném záznamu jako je například: kdo přidal záznam, množství sklizené révy, velikost sklizeného pole na vinohradě, do jakého sklepa se sklizeň převezla, počáteční cukernatost atd.



Obrázek 4 Drátěný model zaměstnanců

Zobrazení stránky s přehledem všech zaměstnanců zaregistrovaných v systému (Obrázek 4), kde můžeme vidět základní informace o uživateli a při kliknutí na tlačítko detaily se zobrazí kompletní informace předané z Microsoft.Asp.Net.Identity, jako je například normalizovaný email a uživatelské jméno, hash hesla, počet neplatných přihlášení apod. Uživatel v našem případě Admin bude moct upravit údaje uživateli a nebo jej smazat.



Obrázek 5 Drátěný model počasí

Zobrazení stránky s předpovědí počasí pro určitou oblast (Obrázek 5) můžeme vidět jednoduché pole pro vyhledávání, kde zadáme, místo které chceme vyhledat a na základě API požadavku se nám vrátí JSON data, které se zpracují a zobrazí na stránce.

7 IMPLEMENTACE

Vytvoření samotné aplikace probíhalo ve vývojovém prostředí od společnosti Microsoft, konkrétně v programu Visual Studio 2022 Community. Aplikační i prezenční část byla vyvíjena společně a současně. Jak už bylo avizováno Entity Framework Core nabízí dva druhy přístupu, pro tento projekt byl zvolen přístup Code First.

7.1 Propojení databáze a kódu

Po vytvoření aplikace první věc, která byla zapotřebí, byla stáhnutí potřebných balíčků z Nugget Package Manageru pro propojení s databází, těmito balíčky jsou:

- Microsoft.EntityFrameworkCore – tento balíček slouží pro mapování objektů, pro následné propojení s databází, podporuje LINQ dotazy, migraci a aktualizaci dat
- Microsoft.EntityFrameworkCore.SqlServer – tento balíček slouží pro propojení databáze s Entity Framework Core
- Microsoft.EntityFrameworkCore.Tools – tento balíček slouží pro zadávání příkazů do konzole, např.: Add-Migration – přidání migrace, Remove-Migration – odstranění poslední migrace, Drop-Database – odstranění objektů z databáze, Update-Database – aktualizace databáze
- Pomelo.EntityFrameworkCore.MySql – prostředník pro spojení mezi MySQL databází a EF Core

Po úspěšném stáhnutí balíčku bylo potřeba nastavit propojení s databází v kódu – to se provedlo v „appsettings.json“, kde byl přidán ConnectionString, který obsahuje informace o serveru, jméno databáze a přihlašovací údaje pro připojení do databáze.

Následně bylo potřeba vytvořit dané třídy (Obrázek 6) a přidat je do DbContextu (Obrázek 7), který je součástí EF Core balíčku a zajišťuje přesun modelu tříd do databáze.

```
[Table(nameof(Species))]  
Počet odkazů: 27  
public class Species  
{  
    [Key]  
    [Required]  
    Počet odkazů: 10  
    public int Id { get; set; }  
    [Required(ErrorMessage = "Prosím zadejte jméno odrůdy")]  
    Počet odkazů: 9  
    public string SpeciesName { get; set; }  
    [Required(ErrorMessage = "Prosím zadejte popis")]  
    Počet odkazů: 8  
    public string SpeciesDescription { get; set; }  
}
```

Obrázek 6 Ukázka třídy

```
Počet odkazů: 23  
public class WineDbContext : IdentityDbContext<User, Role, int>  
{  
    public DbSet<Species> Species { get; set; }  
}
```

Obrázek 7 Kontextové třídy

Následně bylo potřeba přidat `ConnectionString` (Obrázek 8) do souboru „Startup.cs“ do části „ConfigureServices“ – soubor, který se automaticky vytvoří při vytvoření projektu, obsahuje veškerou konfiguraci, která se volá po každém spuštění.

```
services.AddDbContext<WineDbContext>(options =>  
options.UseMySQL(  
    Configuration.GetConnectionString("MySQLConnectionString"),  
    new MySQLServerVersion("8.0.26")));
```

Obrázek 8 Propojení kódu s databází

Po propojení kódu s databází bylo potřeba spustit „Package Manager Console“ a využít již zmíněnou migraci dat do MySQL databáze, migraci vytvoříme zadáním příkazu „Add-Migration“, po zadání příkazu nás konzole upozorní, jestli proběhlo vše v pořádku a vytvoří se nový soubor, ve kterém můžeme migraci zkontrolovat (Obrázek 9), po kontrole už stačí jen zadat příkaz „Update-Database“ a entita „Species“ se přidá do databáze. Pro kontrolu aktualizace databáze byl použit nástroj MySQL Workbench, který je popsán v použitých technologiích.


```
migrationBuilder.CreateTable(  
    name: "Species",  
    columns: table => new  
    {  
        Id = table.Column<int>(type: "int", nullable: false)  
            .Annotation("MySQL:ValueGenerationStrategy", MySQLValueGenerationStrategy.IdentityColumn),  
        SpeciesName = table.Column<string>(type: "longtext", nullable: false)  
            .Annotation("MySQL:CharSet", "utf8mb4"),  
        SpeciesDescription = table.Column<string>(type: "longtext", nullable: false)  
            .Annotation("MySQL:CharSet", "utf8mb4")  
    },  
    constraints: table =>  
    {  
        table.PrimaryKey("PK_Species", x => x.Id);  
    })  
    .Annotation("MySQL:CharSet", "utf8mb4");
```

Obrázek 9 Migrace tabulky

7.2 Autorizace a Autentifikace

Pro autorizaci a autentifikaci byl využit další balíček, který je součástí EF Core s názvem Microsoft.AspNetCore.Identity.EntityFrameworkCore, jak už si bylo možné povšimnout na obrázku výše (Obrázek 7) DbContext naší aplikace dědí IdentityDbContext, který je určen pro správu uživatelů a rolí v systému a databázi. Pro naši aplikaci bylo využito několik vestavěných funkcí tohoto balíčku, které po propojení (Obrázek 10) vytvoří třídyAspNetRoles – obsahuje jména všech rolí v systému, v našem případě Admin, Manažer a Zaměstnanec, AspNetUsers – která obsahuje uživatelská jména, jména a příjmení uživatelů, emailové adresy atd. a v poslední řadě tabulka AspNetUsersRoles – která slouží jako propojovací entita mezi dvěma již zmíněnými tabulkami. Pro hashování hesel se používá PBKDF2, která funguje na bázi, že přijímá heslo jako vstup, tzv. salt o délce 128 bitů, celé číslo určující počet iterací hashovací funkce, které se mají provést, v našem případě 1000 iterací a celé číslo popisující požadovanou délku klíče na výstupu. Pro použití celého Autorizačního a Autentifikačního nástroje bylo potřeba přidat do souboru „Startup.cs“ funkce „UseAuthentication()“ a „UseAuthorization()“.

```
services.AddIdentity<User, Role>()  
    .AddEntityFrameworkStores<WineDbContext>()  
    .AddDefaultTokenProviders();
```

Obrázek 10 Přidání identity rolí

Pro registraci nového uživatele a přihlášení už vytvořeného uživatele do systému byl vytvořen „AccountController“, který obsahuje HttpGet a HttpPost metody pro přesměrování a tzv. Service, která se stará o přihlášení a registraci uživatelů a také o systém rolí, který je v aplikaci použit.

7.3 Funkce

Po napojení na databázi, vytvoření příslušných entit a vytvoření autentifikace a autorizace aplikace přichází na řadu implementování kontrolérů a zobrazení stránek. Pro každý model, který je uložen do databáze byl vytvořen kontrolér, pro zajištění funkcionality. Níže si ukážeme a popíšeme to jak kontrolér a zobrazení funguje a jak reaguje na vstupní data uživatele.

Pro ukázkou byl vybrán kontrolér Sklizeň (Obrázek 11).

```
[Area("Employee")]
Počet odkazů: 2
public class HarvestController : Controller
{
    private readonly WineDbContext _context;
    private readonly IWebHostEnvironment _env;
    private readonly IConfiguration _config;

    Počet odkazů: 0
    public HarvestController(WineDbContext context, IWebHostEnvironment env, IConfiguration config)
    {
        _context = context;
        _env = env;
        _config = config;
    }
}
```

Obrázek 11 Kontrolér sklizně

Jak je z obrázku zřetelné, třída dědí funkce od třídy „Controller“, která obsahuje všechny potřebné akce pro zajištění komunikace mezi kontrolérem a zobrazením, dále je třída přidána do oblasti zaměstnanec, pro lepší navigaci v aplikaci, třída obsahuje privátní atributy pro naši databázi, ze které získává objekty, konfiguraci a interface, které získává informace o hostování webu, na kterém aplikace běží.

Pro výpis všech záznamů na stránce je implementována metoda „Index()“, kde se využívají LINQ dotazy, po navázání všech potřebných tabulek, metoda vrácí list, který se předává danému zobrazení.

Pro vytvoření nového záznamu byla implementována metoda „Create()“, která má atribut [HttpGet], atribut zajišťuje správné mapování kdykoliv je příslušná akce v zobrazení zavolána. Pro správné mapování jednotlivých tříd, je uvnitř aplikace volána metody akce „ViewData“, která předává danému zobrazení příslušná data ve formě listu. Pro zajištění správného uložení do databáze a odeslání na server, je vytvořena další metoda se stejným názvem, ale s atributem [HttpPost]. Atribut zajišťuje, že funkce se bude volat jen v případě, kdy chceme odeslat data, to se většinou děje v případě stisku nějakého tlačítka pro vytvoření, editaci nebo smazání. Uvnitř metody se první provede kontrola, jestli jsou všechna data

předané ze zobrazení validní a pak se volají funkce pro přidání a uložení do databáze, nakonec je ještě předána hláška o úspěšném přidání do modalu, který se zobrazí po přesměrování.

Pro úpravu záznamu byla implementována metoda „Edit()“ s atributem [HttpGet] a parametrem id, jelikož každý záznam má své id, tak první věc která musí v metodě proběhnout je jestli id předané v parametru není null, pokud není, vytvoříme si novou proměnnou a zavoláme funkci, která podle id najde příslušnou třídu a uloží ji do naší proměnné, zajistíme kontrolu dat a opět jako ve funkci pro vytvoření sklizně, předáme našemu zobrazení data z příslušných tříd. Druhá metoda se stejným názvem zajišťuje správnou aktualizaci dat v dané třídě, v parametru předáváme opět id a samotnou třídu, první věc která probíhá je kontrola jestli se id v parametru předané zobrazení, shoduje s primárním klíčem dané třídy, pokud ano, provede se validace dat a je přidán „try“ a „catch“ blok, kde v prvním bloku se provede aktualizace dat a v druhém bloku, dochází ke kontrole jestli nedochází k duplicitě dat, po úspěšné aktualizaci je uživatel přesměrován zpátky na stránku se záznamy a je mu zobrazen modal, jestli vše proběhlo v pořádku.

Obdobným způsobem jsou implementované ostatní metody v kontroléru a v ostatních kontrolérech jednotlivých tříd.

7.4 Zabezpečení proti útokům

7.4.1 SQL Injection

Na zabezpečení aplikace proti SQL Injection útokům je využit již několikrát zmíněný EF Core, který na přístup k datům nevyužívá klasické SQL dotazy, ale tzv. LINQ dotazy, které jsou součástí balíčku, tyto dotazy jsou normálně psané v kódu a posléze transformované do SQL dotazu. Při použití LINQ dotazů omezujeme útoky SQL Injection. Pro ukázkou byl vytvořen dotaz pro výpis všech pracovních úkonů v systému (Obrázek 12).

```
var dbcontext = _context.Works
.Include(c => c.WorkType)
.Include(c => c.Cellar)
.Include(c => c.Wineyard)
.Include(c => c.User)
.Include(cy => cy.WorkName).ToListAsync();
```

Obrázek 12 Ukázka LINQ dotazu

7.4.2 CSRF útok

Pro ochranu proti CSRF útokům je možno použít několik technik nabízených samotným ASP.NET Core, ku příkladu ke každé metodě, která volá změnu nějaké stavu, stačí přidat atribut „ValidateAntiForgeryToken“, který blokuje nepovolené akce, které obsahují nevalidní antiforgery tokeny (Obrázek 13).

```
[HttpPost]
[ValidateAntiForgeryToken]
Počet odkazů: 0
public async Task<IActionResult> Create(Harvest harvest)
{
```

Obrázek 13 Ukázka antiforgery tokenu

Automatické generování antiforgery tokenů lze nastavit přímo v HTML formuláři, kdy k tagu <form> připneme atribut „method=“post““. Další forma ochrany proti útokům je samotné používání autentizace, kdy po přihlášení se vygeneruje token, který se předává, ale je nutno podotknout, že i takto vygenerované cookies lze napadnout. Aplikace také přímo po vytvoření využívá funkci „UseHttpsRedirection()“, která slouží pro vytvoření prostředníka mezi HTTP a HTTPS.

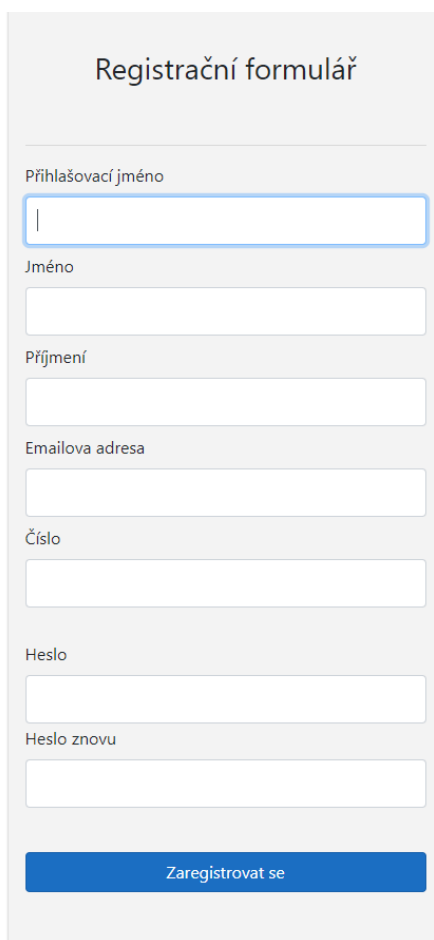
8 PRŮVODCE APLIKACÍ

Tato část se bude věnovat samotnému rozhraní celé aplikace a bude zde ukázáno grafické zpracování aplikace a několik funkcí, jako například registrace a přihlášení nového uživatele, vytvoření sklepa a vinohradu a její úprava nebo smazání a vytvoření pracovního úkonu, který používá atributy sklepa nebo vinohradu, to záleží podle toho, o jaký typ úkonu jde.

8.1 Registrace a přihlášení

Pro samotné používání aplikace je potřeba se do systému zaregistrovat a přihlásit, pokud nebudou tyto podmínky splněny, uživatel se dostane jen na hlavní stránku a nebude moci používat žádné funkcionality systému.

Pro registraci musí uživatel kliknout na tlačítko Registrace v navigační liště, po kliknutí, aplikace zobrazí novou stránku s formulářem pro registraci (Obrázek 14).



Registrační formulář

Přihlašovací jméno

Jméno

Příjmení

Emailova adresa

Číslo

Heslo

Heslo znovu

Zaregistrovat se

Obrázek 14 Registrační formulář

Přihlašovací jméno, emailová adresa, heslo a zopakování hesla jsou povinné atributy, když některé z těchto podmínek uživatel zadá nevalidně, objeví se příslušná hláška, kde nastala chyba (Obrázek 15), validní zadání emailové adresy obsluhuje samotná aplikaci.

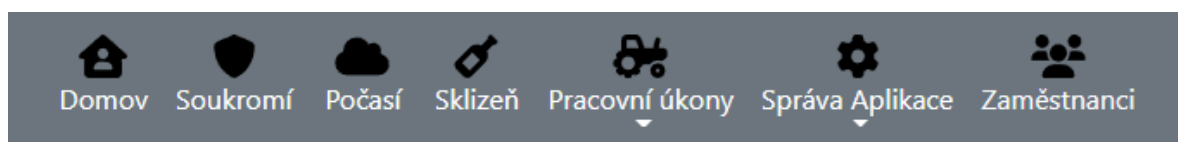
- Zadejte přihlašovací jméno
- Zadejte emailovou adresu
- Zadejte heslo
- Zadané hesla se neshodují

Obrázek 15 Ukázka upozornění

Po úspěšné registraci se může uživatel přesunout k přihlášení, když u přihlášení zadá stejná data, jaké použil pro registraci, systém jej přesměruje na hlavní stránku aplikace, při zadání nevalidních údajů jej systém upozorní obdobným způsobem jako na obrázku výše (Obrázek 15).

8.2 Navigační lišta a role

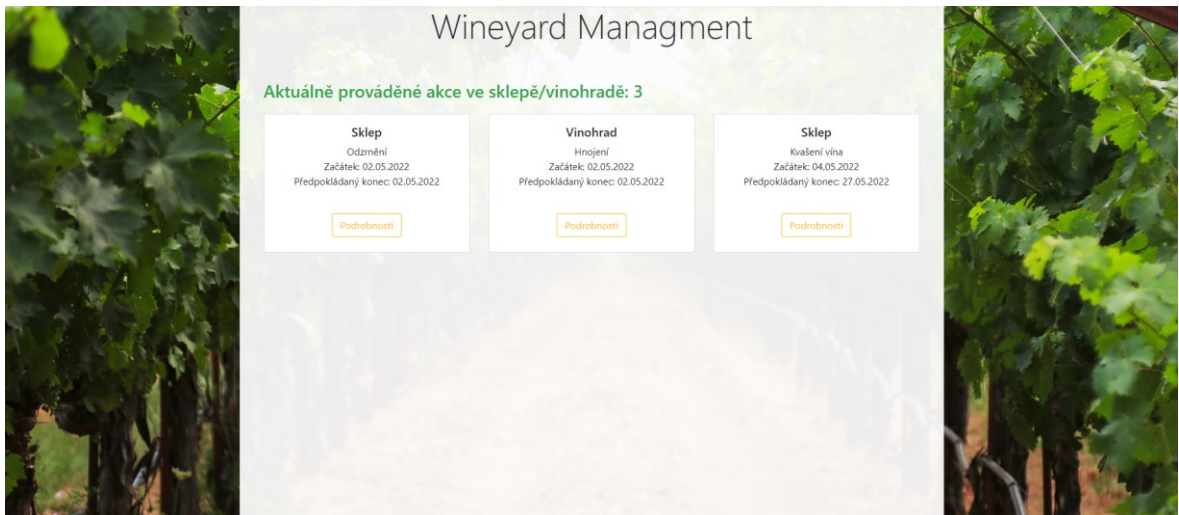
Jak už bylo ukázáno v návrhu aplikace, kapitola výše, tak aplikace je rozdělená do rolí, kde po přihlášení uživatel automaticky dostává roli zaměstnance, takovému uživateli je zpřístupněno pouze použití „Počasí“, „Sklizně“ a „Pracovních úkonů“ v navigační liště, zbytek funkcí je před tímto uživatelem skryta a nemůže se k nim dostat. Manažer má navíc zpřístupněné tlačítko pro „Správu Aplikace“, kde se nachází kompletní přidání činností dostupných v aplikaci, jména vinohradu, jména odrůd pěstovaných a zpracovávaných v daném vinohradě a sklepě a také jména sklepů. Admin má kompletní přístup do všech částí aplikace, má navíc přidané tlačítko „Zaměstnanci“, kde se nachází kompletní přehled uživatelů zaregistrovaných do systému (Obrázek 16). V následujících podkapitolách budou tyto činnosti blíže rozebrány.



Obrázek 16 Navigační lišta

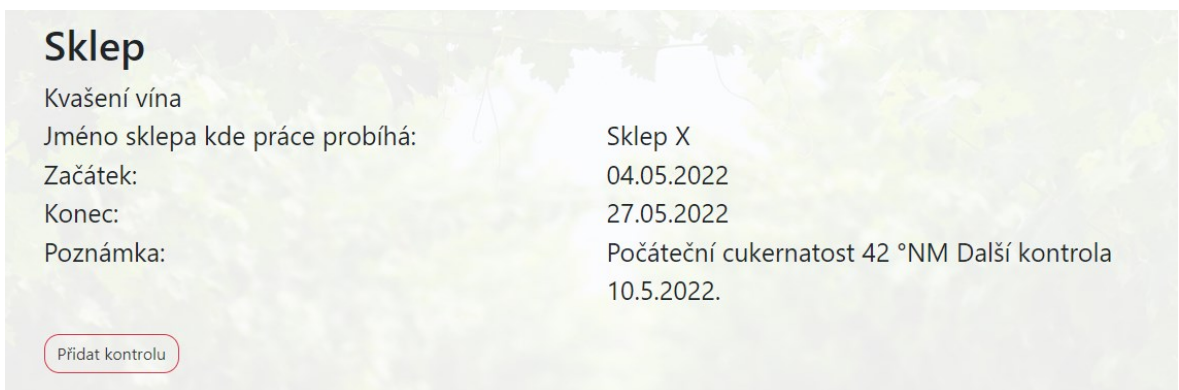
8.3 Hlavní stránka

Po popsání navigační lišty se můžeme přesunout k hlavní stránce aplikace, na hlavní stránce se dynamicky zobrazují právě prováděné práce, uživatel dostává informaci, kolik přesně pracovních úkonů je teď aktivních a jaké to jsou (Obrázek 17).



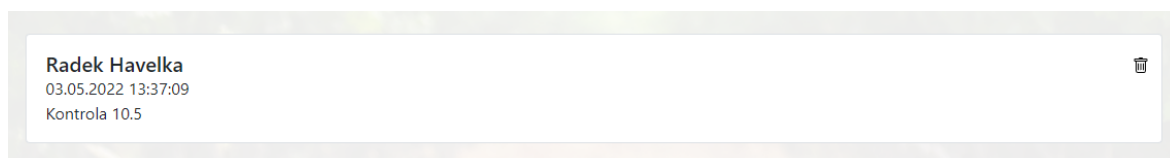
Obrázek 17 Titulní strana

Dále si uživatel může pracovní úkon zobrazit detailněji na další stránce, to provede kliknutím na tlačítko „Podrobnosti“, po stisknutí se uživateli zobrazí nová stránka, kde uživatel dostane bližší informace k danému úkolu a má zde možnost přidat komentář (Obrázek 18).



Obrázek 18 Podrobnosti pracovního úkonu

Pro přidání komentáře je potřeba kliknout na tlačítko „Přidat kontrolu“ po stisknutí tlačítka je uživateli zobrazen modal s textovým polem, který má nastavený maximální délku 250 znaků, po vyplnění textového pole a stisknutí tlačítka pro přidání, je komentář zobrazen na stránce, jeden uživatel zde může přidávat komentáře podle své libosti. U komentáře lze vidět kdo jej napsal, datum přidání komentáře, samotný text a je zde tlačítko pro smazání komentáře (Obrázek 19).



Obrázek 19 Přidání komentáře

8.4 Pracovní úkony

Po stisknutí tlačítka „Pracovní úkony“ v navigační liště se uživateli zobrazí rozbalovací menu, kde jsou dvě tlačítka, jedno pro přidání pracovního úkonu, druhé pro zobrazení. Při stisku tlačítka pro vytvoření je uživatel přesměrován na novou stránku, kde může daný úkon vytvořit, pro typ práce a jméno práce je vytvořené dynamické obnovování pole pomocí JavaScriptu, kde při zvolení typu práce sklep se zobrazí práce, které jsou dostupné pouze ve sklepech, při vybrání vinohradu je to přesně naopak, dále zde uživatel vybere začátek a předpokládaný konec práce a podle toho jaký typ pracovního úkonu vybral, vybere buď část vinohradu nebo část sklepa, závěrem ještě může přidat poznámku a stiskne tlačítko „Vytvořit“. Po vytvoření záznamu je uživatel přesměrován na stránku s výpisem všech záznamů přidávaných do aplikace (Obrázek 20). Uživatel se může podívat na výpis všech prací ve vinohradě nebo sklepech s možností upravit daný záznam v případě zadání nepřesných informací anebo celý záznam odstranit. Záznamy jsou na stránce řazené podle předpokládaného konce pracovního úkonu, tím se rozumí, že práce, které se mají dokončit v nejbližších dnech jsou zobrazené navrchu. Po stisknutí tlačítka „Detaily“ se uživateli zobrazí kompletní výpis se všemi náležitostmi. Je zde také implementované pole pro vyhledávání podle jména dané práce. Po vyplnění pole, jsou uživateli vyfiltrovány záznamy. Počet záznamů na stránce je omezen na 10, pak se stránka rozdělí a záznamy se přesunou na další stránky, pro přechod mezi stránkami je implementovaný číselník, který se nachází v pravém dolním rohu stránky (Obrázek 20).

Zobrazení pracovních úkonů

Hledej podle názvu

Typ práce	Jméno práce	Od	Do	Autor	Poznámka	Akce
Vinohrad	Hnojení	02-05-2022	02-05-2022	AdminRadek	dsadas	Úprava Detaily Smazat
Sklep	Odzměnění	02-05-2022	02-05-2022	AdminRadek	dsdsdsxxxxx	Úprava Detaily Smazat
Sklep	Kvašení vína	03-05-2022	03-05-2022	AdminRadek	dsa	Úprava Detaily Smazat
Sklep	Kvašení vína	04-05-2022	27-05-2022	AdminRadek	Počáteční cukernatost 42 °NM Další kontrola 10.5.2022.	Úprava Detaily Smazat

1

Obrázek 20 Zobrazení všech záznamů

8.5 Sklizeň

Jako další tlačítko v navigační liště je „Sklizeň“, po stisknutí tlačítka vyjede obdobné menu jako v kapitole výše, kde uživatel může vytvořit záznam o sklizni nebo se podívat na všechny záznamy v systému, pro vytvoření je nutné zadat jaká odrůda byla sklizena, jaké množství bylo sklizeno, počáteční cukernatost a obsah kyselin, jako doplňková informace slouží teplota na vinohradě v době sklizně a stav počasí, dále uživatel musí zadat datum startu a konce sklizně a přiřadit část vinice, kde sklizeň probíhala a sklep do kterého bude vinná réva uskladněna před dalším zpracováním, po splnění uživatel klikne na tlačítko vytvořit, systém přesměruje uživatele na stránku se všemi záznamy. Stejně jako v kapitole výše stránka obsahuje filtrační vyhledávací pole, které jde filtrovat podle jména odrůdy a také číselník pro lepší orientaci mezi záznamy, navíc je zde možnost vytvoření PDF záznamu a uložení do lokálního počítače anebo zaslání na emailovou adresu, pro tyto funkce slouží separátní tlačítka ve sloupci pro akce. Uživatel také může daný záznam upravit pro korekci dat, odstranit jej anebo se podívat na kompletní detaily.

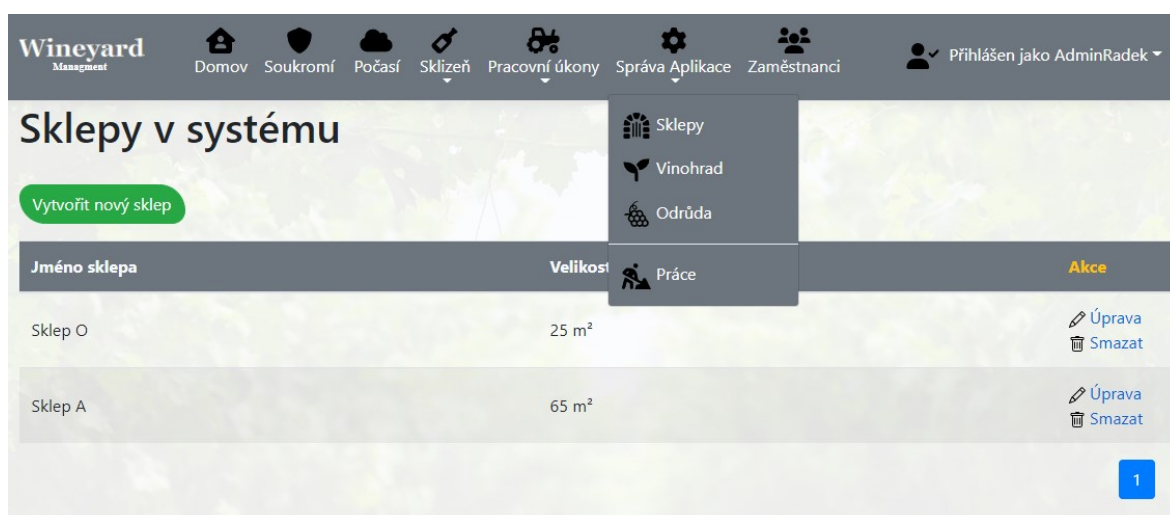
8.6 Počasí

Poslední položkou dostupnou pro normálního pracovníka je „Počasí“. Stránka slouží jako informativní nástroj pro vinaře. Je zde implementované vyhledávací pole, kde je potřeba zadat danou oblast pro vytvoření požadavku. Po stisknutí tlačítka pro vyhledávání je

vytvořen nový požadavek na API propojenou s aplikací, pro účely bakalářské práce byla použita API ze stránky „openweathermap.org“ [24].

8.7 Správa aplikace

Tlačítko v navigační liště pro správu aplikace je jen pro uživatele v rolích manažer a admin. Po rozkliknutí vyjede rozbalovací menu s několika dostupnými funkcemi, funkce obsažené v tomto menu plní funkcionalitu pro počáteční nastavení aplikace, uživatel vytvoří nový sklep, nějak jej pojmenuje a zadá jeho velikost (Obrázek 21), vytvoří nový vinohrad, jeho rozlohu a nadmořskou výšku a vytvoří odrůdy které se na vinohradě pěstují, všechny záznamy může kdykoliv upravit, smazat a nebo vytvořit nový, poslední funkce je pro vytvoření jména pracovního úkonu, které se používá jako jedno z polí v kapitole 8.4 pro vytvoření pracovního úkonu.



Obrázek 21 Zobrazení sklepů v systému

8.8 Zaměstnanci a panel uživatele

Poslední položkou v navigaci, která je dostupná pouze pro administrátora, je „Zaměstnanci“, tato položka zajišťuje administrátorovi přehled o tom, kdo je v aplikaci registrován. Jsou zde funkce pro úpravu uživatele v případě, že by bylo potřeba změnit přihlašovací jméno, nebo jméno a příjmení uživatele, emailovou adresu nebo telefonní číslo, je zde i funkce pro smazání uživatele ze systému.

Poslední položka dostupná pro všechny uživatele v systému, je v pravé části navigace, kde uživatel vidí, pod kým je přihlášen, po rozkliknutí se rozbalí menu, kde se uživatel může

odhlásit anebo změnit heslo, pro změnu hesla je vyžadována znalost původního hesla, pak jen stačí napsat nové heslo a potvrdit jej.

9 TESTOVÁNÍ

Za účelem otestování aplikace byl do projektu zakomponován nový projekt, který se specializuje na automatizovaný testování komponent v aplikaci. Bylo využito SeleniumLibrary pro testování ASP.NET projektů. Cílem testů je najít a odstranit bugy v aplikaci, případně najít neošetřené vstupy a výstupy. Pro lepší přehled bylo několik testů vypsáno v tabulkách níže.

Tabulka 23. Testovací případ přímé navigace

ID: 001	
Testovací případ: Přímá navigace	
Vstupní podmínka: /	
Testovací kroky	
Kroky	Popis
1.	Vyplnění linku (" https://localhost:5001/Admin/Users ") do vyhledávacího pole.
2.	Stisknutí tlačítka Enter.
Očekávaný výsledek	Aplikace přesměruje uživatele na stránku přihlášení, uživatel se nedostane na příslušný link.

Tabulka 24. Testovací případ pro validní přihlášení.

ID: 002	
Testovací případ: Validní přihlášení	
Vstupní podmínka: /	
Testovací kroky	
Kroky	Popis
1.	Uživatel klikne na tlačítko přihlášení v navigační liště.
2.	Uživatel vyplní všechna pole pro přihlášení a stiskne tlačítko přihlásit.

3.	Aplikace ověří data.
Očekávaný výsledek	Aplikace přesměruje uživatele na hlavní stránku.

Tabulka 25. Testovací případ pro ne-validní přihlášení

ID: 003	
Testovací případ: Ne-validní přihlášení	
Vstupní podmínka: /	
Testovací kroky	
Kroky	Popis
1.	Uživatel klikne na tlačítko přihlášení v navigační liště.
2.	Uživatel vyplní všechna pole pro přihlášení a stiskne tlačítko přihlásit.
3.	Aplikace ověří data.
Očekávaný výsledek	Aplikace upozorní uživatele na to, že zadal neplatné údaje.

Tabulka 26. Testovací případ pro zobrazení sklepů

ID: 004	
Testovací případ: Zobrazení sklepů	
Vstupní podmínka: Uživatel je přihlášen	
Testovací kroky	
Kroky	Popis
1.	Uživatel klikne na tlačítko správa aplikace.
2.	Uživatel klikne na tlačítko sklepy.
Očekávaný výsledek	Aplikace přesměruje uživatele na stránku se zobrazením sklepů.

Tabulka 27. Testovací případ pro vytvoření sklepa validními daty

ID: 005	
Testovací případ: Vyplnění formuláře pro vytvoření sklepa – validní	
Vstupní podmínka: Uživatel je přihlášen	
Testovací kroky	
Kroky	Popis
1.	Vyplnění linku (" https://localhost:5001/Admin/Cellar/Create ") do vyhledávacího pole.
2.	Uživatel vyplní pole se jménem řetězcem („Sklep T“) a pole s velikostí („25“).
3.	Uživatel klikne na tlačítko „Vytvořit“
Očekávaný výsledek	Aplikace přesměruje uživatele na stránku se záznamy, zobrazí modal o úspěšném přidání.

Tabulka 28. Testovací případ pro vytvoření sklepa ne-validními daty

ID: 006	
Testovací případ: Vyplnění formuláře pro vytvoření sklepa – ne-validní	
Vstupní podmínka: Uživatel je přihlášen	
Testovací kroky	
Kroky	Popis
1.	Vyplnění linku (" https://localhost:5001/Admin/Cellar/Create ") do vyhledávacího pole.
2.	Uživatel vyplní pole se jménem řetězcem („Sklep X“) a pole s velikostí („123456789123456“).
3.	Uživatel klikne na tlačítko „Vytvořit“
Očekávaný výsledek	Aplikace zobrazí chybovou hlášku u pole s velikostí.

10 ZHODNOCENÍ A PŘÍPADNÉ VYLEPŠENÍ

Výstupem bakalářské práce je aplikace pro podporu malovinařů, implementovaný systém nabízí několik základních funkcí pro správu pracovních úkonů ve sklepě nebo vinohradě a zaznamenávání jednotlivých sklizní pro lepší porovnání a zhodnocení celého ročníku. Aplikace byla realizována pouze lokálně, jako jedno z hlavních vylepšení při nasazení v reálném prostředí je umístění na host server, například Azure nabízí studentům roční pronájem zdarma. V rámci otestování celé aplikace a zkoušky praktického použití byli navrhnuté vylepšení které si představíme v bodech níže:

- Admin by měl mít možnost měnit role pro jednotlivé uživatele
- Přidat modul pro pracovní směny, kde by byla možnost plánovat si směny, tak aby se nepřekrývali
- Přidat funkcionalitu pro resetování hesla v případě, že bylo zapomenuto, například emailem
- Vylepšit modul s předpovědí počasí pro celý týden
- Zaslání PDF emailem na vybranou emailovou adresu
- Přidání kalendáře pro lepší přehlednost pracovních úkonů
- Lepší responzivita pro mobilní zařízení

ZÁVĚR

Cílem této bakalářské práce bylo navrhnout a implementovat aplikaci pro podporu malovínařů. Důležitým aspektem při návrhu a implementaci bylo si cíleně stanovit postup, který bylo za potřebí striktně dodržovat. Bakalářská práce je rozdělena do 10 kapitol.

Úvod bakalářské práce přibližuje, kdo nebo co jsou malovínaři. Stručně popisuje základní postup pro pěstování vinné révy a následné výroby vína. Závěrem jsou popsány jednotlivé druhy vín a rozdíl mezi nimi.

Druhá kapitola probírá již dostupné řešení, pro tuhle kapitolu byly vybrány řešení AgroWin, VineInfo, Vintrace, eVineyard a Process2Wine. Pro každou aplikaci jsou stanoveny výhody a nevýhody a letmý popis funkcionalit. Závěrem je zhodnocení všech vybraných řešení.

Třetí kapitola se zabývá sběrem a analýzou požadavků. Sběr a analýza požadavků byla zpracována v rámci unifikovaného procesu vývoje aplikací. Proto jsou požadavky rozděleny na funkční a nefunkční, kde funkční popisují to, jak by měl systém fungovat, kdežto nefunkční popisují vlastnosti systému jako je rychlost, bezpečnost, přenositelnost apod.

Čtvrtá kapitola řeší nejčastější útoky na webové aplikace, před kterými je aplikaci nutné zabezpečit. Popisuje to, jak k takovým útokům dochází a ke každému útoku je vybrán příklad.

Pátá kapitola popisuje technologie, které byly použity pro vývoj webové aplikace pro podporu malovínařů. Technologie byly vybrány na základě zkušeností autora. Zároveň pátá kapitola ukončuje teoretickou část bakalářské práce.

Šestá kapitola začíná s návrhem samotného systému. V první části této kapitoly byl vytvořen diagram případů užití, kde jsou vyobrazeni aktéři, kteří budou v systému působit. Byli zde vytvořeny hlavní případy užití systému, kde ke každému případu byl vytvořen scénář, který se podle potřeby dělí na hlavní neboli úspěšný scénář a alternativní neboli neúspěšný scénář. V druhé části byl vytvořen entitně relační diagram, kde byly vytvořeny třídy systému a vazby mezi jednotlivými třídami. Z entitně relačního digramu byl vytvořen databázový model, který z něj přímo vychází. Závěrem bylo popsání a vytvoření drátěných modelů.

Sedmá kapitola obsahuje samotnou implementaci. Byla zde popsána základní konfigurace aplikace, propojení s databází a stažení potřebných balíčků. Také zde byla popsána autentifikace a autorizace a v návaznosti na čtvrtou kapitolou, zde byla vyčleněná část na ochranu

systemu proti útokům na webové aplikace. Mezi část této kapitola též patří popis metody kontroléru.

V osmé kapitole byl vytvořen průvodce aplikací, který může sloužit i jako příručka. V této části byli popsány jednotlivé stránky, které se v systému nachází a popis několika funkcionalit.

V deváté kapitole bylo vytvořeno několik automatizovaných testů pro odstranění bugů a ošetření výjimek.

Desátá a poslední kapitola se zabírala celkovým zhodnocením vytvořené aplikace a na základě zjištěných nedostatků nebo nových nápadů byl vytvořen seznam případných vylepšení.

SEZNAM POUŽITÉ LITERATURY

- [1] ROTH, Daniel, Rick ANDERSON a Shaun LUTTIN. *Přehled ASP.NET Core* [online]. Microsoft, c2022 [cit. 2022-05-08]. Dostupné z: <https://docs.microsoft.com/cs-cz/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-5.0>
- [2] ANDERSON, Rick, Dave BROCK a Kirk LARKIN. *Úvod do Razor stránek v ASP.NET Core* [online]. Microsoft, c2022 [cit. 2022-05-08]. Dostupné z: <https://docs.microsoft.com/cs-cz/aspnet/core/razor-pages/?view=aspnetcore-6.0&tabs=visual-studio>
- [3] *Entity Framework Core* [online]. Microsoft, c2022 [cit. 2022-05-08]. Dostupné z: <https://docs.microsoft.com/cs-cz/ef/core/>
- [4] PAVLOUŠEK, Pavel. *Výroba vína u malovinařů* [online]. 2., aktualiz. a rozš. vyd. Grada, 2010 [cit. 2022-05-08]. ISBN 978-80-247-7472-5.
- [5] *Postup výroby vína* [online]. VD Vinotéky, c2013-2022 [cit. 2022-05-08]. Dostupné z: <https://www.vinoadestilaty.cz/vinny-radce/postup-vyroby-vina/>
- [6] HEINISCH, Jiří. *AgroWin software pro zemědělce a vinaře* [online]. [cit. 2022-05-08]. Dostupné z: <https://agrowin.cz/>
- [7] *VineInfo Vineyard Software* [online]. PremiereVision, c2004-2021 [cit. 2022-05-08]. Dostupné z: <https://vineinfo.com/index.html>
- [8] *Vintrace* [online]. c2022 [cit. 2022-05-08]. Dostupné z: <https://www.vintrace.com/cs/tgl/>
- [9] *Vineyard management software* [online]. Elmibit, c2022 [cit. 2022-05-08]. Dostupné z: <https://www.evineyardapp.com/>
- [10] *Process2Wine* [online]. Ertus Group, c2022 [cit. 2022-05-08]. Dostupné z: <https://www.process2wine.com/en/>
- [11] *SQL injection* [online]. PortSwigger, c2022 [cit. 2022-05-08]. Dostupné z: <https://portswigger.net/web-security/sql-injection>
- [12] *Cross-site scripting* [online]. PortSwigger, c2022 [cit. 2022-05-08]. Dostupné z: <https://portswigger.net/web-security/cross-site-scripting>
- [13] KIRSTENS. *Cross Site Request Forgery (CSRF)* [online]. OWASP Foundation, c2022 [cit. 2022-05-08]. Dostupné z: <https://owasp.org/www-community/attacks/csrf>

- [14] *Cross-site request forgery (CSRF)* [online]. PortSwigger, c2022 [cit. 2022-05-08]. Dostupné z: <https://portswigger.net/web-security/csrf>
- [15] *MySQL Workbench* [online]. Oracle, c2022 [cit. 2022-05-08]. Dostupné z: <https://www.mysql.com/products/workbench/>
- [16] ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky. 2.*, aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 9788025115039.
- [17] *Six Benefits Of Using MVC Model For Effective Web Application Development* [online]. Brainvire Infotech, c2022 [cit. 2022-05-08]. Dostupné z: <https://www.brainvire.com/six-benefits-of-using-mvc-model-for-effective-web-application-development/>
- [18] *What is wireframing?* [online]. Experience UX, c2022 [cit. 2022-05-08]. Dostupné z: <https://www.experienceux.co.uk/faqs/what-is-wireframing/>
- [19] *Entity Framework Core* [online]. EntityFrameworkTutorial.net, c2020 [cit. 2022-05-08]. Dostupné z: <https://www.entityframeworktutorial.net/efcore/entity-framework-core.aspx>
- [20] SMITH, Steve. *Overview of ASP.NET Core MVC* [online]. Microsoft, c2022 [cit. 2022-05-08]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?msclkid=663a461bc4c311ecac603cdac032439e&view=aspnetcore-5.0>
- [21] ANDERSON, Rick, N. Taylor MULLEN, Dave PAQUETTE a Jerrie PELSER. *Pomocné rutiny značek ve formulářích v ASP.NET Core* [online]. Microsoft, c2022 [cit. 2022-05-08]. Dostupné z: <https://docs.microsoft.com/cs-cz/aspnet/core/mvc/views/working-with-forms?view=aspnetcore-6.0>
- [22] *Data Modeling* [online]. IBM Cloud Education, c2020 [cit. 2022-05-08]. Dostupné z: <https://www.ibm.com/cloud/learn/data-modeling>
- [23] MEGIDA, Dillion. *What is JavaScript? A Definition of the JS Programming Language* [online]. Free Code Camp, 2021 [cit. 2022-05-08]. Dostupné z: <https://www.freecodecamp.org/news/what-is-javascript-definition-of-js/>
- [24] *Weather API* [online]. OpenWeather, c2012 — 2022 [cit. 2022-05-08]. Dostupné z: <https://openweathermap.org/api>

[25] *What is CSS?* [online]. Tutorial Point, c2022 [cit. 2022-05-08]. Dostupné z: https://www.tutorialspoint.com/css/what_is_css.htm

[26] OUELLETTE, Alexandre. *What is Bootstrap: A Beginner's Guide* [online]. Career-Foundry, 2021 [cit. 2022-05-08]. Dostupné z: <https://careerfoundry.com/en/blog/web-development/what-is-bootstrap-a-beginners-guide/>

[27] *Pro ASP.NET Core 3: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages* [online]. 8th ed. Berkeley, CA: Apress, 2020 [cit. 2022-05-08]. ISBN 9781484254394.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

SQL	Structured Query Language
MVC	Model View Controller
LINQ	Language Integrated Query
CSRF	Cross-site Request Forgery
XSS	Cross-site Scripting
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secured
PBKDF2	Password-Based Key Derivation Function 2
PDF	Portable Document Format
API	Application Programming Interface
CSS	Cascading Style Sheets
XML	Extensible Markup Language
SVG	Scalable Vector Graphics
ORM	Object-Relational Mapping
ERD	Entity Relationship Diagram
JSON	JavaScript Object Notation

SEZNAM OBRÁZKŮ

Obrázek 1 Diagram případů užití.....	28
Obrázek 2 Drátěný model registrace	42
Obrázek 3 Drátěný model záznamů.....	43
Obrázek 4 Drátěný model zaměstnanců	44
Obrázek 5 Drátěný model počasí.....	45
Obrázek 6 Ukázka třídy	47
Obrázek 7 Kontextové třídy.....	47
Obrázek 8 Propojení kódu s databází	47
Obrázek 9 Migrace tabulky.....	48
Obrázek 10 Přidání identity rolí.....	48
Obrázek 11 Kontrolér sklizně	49
Obrázek 12 Ukázka LINQ dotazu	50
Obrázek 13 Ukázka antiforgery tokenu	51
Obrázek 14 Registrační formulář.....	52
Obrázek 15 Ukázka upozornění.....	53
Obrázek 16 Navigační lišta.....	53
Obrázek 17 Titulní strana	54
Obrázek 18 Podrobnosti pracovního úkonu.....	54
Obrázek 19 Přidání komentáře	55
Obrázek 20 Zobrazení všech záznamů	56
Obrázek 21 Zobrazení sklepů v systému	57
Obrázek 22 ERD diagram.....	72
Obrázek 23 Databázový model.....	73

SEZNAM TABULEK

Tabulka 1. Poplatky za systém AgroWin	13
Tabulka 2. Poplatky za systém AgroWin	13
Tabulka 3. Poplatky za systém VineInfo	14
Tabulka 4. Poplatky za systém eVineyard.....	16
Tabulka 5. Scénář případu užití Registrace	29
Tabulka 6. Scénář případu užití Přihlášení	30
Tabulka 7. Scénář případu užití Zobrazení informací o sklizni.....	31
Tabulka 8. Scénáře případu užití Vytvoření záznamu.....	31
Tabulka 9. Scénář případu užití Úprava záznamu	32
Tabulka 10. Scénář případu užití Smazání záznamu	33
Tabulka 11. Scénář případu užití Filtrování dle záznamu	33
Tabulka 12. Scénář případu užití Zobrazení pracovních úkonů	34
Tabulka 13. Scénář případu užití Vytvoření pracovního úkonu.....	34
Tabulka 14. Scénář případu užití Zobrazení detailu	35
Tabulka 15. Scénář případu užití Smazání pracovního úkonu	35
Tabulka 16. Scénář případu užití Přidání komentáře.....	36
Tabulka 17. Scénář případu užití Zobrazení předpovědi o počasí.....	36
Tabulka 18. Scénáře případu užití Vytvoření PDF záznamu.....	37
Tabulka 19. Scénář případu užití Zaslání na e-mail	37
Tabulka 20. Scénář případu užití Stáhnutí do počítače	38
Tabulka 21. Scénář případu užití Zobrazení informací o zaměstnancích.....	38
Tabulka 22. Scénář případu užití Změna detailů	39
Tabulka 23. Testovací případ přímé navigace	59
Tabulka 24. Testovací případ pro validní přihlášení.	59
Tabulka 25. Testovací případ pro ne-validní přihlášení	60
Tabulka 26. Testovací případ pro zobrazení sklepů	60
Tabulka 27. Testovací případ pro vytvoření sklepa validními daty	61
Tabulka 28. Testovací případ pro vytvoření sklepa ne-validními daty	61

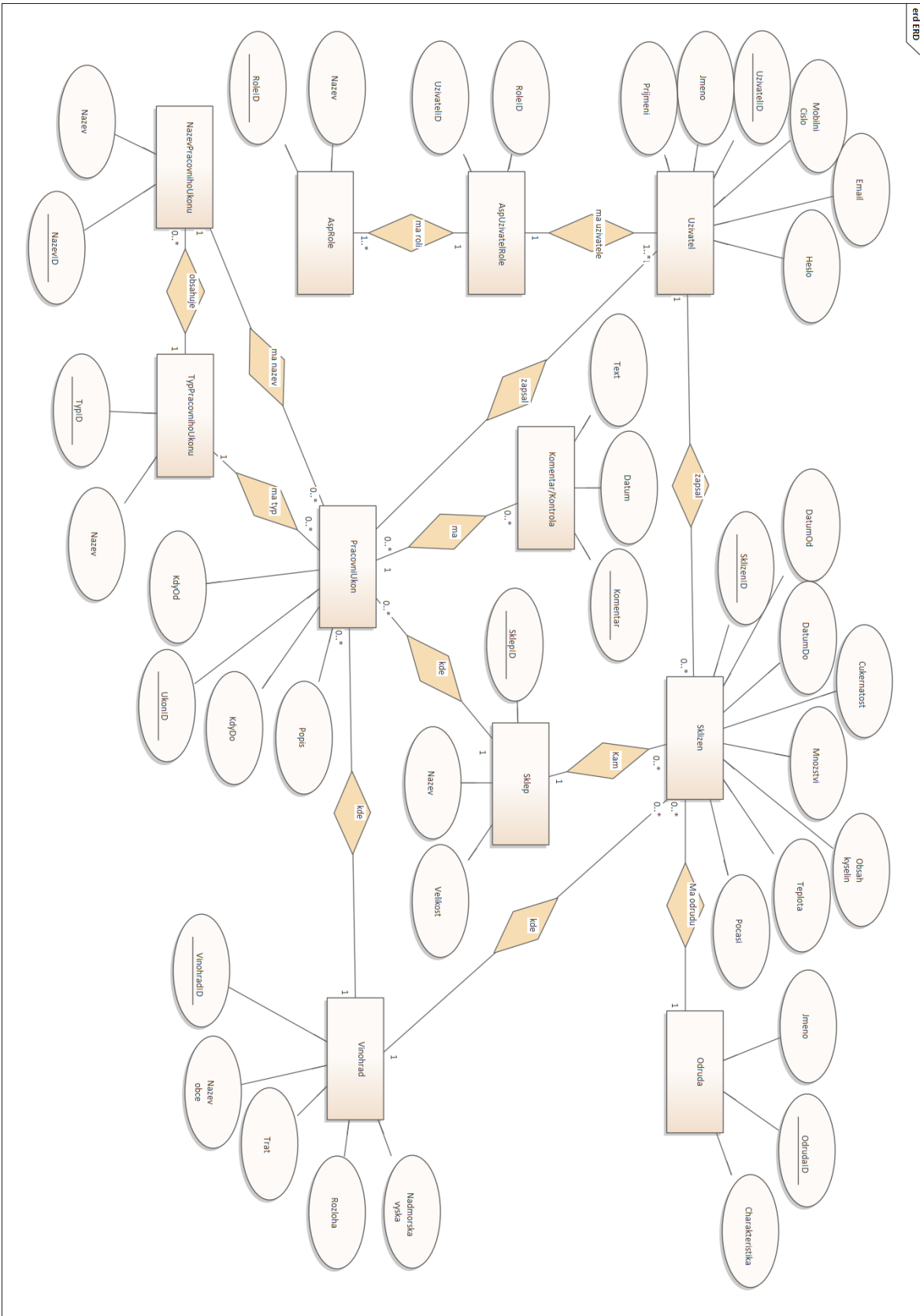
SEZNAM PŘÍLOH

P I. ERD

P II. ERDDataModeling

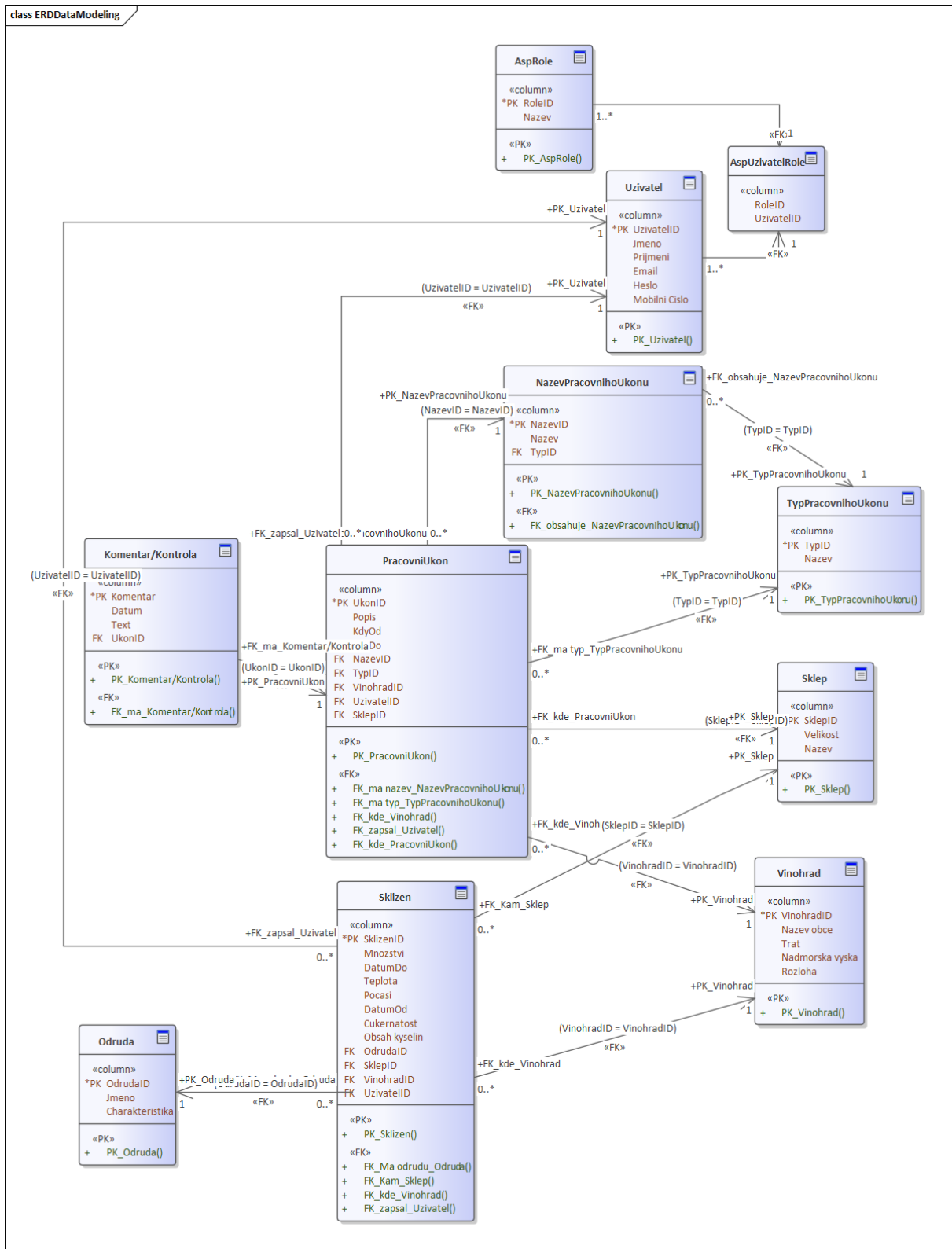
P III. CD s přílohami a zdrojovým kódem

PŘÍLOHA P I: ERD



Obrázek 22 ERD diagram

PŘÍLOHA P II. ERDDATAMODELING



Obrázek 23 Databázový model