



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

NÁVRH A IMPLEMENTACE REPORTINGOVÉHO MODULU PRO AUTOMATIZOVANÉ PILNICE

DESIGN AND IMPLEMENTATION OF A REPORTING MODULE FOR AUTOMATED SAWMILLS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jan Adam

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Lukáš Novák, Ph.D.

BRNO 2024

Zadání diplomové práce

Ústav:	Ústav informatiky
Student:	Bc. Jan Adam
Vedoucí práce:	Ing. Lukáš Novák, Ph.D.
Akademický rok:	2023/24
Studijní program:	Informační management

Garant studijního programu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává diplomovou práci s názvem:

Návrh a implementace reportingového modulu pro automatizované pilnice

Charakteristika problematiky úkolu:

Úvod
Vymezení problému a cíle práce
Teoretická východiska práce
Analýza problému a současné situace
Vlastní návrhy řešení, přínos návrhů řešení
Závěr
Seznam použité literatury
Přílohy

Cíle, kterých má být dosaženo:

Cílem diplomové práce je navrhnout a implementovat reportingový modul pro software užívaný v automatizovaných pilnicích.

Základní literární prameny:

BASL, Josef a BLAŽÍČEK, Roman, 2012. Podnikové informační systémy: podnik v informační společnosti. 3., aktualiz. a dopl. vyd. Management v informační společnosti. Praha: Grada. ISBN 978-80-247-4307-3.

GÁLA, Libor; POUR, Jan a ŠEDIVÁ, Zuzana, 2015. Podniková informatika: počítačové aplikace v podnikové a mezipodnikové praxi. 3., aktualizované vydání. Management v informační společnosti. Praha: Grada Publishing. ISBN 978-80-247-5457-4.

HANZELKOVÁ, Alena; KEŘKOVSKÝ, Miloslav a VYKYPĚL, Oldřich, 2017. Strategické řízení: teorie pro praxi. 3. přepracované vydání. C.H. Beck pro praxi. V Praze: C.H. Beck. ISBN 978-80-7400-637-1.

SCHWALBE, Kathy, 2011. Řízení projektů v IT: kompletní průvodce. Brno: Computer Press. ISBN 978-80-251-2882-4.

SODOMKA, Petr a KLČOVÁ, Hana, 2010. Informační systémy v podnikové praxi. 2., aktualiz. a rozš. vyd. Brno: Computer Press. ISBN 978-80-251-2878-7.

PROCHÁZKA, Jaroslav a KLIMEŠ, Cyril, 2011. Provozujte IT jinak: agilní a štíhlý provoz, podpora a údržba informačních systémů a IT služeb. Průvodce (Grada). Praha: Grada. ISBN 978-80-247-4137-6.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2023/24

V Brně dne 4.2.2024

L. S.

doc. Ing. Miloš Koch, CSc.
garant

doc. Ing. Vojtěch Bartoš, Ph.D.
děkan

Abstrakt

Tato diplomová práce se zabývá návrhem a implementací reportingového modulu SMDH Report pro software využívaný v automatizovaných pilnicích. Cílem bylo vyvinout modul, který umožňuje efektivní sledování, úpravu a sdílení dat z měřících stanic. Práce začíná teoretickým základem, kde jsou vysvětleny klíčové koncepty. Následuje analýza aktuálního stavu a potřeb, na jejímž základě bylo možné navrhnout aplikaci. Implementace zahrnovala integraci s existujícími systémy a testování nového modulu. Výsledkem je plně funkční aplikace, která je již nasazena v produkčním prostředí a přináší značné zlepšení v procesech zpracování dat ve firmách.

Klíčová slova

desktopová aplikace, smdh, reportingový modul, C#, winforms

Abstract

This thesis deals with the design and implementation of the SMDH Report reporting module for software used in automated sawmills. The aim was to develop a module that allows efficient monitoring, editing and sharing of data from measuring stations. The thesis starts with a theoretical background where key concepts are explained. This is followed by an analysis of the current state and the needs, based on which it was possible to design an application. Implementation included integration with existing systems and testing of the new module. The result is a fully functional application that has already been deployed in a production environment and brings significant improvements to the data processing in companies.

Key words

desktop application, smdh, reporting module, C#, winforms

Bibliografická citace

ADAM, Jan. *Návrh a implementace reportingového modulu pro automatizované pilnice* [online]. Brno, 2024 [cit. 2024-05-12]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/158852>. Diplomová práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Ing. Lukáš Novák, Ph.D.

Čestné prohlášení

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 12. 5. 2024

.....

Podpis autora

Poděkování

Ze všeho nejdříve bych rád poděkoval vedoucímu diplomové práce Ing. Lukáši Novákovi, Ph.D., za jeho vedení a cenné rady při tvorbě této práce. Zároveň také velký dík patří mé rodině za poskytnutou podporu během celého studia a mé přítelkyni, která mi byla oporou.

OBSAH

ÚVOD	11
VYMEZENÍ PROBLÉMU A CÍLE PRÁCE	12
1 TEORETICKÁ VÝCHODISKA PRÁCE	13
1.1 Automatizace a robotizace	13
1.1.1 RPA	14
1.1.2 Automatizace prostřednictvím vlastního softwaru.....	15
1.2 Industry 4.0	16
1.3 Životní cyklus softwaru	18
1.3.1 Vodopádový model	18
1.3.2 Agilní přístup.....	19
1.3.3 Iterativní přístup	20
1.3.4 Hybridní a kombinované přístupy	21
1.3.5 Práce v týmu.....	22
1.4 Vývojové prostředí a nástroje	23
1.4.1 Visual Studio	24
1.4.2 Git, GitHub, GitLab	25
1.5 Programovací jazyky	26
1.5.1 Typy programovacích jazyků.....	26
1.5.2 C#	29
1.5.3 SQL	30
1.5.4 Inno Setup	31
1.6 Možnosti vývoje desktopových aplikací	32
1.6.1 .NET Framework.....	32
1.6.2 .NET Core	32
1.6.3 WinForms.....	33
1.6.4 WPF	34
1.6.5 .NET MAUI	34
1.6.6 Avalonia	34
1.7 Databáze	35
1.7.1 SQLite	35
1.7.2 MySQL.....	35
1.7.3 Maria DB.....	36

1.8	Testování softwaru	36
1.8.1	Unit Testy.....	37
1.9	Analytické nástroje	38
1.9.1	SLEPT.....	38
1.9.2	Porterova analýza.....	39
1.9.3	Model 7S.....	39
1.9.4	SWOT	40
2	ANALÝZA PROBLÉMU A SOUČASNÉ SITUACE	41
2.1	Představení společnosti	41
2.2	Analýza vnějšího prostředí	42
2.2.1	SLEPT	42
2.2.2	Porterova analýza 5 sil	45
2.3	Analýza vnitřního prostředí	49
2.3.1	Model 7S.....	49
2.4	Sumarizace analýz.....	54
2.4.1	SWOT	54
2.5	Použité programovací jazyky, frameworky a knihovny	58
2.6	Databázová integrace a správa	59
2.7	Architektura a design aplikací	60
2.7.1	System měření a manipulace dřevní hmoty II (SMMDH2).....	60
2.7.2	System měření pro pásové dopravníky (SMPD)	60
2.7.3	System měření pro pásové pily (SMPP)	61
2.7.4	System pro měření a manipulaci pro omítací pily (SMMOP)	61
2.8	Testování, kvalita a správa verzí softwaru	61
3	VLASTNÍ NÁVRH ŘEŠENÍ	63
3.1	Požadavky a očekávání	63
3.2	Analýza rizik.....	64
3.2.1	Identifikace a ohodnocení rizik.....	65
3.2.2	Opatření k rizikům	70
3.3	Aplikace SMDH Report.....	73
3.3.1	ReportForm.....	74
3.3.2	Procházení a editace dat.....	80
3.3.3	Přehled skládek	89
3.3.4	Export a reportování dat.....	93

3.4	Integrace a nasazení nové verze	103
3.4.1	Integrace s existujícími aplikacemi	103
3.4.2	Nasazení aplikace, uvedení do provozu a správa verzí	104
3.5	Ekonomické zhodnocení.....	105
3.6	Přínosy řešení.....	106
3.7	Možné budoucí vylepšení	107
ZÁVĚR.....	108
SEZNAM POUŽITÉ LITERATURY	109
SEZNAM OBRÁZKŮ.....	116
SEZNAM TABULEK	118

ÚVOD

V dnešní době je automatizace a robotizace v průmyslu stále důležitější, což je dáno několika faktory. Jedním z hlavních důvodů je potřeba zvyšování efektivity a snižování nákladů provozu, kdy automatizace umožňuje rychlejší a přesnější výrobní procesy s menším množstvím odpadu. Dále automatizace přispívá ke zlepšení pracovních podmínek tím, že eliminuje nutnost lidské práce v nebezpečných nebo monotónních pracovních pozicích. S rostoucími požadavky na kvalitu a sledovatelnost výrobních procesů se stává automatizace nezbytnou pro zajištění konzistence a dodržování průmyslových standardů.

Zároveň, s rostoucí potřebou efektivně sbírat a analyzovat data z různých částí výrobního procesu se stává nezbytné mít k dispozici pokročilé reportovací nástroje. Tato diplomová práce se proto zaměřuje na návrh a implementaci reportingového modulu SMDH Report, který je určen pro software v automatizovaných pilnicích. Tento modul poskytne uživatelům možnost nejen sledovat data z měřících stanic, ale i je účinně analyzovat a sdílet.

Tento modul, vyvíjený ve spolupráci s firmou BJSW, umožňuje uživatelům efektivní sledování, úpravu a sdílení dat z měřících stanic, což přináší lepší využití shromážděných dat pro strategické rozhodování a optimalizaci firemních procesů.

Práce je strukturovaná do několika hlavních kapitol. První část pokrývá základní teoretické koncepty včetně automatizace, Industry 4.0 a životního cyklu softwaru. Druhá část se zaměřuje na představení společnosti, analýzu jejího vnitřního a vnějšího prostředí a následnou sumarizaci těchto analýz. Ve třetí části jsou specifikovány požadavky a očekávání projektu, následované detailním popisem aplikace SMDH Report a jejím způsobu nasazení a udržováním. Závěrečné kapitoly se věnují ekonomickému zhodnocení projektu, přínosům řešení a diskusi o možných budoucích vylepšeních.

VYMEZENÍ PROBLÉMU A CÍLE PRÁCE

Cílem této diplomové práce je analyzovat současný stav, navrhnout a implementovat reportingový modul pro software užívaný v automatizovaných pilnicích. Tento modul, označovaný jako SMDH Report, bude integrován do stávajícího systému SMDH a poskytne uživatelům možnost sledování, úpravy a exportu dat z měřících stanic. Zásadním požadavkem je, aby bylo možné data efektivně sdílet a exportovat, například do formátu PDF.

K docílení těchto cílů je nutné splnit následující body:

- provést analýzu trhu a vnitřního prostředí,
- navrhnout uživatelské rozhraní, které umožní efektivní interakci s daty,
- implementovat logiku aplikace zajišťující správu a zpracování dat,
- integrace nového modulu s existujícími aplikacemi v rámci projektu SMDH,
- provést testování funkcionality a uživatelské přívětivosti nově implementovaného modulu,
- analyzovat možné vylepšení procesů.

Diplomová práce se zaměřuje na vytvoření řešení, které zvýší produktivitu práce s daty v rámci aplikací pro automatizované pilnice a umožní uživatelům lépe využívat nasbíraná data pro další zpracování a rozhodování.

1 TEORETICKÁ VÝCHODISKA PRÁCE

Následující podkapitoly poskytují základní teoretické koncepty důležité pro vypracování vlastního návrhu řešení této diplomové práce. Jsou zde vysvětleny principy automatizace, použité programovací jazyky a vývojová prostředí, které jsou klíčové pro realizaci projektu. Dále jsou popsány databázové systémy a analytické nástroje, které byly použity. Tyto teoretické základy umožňují hlubší pochopení technického řešení představeného v dalších částech práce.

1.1 Automatizace a robotizace

Automatizace a robotizace představují nezbytné části současného průmyslového rozvoje, neboť značně přispívají k efektivnějšímu, bezpečnějšímu a kvalitnějšímu provozu výrobních procesů. Tyto technologie umožňují delegovat opakované a fyzicky náročné úkoly na systémy a roboty, což má za následek výrazné zlepšení pracovního prostředí a zvýšení produktivity. Tato kapitola se věnuje popisu základních principů automatizace a robotizace a jejich využití v konkrétních průmyslových sektorech, včetně pilnic pro zpracování dřeva.

Automatizace se vztahuje k aplikování kontrolních systémů, softwaru a jiných technologií pro vykonávání úloh, které nevyžadují lidský zásah. Robotizace, specifické odvětví automatizace, pak využívá roboty k vykonávání fyzických činností. Tyto technologie společně nabízejí řadu výhod, jako je například zvýšení přesnosti a konzistence v procesech, snížení rizika pracovních úrazů tím, že převezmou nebezpečné nebo monotónní úlohy, a celkové zvýšení efektivity práce. (1) (2)

V kontextu průmyslu zpracování dřeva, automatizace a robotizace znamenají posun v pořezech dřeva, třídění a zpracování kmenů. Díky moderním kontrolním systémům a robotické technice je možné minimalizovat množství odpadu a maximalizovat využití surovin. Tato technologická vylepšení nejen zvyšují produktivitu a rentabilitu, ale také podporují udržitelnější využívání dřevních zdrojů. (3)

Je však důležité si uvědomit, že i přes zjevné výhody mohou automatizace a robotizace přinést potenciální nevýhody. Vysoké počáteční investice do zavedení těchto technologií a potřeba specializované údržby a oprav mohou představovat překážky pro některé

podniky. Dále existují obavy ohledně potenciální ztráty pracovních míst pro lidi v důsledku nahrazení tradičních manuálních úkolů automatizovanými procesy.

Přesto automatizace a robotizace zůstávají nezbytnými součástmi moderního průmyslového vývoje, nabízejí cestu k větší efektivitě, bezpečnosti a kvalitě. V průmyslu zpracování dřeva, stejně jako v mnoha dalších odvětvích, tyto technologie představují klíčový prvek pro růst a udržení konkurenceschopnosti podniků. (4) (5)

1.1.1 RPA

Robotic Process Automation (*RPA*) je inovativní technologie, která umožňuje efektivní automatizaci opakovaných a rutinních úkolů v digitálních procesech. Použitím softwarových "robotů" RPA napodobuje interakce uživatelů s aplikacemi prostřednictvím uživatelského rozhraní, což umožňuje automatizaci široké škály procesů bez zásahu do základních systémových infrastruktur.

Mezi hlavní přínosy RPA patří výrazné zvýšení produktivity, kdy roboti přebírají úkoly, čímž se uvolňuje kapacita zaměstnanců pro zvládání komplexnějších úkolů. Dále RPA minimalizuje možnost lidských chyb, zvyšuje efektivitu pracovních postupů, snižuje provozní náklady a zlepšuje spokojenost zaměstnanců tím, že eliminuje repetitivní činnosti. Využitím RPA pro sběr a analýzu dat dochází ke zvýšení dostupnosti a efektivního využívání zásadních informací.

Ačkoliv RPA přináší mnoho výhod, je třeba si být vědom jejích omezení. Počáteční implementace a vývoj řešení RPA může vyžadovat značné investice a odborné znalosti. Pro úkoly vyžadující komplexní rozhodování nebo kreativní myšlení se RPA může jevit jako méně účinná a existuje riziko ztráty pracovních míst v důsledku automatizace. Také je nutné dodržovat etické standardy, především v oblasti transparentnosti automatizovaných rozhodnutí.

RPA nachází uplatnění v širokém spektru aplikací, od zpracování a analýzy dat, přes vkládání dat do systémů, až po automatizaci komunikace se zákazníky a optimalizaci back-office procesů. Díky své schopnosti neinvazivně integrovat s existujícími systémy umožňuje RPA rychlou implementaci s minimálními technickými požadavky.

V kontextu automatizovaných pilnic nebo širšího dřevozpracujícího průmyslu může RPA optimalizovat a zefektivnit celou řadu procesů. Například, RPA může být využita pro

automatizaci zadávání dat do systémů správy inventáře, což minimalizuje chyby a zvyšuje efektivitu správy zásob surovin a hotových produktů. Další možnou aplikací je automatizace procesů plánování údržby strojů, kde RPA může analyzovat data ze senzorů a určit optimální časování údržby, což zvyšuje produktivitu a snižuje neplánované výpadky.

RPA může také efektivně automatizovat sledování kvality výrobních procesů tím, že shromažďuje a analyzuje data z výrobních linek, což umožňuje rychlé identifikování a řešení problémů dříve, než ovlivní kvalitu produktu. V oblasti logistiky a distribuce může RPA automatizovat zadávání a sledování objednávek, což zlepšuje přesnost a efektivitu distribučních procesů.

Výběr správné RPA platformy a úzká spolupráce s expertem na implementaci může výrazně zvýšit účinnost. Při zavádění RPA je nezbytné důkladně posoudit, zda je technologie vhodná pro specifické potřeby organizace a zda přínosy převažují nad potenciálními nevýhodami. (6) (7) (8)

1.1.2 Automatizace prostřednictvím vlastního softwaru

Automatizace pomocí vlastního softwaru nabízí organizacím příležitost k přizpůsobení technologických řešení jejich specifickým potřebám, což vede k efektivnější správě procesů a zvýšení produktivity napříč různými sektory. Například ve dřevozpracujícím průmyslu, kterým se tato práce zabývá, mohou najít uplatnění aplikace pro automatizované monitorování zásob nebo řízení pořezů a třídění dřeva. Zvýšená efektivita, minimalizace chyb, lepší správa dat a adaptabilita softwaru k unikátním potřebám podniku jsou základními přínosy, které umožňují organizacím optimalizovat využití materiálů a zpracovávat objednávky s vyšší přesností. Tyto výhody nejenže podporují snížení nákladů na pracovní sílu, ale také zvyšují konkurenceschopnost a inovativní potenciál v současném dynamickém průmyslovém prostředí.

Pro úspěšný vývoj softwaru je zásadní vhodně zvolit technologie a programovací přístupy, které odpovídají typu plánované aplikace a jejím specifickým požadavkům. Vývojáři mají k dispozici širokou škálu možností, které umožňují přizpůsobit software pro desktopové, webové, nebo multiplatformní aplikace. Každý z těchto typů aplikací vyžaduje odlišné technické řešení a přístup. Desktopové aplikace často vyžadují

hlubokou integraci s operačním systémem a využívají výhod nativního prostředí pro vysoký výkon a plnou funkčnost. Webové aplikace na druhou stranu poskytují univerzálnost a snadný přístup prostřednictvím webového prohlížeče, což eliminuje potřebu instalace na koncovém zařízení. Multiplatformní vývoj pak umožňuje s jedním kódovým základem cílit na více platform, což je efektivní řešení pro dosažení širšího publika bez nutnosti vývoje více samostatných aplikací. Správná volba technologií a architektury softwaru je důležitá nejen pro efektivitu vývoje a údržby, ale také pro dosažení optimálního výkonu, zabezpečení a schopnosti integrace s existujícími systémy a platformami. Díky rozvoji technologií, jako je umělá inteligence a strojové učení, se otevírají nové možnosti pro automatizaci v nejrůznějších procesech. Příkladem takového užití může být rozpoznávání tvarů kmenů umělou inteligencí a jejich případným vyřazením, aby se zamezilo zaseknutí špatných kmenů ve strojích.

V kontrastu s robotickou procesní automatizací, která se zaměřuje na simulaci lidských akcí pro vykonávání úloh, poskytuje automatizace prostřednictvím vlastního softwaru více přizpůsobených a integrovaných řešení. Tím se stává klíčovým faktorem pro udržení konkurenceschopnosti a podporu inovací v dynamickém průmyslovém prostředí, zdůrazňující význam takové automatizace pro modernizaci operací a zvyšování produktivity napříč odvětvími. (9) (10) (11)

1.2 Industry 4.0

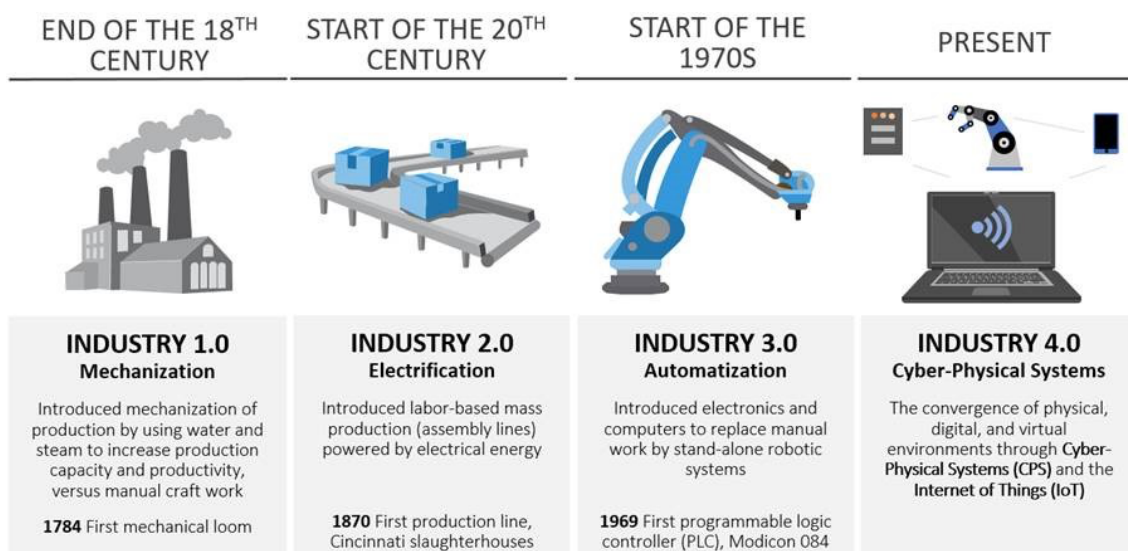
Průmysl 4.0, známý také jako Čtvrtá průmyslová revoluce, signalizuje zásadní proměnu v průmyslovém sektoru. Tato proměna je založena na využití pokročilých technologií. Charakteristickým rysem této éry je hluboká integrace informačních a komunikačních technologií do výrobních procesů. Tím překračuje dosavadní fáze mechanizace, masové produkce a automatizace. Rozlišovacími prvky Průmyslu 4.0 jsou propojení strojů, zařízení a senzorů prostřednictvím internetu věcí (*IoT*). Toto propojení umožňuje sběr a analýzu dat v reálném čase. Integrace kybernetických fyzických systémů, cloud computingu a aplikace využívající umělou inteligenci otevírá cestu k optimalizaci a automatizaci úkolů. (12)

Přechod do éry Průmyslu 4.0 s sebou přináší značné změny pro podniky. Mezi ty patří další zvýšení produktivity a efektivity, stejně jako zlepšení flexibility. V důsledku

propojenosti systémů mohou firmy dosahovat vyšší kvality a bezpečnosti výrobních procesů. Rovněž se naskytují možnosti pro vznik nových obchodních modelů založených na analýze a využití dat. Současně však Průmysl 4.0 přináší nové výzvy, včetně rizik v oblasti kybernetické bezpečnosti. Je také nutné zaměřit se na rekvalifikaci pracovní síly a na udržování etických a sociálních standardů v měnícím se prostředí.

V porovnání s Průmyslem 3.0, který byl charakterizován automatizací jednotlivých strojů a procesů, Průmysl 4.0 posunuje paradigma k propojení celých výrobních systémů. Tento posun zahrnuje nejen technologický pokrok, ale i změnu v přístupu k řízení a využití dat. Zatímco Průmysl 3.0 se soustředil převážně na monitorování a reporting, Průmysl 4.0 využívá data pro hlubší analýzu a optimalizaci procesů.

Pro úspěšnou implementaci Průmyslu 4.0, je velice důležité strategické plánování a investice do nových technologií. Tento krok umožňuje firmám zvýšit konkurenceschopnost, efektivitu a inovační potenciál. Zároveň je důležité řešit s tím spojené výzvy a rizika. Průmysl 4.0 se tedy stává základem pro budoucí inovace v průmyslovém sektoru, otevírající nové možnosti založené na inteligentních systémech, propojenosti a digitalizaci. (13) (14)



Obrázek 1.1 - Přehled čtyř průmyslových revolucí
(Zdroj: (14))

1.3 Životní cyklus softwaru

Životní cyklus vývoje softwaru (SDLC – *Software Development Lifecycle*) slouží jako základní proces pro efektivní vytváření softwaru, který organizuje celý projekt od ideje až po uvedení produktu na trh a jeho následnou údržbu. Tato kapitola přibližuje SDLC, včetně jeho principů, hlavních funkcí a významného dopadu na tvorbu softwaru.

Pochopení a aplikování SDLC umožňuje týmům optimalizovat plánování, zabezpečit hladký průběh vývoje a v konečném důsledku přináší produkty odpovídající nejvyšším standardům kvality a přesným očekáváním klientů. Jak bylo zmíněno výše, SDLC zahrnuje systematickou metodiku pro řízení celé cesty vývoje softwaru, která působí jako plán rozdělující vývojový proces do spravovatelných fází s jasnými cíli a výstupy. Tento strukturovaný přístup podporuje jasnost, organizaci a kontrolu, minimalizuje rizika a zajišťuje soulad s cíli projektu.

Základní záměr SDLC spočívá v optimalizaci procesu vývoje tak, aby výsledný software nejen vyhověl požadavkům a očekáváním zákazníků, ale také byl vyvinut s minimálními riziky a náklady. Efektivní aplikace SDLC umožňuje týmovou spolupráci a předcházení překážkám díky včasnému rozpoznání a adresaci možných komplikací, čímž se zajišťuje hladká funkcionalita a vysoká kvalita produktu.

Konkrétní modely a implementace SDLC mohou přistupovat k vývoji softwaru odlišně, avšak obvykle zahrnují stěžejní fáze, a to od plánování a analýzy požadavků, přes návrh a architekturu, vývoj, testování, až po nasazení a údržbu. Každá z těchto fází má svoji jedinečnou roli, která přispívá k zajištění kvality, optimalizaci procesů a minimalizaci rizik a nákladů. Zároveň prioritizace spolupráce a komunikace mezi všemi zainteresovanými stranami napomáhá dosažení projektových cílů a splnění uživatelských požadavků s větší přesností a lepšími výstupy. (12) (15) (16)

1.3.1 Vodopádový model

Vodopádový model, zavedený v roce 1970 Dr. Winstonem W. Roycem, patří mezi nejstarší a tradiční metody softwarového vývoje. Tento model se vyznačuje lineární strukturou, kde fáze projektu postupují logicky po sobě bez možnosti vracet se

k předchozím etapám. Ačkoliv je vnímán jako zastaralý ve srovnání s agilními metodami, stále nalézá uplatnění v projektech vyžadujících pevnou strukturu.

Vývoj softwaru dle vodopádového modelu začíná analýzou a shromažďováním požadavků, následuje návrh, programování a implementace systému. Klíčové jsou fáze integrace a testování, které předchází finálnímu nasazení do provozu. Údržba systému pak zajišťuje jeho bezproblémový chod. Strukturovanost modelu umožňuje efektivní plánování a monitorování projektu, což je výhodné pro projekty s pevnými požadavky a malou pravděpodobností změn.

Každá fáze projektu produkuje dokumentaci a výstupy, které slouží pro další kroky, čímž zajišťuje průběžnou sledovatelnost a předvídatelnost celého projektu. Nicméně, hlavní slabiny modelu spočívají v jeho nepružnosti a vysokém riziku nákladných oprav v případě problémů, které se objeví ve pozdějších fázích. Pozdní testování a omezená zpětná vazba od uživatelů také představují problematické aspekty.

Přestože má vodopádový model svá omezení, jeho schopnost poskytnout pevné směřování a strukturu zůstává ceněna v projektech s jasně definovanými a stabilními požadavky. Správně implementovaný, vodopádový model může přinést silné a kvalitní softwarové řešení. (17) (18) (19)

1.3.2 Agilní přístup

Agilní přístup transformuje tradiční procesy vývoje softwaru tím, že podporuje flexibilní a iterativní pracovní rámec, který reaguje na proměnlivost požadavků a podněty z trhu. Jeho užití spočívá v rozdělení projektu na krátké, spravovatelné segmenty známé jako sprinty, během nichž týmy pracují na vytvoření či vylepšení specifických částí softwaru, což umožňuje průběžnou adaptaci a inovaci.

Jednou z nejrozšířenějších implementací agilního modelu je Scrum, metodika, která se řídí pevně stanovenými rolemi, rituály a artefakty pro zajištění plynulého vývoje produktu. Týmy organizované podle Scrumu pracují ve sprintech, což jsou typicky dvoutýdenní až měsíční cykly, ve kterých se vyvíjí nové funkce softwaru. Každý sprint začíná plánovacím setkáním, během něhož se stanovují cíle a přiřazují úkoly, a končí retrospektivou, kde se zhodnocuje průběh práce a hledají se možnosti pro další zlepšení.

Scrum dále zdůrazňuje význam denních stand-upů, krátkých koordinačních schůzek, které pomáhají udržet tým na stejné vlně a rychle řešit případné problémy. Každý sprint směřuje k vytvoření hmatatelného, funkčního produktu nebo jeho části, který lze okamžitě testovat a představit zákazníkovi. (20) (21) (22)

Agilní přístup nabízí vývojovým týmům zvýšenou pružnost, která umožňuje svižnější reakce na změny, což je nezbytné pro udržení kroku s dynamickým tržním prostředím. Klienti oceňují možnost aktivního zapojení do procesu a průběžnou kontrolu nad vývojem produktu, což vede ke zvýšení jejich spokojenosti. Týmy jsou produktivnější a více motivované, jelikož jsou více samostatné a pro každý sprint mají jasné cíle.

Avšak, agilní metody, obzvláště Scrum, si kladou vysoké nároky na komunikační dovednosti a seberegulaci v rámci týmu. Pro velké a složité organizace může být výzvou efektivně řídit agilní procesy bez silného vedení a koordinace. Úspěch agilního přístupu tak závisí na zkušenostech a dovednostech projektového manažera, často označovaného jako scrum master, který musí být schopen udržet tým v souladu s principy agilit a směřovat projekt k cíli. (23) (24)

1.3.3 Iterativní přístup

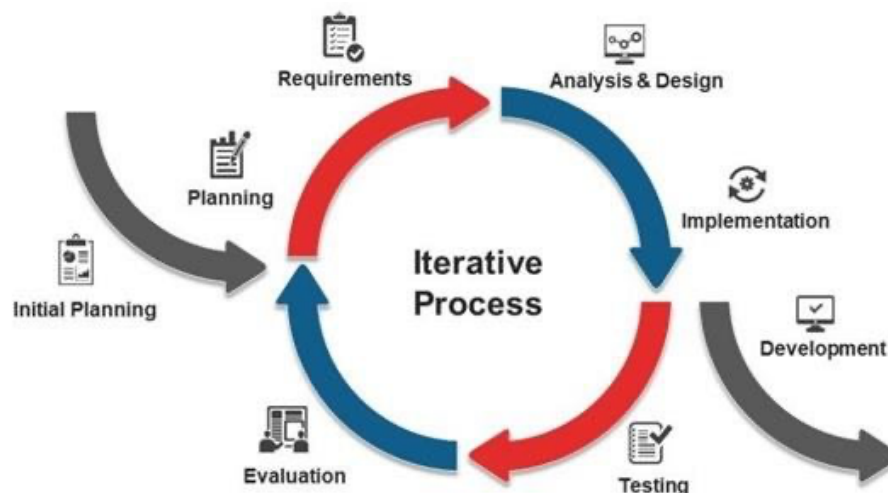
Iterativní model je považován za jeden z nejpobulárnějších a nejpoužívanějších přístupů v rámci SDLC. V tomto modelu je proces vývoje zahájen s implementací malé části softwarových požadavků, což představuje první krok k postupnému rozvíjení a zdokonalování celkového řešení. Po každé iteraci neboli cyklu vývoje, je výsledná část produktu pečlivě přezkoumána a otestována. To umožňuje identifikovat a začlenit nové požadavky, což je proces, který se opakuje v pravidelných cyklech. Tyto cykly pak probíhají kontinuálně, dokud není systém zcela implementován. Iterativní model je zvláště vhodný pro projekty, které jsou charakteristické neúplně definovanými požadavky na začátku, nebo když jsou vývojáři postaveni před úkol integrace nových technologií, či když projekt čelí časovým omezením, která brání klasickému postupnému vývoji.

Hlavní silou iterativního modelu je jeho schopnost flexibilně reagovat na proměnlivé požadavky a poskytovat prostor pro brzkou zpětnou vazbu a identifikaci rizik. Tato adaptabilita umožňuje nejen zlepšit kvalitu finálního produktu, ale také efektivněji řídit

náklady na vývoj, neboť umožňuje snadné provádění změn v raných fázích vývoje, což snižuje potřebu nákladných úprav v pozdějších fázích. Rychlé prototypování a průběžné testování v každé fázi iterace napomáhají včasnému odhalení problémů a zvyšují kvalitu vývojového procesu.

Nicméně, iterativní model přichází i s určitými výzvami. Jeho dynamický charakter může znesnadnit sledování pokroku a vyžaduje pečlivou správu zdrojů. Složitější koordinace fází vývoje a potenciální obtíže při definování architektury systému mohou vyústit ve zvýšenou náročnost na projektový management. Navíc, časté změny požadavků mohou vést k nutnosti angažovat vysoce kvalifikované specialisty pro analýzu rizik, což přináší dodatečné náklady.

Celkově iterativní model nabízí efektivní rámec pro projekty vyžadující adaptabilitu a schopnost reagovat na změny, a to i přes některé jeho výzvy. Jeho aplikace v praxi vyžaduje strategické plánování a správné projektové řízení, aby byly zajištěny optimální výsledky a byla naplněna očekávání všech zainteresovaných stran. (20) (25) (26)



Obrázek 1.2 - Iterativní model SDLC
(Zdroj: (26))

1.3.4 Hybridní a kombinované přístupy

V současné době se v praxi stále častěji používá přístup, kdy nejsou vývojové týmy omezeny pouze na jeden rigidní model řízení softwarového vývoje, ale naopak využívají kombinace různých metodik nebo si vybrané modely upravují tak, aby co nejlépe odpovídaly specifikám a potřebám konkrétního projektu. Tento trend vede ke zjištění, že

neexistuje univerzální řešení vhodné pro všechny typy projektů. Flexibilní a adaptabilní řešení mohou přinést lepší výsledky a efektivnější řízení procesu vývoje.

Hybridní metody, spojující prvky klasických postupů jako vodopádový model s agilními strategiemi typu Scrum či Kanban, nabízejí týmům možnost těžit z předností obou přístupů. Projekt může odstartovat důkladným vyhodnocením a přípravou pomocí tradičního modelu, aby následně přešel k adaptabilnímu agilnímu stylu, reagujícímu na dynamiku potřeb a preferencí klienta.

Tato řešení umožňují efektivnější reakci na překvapivé situace, zjednodušení řešení výzev a tvorbu softwaru, který lépe reflektuje aktuální tržní a zákaznické požadavky. Zároveň podporují efektivnější alokaci zdrojů, umožňující týmům rychle se přizpůsobit bez zbytečného zdržování pevnými pravidly.

Úspěšná implementace těchto inovativních přístupů si žádá výbornou týmovou komunikaci, schopnost rychlého přechodu na nové metody a efektivní vedení schopné sjednotit práci podle různých metodologických rámců. Zásadní roli hraje i pečlivá dokumentace, která odráží tuto unikátní kombinaci přístupů a zajišťuje srozumitelnost projektových kroků pro všechny účastníky. (27)

Příkladem hybridního přístupu v praxi může být spirálový model, který rozšiřuje lineární strukturu vodopádového modelu o iterativní cykly plánování, rizikové analýzy, vývoje a hodnocení. Tento model poskytuje rámec pro komplexní řízení projektů s vysokým důrazem na identifikaci a minimalizaci rizik v každé fázi vývoje. Jeho cyklická povaha umožňuje projektům postupně se vyvíjet a začleňovat nově objevené požadavky nebo technologické výzvy, což z něj činí ideální volbu pro projekty, kde jsou počáteční specifikace nejasné nebo se očekává jejich změna. Spirálový model tak přináší možnost hlouběji porozumět projektu a zvyšuje šance na úspěch tím, že podporuje průběžné testování a optimalizaci. Využití tohoto modelu však vyžaduje pečlivé řízení a schopnost týmu efektivně zvládat proces hodnocení a přizpůsobování. (17) (28)

1.3.5 Práce v týmu

Vývoj softwaru je komplexní a více fázový proces, který vyžaduje úzkou spolupráci mezi různými členy týmu. Úspěch projektu se opírá nejen o technické dovednosti, ale také o schopnost efektivní týmové práce.

Důležitost kvalitní komunikace a vzájemné důvěry v týmu nelze podcenit. Efektivní komunikace pomáhá překonávat překážky a vyjasňuje očekávání, zatímco důvěra podporuje otevřenou výměnu názorů a myšlenek. Tým, který spolupracuje na základě jasně definovaných cílů a má jasné rozdělení úkolů, je více motivovaný a zaměřený na dosažení společného cíle.

Flexibilita a přizpůsobivost jsou vlastnosti, které týmu umožňují rychle reagovat na nečekané změny nebo výzvy během vývoje. Schopnost týmu adaptovat se na nové informace nebo změněné požadavky může být rozhodující pro úspěch projektu.

Zapojení týmu do rozhodovacích procesů a podpora osobního i profesního růstu jednotlivců jsou dalšími aspekty, které přispívají k úspěšné týmové spolupráci. Když se členové týmu cítí hodnotní a vidí možnosti pro svůj růst, jsou více angažovaní a připraveni přinášet inovativní řešení.

Klíčem k úspěšnému softwarovému projektu tedy není pouze výběr správného vývojového modelu, ale také budování silného a kooperativního týmu. Silná týmová kultura, která podporuje spolupráci, otevřenou komunikaci a vzájemnou podporu, je velice důležitá pro dosažení dobrých výsledků v jakémkoliv projektu. (20) (29)

1.4 Vývojové prostředí a nástroje

Tato kapitola poskytuje obecný přehled o významu integrovaných vývojových prostředí (IDE) a systémů pro správu verzí v kontextu softwarového inženýrství. Zatímco teoreticky je možné vyvíjet software pouze s použitím základního textového editoru a manuálně kompilovat a spouštět kód, v praxi IDE poskytují nástroje a funkce, které tento proces značně zjednodušují a automatizují. IDE centralizují vývojové úkoly, jako je psaní, testování a ladění kódu, a nabízejí vývojářům užitečné funkce, jako jsou syntaktické zvýraznění, automatické doplňování kódu a integrované debuggery, což zvyšuje efektivitu a snižuje možnost chyb. (30)

Podobně systémy pro správu verzí umožňují vývojářům efektivně sledovat a spravovat změny v kódu, což je zásadní pro kolaboraci v týmových projektech. Tyto systémy nabízejí možnost vytvářet zálohy různých verzí kódu a snadno přecházet mezi nimi, což umožňuje bezpečné experimentování s novými funkcemi bez rizika ztráty stávajícího práce. (31)

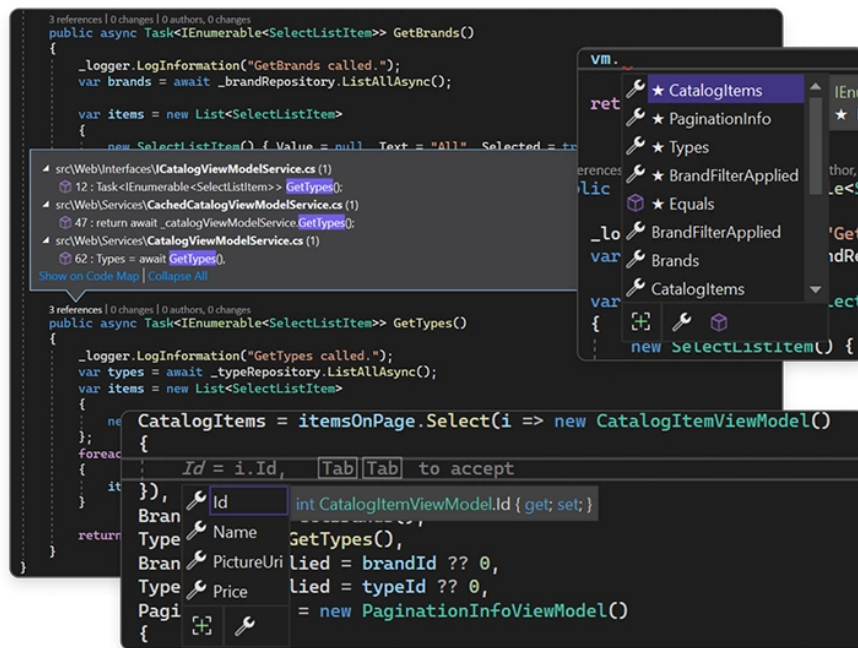
1.4.1 Visual Studio

Visual Studio od společnosti Microsoft je komplexní IDE, které je určené pro vývojáře softwaru pracující na aplikacích pro širokou škálu platform, včetně Windows, Android, iOS, webu a cloudu. Nabízí rozsáhlé nástroje a služby, které usnadňují vytváření, ladění, testování a nasazování aplikací. Díky pokročilým diagnostickým a ladícím funkcím, včetně integrovaného debuggeru a profilovacích nástrojů, mohou vývojáři rychle identifikovat problémy, optimalizovat výkon a zvyšovat stabilitu aplikací, což je klíčové pro projekty různého rozsahu a typu.

Visual Studio podporuje řadu programovacích jazyků, včetně C#, Visual Basic.NET, C++, a JavaScript, to umožňuje vybrat si jazyk, který nejlépe vyhovuje potřebám každého projektu. Kromě toho IDE nabízí integraci s různými frameworky a knihovnamí, jako jsou .NET Framework, .NET Core a Entity Framework, které jsou podrobněji přiblíženy v následujících kapitolách.

Mezi další populární IDE patří například Eclipse a IntelliJ IDEA pro Javu či PyCharm pro Python, přičemž každé z nich má své specifické zaměření a sadu funkcí.

Visual Studio je dostupné v několika verzích, od bezplatné Community verze, která je ideální pro studenty a malé týmy, po Professional a Enterprise edice určené pro rozsáhlejší projekty a organizace. Každá verze je navržena tak, aby vyhovovala specifickým potřebám vývojářů, přičemž Community verze představuje významnou výhodu díky své dostupnosti zdarma, na rozdíl od některých konkurenčních IDE, která jsou dostupná pouze za poplatek. (32)



Obrázek 1.3 - Vzhled a možnosti programu Visual Studio
(Zdroj: (32))

1.4.2 Git, GitHub, GitLab

Git, GitHub a GitLab představují základ pro moderní správu verzí a spolupráci v softwarovém vývoji. Tyto nástroje jsou nezbytné pro efektivní sledování změn, revizi kódu a správu projektů, což zvyšuje transparentnost a usnadňuje spolupráci mezi členy týmu.

Git je distribuovaný systém správy verzí, který umožňuje vývojářům ukládat snímky (*snapshoty*) svého kódu v čase. Nabízí flexibilitu v práci s různými verzemi kódu a podporuje paralelní vývoj, což značně zlepšuje produktivitu a efektivitu.

GitHub je cloudová platforma, která rozšiřuje možnosti Gitu o sociální a kolaborativní prvky, jako jsou pull requesty a code reviews. Umožňuje vývojářům sdílet kód a spolupracovat na projektech s lidmi z celého světa. GitHub také poskytuje nástroje pro sledování problémů, projektové plánování a integraci s dalšími službami. (33)

GitLab nabízí srovnatelné funkce jako GitHub, ale s jedinečnou vlastností – a to možností provozu na vlastním serveru. Tato možnost dává organizacím kontrolu nad svými daty a umožňuje přizpůsobení infrastruktury podle vlastních bezpečnostních politik a potřeb. GitLab je tedy ideální volbou pro týmy, které vyžadují vysoký stupeň přizpůsobení

a kontrolu, a nabízí integrované CI/CD pipeline, což zjednodušuje automatizaci testování a nasazování aplikací. (34)

Výběr mezi GitHubem a GitLabem závisí na konkrétních potřebách projektu a týmu. Zatímco GitHub je ideální pro projekty hledající širokou komunitní podporu a snadnou integraci s externími aplikacemi, GitLab poskytuje větší flexibilitu a kontrolu pro týmy, které preferují hostování na vlastním serveru a potřebují rozsáhlé CI/CD nástroje. (35) (36)

1.5 Programovací jazyky

Programovací jazyky lze chápat jako ucelený soubor instrukcí, které programátor vytváří pro řízení počítače při plnění specifických úkolů. Tyto pokyny jsou formulovány ve strukturovaném kódu, který je určen sémantikou a syntaxí daného jazyka. Následně je kód, zapsaný v daném jazyce přeložen nebo interpretován do strojového kódu – posloupnosti instrukcí, které jsou srozumitelné pro procesor počítače. (37) (38)

Nicméně je důležité si uvědomit, že pojem „programovací jazyky“ má abstraktní charakter a odborníci jej interpretují různými způsoby. Obecně však všechny definice souhlasí s tím, že tyto jazyky převádějí komplexní myšlenky a algoritmy do kódu, jenž je následně počítačem zpracováván. Samozřejmě je možné zapisovat instrukce přímo v kódu pro procesor, nicméně pro lidi je toto rozhraní nepraktické a zdlouhavé pro efektivní programování. Aby lidé urychlili a zefektivnili vytváření softwaru, využívají právě různých programovacích jazyků, což zároveň umožňuje využívat různé techniky a přístupy při řešení problémů. Současně překladače zajišťují, že díky mnoha různým optimalizacím jsou zdroje počítače využity co nejefektivněji. (39)

1.5.1 Typy programovacích jazyků

V dnešní době existuje široké spektrum programovacích jazyků, z nichž každý přináší jedinečné vlastnosti a specifické oblasti použití. Toto umožňuje programátorům vybírat tu nejlepší možnost pro konkrétní úkol nebo projekt. Místo zaměření na detailní znalost jednotlivých jazyků by se programátoři měli soustředit na osvojení schopnosti programovat a algoritmičky uvažovat. Správný výběr programovacího jazyka může být rozhodující pro úspěch projektu, neboť různé jazyky jsou optimalizovány pro specifické

cíle – od rychlého vývoje po dosažení maximálního výkonu nebo aplikace v určitých odvětvích. (40)

- **Procedurální programovací jazyky**

Procedurální jazyky postupují podle sekvence příkazů nebo instrukcí, aby dosáhly požadovaného výsledku. Každá sada kroků je známa jako procedura, a program napsaný v těchto jazycích obsahuje jednu nebo více procedur. Procedurální jazyky jsou často základní zkušeností juniorních vývojářů. Mezi nejpoužívanější procedurální jazyky se řadí C, Pascal, nebo BASIC. (41)

- **Funkcionální programovací jazyky**

Funkcionální programování je paradigma, ve které se vývojáři soustředí na definování výsledků funkcí místo na postup kroků k jejich dosažení. Jedná se o deklarativní přístup, kde se tedy hlavní důraz klade na „co řešit“ namísto „jak to řešit“. V tomto paradigmatu jsou výrazy upřednostňovány před příkazy, což znamená, že se zaměřují na generování hodnot místo na vykonávání akcí. Základem funkcionálního programování je lambda kalkulus Alonza Churcha, který poskytuje rámec pro pochopení funkcí a jejich aplikací a je považován za jeden z nejjednodušších programovacích modelů. Tento přístup je klíčový pro jazyky jako Haskell, Scala nebo F#, které jsou oblíbené pro aplikace vyžadující vysokou úroveň abstrakce a neměnnost výsledků. (42)

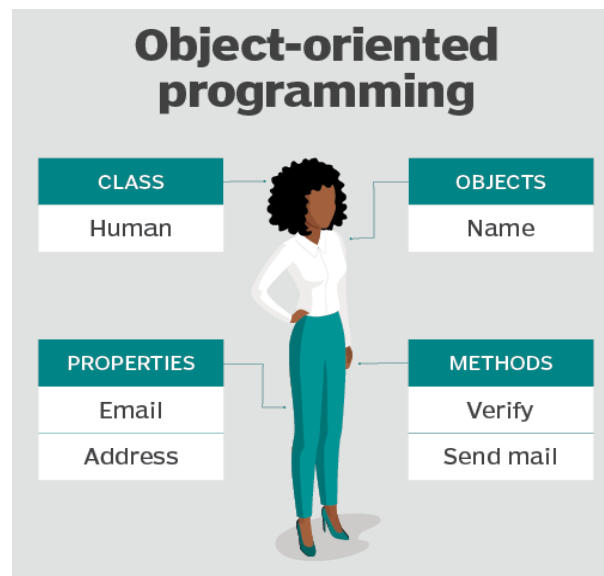
- **Objektově orientované programovací jazyky (OOP)**

Objektově orientované programování neboli OOP, je programovací paradigma, které se zaměřuje na objekty jako základní jednotky softwarového vývoje. Podle Stroustrupa, autora článku „*What is object-oriented programming?*“, je OOP definováno právě tímto zaměřením na objekty, které jsou reprezentacemi reálných věcí nebo konceptů. Každý objekt v OOP obsahuje soubor vlastností a chování, reprezentovaných daty a funkcemi, které spolu úzce souvisejí.

OOP se vyznačuje třemi hlavními principy, a to zapouzdřením, dědičností a polymorfismem. Zapouzdření skrývá interní data objektu a umožňuje interakci s objektem pouze prostřednictvím definovaného rozhraní. Dědičnost umožňuje objektům zdědit vlastnosti a chování od jiných objektů, což vede k větší efektivitě a snadnějšímu

znovupoužití kódu. Polymorfismus dovoluje objektům různých tříd sdílet stejné rozhraní, ale implementovat je různými způsoby, což zvyšuje flexibilitu a rozšiřitelnost softwaru.

Stroustrup zdůrazňuje, že OOP poskytuje schopnost modelovat reálný svět, což je zásadní pro vývoj aplikací, které musí interagovat s reálnými objekty nebo koncepty. OOP také usnadňuje údržbu a rozšiřování softwaru, což je důležité pro dlouhodobě udržitelný vývoj. Tato kombinace schopnosti modelování, snadné údržby a možnosti recyklace kódu činí OOP cenným nástrojem v rukou softwarových vývojářů. Mezi jazyky podporující objektově orientované paradigma se řadí například C#, Java nebo Python. (43) (44)



Obrázek 1.4 - Příklad struktury objektu používané v OOP
(Zdroj: (44))

- **Logické programovací jazyky**

V rámci programovacích paradigmat se logické programovací jazyky odlišují svým způsobem definování cílů programu skrze logická pravidla, nikoli přesným popisem postupu jejich provádění. Toto umožňuje vyjadřovat řešení problémů více abstraktně, což vede k lepší pochopitelnosti a jednoduššímu spravování kódu. Příkladem těchto jazyků jsou Prolog a Datalog, které jsou široce využívány v oblastech jako umělá inteligence nebo databázové systémy. Ačkoliv logické jazyky nemusí být vždy optimální z hlediska rychlosti výpočtu a mohou představovat výzvy v implementaci některých typů algoritmů, jejich přístup k řešení komplexních problémů je nepopíratelně silný a významný v kontextu pokročilého softwarového vývoje. (45)

- **Další možnosti klasifikace programovacích jazyků**

Kromě tradičního dělení programovacích jazyků podle paradigmat lze jazyky kategorizovat i podle úrovně abstrakce, způsobu překladu a účelu použití. Nižší jazyky, jako Assembler a C, jsou blíže strojovému kódu, což umožňuje přesné řízení hardwaru, zatímco vyšší jazyky jako C#, Python nebo Java usnadňují vývoj díky větší abstrakci, což však může zpomalit výkon a paměťové operace.

Jazyky se dále rozlišují na kompilované, jako C a Rust, které jsou překládány do strojového kódu před spuštěním, a interpretované, jako JavaScript a Python, zpracováváné interpretem v reálném čase, což zvyšuje flexibilitu a usnadňuje ladění, ale může zpomalit provoz.

Podle účelu použití se mohou jazyky dělit na systémové, jako je C++, vhodné pro vývoj operačních systémů, a aplikační, jako PHP a Java, zaměřené na vývoj uživatelských aplikací. Specializované jazyky, například SQL, se využívají v oblastech vyžadujících konkrétní technické znalosti, jako je databázový management.

Další rozdělení je mezi back-end a front-end jazyky, kde back-end jazyky jako Python se zaměřují na serverovou logiku a zpracování dat, zatímco front-end jazyky jako JavaScript se specializují na tvorbu uživatelských rozhraní v prohlížečích. Popularita jazyka může ovlivnit dostupnost komunitních zdrojů a podporu, což je důležité při výběru jazyka pro specifické projekty.

Je důležité si uvědomit, že mnoho programovacích jazyků kombinuje rysy z různých paradigmat, což umožňuje vývojářům adaptovat techniky řešení problémů. Tato schopnost adaptace je zásadní pro efektivní vývoj softwaru, zejména v projektech, kde je potřeba kombinovat různé programovací přístupy nebo technologie. (46)

1.5.2 C#

C# je moderní programovací jazyk s důrazem na objektovou orientaci a typovou bezpečnost, umožňující vytvářet různorodé, bezpečné a odolné aplikace na platformě .NET. Tato platforma je více přiblížena v kapitolách 1.6.1 a 1.6.2. Díky svému původu v rodině jazyků C je tento jazyk snadno pochopitelný pro vývojáře, kteří mají zkušenosti s jazyky jako C, C++, Java nebo JavaScript.

V jádru C# leží objektově orientované paradigma, které umožňuje definovat typy a specifikovat jejich chování. Jazyk nabízí klíčové vlastnosti, jako je automatická správa paměti, eliminující potřebu manuálního uvolňování paměti, a podpora *nullable* typů pro zvýšení bezpečnosti kódu. C# dále poskytuje nástroje pro efektivní zachytávání a zpracování výjimek, podporuje funkcionální programování skrze lambda výrazy a obsahuje integrovaný dotazovací jazyk *LINQ* pro efektivní práci s daty.

Jazyk se vyznačuje sjednoceným typovým systémem, ve kterém všechny typy, včetně primitivních, dědí od základního objektu, což umožňuje konzistentní manipulaci s daty. C# podporuje jak hodnotové, tak referenční typy a nabízí efektivní práci s kolekcemi prostřednictvím generických metod a typů.

Zvláštní pozornost je v C# věnována verzování a kompatibilitě, což usnadňuje evoluci programů a knihoven v průběhu času. Specifické modifikátory a pravidla pro řešení přetížení metod přispívají k rozvoji stabilních a udržitelných aplikací.

V současnosti je C# ceněn pro svou univerzálnost, bezpečnost a výkon a najde uplatnění v široké škále aplikací – od webových a desktopových aplikací až po mobilní a cloudové služby. (47)

1.5.3 SQL

SQL neboli *Structured Query Language*, je programovací jazyk speciálně navržený pro správu a manipulaci s daty v relačních databázích. Tyto databáze organizují informace do tabulek, s řádky obsahujícími jednotlivé záznamy a sloupci, které reprezentují různé datové atributy. SQL umožňuje uživatelům provádět rozmanité operace, včetně ukládání, aktualizace, mazání a vyhledávání dat, a zároveň poskytuje nástroje pro údržbu a optimalizaci výkonnosti databází.

Díky své flexibilitě a kompatibilitě s jinými programovacími jazyky se SQL stalo oblíbeným nástrojem mezi datovými analytiky a vývojáři pro vývoj výkonných databázových aplikací. Tento jazyk je snadno osvojitelný díky používání běžných anglických termínů ve svých příkazech. Jeho kořeny sahají do 70. let minulého století, kdy byl vyvinut na základě relačního modelu dat a původně pojmenován „*SEQUEL*“. Název byl později zkrácen na SQL. Společnost Oracle se stala prvním poskytovatelem komerčního relačního databázového systému založeného na SQL. (48)

V SQL systému je základní prvek SQL tabulka, skládající se z řádků a sloupců. Tyto tabulky jsou zásadní pro vytváření a definování vztahů mezi daty uloženými v různých tabulkách, což vede k efektivnějšímu ukládání a správě dat. SQL příkazy, známé také jako dotazy, jsou využívány k interakci s databází. Tyto dotazy jsou formulovány pomocí různých prvků jazyka, včetně identifikátorů, proměnných a podmínek. (12)

Dalším klíčovým aspektem SQL jsou uložené procedury, což jsou soubory obsahující jeden nebo více SQL příkazů uložených přímo v databázi. Tyto procedury zvyšují efektivitu a výkon při práci s daty. Zpracování SQL dotazů se odehrává na serveru, který následně vrací výsledky. Tento proces zahrnuje několik kroků, od analýzy dotazu parserem, přes vytváření plánu pro manipulaci s daty relačním enginem, až po zpracování kódu ukládacím enginem, který čte a ukládá data na fyzická disková úložiště. (49)

1.5.4 Inno Setup

Inno Setup je bezplatné, open-source řešení pro tvorbu instalačních balíčků na platformě Windows, který vyniká svou výkonností a adaptabilitou. Podpora instalace i odinstalace široké škály Windows aplikací činí z tohoto nástroje komplexní řešení pro správu softwaru, oblíbené mezi vývojáři na všech úrovních zkušeností.

Inno Setup kombinuje vizuální editor, známý jako Inno Setup Wizard, pro snadné vytváření instalačních skriptů, a Inno Setup Compiler, který tyto skripty transformuje na spustitelné *EXE* soubory. Vytvoření instalačního programu zahrnuje výběr typu instalace, přidání souborů, nastavení možností instalace a generování finálního *EXE* souboru.

Mezi klíčové vlastnosti Inno Setup patří podpora pro všechny současné verze Windows, včetně 64bitových aplikací, možnost instalace bez nutnosti administrátorských oprávnění, tvorba kompaktního jedno-souborového instalačního balíčku a další. Tyto funkce zajišťují vysokou flexibilitu při tvorbě instalačních balíčků, které mohou být přizpůsobeny specifickým potřebám vývojových projektů.

Pro skriptování Inno Setup využívá jazyk inspirovaný Pascalem a Delphi, což vývojářům umožňuje efektivně přizpůsobovat instalační procesy vkládáním specifické logiky a kontrolních struktur do skriptů. Tento skriptovací jazyk zvyšuje flexibilitu a umožňuje tvorbu pokročilých, uživatelsky přívětivých instalačních balíčků pro Windows, čímž Inno Setup představuje ideální řešení pro distribuci softwaru. (50)

1.6 Možnosti vývoje desktopových aplikací

Tato kapitola představuje možnosti vývoje desktopových aplikací se zaměřením na programovací jazyk C#, který byl zvolen pro realizaci projektu, jímž se tato práce zabývá. Tento výběr je strategický pro vývoj aplikací pro pilnice, kde je zásadní spolehlivá komunikace se senzory a přesné zpracování dat. C# a platformy jako .NET Framework a .NET Core poskytují potřebné nástroje pro tvorbu aplikací, které efektivně interagují s hardwarem a zajišťují hladký průběh průmyslových procesů.

Zařazení WinForms či WPF do vývoje umožňuje tvorbu uživatelsky přívětivých rozhraní, což je důležité pro uživatele – v tomto případě zaměstnance obsluhující stroje. Proto se následující podkapitoly zaměřují na přidružené frameworky pro vývoj v C# a rovněž přibližují možnosti vývoje uživatelských rozhraní. (51)

1.6.1 .NET Framework

.NET Framework byl uveden na trh společností Microsoft v roce 2002 a představuje důležitý moment v historii vývoje softwaru pro Windows. Tato platforma totiž zpřístupnila vývojářům soubor konzistentních API (*Application Programming Interface*) pro vytváření aplikací, které mohou běžet na jakékoli verzi Windows. Díky tomu se výrazně snížila potřeba pro zvláštní přizpůsobení každé verzi operačního systému. Tento přístup tak pomohl omezit fragmentaci trhu s Windows aplikacemi, usnadňující vývojářům tvorbu univerzálně použitelných řešení.

Jádrem .NET Frameworku je Common Language Runtime (*CLR*). CLR umožňuje spouštět aplikace napsané v různých programovacích jazycích, což výrazně obohatilo ekosystém vývojářských nástrojů. Spolu s rozsáhlou knihovnou tříd pro běžné úkoly se .NET Framework stal základem pro vývoj řady aplikací. Rozvoj .NET Frameworku také otevřel cestu pro následující generaci .NET technologií. (51) (52)

1.6.2 .NET Core

.NET Core byl představen v roce 2016 jako reakce Microsoftu na rostoucí poptávku po multiplatformním vývoji aplikací. Platforma umožňuje vývoj pro Linux, macOS a Windows z jednotné základny kódu, což představuje odchod od tradičního zaměření

pouze na Windows. Hlavními motivy pro vývoj .NET Core byly modularita, optimalizace pro cloudové nasazení a schopnost efektivního spouštění aplikací na různých platformách. Platforma přináší vysokou výkonnost, lepší kontrolu nad správou paměti a snadnější správu závislostí díky modulární architektuře.

V roce 2022 byl uveden .NET 8, který dále rozšiřuje funkce a sjednocuje vývojářské prostředky napříč různými aplikacemi a platformami. Tato verze pokračuje ve zlepšování výkonnosti, modularity a cloudové optimalizace. Současně se již pracuje na .NET 9, který slibuje další inovace a vylepšení, zaměřené na zvýšení efektivity a produktivity vývojářů při práci s nejnovějšími technologickými trendy a zařazení nadcházejících potřeb softwarového průmyslu. (53) (54)

1.6.3 WinForms

WinForms, technologie uvedená v roce 2002 jako součást první verze .NET Frameworku, představuje nástupce původních Windows Forms, používaných ve Visual Basicu a C++. Díky své schopnosti zjednodušit design a vývoj se tato platforma rychle stala preferovanou volbou pro tvorbu desktopových aplikací s GUI. S bohatou sadou vizuálních komponent a nástrojů, WinForms výrazně usnadňuje práci vývojářům a poskytuje snadný přístup k databázím, síťovým službám a dalším funkcím .NET Frameworku. Jeho jednoduchost a efektivita vývoje významně přispěly k oblíbenosti WinForms mezi vývojáři. Proto také disponuje tato technologie rozsáhlou komunitou, což je další výhodou, jelikož umožňuje snadné sdílení znalostí a zdrojů, což vývojářům usnadňuje řešení problémů a inovace.

Přestože WinForms nabízí mnoho výhod, má také jisté nedostatky. Například ve srovnání s moderními frameworky může WinForms poskytovat působit zastarale a moderní prvky je složitější do GUI zakomponovat. Zároveň u komplexnějších aplikací může být problematický výkon.

S příchodem .NET Core a jeho nástupců, které cílí na multiplatformní vývoj a optimalizaci pro cloud, WinForms zůstává podporován v nejnovějších verzích .NET. Tato kontinuita umožňuje vývojářům využívat nové funkce a zároveň si zachovat známé postupy při tvorbě desktopových aplikací. (55)

1.6.4 WPF

WPF (*Windows Presentation Foundation*), uvedený jako součást .NET Frameworku 3.0 v roce 2006, umožnil vývoj desktopových aplikací s bohatými grafickými prvky a vektorovou grafikou. WPF byl navržen pro poskytování pokročilých nástrojů pro tvorbu moderních GUI, čímž překonává omezení starších technologií jako WinForms, zejména ve flexibilitě designu a vizuální kvalitě.

Hlavní přednosti WPF spočívají ve schopnosti vytvářet aplikace s moderním vzhledem a vysokou úrovní interaktivity, včetně podpory 3D grafiky a multimédií. Jeho architektura je optimalizovaná pro moderní aplikace, což umožňuje lepší výkon a složitější layouts a animace. Naopak, vyšší složitost a náročnost na systémové zdroje mohou ztěžovat použití WPF pro jednoduché aplikace a na starších systémech může docházet k problémům s kompatibilitou. (56)

1.6.5 .NET MAUI

.NET MAUI (*Multi-platform App UI*) přišel jako součást .NET 6 v roce 2021, je novým frameworkem od Microsoftu pro vývoj multiplatformních aplikací. Jako nástupce Xamarinu, .NET MAUI rozšiřuje možnosti vývoje aplikací nejen pro mobilní zařízení, ale také pro desktop a další platformy z jediné kódové báze. Tím umožňuje vývojářům vytvářet aplikace, které jsou konzistentní napříč různými zařízeními a operačními systémy, což zjednodušuje vývoj a údržbu softwaru. (57)

1.6.6 Avalonia

Avalonia, od svého uvedení na trh v roce 2017, se profiluje jako pokročilý framework pro tvorbu desktopových aplikací. Je postavena na principu otevřenosti (open-source) a kombinací XAML a .NET. Avalonia směřuje k poskytnutí rozsáhlých možností pro překonání omezení tradičních přístupů, jako WinForms či WPF, zejména v oblastech moderního designu a multiplatformní integrace.

Přednosti Avalonie zahrnují nejen její moderní GUI a schopnost implementovat komplexní prvky UI, ale také zlepšenou efektivitu a podporu pro práci s multimédií. Jako open-source řešení nabízí prostor pro rozvoj a přizpůsobení podle specifických potřeb

projektu. Díky podpoře systémů Windows, macOS a Linux, Avalonia přináší významnou flexibilitu pro projekty využívající diverzifikované operační systémy. (58)

1.7 Databáze

Databáze, definované jako strukturované kolekce propojených dat, slouží jako nezbytný nástroj pro ukládání a správu informací. Od svého vzniku v padesátých letech minulého století, kdy byly data uchovávána v jednoduchých souborových systémech, databáze prošly vývojem k sofistikovaným systémům, které umožňují efektivní práci s velkými objemy informací.

Existují primárně dva typy databází: *relační*, které ukládají data do tabulek pro strukturované informace, a *NoSQL* pro flexibilní manipulaci s nestrukturovanými daty, ideální pro webové aplikace potřebující rychlé zpracování rozličných dat. Přestože *NoSQL* databáze rozšiřují možnosti správy dat, relační databáze zůstávají důležité pro aplikace, kde je prioritou pevná struktura a integrita dat. (12) (59)

1.7.1 SQLite

SQLite je kompaktní relační databázový systém, jehož hlavní přednosti – malá velikost a vysoká rychlost – ho činí ideální pro širokou škálu aplikací. Jeho schopnost být snadno integrován přímo do aplikací eliminuje potřebu externích databázových serverů, což je sympatické zejména pro mobilní a některé desktopové aplikace. Dostupnost SQLite jako open-source řešení umožňuje vývojářům přístup k pokročilým databázovým funkcím bez dodatečných nákladů.

SQLite podporuje vývoj aplikací s omezenými systémovými zdroji a nabízí vysokou míru nezávislosti na platformě. Přestože je vhodná pro různé použití, je důležité uvážit specifické požadavky každého projektu, včetně potřeby škálovatelnosti a výkonu, aby bylo zajištěno, že SQLite je pro danou situaci opravdu nejvhodnější řešení. (60)

1.7.2 MySQL

MySQL je dalším uznávaným open-source relačním databázovým systémem. Mnoho softwarových řešení je postaveno právě na tomto systému. Jeho schopnost efektivně organizovat, ukládat a zpřístupňovat data přes relační tabulky, umožňuje vývojářům

a organizacím využívat komplexní databázové funkce bez významných investic do licencí. S podporou většiny operačních systémů a integrací s mnoha programovacími jazyky, MySQL zajišťuje vysokou míru adaptability a je vhodný pro široké spektrum aplikací – od jednoduchých webových stránek až po komplexní podnikové systémy.

Díky své výkonnosti, škálovatelnosti a robustním bezpečnostním opatřením je MySQL preferovanou volbou pro vývojáře hledající efektivní a spolehlivý způsob správy dat. Jeho vlastnosti jako transakční integrita a podpora pro replikaci jsou vhodné pro projekty vyžadující vysokou dostupnost a konzistenci dat. Navíc, aktivní a rozsáhlá komunita MySQL přispívá k jeho neustálému vývoji a zajišťuje bohatý zdroj návodů, nástrojů a doplňků, které usnadňují práci s tímto systémem. (61)

1.7.3 Maria DB

MariaDB vznikla v roce 2009 jako fork MySQL. Jde o open-source relační databázový systém (*RDBMS*), kompatibilní s MySQL a přináší řadu vylepšení a nových funkcí. Díky své kompatibilitě umožňuje snadnou migraci z MySQL a zase zpět, což ji činí atraktivní volbou pro ty, kteří hledají alternativu k MySQL. MariaDB je zdarma a její vývoj a udržování je podporováno aktivní komunitou, což zajišťuje stálý přísun inovací a podpory.

Používá se v řadě aplikací, od webových a e-commerce platforem, přes CRM a ERP systémy, až po analytické nástroje a datové sklady. MariaDB zdůrazňuje flexibilitu a otevřenost, čímž uspokojuje požadavky moderního softwarového vývoje a datové správy. (62) (63)

1.8 Testování softwaru

Testování softwaru je zásadní činností ve vývojovém cyklu, jak bylo zmíněno v kapitole 1.3 a zaměřuje se na ověření funkčnosti a odolnosti softwarových aplikací. Jeho hlavní účel spočívá ve vyhledávání a odstraňování chyb, aby finální produkt splňoval specifikace a očekávání uživatelů.

Existuje mnoho různých metod testování, od unit testů, přes integrační až po systémové a akceptační testy. Každá z těchto metod hraje důležitou roli v detekci specifických typů problémů. Unit testy se soustředí na nejmenší části kódu, jako jsou funkce a metody,

zatímco integrační testy ověřují, jak tyto části spolupracují. Systémové testování zkoumá celý systém v jeho komplexnosti a akceptační testování se zaměřuje na splnění požadavků koncových uživatelů.

Proces testování prochází několika fázemi, od plánování, přes tvorbu a provedení testů, až po analýzu výsledků a jejich reportování. Tento strukturovaný přístup zajišťuje důkladnou kontrolu softwaru a jeho připravenost pro reálné využití.

Automatizace hraje v testování softwaru významnou roli, neboť testovací nástroje mohou značně usnadnit a zrychlit celý proces. Tyto nástroje umožňují vývojářům sledovat a následně analyzovat chování softwaru.

Důkladné testování je nezbytné pro vývoj kvalitního softwaru a jde o komplexní systém, který při jeho správném použití zajišťuje bezpečnost a správnost vyvíjeného softwaru.
(20) (64)

1.8.1 Unit Testy

Unit testing je zásadní technika v procesu vývoje softwaru, zaměřená na verifikaci nejmenších částí kódu, jako jsou jednotlivé metody nebo třídy. Tato metoda umožňuje identifikovat a řešit problémy na nejnižší úrovni, což výrazně přispívá k celkové kvalitě a spolehlivosti finálního produktu.

Hlavním cílem unit testování je zajištění, že každá komponenta softwaru funguje podle očekávání a splňuje předem definované specifikace. Zároveň se chyby v jednotlivých samostatných komponentách či malých částech softwaru odhalují, identifikují, a hlavně opravují výrazně jednodušeji a rychleji než při testování větších částí aplikací. Díky tomu také dochází ke snížení celkových nákladů na vývoj.

Praxe unit testování rovněž podporuje čistý kód a usnadňuje refaktoring, protože každý kus kódu musí být dostatečně modularizovaný, aby bylo možné ho samostatně testovat.

Automatizace unit testů je dalším krokem, který zvyšuje efektivitu vývoje tím, že umožňuje rychlé a opakované spouštění testů. Tato automatizace nejen šetří čas, ale také zajišťuje konzistentní testovací prostředí a snižuje možnost lidských chyb.

Unit testování, včetně technik jako Test-Driven Development (*TDD*), tak vytváří základ pro velmi stabilní a korektní aplikace. Zároveň umožňuje týmům rychle reagovat na

chyby a udržovat kód kvalitní. Jedná se o praktiky, které by si měly najít místo při vývoji jakéhokoli softwaru. (65) (66)

1.9 Analytické nástroje

V podnikovém managementu a strategickém plánování jsou analytické nástroje nesmírně důležité pro identifikaci vnitřních a vnějších faktorů ovlivňujících organizaci. Umožňují manažerům systematicky analyzovat různé aspekty podnikání, od interních procesů až po tržní dynamiku a vnější vlivy, čímž pomáhají při tvorbě efektivních strategií. Nástroje jako model 7S, SWOT, SLEPT a Porterova analýza poskytují podklady pro komplexní hodnocení a umožňují organizacím vytvářet plány, které podporují udržitelný růst a konkurenční výhodu. Tyto nástroje jsou více přiblíženy a popsány v následujících podkapitolách.

1.9.1 SLEPT

SLEPT analýza slouží jako důležitý nástroj pro hodnocení vnějších faktorů, které mohou ovlivnit fungování organizace. Akronym SLEPT zastupuje pět zásadních oblastí: Sociální, Legislativní, Ekonomické, Politické, a Technologické faktory. Tato metoda umožňuje podnikům systematicky prozkoumat vnější prostředí a identifikovat potenciální problémy a příležitosti, které mohou ovlivnit jejich strategii a operace.

Sociální složka se zabývá změnami v populaci, jako jsou demografické trendy a změny v preferencích spotřebitelů, které mohou ovlivnit tržní poptávku. Legislativní faktory se týkají právních a regulačních změn, které mohou ovlivnit způsob, jakým organizace podniká. Ekonomické faktory zahrnují celkové ekonomické klima a makroekonomické ukazatele, jako je inflace, nezaměstnanost, a růst HDP, které mohou mít dopad na nákupní sílu a investiční rozhodování. Politické složky odkazují na politickou stabilitu a politiky, které mohou ovlivnit podnikatelské prostředí. Nakonec, Technologické faktory představují rychlost technologického vývoje a inovací, které mohou poskytnout nové příležitosti pro podnikání nebo zastaralost existujících produktů a služeb.

Provozování SLEPT analýzy umožňuje vedoucím pracovníkům lépe porozumět a předvídat vnější vlivy, čímž se zvyšuje schopnost organizace adaptovat se na měnící se podmínky a maximalizovat svůj potenciál pro růst. Přestože nemohou poskytnout

konkrétní citace, doporučuji pro hlubší studium a aplikaci tohoto nástroje konzultovat odbornou literaturu a akademické zdroje zaměřené na strategické řízení a analýzu podnikového prostředí. (67) (68)

1.9.2 Porterova analýza

Porterova analýza, známá také jako model pěti sil, je klíčovým nástrojem pro strategickou analýzu průmyslového odvětví. Tento model byl vyvinut Michaelem E. Porterem z Harvard Business School v roce 1979 a je zaměřen na identifikaci pěti základních konkurenčních sil, které ovlivňují podniky ve všech odvětvích. Tyto síly určují intenzitu konkurence a tím i atraktivitu a ziskovost odvětví.

První silou je síla *konkurence mezi stávajícími firmami* v odvětví. Vysoký počet konkurentů nebo podobně silné firmy mohou vést k intenzivní konkurenci, která snižuje zisky. Druhou silou jsou *hrozby nových konkurentů*, kteří mohou vstoupit na trh a narušit stávající rovnováhu. Třetí silou jsou *hrozby substitutů*, tedy alternativní produkty nebo služby, které mohou plně nahradit stávající produkty a tím snížit jejich poptávku.

Čtvrtou důležitou silou jsou *vyjednávací síly dodavatelů*, která se týká schopnosti dodavatelů ovlivňovat cenu a dostupnost materiálů nebo služeb. Pátou silou jsou *vyjednávací síly zákazníků*, která odráží schopnost zákazníků vyjednávat nižší ceny a lepší služby.

Porterův model pěti sil poskytuje rámec, který pomáhá podnikům pochopit vnější faktory ovlivňující jejich konkurenční prostředí a umožňuje jim formulovat efektivní strategie pro zvýšení konkurenceschopnosti a obrany proti konkurenčním hrozbám. Tato analýza je zásadní pro každou firmu, která se snaží udržet si nebo zvýšit svou pozici na trhu. Díky porozumění těmto pěti silám se mohou firmy lépe orientovat ve svém průmyslovém prostředí a přizpůsobit své strategické plány tak, aby maximalizovaly svůj zisk a minimalizovaly rizika. (67) (69)

1.9.3 Model 7S

Model 7S je strategický nástroj vytvořený v 80. letech konzultantskou firmou McKinsey & Company, zaměřený na analýzu a následné zlepšení efektivity organizace. Jeho jádrem je přesvědčení, že pro úspěšné dosažení cílů a implementaci strategií není dostatečná

pouze správně formulovaná strategie, ale je nezbytné, aby sedm klíčových vnitřních prvků organizace – strategie, struktura, systémy, sdílené hodnoty, dovednosti, personál a styl vedení – bylo vzájemně sladěno a harmonizováno.

Tento model nabízí komplexní pohled na organizaci, umožňuje identifikovat oblasti, které vyžadují změny nebo zlepšení, a slouží jako vodítko pro vyvážený rozvoj všech aspektů organizace. Sdílené hodnoty, stojící v centru modelu, zdůrazňují význam organizační kultury a etiky jako klíčového řídicího principu, který ovlivňuje všechny ostatní prvky. Model 7S tak poskytuje ucelený rámec pro analýzu, jak strategické, tak organizační a lidské faktory působí a vzájemně se ovlivňují, což umožňuje vedení organizace lépe pochopit složitosti a dynamiku interního prostředí. (67) (70)

1.9.4 SWOT

SWOT analýza je uznávaným nástrojem strategického řízení, který pomáhá organizacím v identifikaci interních a externích faktorů, které jsou klíčové pro dosažení strategických cílů. Tato metoda analyzuje čtyři základní dimenze: Silné stránky (*Strengths*), Slabé stránky (*Weaknesses*), Příležitosti (*Opportunities*) a Hrozby (*Threats*).

Silné stránky představují interní zdroje a schopnosti, které organizace může využít jako výhodu ve srovnání s konkurencí. Slabé stránky jsou interní nedostatky, které mohou omezit efektivitu nebo konkurenceschopnost organizace. Příležitosti jsou faktory, které organizace může využít k dosažení nebo překročení svých strategických cílů. Naopak, hrozby jsou rizika, která by mohla negativně ovlivnit organizaci.

Praktické využití SWOT analýzy spočívá v možnosti systematicky identifikovat a hodnotit faktory ovlivňující projekt nebo podnikání a v následném formulování strategií, které tyto faktory berou v úvahu. SWOT analýza může sloužit jako základ pro strategické plánování, pomáhá při rozhodování a přispívá k lepšímu pochopení trhu a konkurenčního prostředí. (67) (71)

2 ANALÝZA PROBLÉMU A SOUČASNÉ SITUACE

V této kapitole je provedena analýza současné situace firmy BJSW. Nejdříve bude představena společnost a její vnitřní a vnější prostředí pomocí modelů 7S, Porterovy analýzy pěti sil a SLEPT analýzy. Výstupy z jednotlivých analýz budou sumarizovány prostřednictvím SWOT analýzy, která umožní identifikovat hlavní příležitosti a hrozby pro projekt. Zároveň budou v rámci této kapitoly popsány technologické aspekty, včetně architektury aplikací, vývojových nástrojů a testovacích metod.

2.1 Představení společnosti

Tato diplomová práce byla vytvořena ve spolupráci s firmou BJSW. Společnost BJSW byla založena v České republice a má sídlo v Kroměříži a v Brně. V Brně se také nachází větší část zaměstnanců a hlavní pracovní prostory. Firma BJSW se primárně zaměřuje na vývoj zakázkového softwaru, či jeho části, jako externí dodavatel. Klientela je různorodá a zaměřená především na český trh, nicméně mezi další významné, zahraniční, klienty se řadí Dánsko nebo Izrael. Firma poskytuje řešení pro různé průmyslové odvětví a mezi její hlavní projekty patří softwarové nástroje pro kontrolu bezpečnostních systémů, měření hluku a vibrací, vojenské simulace, testování zařízení pro kosmické projekty.

Firma ale také vytváří čistě své vlastní softwarové řešení, například pro řízení a monitorování procesů ve dřevozpracujícím průmyslu. (72)

Právě tímto softwarem se bude práce v praktické části zabývat. Podrobnějším popisem firmy se zabývá kapitola 2.3.1.



Obrázek 2.1 - Logo společnosti BJSW CZ
(Zdroj: (72))

2.2 Analýza vnějšího prostředí

Analýza vnějšího prostředí poskytuje přehled o externích faktorech, které ovlivňují operace a strategické rozhodování společnosti. V této části práce budou použity modely SLEPT a Porterova analýza pěti sil.

2.2.1 SLEPT

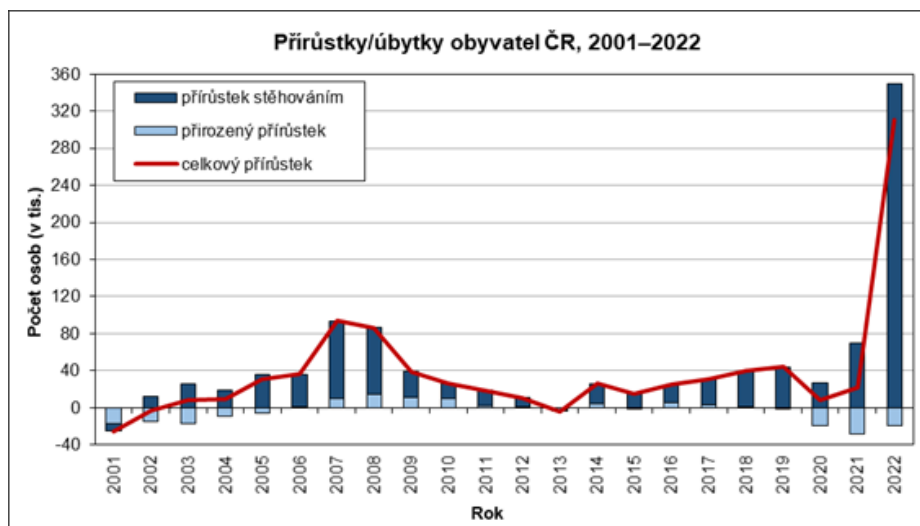
V této kapitole bude provedena SLEPT analýza společnosti, která zkoumá sociální, legislativní, ekonomické, politické a technologické faktory ovlivňující její podnikání. Analýza slouží k identifikaci vnějších vlivů a trendů, které mohou ovlivnit strategické rozhodování a umožnit firmě lépe reagovat na měnící se podnikatelské prostředí.

2.2.1.1 Sociální faktory

V oblasti sociálních faktorů je firma BJSW vystavena dynamickým demografickým trendům, které ovlivňují jak trh, tak pracovní prostředí. Rok 2022 přinesl České republice nárůst populace na 10,828 milionu, z velké části díky imigrační vlně z Ukrajiny. Tento příliv nových obyvatel může ovlivnit poptávku po technologických řešeních, zejména v sektorech, kde jsou tyto komunity aktivní, jako jsou právě operátoři v pilnicích. (73)

Produktivní složka populace, tvořící 63,4 %, a mladší generace, která v roce 2022 zaznamenala nevídaný přírůstek, naznačují trvalou potřebu inovativních a efektivních pracovních nástrojů. Zvláště důležitá je adaptabilita a inovace ve vývoji produktů, aby odpovídaly potřebám mladších demografických skupin a reagovaly na technologický vývoj. (74)

Zároveň demografické prognózy ukazují na stárnutí populace. To s sebou přináší také nutnost přizpůsobit technologická řešení právě stárnoucí společnosti. Implementace reportovacího modulu SMDH, který umožňuje náhled na změřené kmeny a jejich editaci například při chybném změření, se tak jeví jako strategický krok vstříc nejen aktuálním, ale i budoucím tržním požadavkům. (75)



Obrázek 2.2 - Přírůstek obyvatel ČR
(Zdroj: (74))

2.2.1.2 Legislativní faktory

Legislativní faktory hrají klíčovou roli v operacích a strategii podniku, zvláště kvůli její působnosti v různorodých sektorech jako jsou dřevozpracující průmysl, vojenské simulace, vesmírné projekty a technologie pro měření hluku a vibrací. Přísné regulace ochrany životního prostředí mohou ovlivnit vývoj softwaru pro dřevozpracující průmysl, vyžadující inovace pro snižování ekologické stopy. Vojenské a vesmírné aplikace jsou regulovány národními a mezinárodními bezpečnostními standardy, což vyžaduje pečlivé dodržování předpisů ve vývoji softwaru a jeho exportu. Zákony o ochraně dat a soukromí také hrají roli, zvláště v oblasti zpracování osobních dat, což může ovlivnit jak interní procesy, tak vývoj produktů pro trh. Pro firmu je nezbytné udržet si aktuální přehled o legislativě v každém z jejích segmentů trhu a zajistit, že její produkty a operace jsou plně v souladu s platnými zákony a regulacemi. Toto vyžaduje nejen reaktivní přizpůsobení se změnám v legislativě, ale také proaktivní strategické plánování, aby se předešlo potenciálním právním a finančním rizikům.

2.2.1.3 Ekonomické faktory

V roce 2023 česká ekonomika zaznamenala pokles HDP o 0,4 %, primárně v důsledku vysoké inflace a poklesu domácí spotřeby. Tento stav však otevírá dveře oživení v následujících letech, což přináší potenciální příležitosti pro sektor IT, zejména v oblastech digitalizace a automatizace, které mohou být pro firmu klíčovými oblastmi růstu. (76) (77)

Přes celkové zpomalení ekonomiky zůstává trh práce v IT sektoru dynamický, s rostoucí poptávkou po specialistech. To vyžaduje od společností, včetně BJSW, zvýšené investice do získávání a udržení talentů. S očekávaným snižováním inflace v následujících letech by mělo nastat zlepšení kupní síly, což představuje další nové tržní příležitosti. (76) (78) V kontextu nutných strukturálních změn a narůstající potřeby digitalizace se ukazuje, že vládní podpora a fondy EU představují významné zdroje pro financování inovací v IT. Firma by měla aktivně vyhledávat tyto příležitosti, aby podpořila své inovační projekty a udržela si konkurenční výhodu. (78)

Vzhledem k citlivosti české ekonomiky, a tedy i IT sektoru, na globální ekonomické události, je nezbytné, aby firma udržovala flexibilní strategii schopnou reagovat na různé nadcházející výzvy. Flexibilita a schopnost rychle se přizpůsobit novým technologiím jsou klíčové pro zachování konkurenceschopnosti. Ta bude podpořena také při dodávání požadovaných úprav a vylepšení ze strany zákazníků u stávajících řešení, což firmě také pomůže zvýšit hodnotu svých produktů a získat nové zakázky. (78)

2.2.1.4 Politické faktory

V politickém kontextu se společnost vyrovnává s klíčovými regulacemi, které ovlivňují celé IT odvětví. GDPR přináší přísná pravidla pro ochranu dat a soukromí, což vyžaduje významné investice do zabezpečení a dodržování legislativy, ale zároveň umožňuje rozvoj v oblasti kybernetické bezpečnosti. Česká legislativa, včetně podpory výzkumu a vývoje, spolu s politickou stabilitou, podporuje inovace a udržení konkurenceschopnosti. Globální politické napětí a obchodní války, jako jsou ty mezi USA a Čínou, mohou ovlivnit přístup k trhům a zdrojům, což vyžaduje proaktivní sledování a adaptaci na mezinárodní vývoj. Pro firmu je tedy zásadní monitorování politických trendů a rychlé reakce na legislativní změny, aby efektivně navigovala v dynamickém prostředí a využívala dostupné politické a legislativní příležitosti.

2.2.1.5 Technologické faktory

Ve sféře technologických faktorů se podnik musí vyrovnávat s rychlým vývojem technologií a inovací, které ovlivňují celý IT sektor. Zvláště významný je rostoucí důraz na umělou inteligenci (AI) a automatizaci, což otevírá nové tržní příležitosti pro firmu, zároveň však vyžaduje neustálé investice do výzkumu a vývoje pro udržení konkurenceschopnosti. Technologické pokroky, jako je cloud computing, big data a IoT (internet věcí), mění způsob, jakým firmy a odvětví operují, a BJSW musí tyto trendy

nejen sledovat, ale aktivně do nich investovat. Zabezpečení dat a ochrana soukromí jsou další klíčové aspekty, které vyžadují neustálou pozornost, zejména ve světle stále přísnějších regulací v oblasti kybernetické bezpečnosti. Adaptabilita a inovační kapacita jsou tak pro firmu nezbytné k efektivnímu využívání technologických trendů a k ochraně před potenciálními hrozbami spojenými s technologickými změnami.

Ve sféře technologických faktorů se podnik musí vyrovnávat s rychlým vývojem technologií a inovací, které ovlivňují celý IT sektor. Rostoucí důraz na umělou inteligenci (AI) a automatizaci, společně s rozvojem kvantových technologií, nabízí nové tržní příležitosti, avšak vyžaduje neustálé investice do výzkumu a vývoje. Cloud computing, big data, IoT (internet věcí), a nasazování 5G sítí mění způsob, jakým firmy operují, s čímž by měla firma počítat. Bezpečnost dat a ochrana soukromí, zesílené přísnými regulacemi, vyžadují také neustálou pozornost. Zvláště významný je rostoucí důraz na ekologické technologie v rámci "zelené IT", který reflektuje globální důraz na udržitelnost. Adaptabilita a inovační kapacita jsou pro firmu nezbytné k efektivnímu využívání technologických trendů a ochraně před potenciálními hrozbami. Nové bezpečnostní hrozby vyplývající z AI a kvantových výpočtů vyžadují neustálé aktualizace bezpečnostních protokolů a investice do kybernetické bezpečnosti, aby firma mohla identifikovat a minimalizovat potenciální hrozby, udržet si konkurenceschopnost a efektivně reagovat na dynamický technologický vývoj.

2.2.2 Porterova analýza 5 sil

V této kapitole bude využit Porterův model pěti konkurenčních sil, pomocí kterého bude analyzováno průmyslové odvětví a konkurenční prostředí firmy a projektu. Využitím tohoto modelu získávají firmy ucelený pohled na hrozby a příležitosti v jejich odvětví, což jim umožňuje lépe formulovat své strategie, zvýšit konkurenceschopnost a optimalizovat své postavení na trhu.

2.2.2.1 Stávající konkurence

V kontextu obecného sektoru zakázkového softwaru čelí BJSW vysoké konkurenci, kde se řada společností také zaměřuje na vytváření softwarových řešení, které jsou přizpůsobené specifickým potřebám klientů. Tento sektor je charakterizován intenzivním bojem o zákazníky, přičemž některé firmy se specializují výhradně na vývoj

informačních systémů na míru, jiné na aplikace a některé poskytují širší škálu softwarových služeb.

Avšak, ve specifickém segmentu softwaru pro automatizované pilnice, kde BJSW působí se svým projektem SMDH, je konkurenční prostředí méně nasycené. V tomto úzkém odvětví je hlavním konkurentem Kesat, který představuje jediného významného hráče. Společnost Kesat se specializuje na průmyslovou automatizaci a robotizaci, nabízí komplexní služby v oblasti automatizace výrobních linek, robotických pracovišť, a modernizace elektroinstalace. Jednou z hlavních nabízených služeb jsou právě automatizace v dřevoprůmyslu. Zabývá se také integrací speciálních brusek pro průmyslové roboty a vývojem řídicích systémů. Kesat poskytuje své služby od návrhu přes realizaci až po servis, s cílem zvyšovat efektivitu výroby a kvalitu obrábění u svých zákazníků.

Nicméně takto omezená konkurence dává firmě příležitost využít své znalosti a technologické schopnosti, a ještě více posílit svoji tržní pozici.

Příležitost, které může v tomto segmentu BJSW využít, je potenciál nabídnout funkce, které v současnosti nejsou na trhu dostupné, jako je možnost editace už změřených kmenů a tvorba různých PDF sestav. Tato specifická výhoda by mohla být ještě více zvýrazněna implementací reportingového modulu, který by uživatelům umožnil efektivněji spravovat a analyzovat data, čímž by se zvýšila hodnota nabízená BJSW jejich zákazníkům, kteří takové funkce poptávají.

Přestože BJSW čelí silné konkurenci v širším kontextu zakázkového softwaru, zároveň specializace v oblasti softwaru pro automatizované pilnice jí poskytuje strategickou výhodu. Závazek BJSW k adaptaci na specifické požadavky zákazníků a k neustálému inovování je zásadní pro udržení a rozvoj její konkurenční pozice v tomto specifickém segmentu.

2.2.2.2 Hrozba vstupu nové konkurence

Vývoj softwaru obecně čelí nízkým vstupním bariérám, což usnadňuje příchod nových hráčů na trh a zvyšuje konkurenční tlak v oblasti získávání jednotlivých zakázek. V tomto dynamickém prostředí se odlišení a inovace stávají klíčem k udržení atraktivity softwarových řešení. Firmy jsou tak nuceny neustále zdokonalovat své technologie a služby, aby splnily a ideálně překonaly očekávání zákazníků a odlišily se tak od

konkurence. To zahrnuje jak vývoj softwaru na míru, tak poskytování přidané hodnoty skrze pokročilé funkce a vynikající zákaznický servis. Významnou roli zde hraje přístup k inovacím, jako je integrace nových technologií a rozšíření funkčnosti softwaru, což může být klíčové pro udržení konkurenční výhody.

Ve specifickém segmentu softwaru pro pilnice se ukazuje, že potenciální noví konkurenti by mohli přijít ze strany výrobců průmyslových strojů, kteří by mohli nabízet integrované softwarové řešení. Toto představuje značnou hrozbu pro firmy jako BJSW, jelikož výrobci strojů mají již stabilní vztahy se svými zákazníky a mohou využít své technologické a inženýrské zázemí k vývoji přizpůsobených softwarů. Integrace hardwaru a softwaru by mohla poskytnout silnou konkurenční výhodu, která by mohla přilákat stávající i nové zákazníky.

Další hrozbou je možnost, že firmy z kovožpracujícího a jiných průmyslových odvětví, které již mají zkušenosti s vývojem softwaru pro automatizaci a monitorování výrobních procesů, rozšíří své technologické portfolio o řešení pro dřevozpracující průmysl. Tyto firmy, jelikož půjde pravděpodobně o větší společnosti, by mohly snadněji získávat nové zákazníky.

V obou případech je klíčové, aby firmy jako BJSW neustále monitorovaly tržní trendy a technologický vývoj, aby byly schopny rychle reagovat na potenciální nové konkurenty. Investice do výzkumu a vývoje, zejména v oblastech jako machine learning a umělá inteligence, spolu s rozvojem pokročilých analýz dat, umožní firmě udržet technologický náskok. Rozšiřování funkcí softwaru, jako je implementace reportingového modulu, může posílit tržní pozici tím, že nabídne unikátní přínos zákazníkům – zlepšený přehled nad daty a efektivnější řízení operací, což jsou nesmírně důležité aspekty pro fungování společností ve dřevozpracujícím průmyslu.

2.2.2.3 Hrozby substitutů

V segmentu softwaru pro pilnice, kde BJSW nabízí specializovaná řešení, existuje na trhu velice omezený počet přímých substitutů. Hlavní konkurent, Kesat, i když nabízí automatizační technologie pro dřevozpracující průmysl, se nezaměřuje přímo na specifické aplikace pro pořez dřeva, ale pouze na měření kmenů. Tento rozdíl v zaměření znamená, že alternativní produkty mohou nabízet sice rozdílné funkce, ale často nesplňují

přesné požadavky zákazníků BJSW, kteří vyžadují sofistikovaný software pro efektivní řízení a optimalizaci procesů pořezu a měření dřeva.

Nízký počet přímých substitutů znamená, že zákazníci hledající specifické řešení mají omezené možnosti, co se týče přechodu na jiné produkty. Přechodové náklady nejsou omezeny pouze na finanční aspekt, ale zahrnují také vynaložený čas a úsilí na vyhledání vhodné alternativy, integraci nového softwaru a školení zaměstnanců. Tyto náklady mohou být významné, zejména pokud nové řešení neumožňuje snadnou integraci nebo vyžaduje rozsáhlé úpravy pro splnění specifických potřeb.

Navíc, přechod na jiný produkt může být komplikován nedostatkem kompatibility s existujícími hardwarovými systémy, což představuje další technické a operativní problémy. Tím pádem, pokud zákazníci zvažují substituci softwaru od BJSW, musí pečlivě zvážit nejen přímé náklady spojené s nákupem nového softwaru, ale i potenciální dopady na jejich operace a celkovou produktivitu.

Vzhledem k těmto faktorům je zásadní, aby BJSW kontinuálně inovovala a rozšiřovala své softwarové řešení o nové funkce, které zvyšují jeho hodnotu pro zákazníky a udržela si tak konkurenční výhodu. Zároveň substituty aktuálně neposkytují možnosti upravovat naměřené kmeny, či vytvářet tiskové sestavy. Pro projekt SMDH je tedy příležitostí integrovat tyto funkce a získat tak další strategickou výhodu. Tím by firma ukázala také skutečnost, že proaktivně reaguje na dynamické požadavky trhu a může tak předstihnout potenciální vývoj konkurence.

2.2.2.4 Odběratelé a zákazníci

BJSW se setkává s rozmanitou zákaznickou základnou napříč různými sektory, což přináší variabilitu do velikosti a specifikací objednávek. Tato diverzita umožňuje zákazníkům přizpůsobit si softwarová řešení podle svých potřeb. Ačkoliv teoreticky mohou zákazníci přejít k alternativním dodavatelům, praktické a finanční náklady na takovou změnu jsou poměrně vysoké. Tyto bariéry činí přechod k jinému dodavateli komplikovaným a časově náročným procesem.

Konkrétně pak software SMDH je klíčovým prvkem pro správné a efektivní fungování pilnic, včetně ovládání strojů a senzorů. Jeho zásadní role v procesech a operacích extrémně zvyšuje náklady a komplikace spojené s případným přechodem na jiné řešení, což posiluje vyjednávací pozici společnosti. Pro firmu je nezbytné udržet si vysokou

úroveň služeb a neustále inovovat své produkty na základě požadavků od zákazníků, aby si zajistila jejich loajalitu a odolala konkurenci.

2.2.2.5 Dodavatelé

Pokud jde o vyjednávací sílu dodavatelů, BJSW stojí na pevném základě díky internímu vývoji softwaru. Firma není závislá na externích dodavatelích pro žádnou klíčovou komponentu svých softwarových produktů, což eliminuje riziko ovlivnění kvality či dodacích lhůt z externí strany. Tato samostatnost poskytuje firmě významnou kontrolu nad celým procesem vývoje a umožňuje jí rychle reagovat na požadavky zákazníků a měnící se tržní podmínky.

2.3 Analýza vnitřního prostředí

Analýza vnitřního prostředí se zaměřuje na posouzení interních faktorů společnosti, které mohou ovlivnit její schopnost realizovat nové projekty a dosahovat strategických cílů. V této části bude využit model 7S.

2.3.1 Model 7S

Model 7S je komplexní nástroj pro diagnostiku a formování organizací. Tato analýza pomáhá pochopit, jak různé aspekty firmy pracují společně a jaký dopad mají na schopnost firmy dosáhnout svých cílů.

2.3.1.1 Strategie

Strategie firmy BJSW je založena na vývoji customizovaných softwarových řešení pro široké spektrum průmyslových odvětví. Zaměřuje se na využití moderních technologií a vývojových nástrojů, aby její produkty reflektovaly jak aktuální, tak budoucí potřeby klientů. Tento přístup umožňuje firmě neustále inovovat a udržovat si konkurenční výhodu.

V rámci dosahování svých cílů firma klade důraz na integraci a otevřenost řešení. To zahrnuje spolupráci s dalšími softwary a automatizaci procesů, čímž proaktivně reaguje na měnící se požadavky zákazníků s vysokou mírou předvídativosti a adaptability. Díky tomuto přístupu si firma udržuje silnou pozici na trhu a zároveň poskytuje klientům významnou přidanou hodnotu.

Ve vztahu ke konkurenci se BJSW odlišuje především kvalitou svých služeb, inovačním přístupem a schopností rychle reagovat na nové tržní trendy. Co se týče životního prostředí, firma přistupuje zodpovědně, což dokládá například projekt SMDH, kde je kladen důraz na minimalizaci odpadu a maximální využití dřevní hmoty.

Tato strategie, založená na kvalitě, udržitelnosti a inovacích, umožňuje firmě efektivně získávat nové zakázky a podporovat svůj trvalý rozvoj. Díky tomuto přístupu si dokáže firma udržet konkurenční výhodu a zároveň přispívat k pozitivnímu vývoji v oblastech, ve kterých působí.

2.3.1.2 Struktura

Interní struktura BJSW je klíčová pro dynamiku a úspěšný vývoj zakázkového softwaru. Centrální roli v tomto uspořádání hraje vedení, sestávající ze tří vrcholových manažerů, kteří přijímají strategická rozhodnutí a určují směr, kterým se společnost bude ubírat. Tyto rozhodnutí jsou podkladem pro práci specializovaných týmů, přizpůsobených na jednotlivé projekty a zakázky. Zatímco jedna část firmy je zaměřena čistě na dánské klienty, zbytek společnosti je strukturován do menších, autonomních týmů, které se věnují široké škále dalších projektů. Tento přístup umožňuje firmě rychle reagovat na měnící se požadavky a zároveň zajišťuje, že všechny projekty jsou dokončeny včas a na vysoké úrovni kvality.

Vrcholový management nejenže zastřešuje celkovou strategii a kvalitu pracovních procesů, ale zároveň se aktivně věnuje udržování a rozvíjení vztahů s klientelou. Toto zahrnuje komunikaci se současnými klienty, ale také získávání nových zakázek a zákazníků, což je stěžejní pro další, budoucí rozvoj a růst firmy.

Vedle vedení jsou nedílnou součástí managementu i team leaderi, odpovědní za každodenní vedení a koordinaci projektů. Každý team leader vede menší tým, sestávající z několika programátorů. Team leader nese odpovědnost za komunikaci s vrcholovým managementem a klienty, pro které daný tým vytváří řešení. Tyto interakce zajišťují, že týmy mají všechny potřebné informace pro efektivní a přesný vývoj softwaru. Současně je zaručeno, že jakékoliv změny v projektu jsou reportovány zpět vrcholovému managementu, což zajišťuje koherentnost ve všech částech firmy. Tato vzájemná spolupráce mezi vrcholovým managementem a team leadery zajišťuje efektivní tok

informací a umožňuje společnosti reagovat na požadavky klientů s maximální rychlostí, efektivitou a zároveň poskytovat kvalitní a přesná řešení.

Zavedená hierarchie ve firmě umožňuje efektivní přechod od centralizovaných strategických rozhodnutí k decentralizovanému dennímu řízení projektů. Tato flexibilita je zásadní v rychle se měnícím IT odvětví, kde je absolutně nezbytná rychlá reakce na měnící se požadavky ze strany klientů a nově přichozí technologie. Neformální, avšak účinná koordinace týmů odráží filozofii společnosti, která klade důraz na agilitu a inovaci, a současně si zachovává velmi osobní přístup k zaměstnancům a klientům. Zároveň flexibilita takové struktury podporuje neustálý růst firmy.

2.3.1.3 Systémy

Firma využívá několik systémů, aby zajistila hladký a efektivní chod. Mezi tyto systémy se řadí široké spektrum nástrojů a procesů, od finančního řízení a správy HR až po komunikaci a ukládání dokumentů.

Zásadní pro řízení projektů a monitorování pracovního výkonu je systém BPM. Ten je využíván pro zapisování odpracovaných hodin a evidenci úkolů. Tento systém poskytuje přesný přehled o vynaloženém čase na konkrétní úkoly, což je nezbytné pro sledování efektivity a pro fakturaci klientům. Kromě BPM využívá BJSW také GitLab, který slouží nejen pro správu verzí a ukládání kódu, ale také pro sledování úkolů (issues) a plánování pracovních úkolů (to-do). GitLab přispívá k transparentnosti a usnadňuje sledování pokroku v rámci projektů.

Pro interní i externí komunikaci společnost upřednostňuje komunikační kanály jako Telegram a Microsoft Teams. Telegram poskytuje rychlý a neformální komunikační prostředek, zatímco Microsoft Teams se využívá pro koordinovanější a strukturovanější výměnu informací, například pro plánování schůzek, sdílení souborů a správu projektových týmů.

Co se týče interních pravidel a procesů, ty si určuje každý projektový tým individuálně, v závislosti na specifických požadavcích a charakteristikách daných úkolů. Tento přístup podporuje agilitu a pružnost, které jsou důležité pro rychlé přizpůsobení se požadavkům klientů a trhu. Pravidla jsou formulována tak, aby byla v souladu s firemními standardy a zároveň poskytovala dostatek prostoru pro inovace a kreativní řešení výzev.

Mimo tyto systémy se ve společnosti také samozřejmě používají nejrůznější systémy a nástroje pro samotný vývoj softwaru. Mezi ty patří například Visual Studio jakožto nejrozšířenější IDE ve společnosti, Docker pro kontejnerizaci, SQL Server Management Studio, SQLite Browser a mnoho dalších. Tyto nástroje jsou nezbytné pro chod společnosti, jelikož by se bez nich nemohl vývoj softwaru uskutečnit.

2.3.1.4 Styl řízení

Firma BJSW uplatňuje styl řízení, který je převážně demokratický, což vytváří otevřené, přátelské a neformální pracovní klima. Vedení firmy je přístupné a podporuje motivaci zaměstnanců, zároveň jim poskytuje volnost v organizaci práce. Team leader určuje priority projektů a úkolů, ale důraz je kladen na samostatnost a iniciativu každého zaměstnance, což je obzvláště ceněno těmi, kteří při práci současně studují na vysokých školách.

Tento styl řízení se firmě vyplácí, jelikož podporuje nejen produktivitu, ale i spokojenost zaměstnanců. Atmosféra spolupráce převyšuje nad soutěživostí, což napomáhá k vytváření silného týmového ducha a podněcuje k výměně nápadů a řešení. Autonomie poskytnutá zaměstnancům a týmům vede k vyšší efektivitě a umožňuje firmě svižně reagovat na změny v požadavcích u jednotlivých zakázek.

Zaměření na flexibilitu a podpora kontinuálního vzdělávání jsou klíčové pro adaptabilitu firmy v dynamickém IT sektoru. Tento přístup nejenže podporuje rozvoj osobních a profesionálních dovedností zaměstnanců, ale zároveň přispívá k udržitelnému růstu firmy. V konečném důsledku firma ukazuje, že v IT odvětví může být flexibilní a zaměstnanecky orientovaná kultura pevným základem pro úspěch společnosti.

2.3.1.5 Spolupracovníci

Ve společnosti tvoří páteř týmů zejména programátoři, jejichž zkušenosti jsou klíčovými při jejich výběru. S ohledem na lokalitu v Brně, významném akademickém centru, firma čerpá z talentů VŠ, nabízejíc tak příležitosti pro absolventy i aktivní studenty. Kritické kompetence zahrnují nejen znalost programování a technologických trendů, ale také schopnost neustálého učení se, pracovitost a týmovou spolupráci.

Team leadeři jsou vybráni nejenom na základě technické zdatnosti, ale také podle schopnosti vést a motivovat tým lidí, správně rozdělit práci a efektivně komunikovat jak vnitřně, tak se zákazníky. Jsou klíčoví pro udržení morálky a spokojenosti týmu, což je

podporováno různými team-buildingovými aktivitami a společnými obědy, které posilují firemní kulturu a soudržnost týmu.

Společnost si cení a podporuje diverzitu, přestože je, podobně jako většina IT společností, složení týmu většinou mužské. Hlavním motivátorem je konkurenceschopná finanční odměna, ale také velmi flexibilní pracovní doba a možnost práce z domu, což je pro mnoho zaměstnanců významný benefit.

V současné době se BJSW zaměřuje na posílení svých týmů a neustále hledá nové kolegy. Zaměřuje se na vytvoření prostředí, kde jsou zaměstnanci odměněni nejen podle výkonu, ale také za jejich příspěvek k inovativnímu a pozitivnímu pracovnímu prostředí. Spokojenost zaměstnanců je pro firmu prioritou, jelikož se snaží své zaměstnance udržet.

2.3.1.6 Schopnosti

Ve společnosti tvoří jádro schopností zaměstnanců především silná programátorská expertíza. K tomu je nezbytný také kompetentní správce systému a sítí, stejně jako organizační dovednosti sekretářky, která zajišťuje každodenní operace firmy. Důraz na rozvoj a učení je patrný v podpoře neustálého vzdělávání a přijímání moderních přístupů k inovacím v rámci projektů.

Firma si cení schopnosti svých zaměstnanců pracovat pod tlakem a rychle reagovat na naléhavé požadavky, což odráží agilní a dynamický charakter pracovního prostředí. Kritické myšlení a problémové řešení jsou důležité dovednosti, které umožňují týmům překonávat problémy.

Anglický jazyk je základní požadavek pro komunikaci s mezinárodní klientelou a partnery, přičemž psaná forma je nezbytná pro koordinaci a dokumentaci projektů. Při výběru nových zaměstnanců je kladen větší důraz na praktické zkušenosti a osobní projekty než na formální akademické vzdělání, což ukazuje na hodnotový systém společnosti, který preferuje reálné dovednosti a prokazatelné výsledky.

Schopnosti v BJSW jsou systematicky sledovány a hodnoceny, aby se zajistilo, že zaměstnanci mají nejen potřebné technické znalosti, ale také aby byly rozvíjeny jejich měkké dovednosti.

2.3.1.7 Sdílené hodnoty

Sdílené hodnoty společnosti představují jakýsi etický kompas, kterým se zaměstnanci řídí ve své každodenní práci. Tyto hodnoty jsou zakotveny v otevřenosti, inovaci, a respektu k individuálním přínosům každého člena týmu a firmy celkově. Firma byla založena na principu kvality, spolehlivosti a transparentnosti, a těchto klíčových vlastností se drží i nadále.

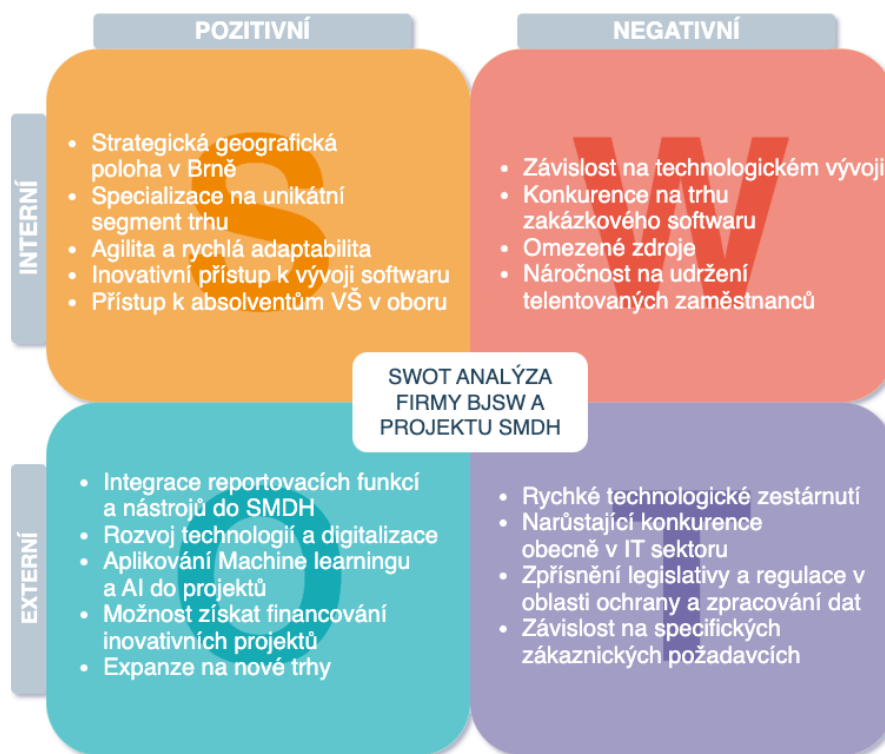
Firemní kultura je charakterizována dynamickým, ale současně podpůrným pracovním prostředím, kde je kladen důraz na samostatnost, ale zároveň týmovou spolupráci a individuální rozvoj. Základní hodnoty BJSW se odrážejí v každodenním jednání – od otevřené komunikace mezi všemi úrovněmi managementu po závazek dodávat klientům včas kvalitní a přesné řešení, i přes případné postupné úpravy ze strany zákazníků. Tyto sdílené hodnoty jsou pravidelně připomínány ve všech firemních aktivitách a jsou integrované do všech aspektů pracovního procesu, od nábory nových talentů až po vedení celých projektů.

2.4 Sumarizace analýz

Kapitola 2.4 se věnuje syntéze a sumarizaci klíčových zjištění z předchozích analýz vnitřního a vnějšího prostředí. Provede se SWOT analýza, která stručně shrne silné a slabé stránky, příležitosti a hrozby, identifikované během analýzy. Tento přístup umožní jasněji definovat strategické směry pro další vývoj a integraci reportingového modulu, zohledňujíc přitom všechny relevantní interní a externí faktory.

2.4.1 SWOT

SWOT analýza je strategický nástroj pro identifikaci Sil (*Strengths*), Slabostí (*Weaknesses*), Příležitostí (*Opportunities*) a Hrozeb (*Threats*) v projektu nebo firmě. Pomáhá organizacím porozumět svému prostředí pro efektivní plánování a rozhodování. V případě BJSW a projektu SMDH nám tato analýza umožňuje identifikovat klíčové oblasti pro využití silných stránek, řešení slabostí, využití příležitostí pro růst a přípravu na potenciální hrozby. Tím podporuje strategické řízení a udržitelný úspěch firmy.



Obrázek 2.3 - SWOT Analýza
(Zdroj: Vlastní zpracování)

2.4.1.1 Silné stránky

Společnost BJSW má hned několik silných stránek, které ji odlišují na trhu a poskytují jí konkurenční výhody, zvláště pak v rámci projektu SMDH. Výhodnou předností je její umístění hlavních prostorů v Brně, akademickém centru s bohatým zdrojem talentovaných absolventů a studentů. Tato strategická poloha umožňuje společnosti nabírat nové kolegy se vzděláním v oboru. Tento přístup podporuje užívání moderních technologií a aktuálních trendů.

Projekt SMDH se specializuje na software pro automatizované pilnice, segment s minimální konkurencí. Díky hlubokým znalostem a technologickému know-how, poskytuje firma vysoce specializovaná řešení, která jdou přesně naproti potřebám klientů a odlišují ji na trhu.

Přestože je BJSW menší společností, její schopnost agilního přizpůsobení se a inovačního přístupu vytváří silný základ pro růst a rozvoj. Zároveň konexe na zahraniční klienty otevírá možnosti budoucí expanze.

2.4.1.2 Slabé stránky

Mezi slabé stránky společnosti BJSW a jejího projektu SMDH patří několik faktorů, které mohou ovlivnit její konkurenceschopnost a efektivitu. Závislost na rychlém technologickém vývoji, představuje první slabinu, jelikož to pro firmu znamená stále investice do inovací a výzkumu. Toto může být zvláště problematické pro menší firmy, jako je BJSW, kde jsou finanční a lidské zdroje více omezené ve srovnání s velkými korporacemi.

Dalším klíčovým omezením je poměrně silná konkurence na trhu zakázkového softwaru, která vyvíjí tlak na firmu v boji o získání a udržení zakázek. I když se projekt SMDH specializuje na specifický segment trhu, obecná konkurence ve vývoji softwaru může firmě znepříjemňovat rozšiřování jejího působení a získávání nových klientů. Zároveň konkurent Kesat je větší firma s působením na širší škále trhů, což může vést k preferenci řešení od této rozsáhlejší společnosti ze strany potenciálních zákazníků.

Současná nutnost manuálního zpracování dat pro tiskové sestavy v rámci projektu SMDH může být také považována za oblast vyžadující pozornost. Primárně z důvodu, že se projekt vyznačuje vysokou mírou přizpůsobení potřebám klientů a je vnímán jako zákaznický orientované řešení.

Rovněž omezené zdroje, jak finanční, tak lidské, mohou bránit společnosti v rychlém škálování a expanzi. Přestože má BJSW přístup k talentovaným absolventům a studentům, díky své lokaci v Brně, získávání, ale především udržení top talentů je v konkurenčním IT odvětví dost náročné.

Tato kombinace faktorů – závislost na neustálých inovacích, silná konkurence a omezené zdroje – tvoří základní slabé stránky, se kterými se BJSW musí vypořádat, aby si udržela svou pozici na trhu a podporovala růst svých projektů.

2.4.1.3 Příležitosti

Příležitosti pro BJSW jsou úzce propojeny s rychlým rozvojem technologií a vzrůstající potřebou digitalizace a automatizace napříč všemi průmyslovými odvětvími. Možnosti získat financování pro inovativní projekty díky legislativním změnám a vládním stimulům představují pro firmu také významné příležitosti. Současně umožňuje využití moderních technologií, jako je cloud computing, umělá inteligence (AI) a internet věcí (IoT), rozšířit funkčnost jejich softwarových řešení a zvýšit jejich tržní přitažlivost.

Specificky pro projekt SMDH představuje integrace reportovacích nástrojů a funkcí do existujícího softwaru klíčovou příležitostí. Tímto krokem by firma vyšla vstříc poptávce zákazníků po automatizaci procesů přípravy dat, což by značně zvýšilo vnímanou hodnotu softwaru a ušetřilo uživatelům čas strávený manuálními operacemi. Implementace reportingového modulu by také posílila konkurenční pozici SMDH, vzhledem k absenci přímých substitutů na trhu, což by firmě umožnilo udržet si náskok před konkurencí. Další příležitostí je aplikace machine learningu a AI pro optimalizaci pracovních postupů, včetně identifikace vadných kmenů, což by přispělo k prevenci závad na strojích. Tato inovace by nejen zefektivnila výrobní procesy, ale také by představovala významný pokrok ve schopnosti softwaru reagovat na specifika provozu.

Tyto inovace by BJSW umožnily nejen posílit svou tržní pozici, ale také nabídnout zákazníkům řešení, která by výrazně překonávala současné možnosti, a tím podpořit jejich operativní efektivitu a konkurenceschopnost.

2.4.1.4 Hrozby

Pro firmu BJSW je jednoznačně hrozbou velmi rychle měnící se technologické prostředí. Tato extrémní rychlost vývoje technologií a obecně trendů v IT vede k tomu, že se vytvořené řešení může stát rychle zastaralé. To vyžaduje od firmy neustálé investice do vývoje a aktualizace. Narůstající konkurence ze strany nových hráčů na trhu může ohrozit její postavení, zejména pokud by tito konkurenti nabízeli lepší nebo levnější alternativy. Zpřísnování legislativy a regulací v oblasti ochrany dat a soukromí může přinést další finanční a operační zátěž. Ekonomické fluktuace a globální události, jako jsou pandemie, představují další nejistotu, která může ovlivnit poptávku po produktech a službách firmy.

Specificky pro projekt SMDH, hlavní hrozby spočívají ve vzniku nových technologií, které by mohly přinést efektivnější nebo inovativnější řešení pro automatizaci pilnic. Rychlá adaptace a implementace těchto nových technologií konkurencí by mohla oslabit jedinečnou tržní pozici, kterou SMDH aktuálně má. Kromě toho, závislost projektu na specifických zákaznických požadavcích a tržních segmentech přináší riziko, že v případě změny tržních trendů nebo zákaznických preferencí by mohl projekt ztratit svou relevanci. Udržení technologického náskoku a poskytování nepřetržité hodnoty v rámci projektu SMDH tak vyžaduje od firmy proaktivní přístup k inovacím a schopnost rychle reagovat na nové výzvy v průmyslu.

2.4.1.5 Výsledek SWOT

SWOT analýza pro BJSW a její projekt SMDH ukazuje, že integrace reportovacího modulu je strategicky výhodná. Tento krok by posílil postavení produktu na trhu tím, že by odpovídal na poptávku zákazníků po efektivnějším zpracování dat, zvýšil by uživatelský komfort a rozšířil funkční spektrum SMDH. Silné stránky firmy jako jsou specializace a technologická zdatnost poskytují pevný základ pro inovaci, zatímco vnější financování a technologický pokrok představují příležitost k dalšímu rozvoji. Navzdory hrozbám rychlého technologického přepracování a zvýšené konkurence, vyplývající z analýzy, že rozvoj reportovacího modulu je klíčový pro udržení a posílení konkurenční výhody SMDH.

2.5 Použité programovací jazyky, frameworky a knihovny

Projekt byl vyvinut s využitím .NET Frameworku verze 4.7.2 a programovacího jazyka C# ve verzi 7.3, což zajišťuje stabilní výkon a rozsáhlou podporu od Microsoftu. Tato kombinace byla zvolena pro svou spolehlivost a kompatibilitu s různými systémovými nastaveními, což usnadňuje integraci s dalšími nástroji a knihovnami.

Logování v aplikacích bylo implementováno pomocí nástroje Log4Net, který je známý pro svou výkonnost a flexibilitu. Díky této knihovně je možné detailně sledovat provoz aplikace a efektivně podporovat její ladění a optimalizaci.

Pro tvorbu uživatelského rozhraní byl vybrán framework WinForms. Ačkoli existují modernější alternativy jako WPF, .NET MAUI nebo Avalonia, preference byla dána WinForms pro jeho osvědčenou spolehlivost a bohatou podporu, což je zásadní pro vývoj stabilních aplikací v průmyslovém prostředí. Efektivní cesta pro rychlý vývoj a nasazení aplikací byly klíčové, zvláště v době, kdy byl projekt zahájen před oznámením .NET MAUI a plným rozvojem Avalonie. Preference WinForms byla rozhodnuta na základě specifických požadavků na adaptabilitu a jednoduchost implementace.

Pro tvorbu a design samotného UI byla zvolena knihovna DevExpress, která umožňuje tvorbu esteticky přitažlivých a funkcionalitou bohatých UI komponent. Volba této knihovny byla motivována její vysokou adaptabilitou a schopností vytvářet intuitivní a vizuálně atraktivní uživatelské rozhraní. Komponenty od DevExpressu představují

modernější a vizuálně přitažlivější alternativu k základním prvkům v ToolBoxu WinForms.

Celkově byl výběr technologického stacku pro projekt zaměřen na maximální efektivitu a minimalizaci rizik při implementaci v existujícím průmyslovém prostředí. Tento přístup zaručuje, že rozšíření stávajícího systému proběhne s minimálními náklady na čas a zdroje.

2.6 Databázová integrace a správa

V rámci projektu byla zvolena kombinace technologií pro správu a ukládání dat, které jsou kritické pro funkčnost a spolehlivost celého systému. Lokální data jsou ukládána v databázích SQLite, což umožňuje efektivní a bezpečné ukládání dat přímo na jednotlivých pracovních stanicích bez nutnosti složité databázové infrastruktury. Tento výběr byl učiněn na základě potřeby rychlého přístupu k datům, snadného nasazení a zálohování dat.

Pro situace, kdy je nutné sdílet data napříč více měřicími stanicemi, je využíváno databázové řešení MySQL. Tato technologie byla zvolena pro svou robustnost, vysokou dostupnost a schopnost zvládat velké objemy dat, které jsou sdílené mezi různými částmi systému. Synchronizace dat mezi lokálními databázemi SQLite a centrálním úložištěm MySQL je klíčová pro integritu a konzistenci dat, což je zásadní pro správné fungování aplikací.

Nástroje jako MariaDB, SQL Server Management Studio a různé DB browsery jsou používány především pro prohlížení a ruční ladění databází a dotazů. Tyto nástroje poskytují uživatelsky přívětivé rozhraní pro monitorování stavu a výkonu databází, což usnadňuje diagnostiku a řešení potenciálních problémů.

Samotná správa databází, včetně aktualizací a správy verzí, je integrována přímo do kódu aplikace napsaného v C#. Tento přístup zajišťuje, že každá nová verze aplikace automaticky zkontroluje aktuální verzi databáze a případně provede potřebné aktualizace pomocí předem definovaných SQL dotazů. Tento přístup nabízí několik výhod: zvyšuje konzistenci a bezpečnost tím, že minimalizuje riziko lidských chyb a zajišťuje, že databázové operace jsou prováděny konzistentně a bezpečně. Automatizace aktualizací zjednodušuje nasazování nových verzí aplikace, což snižuje náklady na údržbu a zvyšuje

efektivitu operací. Navíc, systémová kontrola a aktualizace databází přímo z aplikace zajišťuje, že databáze je vždy synchronizována s nejnovější verzí aplikace, což zabraňuje potenciálním problémům s kompatibilitou.

Celkově je tedy databázová architektura projektu navržena tak, aby poskytovala vysokou úroveň výkonu a flexibility, které jsou nezbytné pro spolehlivý provoz na pilnicích.

2.7 Architektura a design aplikací

Jak již bylo nastíněno v Úvodu, tato práce se zabývá projektem nazvaným SMDH¹ (*Systémy Měření Dřevní Hmoty*), který se specializuje na elektronické měření a manipulaci s dřevem. Tento projekt zahrnuje několik aplikací, které dohromady vytváří software pro automatizované pilnice. Tyto produkty zahrnují aplikace pro samotné měření a manipulaci s dřevní hmotou, dále aplikace pro pásové dopravníky, omítací a pásové pily. Tyto aplikace jsou navrženy pro zvýšení efektivity a produktivity ve dřevozpracujícím průmyslu prostřednictvím automatizace a pokročilé analýzy dat.

2.7.1 Systém měření a manipulace dřevní hmoty II (SMMDH2)

Aplikace *SMMDH2* je navržena pro automatizaci manipulace a evidenci dřevní hmoty, zvyšující efektivitu práce v manipulačních skladech. Systém využívá bezkontaktní elektronické měření pro poskytování přesných údajů o dřevní surovině, jako jsou délka, průměr, kubatura, a křivost, přímo na ovládacím panelu. Zaznamenává data o každém kmeni a výřezu do databáze, což usnadňuje následnou manipulaci a umožňuje detailní sledování a analýzu manipulačních procesů. Tato technologie nejen zjednodušuje manipulaci s dřevem, ale také přispívá k optimalizaci skladových procesů a umožňuje efektivní využití dřevní suroviny.

2.7.2 Systém měření pro pásové dopravníky (SMPD)

Tato aplikace je navržena pro efektivní měření kmenů a výřezů posunovaných pomocí pásových dopravníků nebo jiných přepravních systémů, umožňující detailní evidenci každého kusu. Systém nabízí pracovníkům okamžitý přehled o množství a objemu zpracovaného materiálu během směny pomocí zobrazení na světelném panelu. Toto

¹ Webové stránky projektu jsou dostupné na: www.merenidreva.cz

zařízení, umístěné na vstupu do pilnice, je klíčové pro monitorování produktivity a umožňuje vedoucím výroby efektivně sledovat dodržování výrobních plánů a identifikovat případné prodlevy ve zpracování.

2.7.3 Systém měření pro pásové pily (SMPP)

Aplikace *SMPP* je určena pro přesné měření objemu a počtu kmenů během jejich zpracování na pásové pile. S měřicím zařízením instalovaným přímo u pásové pily je zajištěno, že každý zpracovaný kmen je pečlivě evidován. Toto umožňuje vedoucím výroby efektivně monitorovat průběh výroby a zajistit dodržení plánované produkce, zvyšující celkovou produktivitu a minimalizující zpoždění ve zpracování materiálu.

2.7.4 Systém pro měření a manipulaci pro omítací pily (SMMOP)

SMMOP automatizuje proces omítání dřevěných desek tím, že před zpracováním změří jejich povrch a na základě těchto údajů nastaví šířku omítacích pil. Tento plně automatický systém zaznamenává každý změřený kus do databáze a umožňuje pracovníkům sledovat výkon práce skrze celkový počet kusů a objem zpracovaného materiálu za směnu na světelném displeji. Takto zefektivňuje pracovní procesy a zvyšuje produktivitu.

Každý z produktů společnosti přináší specifické výhody pro procesy v dřevozpracujícím průmyslu, od přesného měření kmenů po automatizaci omítacích procesů až po efektivní zpracování materiálu. Tyto technologie umožňují podnikům nejen zvýšit produktivitu, ale také optimalizovat výrobní postupy. Integrace těchto produktů do výrobních linek podporuje modernizaci a digitalizaci, což vede k lepší přesnosti, vyšší efektivitě a minimalizaci odpadu.

2.8 Testování, kvalita a správa verzí softwaru

V rámci projektu SMDH je kladen důraz na kvalitu softwaru a systematické testování. Základem správy kódu je využití platformy GitLab, kde je pro každý úkol vytvořena nová větev odvozená z hlavní developerské větve (*develop*). Tento přístup umožňuje efektivně implementovat změny a nové funkce s minimálním rizikem pro stávající systém.

Po implementaci změn ve vlastní větvi jsou tyto změny následně integrovány zpět do hlavní větve pomocí procesu *merge request*. Merge request je bodem, kde dochází ke kontrole kódu team leaderem. Během tohoto procesu může dojít k diskusi o implementaci, kde se probírají důvody a metodiky použité při vývoji, což zajišťuje transparentnost a pochopení změn všemi členy týmu. Může být vyžádán i názor nebo revize od dalších členů týmu, což podporuje kolektivní odpovědnost za kvalitu a koherenci kódu.

Schválení merge requestu znamená, že změny jsou připraveny k integraci do hlavní develop větve. Dále se z této větve odvíjejí větve pro konkrétní verze softwaru, nebo se v ní řeší specifické funkce či opravy chyb, které se testují přímo ve větvích dané verze.

Manuální testování probíhá kontinuálně během vývoje v rámci jednotlivých větví, což umožňuje vývojářům rychle identifikovat a opravit chyby předtím, než dojde k jejich integraci do hlavní větve. Kromě toho se provádějí automatizované unit testy pomocí frameworku NUnit, který ověřuje funkčnost jednotlivých komponent nezávisle na ostatních částech systému. Tento přístup zajišťuje, že nově implementované funkce neovlivňují existující funkcionalitu systému.

Při každém commitu, který je nahrán (*pushnut*) do GitLabu, dochází k automatickému sestavení projektu a spuštění unit testů na serveru. Tento proces zahrnuje spuštění všech unit testů a pokud všechny testy projdou, lze změny bezpečně integrovat do develop větve. Tímto způsobem je zajištěno, že opravdu všechny změny musejí projít testy, před začleněním do stávajícího prostředí.

3 VLASTNÍ NÁVRH ŘEŠENÍ

V rámci této diplomové práce byl na základě výsledků předcházejících analýz a požadavků klientů navržen a implementován reportovací modul pro projekt SMDH, nazvaný SMDH Report. Tento modul poskytuje rozšířené reportovací a editační funkce, které jsou klíčové pro efektivní monitorování a řízení procesů v automatizovaných pilnicích. Z důvodu komplexnosti projektu a požadované efektivity byla práce na aplikaci rozdělena mezi členy týmu. Tento přístup umožnil efektivnější detekci a řešení problémů, které by mohly být při samostatné práci přehlédnuty.

Vývoj aplikace byl založen na hybridním modelu SDLC, kde se počáteční fáze řídila metodikou podobnou vodopádovému modelu. Tato fáze byla zvolena z důvodu již známého designu a stylu aplikace. V následujících fázích, zejména při vytváření reportů, byl přístup více agilní a charakterizoval se intenzivním zapojením zákazníků. Reporty byly upravovány a optimalizovány tak, aby co nejvíce odpovídaly zpětné vazbě od zákazníků, což umožňovalo lépe reflektovat jejich aktuální požadavky a představy o funkcionalitě.

Modularita systému pak umožnila využití jednotlivých komponent i v dalších aplikacích. Moduly byly navrženy s důrazem na opakovanou použitelnost kódu, což znamená, že každý modul může být snadno integrován do různých částí projektu SMDH nebo využit v budoucích projektech. Tento přístup nejen zvyšuje flexibilitu a škálovatelnost aplikace, ale také přispívá k ekonomičnosti vývoje, protože umožňuje znovuvyužití již vyvinutých řešení. Celkově byla aplikace SMDH Report navržena tak, aby odpovídala specifickým potřebám klienta a zároveň přinášela vysokou míru adaptability pro budoucí rozvoj a integraci.

3.1 Požadavky a očekávání

V této kapitole jsou specifikovány hlavní požadavky na aplikaci SMDH Report, které byly identifikovány na základě analýz potřeb klientů a uživatelských interakcí. Aplikace SMDH Report, jakožto integrální součást projektu pro automatizované pilnice, byla navržena tak, aby uspokojila specifické nároky klientů v oblasti datové správy a reportingu.

Základním požadavkem je schopnost aplikace procházet data uložená v databázi, přičemž je důležité, aby UI bylo konzistentní s ostatními aplikacemi SMDH, ale zároveň bylo uzpůsobeno používání v kancelářském prostředí. To zahrnuje adaptaci UI pro ovládání pomocí klávesnice a myši, se zachováním možnosti dotykového ovládání pro maximální flexibilitu použití.

Dalším klíčovým aspektem je možnost filtrování dat podle různých kritérií, které umožňují uživatelům efektivně třídit a analyzovat informace relevantní pro jejich specifické potřeby. Aplikace by také měla umožňovat uživatelům vytváření dokumentů z vybraných dat, jako jsou tiskové sestavy, primárně ve formátech PDF, ale také v CSV a XML formátech. Tato funkcionality by měla být optimalizována pro snadné zobrazení dokumentů na různých zařízeních, včetně PC a mobilních telefonů.

V případě, že na dané pile existuje více měřících míst, je vyžadováno, aby SMDH Report umožňoval prohlížení dat ze všech těchto míst, což značně rozšiřuje schopnosti monitorování a analýzy. Aplikace by měla rovněž nabízet nástroje pro základní úpravy dat, jako je hromadná editace vlastností kmenů, což uživatelům umožní korigovat například kvalitu dřeva, která byla původně špatně určena na měřících místech, přímo v reportovacím modulu, bez nutnosti složitějšího zasahování do databáze.

Důležitá je i možnost správy informací o kmenu, včetně mazání nebo zobrazení komplexních dat z databáze. SMDH Report by měl zahrnovat i funkce pro zobrazování informací o skládkách na konkrétní pilnici, jako je počet kmenů na skládce a zpřístupnění podrobných dat o dodávkách, včetně čísel dodacích listů a evidenčních čísel dodávek. Zároveň by mělo být možné nahlížet na a celkově spravovat dodavatele přímo z aplikace.

Takhle definované požadavky a očekávání jsou nezbytné pro návrh a vývoj aplikace SMDH Report, a zajišťují, že výsledný produkt bude odpovídat potřebám klientů.

3.2 Analýza rizik

Tato kapitola se zabývá systematickým přístupem k identifikaci, hodnocení a managementu potenciálních rizik, která by mohla ovlivnit průběh vývoje či implementace aplikace SMDH Report. Analýza rizik je zásadní pro minimalizaci možných negativních dopadů na projekt, což zahrnuje identifikování možných hrozeb a přípravu efektivních opatření pro jejich řešení. Tato kapitola je strukturována do dvou

hlavních částí, které se zaměřují na identifikaci a kvantifikaci rizik a na formulaci odpovídajících opatření.

3.2.1 Identifikace a ohodnocení rizik

Tato část práce se zabývá systematickou identifikací a kvantifikací rizik spojených s vývojem a implementací aplikace SMDH Report. Proces začíná identifikací potenciálních rizik, která mohou projekt ovlivnit, a jejich scénářů, popisujících možné důsledky. Tento krok je nutný pro pochopení a předvídání rizik, které mohou nastat během realizace projektu. Následující tabulka tedy ke každému identifikovanému riziku uvádí také identifikační číslo a scénář.

Tabulka 3.1 - Identifikace rizik
(Zdroj: Vlastní zpracování)

<i>ID</i>	<i>Riziko</i>	<i>Scénář</i>
<i>R1</i>	Nedostatečná kompatibilita s existujícími systémy	Integrace s jinými aplikacemi SMDH je komplikovaná nebo nemožná, což vede k omezení funkčnosti nebo nutnosti drahých úprav.
<i>R2</i>	Chyby ve vývoji software	Během vývoje se narazí na technické problémy, které způsobují zpoždění nebo nutnost redesignu částí systému.
<i>R3</i>	Omezená dostupnost lidských zdrojů	V průběhu vývoje projektu se bude nutné věnovat i jiným projektům, které budou mít přednost a budou muset být dokončeny dříve. To může vést k prodloužení doby vývoje a celkovému zpoždění projektu.
<i>R4</i>	Překročení rozpočtu	Náklady na vývoj, testování a integraci přesáhnou původní rozpočet, což vyvíjí tlak na finanční zdroje firmy.
<i>R5</i>	Neočekávané změny ve specifikacích projektu	Změny ve specifikacích vyžadují náhlé úpravy v plánu, což způsobuje zpoždění a zvýšení nákladů.

R6	Odmítnutí uživatelů přijmout nový systém	Uživatelé mají odpor k používání nového systému z důvodu nedostatečného školení nebo neochoty měnit zavedené postupy.
R7	Nedostatek odborných znalostí u týmu	Tým nemá dostatečné znalosti pro řešení specifických technických problémů, jako jsou specifické PDF experty, což vede k chybám ve vývoji a prodloužení doby projektu.
R8	Ukončení podpory externí knihovny pro práci s PDF, na které je export založený	Závislost na externích knihovnách, které přestanou být podporovány přináší rizika jak bezpečnostní, tak rizika, že knihovna nebude korektně vytvářet PDF soubory.
R9	Nedostatečná škálovatelnost systému pro budoucí potřeby	S rostoucím počtem uživatelů nebo objemem dat systém nedokáže efektivně zpracovávat zátěž.
R10	Zanedbání uživatelské zpětné vazby při vývoji	Ignorování zpětné vazby od uživatelů během vývojového procesu vede k vytvoření produktu, který neodpovídá potřebám trhu.

Následně je provedena kvantifikace každého identifikovaného rizika ve dvou klíčových dimenzích – subjektivní míry pravděpodobnosti výskytu a potenciální dopad na projekt. Toto hodnocení je zpracováno pomocí skórovací metody, která umožňuje analyzovat rizika a stanovit jejich priority.

Pro samotnou kvantifikaci rizik jsou využity dvě pomocné tabulky. První tabulka hodnotí subjektivní míru pravděpodobnosti výskytu jednotlivých rizik na desetibodové stupnici, zatímco druhá tabulka hodnotí potenciální dopad těchto rizik na projekt, také na desetibodové stupnici. Výsledná tabulka 3.4 zobrazuje pro každé riziko přiřazenou míru pravděpodobnosti a dopad, a také celkovou hodnotu rizika, která je vypočtena vynásobením subjektivní míry pravděpodobnosti a dopadu rizika.

Tabulka 3.2 - Ohodnocení míry pravděpodobnosti výskytu rizika
(Zdroj: Vlastní zpracování)

<i>Hodnota</i>	<i>Procentuálně</i>	<i>Slovní hodnocení</i>
1 - 2	1 % - 19 %	Téměř žádná
3 - 4	20 % - 39 %	Nízká
5 - 6	40 % - 59 %	Pravděpodobná
7 - 8	60 % - 79 %	Více pravděpodobná
9 - 10	80 % - 100 %	Vysoce pravděpodobná

Tabulka 3.3 - Ohodnocení dopadu rizika
(Zdroj: Vlastní zpracování)

<i>Hodnota</i>	<i>Slovní popis</i>
1 - 2	Minimální
3 - 4	Málo významný
5 - 6	Významný
7 - 8	Velmi významný
9 - 10	Kritický

Tabulka 3.4 - Kvantifikace rizik
(Zdroj: Vlastní zpracování)

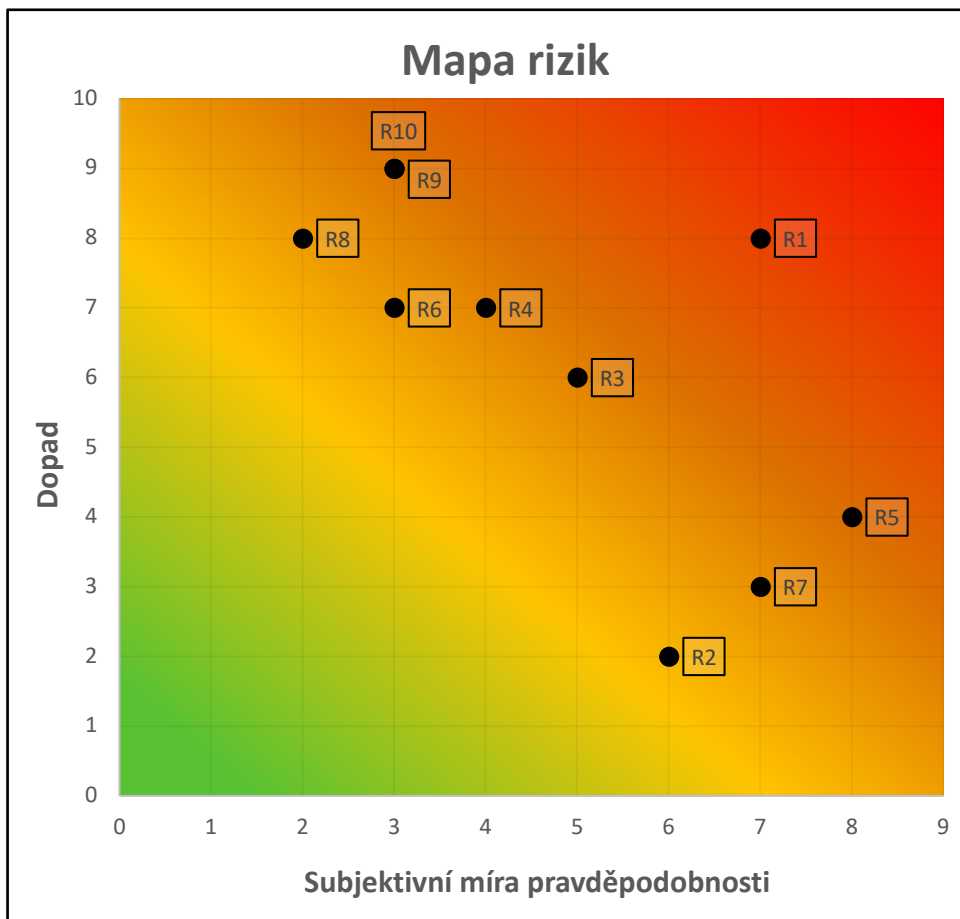
ID	Riziko	SMP²	Dopad	Hodnota rizika
R1	Nedostatečná kompatibilita s existujícími systémy	7	8	56
R2	Chyby ve vývoji software	6	2	12
R3	Omezená dostupnost lidských zdrojů	5	6	30
R4	Překročení rozpočtu	4	7	28
R5	Neočekávané změny ve specifikacích projektu	8	4	32
R6	Odmítnutí uživatelů přijmout nový systém	3	7	21
R7	Nedostatek odborných znalostí u týmu	7	3	21
R8	Ukončení podpory externí knihovny pro práci s PDF, na které je export založený	2	8	16
R9	Nedostatečná škálovatelnost systému pro budoucí potřeby	3	9	27
R10	Zanedbání uživatelské zpětné vazby při vývoji	3	9	27

Tento proces umožňuje nejen identifikovat a kvantifikovat rizika, ale také poskytuje objektivní základ pro jejich další analytické zpracování. Výsledná tabulka tedy poskytuje přehled o celkových hodnotách každého rizika, což umožňuje efektivní monitorování a případnou reakci na vzniklé situace během dalších fází projektu. V této fázi jsou data

² Subjektivní míra pravděpodobnosti

připravena a strukturována tak, aby byla přehledná a snadno interpretovatelná pro všechny zainteresované strany projektu.

Na následujícím obrázku je zobrazena mapa rizik, která poskytuje vizuální reprezentaci hodnot rizik a umožňuje rychle identifikovat oblasti, které vyžadují zvýšenou pozornost. V této mapě jsou rizika rozmístěna podle stupně subjektivní míry pravděpodobnosti jejich výskytu a potenciálního dopadu na projekt. Rizika s vysokou hodnotou jsou umístěná v pravém horním rohu, což jsou rizika, na která je třeba se zaměřit prioritně, neboť kombinují vysokou míru pravděpodobnosti s velkým dopadem. Naproti tomu rizika v levém dolním rohu představují menší hrozbu. Mapa rizik je tedy nástrojem, který napomáhá v efektivním rozhodovacím procesu a vede k proaktivnímu přístupu v řízení rizik.



Obrázek 3.1 - Mapa rizik
(Zdroj: Vlastní zpracování)

3.2.2 Opatření k rizikům

Tato část práce definuje konkrétní opatření, jejichž účelem je minimalizace rizik vyplývajících z implementace aplikace. Opatření jsou cílena na snižování míry pravděpodobnosti vzniku problémů a omezení negativních dopadů, které by mohly vývoj aplikace ohrozit. Tabulka 3.5 popisuje každé identifikované riziko a přiřazuje k němu vhodná opatření. Tato opatření jsou navržena tak, aby byla prakticky aplikovatelná.

Tabulka 3.5 - Opatření k rizikům

(Zdroj: Vlastní zpracování)

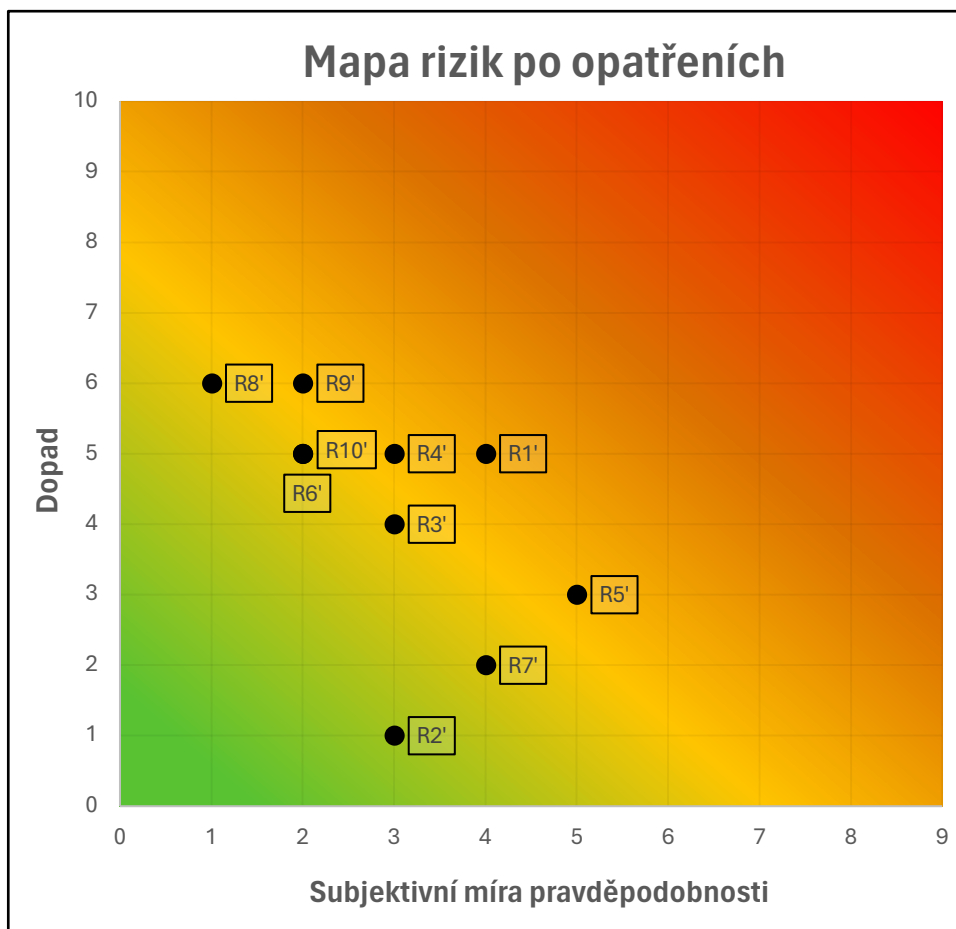
<i>ID</i>	<i>Riziko</i>	<i>Opatření</i>
<i>R1</i>	Nedostatečná kompatibilita s existujícími systémy	Detailní analýza kompatibility a vývoj adaptérů pro integraci
<i>R2</i>	Chyby ve vývoji software	Zavedení agilních metodik a průběžného testování
<i>R3</i>	Omezená dostupnost lidských zdrojů	Plánování zdrojů a zapojení externích programátorů
<i>R4</i>	Překročení rozpočtu	Přísnější kontrola rozpočtu a podrobnější plánování nákladů
<i>R5</i>	Neočekávané změny ve specifikacích projektu	Flexibilní projektový management a zavedení pravidelných re-validace požadavků
<i>R6</i>	Odmítnutí uživatelů přijmout nový systém	Rozsáhlé školení uživatelů a demo se zákazníky
<i>R7</i>	Nedostatek odborných znalostí u týmu	Vzdělávání a rozvoj týmu, případně najmutí expertů
<i>R8</i>	Ukončení podpory externí knihovny pro práci s PDF, na které je export založený	Vývoj vlastního řešení nebo migrace na jinou LTS knihovnu
<i>R9</i>	Nedostatečná škálovatelnost systému pro budoucí potřeby	Architektonický návrh se zaměřením na škálovatelnost od začátku
<i>R10</i>	Zanedbání uživatelské zpětné vazby při vývoji	Implementace procesu pro sběr a začlenění zpětné vazby do vývojového cyklu

Po aplikaci navrhovaných opatření je nezbytné znovu ohodnotit rizika. V následující tabulce jsou prezentována původní rizika, nyní ale s čarou a jsou k nim přiřazeny aktualizované hodnoty míry pravděpodobnosti výskytu a dopadu po aplikaci opatření. Tabulka umožňuje porovnání původních a revidovaných údajů a poskytuje tak přehled o snížení celkové hodnoty rizika.

Tabulka 3.6 - Kvantifikace rizik po opatřeních
(Zdroj: Vlastní zpracování)

ID	Riziko	Nová SMP	Nový dopad	Nová hodnota
<i>R1'</i>	Nedostatečná kompatibilita s existujícími systémy	4	5	20
<i>R2'</i>	Chyby ve vývoji software	3	1	3
<i>R3'</i>	Omezená dostupnost lidských zdrojů	3	4	12
<i>R4'</i>	Překročení rozpočtu	3	5	15
<i>R5'</i>	Neočekávané změny ve specifikacích projektu	5	3	15
<i>R6'</i>	Odmítnutí uživatelů přijmout nový systém	2	5	10
<i>R7'</i>	Nedostatek odborných znalostí u týmu	4	2	8
<i>R8'</i>	Ukončení podpory externí knihovny pro práci s PDF, na které je export založený	1	6	6
<i>R9'</i>	Nedostatečná škálovatelnost systému pro budoucí potřeby	2	6	12
<i>R10'</i>	Zanedbání uživatelské zpětné vazby při vývoji	2	5	10

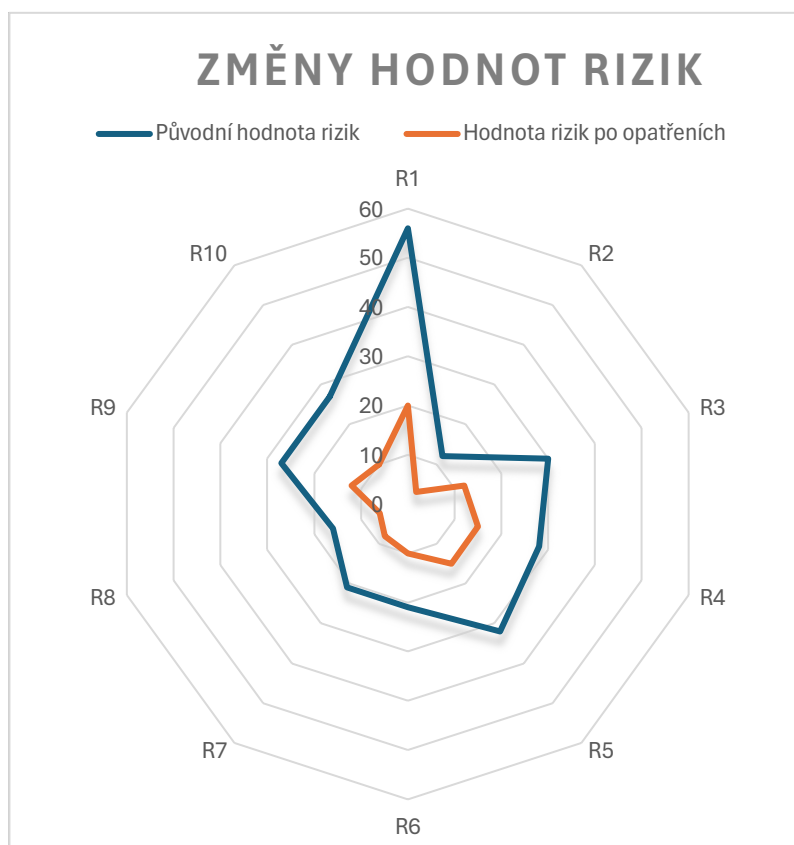
Aktualizovaná mapa rizik po zavedení opatření poskytuje vizualizaci úspěchu těchto opatření v kontextu snižování rizik. Tato mapa ukazuje nově ohodnocená rizika s upravenými hodnotami míry pravděpodobnosti a dopadu. Je možné sledovat posuny jednotlivých rizik směrem k nižším hodnotám, což naznačuje snížení celkového rizika projektu. Vizualizace umožňuje rychlou a intuitivní identifikaci změn v profilu rizik a efektivitu zavedených řízení. Je zde zřetelné, která rizika byla efektivně mitigována a která si stále vyžadují pozornost, umožňující tak zainteresovaným stranám zaměřit se na oblasti, jež vyžadují další zlepšení.



Obrázek 3.2 - Mapa rizik po aplikování opatření
(Zdroj: Vlastní zpracování)

Závěrečná vizualizace opatření k rizikům je prezentována pomocí pavučinového grafu, který demonstruje změny v hodnotách rizik před a po zavedení opatření. Graf ilustruje snížení rizik srovnáním původních a aktualizovaných hodnot, kde každý vrchol reprezentuje jedno konkrétní riziko. Snížený obvod a plocha vnitřního polygonu v grafu

reprezentují efektivní redukci rizikové expozice projektu a poskytují přehled o celkovém snížení rizikosti projektu. Tento graf slouží jako přehledný nástroj pro hodnocení úspěšnosti zavedených opatření a zajišťuje, že rizika jsou řízena efektivně a v souladu s cíli projektu.



Obrázek 3.3 - Pavučinový graf hodnot rizik před a po aplikaci opatření
(Zdroj: Vlastní zpracování)

3.3 Aplikace SMDH Report

Vývoj aplikace si nejdříve vyžaduje nastavení efektivního vývojového prostředí. Přestože příprava development environmentu, zahrnující instalaci nástrojů a software jako je Visual Studio, Git a dalších, spolu s konfigurací ostatních aplikací SMDH, je nezbytným krokem, nebudou tyto počáteční fáze dále podrobně rozebírány, jelikož jsou považovány za standardní část vývojového cyklu a nejsou primárním předmětem této práce.

Hlavním cílem bylo vytvoření projektu WinForms v rámci řešení SMDH (*smdh.sln*), kde prvním úkolem bylo navržení základního uživatelského formuláře (*formu*), což

definovalo UI aplikace. Spolu s UI byla naprogramována back-endová část tohoto formu, tak aby veškeré elementy správně fungovaly. Následovalo etapové rozvíjení klíčových funkcí aplikace, jako jsou prohlížení a filtrování dat, úpravy a správa dat, stejně jako exportování a reportování dat.

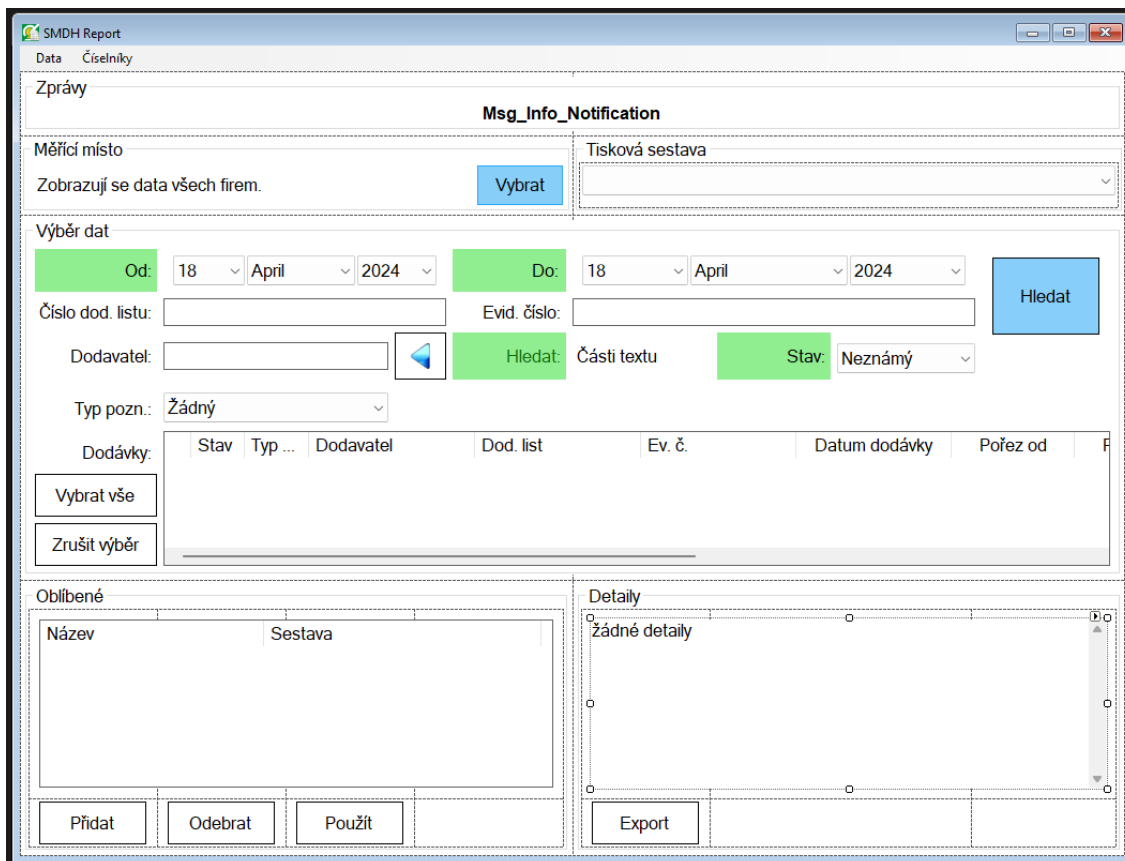
Designem a implementací těchto částí se zabývají následující podkapitoly, kde každá sekce popisuje realizaci jednotlivých funkcí s ohledem na požadavky klientů a technické možnosti systému.

3.3.1 ReportForm

Základním prvkem uživatelského rozhraní aplikace SMDH Report je hlavní formulář pojmenovaný ReportForm (pro dodržení konvence s ostatními aplikacemi projektu). Po spuštění aplikace je jako první inicializován *Program.cs*, který slouží jako vstupní bod a jehož primární a jedinou funkcí je zobrazit samotný ReportForm. Tento formulář pak představuje jádro aplikace, z něhož se řídí další interakce uživatele a z něhož mohou být spuštěna další podokna a nástroje.

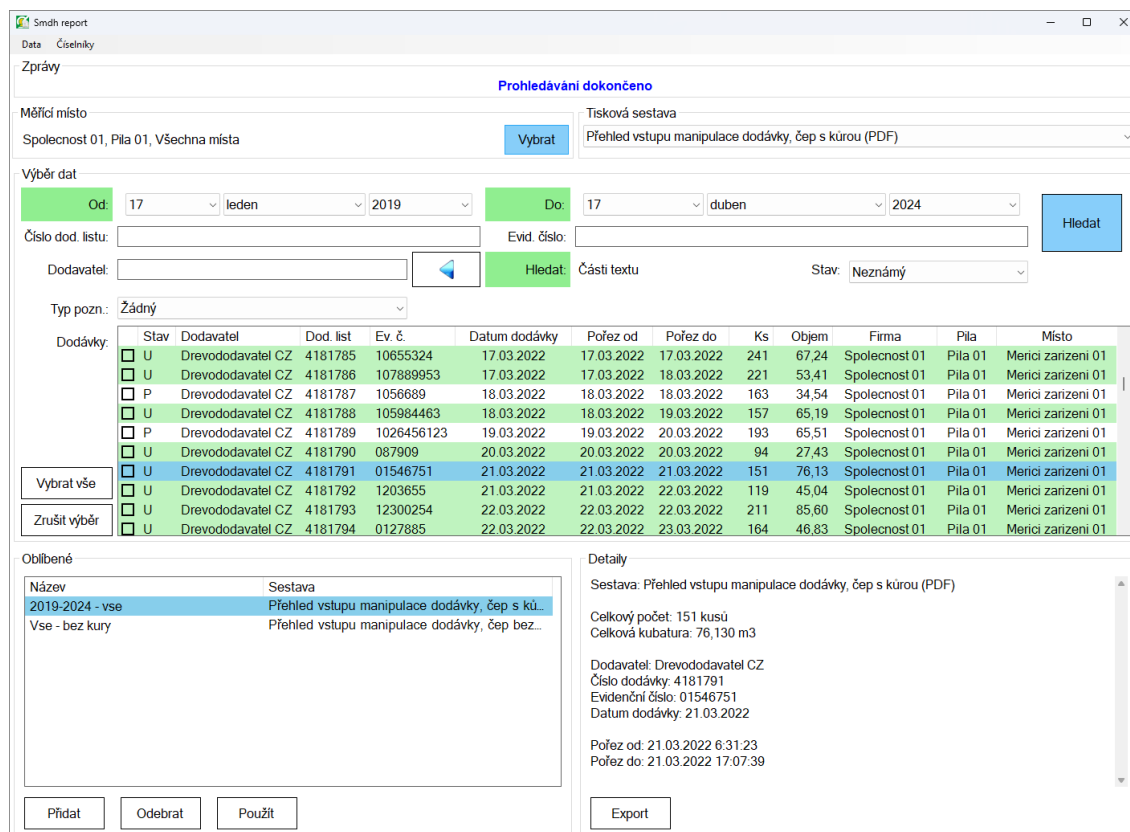
Uživatelské rozhraní ReportForm je navrženo tak, aby umožňovalo intuitivní navigaci a efektivní práci s daty. Jak je patrné ze screenshotu níže, rozhraní je rozčleněno do několika logických částí – komponent, kterým se nazývají *UserControls*. Tento přístup umožňuje modularitu, kde každý *UserControl* může fungovat jako nezávislá jednotka s vlastní logikou a designem, což usnadňuje údržbu kódu a potenciální opětovné použití ve větším projektu. Velikou výhodou je to, že následně není potřeba mít kód pro všechny elementy daného formu v kódu formu, ale logika je v kódu daných *UserControlů* a díky tomu je back-end formu výrazně jednodušší a přehlednější.

Hlavní formulář ReportForm je uspořádán do *TableLayoutPanelu*, což je kontejner, který zajišťuje logické a vizuální uspořádání UI. *TableLayoutPanel* v projektu hraje nenahraditelnou roli, jelikož poskytuje flexibilitu při škálování aplikace na různé velikosti obrazovky a zajišťuje, že každý prvek zůstane správně uspořádán a čitelný bez ohledu na rozlišení nebo velikost okna. Na přiloženém screenshotu níže jsou jednotlivé *UserControly* v *TableLayoutPanelu* ohraničeny čerchovanou čarou, což vizuálně demonstruje jejich rozdělení a funkčnost.



Obrázek 3.4 - ReportForm a rozložení UI elementů v editoru
(Zdroj: Vlastní zpracování)

Použití komponent typu *UserControl* je zásadní nejen pro modularitu a opakované využití kódu, ale také pro zachování konzistentního vzhledu a UX napříč celým souborem aplikací SMDH. Tento přístup zajišťuje, že i přes různé funkcionality jednotlivých aplikací SMDH zůstává uživatelské rozhraní sjednocené, což přispívá k celkové soudržnosti a intuitivnosti ovládání. Tato unifikace je patrná například na prvcích tlačítka typu *TouchButton*, která jsou stejná, jako ty v měřicích aplikacích a která jsou navržena pro efektivní ovládání i na dotykových displejích. Toto využívání již použitých elementů usnadňuje uživatelům navigaci a zvyšuje efektivitu práce s aplikací (zejména při prvotním použití) a zároveň zajišťují soudržnost celého softwarového balíčku.

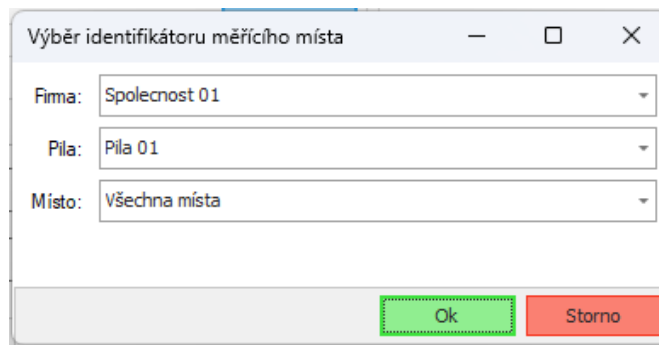


Obrázek 3.5 - Schéma zobrazující ReportForm spuštěné aplikace
(Zdroj: Vlastní zpracování)

Hned pod lištou aplikace se nachází *MenuStrip*, který poskytuje navigaci podoken aplikace, které jsou přiblíženy v nadcházejících kapitolách.

Bezprostředně pod *MenuStripem* je umístěna sekce pro zprávy, sloužící jako centrum oznámení. Tento prostor je určen k výstupu důležitých informací z aplikace, jako jsou notifikace o průběhu akcí – například přihlášení uživatele, procesu vyhledávání v databázi, oznámení o dokončení hledání, potvrzení o exportu souborů a další. Tento element je integrován s cílem zajištění okamžité zpětné vazby pro uživatele.

V ReportFormu aplikace je pod oblastí notifikací umístěn selektor měřících míst – *GlobalMeasurePlace (GMP)*, který je nezbytný pro správnou funkcionality systému. GMP slouží jako unikátní identifikátor každé instalace SMDH, specifikující společnost, působiště pily a měřící místo, což dohromady definuje, jaká data bude aplikace zobrazovat. Tato identifikace zajišťuje, že uživatel má přístup k relevantním informacím, které odpovídají jeho specifickému pracovnímu prostředí a měřící stanici.



Obrázek 3.6 - Okno výběru identifikátoru GMP
(Zdroj: Vlastní zpracování)

GMP je navrženo tak, aby reflektovalo strukturu společnosti, umožňovalo sdílení důležitých informací napříč různými částmi organizace a zároveň chránilo data před neautorizovaným přístupem z jiných míst. Díky tomu, že GMP určuje přístup k datům a operacím, hraje zásadní roli v koordinaci práce mezi stanicemi a v zachování konzistence informací v rámci celého systému SMDH.

```

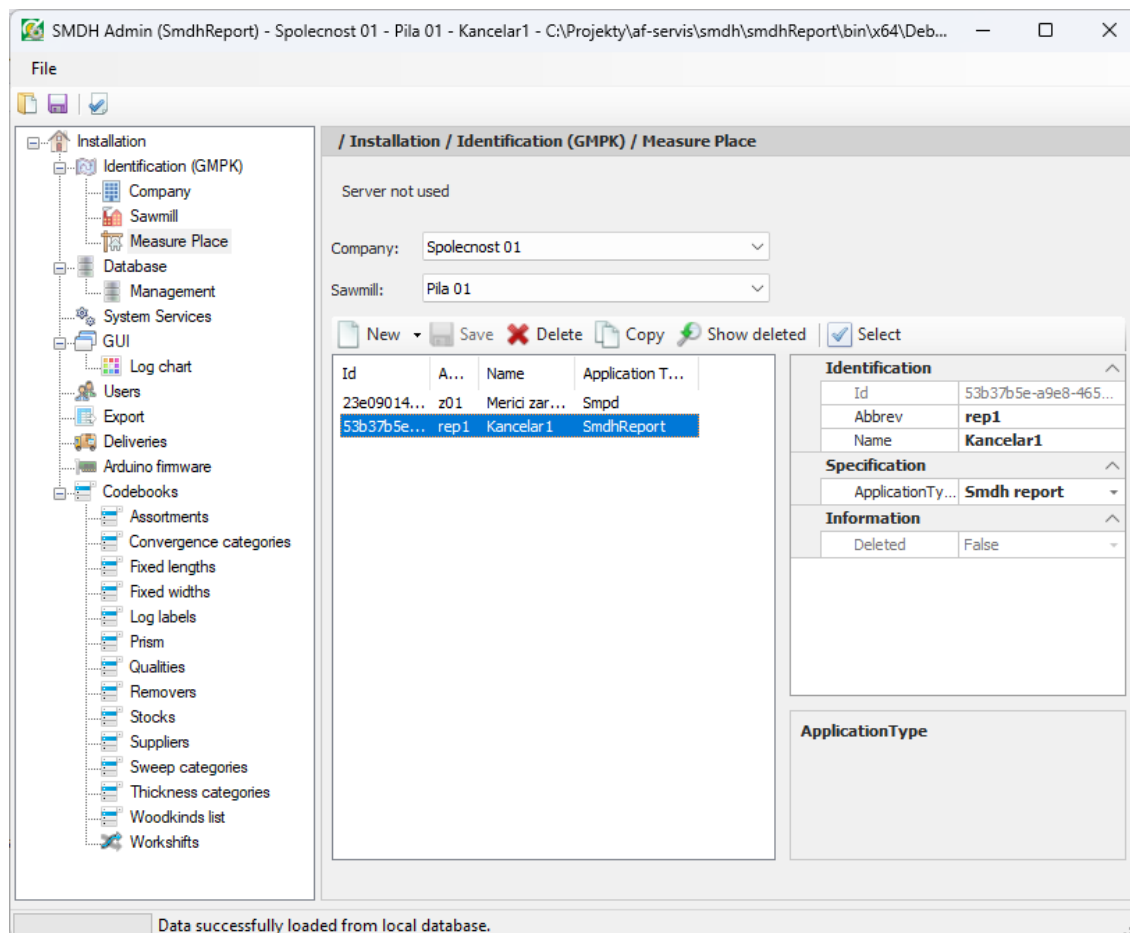
68 references
8 public class GlobalMeasurePlace
9 {
10     [Category("Saw Company")]
11     [TypeConverter(typeof(ExpandableObjectConverter))]
12     public Company Company { get; set; }
13
14     [Category("Sawmill")]
15     [TypeConverter(typeof(ExpandableObjectConverter))]
16     public Sawmill Sawmill { get; set; }
17
18     [Category("Measuring Place")]
19     [TypeConverter(typeof(ExpandableObjectConverter))]
20     public MeasurePlace MeasurePlace { get; set; }
21
22     [Browsable(false)]
23     public GlobalMeasurePlaceKey Gmpk => new GlobalMeasurePlaceKey(Company?.Id, Sawmill?.Id, MeasurePlace?.Id);
24
25     [Browsable(false)]
26     public bool IsDeleted => Company.Deleted ||
27         Sawmill.Deleted ||
28         MeasurePlace.Deleted;
29
30     10 references
31     public GlobalMeasurePlace()
32     {
33     }

```

Obrázek 3.7 - Ukázka části kódu GMP
(Zdroj: Vlastní zpracování)

Konfigurace GMP se provádí v podpůrné aplikaci SMDH Admin, kde je možné spravovat jednotlivé instalace a také je konfigurovat. Zde definované nastavení GMP má bezprostřední vliv na to, jaké informace a data bude aplikace SMDH Report zobrazovat. V praxi to může znamenat, že určitá data, jako jsou kvality dřevin nebo například skládky, mohou být sdílená mezi různými měřicími místy v rámci jedné pily, pokud GMP

odpovídá definované pile ve společnosti. Zároveň se v SMDH Adminu konfiguruje lokální databáze, případně využívání serveru, uživatelé, kteří se mohou přihlásit na konkrétní stanice, stejně jako všechny číselníky – kvalit, dřevin, dodavatelů, skládek a dalších. Aplikace SMDH Admin je vidět na následujícím obrázku níže.



Obrázek 3.8 - Podpůrná aplikace SMDH Admin
(Zdroj: Vlastní zpracování)

Vpravo od selektoru GlobalMeasurePlace (GMP) v ReportFormu se uživatelům nabízí možnost výběru tiskové sestavy. Tento prvek je klíčový pro generování reportů podle specifikace uživatele a je přímo propojen s funkcionalitou exportu a reportování, což je více přiblíženo v kapitole 3.3.4, která se věnuje právě exportu a analýze dat.

Následující element je *DeliverySearchControl*, který má na starosti filtrování a vyhledávání dat dodávek. To umožňuje uživatelům efektivně vyhledávat a třídit informace podle konkrétních parametrů hned na základní obrazovce. Kliknutím na jednotlivé popisky (*labely*), lze aktivovat filtr pro daný atribut, což je uživateli

zvýrazněno změnou barvy pozadí popisku na zelenou – stejně jako je to vidět u filtrů "od" a "do". Uživatelé mají možnost nastavit rozsah datumů a časů pro vyhledávání, určit, zda dodávky obsahují určité evidenční číslo nebo číslo dodacího listu, vybrat dodávky od konkrétního dodavatele, anebo vyhledávat na základě specifické poznámky nebo stavu dodávky – například pouze ty, které jsou již ukončené či uzavřené. Po zvolení filtrů a stisku tlačítka *Hledat* se spustí proces prohledávání databáze pomocí SQL dotazu sestaveného z aktivních filtrů, což uživatelům poskytne přesně specifikované výsledky.

```

30 references
public class DeliverySearchSettings
{
    6 references
    public DeliverySearchSettings() { }

    0 references
    public DeliverySearchSettings(bool fromChecked, DateTime from,
        bool toChecked, DateTime to, bool deliveryNumberChecked,
        string deliveryNumber, bool deliveryEvidenceNumberChecked, string deliveryEvidenceNumber,
        bool supplierChecked, string supplierName, ProcessDirection direction)
    {
        this.FromChecked = fromChecked;
        this.From = from;
        this.ToChecked = toChecked;
        this.To = to;
        this.DeliveryNumberChecked = deliveryNumberChecked;
        this.DeliveryNumber = deliveryNumber;
        this.DeliveryEvidenceNumberChecked = deliveryEvidenceNumberChecked;
        this.DeliveryEvidenceNumber = deliveryEvidenceNumber;
        this.SupplierChecked = supplierChecked;
        this.SupplierName = supplierName;
        this.ProcessDirection = direction;
    }
}
fields
}

```

Obrázek 3.9 - Ukázka kódu pro nastavování vyhledávání dodávek
(Zdroj: Vlastní zpracování)

```

cmd.CommandText = @"SELECT d.*
    FROM Delivery d
    LEFT JOIN Supplier s
    On d.SupplierId = s.Id
    LEFT JOIN (
        SELECT DeliveryId, MIN(Time) AS FirstProcessingTime, MAX(Time) AS LastProcessingTime
        FROM Log
        GROUP BY DeliveryId
    ) AS l ON d.Id = l.DeliveryId {collation} " +
    $"{whereSearchText} " +
    $"{gmpkWhere} " +
    $"{deliveryStateWhere} " +
    $"{deliveryNoteFlagWhere} " +
    $"ORDER BY s.Name, d.DeliveryNumber";

return DoSelectQuery(cmd);

```

Obrázek 3.10 - Část kódu vytvářející SQL dotaz pro vyhledávání
(Zdroj: Vlastní zpracování)

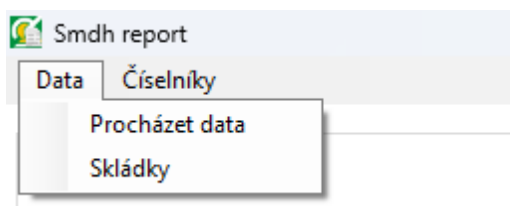
V levé spodní části ReportFormu je uživatelům nabízena funkce uložení a správy oblíbených vyhledávání. Tato část umožňuje uložení specifických filtrů společně s tiskovými sestavami, což následně usnadňuje a urychluje opakovaný přístup k často používaným vyhledáváním. Například účetní, která potřebuje pravidelně kontrolovat dodávky za poslední měsíc, může mít tento filtr přednastavený a rychle přístupný, což výrazně zjednodušuje její pravidelné pracovní postupy.

Na pravé straně formuláře je pak uživatelům prezentován detailnější pohled na informace související s aktuálně vybranou dodávkou. Zde se zobrazují konkrétní údaje, jako jsou například celkový počet kusů, objem dodávky, identifikační číslo a data spojená s časovými údaji o dodávce. Tento detailní přehled umožňuje uživatelům lepší orientaci v datech a efektivnější zpracování informací o konkrétních dodávkách.

3.3.2 Procházení a editace dat

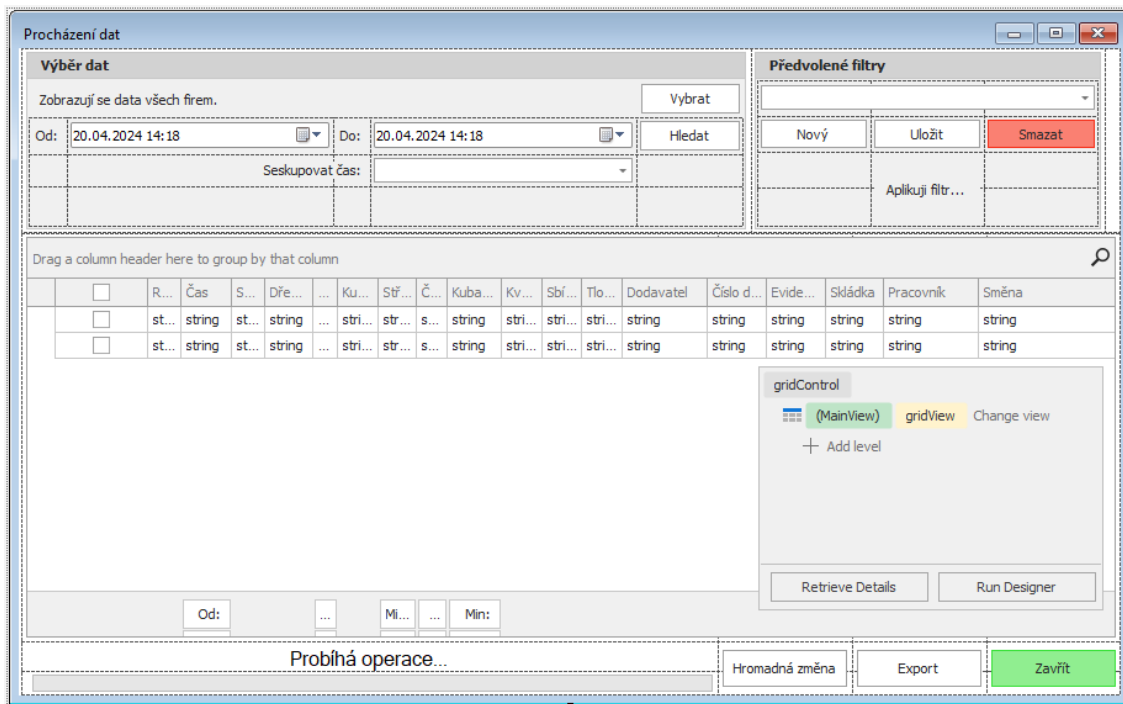
Tato podkapitola se zabývá funkcemi pro procházení a editaci dat, které jsou v rámci aplikace SMDH Report implementovány. Podokno procházení dat je možné spustit, po zvolení záložky "Data" a následně "Procházení dat" z *MenuStripu*. Mimo procházení dat je zde možné také spustit podokno pro správu skládek. Tuto funkcionalitu blíže popisuje kapitola 3.3.3.

Jak již bylo zmíněno, funkce pro procházení dat je přístupná prostřednictvím *MenuStripu*, které po aktivaci otevírá nové podokno aplikace. Toto podokno poskytuje uživatelům rozhraní pro pohodlné procházení dostupných datových záznamů. Cílem je umožnit uživatelům rychle a efektivně nalézt specifické informace v rámci rozsáhlé databáze.



Obrázek 3.11 - MenuStrip v aplikaci SMDH Report
(Zdroj: Vlastní zpracování)

Funkce procházení dat je klíčová pro efektivitu práce s aplikací, neboť zabezpečuje, že uživatelé mohou snadno navigovat databázi a rychle přistupovat k potřebným datům bez nutnosti procházet jednotlivé záznamy ručně. Zároveň je zde možnost si zobrazit a upravit údaje o naměřených kmenech, což je další vyžadovaná funkce ze strany klientů.



Obrázek 3.12 - Schéma Formu pro procházení a editaci dat v editoru
(Zdroj: Vlastní zpracování)

DataBrowseForm je formulář navržený pro efektivní procházení, úpravu a manipulaci s daty v rámci SMDH Reportu. UI pro tento form je navrženo a zamýšleno pro práci s klávesnicí a myší. Vzhled je zachován stejný jako u zbytku aplikací SMDH, ale s optimalizacemi pro myš a klávesnici.

První oblast v horní části formuláře je věnována komponentě "Výběr dat", která je součástí GroupBoxu. GroupBox je užitečná komponenta v oblasti vývoje UI, protože poskytuje vizuální seskupení souvisejících prvků ovládání, zvyšuje přehlednost formu a umožňuje uživatelům lepší orientaci ve funkcích, které spolu souvisí. V tomto případě jsou do GroupBoxu sdruženy funkce pro výběr měřicích míst a filtraci datových záznamů podle časového rozmezí a jednotek seskupení.

Procházení dat

Výběr dat
 Společnost 01, Pila 01, Všechna místa
 Od: 20.02.2022 05:30 Do: 18.04.2022 17:45
 Seskupovat čas: Po dni

Předvolené filtry
 Bez filtru
 Nový Uložit Smazat

Sem přetáhnete záhlaví sloupce, podle kterého chcete seskupovat

<input type="checkbox"/>	Rozříd	Čas	Směr	Dřevina	Délka [m]	Kub. dél.	Sřadový p.	Čep be.	Kubatura [m ³]	Kvalita	Dodavatel	Číslo dodávkového l...	Pracovník
<input type="checkbox"/>	00.00:05:54	21.02.2022 10:44:08	Vstup do procesu	Sm - Smrk bez k...	4,07	4,00	51,0	47,5	0,817	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:04:48	21.02.2022 10:38:14	Vstup do procesu	Sm - Smrk bez k...	4,01	4,00	53,0	51,5	0,882	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:03:37	21.02.2022 10:33:26	Vstup do procesu	Sm - Smrk bez k...	4,05	4,00	43,0	42,0	0,591	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:15:58	21.02.2022 10:29:49	Vstup do procesu	Sm - Smrk bez k...	4,08	4,00	58,0	58,5	1,057	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:12:26	21.02.2022 10:13:51	Vstup do procesu	Sm - Smrk bez k...	4,17	4,10	69,0	66,5	1,533	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:09:38	21.02.2022 10:01:25	Vstup do procesu	Sm - Smrk bez k...	4,42	4,40	62,0	60,5	1,328	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:09:01	21.02.2022 09:51:47	Vstup do procesu	Sm - Smrk bez k...	4,20	4,20	64,0	63,5	1,351	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:08:03	21.02.2022 09:42:46	Vstup do procesu	Sm - Smrk bez k...	4,15	4,10	60,0	57,5	1,199	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:02:42	21.02.2022 09:34:43	Vstup do procesu	Sm - Smrk bez k...	4,10	4,10	60,0	57,5	1,199	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:10:39	21.02.2022 09:32:01	Vstup do procesu	Sm - Smrk bez k...	4,10	4,10	59,0	68,0	1,121	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:03:30	21.02.2022 09:21:22	Vstup do procesu	Sm - Smrk bez k...	4,08	4,00	29,0	27,0	0,264	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:02:55	21.02.2022 09:17:52	Vstup do procesu	Sm - Smrk bez k...	4,02	4,00	39,0	40,5	0,478	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:02:57	21.02.2022 09:14:57	Vstup do procesu	Sm - Smrk bez k...	4,02	4,00	44,0	43,5	0,608	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:05:37	21.02.2022 09:12:00	Vstup do procesu	Sm - Smrk bez k...	4,05	4,00	49,0	47,5	0,754	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:02:06	21.02.2022 09:06:23	Vstup do procesu	Sm - Smrk bez k...	3,99	3,90	26,0	25,0	0,207	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:20:15	21.02.2022 09:04:17	Vstup do procesu	Sm - Smrk bez k...	4,03	4,00	30,0	29,0	0,283	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:02:24	21.02.2022 08:44:02	Vstup do procesu	Sm - Smrk bez k...	3,99	3,90	27,0	23,5	0,223	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:02:13	21.02.2022 08:41:38	Vstup do procesu	Sm - Smrk bez k...	4,04	4,00	31,0	25,5	0,302	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:02:05	21.02.2022 08:39:25	Vstup do procesu	Sm - Smrk bez k...	4,19	4,10	24,0	23,5	0,185	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:02:36	21.02.2022 08:37:20	Vstup do procesu	Sm - Smrk bez k...	4,03	4,00	41,0	40,5	0,528	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:02:57	21.02.2022 08:34:44	Vstup do procesu	Sm - Smrk bez k...	4,03	4,00	46,0	47,0	0,665	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:02:25	21.02.2022 08:31:47	Vstup do procesu	Sm - Smrk bez k...	4,00	4,00	26,0	24,5	0,212	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:03:29	21.02.2022 08:29:22	Vstup do procesu	Sm - Smrk bez k...	4,07	4,00	41,0	41,0	0,528	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:02:28	21.02.2022 08:25:53	Vstup do procesu	Sm - Smrk bez k...	4,05	4,00	38,0	35,0	0,454	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:05:31	21.02.2022 08:23:25	Vstup do procesu	Sm - Smrk bez k...	4,04	4,00	46,0	45,0	0,665	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:04:14	21.02.2022 08:17:54	Vstup do procesu	Sm - Smrk bez k...	4,05	4,00	32,0	27,5	0,322	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan
<input type="checkbox"/>	00.00:11:16	21.02.2022 08:13:40	Vstup do procesu	Sm - Smrk bez k...	4,13	4,10	71,0	70,5	1,623	D - Kvalita D	dob01 - Drevododa...	4181744	Operator Milan

Od: 21.02.2022 6:47:14
 Do: 18.04.2022 17:13:13...
 Ksůů: 18873

Min: 0,51
 Max: 38,80
 Průměr: 3,91

Min: 16,0
 Max: 90,0
 Průměr: 3...

Min: 1...
 Max: 9...
 Průměr: 0...

Min: 0,014
 Max: 5,376
 Průměr: 0...
 Suma: 585...

Hromadná změna Export Zavřít

Obrázek 3.13 - Okno DataBrowseForm s vyhledanými daty
 (Zdroj: Vlastní zpracování)

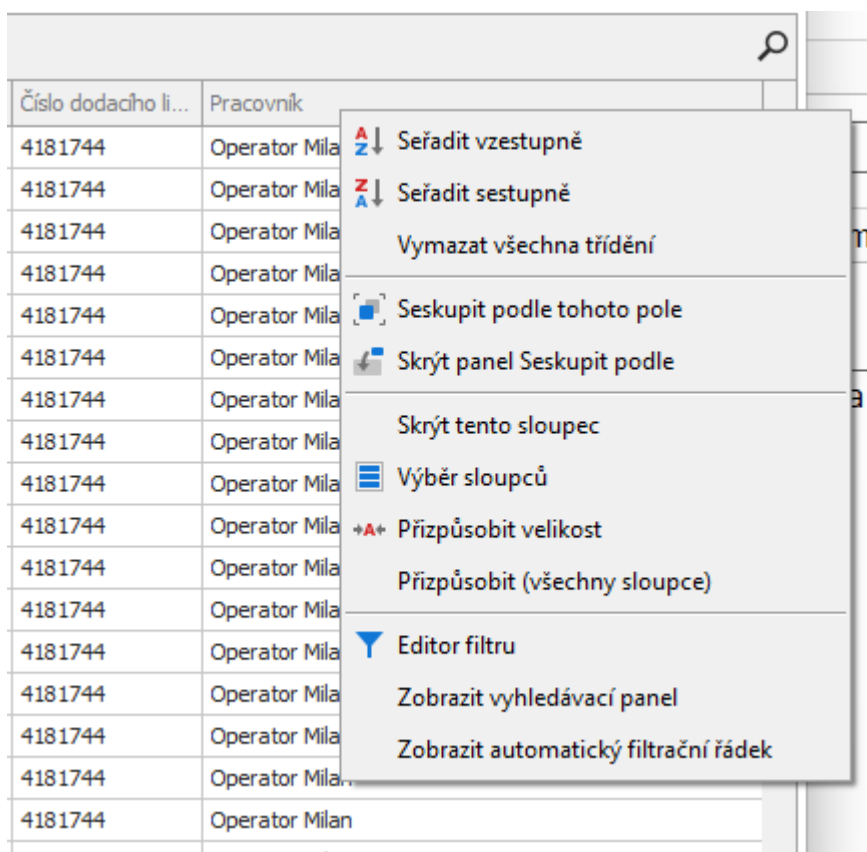
Selektor GMP, také upravený pro optimalizovanou práci s klávesnicí a myší, slouží k výběru konkrétní stanice, pily, či společnosti, jejichž data chce uživatel zobrazit (stejně jako na hlavním *ReportFormu*). Tato komponenta tedy umožňuje rychlé a přesné vyhledávání v databázi měřicích míst. Pod selektorem GMP se nachází ovládací prvky pro nastavení požadovaného datumu a času, v nichž mají být záznamy zobrazeny. Uživatelé mají možnost seskupovat zobrazená data podle různých časových jednotek – dne, měsíce či roku, což usnadňuje analýzu dat v dlouhodobém kontextu.

Druhou komponentou napravo od výběru dat je předvolba a aplikace filtrů dat. Uživatelé zde mohou využít uložení specifických filtrů, které reflektují časté potřeby nebo preferované pohledy na data. Tyto přednastavené filtry lze následně rychle aplikovat na data bez nutnosti opakovaného nastavování kritérií (obdobně jako filtry v *ReportFormu*). Tato logika opět uživatelům umožňuje zrychlit práci s daty a zjednodušuje používání aplikace.

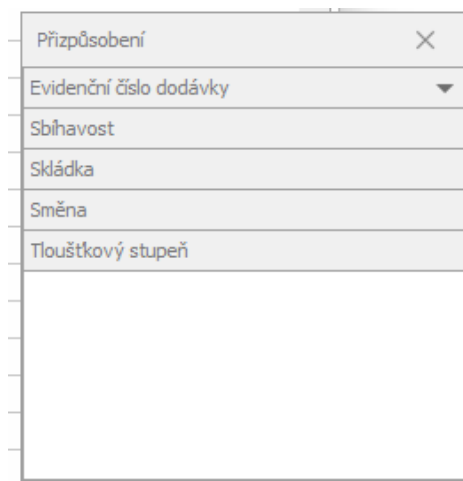
Střední, největší část *DataBrowseFormu* představuje uživatelům hlavní tabulku, která slouží jako zobrazovací rozhraní pro data získaná na základě specifikací z části výběru dat. Tato sekce je zásadní pro interakci uživatelů s daty a jejich analytické zpracování.

V tabulce je možné vidět záznamy o kmenech, kde jsou prezentovány různé atributy z databáze a pospojovaných tabulek. Sloupce ukazují naměřené hodnoty jako jsou rozměry kmene, typ dřeviny, kvalita a mnoho dalších parametrů, což poskytuje komplexní pohled na jednotlivá měření. Unikátní je sloupec zobrazující časový rozdíl mezi měřeními, což umožňuje identifikovat potenciální prostoje v procesu a analyzovat jejich příčiny.

Uživatelé mohou v této tabulce provádět širokou škálu úprav – sloupce lze přesouvat pomocí drag and drop techniky, seřadit data kliknutím na záhlaví sloupce s automatickým zobrazením šipky indikující směr řazení, a upravovat šířku sloupců podobně pomocí myši, obdobně jako v aplikaci Excel. Tato úroveň customizace usnadňuje orientaci v datech a zvyšuje efektivitu práce s celou aplikací.



Obrázek 3.14 - Možnosti úprav filtrování dat v tabulce
(Zdroj: Vlastní zpracování)



Obrázek 3.15 - Možnost skrýt sloupce v tabulce – funkce výběr sloupců
(Zdroj: Vlastní zpracování)

Aplikace inteligentně využívá výčetové typy (*enums*), což jsou datové struktury umožňující definovat soubor pojmenovaných konstantních hodnot. Například, u atributů jako typ dřeviny nebo kvalita, enum zajišťuje, že při editaci hodnot mohou uživatelé vybírat pouze z předdefinovaných platných možností, což eliminuje riziko chyb při ručním zadávání. Aby se předešlo nesrovnalostem, je při vytváření nové kvality dřeva uživatel nucen použít číselník kvalit, kde ji musí nejdříve explicitně definovat, což zajišťuje konzistenci dat.

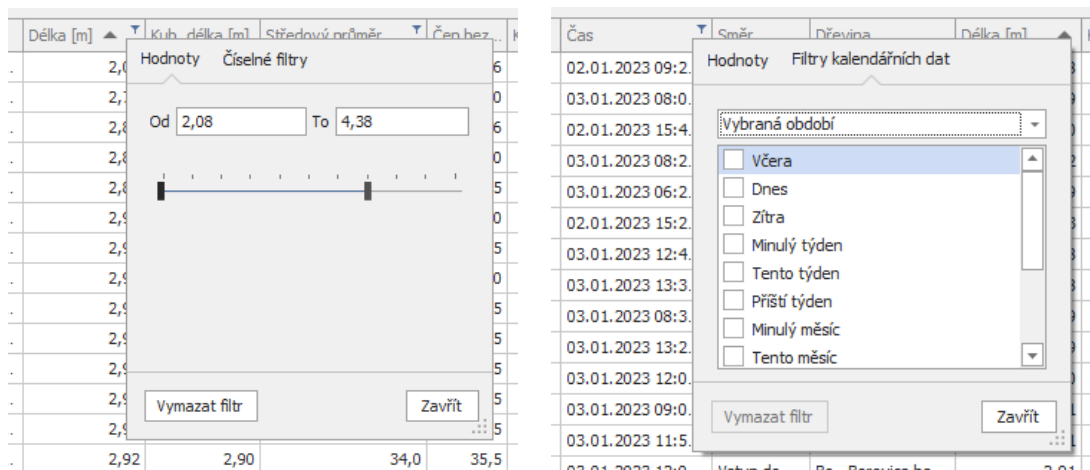
Sem přetáhněte záhlaví sloupce, podle kterého chcete seskupovat

	<input type="checkbox"/>	Rozdíl	Čas	Směr	Dřevina	Délka	Kub. délka [m]	Sřředový pr...	Čep bez...	Kubaturn [m3]	Kvalita	Dodavatel	Číslo dodacho listu
	<input type="checkbox"/>	00.00:01:...	03.01.2022 06:29:13	Vstup do procesu	Bo - Borovice	4,04	4,00	26,0	23,7	0,198	D - Kvalita D	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:02:...	03.01.2022 06:31:18	Vstup do procesu	Bo - Borovice	4,03	4,00	25,0	22,2	0,183	D - Kvalita D	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:01:...	03.01.2022 06:33:05	Vstup do procesu	Bo - Borovice	4,04	4,00	33,0	27,5	0,317	D - Kvalita D	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:01:...	03.01.2022 06:34:20	Vstup do procesu	Bo - Borovice	4,03	4,00	24,0	22,2	0,169	D - Kvalita D	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:02:...	03.01.2022 06:36:23	Vstup do procesu	Bo - Borovice	4,07	4,00	35,0	33,2	0,356	D - Kvalita D	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:02:...	03.01.2022 06:38:29	Vstup do procesu	Bo - Borovice	4,03	4,00	43,0	41,6	0,533	D - Kvalita D	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:10:...	03.01.2022 06:49:08	Vstup do procesu	Bk - Buk	5,06	5,00	45,0	42,2	0,748	BC - Kvalita B ne...	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:03:...	03.01.2022 06:52:35	Vstup do procesu	Bk - Buk	5,02	5,00	37,0	33,3	0,503	BC - Kvalita B ne...	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:02:...	03.01.2022 06:54:48	Vstup do procesu	Bk - Buk	5,04	5,00	31,0	32,8	0,351	BC - Kvalita B ne...	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:06:...	03.01.2022 07:01:08	Vstup do procesu	Sm - Smrk	5,00	5,00	47,0	43,6	0,817	BC - Kvalita B ne...	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:03:...	03.01.2022 07:04:49	Vstup do procesu	Sm - Smrk bez kůry	5,00	5,00	44,0	42,7	0,714	BC - Kvalita B ne...	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:03:...	03.01.2022 07:07:49	Vstup do procesu	Bo - Borovice	4,90	4,90	38,0	28,9	0,520	BC - Kvalita B ne...	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:02:...	03.01.2022 07:10:44	Vstup do procesu	Bo - Borovice bez kůry	5,00	5,00	34,0	27,5	0,423	BC - Kvalita B ne...	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:05:...	03.01.2022 07:16:17	Vstup do procesu	Md - Modřín	5,00	5,00	51,0	43,6	0,963	BC - Kvalita B ne...	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:12:...	03.01.2022 07:28:20	Vstup do procesu	Md - Modřín bez kůry	5,00	5,00	66,0	59,4	1,623	BC - Kvalita B ne...	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:05:...	03.01.2022 07:34:10	Vstup do procesu	Bk - Buk	5,18	5,10	45,0	34,3	0,763	BC - Kvalita B ne...	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:16:...	03.01.2022 07:50:49	Vstup do procesu	Bk - Buk	4,98	4,90	39,0	27,5	0,548	BC - Kvalita B ne...	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:02:...	03.01.2022 07:53:39	Vstup do procesu	Bk - Buk	5,00	5,00	44,0	33,3	0,714	BC - Kvalita B ne...	do01 - Drevododavatel CZ	4181661
	<input type="checkbox"/>	00.00:05:...	03.01.2022 07:59:21	Vstup do procesu	Bk - Buk	5,10	5,10	45,0	43,6	0,763	BC - Kvalita B ne...	do01 - Drevododavatel CZ	4181661

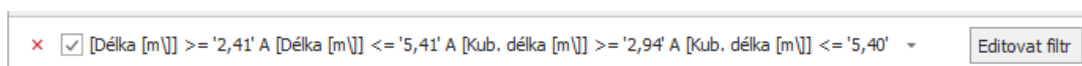
Obrázek 3.16 - Práce s výčtem typů z pohledu UI
(Zdroj: Vlastní zpracování)

Tabulka také obsahuje systém agregace hodnot u vybraných sloupců, jak je vidět na Obrázek 3.13. Například, pokud uživatel zvolí časové rozmezí jednoho týdne a všechna měření proběhla pouze ve dvou dnech, dole pod tabulkou jsou zobrazeny agregované informace jako první a poslední den měření, což poskytuje okamžitý přehled o aktivitách během zvoleného období. Dále jsou zde zobrazeny minimální a maximální hodnoty, průměry a suma pro relevantní sloupce jako délka a kubatura, což umožňuje rychlou kontrolu dat a identifikaci potenciálních chyb, jako jsou nereálné maximální hodnoty, které lze pak snadno upravit nebo odstranit.

Aplikace poskytuje uživatelům pokročilé možnosti filtrování dat přímo v tabulce zobrazení. Po najetí kurzorem na záhlaví jakéhokoliv sloupce se v jeho pravém horním rohu zobrazí ikona filtru, která indikuje možnost aktivovat filtrační funkce. Po kliknutí na tuto ikonu se uživatelům otevře menu s možnostmi filtrování, které jsou vždy specifické pro daný sloupec. Uživatelé mohou aplikovat různé kombinace filtrů podle potřeb, což jim umožňuje lépe analyzovat data z tabulky. Nastavený filtr se pak zobrazuje na spodní části obrazovky, což poskytuje přehled o aktivních filtračních kritériích. Pro efektivnější práci s aplikací lze vybrané filtry uložit prostřednictvím komponenty pro předvolené filtry umístěné v pravém horním rohu obrazovky (jak bylo popisováno výše).



Obrázek 3.17 - Náhled několika možností filtrování dat
(Zdroj: Vlastní zpracování)



Obrázek 3.18 - Zobrazení aktivního filtru v tabulce
(Zdroj: Vlastní zpracování)

Detailní zobrazení kmene v aplikaci umožňuje uživatelům nahlédnout do kompletního seznamu atributů, které jsou o vybraném kmeni uloženy v databázi, a to prostřednictvím dialogového okna, které se aktivuje dvojitým kliknutím na záznam v tabulce. Okno prezentuje ID kmene, časové údaje, informace o společnosti a místě měření (GMP), a také údaje o pracovnících, kteří se kmenem pracovali, typu kmene a mnoho dalších. Tato funkce poskytuje rychlý a strukturovaný přehled o všech informacích kmene, v případě, že by uživatel potřeboval vyhledat specifickou informaci.

Detail kmene	
Id	20220221104408686s01p01z013527
> Gmpk	C: [40dcb2ed-c94c-4af1-b3c1-c4f582a0db4...
DirectionId	Vstup do procesu
DirectionTxt	Vstup do procesu
Time	21.02.2022 10:44
TimeDiff	00:05:54
TimeZone	1
ProcessingCompanyId	40dcb2ed-c94c-4af1-b3c1-c4f582a0...
ProcessingCompanyAbbrev	s01
ProcessingCompanyTxt	Společnost 01
ProcessingCompanyIdentificationNumber	
ProcessingCompanyVatNumber	
SawmillId	cdb68ca-0846-463f-bb89-0095410...
SawmillAbbrev	p01
SawmillTxt	Pila 01
MeasurePlaceId	23e09014-189f-4693-844d-227f42fe...
MeasurePlaceAbbrev	z01
MeasurePlaceTxt	Merici zarizeni 01
WorkerId	a23f460c-a1f0-4f9e-aaa9-ded94ef7...
WorkerAbbrev	MO
WorkerTxt	Operator Milan
TypeId	Surový kmen
TypeTxt	Surový kmen
LinkedWithId	
WoodKind_Id	2
WoodKind_ProcessingCompanyId	40dcb2ed-c94c-4af1-b3c1-c4f582a0...
WoodKind_SawmillId	cdb68ca-0846-463f-bb89-0095410...
WoodKind_MeasurePlaceId	_ALL_

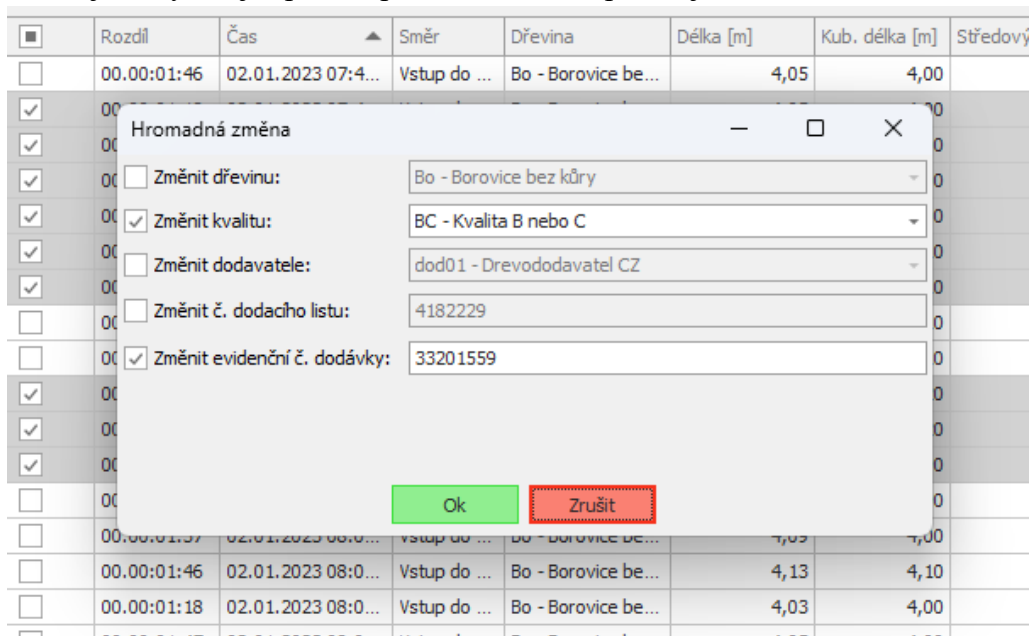
Id
Id kmene

Zavřít

Obrázek 3.19 - Detail naměřeného kmene po výběru z tabulky
(Zdroj: Vlastní zpracování)

Funkce hromadné změny v SMDH Reportu představuje zjednodušení pracovních procesů tím, že umožňuje uživatelům efektivně upravovat vlastnosti několika kmenů najednou. Po selekci požadovaných kmenů za pomoci checkboxů je možné provést jejich hromadnou editaci kliknutím na tlačítko "Hromadná změna" v pravém dolním rohu okna. Následně se zobrazí dialogové okno, kde lze najednou změnit typ dřeviny, kvalitu, dodavatele, číslo dodacího listu, nebo evidenční číslo dodávky. Tato funkce byla vyvinuta

na základě požadavků zákazníků, kteří často čelili potřebě opravovat chybně zadané informace spojené s dodávkou nebo kvalitou dřeva, a její implementace významně zjednodušuje a zrychluje správné přiřazení dat k odpovídajícím dodávkám.



Obrázek 3.20 - Možnosti hromadné změny v aplikaci
(Zdroj: Vlastní zpracování)

V procházení dat je uživatelům poskytnuta i možnost exportu dat z tabulky. Vedle tlačítka "Hromadné změny" se nachází tlačítko "Export", které slouží k vytvoření externí kopie aktuálně zobrazených dat. Jestliže uživatel nevybral žádný kmen v tabulce a pokusí se data exportovat, systém ho upozorní, že je třeba nejprve vybrat konkrétní kmeny. Při výběru jednoho či více kmenů umožňuje tlačítko "Export" generování tabulky, která bude obsahovat data zvolená uživatelem, včetně aplikovaných filtrů a konfigurace sloupců. Takto vyexportovaná tabulka tedy přesně reflektuje tabulku, jak ji uživatel nastavil a včetně vybraných dat a agregovaných informací.

Čas	Dřevina	Délka [m]	Kubatura [m3]	Kvalita	Číslo dodacího listu
23.04.2022 00:32:13	Bo - Borovicebez kůry	3,95	0,223	D - Kvalita D	4181845
23.04.2022 00:34:47	Bo - Borovicebez kůry	4,41	0,354	D - Kvalita D	4181845
23.04.2022 00:37:44	Bo - Borovicebez kůry	3,99	0,515	D - Kvalita D	4181845
23.04.2022 00:48:52	Bo - Borovicebez kůry	3,92	0,176	D - Kvalita D	4181845
23.04.2022 00:51:00	Bo - Borovicebez kůry	3,94	0,223	D - Kvalita D	4181845
23.04.2022 00:55:23	Bo - Borovicebez kůry	4,03	0,454	D - Kvalita D	4181845
23.04.2022 01:00:12	Bo - Borovicebez kůry	3,97	0,176	D - Kvalita D	4181845
23.04.2022 01:02:31	Bo - Borovicebez kůry	3,99	0,294	D - Kvalita D	4181845
23.04.2022 01:06:21	Bo - Borovicebez kůry	3,91	0,620	D - Kvalita D	4181845
23.04.2022 01:10:02	Bo - Borovicebez kůry	4,19	0,394	D - Kvalita D	4181845
23.04.2022 01:12:57	Bo - Borovicebez kůry	4,00	0,246	D - Kvalita D	4181845
23.04.2022 01:14:28	Bo - Borovicebez kůry	3,97	0,176	D - Kvalita D	4181845
23.04.2022 01:17:30	Bo - Borovicebez kůry	3,98	0,240	D - Kvalita D	4181845
23.04.2022 01:19:08	Bo - Borovicebez kůry	4,00	0,181	D - Kvalita D	4181845
Od: 23.04.2022 0:29:42		Min: 3,89	Min: 0,123		
Do: 23.04.2022 13:35:29		Max: 4,41	Max: 0,849		
Kusů: 280		Průměr: 4,01	Průměr: 0,285		
			Suma: 79,779		

Obrázek 3.21 - Vyexportovaná tabulka z procházení dat
(Zdroj: Vlastní zpracování)

V celém projektu SMDH je kladen důraz na plynulý a responzivní UI, což je důležité zvláště při manipulaci s velkými objemy dat. Aby se předešlo zamrznutí UI během vyhledávání, které by mohlo nastat v důsledku dlouhotrvajících operací prováděných v hlavním UI vlákně, je proces vyhledávání realizován v samostatných vláknech. Při vyhledávání může být UI vlákno zahlceno požadavky na zpracování, což by vedlo k jeho neodpovídání. Aby se tomuto zabránilo, je vyhledávání dat delegováno na *Background Workery*, což jsou specializované komponenty určené k provádění asynchronních operací na pozadí bez narušení uživatelské interakce. V jiných částech projektu jsou použity také *Tasky*, ale pro účely vyhledávání byla z důvodu konzistence a efektivity zvolena metoda *Background Workerů*. Tento přístup umožňuje aplikaci zpracovávat vyhledávací dotazy na pozadí, zatímco uživatelské rozhraní zůstává vždy reagující a dostupné pro další uživatelské akce.

```

1 reference
private void bwLoadData_RunWorkerCompleted(object sender, RunWorkerCompletedEventArgs e)
{
    setProgressBarVisible(false);
    btnSearch.ButtonType = DXButton.DXButtonType.Normal;
    btnSearch.Text = Resources.strSearch;
    btnSearch.Enabled = true;

    if (e.Cancelled)
        return;

    if (e.Result is Exception exception)
    {
        log.Error($"Failed to load data: {exception.Message}", exception);

        if (exception is OutOfMemoryException)
        {
            SimpleDialogForm.Dialog(Resources.strErrorTooMany,
                DialogTypeIcon.Error, MessageBoxButtons.OK, KeyboardControl, this);
        }
        else
        {
            SimpleDialogForm.Dialog(string.Format(Resources.strErrorLoadFail,
                exception.Message),
                DialogTypeIcon.Error, MessageBoxButtons.OK, KeyboardControl, this);
        }
        return;
    }

    logs = (List<WoodData>)e.Result;
    gridControl.DataSource = logs;
}

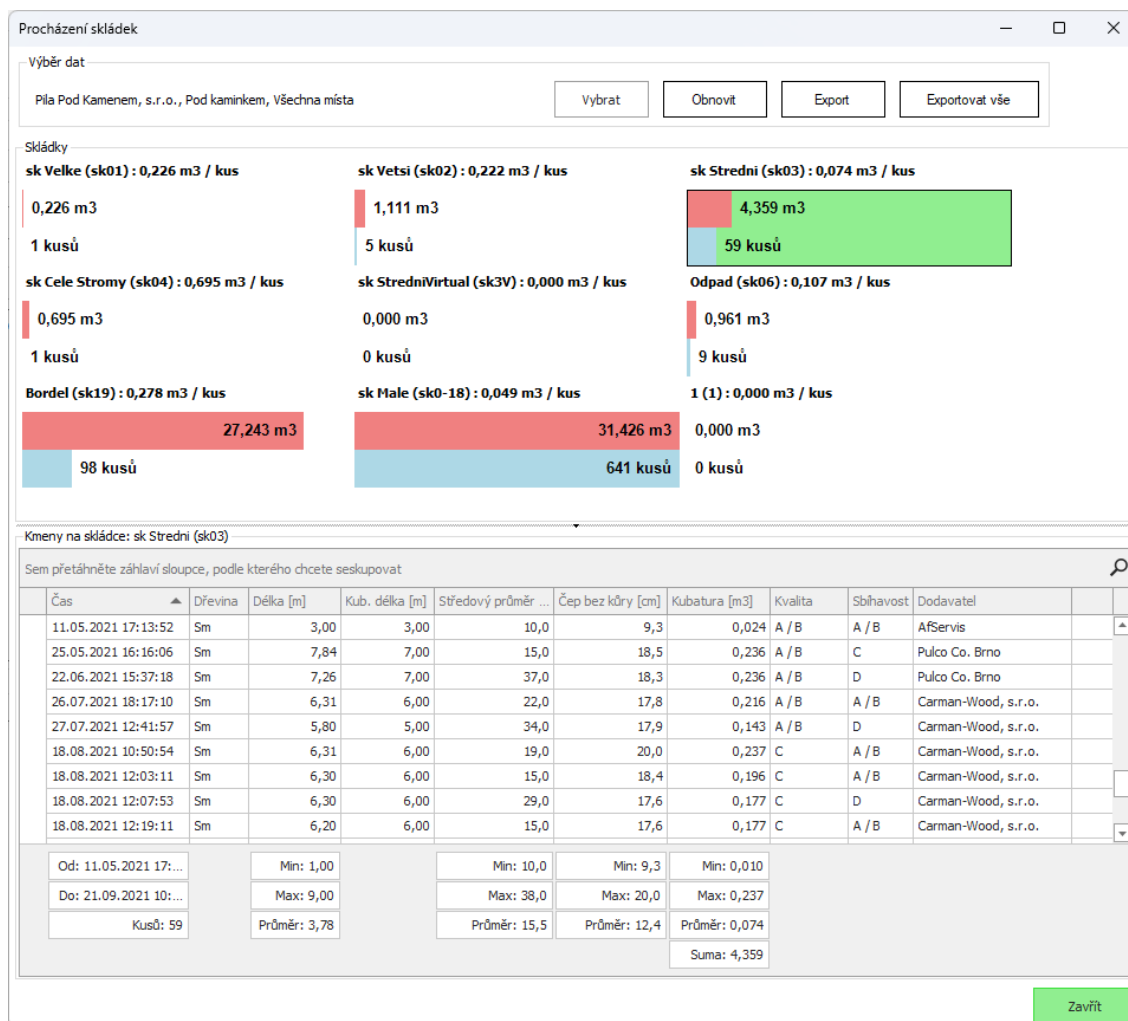
```

Obrázek 3.22 - Ukázka kódu při dokončení prohledávání *BackgroundWorkeru*
(Zdroj: Vlastní zpracování)

3.3.3 Přehled skládek

V rámci reportingového modulu SMDH Report, bylo implementováno podokno pro přehled skládek. Skládky v aplikaci představují skládky dřeva, které obsahují kmeny, nebo jejich pořezané části. Kmeny jsou zařazeny na skládku na základě definovaných kritérií dané skládky, což bývá nejčastěji maximální naměřená délka a maximální průměr daného kmene. To umožňuje efektivní třízení a následné využití kmenů. Uživatelé samozřejmě mají možnost definovat vlastní skládky a nastavit pravidla pro třízení naměřených kmenů, aby bylo zajištěno, že kmeny budou správně zařazeny na příslušné skládky.

Podokno přehledu skládek je přístupné přes *MenuStrip* aplikace volbou "Data" a následně "Skládky". Je navrženo pro intuitivní navigaci a efektivní manipulaci s daty, což zahrnuje možnosti prohlížení, aktualizace a analýzu informací spojených se skládkami.



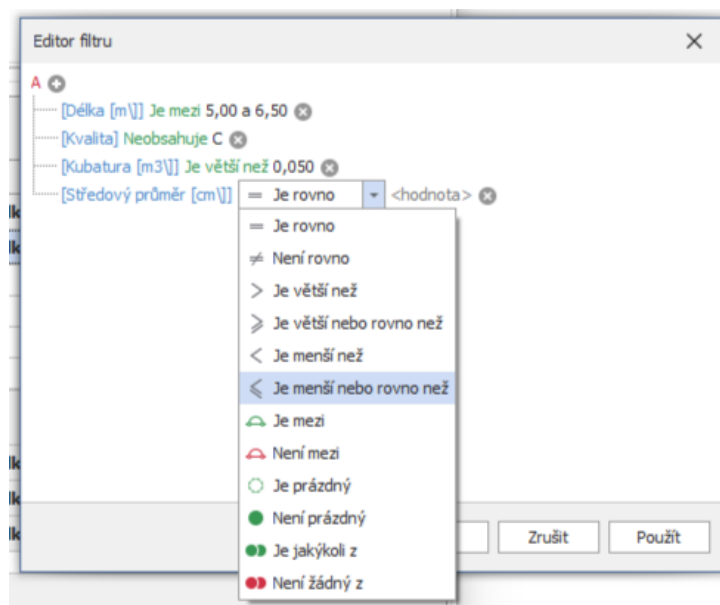
Obrázek 3.23 - Procházení skládek v aplikaci SMDH Report
(Zdroj: Vlastní zpracování)

Celý přehled skládek je v aplikaci koncipován tak, aby odrážel jednotný design a UX, které je typické pro celý projekt SMDH. V jeho horní části se nachází hlavní panel s funkcí výběru GMP, což umožňuje sledování skládek specifických pro různá měřicí místa nebo jejich sdílení mezi více místy či společnostmi. Tato funkcionality a její důležitost více rozpracovány v předešlých kapitolách práce. Vedle tlačítka pro výběr GMP je umístěno tlačítko pro obnovu dat, které slouží k aktualizaci informací a zajišťuje, že uživatelé pracují s nejnovějšími dostupnými daty. Tlačítka pro export dat jsou rovněž umístěna na panelu a jejich funkcionality je více přiblížena v kapitole 3.3.4.

Střední část okna je věnována přehledu skládek, kde jsou jednotlivé skládky graficky prezentovány s barevným kódováním, reprezentujícím jejich zaplněnost v porovnání s ostatními skládkami. Každá skládka je zde zobrazena s uvedením celkového objemu

dřeva v kubických metrech (m³) a počtem kusů na skládce, což poskytuje uživatelům okamžitý přehled o objemu materiálu.

Po selekci konkrétní skládky, což je indikováno barevným zvýrazněním, je uživatelům poskytnuta detailní tabulka kmenů patřících k vybrané skládce, přístupná přes šipku v dolní části okna. Pomocí této šipky je možné zobrazit tabulku, případně nastavit, jak velkou část obrazovky bude tabulka zabírat. Tato interaktivní tabulka, podobně jako tabulka procházení dat popsaná v kapitole 3.3.2, je navržena pro intuitivní ovládání a umožňuje uživatelům flexibilně přizpůsobit její vzhled a rozložení sloupců. Integrované funkce vyhledávání a filtraci poskytují rozšířené možnosti pro analýzu a organizaci dat, což přispívá k efektivnější správě skládek v rámci aplikace.



Obrázek 3.24 - Možnosti vytváření filtrů v tabulce
(Zdroj: Vlastní zpracování)

Jednou z funkcí, která ještě nebyla představena je schopnost seskupování dat v tabulce podle zvolených kritérií. Tato funkcionalita je velmi užitečná pro logické kategorizování dat, i uvnitř specifických skládek. Uživatelé mohou například kmeny seskupit podle kubírovací délky, která je běžně zaokrouhlována na celé metry nebo na desítky centimetrů. Díky tomu je možné rychle identifikovat a porovnat kmeny patřící do stejné kategorie, neboť reálná délka každého kmene je zřetelně uvedena vedle seskupených hodnot.

Toto rozdělení podle seskupení sloupců je aplikovatelné a užitečné také v ostatních částech aplikace, zejména v procházení dat, kde uživatelé mohou efektivně strukturovat a analyzovat informace dle individuálních požadavků.

Další z mnoha funkcí, které tabulka nabízí, je také nástroj pro vyhledávání, zasazený v pravém horním rohu ve formě symbolu lupy. Vyhledávací lišta, která se zde nachází, umožňuje uživatelům rychle nalézt specifické záznamy nebo skupiny dat podle klíčových slov nebo hodnot. Tato funkce značně usnadňuje práci s rozsáhlými sadami dat a zvyšuje efektivitu při správě a sledování informací o skládkách dřeva. Snadná dostupnost a intuitivní ovládání vyhledávacího nástroje přispívá k celkové uživatelské spokojenosti při práci s aplikací.

Kmeny na skládce: sk Střední (sk03)

Kub. délka [m]

Čas	Dřevina	Délka [m]	Středový průměr [c...	Čep bez kůry [cm]	Kubaturn [m3]	Kvalita	Sbiřavost	Dodavatel
> Kub. délka [m]: 4,00 Délka: 4,16 m, Střed: 19,3 cm, Čep bez kůry: 13,2 cm, Kubatura: 0,081 m3, Kubatura celkem: 0,486 m3, Kusů: 6								
▼ Kub. délka [m]: 5,00 Délka: 5,30 m, Střed: 21,5 cm, Čep bez kůry: 14,2 cm, Kubatura: 0,097 m3, Kubatura celkem: 0,390 m3, Kusů: 4								
11.05.2021 17:13:09	Sm	5,00	11,0	10,8	0,048	TEST	A / B	AfServis
11.05.2021 17:13:20	Sm	5,00	12,0	10,9	0,057	TEST	A / B	AfServis
27.07.2021 12:41:57	Sm	5,80	34,0	17,9	0,143	A / B	D	Carman-Wood, s.r.o.
20.09.2021 13:25:36	Sm	5,40	29,0	17,1	0,143	TEST	D	Carman-Wood, s.r.o.
Kusů: 4		Průměr: 5,30	Průměr: 21,5	Průměr: 14,2	Průměr: 0,097			
> Kub. délka [m]: 6,00 Délka: 6,25 m, Střed: 19,9 cm, Čep bez kůry: 16,6 cm, Kubatura: 0,172 m3, Kubatura celkem: 1,717 m3, Kusů: 10								
> Kub. délka [m]: 7,00 Délka: 7,33 m, Střed: 21,8 cm, Čep bez kůry: 16,2 cm, Kubatura: 0,188 m3, Kubatura celkem: 0,754 m3, Kusů: 4								
> Kub. délka [m]: 9,00 Délka: 9,00 m, Střed: 13,0 cm, Čep bez kůry: 10,5 cm, Kubatura: 0,119 m3, Kubatura celkem: 0,119 m3, Kusů: 1								
Od: 11.05.2021 17:12:...		Min: 1,00	Min: 10,0	Min: 9,3	Min: 0,010			
Do: 21.09.2021 10:19:...		Max: 9,00	Max: 38,0	Max: 20,0	Max: 0,237			
Kusů: 59		Průměr: 3,78	Průměr: 15,5	Průměr: 12,4	Průměr: 0,074			
					Suma: 4,359			

Zavřít

Obrázek 3.25 - Možnosti seskupení dat ve skládkách
(Zdroj: Vlastní zpracování)

```

1 reference
private List<WoodData> loadStockLogs(Stock stock)
{
    setProgressBarStyle(ProgressBarStyle.Marquee);

    List<WoodData> stockLogs = stock.GetLogs(Dal, true, true);

    setProgressBarStyle(ProgressBarStyle.Continuous);
    setProgressBarMaximum(stockLogs.Count);

    DateTime lastTime = DateTime.MinValue;
    ...

    int cnt = 0;
    foreach (WoodData w in stockLogs)
    {
        if (bwLoadStockLogs.CancellationPending)
            return null;

        TimeSpan diff = w.Time - lastTime;

        if (diff.TotalMinutes > 90)
            w.SetTimeDiff(null);
        else
            w.SetTimeDiff(diff);

        lastTime = w.Time;

        cnt++;

        if (cnt % 10 == 0)
            bwLoadStockLogs.ReportProgress(cnt);
    }

    bwLoadStockLogs.ReportProgress(stockLogs.Count);

    return stockLogs;
}

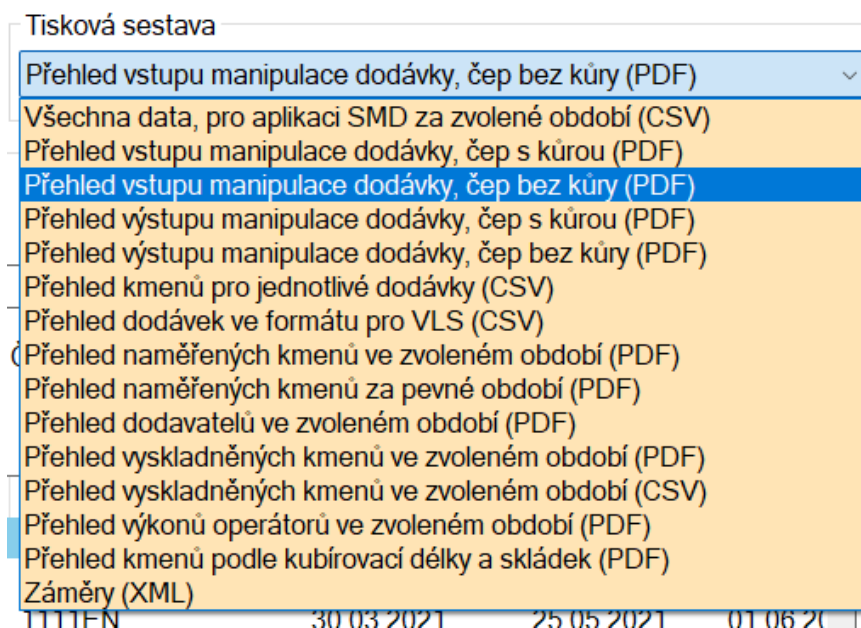
```

Obrázek 3.26 - Ukázka části kódu načítající data pro skládku
(Zdroj: Vlastní zpracování)

3.3.4 Export a reportování dat

Tato kapitola se věnuje klíčovému aspektu práce a aplikace – a to možnostem exportu dat do různých formátů a jejich následného zpracování. Cílem této části je ukázat, jak lze data přenést z aplikace do vnějšího prostředí, tak aby odpovídala potřebám uživatelů. Tím je splněn jeden z hlavních cílů této aplikace, který spočívá ve vytvoření efektivního a flexibilního nástroje pro reportování a sdílení naměřených dat.

Na hlavní obrazovce aplikace, jak již bylo zmíněno v předchozích kapitolách, se v pravém horním rohu nachází možnosti tiskových sestav a exportů (viz Obrázek 3.5). Tyto funkce umožňují uživatelům využít data aktuálně zobrazená v prostřední části okna a přizpůsobit je konkrétním potřebám. UI aplikace se dynamicky mění podle typu vybraných dat a interakcí uživatele. V případě, že je úkolem vyhledat dodávky, UI se přizpůsobí tak, aby reflektovalo relevantní informace, jako jsou dodavatelé, dodací listy, evidenční čísla a další, a zároveň poskytlo uživatelům flexibilní filtry pro specifické dotazy.

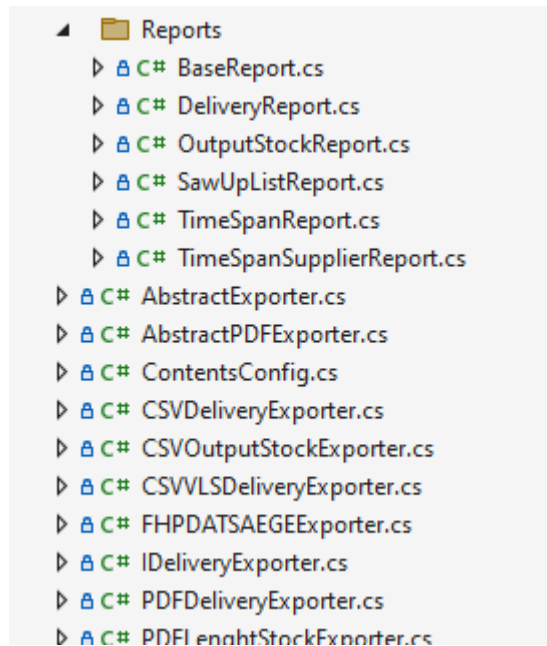


Obrázek 3.27 - Přehled tiskových sestav a možností exportu dat
(Zdroj: Vlastní zpracování)

Exportování dat je v SMDH Reportu navrženo s ohledem na modulárnost a rozšiřitelnost, což umožňuje s lehkostí přidávat nové typy reportů podle potřeb uživatelů nebo evoluce samotného projektu. Základem tohoto systému je využití principů objektově orientovaného programování, zejména dědičnosti. Díky tomu mohou nově vytvořené reporty efektivně dědit vlastnosti a metody od svých předchůdců a dále je rozšiřovat o novou funkcionalitu, která je potřebná pro specifické typy reportů a tiskových sestav.

Ve stávající struktuře kódu jsou reporty rozčleněny do pěti základních kategorií, odpovídajících datům nebo typům dat, na které jsou zaměřeny, jako například skládky, dodávky či časové rozmezí. Každý report přenáší s sebou specifická data, která jsou nezbytná pro jeho účel. Tento "nosič dat" je následně předáván konkrétnímu exportéru, který se specializuje na generování výstupu v nějakém z konkrétních formátů – ať už se jedná o PDF, CSV, XML nebo jiné formáty.

Pro tvorbu reportů ve formátu PDF je využívána knihovna *PDFSharp*, což je robustní nástroj umožňující vysokou míru přizpůsobení výstupu při zachování přesnosti a kvality prezentace dat. Implementace této knihovny do architektury reportingového modulu zajišťuje, že i komplexní reporty mohou být převedeny do PDF formátu s důrazem na vizuální přesnost.



Obrázek 3.28 - Soubory reportů a exportérů
(Zdroj: Vlastní zpracování)

```

99+ references
public enum Exporters
{
    [LocalizedDescriptionResources(typeof(Resources), nameof(Resources.ExportAllSMD))]
    AllDataSMD = 1,

    [LocalizedDescriptionResources(typeof(Resources), nameof(Resources.ExportOneDelivRepPDFWBark))]
    OneDeliveryReportPDFWithBark = 2,

    [LocalizedDescriptionResources(typeof(Resources), nameof(Resources.ExportOneDelivRepPDFWOBark))]
    OneDeliveryReportPDFWithoutBark = 3,

    [LocalizedDescriptionResources(typeof(Resources), nameof(Resources.ExportOneDelivRepPDFWOutputWBark))]
    OneDeliveryReportPDFWithOutputWithBark = 4,

    [LocalizedDescriptionResources(typeof(Resources), nameof(Resources.ExportOneDelivRepPDFWOutputWOBark))]
    OneDeliveryReportPDFWithOutputWithoutBark = 5,

    [LocalizedDescriptionResources(typeof(Resources), nameof(Resources.ExportTimeSpanRepPDF))]
    TimeSpanReportPDF = 6,

    [LocalizedDescriptionResources(typeof(Resources), nameof(Resources.ExportFixedIntervalRepPDF))]
    FixedIntervalReportPDF = 7,

    [LocalizedDescriptionResources(typeof(Resources), nameof(Resources.ExportSawUpListPDF))]
    SawUpListPDF = 8,

    [LocalizedDescriptionResources(typeof(Resources), nameof(Resources.ExportTimeSpanSuppRepPDF))]
    TimeSpanSupplierReportPDF = 9,

    [LocalizedDescriptionResources(typeof(Resources), nameof(Resources.ExportOutStockRepPDF))]
    OutputStockReportPDF = 10,

    [LocalizedDescriptionResources(typeof(Resources), nameof(Resources.ExportOutStockRepCSV))]
    OutputStockReportCSV = 11,
}

```

Obrázek 3.29 - Ukázka enumu exportérů v aplikaci
(Zdroj: Vlastní zpracování)

```

1 reference
public static BaseReport CreateReport(Exporters exporter)
{
    switch (exporter)
    {
        case Exporters.AllDataSMD:
        case Exporters.AllDataSMDTimeSpan:
        case Exporters.TimeSpanReportPDF:
        case Exporters.FixedIntervalReportPDF:
        case Exporters.WorkersTimeSpanReportPDF:
        case Exporters.LengthsAndStockTimeSpanReportPDF:
            return new TimeSpanReport(exporter);

        case Exporters.PrismTimeSpanReportPDF:
            return new TimeSpanReport(exporter) { PrismExport = true };

        case Exporters.OneDeliveryReportPDFWithBark:
        case Exporters.OneDeliveryReportPDFWithoutBark:
        case Exporters.OneDeliveryReportPDFWithOutputWithBark:
        case Exporters.OneDeliveryReportPDFWithOutputWithoutBark:
        case Exporters.FHPDATESAEGEReport:
        case Exporters.VLSReportCSV:
        case Exporters.DeliveryReportCSV:
            return new DeliveryReport(exporter);

        case Exporters.SawUpListPDF:
            return new SawUpListReport(exporter);

        case Exporters.TimeSpanSupplierReportPDF:
            return new TimeSpanSupplierReport(exporter);

        case Exporters.OutputStockReportPDF:
        case Exporters.OutputStockReportCSV:
            return new OutputStockReport(exporter);

        default:
            throw new ArgumentOutOfRangeException($"Unknown exporter [{exporter}]!");
    }
}

```

Obrázek 3.30 - Kód vytvářející reporty na základě typu exportéru
(Zdroj: Vlastní zpracování)

```

2 references
private void _setExporter(Exporters exporter)
{
    this.exporterType = exporter;

    switch (exporter)
    {
        case Exporters.OneDeliveryReportPDFWithBark:
        case Exporters.OneDeliveryReportPDFWithoutBark:
        case Exporters.OneDeliveryReportPDFWithOutputWithBark:
        case Exporters.OneDeliveryReportPDFWithOutputWithoutBark:
            this.exporter = new PDFDeliveryExporter();
            break;

        case Exporters.FHPDATESAEGEReport:
            this.exporter = new FHPDATESAEGEExporter();
            break;

        case Exporters.VLSReportCSV:
            this.exporter = new CSVVLSDeliveryExporter();
            break;

        case Exporters.DeliveryReportCSV:
            this.exporter = new CSVDeliveryExporter();
            break;
    }
}

```

Obrázek 3.31 - Přiřazení konkrétního exportéru v DeliveryReportu
(Zdroj: Vlastní zpracování)

```

4 references
public override void Export()
{
    LoadDeliveryData();

    filePath = Path.Combine(this.exportPath, exportFilename);

    CSVVLSDeliveryHeader header = createDeliveryHeader();
    List<CSVVLSDeliveryRow> rRows = createLogRows();

    bool fileExists = File.Exists(filePath);

    StreamWriter sw = null;

    try
    {
        // If the file already exists this StreamWriter will be set to 'append' mode
        // e.g. when more than one delivery is selected for export
        sw = new StreamWriter(filePath, fileExists, System.Text.Encoding.UTF8);

        if (!fileExists)
        {
            sw.WriteLine(getPHeaders());
        }

        // CSV Delivery Header
        sw.WriteLine(header.ToString());

        // All logs in delivery
        foreach (CSVVLSDeliveryRow row in rRows)
        {
            sw.WriteLine(row.ToString());
        }
    }
    catch (IOException ex)
    {
        throw new ExporterException(
            $"Failed to write to file [{filePath}]: {ex.Message}", ex);
    }
    finally
    {
        sw?.Close();
    }

    this.Filename = filePath;
}

```

Obrázek 3.32 - Příklad exportu dat dodávky do formátu CSV
(Zdroj: Vlastní zpracování)

```

5 references
public interface IDeliveryExporter
{
    34 references
    string Filename { get; set; }

    7 references
    string DeliveryId { get; }

    5 references
    int? DeliverySupplierId { get; }

    9 references
    GlobalMeasurePlaceKey Gmpk { get; set; }

    6 references
    bool RoundVolume();

    6 references
    int RoundDecimalNumbers();

    5 references
    void Export();
}

```

Obrázek 3.33 - Příklad využití rozhraní
(Zdroj: Vlastní zpracování)

```

1 reference
private void drawDeliveryNote()
{
    this.tf.Alignment = XParagraphAlignment.Right;

    int oneEighthOfDrawWidth = getColumnSpanWidth(8, 1);
    this.tf.DrawString($"Příznaky:",
        this.fontBold,
        this.defaultBrush,
        new XRect(DefaultHorizontalPagePadding, this.drawPosition, oneEighthOfDrawWidth - 5, 13),
        XStringFormats.TopLeft);

    this.tf.Alignment = XParagraphAlignment.Left;

    this.tf.DrawString(LocalizeUtils.GetShortTextFlags(this.delivery.NoteType),
        this.fontRegular,
        this.defaultBrush,
        new XRect(DefaultHorizontalPagePadding + oneEighthOfDrawWidth, this.drawPosition, oneEighthOfDrawWidth, 13),
        XStringFormats.TopLeft);
    this.drawPosition += 13;

    if (!string.IsNullOrEmpty(this.delivery.Note))
    {
        string noteDrawString
            = this.delivery.Note.Trim($" {Environment.NewLine}").ToCharArray();

        this.tf.Alignment = XParagraphAlignment.Right;
        this.tf.DrawString("Poznámka:",
            this.fontBold,
            this.defaultBrush,
            new XRect(DefaultHorizontalPagePadding, this.drawPosition, oneEighthOfDrawWidth - 5, 13),
            this.format);

        this.tf.Alignment = XParagraphAlignment.Left;

        XRect shortNoteDrawRect
            = new XRect(DefaultHorizontalPagePadding + oneEighthOfDrawWidth,
                this.drawPosition,
                getColumnSpanWidth(8, 7),
                13);
        this.tf.PrepareDrawString(noteDrawString,
            this.fontRegular,
            shortNoteDrawRect,
            out int lastChar,
            out double neededHeight);
    }
}

```

Obrázek 3.34 - Část kódu vytvářející PDF dokument
(Zdroj: Vlastní zpracování)

```

171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
    bool commentOnLeft = true;
    int commentCount = 0;
    int commentStartPosition = this.drawPosition;
    foreach (string comment in comments)
    {
        this.tf.DrawString(comment,
            this.fontSmall,
            this.defaultBrush,
            new XRect(commentOnLeft ? 50 : this.drawWidth / 2 + 50,
                this.drawPosition, this.drawWidth, 11),
            this.format);
        commentCount++;
        this.drawPosition += 11;
        if (commentOnLeft && (commentCount * 2) >= comments.Count)
        {
            this.drawPosition = commentStartPosition;
            commentOnLeft = false;
        }
    }
    if ((commentCount % 2) > 0)
    {
        this.drawPosition += 11;
    }

    this.drawPosition += 5;
    drawLineAndAdvancePosition(advancePosition: DefaultVerticalSpaceLine);

    table_LogList(this.logs, this.delivery, out decimal precomputedTotalCubage,
        out int precomputedTotalInputLogs);

    this.drawPosition += 26;

```

Obrázek 3.35 - Kód vykreslující spodní část PDF reportu dodávky
(Zdroj: Vlastní zpracování)

Aplikace SMDH Report obsahuje také integrovaný PDF prohlížeč sestavený s využitím komponent DevExpressu. Tato funkčnost umožňuje uživatelům bezprostředně zobrazit generované dokumenty, což usnadňuje jejich kontrolu a případnou korekci dat, před uložením PDF. Soubor se nejdříve vytvoří v dočasné složce, odkud je předán na zobrazení a následně ho může uživatel uložit, což již vygenerovaný soubor přesune na zvolené místo a případně ho přejmenuje. V případě, že soubor není potřeba uchovávat, bude automaticky odstraněn spolu s ostatními dočasnými soubory v rámci pravidelného čištění. Interval čištění lze nastavit v podpůrné Admin aplikaci. Prohlížeč dále nabízí funkci tisku, což zvyšuje efektivitu práce a optimalizuje workflow.

Protokol manipulace jednotlivých kmenů
Drevozprac s.r.o., Pila BRLH, Manipulační linka ZEMBROD

Obsluha: Novotny Jan	Dodavatel: 1.P
Manipulace začátek: 15.04.2024 11:06:23	Číslo dodacího listu: 40050025
Manipulace konec: 17.04.2024 13:36:41	Datum dodávky: 15.04.2024
Evidenční č. dodávky: 230	Dopravce: PPLD
Deklarovaný objem: 27,5 m3	Vozidlo: 9C69168
Deklarovaný počet: 0 kusů	ID konfigurace: PPlasS2015805
Stav: Uzavřená	Certifikace FSC: neudáno
	Certifikace PEFC: 100 %

Příznaky:

Délka = Naměřená délka	Čepový průměr = Čepový průměr v kůře po zaokrouhlení
Průměr osa X = Středový průměr měřený ve vodorovné ose	Odpoč. kůra = Odpočet kůry na středovém průměru
Průměr osa Y = Středový průměr měřený ve svislé ose	Objem = Kubatura dřevní hmoty
Jmen. délka = Zaokrouhlená délka pro výpočet kubatury	Křív. = Křivost kmene
Střed. průměr = Středový průměr po zaokrouhlení	Sbív. = Sbíhavost kmene

Č.	Třída	Kvalita	Dřev.	Délka (m)	Průměr osa X (cm)	Průměr osa Y (cm)	Jmen. délka (m)	Střed. průměr (cm)	Čepový průměr v kůře (cm)	Odpoč. kůra (cm)	Křív. (%)	Sbív. (cm/m)	Objem (m3)
1	D 3b	B	Bo	4,10	39,1	39,1	4	39,00	37,3	1,60	5,1	0,9	0,440
2	D 3a	B	Bo	4,10	32,7	32,8	4	32,00	30,6	1,20	4,1	1,1	0,300
3	D 3a	B	Bo	4,14	32,5	32,4	4	32,00	31,0	1,20	4,4	0,7	0,300
4	D 3b	B	Bo	4,26	38,6	38,6	4	38,00	37,7	1,50	4,5	0,4	0,420
5	D 3a	B	Bo	4,13	34,3	34,3	4	34,00	32,1	1,30	2,6	1,1	0,340
6	D 3a	B	Bo	4,10	35,3	35,3	4	35,00	32,6	1,30	2,9	1,3	0,360
7	D 3a	B	Bo	4,10	33,5	33,4	4	33,00	31,4	1,20	7,9	1,1	0,320
8	D 3b	B	Bo	4,13	37,5	37,5	4	37,00	34,7	1,50	3,8	1,3	0,400
9	D 3a	B	Bo	4,11	34,5	34,5	4	34,00	33,6	1,30	5,9	0,5	0,340
10	D 3b	B	Bo	4,11	39,5	39,5	4	39,00	39,5	1,60	0,0	0,0	0,440
11	D 3b	B	Bo	4,11	38,8	38,7	4	38,00	36,7	1,50	3,2	1,0	0,420
12	D 3a	C	Bo	4,10	32,9	32,9	4	32,00	32,0	1,20	29,4	0,5	0,300

Obrázek 3.36 - Zobrazení vygenerovaného PDF dokumentu v aplikaci
(Zdroj: Vlastní zpracování)

PDF reporty generované v aplikaci zahrnují nejen přehledné výstupy dat, ale také srovnání s deklaroványi hodnotami a zaznamenávají případné odchylky. Zároveň je při exportu uživatelům nabídnuta možnost, které sumáře chtějí zahrnout do dokumentu a případně zda chtějí k exportu připsat poznámku. Aktuálně mohou uživatelé zvolit rozpad dat dodávky podle kubírovacích délek, typů dřevin a kvalit.

97	D 2b	C	Bo	4,09	30,0	29,9	4	29,00	28,5	1,00	3,1	0,7	0,250
98	D 3b	B	Bo	4,10	38,0	37,9	4	37,00	36,8	1,50	2,4	0,6	0,400
99	D 2a	D	Bo	4,10	22,8	22,8	4	22,00	21,7	0,70	3,6	0,6	0,140

Celkový objem: 28,44 m3
 Celkem přijatých kusů: 99
 Deklarovaný objem: 27,5
 Překontrolovaný objem: 0
 Přijatý objem: 28,44
 Celkem kusů: 99

Rozdíl dekl. objemu: -0,94 m3 (-3,4 %)
 Rozdíl dekl. přijatých kusů: -99

Vygenerováno 27.04.2024, Smdh Report 1.14.1.1
 Drevozprac s.r.o., Pila BRLH, Manipulační linka ZEMBROD

Strana 3

Obrázek 3.37 - Porovnání s deklaroványi hodnotami v reportu
(Zdroj: Vlastní zpracování)

Celkový protokol						
Drevozprac s.r.o., Pila BRLH, Manipulační linka ZEMBROD						
Obsluha: Novotny Jan		Dodavatel: LCR				
Manipulace začátek: 03.04.2024 13:17:07		Číslo dodacího listu: 2405463				
Manipulace konec: 03.04.2024 14:07:05		Datum dodávky: 28.03.2024				
Evidenční č. dodávky: 13		Dopravce: PPLD				
Deklarovaný objem: 25,78 m3		Vozidlo: 9C30375				
Deklarovaný počet: 0 kusů		ID konfigurace: PFI5asB2015805				
Stav: Uzavřená		Certifikace FSC: neudáno				
		Certifikace PEFC: 100 %				
Příznaky:						
Dřevina	Třída	Kvalita	Počet (ks)	Objem (m3)	Podíl (%)	
Borovice	D 2a	C	1	0,18	0,69	
		Celkem	1	0,18	0,69	
	D 2b	C	3	0,67	2,55	
		VL	1	0,20	0,76	
		Celkem	4	0,87	3,31	
	D 3a	C	1	0,34	1,30	
		D	1	0,30	1,14	
		Celkem	2	0,64	2,44	
	D 3b	B	3	1,39	5,30	
		C	13	5,82	22,17	
		D	3	1,37	5,22	
		Celkem	19	8,58	32,69	
	D 4	B	9	5,60	21,33	
		C	9	4,95	18,86	
		D	6	3,52	13,41	
		Celkem	24	14,07	53,60	
	D 5	C	2	1,91	7,28	
		Celkem	2	1,91	7,28	
	Celkem			52	26,25	100,00
	Kubírovací délka (m)	Počet (ks)	Kubatura celkem (m3)	Kubatura na kus (m3)	Podíl (%)	
4,00	52	26,25	0,505	100,00		

Obrázek 3.38 - Rozpad dat do kategorií v reportu dodávky
(Zdroj: Vlastní zpracování)

Aplikace nabízí také možnost vytvářet přehledy dodavatelů. Tento report poskytuje vedení přehledné informace o dodavatelích, množství jimi dodaného dřeva, jeho kubatuře a kvalitě. Tato data jsou velmi užitečná pro strategické rozhodování a optimalizaci dodavatelských vztahů. Díky tomuto reportu může vedení efektivněji plánovat a spravovat zdroje.

Další užitečnou funkcionalitou je sledování výkonosti operátorů, které se rovněž exportuje do PDF reportů. Tato možnost umožňuje vedení získat podrobný přehled o pracovním nasazení jednotlivých zaměstnanců. Reporty o výkonosti operátorů mohou sloužit jako nástroj pro rozhodování o odměnách a jsou základem pro další kroky vedoucí k optimalizaci pracovního procesu. Tento report je zvláště užitečný pro hodnocení efektivity a produktivity zaměstnanců na jednotlivých úsecích práce.

Protokol výkon operátorů				
Společnost 01, Pila 01, všechna měřicí místa				
Od: 22.04.2022		Směr zpracování: Vstup do procesu		
Do: 30.04.2022				
Operator Andrej (operator02)				
Datum	Počet (ks)	Kubatura na kus (m3)	Podíl (%)	Kubatura celkem (m3)
PÁ 22.04.2022	120	0,251	6,85	30,17
SO 23.04.2022	280	0,285	18,10	79,73
NE 24.04.2022	---	---	---	---
PO 25.04.2022	98	0,318	7,07	31,16
ÚT 26.04.2022	262	0,251	14,95	65,84
ST 27.04.2022	221	0,354	17,74	78,13
ČT 28.04.2022	159	0,358	12,91	56,86
PÁ 29.04.2022	152	0,442	15,27	67,24
SO 30.04.2022	93	0,337	7,11	31,30
Celkem	1385	0,318	100,00	440,43
Operator Milan (operator01)				
Datum	Počet (ks)	Kubatura na kus (m3)	Podíl (%)	Kubatura celkem (m3)
PÁ 22.04.2022	---	---	---	---
SO 23.04.2022	---	---	---	---
NE 24.04.2022	---	---	---	---
PO 25.04.2022	202	0,446	24,19	90,06
ÚT 26.04.2022	141	0,501	18,97	70,60
ST 27.04.2022	249	0,289	19,36	72,06
ČT 28.04.2022	181	0,398	19,35	72,02
PÁ 29.04.2022	163	0,414	18,14	67,52
SO 30.04.2022	---	---	---	---
Celkem	936	0,398	100,00	372,26

Obrázek 3.39 - Report výkonu zaměstnanců
(Zdroj: Vlastní zpracování)

Jak již bylo zmíněno v kapitole 3.3.3, reportovací modul umožňuje také vytvářet inventury konkrétních skládek, které jsou prezentovány v podobě PDF reportů. Tyto reporty poskytují detailní přehled kmenů na dané skládce a zahrnují také možnost volby sumářů, které data rozdělují do specifických kategorií. Tato funkce umožňuje uživatelům lepší orientaci v množství a typu dřeva konkrétní skládky, což je důležité pro efektivní správu zásob.

Inventura skládky sk Stredni k datu 26.04.2024 Pila Pod Kamenem, s.r.o., Pod kaminkem, Kancel												
Název skládky: sk Stredni Zkratka skládky: sk03								Datum: 26.04.2024 21:15:17 Vystavil: Adam Jan				
Č.	Třída	Kvalita	Dřev.	Délka (m)	Průměr osa X (cm)	Průměr osa Y (cm)	Jmen. délka (m)	Střed. průměr (cm)	Čepový průměr (cm)	Kůra	Odpoč. kůra (cm)	Objem (m3)
1	D 1a	C	Sm	4,00	11,0	11,0	4	11,00	11,0	Ano	0,70	0,038
2	D 1a	C	Sm	3,00	14,0	14,0	3	14,00	11,0	Ano	0,80	0,046
3	D 1a	TEST	Sm	7,00	11,0	11,0	7	11,00	11,2	Ano	0,70	0,067
4	D 1a	TEST	Sm	5,00	11,0	11,0	5	11,00	11,5	Ano	0,70	0,048
5	D 1a	TEST	Sm	3,00	11,0	11,0	3	11,00	11,4	Ano	0,70	0,029
6	D 1a	TEST	Sm	2,00	11,0	11,0	2	11,00	10,1	Ano	0,70	0,019
7	D 1a	TEST	Sm	2,00	14,0	14,0	2	14,00	11,8	Ano	0,80	0,031
8	D 1a	TEST	Sm	3,00	11,0	11,0	3	11,00	10,7	Ano	0,70	0,029
9	D 1b	TEST	Sm	3,00	16,0	16,0	3	16,00	11,6	Ano	0,80	0,060
10	D 1a	TEST	Sm	4,00	11,0	11,0	4	11,00	10,5	Ano	0,70	0,038
11	D 1a	TEST	Sm	9,00	13,0	13,0	9	13,00	11,2	Ano	0,80	0,119
12	D 1a	TEST	Sm	5,00	12,0	12,0	5	12,00	11,6	Ano	0,80	0,057
13	D 1a	A/B	Sm	2,00	11,0	11,0	2	11,00	11,1	Ano	0,70	0,019
14	D 1a	A/B	Sm	1,00	11,0	11,0	1	11,00	10,1	Ano	0,70	0,010
15	D 1a	A/B	Sm	3,00	10,0	10,0	3	10,00	10,2	Ano	0,70	0,024
16	D 1a	A/B	Sm	4,00	12,0	12,0	4	12,00	12,0	Ano	0,80	0,045
17	D 1a	A/B	Sm	6,00	12,0	12,0	6	12,00	11,2	Ano	0,80	0,068
18	D 1a	A/B	Sm	6,00	11,0	11,0	6	11,00	10,6	Ano	0,70	0,057

Obrázek 3.40 - Část inventury skládky v exportovaném PDF souboru
(Zdroj: Vlastní zpracování)

Při generování PDF dokumentů v aplikaci SMDH Report se data vykreslují od horní části dokumentu směrem dolů. Tento způsob tvorby vyžaduje specifické úpravy v procesu vývoje, aby bylo zajištěno správné uspořádání a prezentace informací. Jak již bylo zmíněno, aplikace umožňuje, kromě PDF formátu, exportovat data také do CSV souborů. V tomto případě se vytvoří kolekce dat, která se postupně zapisuje do souboru podle definovaných hlaviček, řádek po řádku. Další možností je export do XML, kde jsou data strukturována a uzavřena do předem definovaných tagů, což zajišťuje jejich správné uspořádání a snadnou čitelnost. Tyto metody reportů poskytují flexibilní a efektivní způsoby pro zpracování a také sdílení informací podle potřeb uživatelů.

3.4 Integrace a nasazení nové verze

Tato kapitola se věnuje integraci a nasazení aplikace SMDH Report v rámci projektu SMDH. Jsou zde popsány postupy a techniky propojení této nové aplikace se stávajícími systémy projektu, což zahrnuje integraci jednotlivých komponent aplikace do celkové struktury projektu. Dále kapitola představuje proces nasazení aplikace a její uvedení do provozu u zákazníků, včetně nezbytných kroků zajišťujících plynulý přechod a optimální funkčnost systému.

3.4.1 Integrace s existujícími aplikacemi

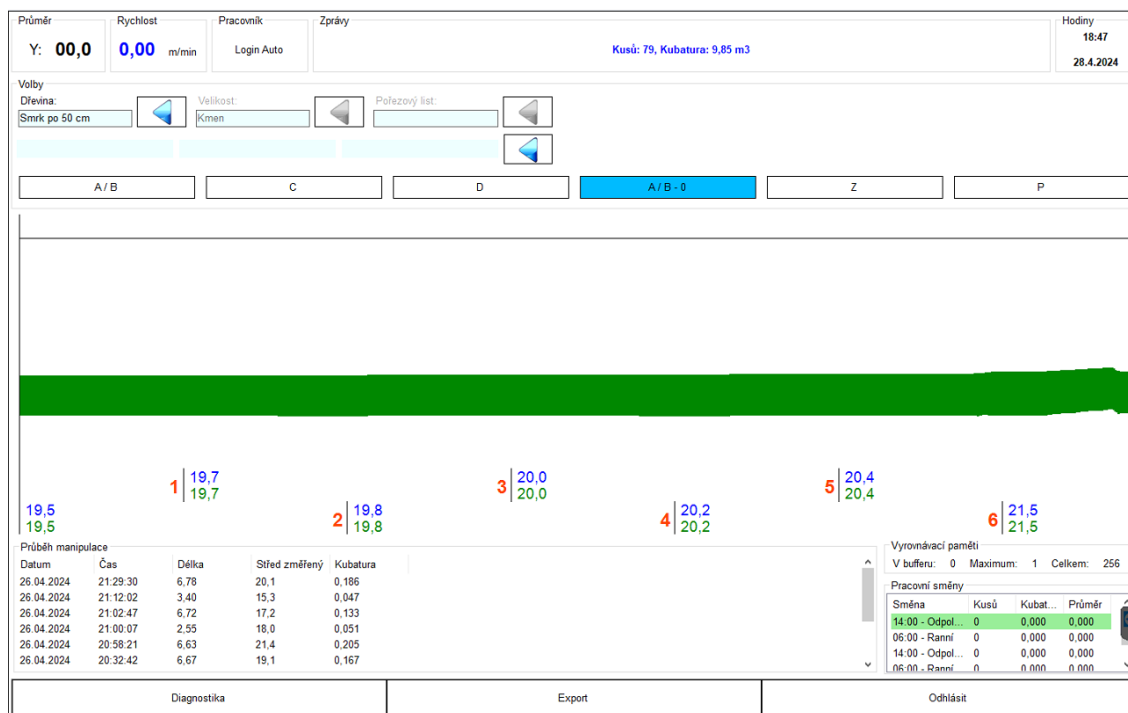
Tato část práce zdůrazňuje význam modulární architektury, která je v projektu SMDH klíčová. Vzhledem k tomu, že aplikace SMDH Report je integrována do solution SMDH, je možné jednoduše a rychle sdílet komponenty napříč projekty, což napomáhá efektivitě a kohezi systému.

Významnou součástí integrace je rovněž rozšíření možností exportu dat. Tato funkcionality je přidána i do ostatních aplikací a zároveň byla přizpůsobena dotykovému ovládání, aby byla přístupná pro operátory přímo na měřicích místech. Tito uživatelé mají možnost generovat sestavy přímo relevantní pro své specifické měřicí místo, ale pouze z dat, která jsou lokálně dostupná na dané stanici. Typy povolených exportů jsou spravovány prostřednictvím licencí, které se načítají při každém spuštění aplikace.

Jak bylo vysvětleno v kapitole 3.3.1 i naopak do aplikace SMDH Report jsou integrovány některé *UserControls*, které byly dříve použité na jiných místech. Jedním z takových příkladů je zmiňovaný typ tlačítka *TouchButton*. Tyto tlačítka jsou navrženy tak, aby poskytovaly pouze základní vzhled a logiku. Konkrétní akce vyvolané po aktivaci tlačítka, tj. co se má stát po jeho zmáčknutí, je nutné definovat individuálně pro každé místo použití v aplikaci. Tento přístup umožňuje vývojářům flexibilně přizpůsobit funkčnost tlačítek specifickým potřebám jednotlivých částí aplikace, což zvyšuje jejich praktickou užitečnost a integraci do UI.

Další integrovanou komponentou je správa číselníků, přístupná v *MenuStripu* pod záložkou "*Číselníky*". Zde mohou uživatelé spravovat seznamy dodavatelů, kvalit a uživatelské účty. Tyto možnosti jsou běžně dostupné v podpůrné aplikaci SMDH

Admin, ale pro zvýšení uživatelského komfortu a autonomie byly implementovány i do Reportu. Tímto způsobem mohou zákazníci jednoduše a efektivně spravovat nezbytné informace přímo z jednoho místa.



Obrázek 3.41 - Aplikace SMPD s možností exportovat data
(Zdroj: Vlastní zpracování)

3.4.2 Nasazení aplikace, uvedení do provozu a správa verzí

Tato kapitola se zaměřuje na postupy a metodiky, které jsou nezbytné pro správnou instalaci a konfiguraci aplikace v reálném prostředí u zákazníka. Vzhledem k tomu, že SMDH Report je samostatná aplikace, je možné ji nasazovat obdobným způsobem jako ostatní měřicí aplikace v rámci projektu.

Instalační balíčky pro SMDH Report a další aplikace projektu byly vytvořeny pomocí nástroje InnoSetup. Tento nástroj umožňuje sestavení instalačního scriptu, který generuje instalační balíčky ve vybraných verzích. Díky této metodě je možné vytvářet updaty jak pro nejnovější verze, tak i updaty, které opravují jen konkrétní chyby ve starších verzích.

Proces instalace na měřicích stanicích zahrnuje kroky, jako jsou kalibrace senzorů a další technická nastavení, které jsou nutné pro zajištění přesnosti měření. Z tohoto důvodu je důležité mít možnost efektivně spravovat zařízení na dálku. V současnosti se pro tyto

účely využívá software TeamViewer, což umožňuje programátorům snadno a rychle přistupovat k počítačům klientů a provádět potřebné úkony vzdáleně.

Nasazení aplikace SMDH Report tedy probíhá tak, že instalátor aplikace je přenesen na cílový počítač prostřednictvím TeamViewera. Následně je aplikace nainstalována a přizpůsobena podle specifických požadavků a preferencí zákazníka. Tento přístup je velmi jednoduchý a je stejný napříč všemi aplikacemi SMDH.

Aktualizace verze na novější probíhá obdobným způsobem jako původní instalace. Software SMDH, včetně aplikace SMDH Report, je nastaven tak, že zákazník při koupi získává konkrétní verzi aplikace spolu s údržbou. Jakékoliv další úpravy nebo rozšíření funkcionalit si však musí zákazník dokoupit, jelikož je potřeba software konkrétně upravit. Tento model způsobuje, že aplikace nejsou aktualizovány příliš často, což umožňuje udržet proces aktualizace v režimu manuálního updatování. V případě, že zákazník vyžaduje specifickou úpravu nebo přidání nové funkcionality, je nová verze aplikace nainstalována ručně, opět prostřednictvím TeamViewera.

Tento postup zajišťuje, že všechny aplikace v rámci projektu SMDH, včetně SMDH Report, jsou aktualizovány konzistentně a v souladu s potřebami a požadavky klientů. Díky tomu je možné zajistit, že každý zákazník má přístup k funkcím, které jsou pro něj potřebné.

3.5 Ekonomické zhodnocení

Tato kapitola se zabývá vyčíslením celkových nákladů na návrh a implementaci aplikace SMDH Report. Aktivita jsou pro přehlednost spojeny do větších aktivit a doby trvání jsou zaokrouhleny na čtvrtiny *man day* (MD), vzhledem k tomu, že ve firemním systému jsou časové údaje zaznamenávány v tomto formátu. Následující tabulka ukazuje přehled hlavních činností, jejich dobu trvání a také vyčíslení celkové ceny. Tato tabulka tak poskytuje přehled o celkovém rozpočtu projektu.

Tabulka 3.7 - Přehled hlavních činností
(Zdroj: Vlastní zpracování)

Činnost	Doba trvání (MD)	Suma (Kč)
Analýza požadavků	2,5	7000
Návrh systému	4	11200
Implementace UI	7	19600
Implementace back-endu	14	39200
Integrace a testování	8,5	23800
Dokumentace	1,25	3500
Sestavení a nasazení	0,5	1400
Celkem	37,75	105700

Celkové náklady projektu byly vyčísleny na 105700 Kč. Tato suma byla odvozena od počtu man days násobeným průměrnou hodinovou mzdou ve výši 350 Kč za hodinu, přičemž jeden man day představuje standardní osmihodinový pracovní den. Tento přístup k odhadu nákladů poskytuje transparentní a opakovatelný základ pro finanční plánování a odůvodnění investic do vývoje softwaru.

3.6 Přínosy řešení

Tato část práce se soustředí na výhody, které přináší aplikace SMDH Report do procesů automatizovaných pilnic. Jedním z klíčových přínosů je schopnost zaměstnanců v kancelářích prohlížet a analyzovat naměřená data prostřednictvím uživatelsky přívětivé aplikace. Díky centralizovanému náhledu na data ze všech měřicích míst mají uživatelé možnost data efektivně spravovat, filtrovat a porovnávat přímo v rámci jedné aplikace, aniž by museli přistupovat k samotné databázi a složité data získávat.

Dalším významným přínosem je možnost úpravy již naměřených dat přímo uživatelem aplikace, což původně vyžadovalo složité a rizikové zásahy do databáze externím programátorem. Uživatelé nyní mohou přímo z aplikace upravovat nebo odstraňovat chybné záznamy, čímž je zajištěna vyšší integrita dat a eliminace rizik spojených s přímou manipulací databáze. Tato funkcionality umožňuje i korekci špatně zvolené kvality, či typu dřeviny nebo jiných parametrů, které byly dříve zadány nesprávně. Díky tomu již firmy nemusí zajistit služby externího programátora k opravám těchto chyb přímo

v databázi, což představuje finanční úsporu. Eliminace potřeby externího zásahu nejen snižuje náklady, ale také zkracuje čas potřebný pro řešení těchto problémů, což přispívá k celkové efektivitě procesů v rámci firmy.

Významnou inovací je rovněž implementace exportní funkce, která nahrazuje dříve pracné a časově náročné ruční vytváření reportů. Aplikace SMDH Report poskytuje možnost generování mnoha reportů, které lze snadno exportovat pro další použití. Tento přístup nejenže značně usnadňuje práci s daty, ale také umožňuje společně efektivně přenášet data do dalších systémů, které jsou součástí jejich pracovních procesů, pomocí specifických CSV, XML či jiných souborů.

Navíc, díky těmto reportům již u měřicích stanic není třeba osobní záznam o příchozích kmenech. Měření probíhá standardně a výsledky jsou dostupné v reportech, které lze v kanceláři porovnat s očekávanými hodnotami. Tento systém nejen zvyšuje přesnost záznamů, ale také výrazně zefektivňuje logistické operace a správu inventáře, což přináší další úspory času a nákladů pro firmu.

3.7 Možné budoucí vylepšení

Projekt SMDH má samozřejmě také řadu možností, které by se do budoucna daly implementovat, aby přinesli další výhody jak pro uživatele, tak pro provozovatele. Jako první krok se nabízí přenesení správy licencí do cloudu, což by umožnilo ověřování licencí v cloudovém prostředí a zjednodušilo správu. Nevýhodou je nutnost stálého internetového připojení měřicích stanic.

Další možnou změnou by mohl být přechod na model založený na předplatném. Ten by zákazníkům garantoval přístup k nejnovějším verzím softwaru, což by zahrnovalo všechny aktualizace. Tento přístup by zjednodušil správu verzí softwaru u klientů, což by vedlo k lepší kompatibilitě a snadnějšímu řešení problémů. Dalším krokem by mohlo být přenesení celé aplikace SMDH Report do cloudu, což by umožnilo přístup přes internet z jakéhokoli místa a zařízení, a poskytlo ještě větší flexibilitu a přístupnost.

Inovace v oblasti správy databází by mohla zahrnovat implementaci Entity Frameworku a jeho migrací, pro zjednodušení změn v databázi a jednodušší integraci nových funkcí. Tento přístup by zrychlil vývoj nových funkcí a zajistil vysokou úroveň integrity a bezpečnosti dat.

ZÁVĚR

Závěr této diplomové práce slouží jako shrnutí dosažených cílů a hlavních výsledků. Cílem práce bylo navrhnout a implementovat reportingový modul SMDH Report pro software využívaný v automatizovaných pilnicích. Modul měl poskytnout uživatelům efektivní nástroje pro sledování, úpravu a sdílení dat z měřicích stanic, a to vše bylo úspěšně realizováno.

První část práce se zaměřila na teoretická východiska, kde byly vysvětleny základní pojmy a koncepty jako automatizace a životní cyklus softwaru. Důraz byl kladen na pochopení potřebných technologií a metodologií, které byly použity v dalších fázích projektu.

Druhá kapitola analyzovala aktuální situaci a zdůvodnila potřebu implementace nového modulu. Zahrnovala analýzy vnitřního a vnějšího prostředí společnosti a poukázala na specifické nedostatky, které reportovací modul má řešit.

Třetí část práce se podrobně věnovala samotnému návrhu a implementaci řešení. Byl navržen UI, implementována back-endová logika a provedena integrace s existujícími systémy. V této části byly také popsány přínosy nově vytvořeného řešení, navržena potenciální vylepšení pro budoucí rozvoj a provedeno ekonomické zhodnocení projektu.

Závěrem lze konstatovat, že všechny stanovené cíle byly úspěšně splněny a aplikace SMDH Report je nyní nasazena v produkčním prostředí projektu SMDH a již byla implementována u zákazníků. Díky tomu mohou firmy efektivněji využívat data z měřicích stanic, což vede k lepšímu rozhodování a optimalizaci firemních procesů.

SEZNAM POUŽITÉ LITERATURY

1. Sodomka, Petr a Klčová, Hana. *Informační systémy v podnikové praxi. 2., aktualiz. a rozš. vyd.* Brno : Computer Press, 2010. 978-80-251-2878-7.
2. Automation. *Britannica*. [Online] [Citace: 17. Únor 2024.] <https://www.britannica.com/technology/automation>.
3. Výroba dřevoobráběcích strojů. *Af-Servis*. [Online] [Citace: 17. Únor 2024.] <https://af-servis.cz/>.
4. Automation and Robotization of processes across applications. *Intedo*. [Online] [Citace: 17. Únor 2024.] <https://intedo.cz/en/home/automation-and-robotization/>.
5. Tajak, Mateusz. What is the difference between automation and robotization? *GGs IT Consulting*. [Online] 2. 2 2021. [Citace: 17. Únor 2024.] <https://ggsitc.com/blog/what-is-the-difference-between-automation-and-robotization>.
6. What is robotic process automation (RPA)? *IBM*. [Online] [Citace: 18. Únor 2024.] <https://www.ibm.com/topics/rpa>.
7. What is RPA? *Microsoft PowerAutomate*. [Online] [Citace: 18. Únor 2024.] <https://powerautomate.microsoft.com/en-us/what-is-rpa/>.
8. Robotic Process Automation (RPA). *Ui Path*. [Online] [Citace: 18. Únor 2024.] <https://www.uipath.com/rpa/robotic-process-automation>.
9. Basl, Josef a Blažíček, Roman. *Podnikové informační systémy: podnik v informační společnosti. 3., aktualiz. a dopl. vyd.* Praha : Grada, 2012. 978-80-247-4307-3.
10. Govender, Dhashen. SOVTECH. *The Power of Automation with Custom Software in Your Business*. [Online] 30. Květen 2023. [Citace: 22. Únor 2024.] <https://www.sovtech.com/blog/the-power-of-automation-with-custom-software-in-your-business>.
11. Schalkwyk, Alexis van. SOVTECH. *The Benefits Of Custom Software For Your Business*. [Online] [Citace: 22. Únor 2024.] <https://www.sovtech.com/blog/the-benefits-of-custom-software-for-your-business>.

12. Gála, Libor, Pour, Jan a Šedivá, Zuzana. *Podniková informatika: počítačové aplikace v podnikové a mezipodnikové praxi. 3., aktualizované vydání*. Praha : Grada, 2015. 978-80-247-5457-4.
13. What is Industry 4.0? *IBM*. [Online] [Citace: 25. Únor 2024.] <https://www.ibm.com/topics/industry-4-0>.
14. What is Industry 4.0? *ISA*. [Online] [Citace: 25. Únor 2024.] <https://blog.isa.org/what-is-industry-40>.
15. Boyde, Joshua. *A Down-To-Earth Guide To SDLC Project Management*. místo neznámé : CreateSpace Independent Publishing Platform, 2014. 1499619189.
16. What is SDLC (Software Development Lifecycle)? *AWS*. [Online] [Citace: 30. Leden 2024.] [https://aws.amazon.com/what-is/sdlc/#:~:text=The%20software%20development%20lifecycle%20\(SDLC,expectation s%20during%20production%20and%20beyond..](https://aws.amazon.com/what-is/sdlc/#:~:text=The%20software%20development%20lifecycle%20(SDLC,expectation s%20during%20production%20and%20beyond..)
17. Pressman, Roger. *Software Engineering: A Practitioner's Approach*. New York : McGraw-Hill Education, 2009. 978-0-07-337597-7.
18. SDLC - Waterfall Model. *TutorialsPoint*. [Online] [Citace: 11. Únor 2024.] https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm.
19. Lutkevich, Ben. What is the waterfall model? *TechTarget*. [Online] [Citace: 11. Únor 2024.] <https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model>.
20. Procházka, Jaroslav a Klimeš, Cyril. *Provozujte IT jinak: agilní a štíhlý provoz, podpora a údržba informačních systémů a IT služeb*. Praha : Grada, 2011. 978-80-247-4137-6.
21. Sutherland, Jeffrey. *Scrum: the art of doing twice the work in half the time*. New York : Crown Business, 2014. 978-0385346450.
22. The Home of Scrum. *SCRUM*. [Online] [Citace: 15. Únor 2024.] <https://www.scrum.org/>.
23. CLAIRE DRUMOND. Is the Agile Manifesto still a thing? *Atlassian*. [Online] [Citace: 15. Únor 2024.] <https://www.atlassian.com/agile/manifesto>.

24. What Is Agile? And When to Use It. *Coursera*. [Online] 20. Nov 2023. [Citace: 15. Únor 2024.] <https://www.coursera.org/articles/what-is-agile-a-beginners-guide>.
25. SDLC - Iterative Model. *TutorialsPoint*. [Online] [Citace: 4. Únor 2024.] https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm.
26. Miraz, Mahdi a Ali, Maaruf. Blockchain Enabled Smart Contract Based Applications: Deficiencies with the Software Development Life Cycle Models. *Baltica Journal*. 1, 2020, Sv. 33.
27. SDLC - Hybrid Model. *TutorialsCampus*. [Online] [Citace: 16. Únor 2024.] <https://www.tutorialscampus.com/sdlc/hybrid-model.htm>.
28. SDLC - Spiral Model. *TutorialsPoint*. [Online] [Citace: 16. Únor 2024.] https://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm.
29. Schwalbe, Kathy. *Řízení projektů v IT: kompletní průvodce*. Brno : Computer Press, 2011. 978-80-251-2882-4.
30. What is an IDE (Integrated Development Environment)? *AWS*. [Online] [Citace: 27. Únor 2024.] <https://aws.amazon.com/what-is/ide/>.
31. What is version control? *Atlassian*. [Online] [Citace: 27. Únor 2024.] <https://www.atlassian.com/git/tutorials/what-is-version-control>.
32. Visual Studio. *Microsoft*. [Online] [Citace: 27. Únor 2024.] <https://visualstudio.microsoft.com/vs/>.
33. *GitHub*. [Online] [Citace: 28. Únor 2024.] <https://github.com/>.
34. About. *GitLab*. [Online] [Citace: 28. Únor 2024.] <https://about.gitlab.com/>.
35. Shah, Jigar. GitHub vs GitLab: Which is the Best in 2024? *Radix*. [Online] 6. Nov 2023. [Citace: 28. Únor 2024.] <https://radixweb.com/blog/github-vs-gitlab>.
36. Getting Started - What is Git? *Git-SCM*. [Online] [Citace: 28. Únor 2024.] <https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F>.
37. Programming languages. *CodeCademy*. [Online] [Citace: 16. Leden 2024.] <https://www.codecademy.com/resources/blog/programming-languages/>.

38. Tuama, Daragh Ó. What is Programming Language. *CodeInstitute*. [Online] [Citace: 18. Leden 2024.] <https://codeinstitute.net/global/blog/what-is-a-programming-language/>.
39. Ferguson, Stuart J. a Hebels, Rodney. *Computers for Librarians: An Introduction to the Electronic Library*. místo neznámé : Chandos Publishing, 2003. 1876938609.
40. Gabbrielli, Prof. Dr. Maurizio a Martini, Prof. Dr. Simone. *Programming Languages: Principles and Paradigms*. Bologna, Italy : Springer, 2010. 978-1-84882-913-8.
41. Preston, Robert. What Is a Procedural Programming Language? *indeed*. [Online] [Citace: 10. Leden 2024.] <https://www.indeed.com/career-advice/career-development/procedural-programming-language#:~:text=A%20procedural%20programming%20language%20is%20one%20th at%20uses%20sets%20of,incluing%20BASIC%2C%20C%20and%20Pascal..>
42. Functional Programming Paradigm. *GeeksForGeeks*. [Online] [Citace: 14. Leden 2024.] <https://www.geeksforgeeks.org/functional-programming-paradigm/>.
43. Stroustrup, Bjarne. *What is object-oriented programming?* New Jersey : IEEE, 1988. DOI 10.1109/52.2020.
44. Object Oriented Programming. *TechTarget*. [Online] [Citace: 7. Leden 2024.] <https://www.techtarget.com/searcharchitecture/definition/object-oriented-programming-OOP>.
45. Logic Programming Languages. *Linode*. [Online] [Citace: 20. Leden 2024.] <https://www.linode.com/docs/guides/logic-programming-languages/>.
46. Coursera. *Types of Programming Languages*. [Online] [Citace: 29. Listopad 2023.] <https://www.coursera.org/articles/types-programming-language>.
47. A tour of the C# language. *Microsoft Documentation*. [Online] 04. 05 2023. [Citace: 15. Leden 2024.] <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>.
48. History of SQL. *Oracle Docs*. [Online] Oracle. [Citace: 27. Prosinec 2023.] <https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/History-of-SQL.html#GUID-4DD5E1B6-BEC7-4E9B-B369-1466F93ACA28>.
49. What is SQL? *AWS Amazon*. [Online] AWS. [Citace: 27. Prosinec 2023.] <https://aws.amazon.com/what-is/sql/>.

50. Russell, Jordan. Inno Setup. *Jordan Russell's Software*. [Online] [Citace: 27. Leden 2024.] <https://jrsoftware.org/isinfo.php>.
51. .NET documentation. *Microsoft*. [Online] [Citace: 29. Únor 2024.] <https://learn.microsoft.com/en-us/dotnet/>.
52. Overview of .NET Framework. *Microsoft Developer Network*. [Online] [Citace: 1. Březen 2024.] <https://learn.microsoft.com/en-us/dotnet/framework/get-started/overview>.
53. Introduction to NET. *Microsoft Developer Network*. [Online] 10. 1 2024. [Citace: 1. Březen 2024.] <https://learn.microsoft.com/en-us/dotnet/core/introduction>.
54. What Is .Net? *Microsoft Developer Network*. [Online] [Citace: 1. Březen 2024.] <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>.
55. Windows Forms overview. *Microsoft Developer Network*. [Online] [Citace: 3. Březen 2024.] <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/windows-forms-overview?view=netframeworkdesktop-4.8>.
56. WPF overview. *Microsoft Developer Network*. [Online] [Citace: 2. Březen 2024.] <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/introduction-to-wpf?view=netframeworkdesktop-4.8>.
57. What is .NET MAUI? *Microsoft Developer Network*. [Online] [Citace: 3. Březen 2024.] <https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui?view=net-maui-8.0>.
58. Avalonia UI. *Avaloniaui.net*. [Online] [Citace: 4. Březen 2024.] <https://avaloniaui.net/>.
59. What Is a Database? *Oracle*. [Online] [Citace: 2. Březen 2024.] <https://www.oracle.com/database/what-is-database/>.
60. SQLite.org. *What Is SQLite?* [Online] [Citace: 4. Březen 2024.] <https://www.sqlite.org/>.
61. What is MySQL? *MySQL*. [Online] [Citace: 4. Březen 2024.] <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>.
62. MariaDB. *GitHub*. [Online] [Citace: 5. Březen 2024.] <https://github.com/MariaDB/server>.
63. *MariaDB*. [Online] [Citace: 5. Březen 2024.] <https://mariadb.org/>.

64. What is software testing? *IBM*. [Online] [Citace: 6. Březen 2024.] <https://www.ibm.com/topics/software-testing>.
65. Martin, Robert C. *Clean Code: A Handbook of Agile Software Craftsmanship*. Upper Saddle River : Pearson, 2008. 9780132350884.
66. *Unit Testing and Analysis*. Morell, Larry. Arkansas : Arkansas Tech University, 1989.
67. Alena, Hanzelková, Miloslav, Keřkovský a Vykypěl, Oldřich. *Strategické řízení: teorie pro praxi. 3. přepracované vydání*. Praha : C.H. Beck, 2017. 978-80-7400-637-1.
68. Ortiz, Jaime. *The global environment through the SLEPT framework*. místo neznámé : International Journal of Business and Globalisation, 2010. 10.1504/ijbg.2010.035479.
69. Gratton, Peter. Porter's Five Forces Explained and How to Use the Model. *Investopedia*. [Online] [Citace: 27. Březen 2024.] <https://www.investopedia.com/terms/p/porter.asp>.
70. Enduring Ideas: The 7-S Framework. *McKinsey & Company*. [Online] 1. Březen 2008. [Citace: 28. Březen 2024.] <https://www.mckinsey.com/capabilities/strategy-and-corporate-finance/our-insights/enduring-ideas-the-7-s-framework>.
71. Kenton, Will. SWOT Analysis: How To With Table and Example. *Investopedia*. [Online] 30. Říjen 2023. [Citace: 29. Březen 2024.] <https://www.investopedia.com/terms/s/swot.asp>.
72. BJSW. Oficiální stránky BJSW. *About Us*. [Online] 2023. [Citace: 12. Prosinec 2023.] <https://www.bjsw.cz/index.php?lang=en>.
73. Vyšla Statistická ročenka 2023. *Český Statistický Úřad*. [Online] 27. Listopad 2023. [Citace: 15. Březen 2024.] <https://www.czso.cz/csu/czso/vysla-statisticka-rocenka-2023>.
74. Aktuální populační vývoj v kostce. *Český Statistický Úřad*. [Online] [Citace: 15. Březen 2024.] <https://www.czso.cz/csu/czso/aktualni-populacni-vyvoj-v-kostce>.
75. Stárnutí se nevyhneme. *Český Statistický Úřad*. [Online] [Citace: 15. Březen 2024.] <https://www.czso.cz/csu/czso/ea002b5947>.
76. Economic forecast for Czechia. *European Commission*. [Online] [Citace: 16. Březen 2024.] https://economy-finance.ec.europa.eu/economic-surveillance-eu-economies/czechia/economic-forecast-czechia_en.

77. Czech Republic GDP. *Focus Economics*. [Online] [Citace: 16. Březen 2024.]
<https://www.focus-economics.com/country-indicator/czech-republic/gdp/>.
78. Czech Republic: Staff Concluding Statement of the 2023 Article IV Mission. *International Monetary Fund*. [Online] [Citace: 16. Březen 2024.]
<https://www.imf.org/en/News/Articles/2023/11/21/mcs112123-czech-republic-staff-concluding-statement-of-the-2023-article-iv-mission>.

SEZNAM OBRÁZKŮ

Obrázek 1.1 - Přehled čtyř průmyslových revolucí	17
Obrázek 1.2 - Iterativní model SDLC	21
Obrázek 1.3 - Vzhled a možnosti programu Visual Studio	25
Obrázek 1.4 - Příklad struktury objektu používané v OOP	28
Obrázek 2.1 - Logo společnosti BJSW CZ	41
Obrázek 2.2 - Přírůstek obyvatel ČR	43
Obrázek 2.3 - SWOT Analýza	55
Obrázek 3.1 - Mapa rizik	69
Obrázek 3.2 - Mapa rizik po aplikování opatření	72
Obrázek 3.3 - Pavučinový graf hodnot rizik před a po aplikaci opatření	73
Obrázek 3.4 - ReportForm a rozložení UI elementů v editoru	75
Obrázek 3.5 - Schéma zobrazující ReportForm spuštěné aplikace	76
Obrázek 3.6 - Okno výběru identifikátoru GMP	77
Obrázek 3.7 - Ukázka části kódu GMP	77
Obrázek 3.8 - Podpůrná aplikace SMDH Admin	78
Obrázek 3.9 - Ukázka kódu pro nastavování vyhledávání dodávek	79
Obrázek 3.10 - Část kódu vytvářející SQL dotaz pro vyhledávání	79
Obrázek 3.11 - MenuStrip v aplikaci SMDH Report	80
Obrázek 3.12 - Schéma Formu pro procházení a editaci dat v editoru	81
Obrázek 3.13 - Okno DataBrowseForm s vyhledanými daty	82
Obrázek 3.14 - Možnosti úprav filtrování dat v tabulce	83
Obrázek 3.15 - Možnost skrýt sloupce v tabulce – funkce výběr sloupců	84
Obrázek 3.16 - Práce s výčtem typů z pohledu UI	84
Obrázek 3.17 - Náhled několika možností filtrování dat	85
Obrázek 3.18 - Zobrazení aktivního filtru v tabulce	85
Obrázek 3.19 - Detail naměřeného kmene po výběru z tabulky	86
Obrázek 3.20 - Možnosti hromadné změny v aplikaci	87
Obrázek 3.21 - Vyexportovaná tabulka z procházení dat	88
Obrázek 3.22 - Ukázka kódu při dokončení prohledávání <i>BackgroundWorkeru</i>	89
Obrázek 3.23 - Procházení skládek v aplikaci SMDH Report	90
Obrázek 3.24 - Možnosti vytváření filtrů v tabulce	91

Obrázek 3.25 - Možnosti seskupení dat ve skládkách	92
Obrázek 3.26 - Ukázka části kódu načítající data pro skládku	93
Obrázek 3.27 - Přehled tiskových sestav a možností exportu dat	94
Obrázek 3.28 - Soubory reportů a exportérů	95
Obrázek 3.29 - Ukázka enumu exportérů v aplikaci	95
Obrázek 3.30 - Kód vytvářející reporty na základě typu exportéru	96
Obrázek 3.31 - Přiřazení konkrétního exportéru v DeliveryReportu	96
Obrázek 3.32 - Příklad exportu dat dodávky do formátu CSV	97
Obrázek 3.33 - Příklad využití rozhraní	97
Obrázek 3.34 - Část kódu vytvářející PDF dokument	98
Obrázek 3.35 - Kód vykreslující spodní část PDF reportu dodávky	98
Obrázek 3.36 - Zobrazení vygenerovaného PDF dokumentu v aplikaci	99
Obrázek 3.37 - Porovnání s deklaroványi hodnotami v reportu	100
Obrázek 3.38 - Rozpad dat do kategorií v reportu dodávky	100
Obrázek 3.39 - Report výkonu zaměstnanců	101
Obrázek 3.40 - Část inventury skládky v exportovaném PDF souboru	102
Obrázek 3.41 - Aplikace SMPD s možností exportovat data	104

SEZNAM TABULEK

Tabulka 3.1 - Identifikace rizik	65
Tabulka 3.2 - Ohodnocení míry pravděpodobnosti výskytu rizika	67
Tabulka 3.3 - Ohodnocení dopadu rizika	67
Tabulka 3.4 - Kvantifikace rizik	68
Tabulka 3.5 - Opatření k rizikům	70
Tabulka 3.6 - Kvantifikace rizik po opatřeních	71
Tabulka 3.7 - Přehled hlavních činností	106