



Zadání diplomové práce

Název:	Portál pro pojišťovací zprostředkovatele
Student:	Bc. Roman Soběslav
Vedoucí:	Ing. Petr Bodnár
Studijní program:	Informatika
Obor / specializace:	Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2023/2024

Pokyny pro vypracování

Cílem této práce je na základě poskytnutých user stories provést analýzu, návrh a implementaci portálu pro zprostředkovatele v oblasti aplikací pro sjednání pojištění.

Student se seznámí se zadáním formou user stories a fungováním pojišťoven v oblasti zprostředkovatelů pojištění. Na základě této analýzy provede návrh a implementaci prototypu portálového řešení pro poskytování služeb partnerům pojišťovny. Tento prototyp bude obsahovat rozhraní s přehledem klientů a jejich pojistných smluv a umožní sjednat novou pojistnou smlouvu pro vybraného klienta. Vzniklý kód otestuje a zdokumentuje.

Diplomová práce

PORTÁL PRO POJIŠŤOVACÍ ZPROSTŘEDKOVATELE

Bc. Roman Soběslav

Fakulta informačních technologií
Katedra softwarového inženýrství
Vedoucí: Ing. Petr Bodnár
15. února 2024

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2024 Bc. Roman Soběslav. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení, je nezbytný souhlas autora.

Odkaz na tuto práci: Soběslav Roman. *Portál pro pojišťovací zprostředkovatele*. Diplomová práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2024.

Obsah

Poděkování	vii
Prohlášení	viii
Abstrakt	ix
Seznam zkratek	x
Úvod	1
0.1 Cíl práce	1
1 Analýza	3
1.1 Definice pojmů	3
1.2 Sjednání pojištění vozidla na německém trhu	4
1.2.1 Druhy pojištění	4
1.2.2 Slevy na pojištění	5
1.2.3 Získávané informace pro výpočet pojištění	6
1.2.4 Obecné poznatky	8
1.2.5 Proces sjednání pojištění	8
1.3 Analýza požadavků	8
1.3.1 Funkční	8
1.3.2 Nefunkční	9
1.4 Analýza existujících řešení	9
1.5 Doménový model	10
1.5.1 Správa osob	10
1.5.2 Pojištění	10
1.5.3 Dokumenty	15
1.5.4 Lokace	15
1.5.5 Kalendář	17
1.5.6 Zprávy	17
1.5.7 Obchodní příležitost	17
2 Návrh	25
2.1 Návrh uživatelského rozhraní	25
2.2 Technologie	33
2.2.1 Liferay	33
2.2.2 Databáze	34
2.2.3 Framework pro tvorbu uživatelského rozhraní	34
2.3 Logický datový model	35
2.4 Databázový model	41
2.5 Architektura	41

3 Implementace	47
3.1 Vývojové prostředí	47
3.2 Verzování kódu	47
3.3 Docker	47
3.4 Vývoj portálu pomocí Liferay 7.4	49
3.4.1 Liferay workspace	49
3.4.2 Tvorba servisní vrstvy	49
3.4.3 Tvorba pravidelných úkolů	50
3.4.4 Tvorba prezenční vrstvy	50
4 Testování	51
5 Závěr	53
A Administrátorská příručka	55
Obsah příloh	59

Seznam obrázků

1.1	Ukázka kompletního BOM	11
1.2	BOM - Přehled osob a rolí	12
1.3	BOM - správa klientů	12
1.4	BOM - Pojistný plán	13
1.5	BOM - Kalkulace	14
1.6	BOM - Pojistná smlouva	14
1.7	BOM - Pojistný plán pro pojištění vozidla	15
1.8	BOM - Atributy kalkulace pro pojištění vozidla	16
1.9	BOM - Vazba kalkulace pro pojištění vozidla na okolní entity	16
1.10	BOM - Pojistná smlouva pro pojištění vozidla	17
1.11	BOM - Vozidlo	18
1.12	BOM - Automobil	19
1.13	BOM - Dokumenty	20
1.14	BOM - lokace	20
1.15	BOM - Kalendář	21
1.16	BOM - Textová komunikace	21
1.17	BOM - Atributy pro obchodní příležitost	22
1.18	BOM - výstupy obchodní příležitosti	23
2.1	WF - seznam smluv	26
2.2	WF - seznam klientů	26
2.3	WF - profil klienta	28
2.4	WF - aplikace pro sjednání pojištění vozidla - 1. krok	29
2.5	WF - aplikace pro sjednání pojištění vozidla - 2. krok	30
2.6	WF - aplikace pro sjednání pojištění vozidla - 3. krok	30
2.7	WF - aplikace pro sjednání pojištění vozidla - 5. krok - rekapitulace osoby	31
2.8	WF - aplikace pro sjednání pojištění vozidla - 5. krok - rekapitulace pojistného plánu	32
2.9	LDM - kompletní model	36
2.10	LDM - osoby	37
2.11	LDM - agent	37
2.12	LDM - společná část pojištění	38
2.13	LDM - pojištění vozidla	39
2.14	LDM - dokumenty	40
2.15	PDM - fyzický datový model	42
2.16	PDM - číselníky	43
2.17	SA - návrh architektury u zákazníka	44
2.18	SA - plánovaný provoz na demo prostředí	45

Seznam tabulek

1.1	Minimální krytí pro povinné ručení definované zákonem.[13]	4
1.2	Minimální krytí pro povinné ručení definované zákonem.[13]	5
1.3	Srovnání druhů pojištění	5

Seznam výpisů kódu

Nejprve bych chtěl poděkovat panu Michalovi Štěpánkovi za umožnění psaní této diplomové práce v jeho společnosti Cleverbee Solutions s.r.o. Dále bych chtěl poděkovat panu Ing. Petru Bodnárovi za skvělé rady a vedení této diplomové práce. Dále bych rád poděkoval také paní Ireně Skotálkové a panu Filipu Popperovi za cenné konzultace v oblasti pojišťovnictví a zadání byznys části práce. Nakonec bych chtěl také poděkovat svoji rodině a přátelům, kteří mne podporovali ve studiu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. Dále prohlašuji, že jsem s Českým vysokým učení technickým v Praze uzavřel dohodu, na jejímž základě se ČVUT vzdalo práva na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona. Tato skutečnost nemá vliv na ust. § 47b zákona č. 111/1998 Sb., o vysokých školách, ve znění pozdějších předpisů.

V Praze dne 15. února 2024

Abstrakt

Cílem této práce je analýza, návrh a implementace portálu pro zprostředkovatele. Nejprve byla provedena analýza fungování pojištění na německém trhu. Na základě této analýzy došlo k návrhu portálu pro zprostředkovatele, který jsem implementoval pomocí portálového řešení Liferay 7.4. Pro implementaci uživatelského rozhraní jsem použil framework Angular2.

Klíčová slova zprostředkovatelé, pojištění, auto pojištění, B2B, Liferay

Abstract

The aim of this work is the analysis, design, and implementation of a portal for intermediaries. Initially, an analysis of insurance operations in the German market was conducted. Based on this analysis, a portal for intermediaries was designed, which I implemented using the Liferay 7.4 portal solution. For the implementation of the user interface, I utilized the Angular2 framework.

Keywords insurance agent, insurance, car insurance, B2B, Liferay

Seznam zkratk

API	Application programming interface
BOM	Business object model (byznys objektový model)
DACH	Akronym pro Německo, Rakousko, Švýcarsko
DIČ	Daňové identifikační číslo
DOM	Document Object Model
ESB	Enterprise service bus
eVB	elektronische Versicherungsbestätigung (elektronické potvrzení pojištění)
HSN	Herstellerschlüsselnummer (číslo výrobce)
HTTP	Hypertext Transfer Protocol
LDM	Logical data model (logický datový model)
OCR	Optical Character Recognition (Optické rozpoznávání znaků)
OSVČ	Osoba samostatně výdělečně činná
PDF	Portable Document Format (přenosný formát dokumentů)
PDM	Physical data model (fyzický datový model)
RDBMS	Relational database management system (Relační databázový systém)
SA	System architecture (systémová architektura)
SQL	Structured Query Language
TSN	Typschlüsselnummer (číslo typu)
UI	User interface (uživatelské rozhraní)
VAT	Value added tax (daň z přidané hodnoty)
VIN	Vehicle identification number (Identifikační číslo vozidla)
WF	Wireframe (drátěný model)

Úvod

Společnost Cleverbee solutions s.r.o. se zabývá tvorbou informačních systémů ve finančním sektoru, převážně v oblasti pojišťovnictví. Při nástupu na pozici vývojáře mi bylo velmi sympatické téma digitalizace, které si společnost bere jako hlavní cíl. Toto téma je mi velmi blízké, jelikož jsem fanoušek informačních technologií a vidím v tomto směru budoucnost. Jsem zastáncem myšlenky, že mezi úřady a firmami mají putovat data, ne lidé.

Při jedné interní schůzce byla zmíněna analýza, kdy na německém trhu je stále mnoho pojišťoven, které stále operují čistě mimo digitální svět. Jedná se převážně o malé lokální pojišťovny, které si svým rozpočtem nemohou dovolit konkurovat velkým nadnárodním pojišťovnám. Aby zůstaly cenově konkurenceschopné, využívají stále jako primární kanál pro prodej osobní setkání. Následná kalkulace ceny pojištění probíhá ručně, například pomocí předem připravených Excelových šablon. Díky tomuto způsobu práce dochází k prodlevě mezi sepsáním klientových požadavků a následným získáním ocenění těchto požadavků. Pokud je třeba upravit jakékoliv parametry smlouvy, je potřeba znovu navštívit pobočku pojišťovny.

Další zátěž na personál tvoří i samotný proces hlášení škody, kdy klient je nucen hlásit i banální záležitost skrze telefonní linku. Pokud dojde například ke změně příjmení, nebo korespondenční adresy, klient je opět nucen navštívit pobočku, případně zaslat dopis.

Je potřeba si také říct, že nevýhody se netýkají jen komfortu klienta, ale i pojišťovny jako takové. Proces výpočtu pojistné smlouvy je dnes velmi náročný obecně, jelikož je třeba vhodně odhadnout rizikovitost klienta. Při špatném odhadu se snadno může stát, že je částka podhodnocená, a nereflakuje tak případné náklady při vzniku škodní události. Naproti tomu jde dnešní poptávka, kdy klienti chtějí ideálně co nejvíce personalizované nabídky, které pokrývají přesně jejich potřeby a zajistí jim nejnižší cenu na trhu. Na základě toho pojišťovny dnes při výpočtu částky sledují více než stovku parametrů. Pokud tedy malá pojišťovna nemá standardizovaný a automatizovaný proces pro výpočet pojistného, jednoduše se stane, že podcení rizikovitost klienta a následně náklady na likvidaci pojistné události převyší vyčleněnou částku. Pokud je naopak částka vyčíslena výrazně nad průměr, riskuje tak ztrátu klienta.

Abychom pomohli výše popsaným pojišťovnám, rozhodli jsme se v Cleverbee Solutions s.r.o. vytvořit komplexní řešení pod názvem Cleeo. Účelem tohoto balíčku mají být předem připravené části systémů pro pojišťovny, které mají snížit čas na vývoj finálního řešení. V této práci se konkrétně zabývám portálem pro zprostředkovatele.

0.1 Cíl práce

V této práci se zabývám analýzou, návrhem a implementací portálu pro zprostředkovatele. Hlavním cílem je vytvořit obecný prototyp, který bude sloužit jako základ pro tvorbu finálních řešení pro konkrétní pojišťovny. Zabývám se návrhem a implementací pouze front-endové části. Tvorba core-systému (systém obsahující byznys logiku pro fungování pojišťovny) je mimo rozsah

této práce vzhledem ke své komplexnosti. Volání tohoto systému je tak pouze simulované, stejně jako volání ostatních systémů.

Cílem Kapitoly 1 je definovat pojmy, které se budou používat v rámci této práce, popsat princip pojištění vozidel na německém trhu, definovat případy užití. Posledním cílem je návrh byznys objektového modelu dle dřívějších zjištění.

Kapitola 2 se zabývá návrhem konkrétního řešení. Konkrétně se jedná o návrh uživatelského rozhraní, volbu technologií, vytvoření datového modelu a návrh architektury.

Cílem Kapitoly 3 je popsat postup vývoje samotného portálu - použité nástroje a postup implementace dílčích částí.

Poslední kapitola 4 se věnuje testování.

Kapitola 1

Analýza

Tato kapitola staví teoretické základy, ze kterých vychází zbylá práce. Nejprve definuji pojmy, které se budou vyskytovat. Následně popíši princip sjednání pojištění vozidla na německém trhu. Definuji požadavky a případy užití portálu. Dále popíši výzkum již existujících řešení. Nakonec navrhnu a popíši byznys objektový model portálu dle předchozího výzkumu.

1.1 Definice pojmů

Před tím, než se začnu zabývat analýzou požadavků a existujících řešení, je třeba si definovat pojmy, které se budou vyskytovat v této práci. Důvodem není pouze seznámení čtenáře s problematikou, ale i upřesnění významu pojmu, ve kterém se bude vyskytovat. Některé pojmy si lze vyložit více způsoby, které se mohou lišit v závislosti na pojišťovně. Následující seznam obsahuje pojmy v češtině a angličtině spolu s definicí pojmu.

Fyzická osoba - *Natural person* - Představuje fyzickou osobu.

Jmenovaný pojištěný - *Named insured* - Osoba uvedená na smlouvě, která má stejná práva jako pojistník. Může tedy rozhodovat o pojistné smlouvě.[1]

Kalkulace - *Insurance quote* - Návrh pojistné smlouvy, která nemusí být ohodnocená. Vzniká postupně vyplňováním předem definovaných parametrů dle přání pojistníka.[2]

Osoba samostatně výdělečně činná - *Self-employed* - Fyzická osoba, která má příjmy z podnikání.

Podnikatel - *Entrepreneur* - Fyzická nebo právnická osoba, která vykonává soustavnou činnost pro vytváření zisku.

Pojistitel - *Insurer* - právnická osoba, která je oprávněna provozovat pojišťovací činnost, tj. pojišťovna.[3]

Pojistná smlouva - *Insurance policy* - Smlouva mezi pojistitelem a pojistníkem o uzavření pojištění. Pojistitel se zavazuje k vyplacení smluvené částky při vzniku pojistné události, na kterou se vztahuje pojistná smlouva.[3] Jedná se o zafixovanou kalkulaci.

Pojistné - *Insurance premium* - Částka uvedená na smlouvě, kterou musí uhradit pojistník pojistiteli.

Pojistník - *Policyholder* - Osoba, která uzavírá pojistnou smlouvu s pojistitelem. Zavazuje se platit pojistné.[3]

Pojistný plán - *Insurance plan* - Konkrétní ohodnocení pojistného produktu. Definuje seznam pojistných událostí, na které se vztahuje pojistná smlouva a částky, které jsou vypláceny.

Pojištěný - *Additional insured* - Osoba, na kterou se vztahuje pojištění definované pojistnou smlouvou.[3]

Portál pro zprostředkovatele - *Insurance agent portal* - Portál, který využívají externí pracovníci pojišťovny. Typicky se jedná o OSVČ nebo firmy, které se zaměřují na poradenství s následným zprostředkováním pojištění. Agenti jej využívají ke správě svých klientů u dané pojišťovny, spolu s jejich pojistnými smlouvami.

Právnícká osoba - *Juridical person* - Právní společnost (firma).

Zprostředkovatel - *Insurance agent* - Osoba, která je oprávněna uzavírat jménem pojistné smlouvy jménem pojišťovny.

1.2 Sjednání pojištění vozidla na německém trhu

Pro výzkum pojištění na německém trhu jsem využil webových stránek, které pomáhají cizincům se zorientovat v německém prostředí[4, 5], pokud přemýšlí o přestěhování se. Dále jsem se zaměřil na následující pojišťovny, ze kterých jsem čerpal pro výzkum této sekce. Jedná se o: Allianz[6], DA Direkt[7], HDI[8], Helvetia[9], AXA[10], VERIVOX (srovnávací portál)[11] a Tarifcheck(srovnávací portál)[12].

1.2.1 Druhy pojištění

V Německu, stejně jako ve většině státech světa, je požadováno po majiteli vozidla uzavřené *pojištění odpovědnosti z provozu vozidla*, hovorově známé pod pojmem *povinné ručení*. [13] Tento typ pojištění je aplikován na kompenzaci za léčbu, škodu na majetku a ušlém zisku v případě způsobení dopravní nehody řidičem vozidla. Pojištění je svázané s vozidlem, není tedy nutné aby si každý řidič uzavíral pojištění sám. Existují však případy, kdy pojištění nelze aplikovat, přestože vozidlo je pojištěné. Typicky se jedná o pronajatá/zapůjčená vozidla, případně zapůjčení vozidla osobě mimo domácnost, která není uvedena na smlouvě. Pro tento případ si musí řidič pojistit vozidlo sám.

Uzavřené *povinné ručení* je jedním z požadavků způsobilosti vozidla pro provoz na silnici. V případě, že řidič řídí vozidlo bez platného povinného ručení nebo bez registračních značek, dopouští se porušení zákona. Veškerá škoda, nehlédě na viníka nehody, je vymáhána po řidiči, který řídil toto vozidlo. Tabulka 1.1 definuje minimální částky stanovené zákonem, které musí být kryt každé nabízené povinné ručení na německém trhu. Všechny zkoumané pojišťovny, alespoň v jedné úrovni pojištění, nabízejí shodně částky definované v tabulce 1.2.

Typ škody	minimální krytá částka
Újma na zdraví	7,5 milionu EUR
Poškození na majetku	1,12 milionu EUR
Ušlý zisk	50 000 EUR

■ **Tabulka 1.1** Minimální krytí pro povinné ručení definované zákonem.[13]

Při sjednání povinného ručení na nové, nebo dohlášené vozidlo, je pojišťovna povinna vydat a zaslat pojistníkovi do dvaceti čtyř hodin od sjednání *eVB*. Jedná se o kód pro elektronické ověření platnosti povinného ručení pro daný vůz. Toto potvrzení je vyžadováno úřadem při přihlášení vozidla do registru vozidel.[14]

Typ škody	nejčastěji nabízená částka
Újma na zdraví	15 milionu EUR
Poškození na majetku	100 milionu EUR

■ **Tabulka 1.2** Minimální krytí pro povinné ručení definované zákonem.[13]

Kromě povinného ručení pojišťovny nabízí dobrovolné soukromé pojištění, které je v Čechách známé pod pojmem havarijní pojištění. Obecně se označuje též jako *kasko*. Na německém trhu si lze pořídit jednu ze dvou úrovní – komplexní (anglicky *fully comprehensive*, německy *Teilkasko*) a částečné (*partially comprehensive/Vollkasko*). Povinné ručení se vztahuje pouze škody způsobené ostatním účastníkům provozu, zatímco havarijní pojištění se snaží krýt škody, které vzniknou na vlastním vozidle. Částečné havarijní pojištění kryje škody na vozidle, které vzniknou bez zavinění pojištěnou osobou. Jedná se o škody vzniklé živelnými pohromami, požárem a krádeží. Výjimkou je vandalismus, který je až součástí komplexního pojištění. Součástí částečného havarijního pojištění je standardně i ochrana světel a skel, střet se zvěří a poškození vlastního majetku z nedbalosti. Komplexní havarijní pojištění chrání vozidlo proti výše uvedeným škodám. Kromě toho také kryje škody vzniklé na vlastním vozidle při zavinění nehody. Zajímavé je porovnání se Švýcarskem, kde se nabízí stejné typy pojištění. Na rozdíl od německého trhu částečné havarijní pojištění kryje škody vzniklé na vlastním vozidle v případě zavinění nehody. Pro ochranu vozidla před živelnými katastrofami je třeba sjednat komplexní havarijní pojištění.[15] Tabulka 1.3 porovnává standardní typy škod, které jsou kryty různými typy pojištění vozidel.

Typ škody	Povinné ručení	Částečné	Komplexní
Škoda na cizím majetku	ANO	ANO	ANO
Náklady na léčbu	ANO	ANO	ANO
Krádež vozidla		ANO	ANO
Požár a živelné pohromy		ANO	ANO
Kolize se zvěří		ANO	ANO
Pojištění skel		ANO	ANO
Výměna zámků při ztrátě klíčů		Volitelné	Volitelné
Způsobení nehody z nedbalosti		Volitelné	Volitelné
Poškození vlastního vozidla při nehodě			ANO
Vandalismus			ANO
*Hit-and-run			ANO

*Hit-and-run je typ poškození vozidla, kdy pachatel ujede z místa nehody. Vztahuje se primárně na zaparkované vozidlo, kdy je pachatel neznámý.

■ **Tabulka 1.3** Srovnání druhů pojištění

1.2.2 Slevy na pojištění

První slevou, kterou bych chtěl probrat, je sleva na typ registračních značek. Německo dovoluje registrovat si značky pouze na část roku. Jedná se o sezonní registrační značky. Podmínkou je provoz alespoň dva souvislé měsíce, nejvýše však jedenáct měsíců. Využívají se například na motocyklech, nebo pokud někdo vlastní letní a zimní automobil. Zákon umožňuje platit pojištění pouze za měsíce, kdy je vozidlo v provozu.[16]

Další slevu může klient dostat v případě, že si chce pojistit další vozidlo. Předpokladem je, že druhé vozidlo je například víkendové a tak se nebude vyskytovat tolik v rizikovém provozu.[4]

Stejně jako mnoho zemí na světě, i Německo používá systém zvaný *bonus-malus*, který odměňuje pojistníky za každý rok bez škodní události. Naopak, v případě hlášení pojistných událostí je pojistník penalizován, kdy se místo padesáti procentní slevy dostane až na stoprocentní příplatek za pojištění. Německý ekvivalent se nazývá SF-třída (*Schadenfreiheitsklasse*). Číslo za pomlčkou určuje počet bezeškodních let. Maximální slevu, kterou lze přes SF-třídou získat je sedmdesát procent. Každá pojišťovna si definuje vlastní tabulku, která definuje degradační třídy při nahlášení škodní události. Většinou se jedná o dvojrozměrnou tabulku, kdy každá další škodní událost ve stejném roce znamená větší postih. Například má-li pojistník třídu SF-8, je po nahlášení škodní události degradován na SF-4. Pokud již předtím hlásil škodní událost, je degradován na SF-1 místo toho. Speciálními třídami jsou SF-1/2 a SF-M. SF-1/2 je přidělen pojistníkovi, který si pojistil své první vozidlo, avšak je držitelem řidičského oprávnění alespoň tři roky. Díky této třídě dostane slevu na pojištění, která je nižší než v případě SF-1, ale pokud nenahlásí škodní událost během roku, je povýšen na SF-2. SF-0 je naopak přiřazena řidiči, který nemá řidičské oprávnění alespoň tři roky. Třída SF-M naopak označuje osobu, která je příliš riziková. Pojistník s touto třídou standardně platí 100 procentní příplatek. Při přechodu mezi pojišťovnami se třídy přenáší. Rozlišuje se zvláště třída pro povinné ručení a pro komplexní pojištění.[4, 5]

Pojistník dostane slevu na pojistné také za spoluúčast. Jedná se o částku, kterou klient hradí v případě škodní události. Pokud si zvolí spoluúčast 150 euro, pojišťovna vyplácí částku ve vyšší cena za opravu mínus spoluúčast. Pojistník si volí zvláště spoluúčast pro částečné havarijní pojištění a zvláště pro komplexní.[4, 5]

Pokud vozidlo řídí více jak jedna osoba, obecně dochází ke zdražení pojištění. Výjimkou jsou děti a partneři, kteří jsou starší dvaceti pěti let a vlastní řidičské oprávnění alespoň tři roky. Ke zdražení také dochází v případě, že jeden z řidičů je starší šedesáti pěti let. Pokud pojistník chce, aby se pojistná smlouva vztahovala i na osobu mimo domácnost, musí ji buď jmenovat na smlouvě a vlastnit vlastní technický průkaz k vozidlu, nebo musí zvolit na smlouvě možnost, kdy vozidlo může používat kdokoliv. Standardně má tato volba za následek zdražení smlouvy o sto procent, kvůli vyššímu riziku.[4, 5]

Roční nájezd, parkování a způsob užívání vozidla jsou další z parametrů, které mají vliv na výslednou cenu pojistného. Čím více se vozidlo vyskytuje v provozu, tím je rizikovější.[4, 5]

Volba způsobu financování vozidla má na pojištění zásadní vliv. V případě, že je vozidlo pořízeno na leasing, pojišťovny automaticky nabízí zájemci o pojištění pouze komplexní pojištění. Vychází to z praxe, kdy poskytovatelé leasingů požadují ve smlouvách automaticky maximální možné pojištění vozidla pro minimalizaci rizika finanční ztráty.

Pokud chce pojistník dosáhnout dalších slev, může si zvolit možnost oprav v partnerských autoservisech. V případě nehody je pojistník nucen provést opravu v jednom z partnerským servisů. Tato volba se nemusí vyplatit pojistníkovi v případě, že se žádný partnerský autoservis nenachází v jeho okolí, případně s horší dopravní obslužností.[4, 5]

Posledním způsobem, jak lze získat slevu na pojistném je zaplatit celou roční sumu nejednou. Pojišťovny preferují zaplacení ročního pojistného předem v celé své výši.[4, 5]

1.2.3 Získávané informace pro výpočet pojištění

Na základě analýzy kalkulaček pro auto pojištění výše vyjmenovaných pojišťoven a srovnávacích portálů, jsem sestavil následující seznamy atributů, které pojišťovny používají při výpočtu pojistného pro fyzické osoby.

Osoby:

Jméno a příjmení

Adresa PSČ je jeden z prvních požadovaných údajů. Využívá se pro ocenění rizikovosti klienta na základě místa, kde se vozidlo primárně pohybuje. Například osoba bydlící v Berlíně zaplatí

na pojistném více, než identická osoba s identickým vozidlem na vesnici. Celá adresa je následně využívána pro korespondenční účely.

Datum narození

Rok získání řidičského průkazu Rozhoduje v případě, že si pojistník chce sjednat své první pojištění. Pokud jej vlastní alespoň tři roky, přidělí se mu třída SF-1/2, která nabízí mírně nižší pojistné než SF-0.

Trestné body Pokud pojistník páchá dopravní přestupky, jedná se o rizikovějšího klienta.

Profese Určité profese ze zákona mají nárok na slevu (např. úředníci po pěti letech služby[17])

Vozidlo:

Druh registrační značky Standardní nebo historické. V případě že se jedná o sezonní značky, je třeba uvést počáteční a koncový měsíc.

Sekundární vozidlo Informace, že se jedná o méně používané vozidlo a lze tedy aplikovat slevu.

Identifikátor modelu vozidla Kombinace HSN/TSN nebo ruční vyhledání.

Existující pojištění vozidla Uvádí se z důvodu přenosu bonusů mezi pojistnými smlouvami.

Datum první registrace Slouží při výpočtu hodnoty vozidla.

Majitel vozidla Vynucení vyplnění majitele vozidla pokud jej pojistník nevlastní. (např. firmní vozidlo, leasing atd.)

Užití vozidla:

Způsob Pro soukromé nebo pracovní účely

Roční nájezd Čím méně vozidlo jezdí po komunikacích za rok, tím více pojistník ušetří. Uvedený nájezd se kontroluje (např. na stanici technické kontroly). Při způsobení nehody po překročení limitu může pojišťovna požadovat po pojistníkovi kompenzaci za způsobenou škodu. Důvodem je porušení smluvních podmínek. Aby se pojistník vyhnul pokutě a měl stále platné pojištění, je třeba nahlásit změnu ročního nájezdu včas.

Způsob financování Vlastní finance nebo splacená půjčka, leasing, bankovní půjčka. Při volbě leasingu vynutí sjednání komplexního pojištění.

Počet řidičů a vztah vůči pojistníkovi Volba manžela/manželky standardně nemá vliv na cenu pojistného. Zvolením možnosti *ostatní* dochází k navýšení pojistného na dvojnásobek.

Způsob parkování

Škodní historie v případě existence předchozího pojištění:

Jméno pojišťovny

Počet hlášení škodních událostí v posledních třech letech

Dosažené SF třídy Povinné ručení je přenášeno vždy. Pro přenos třídy komplexního pojištění je nutné jej mít sjednané v posledních dvanácti měsících.

Parametry pojištění:

Počátek platnosti smlouvy

Varianta pojištění

Úroveň pojistného plánu

Výše spoluúčasti

Frekvence platby Měsíční, čtvrtletní, pololetní, roční. Při zaplacení celého pojistného předem získá pojistník slevu.

1.2.4 Obecné poznatky

Pro identifikaci osob se používá kombinace jména, příjmení a data narození. Německo nepoužívá rodná čísla nebo případný ekvivalent (národní identifikační číslo), který jednoznačně identifikuje osobu.

Posledním poznatkem, který mne překvapil, je téměř absence příplatkových balíčků. Na rozdíl od švýcarského a českého trhu se příplatkové služby vyskytují minimálně. Jedná se především o prémiové asistenční služby a právní ochrana, která hradí náklady na soudní řízení, kdy je nehoda řešena soudní cestou. Pojišťovny tak nabízí většinou dvě až čtyři předem připravené varianty pojištění, ze kterých si klient vybírá.

1.2.5 Proces sjednání pojištění

Proces sjednání pojištění začíná buď sjednáním pojištění on-line pomocí zjednodušené kalkulačky nebo osobní návštěvou pobočky pojišťovny či zprostředkovatele, kteří jsou vzdělaní v dané problematice. Díky tomu mají přístup k rozšířené kalkulačce, která dovoluje komplexněji a podrobněji nastavovat parametry pojištění. Potenciální pojistník vyplní/sdělí své požadavky na pojištění a doplní údaje o sobě, případně dalších osobách, a předmětu pojištění (vozidla pro auto-pojištění).

Po vyplnění parametrů dojde k ocenění kalkulace. Pokud je zájemce o pojištění spokojen s nabídkou, rozhodne se ji podepsat - ruční podpis fyzicky na pobočce, uhrazením pojistné při sjednání on-line. Od této chvíle je smlouva sice platná, ale rozjíždí se proces ověření údajů. Při nesrovnalosti dochází k vypovězení smlouvy, případně je pojistník vyzván k doplnění a opravě chybějících/chybných údajů.

1.3 Analýza požadavků

Tato kapitola definuje požadavky na výsledné řešení formou seznamů. Položky jsou řazeny od nejdůležitější po nejméně důležité.

1.3.1 Funkční

Parametry pojištění:

Zobrazení profilu agenta Agent si může prohlédnout informace o svém profilu, které si o něm pojišťovna uchovává.

Aktualizace profilu agenta Agent může aktualizovat informace, pokud objeví nesrovnalosti.

Správa svých klientů Agent může spravovat své portfolio. Může si zobrazit klienty a jejich profily, zakládat pojistné smlouvy a zobrazovat si detaily pojistných smluv, které sám založil.

Podpora fyzických a právnických osob a OSVČ Aplikace musí umět založit jakýkoliv typ osoby, nejen fyzické osoby.

Podpora dokumentů Aplikace musí umožnit nahrání osobních dokumentů klienta a fotodokumentaci pojištěného majetku.

Notifikace Portál podporuje notifikace o změně stavů smluv, novinky o pojistných produktech a upozornění na zprávy a události.

Práce s kalendářem Portál umožňuje vytvářet schůzky a evidovat si úkoly.

Podpora komunikace Portál umožní komunikaci mezi agentem a klientem skrze portál.

Evidence obchodních příležitostí Portál bude evidovat klientovu žádost o pojištění jako obchodní příležitost.

1.3.2 Nefunkční

Webová aplikace Portál bude implementován formou webové aplikace. Odpadá tak nutnost instalace aplikace na koncové zařízení uživatele. Zároveň je aplikace přístupná odkudkoliv. Také je zajištěna široká kompatibilita s koncovými zařízeními, jelikož jediným požadavkem je zařízení s moderním webovým prohlížečem (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge, Opera). Kombinace těchto faktorů je klíčová pro tento projekt, jelikož uživatelé této aplikace nejsou interní zaměstnanci, ale partnerské 3. strany – [glossary:OSVC]OSVČ nebo firmy a jejich zaměstnanci, kteří se specializují na poradenskou a sjeďnávací činnost v oblasti pojišťovnictví.

Responzivní design Přestože se nepředpokládá, že by uživatel přistupoval k portálu z mobilního zařízení, je potřeba zajistit škálování GUI aplikace dle velikosti displaye zařízení. Uživatelé budou přistupovat k aplikaci převážně z přenosných zařízení s úhlopříčkou 7 až 17 palců.

Kompatibilita s Liferay 7.4 Pro tvorbu portálu musí být použita technologie Liferay ve verzi 7.4. Důvodem je kompatibilita s dalšími portály a portlety, které budou v rámci komplexního řešení, kterého je tento portál součástí.

Přístup k aplikaci Přístup do aplikace mají pouze oprávnění uživatelé, kteří mají zajištěné přístupy pojišťovnou. Uživatel bez přístupových údajů se nesmí dostat do portálu, ani získat jakákoliv data z vystaveného API pro potřeby portálu.

Jazyková podpora Portál musí podporovat možnost přepínání jazyků, ať již uživatelsky nebo pomocí konfigurace. Přestože je aplikace cíl na německý trh, počítá se s použitím na celém DACH trhu v upravené variantě pro daný trh.

Podpora více měn Portál musí umožnit změnu zobrazované měny pomocí konfigurace. Počítá se pouze s vizuální změnou, jelikož zobrazované ceny jsou čistě vizuálního charakteru a jsou tak závislé na výchozí měně, ve které fungují ostatní systémy.

1.4 Analýza existujících řešení

Při hledání existujících řešení jsem zjistil, že neexistuje přímý konkurent vytvářené aplikace. Čistě frontové řešení nenabízí na trhu nikdo, většinou je součástí kompletního systému včetně backandové části dodávající byznys logiku.

Dalším problémem je, že většina nalezených řešení není veřejně dostupná. Vývojáři těchto aplikací je sice nabízejí, ale prezentace a dokumenty jsou součástí osobního jednání a nejsou veřejně k dispozici.

1.5 Doménový model

Na základě kapitol výše jsem vytvořil byznys doménový model v nástroji Enterprise Architect. Ten lze rozdělit do osmi částí - lokace, správa osob, pojištění, dokumenty, kalendář, zprávy, notifikace a obchodní příležitost. Celý BOM si lze prohlédnout na obrázku 1.1, případně v příloze pod názvem *BOM-00 - General overview.png*. Následující popis jednotlivých částí nebude v pořadí, ve kterém byly vydefinovány, ale dle významnosti.

1.5.1 Správa osob

První část, kterou bych chtěl popsat, je správa osob. Do této části jsem spojil správu klientů a agentů, jelikož se principiálně jedná pouze o různé role pro stejné objekty, které definují reálné osoby. Kompletní návrh této části zachycuje diagram na obrázku 2.10.

Hlavním objektem je *Entita*, která představuje abstrakci nad osobou. Ta má dva potomky – *Natural_person* a *Entrepreneur*. *Natural_person* představuje fyzickou osobu, jejíž atributy jsou následující – jméno, příjmení, datum narození, pohlaví, národnost, vysokoškolský titul, kontaktní informace, kontaktní a trvalá adresa.

Entrepreneur je sdružující objekt pro své potomky *Juridical_person* a *Self-employed* s jediným společným atributem VAT. V Němemcku neslouží VAT jen pro mezinárodní identifikaci obchodníků, kteří jsou plátcí daně, ale i k vnitrostátní identifikaci pro právnické osoby a OSVČ.[18] Jedná se o ekvivalent DIČ, které používáme zde v České republice. Důvodem je mít jednoznačný identifikátor, kterým lze jednoznačně odlišit osoby.

Juridical_person je objekt, který reprezentuje právnické osoby, tedy firmy. U nich je evidován název společnosti, typ společnosti (společnost s ručením omezeným, akciová společnost a jiné) a předmět hlavního zájmu společnosti.

Self-employed objekt je vazebním objektem mezi *Entrepreneur* a *Juridical_person*, jelikož OSVČ je fyzická osoba, která má vlastní živnost, a tedy má vlastní identifikační číslo.

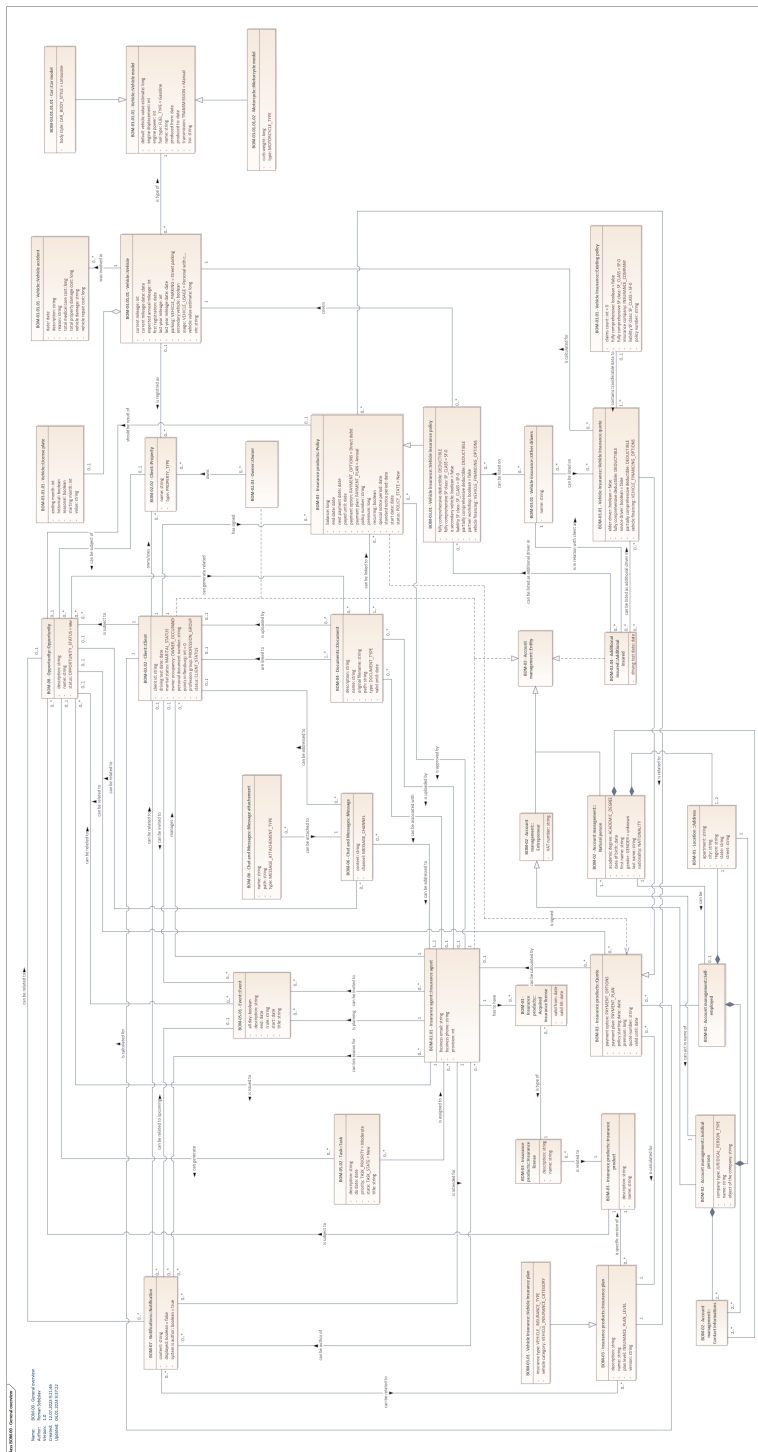
Takto definované objekty dovolují pokrýt všechny aktéry, které vystupují v různých rolích v rámci sjednávání pojištění. Tyto role jsou následující – agent, klient, majitel, další pojištěná osoba a jmenovaný pojištěný. V kontextu BOM je klient brán jako osoba, která uzavírá pojistnou smlouvu s agentem/pojišťovnou. Tento kontext se následně při návrhu finálního datového modelu změnil na výše definovaný. Majitel je na smlouvě uveden v případě, že pojistník(klient) není majitelem pojištěného majetku. Tato skutečnost je důležitá hlavně při vzniku škodní události, kdy kompenzace je adresována majiteli, kterému vznikla reálná škoda, nikoliv pojistníkovi. Speciálními rolemi jsou *další pojištěná osoba* a *jmenovaný pojištěný*. V obou případech se jedná o další osoby, na které se vztahuje uzavřená pojistná smlouva. Rozdíl je však ten, že *jmenovaný pojištěný* má navíc oprávnění měnit parametry smlouvy, stejně jako pojistník/klient.

Zvláštní roli tvoří *agent*, který označuje osobu na smlouvě, která uzavírá pojistnou smlouvu s klientem, jménem pojišťovny. Dále dovoluje dané osobě se přihlásit do systému a spravovat své portfolio klientů u dané pojišťovny. Správa portfolia zahrnuje vytváření svých klientů (vazba *Client-Agent*), aktualizace informací o klientovi, zakládání a správa kalkulací pro své klienty a nakonec uzavírání pojistných smluv jménem pojišťovny se svými klienty. Tyto vztahy zachycuje diagram na obrázku 1.3.

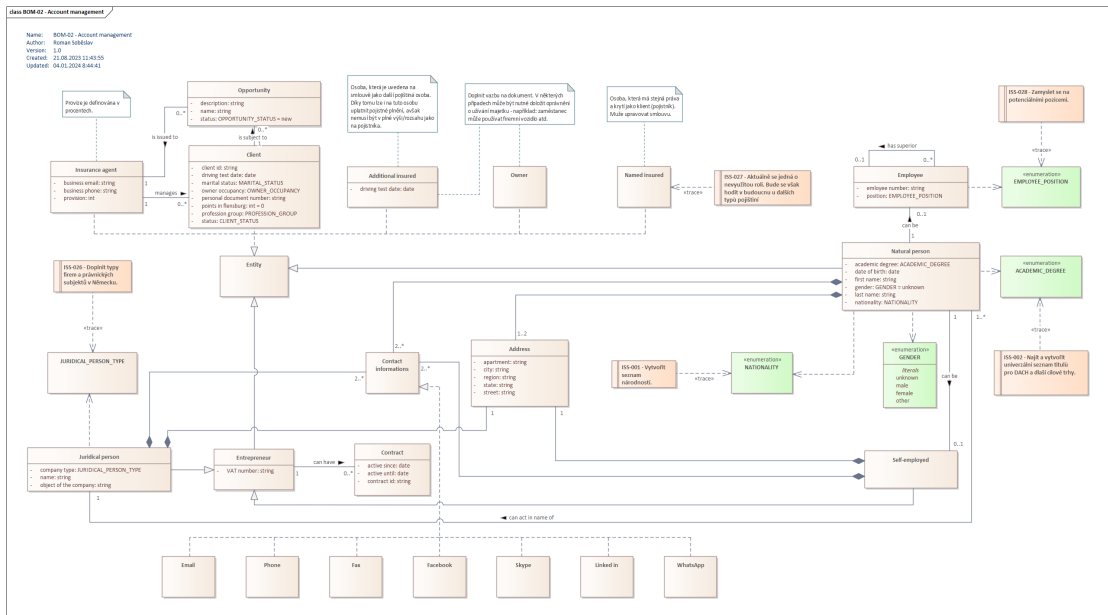
1.5.2 Pojištění

Část pojištění se zabývá analýzou datového modelu pro pojistné produkty, pojistné smlouvy a kalkulace. Výsledný model lze rozdělit na dvě části – společnou pro všechny typy pojištění a specifickou pro pojištění vozidel.

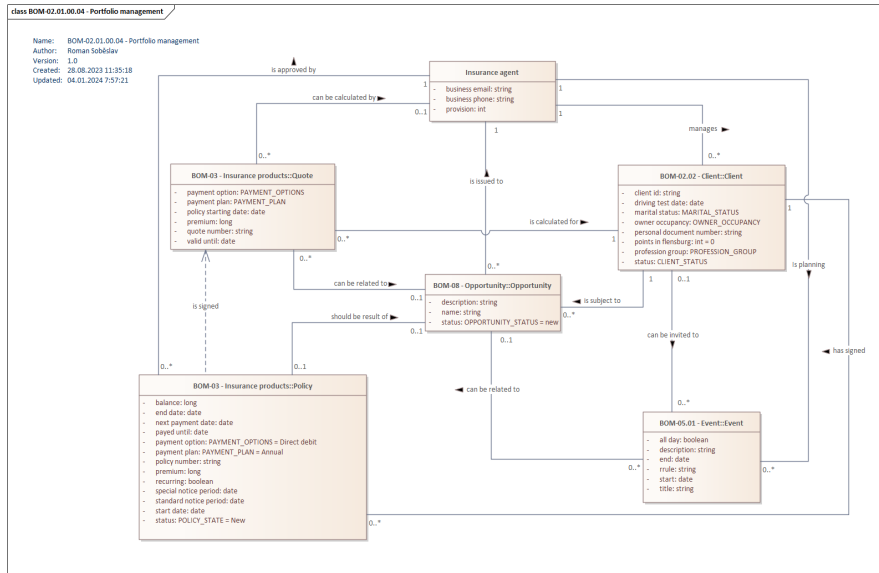
Předtím, než mohu popsat pojistnou smlouvu, je třeba si definovat pojistný produkt, který si klient může sjednat. Nejvyšším objektem v tomto směru je *Insurance_product*, který reprezentuje



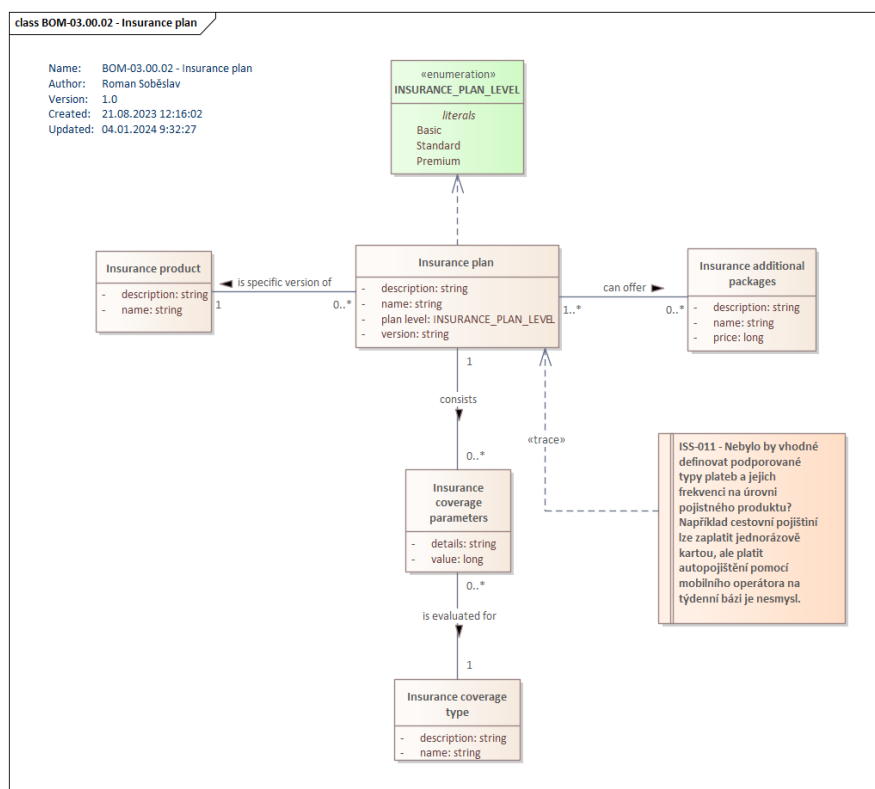
■ Obrázek 1.1 Ukázka kompletního BOM



■ Obrázek 1.2 BOM - Přehled osob a rolí



■ Obrázek 1.3 BOM - správa klientů



■ Obrázek 1.4 BOM - Pojistný plán

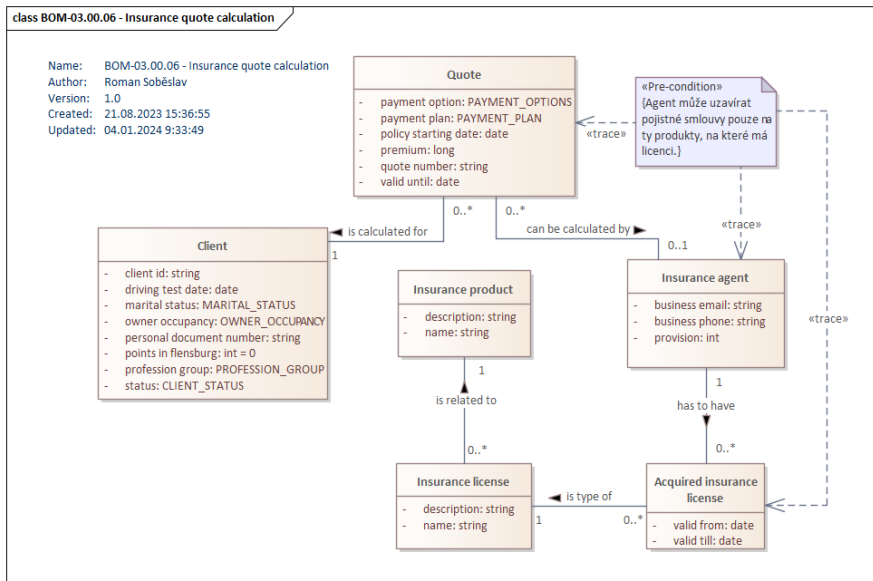
pojistné produkty (cestovní pojištění, pojištění vozidel a jiné). S produktem souvisí i objekt *Insurance_license*, která definuje oprávnění pro agenty. Stejně jako je tomu v České republice, i v Německu je vyžadováno získání licence od státu pro sjednání konkrétního typu pojištění. Tato licence se získává složením znalostní zkoušky specifické pro daný pojistný produkt.

Insurance_plan reprezentuje již konkrétní nabídku produktu. Například pro pojištění vozidla v Německu existují tři druhy (povinné ručení, částečné havarijní pojištění a komplexní havarijní pojištění). Každý z těchto druhů má většinou ještě různé varianty - základní a prémiová úroveň. V tomto případě existuje šest různých instancí třídy *Insurance_plan*, které reprezentují vyjmenované plány. Každý plán obsahuje seznam krytých škod, spolu s případnou maximální částkou pro daný typ škod a případně omezením, kdy lze pojištění uplatnit. Pro volitelné příplatkové služby je určen objekt *Insurance_additional_package*. Tato skutečnost je zachycena na obrázku 1.4.

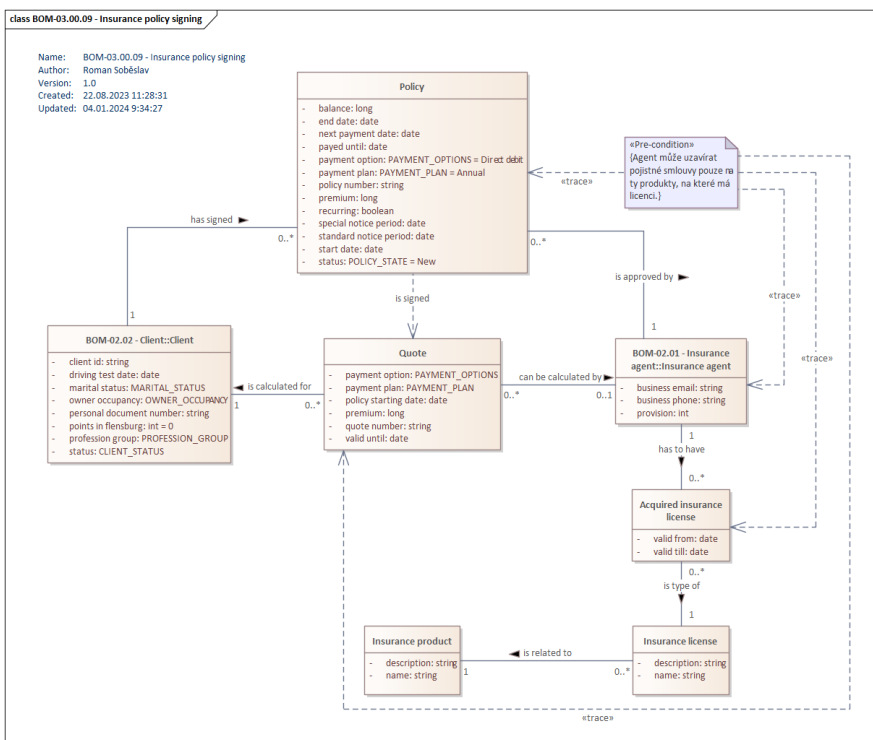
Pro výpočet pojistné kalkulace je určen objekt *Quote*, který reprezentuje abstraktního předka všech návazných kalkulací (pojištění vozidla, cestovní pojištění a další). Společnými atributy jsou zvolený pojistný produkt, zvolený plán, příplatkové služby, způsob platby a její frekvence a datum začátku platnosti. Tuto skutečnost zachycuje obrázek 1.5.

Uzavřená pojistná smlouva je reprezentována objektem *Policy*, která je principiálně podobná kalkulaci. Pojistnou smlouvou lze nazvat zafixovanou kalkulaci, kterou podepíše obě strany. Oproti kalkulaci je rozšířena o následující atributy – zůstatek (pozitivní v případě přeplatku, záporný v případě nedoplatku), datum následující platby, datum zaplacení, výpovědní lhůta a aktuální stav smlouvy. Konkrétní návrh je vyobrazen na obrázku 1.6.

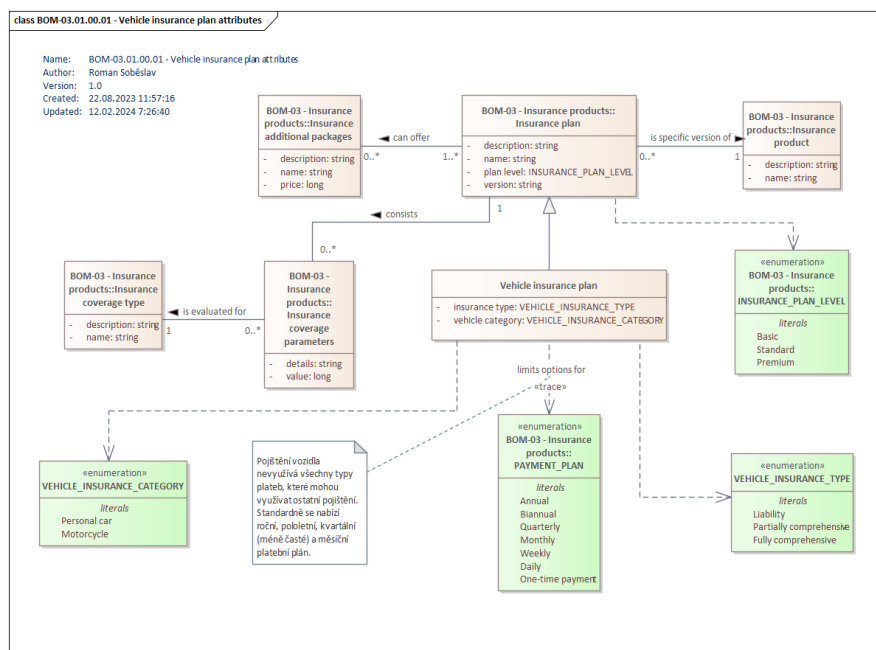
Obrázky 1.7, 1.8, 1.10 a 1.9 zachycují upravené objekty, které jsou specifické pro pojištění vozidel. Oproti generické variantě je třeba přidat informace o řidiči (rok získání řidičského oprávnění, body v bodovém systému, bonus-malus) a vazbu na pojištěné vozidlo. Specifikou kalkulace pro pojištění vozidla je entita, která slouží pro přenos informací při přechodu mezi pojišťovnami.



■ Obrázek 1.5 BOM - Kalkulace



■ Obrázek 1.6 BOM - Pojistná smlouva



■ Obrázek 1.7 BOM - Pojistný plán pro pojištění vozidla

Jedná se o název původní pojišťovny, číslo původní pojistné smlouvy, dosažený bonus-malus a počet pojistných událostí během posledních pěti let.

Pro reprezentaci vozidla jsem zvolil jedinou entitu jménem *Vehicle*. Jedná se o konkrétní vozidlo, které je předmětem pojistné smlouvy. Hlavními atributy jsou: VIN (jednoznačný identifikátor vozidla), odhadovaný roční nájezd, datum první registrace, způsob parkování po většinu času, způsob užívání a případné registrační značky (nikdy nepřihlášené, nebo odhlášené vozidlo nemá registrační značky). Více informací lze vidět na obrázku 1.11.

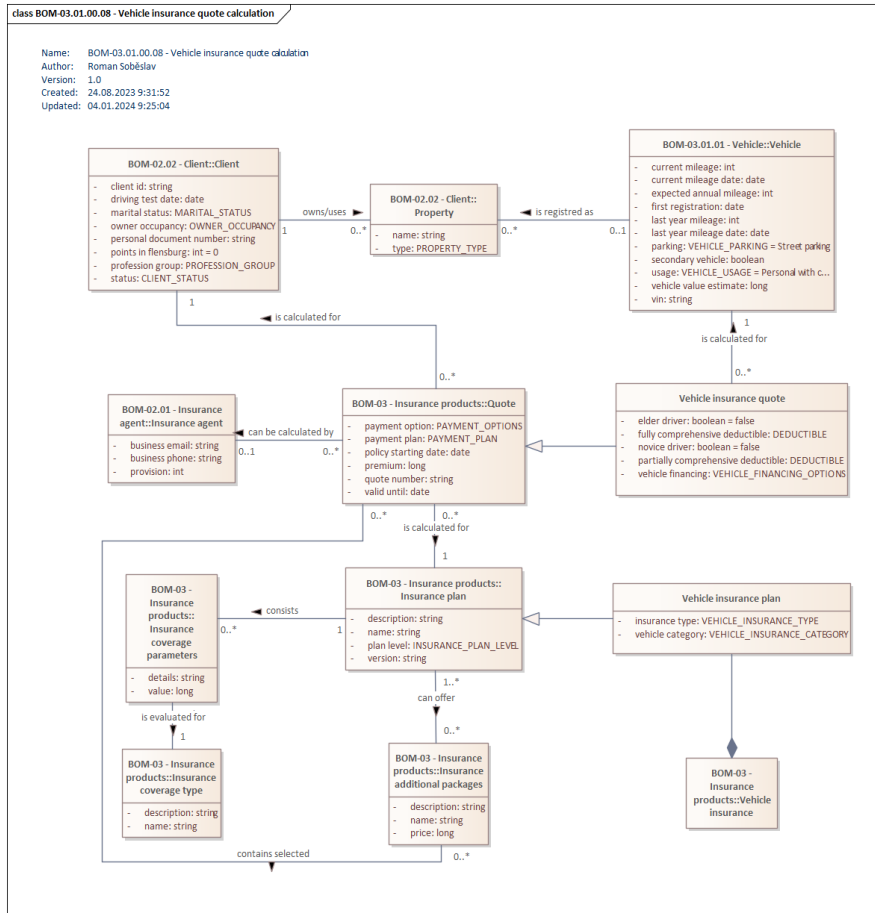
Pro identifikaci konkrétního modelu vozidla slouží datová struktura (konkrétně pro automobil) popsaná na obrázku 1.12. Jedná se o sadu tabulek, která slouží k vyhledávání modelu vozidel dle jména výrobce, roku výroby, modelové řady, generace a konkrétní motorizace. Zajímavostí je, že v Německu, na rozdíl od České republiky a Švýcarska, neberou v potaz druh převodové skříně. Ekvivalentní struktura je také definovaná pro motocykly, jelikož dle výzkumu výše se princip sjednání pojištění na automobil a motocykl nijak zásadně neliší.

1.5.3 Dokumenty

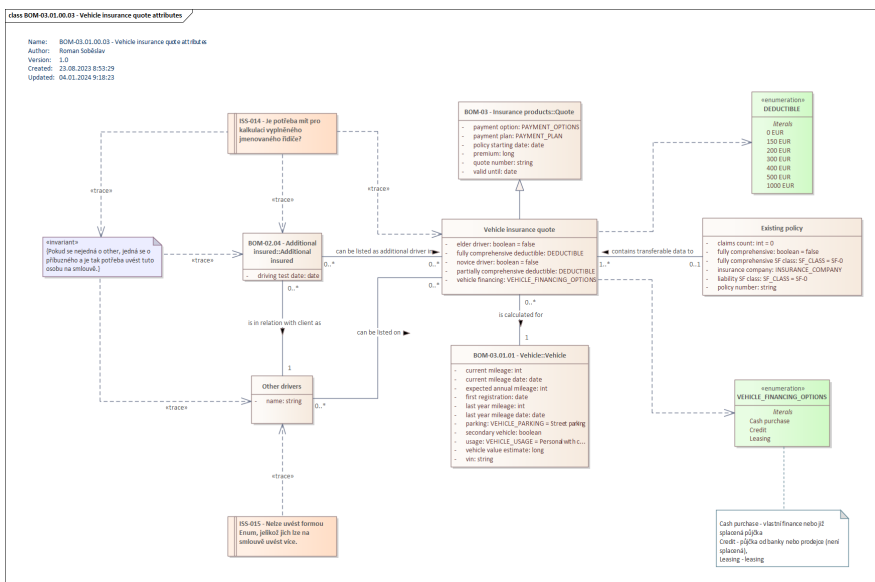
Velkou část tvoří správa dokumentů. Jedná se o uchování skenů občanských a řidičských průkazů, fotografií pojištěného majetku, ale i PDF výtisk pojistné smlouvy. Vytvořený model je navržen tak, že se budou ukládat dokumenty na souborový systém serveru a do databáze se zapíše cesta k tomuto souboru. Pro uchování těchto informací, spolu s metadaty (typ souboru, datum nahrání, informace o uživateli, který nahrál dokument, název souboru, asociovaný klient a smlouva) slouží objekt *Document*. Tuto informaci zachycuje obrázek 1.13.

1.5.4 Lokace

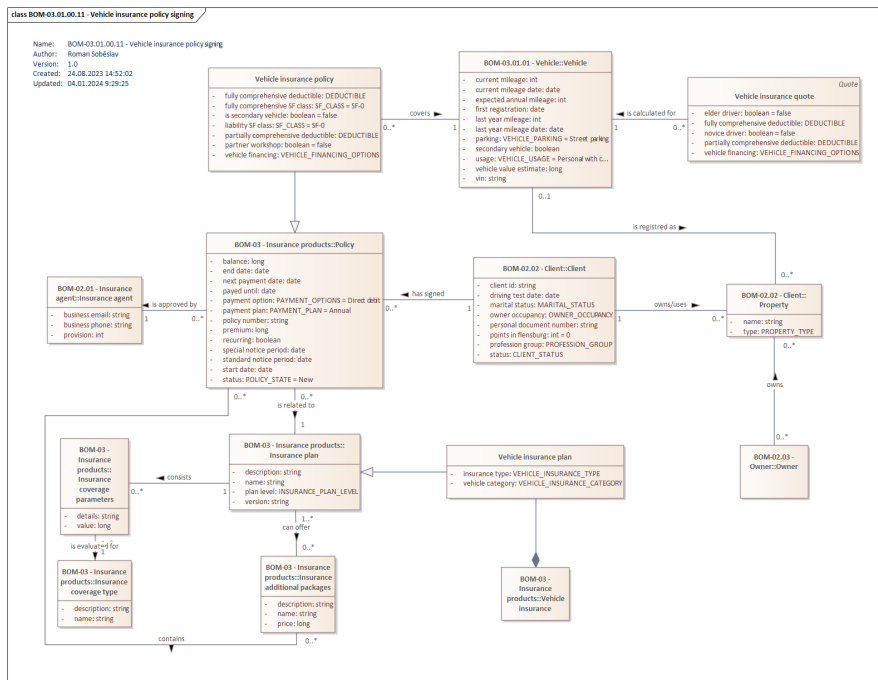
Lokace definuje dva datové objekty – *Branch* (pobočka) a *Address* (adresu). Pobočka definuje pracoviště zprostředkovatele a má přiřazenou adresu, která slouží k zaslání poštovní korespondence. Každá osoba má alespoň jednu adresu (trvalé bydliště/sídlo společnosti), nejvýše dvě



■ Obrázek 1.8 BOM - Atributy kalkulace pro pojištění vozidla



■ Obrázek 1.9 BOM - Vazba kalkulace pro pojištění vozidla na okolní entity



■ Obrázek 1.10 BOM - Pojistná smlouva pro pojištění vozidla

(trvalou a korespondenční adresu). Pro trvalou adresu a adresu sídla společnosti platí také to, že se jedná o jeden z parametrů pro výpočet pojistné částky. Tato část modelu je zachycena na obrázku 1.14.

1.5.5 Kalendář

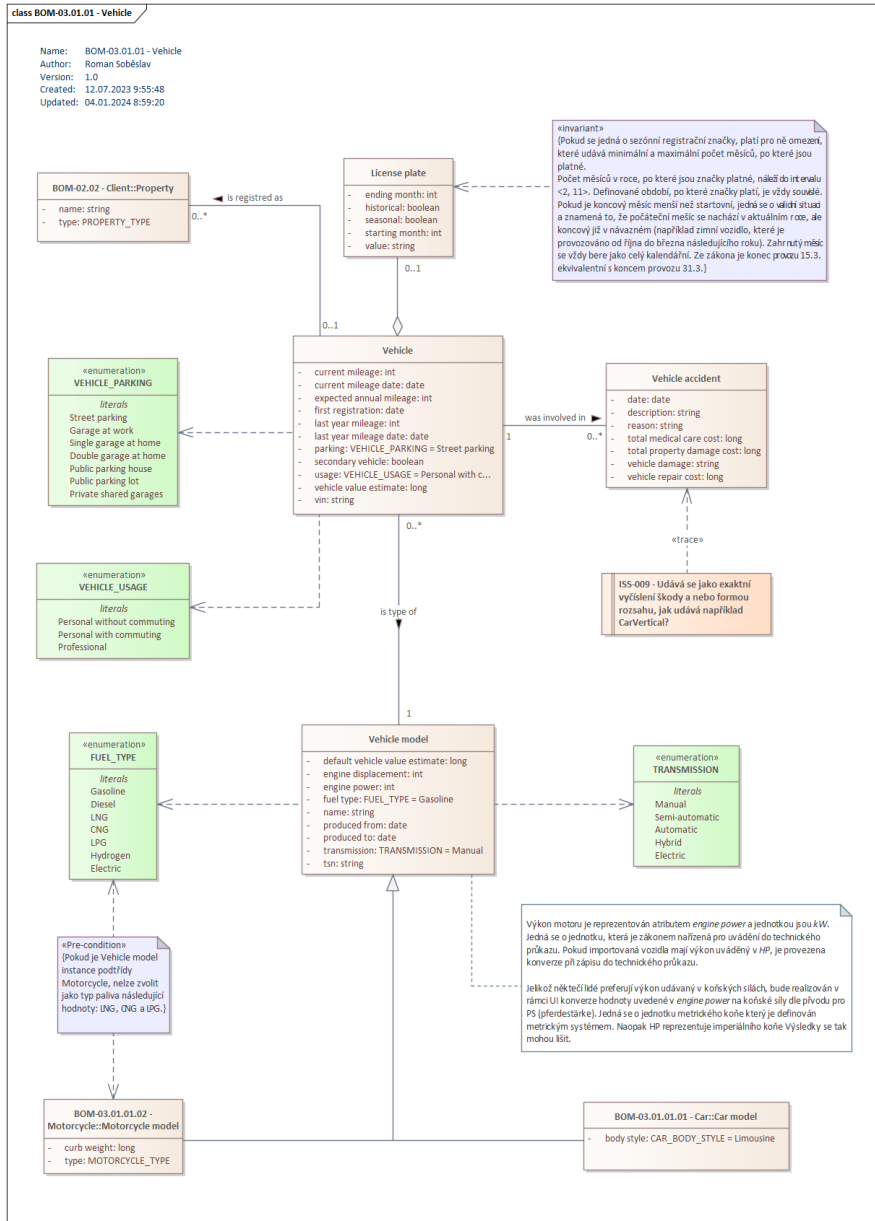
Jeden z funkčních požadavků je možnost plánování schůzek a úkolů pomocí kalendáře. Tento požadavek je realizován pomocí entit *Event* a *Task*. Jedná se o jednoduché entity, které drží název, stav a vazby na zprostředkovatele a klienta. Smyslem je implementovat jednoduchý TODO list spolu s kalendářem. Konkrétní návrh zachycuje obrázek 1.15.

1.5.6 Zprávy

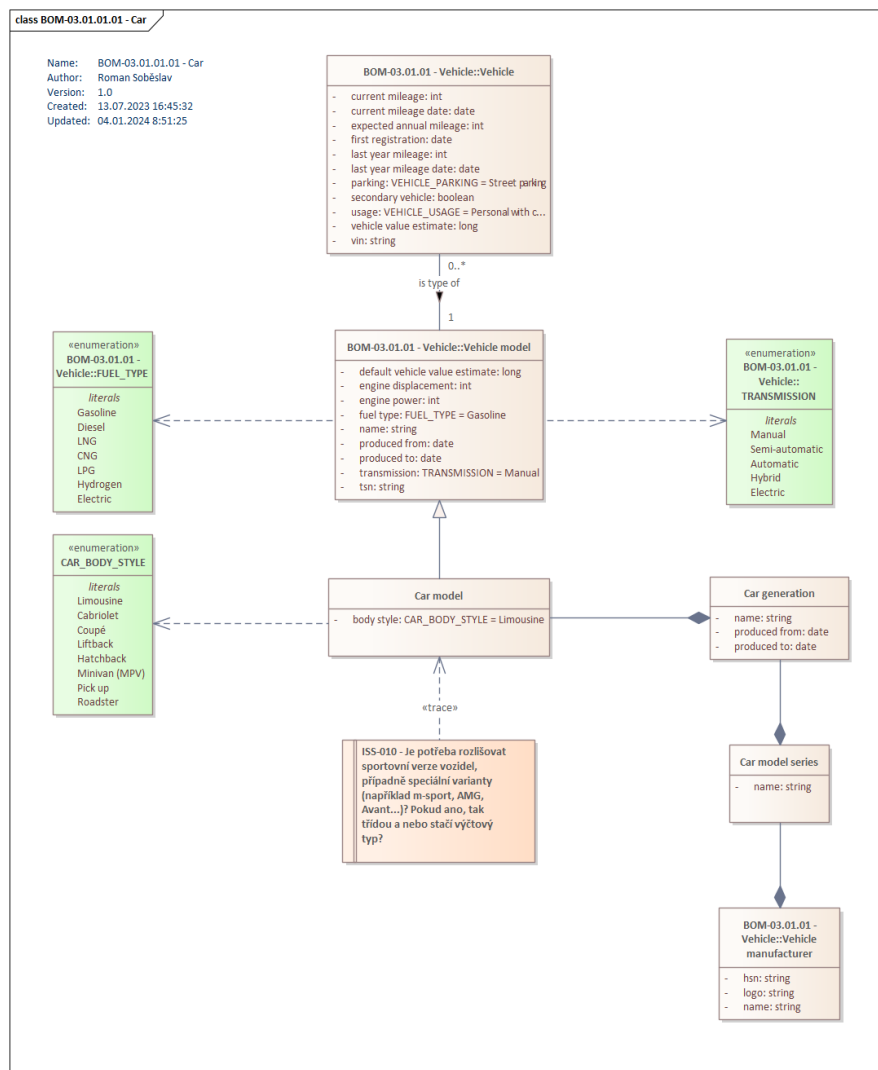
Dalším funkčním požadavkem je implementace jednoduchého rozhraní pro zaslání zpráv mezi zprostředkovatelem a klientem. Zaslou zprávu realizuje objekt *Message*. Jedná se o vazební entitu mezi entitami *Client* a *Agent*, která je obohacena o atributy obsahu zprávy. Důvodem existence číselníku pro způsob zaslání zprávy je myšlenka implementace této funkcionality skrze omni-channel řešení. Takto by bylo umožněna komunikace skrze platformu, která je nepohodlnější pro klienta. Návrh je zachycen na obrázku 1.16.

1.5.7 Obchodní příležitost

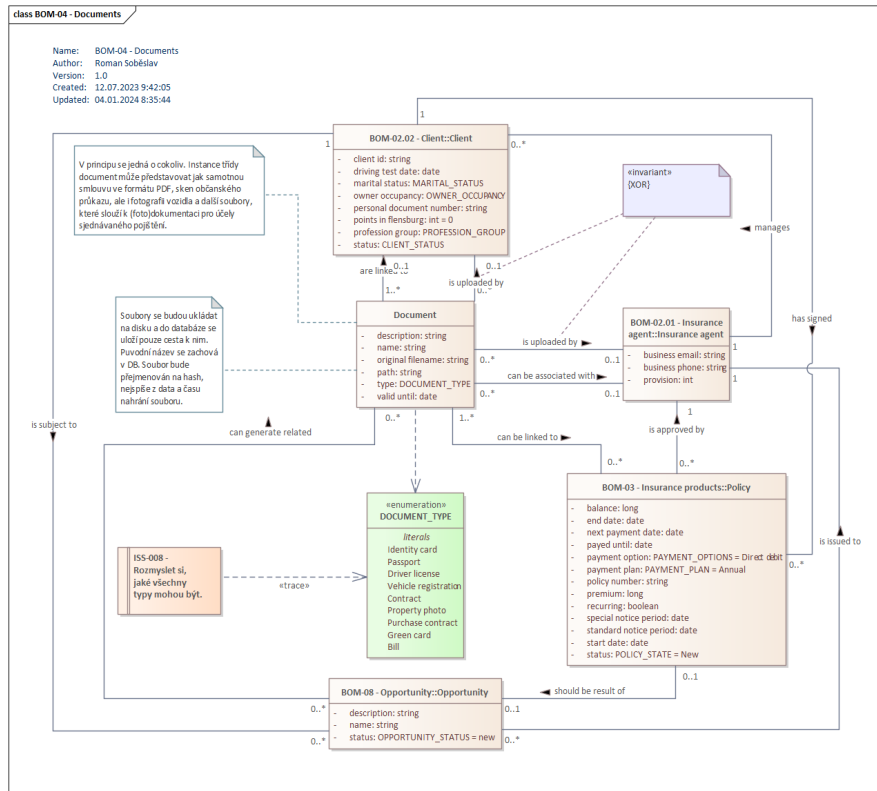
Posledním požadavkem pro namodelování je obchodní příležitost. Jedná se o dokumentační entitu, která má za úkol zaevidovat projevení zájmu klienta o sjednání pojištění a samotný průběh. Samotná entita má vazbu na klienta, který projevil zájem, agenta, který se touto obchodní záležitostí zabývá, přidruženou komunikaci, schůzky a úkoly. Výstupem jsou následně



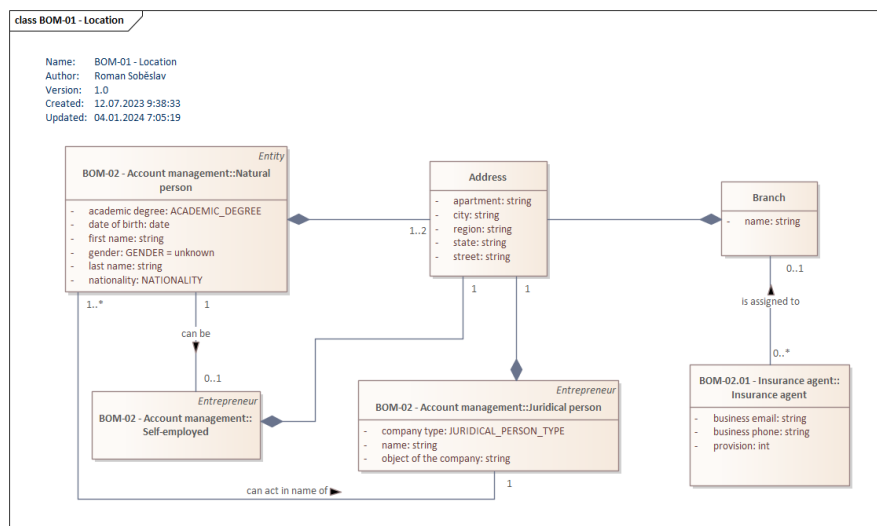
■ Obrázek 1.11 BOM - Vozidlo



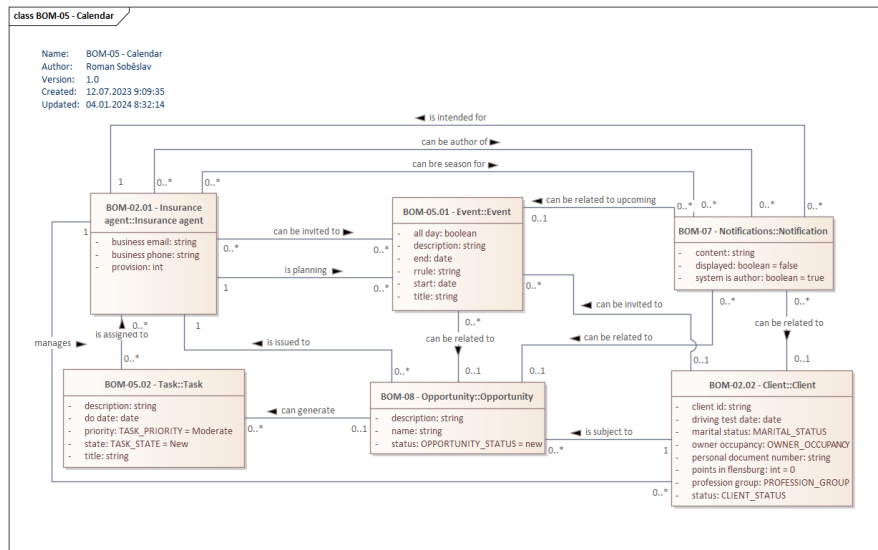
■ Obrázek 1.12 BOM - Automobil



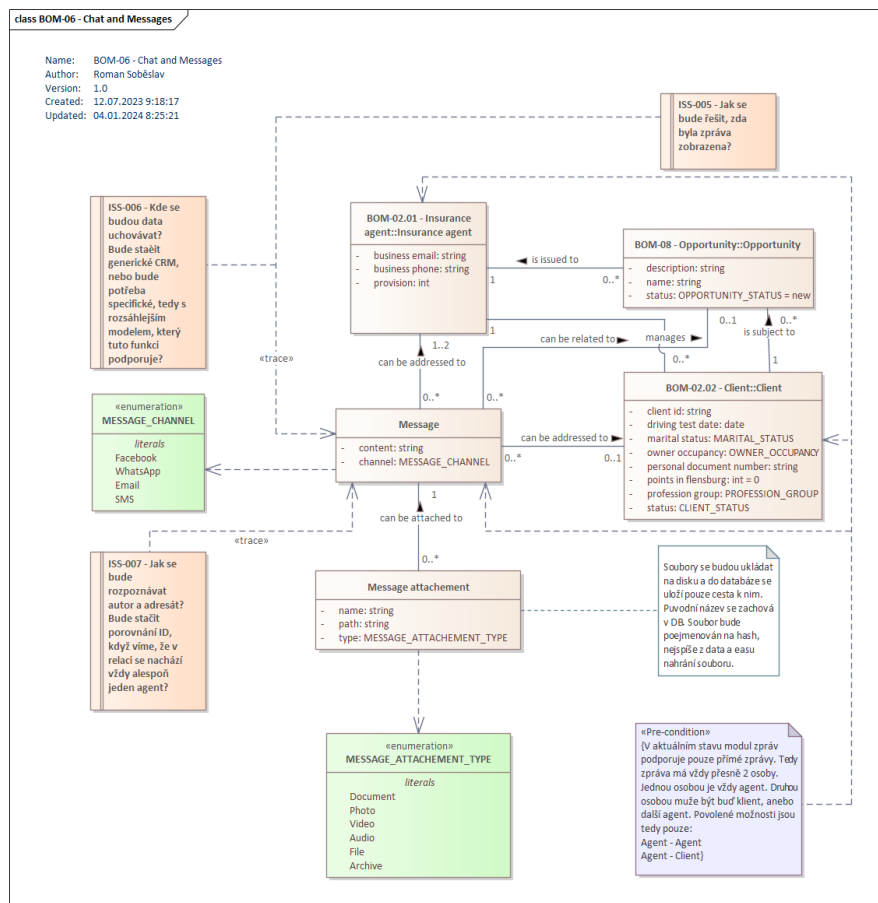
■ Obrázek 1.13 BOM - Dokumenty



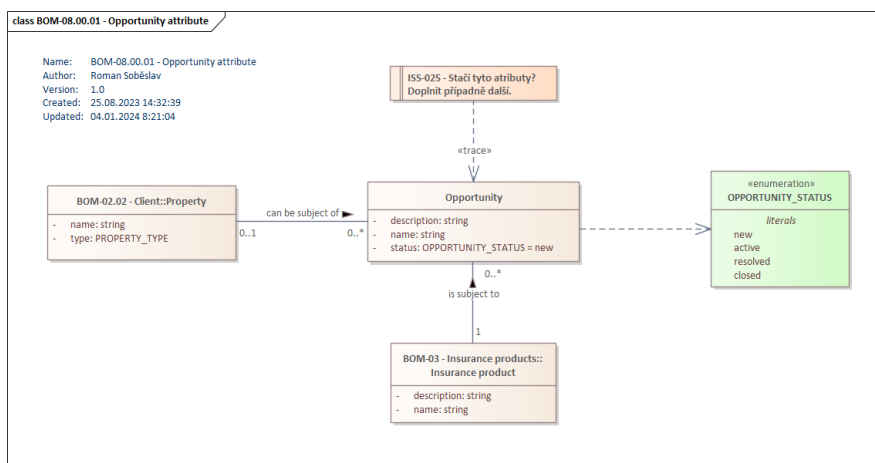
■ Obrázek 1.14 BOM - lokace



■ Obrázek 1.15 BOM - Kalendář

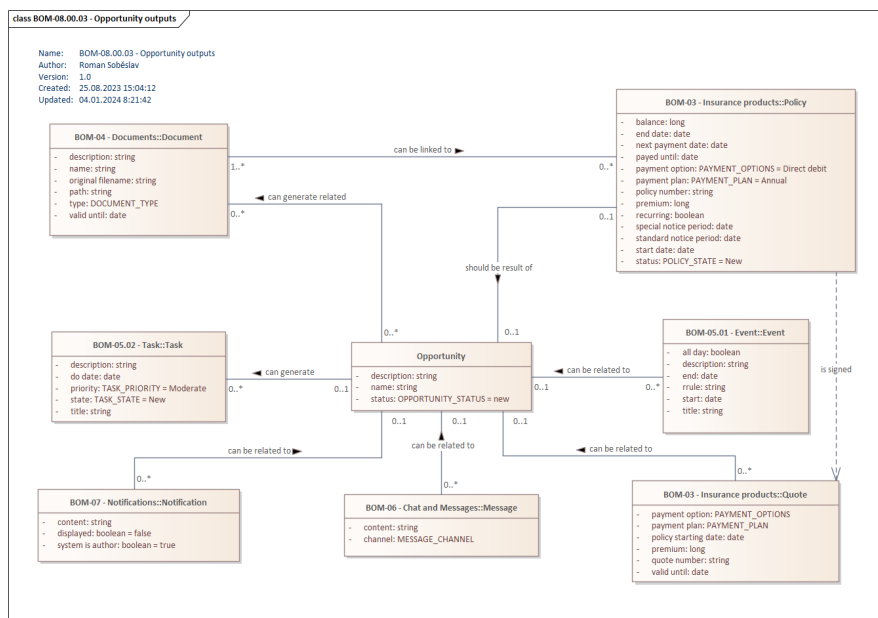


■ Obrázek 1.16 BOM - Textová komunikace



■ **Obrázek 1.17** BOM - Atributy pro obchodní příležitost

kalkulace, dokumenty a uzavřená pojistná smlouva. Více si lze prohlédnout na obrázcích 1.17 a 1.18.



■ Obrázek 1.18 BOM - výstupy obchodní příležitosti

Kapitola 2

Návrh

V této kapitole se zabývám konkrétním návrhem portálu, ze kterého vychází následná implementace. Nejprve se zabývám návrhem uživatelského rozhraní na základě poznatků z předchozí kapitoly. Následně zvolím technologie, které použiji pro implementaci finálního řešení. Jelikož dříve definovaný BOM je velmi rozsáhlý, vytvořím na jeho základě konkrétní datový model, který bude implementován. Nakonec představím návrh architektury, na které by se měl portál provozovat.

2.1 Návrh uživatelského rozhraní

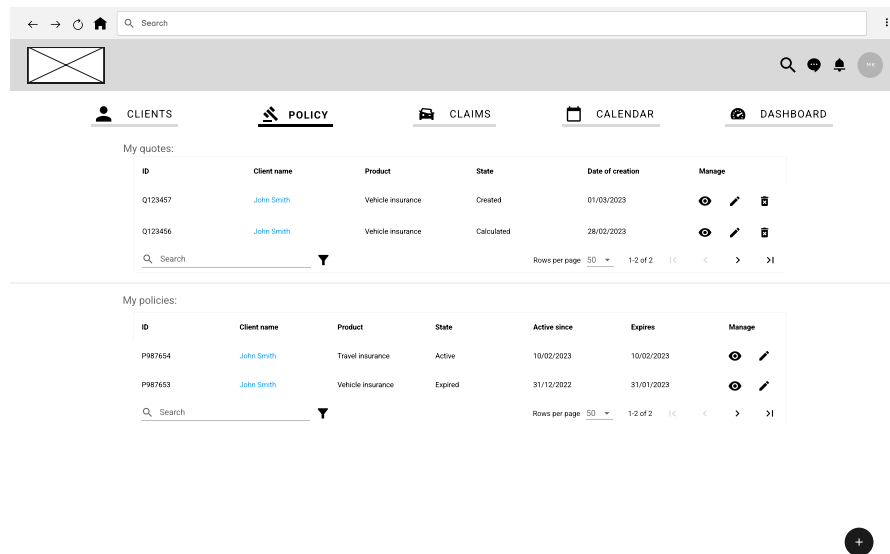
V příloze je přiložený kompletní návrh WF, který jsem vytvořil během analýzy a návrhu BOM v nástroji *Figma*. Při navrhování UI jsem využil designového jazyku Material design od Google[19]. Material design je soubor pravidel a doporučení pro tvorbu intuitivního a srozumitelného UI pro webové a mobilní rozhraní. Při návrhu byly použity pluginy *Ink Wireframe* a *Material Design Icons*. Vzniklý WF tvoří přibližně dvacet jedna typů obrazovek. Některé obrazovky mají odlišný design. Důvodem je časové rozpětí, ve kterém návrh vznikal. Během designování došlo ke změně designu. Z časových důvodů již ale nedošlo k úpravě původních obrazovek, čímž by došlo k sjednocení vzhledu.

První obrazovku tvoří stránka pro přihlášení. Skládá se pouze z loga pojišťovny a vstupů pro zadání emailu/uživatelského jména a hesla. Po úspěšném přihlášení je uživatel přenesen na hlavní obrazovku, kterou tvoří záložky s jednotlivými kategoriemi – klienti, smlouvy, kalkulace, škodní události, statistiky a kalendář. První čtyři kategorie tvoří přehledy o svém portfoliu formou tabulek a případných statistik. V pravém dolním rohu se nachází tlačítko pro vytvoření nového záznamu. Typ vytvořeného záznamu je určen zvolenou kategorií. Příklad si lze prohlédnout na obrázcích 2.1 a 2.2.

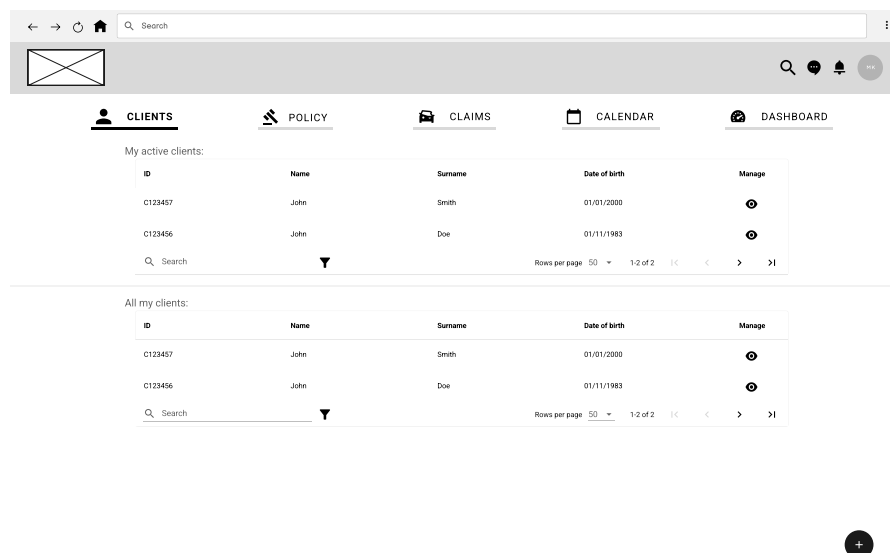
Kategorie kalendář slouží ke zobrazení úkolů a schůzek. Lze přepínat mezi agendou, pracovním týdnem, týdenním a měsíčním zobrazením. Kliknutí na událost v kalendáři se zobrazí podrobnosti. Pravé dolní tlačítko dovolí přidat další novou událost nebo úkol. Preferovaná forma implementace je zobrazovač dat z Google Gmail nebo Microsoft Outlook.

Navrchu každé obrazovky se nachází aplikační lišta, která obsahuje logo (slouží pro návrat na domovskou obrazovku), ikonu pro vyhledávání, komunikaci, notifikaci a profil přihlášeného uživatele. Kliknutím na ikonku vyhledávání se otevře dialogové okno. Nahoře se nachází lišta s kategoriemi pro vyhledávání. Automaticky je zvolena kategorie dle typu okna, ze kterého je vyhledávání vyvoláno. Pod lištou kategorií se nachází filtr a tabulka s výsledky. Všechny varianty si lze prohlédnout v příloze pod názvem *search dialog*.

Kliknutím na ikonu komunikace je uživatel přesunut na obrazovku s komunikačním rozhraní. V levé části okna se nachází seznam posledních konverzací a polem pro vyhledávání kontaktu



■ Obrázek 2.1 WF - seznam smluv



■ Obrázek 2.2 WF - seznam klientů

pro zahájení konverzace. Vpravo se nachází samotné okno komunikace. Uživatel si selektorem zvolí typ komunikačního kanálu. Výchozí hodnota je určena poslední zprávou. Pod selektorem se nachází okno se zprávami a polem pro vložení textu s tlačítkem pro odeslání. Uživatel má možnost připojit ke správě soubory a multimédia.

Kliknutím na ikonu zvonečku dojde k otevření menu, které obsahuje poslední notifikace a tlačítko pro přesunutí se do notifikačního centra. Notifikační centrum je rozdělená na dvě části – levá část obsahuje časovou osu notifikací od nejnovější po nejstarší notifikaci. Nezobrazená notifikace má v pravém horním rohu tečku. Každá notifikace nabízí seznam akcí formou tlačítek v závislosti na kontextu. Pravá část obsahuje seznam novinek, například aktualizace parametrů pojistného produktu. Kliknutím na náhled článku dojde k otevření článku formou dialogového okna.

Kliknutím na ikonu profilu je uživatel přesunut na stránku se svým profilem. Zde uživatel (zprostředkovatel) vidí své údaje, které si o něm pojišťovna eviduje. Jedná se o jméno, příjmení, datum narození, adresa, pracovní telefon a email. Pokud informace nesedí, může si zažádat o jejich aktualizaci.

Kliknutím na klienta v seznamu osob dojde k otevření jeho profilu. Zprostředkovatel vidí osobní informace o klientovi, kontaktní údaje, dokumenty, seznam kalkulací, seznam pojistných smluv a seznam nahlášených škodních událostí. Kliknutím na dokument dojde k zobrazení detailu. Při kliknutí na jakoukoliv položku z výše vyjmenovaných seznamů dojde k přesunutí na obrazovku s detailem. Kliknutím na ikonku plus v dolním pravém rohu dojde k vyvolání menu, ve kterém si uživatel může zvolit typ objektu, který chce vytvořit. Jedná se o nahrání dokumentu, nahlášení škodní události, vytvoření schůzky a založení nové kalkulace. Volba nahrání dokumentu otevře dialogové okno, kde uživatel zvolí název a druh dokumentu. Následně klikne na tlačítko vybrat soubor nebo jej přetáhne a potvrdí nahrání. Volba možnosti nahlášení škodní události přesune uživatele na obrazovku s aplikací pro nahlášení. Zvolením možnosti kalkulace dojde k vysunutí zespodu okna s nabídkou pojistných produktů ve formě kolotoče s kartami. Kliknutím na kartu produktu dojde k přesunutí do kalkulačky daného pojištění. Návrh přehledu klienta si lze prohlédnout na obrázku 2.3.

Pro návrh kalkulačky auto-pojištění jsem použil vícekrokový formulář. Při návrhu jsem vycházel z analýzy sjednávání aplikací, jelikož specializované kalkulačky pro zprostředkovatele nejsou veřejně přístupné. Zároveň platí, že v nejhorším případě má externí zprostředkovatel k dispozici stejný kalkulátor pojištění, jako klienti při sjednávání auto-pojištění on-line přes webový formulář. Rozhodl jsem se zvolit přístup, kdy je třeba vyplnit osobní informace hned na začátku, aby došlo k založení účtu, na který se následně naváží kalkulace. Důvodem je možnost vytvářet více podobných kalkulací, které jsou snadno zpětně vyhledatelné při jejich přepočtu. Nabídka je platná po určitou dobu a lze tedy uzavřít kalkulaci později. Klient tím dostává čas na rozmyšlenou.

Jak již bylo řečeno, první krok slouží ke sběru informací o pojistníkovi. Uživateli je nabídnuta možnost zvolit existujícího klienta nebo vytvořit nového. Následně je přesunut do prvního formuláře, kde je třeba vyplnit osobní informace o pojistníkovi. Při přístupu ke kalkulačce z přehledu klienta je automaticky formulář předvyplněn. Definovaná pole formuláře si lze prohlédnout na obrázku 2.4.

Po přesunutí se do dalšího kroku je zprostředkovatel vyzván k doplnění informací o vozidle. Uživatel má možnost vyhledat vozidlo ručně, nebo zvolit jednu z metod automatického vyhledání. Tyto možnosti jsou HSN/TSN, registrační značka a VIN. Po zvolení modelu vozidla je třeba doplnit další atributy jako je datum první registrace, registrační značka, použití a další. Kompletní seznam si lze prohlédnout na obrázku 2.5.

Třetí krok formuláře slouží k vyplnění parametrů pojištění. Je třeba zvolit počátek typ pojištění, platnosti, definovat seznam řidičů, nastavit spoluúčast atd. Konkrétní parametry si lze prohlédnout na obrázku 2.6

Krok čtvrtý slouží pro volbu pojistného plánu a příplatkových balíčků. Nabídka plánů je realizovaná pomocí plaveckých drah, kdy hlavička obsahuje název plánu a pojistné. Jednotlivé řádky

← → ↻ 🏠

🔍 🗲 🔔 ⋮

✕

Personal information:

Client ID: 123456789 ✕	Personal ID: 987654321 ✎	ID card number: 147258369 ✎
First name: John ✎	Last name: Smith ✎	Degree: M.C.S. ✎
Sex: Male ✎	Date of birth: 01/01/2000 ✎	Nationality: British ✎

Contact information:

E-mail: john.smith@example.com ✎	Phone: +44 07712 345 678 ✎
----------------------------------	----------------------------

Permanent address:

Street: Cherry Court ✎	Apartment: 71 ✎	City: SOUTHAMPTON ✎
Zip code: SO53 SPD ✎	Region: Hampshire ✎	State: United Kingdom ✎

Mailing address:

Street: Cherry Court ✎	Apartment: 71 ✎	City: SOUTHAMPTON ✎
Zip code: SO53 SPD ✎	Region: Hampshire ✎	State: United Kingdom ✎

Documents:

Id card front side:

✕

Id card rear side:

✕

Document name	Document type	Date	Manage
Driving license	Driving license	25/12/2022	👁️ 🗑️
Car registration	Vehicle registration	25/12/2022	👁️ 🗑️
Vehicle insurance card	Insurance card	30/12/2022	👁️ 🗑️
Rear bumper photo	Vehicle photo	25/01/2023	👁️ 🗑️
Rear left fender photo	Vehicle photo	25/01/2023	👁️ 🗑️

1-5 of 8 |< >|

Advisors:

Personal advisor: 963852741 ✎	Registered since: 31/12/2022 ✕	Registered by: 963852741 ✕
-------------------------------	--------------------------------	----------------------------

Quotes:

ID	Product	State	Date of creation	Manage
Q123457	Vehicle insurance	Created	01/09/2023	👁️ ✎ 🗑️
Q123456	Vehicle insurance	Calculated	28/02/2023	👁️ ✎ 🗑️

Rows per page: 50 | 1-2 of 2 |< >|

Policies:

ID	Product	State	Active since	Expires	Manage
P987654	Travel insurance	Active	10/02/2023	10/02/2023	🚗 👁️ 🗑️
P987653	Vehicle insurance	Expired	31/12/2022	31/01/2023	🚗 👁️ 🗑️

Rows per page: 50 | 1-2 of 2 |< >|

Claims

Personal car	Vehicle	No	102 3456	👁️ 🚗 👁️ 🗑️
Flat at 10th floor in block C	Building	Yes	Cherry Court 71, SOUTHAMPTON, SO53 SPD	👁️ 🚗 👁️ 🗑️

Rows per page: 50 | 1-2 of 2 |< >|

ID	Product	Reason	Date	State	Manage
	Personal car	Vehicle	No	102 3456	👁️ 🚗 👁️ 🗑️
	Flat at 10th floor in block C	Building	Yes	Cherry Court 71, SOUTHAMPTON, SO53 SPD	👁️ 🚗 👁️ 🗑️

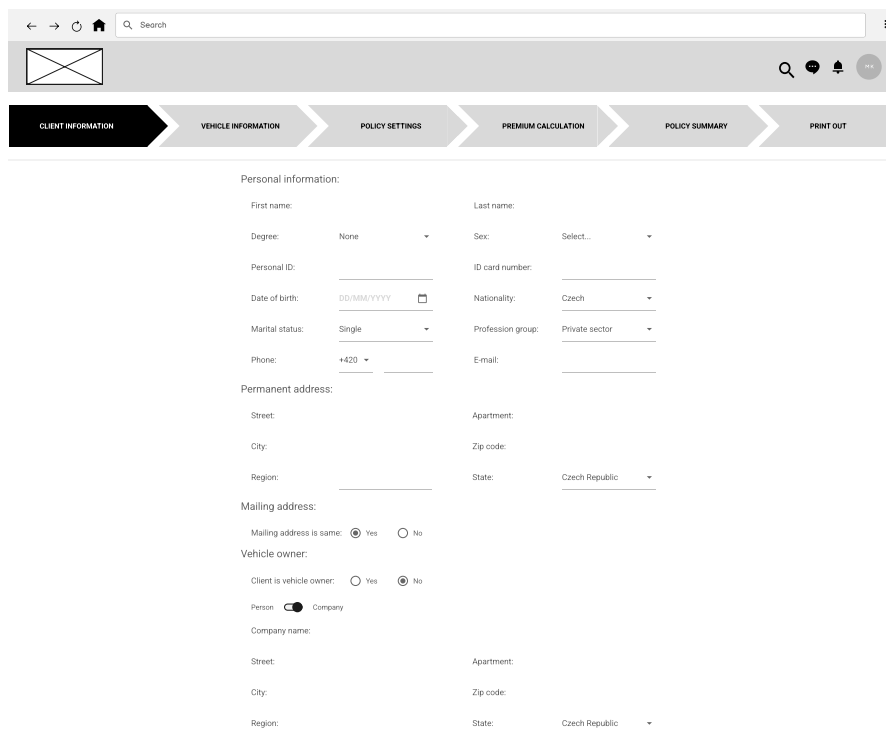
Rows per page: 50 | 1-2 of 2 |< >|

Known insurable property:

Name	Type	Is insured	Identification	Manage
Personal car	Vehicle	No	102 3456	👁️ 🚗 👁️ 🗑️
Flat at 10th floor in block C	Building	Yes	Cherry Court 71, SOUTHAMPTON, SO53 SPD	👁️ 🚗 👁️ 🗑️

Rows per page: 50 | 1-2 of 2 |< >|

■ Obrázek 2.3 WF - profil klienta



← → ↻ 🏠 Search

🔍 🗨️ 🔔 🌐

CLIENT INFORMATION VEHICLE INFORMATION POLICY SETTINGS PREMIUM CALCULATION POLICY SUMMARY PRINT OUT

Personal information:

First name:		Last name:	
Degree:	None ▾	Sex:	Select... ▾
Personal ID:	<input type="text"/>	ID card number:	<input type="text"/>
Date of birth:	DD/MM/YYYY 📅	Nationality:	Czech ▾
Marital status:	Single ▾	Profession group:	Private sector ▾
Phone:	+420 ▾ <input type="text"/>	E-mail:	<input type="text"/>

Permanent address:

Street:	<input type="text"/>	Apartment:	<input type="text"/>
City:	<input type="text"/>	Zip code:	<input type="text"/>
Region:	<input type="text"/>	State:	Czech Republic ▾

Mailing address:

Mailing address is same: Yes No

Vehicle owner:

Client is vehicle owner: Yes No

Person Company

Company name:

Street:	<input type="text"/>	Apartment:	<input type="text"/>
City:	<input type="text"/>	Zip code:	<input type="text"/>
Region:	<input type="text"/>	State:	Czech Republic ▾


■ Obrázek 2.4 WF - aplikace pro sjednání pojištění vozidla - 1. krok

CLIENT INFORMATION VEHICLE INFORMATION POLICY SETTINGS PREMIUM CALCULATION POLICY SUMMARY PRINT OUT

CAR MOTORCYCLE

VIN HSN / TSN LICENSE PLATE **MANUAL** REGISTRATION CERTIFICATE

Vehicle


BLITZ ALPHA A30I
 HSN / TSN: 0005/AFF

Body	Coupe
Fuel type	Gasoline
Engine power	306 HP / 225 kW
Engine displacement	2979 cm ³
Transmission	Manual

License plate
 License plate type: Standard Historical
 Seasonal license plate: Yes No
 License plate number: I.e. M-B 1234

Vehicle usage
 Primary usage: Personal vehicle
 Primary parking type: Garage at home
 Annual driven mileage: 5.000 km
 Estimated value of the vehicle: 12.358 EUR

Vehicle identification
 VIN
 First registration: 15/06/2009

■ Obrázek 2.5 WF - aplikace pro sjednání pojištění vozidla - 2. krok

CLIENT INFORMATION VEHICLE INFORMATION **POLICY SETTINGS** PREMIUM CALCULATION POLICY SUMMARY PRINT OUT

Driver information:

Driving test date: DD/MM/YYYY Points in Flensburg: 0
 Vehicle financing: Cash purchase Type of dwelling: Apartment

Other drivers

Other car driver(s): Yes No
 Spouse Life partner Parents Children Other

Drivers younger than 25: Yes No Drivers older than 65: Yes No

SF class

Existing car insurance: Yes No
 Insurance company: Car Insurance LLC Policy number:
 Claims in last 3 years: None Fully comprehensive in last 12 months: Yes No
 Liability SF class: Select class Fully comprehensive SF class: None

Insurance settings:

Start of insurance: DD/MM/YYYY Insurance type: Fully comprehensive
 Secondary vehicle: Yes No Partner car workshop: Yes No

Deductible:

Partially: 150 EUR Fully: 300 EUR

Billing:

Billing cycle: Annual Payment option: Cash collection

■ Obrázek 2.6 WF - aplikace pro sjednání pojištění vozidla - 3. krok

← → ↻ 🏠 🔍 Search

CLIENT INFORMATION VEHICLE INFORMATION POLICY SETTINGS PREMIUM CALCULATION **POLICY SUMMARY** PRINT OUT

☑ Client information: ▾

☑ Vehicle information: ▾

○ Policy settings: ▴

Driver information

Driving test date
13/06/2018

Points in Flensburg
0

Vehicle financing
Cash purchase

Type of dwelling
Apartment

Insurance settings

Start of insurance
18/05/2023

Insurance type
Fully comprehensive

Secondary vehicle
No

Partner car workshop
Yes

Existing car insurance

Insurance company
Car Insurance LLC

Policy number
CPN123456789

Deductible

Liability insurance
150 EUR

Fully comprehensive
300 EUR

← PREVIOUS STEP

SAVE AS QUOTE

NEXT STEP →

SP-3

Fully comprehensive SP class
SF-3

Annual

Payment option
Cash collection

Other drivers

Selected other drivers

Spouse Life partner

Parents Children Other

Drivers younger than 25
Yes

Drivers older than 65
No

UPDATE VALIDATE

○ Insurance plan: ▾

○ Price summary: ▾

○ Documents: ▾

■ **Obrázek 2.7** WF - aplikace pro sjednání pojištění vozidla - 5. krok - rekapitulace osoby

tabulky představují druhy škod. Buňky tabulky obsahují informaci, zda je daný typ poškození zahrnuto do daného plánu, či nikoliv. Alternativou jsou doplňující informace, například omezující podmínky nebo definice maximální částky, kterou pojišťovna vyplácí. Kliknutím na tlačítko zvolit je uživatel přesunut na další stránku, kde lze přidat volitelné příplatkové balíčky.

Posledním krokem je rekapitulace formuláře. Zprostředkovateli jsou postupně ukazovány vyplněné části formuláře pro kontrolu. Pokud je nalezena chyba, lze se kliknutím na tlačítko *update* přesunout na příslušnou obrazovku pro opravení chyby. V opačném případě lze kliknout na tlačítko *validate*, které slouží k potvrzení údajů. Na konci je možné kliknout na tlačítko pro uzavření smlouvy, čímž dojde k zafixování parametrů kalkulace a vytvoření pojistné smlouvy. Vzniklou smlouvu lze také vytisknout. Ukázkou rekapitulace si lze prohlédnout na obrázcích 2.7 a 2.8.

Při zvolení kalkulace ze seznamu je uživatel přeměrován do kalkulačky, která je předvyplněna dle uložených informací. Pro uložení kalkulace je třeba se dostat do čtvrtého kroku, kdy dojde k uložení hodnot do databáze a ocenění kalkulace. Alternativou je vytvoření kopie kalkulace pro vytvoření další nabídky.

Kliknutím na pojistnou smlouvu je uživatel přesunut na stránku s přehledem smlouvy. V horní části se nachází základní údaje o pojistné smlouvě – číslo pojistné smlouvy, stav smlouvy (nová, běžící, ukončená nebo zrušená), pojistné, zůstatek, datum další splátky a výpovědní lhůty.

← → ↻ 🏠 Search

📧 🔍 🔔 🔒

CLIENT INFORMATION VEHICLE INFORMATION POLICY SETTINGS PREMIUM CALCULATION **POLICY SUMMARY** PRINT OUT

Client information: ▾

Vehicle information: ▾

Policy settings: ▾

Insurance plan: ▸

Content of the insurance plan

Covered insurance events	Additional specifications
Damage you cause to others	Damage caused to other people's property, financial loss and personal injury are covered up to a total of EUR 75 million, with a maximum of EUR 15 million per person.
Towing in the event of an accident	
Foreign Damage Protection	Includes only EU countries, Liechtenstein, Monaco, San Marino and Switzerland.
Damage to own property	Up to 50.000 EUR per insured year.
Collision with animals	

← PREVIOUS STEP

SAVE AS QUOTE

NEXT STEP →

Theft

Forces of nature

Fire

Purchase price guarantee Up to 24 months.

New price guarantee Up to 24 months.

Car sharing protection Up to 500 EUR per claim.

Damage to your own car due to accidents you are at fault

Vandalism

Additional packages

Package name	Description
Premium Car assistance and temporary vehicle	In the event of an accident or car breakdown, we will tow your car away and provide you with a replacement vehicle.

UPDATE VALIDATE

Price summary: ▾

Documents: ▾

■ Obrázek 2.8 WF - aplikace pro sjednání pojištění vozidla - 5. krok - rekapitulace pojistného plánu

Následuje přehled informací o pojistníkovi, specifická část pro typ pojistné smlouvy (v případě pojištění vozidel jsou uvedeny informace pojištěném vozidle), parametry sjednané smlouvy, seznam dalších osob na smlouvě (majitel a další pojištěné osoby), přiložené dokumenty, výčet škodních událostí kryté danou smlouvou a výčet příplatkových služeb.

Interaktivní prototyp je k dispozici na této URL: <https://figmashort.link/2bFcAT>

2.2 Technologie

V této sekci se zabývám výběrem technologií pro vývoj samotného řešení. První podsekcí se věnuje popisu portálového řešení s názvem Liferay od stejnojmenného výrobce. Jedná se o funkční požadavek, kdy vzniklý portál pro zprostředkovatele musí být možné integrovat do této technologie z důvodu kompatibility s ostatními částmi produktu Cleeo. Následuje výběr databázového systému, který je využit jakožto datové úložiště pro simulaci infrastruktury zákazníka. Poslední podsekcí se zabývá výběrem JavaScriptového frameworku pro implementaci uživatelského rozhraní.

2.2.1 Liferay

Liferay[20] je technologie pro vývoj podnikových portálových řešení v jazyce Java skrze specifikaci Java portletů. Java portlet je specifikace, která umožňuje vytvářet malé nezávislé webové aplikace, které jsou následně nasazovány/integrované do portálového řešení.[21, 22] Od servletů se liší primárně tím, že servlet běží na serveru a stará se o zpracování HTTP požadavků, kdy vrací celou stránku. Naopak portlet je určen k implementaci malých modulů (widgetů), které se následně dynamicky umísťují na stránky vytvořené v portálu. Pro svoji správnou funkčnost mají rozšířený životní cyklus (inicializace, renderování, zpracování požadavku, ...). Standardně se umísťuje na stránku více různých portletů, které mohou být na sobě nezávislé nebo mohou spolu spolupracovat. Často se využívají jakožto náhled do části okolních systémů.

Samotný Liferay portál je vydáván ve dvou variantách – komunitní edice a rozšířená placená varianta s názvem DXP (Digital experience platform). Pro účely produktu Cleeo je plně dostatečná základní komunitní edice, která je k dispozici zdarma pod open-source licencí. Nabízí základní portálové řešení, které obsahuje část pro správu webového obsahu, účtů, stránek a widgetů. Od verze 7.4 je nově součástí i do té doby samostatný modul pro e-commerce pro tvorbu e-shopů.

Placená DXP verze vychází z komunitní edice, kdy poskytuje pokročilejší funkce pro správu. Hlavními výhodami placené verze je nejaktuálnější dokumentace, placená podpora (včetně hot-line) a volba z předem připravených specializovaných prototypů dle odvětví (pojišťovny, zdravotnictví, ...). Jedná se tedy o nadstavbu komunitní edice pro zrychlení nasazení a integrace v korporátním prostředí.

Alternativním řešením k Liferay portálu:

IBM WebSphere Portal Jedná se o podobné řešení jako je Liferay. Vznikl v roce 2001. V roce 2019 jej IBM prodal společnosti HCL Technologies. Poslední verzí je 9.5 z roku 2019.

SharePoint Slouží pro tvorbu intranetů, sdílení dokumentů a spolupráci. Vytvořil jej Microsoft v roce 2001.[23]

Kromě výše vyjmenovaných existovalo mnohem více produktů (například Oracle WebLogic Portal, Sun Java System Portal Server, JBoss Enterprise Portal Platform), které poskytovali funkcionalitu pro tvorbu podnikových portálů pomocí Java portletů. V současné době je však většina projektů již ukončených a Liferay tak zůstává jako jediná dále se rozvíjející platforma tohoto typu.

2.2.2 Databáze

Pro účely své práce jsem se rozhodl použít relační databázové systémy. Důvodem je možnost snadného vyhledávání v tabulkách a nativní podpoře Liferay pro práci s RDBMS. Pro účely ukládání dokumentů lze využít no-SQL databázi (například MongoDB). Pro účely tohoto projektu se jedná o zbytečnou komplikaci, kdy je třeba si vytvořit vlastní rozhraní pro komunikaci s MongoDB a Liferay. Rozhodl jsme se tedy ukládat dokumenty na disk a do relační databáze ukládat pouze cestu k souboru.

Při výběru databáze jsem primárně zohledňoval požadavek na dostupnost databáze pod open-source licencí. Databáze slouží pouze pro testovací a prezentační účely a není tedy třeba volit placené varianty, které nabízí nevyužívané funkcionality. Liferay portál přistupuje k databázi skrze Hibernate framework. Lze tak používat pouze standardní funkcionality, které jsou univerzální pro všechny typy SQL databází.

Posledním požadavkem je podpora více paralelních databází v rámci jedné instance. Důvodem je provozování projektové a Liferay operační databáze na jedné databázi.

Seznam uvažovaných databází:

Oracle Oracle poskytuje svůj vlastní databázový systém pro enterprise řešení. Jedná se o velmi kvalitní databázi škálovanou na velký provoz v enterprise prostředí. Dále poskytuje velmi kvalitní dokumentaci. Toto je bohužel vykoupeno vyšší pořizovací cenou, která nesplňuje výše uvedené kritérium.

PostgreSQL Jedná se o velmi populární open-source databázové řešení od PostgreSQL Global Development Group. V roce 2023 udržela již po čtvrté ocenění databáze roku.[24] Mezi výhody patří škálovatelnost, podpora různých datových formátů, uživatelská podpora, rozsáhlá komunita, a jednoduchá integrace nástrojů třetích stran.

MySQL Jedná se o původně open-source řešení, které bylo koupeno Sun Microsystems. Aktuálním majitelem je Oracle. Hlavní výhodou je snadná integrace s cloudovými poskytovateli díky Oracle. Přestože je k dispozici licence zdarma, pro komerční účely může být zpoplatněna.

Microsoft SQL Relační databázový systém vytvořený firmou Microsoft. Nabízí zdarma verzi pro vývojáře, případně edici Express. Omezením této licence je hlavně hardwarové a slouží pro výuku. Pro reálné použití je třeba zaplatit licenci business licenci. Pro účely nasazení projektu na testovací a prezentační prostředí se jedná o horší řešení než volba PostgreSQL.

Pro účely této práce jsem se rozhodl využít PostgreSQL. Poskytuje veškerou svoji funkcionalitu bez nutnosti zakoupení licence. Lze zároveň provozovat na jedné RDBMS obě požadované databáze. Zároveň nebude třeba řešit migraci mezi databázovými systémy v případě, že je potřeba dodat část databáze, a narazí se na problém licencí.

2.2.3 Framework pro tvorbu uživatelského rozhraní

Liferay nabízí tvorbu portletů formou JSP, případně použití JavaScriptu (je k dispozici JQuery). Zároveň poskytují nástroje pro podporu tvorby portletů formou čistě JavaScriptových frameworků pro tvorbu single page aplikací. Lze tedy vytvořit takzvaný *js-widget* pomocí jejich nástroje *liferay-js-generator*. Podporované varianty jsou čistý JavaScript s HTML v kombinaci s JQuery, Angular2, React, Vue.js a jejich knihovna Metal.js. Jelikož v dnešní době uživatelé požadují moderní intuitivní a přehledné aplikace, rozhodl jsem se využít frameworku pro single page aplikaci. Důvodem je použití frameworku i v dalších komponentech Cleeo řešení (například klientská zóna, hlášení škodních událostí a další), které mohou využít možnosti vytvoření portletu jako single page aplikace.

Po aplikaci výše popsaného omezení lze uvažovat pouze nad implementací skrze Angular, React nebo Vue.js.

Angular 2 je framework postavený nad TypeScriptem, který doplňuje klasický JavaScript o datové typy. Díky tomu je údržba rozsáhlých projektů mnohem jednodušší. Autorem frameworku je Google, který jej vydal v roce 2016. Přestože dlouhodobě užívatelská základna klesá, zvládl se udržet v korporátním prostředí. V posledních letech prošel velkými updaty, které opravili některé nedostatky. Aktuální verze 17 přinesla spoustu novinek včetně takzvaných signálů, čímž si framework získal opět na popularitě. Hlavními přednostmi frameworku je obousměrný data-binding.

React je JavaScript knihovna vytvořená Facebookem v roce 2013. Od roku 2022 podporuje také TypeScript. V době psaní se stále jedná o nejpoblárnější JavaScript framework, ale v posledních letech jeho obliba lehce klesá díky častým změnám, které nejsou zpětně kompatibilní. Za poslední dva roky se část komunity přesunula k jiným frameworkům, které staví na principech Reactu, nebo jej rozšiřují. (Například Next.js). Mezi hlavní výhody frameworku patří podpora JSX a s tím spojených virtuálních DOMů.

Vue.js je druhý nejpoblárnější framework, který staví na podobných principech jako React. Původně vznikl v roce 2014 v Googlu jakožto nástroj pro prototypování. V současnosti se o vývoj stará sám autor Evan You, který jej poskytuje open-source pod licencí MIT. Na rozdíl od Reactu podporuje obousměrný data-binding. Stejně jako React, i Vue.js podporuje TypeScript.

Přestože preferuji React framework, rozhodl jsem se použít Angular2. Za posledních pět let prošel velkými změnami, které odstranili většinu neduhů. Autoři přidali takzvaný *Treeshaking*, který dokáže ořezat sestavenou aplikaci o nepoužívané části knihoven, čímž dojde k zmenšení výsledného balíčku, který musí uživatel stahovat. Dalším důvodem pro použití je rozšířenost frameworku ve firmě Cleverbee.

2.3 Logický datový model

Navržený BOM je pro účely této práce zbytečně komplexní. Rozhodl jsem se jej zjednodušit. Došlo k úplnému odebrání sekcí lokace, komunikace, kalendář, notifikace a obchodní případ. Důvodem bylo soustředění se čistě na externí partnery, kteří pro většinu těchto funkcí používají vlastní systémy. Pokud by byla potřeba tyto informace později doplnit (například pro rozšíření použití portálu i pro použití interních zaměstnanců), lze je vhodně doplnit. Výsledek si lze prohlédnout na obrázku 2.9.

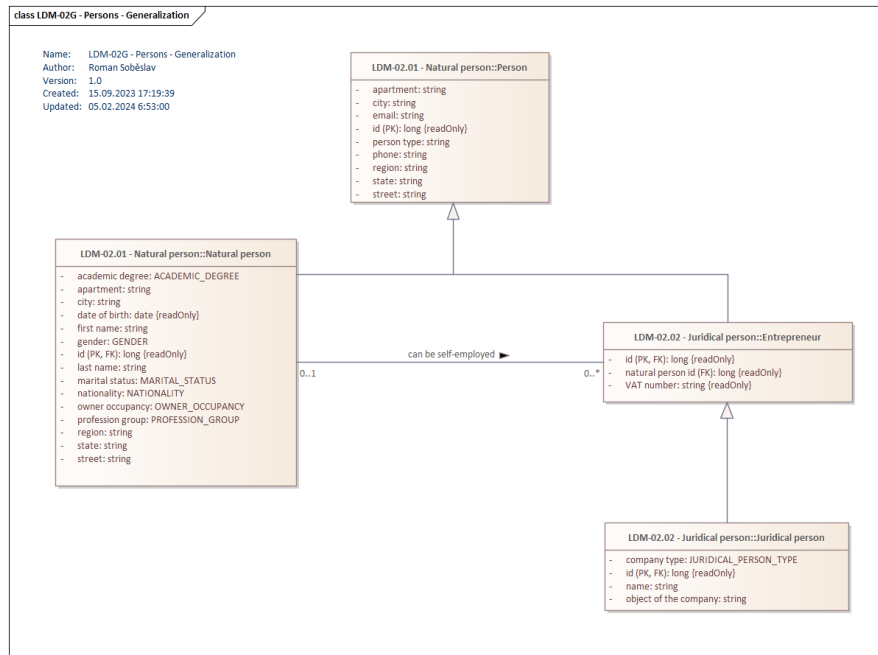
Objekt adresy byl sloužen s objekty *Insurance_agent*, *Person* a *Natural_person*. Koncept dědění pro sestavení osoby byl zachován a předek všech osob byl přejmenován na *Person*. Došlo k odebrání rolí. Nově jsou role definovány jako parametr ve vazebním objektu *Involved_peron*. Výsledek si lze prohlédnout na obrázku 2.10

Přepracování se dočkal i zprostředkovatel, který byl extrahován do vlastní nezávislé entity. Část pro evidování smluv pro účely human resources systému byl odebrán pro nadbytečnost. Zjednodušen byl i princip licencí, které byly nahrazeny tabulkou pro oprávnění sjednání daného pojistného produktu. Nově byl přidán atribut, který ovlivňuje viditelnost pojistných smluv agentem. Během návrhu byl přidán požadavek, že zprostředkovatel standardně vidí pouze své sjednané smlouvy. Tuto skutečnost zachycuje obrázek 2.11.

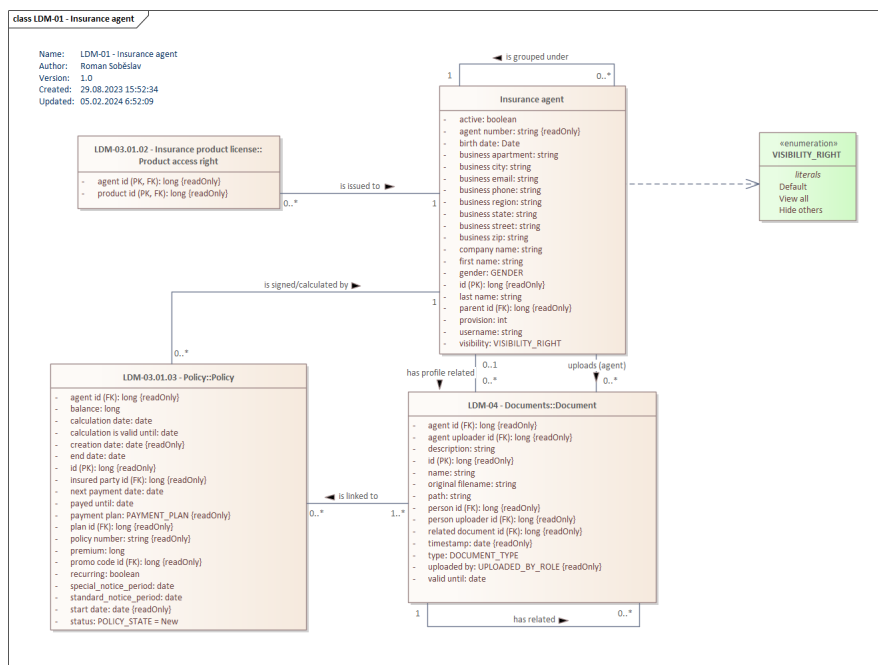
Největší změnou prošla část pro práci s pojistnými smlouvami. Hlavní změnou je odebrání kalkulace. Nově je kalkulace speciální případ pojistné smlouvy, která je ve fázi přípravy. Tato změna měla za následek velké zjednodušení výsledného modelu de-duplikací tabulek a vazeb. Zbytek modelu byl zachován. Výsledek je zachycen na obrázku 2.12

Zachování se dočkala i sekce vozidla, která byla převzatá beze změny. Výsledek zachycuje obrázek 2.13

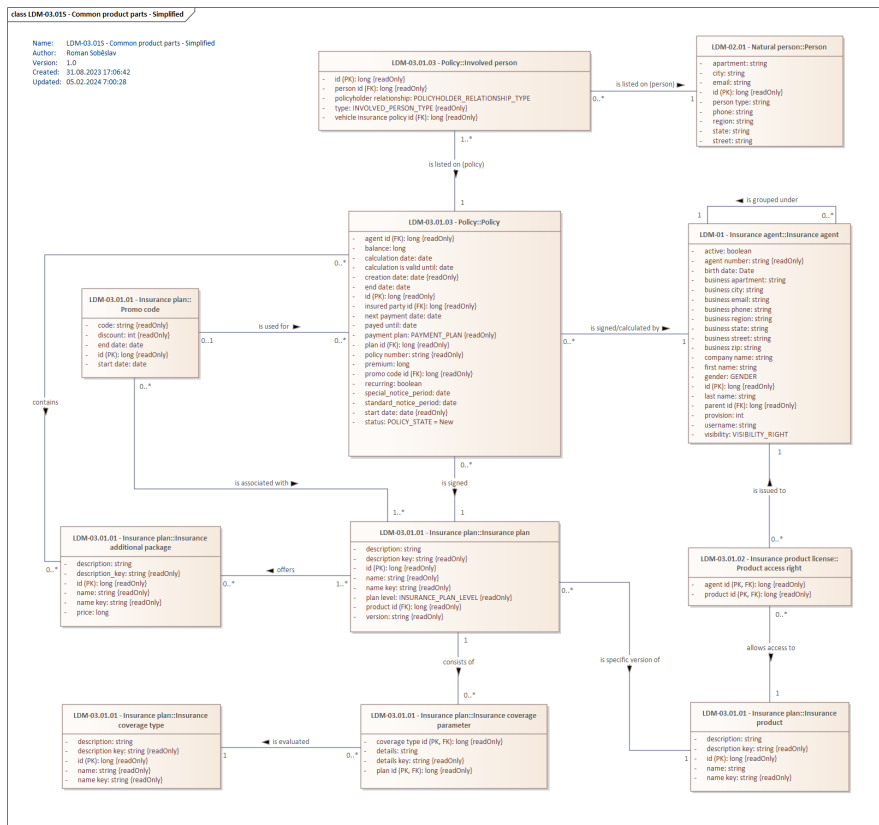
Bez úpravy zůstala také sekce dokumenty. Pouze byly odebrány vazby na obchodní příležitost, viz obrázek 2.14



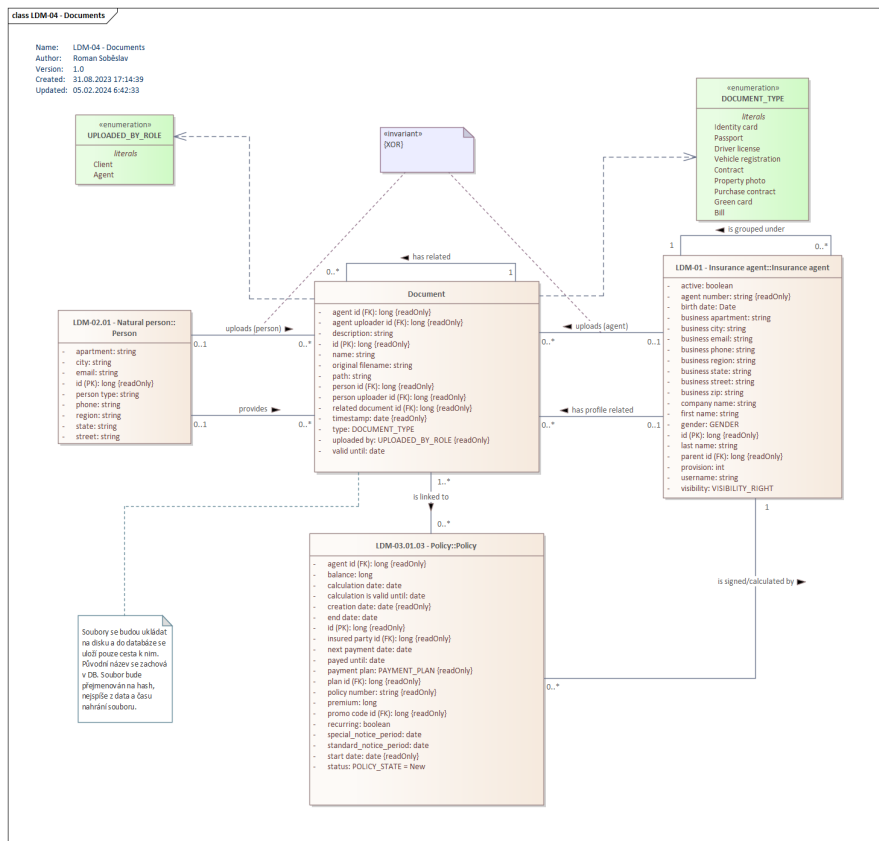
■ Obrázek 2.10 LDM - osoby



■ Obrázek 2.11 LDM - agent



■ Obrázek 2.12 LDM - společná část pojištění



■ Obrázek 2.14 LDM - dokumenty

2.4 Databázový model

Následující obrázky 2.15 a 2.16 zachycují fyzický model. Jedná se o transformovaný logický datový model pomocí Enterprise Architect. Tabulky jsou totožné s třídami v LDM. Číselníky byly transformovány na tabulky, které obsahují pouze ID a hodnotu, kterou využívají kombo-boxy. Vazby typu many-to-many byly transformovány na vazební tabulky, pokud již nebyly dekomponovány v LDM.

2.5 Architektura

V příloze se nachází diagramy pro tři návrhy architektur, na kterých bude vzniklý portál provozován. Obrázky 2.17 a 2.18 popisují pouze dva případy, jelikož třetí *SA-02 - On-premise with ESB* diagram popisuje stejnou architekturu jako *SA-01 - On-premise*, akorát se zde navíc nachází ESB, přes který probíhá veškerá komunikace mezi systémy.

První diagram, zachycený na obrázku 2.17, popisuje ideální infrastrukturu zákazníka. Diagram je rozdělený do čtyř barevných částí. Modrá část reprezentuje server, na který se bude nasazovat portál. Zelená část reprezentuje infrastrukturu zákazníka, na kterou se portál integruje. Žlutá část reprezentuje uživatelská zařízení, ze kterých se přistupuje k portálu. Poslední červená část reprezentuje služby třetích stran, které potřebuje portál ke svému provozu. Toto barevné schéma je shodné u všech diagramů.

Ze služeb třetích stran portál využívá napojení na Google API pro automatické doplňování adres. U zařízení, která zprostředkovatelé budou používat, se jedná převážně o tablety a notebooky, které mají nainstalovaný internetový prohlížeč.

Infrastruktura zákazníka slouží jakožto úložiště a zdroj dat, včetně modulů pro modelování pojistných produktů a jejich ocenění. Samotná infrastruktura se skládá z následujících částí:

Adresářový server Poskytuje informace o externích zprostředkovatelích, kteří mohou využívat portál. Server je využíván jako zdroj identit.

Billing Poskytuje informace o platbách a zůstatku na smlouvách.

Core Stará se o vytvoření a ocenění kalkulací. Poskytuje informace o pojistných produktech.

Customer relation management Datové úložiště pro klienty.

Data warehouse Obsahuje veškeré informace, které se následně vizualizují pomocí grafů.

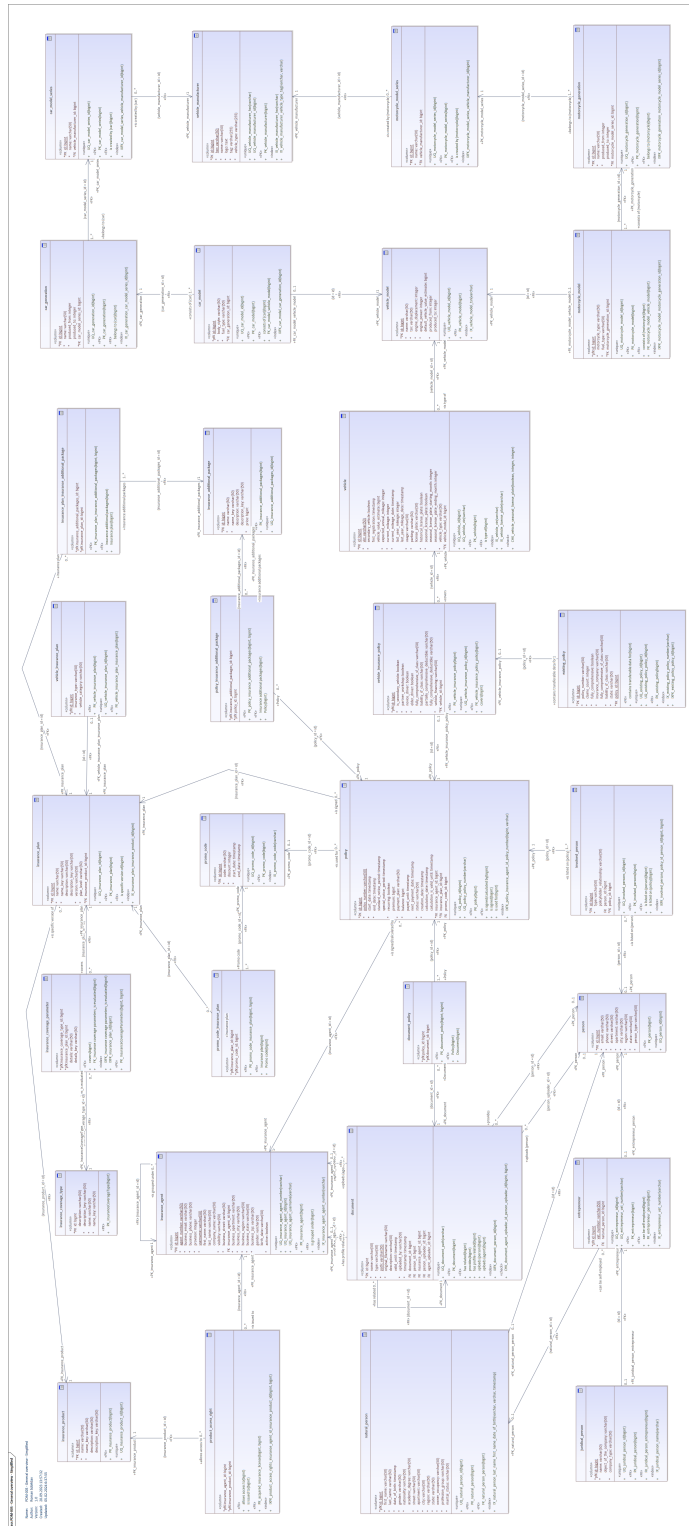
Document management system Systém, který bude využíván pro správu dokumentů.

Operational data storage Slouží jakožto zdroj informací pro klientskou zónu, která bude nasazována na stejný Liferay portál.

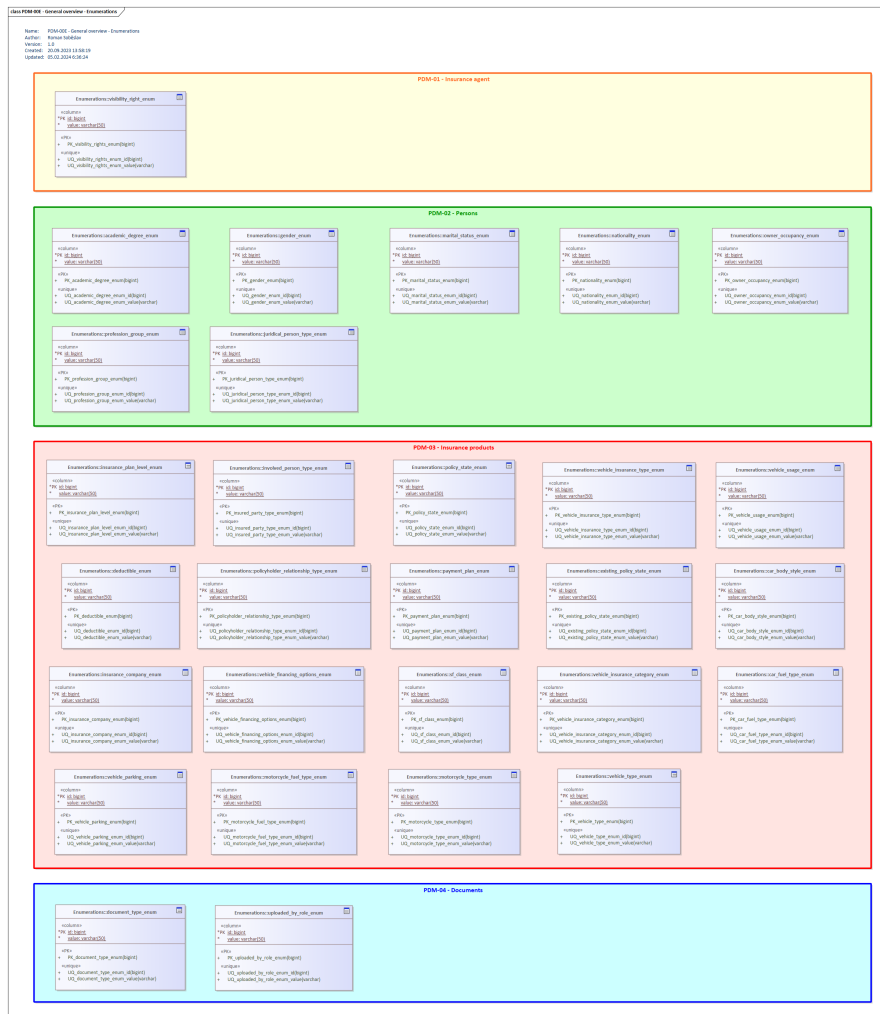
Output management system Systém, který je využíván pro generování smluv v PDF.

Server pro poskytování informací o vozidlech Server, který obsahuje aplikace pro ověření existence vozidla, ověření škodní historie vozidla a získávání informací o vozidlech.

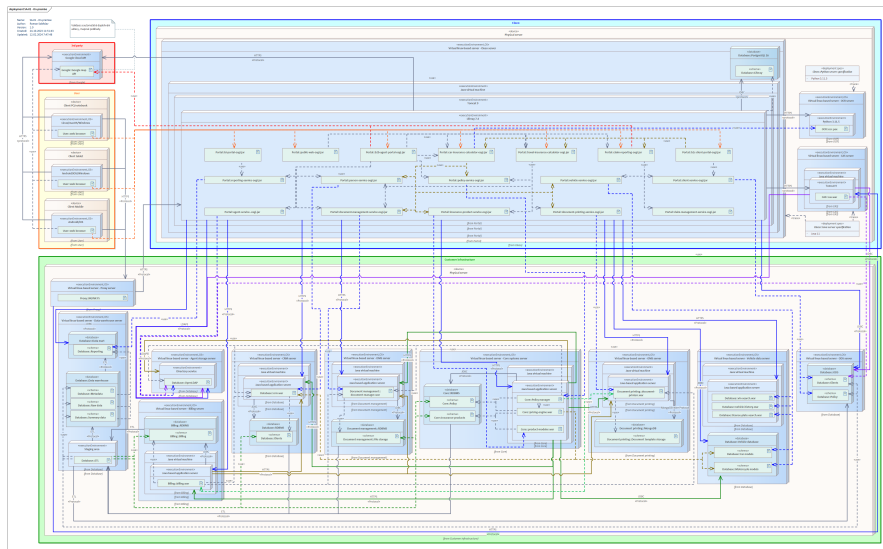
Kompletní řešení Cleo se bude nasazovat formou modulů na jeden Liferay server. Servisní vrstva obsahuje specifickou implementaci pro klienta, aby byla schopna komunikovat s již existující infrastrukturou. Prezenční vrstva, která obsahuje před-připravené univerzální moduly, využívá servisní vrstvu pro komunikaci s klientskou infrastrukturou tak, aby bylo možné získat požadovaná data. Nasazená PostgreSQL databáze slouží jakožto provozní databáze samotného Liferay portálu. Kromě samotného Liferay portálu se předpokládá i nasazení OCR technologie, která umožní před-vyplňovat formuláře na základě informací, které jsou extrahovány z povinně přikládaných dokumentů, jako je občanský a řidičský průkaz, nebo technický průkaz vozidla.



■ Obrázek 2.15 PDM - fyzický datový model



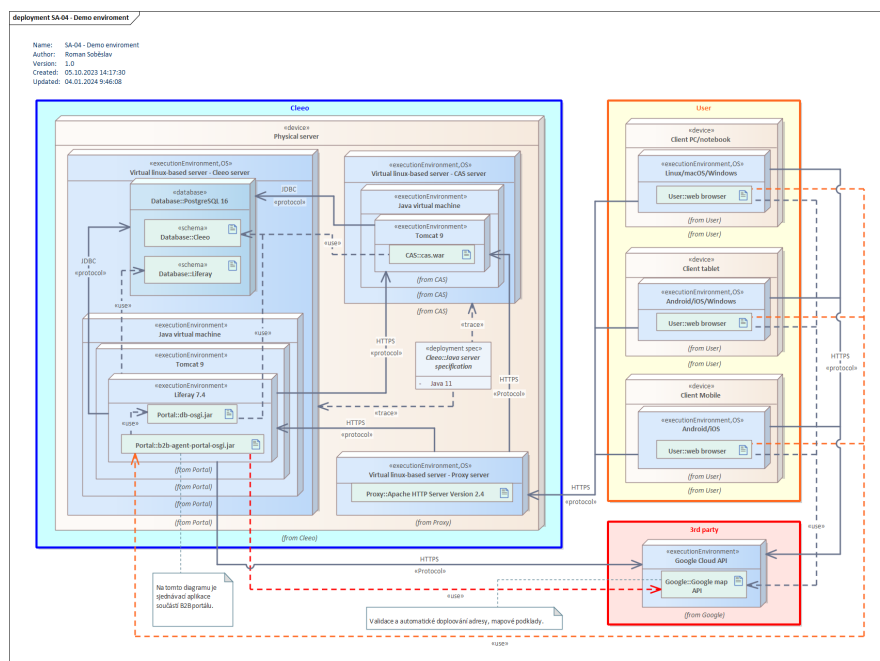
■ Obrázek 2.16 PDM - číselníky



■ **Obrázek 2.17** SA - návrh architektury u zákazníka

Posledním zamýšleným systémem pro případné nasazení je Centrální autentizační systém pro implementaci Single-Sign-On řešení.

Varianta pro provoz na Demo prostředí vychází z výše popsaného návrhu. Hlavním rozdílem je nahrazení infrastruktury zákazníka databází, se kterou bude komunikovat servisní vrstva a simulovat tak interakci s reálnými systémy. Pokud je třeba implementovat složitější byznys logiku, jako je například generování PDF nebo ocenění kalkulace, servisní vrstva bude tuto činnost jen simulovat a bude vracet statické hodnoty.



■ Obrázek 2.18 SA - plánovaný provoz na demo prostředí

Implementace

Tato kapitola pojednává o vývoji navrženého portálu z předchozí kapitoly. Nejprve definuji nástroje, které použiji pro vývoj a způsob verzování kódu. Následně popíši princip vývoje jednotlivých částí s ukázkami.

3.1 Vývojové prostředí

Pro vývoj servisní vrstvy jsem použil *IntelliJ Idea*. Ostatní části jsem vyvíjel s pomocí *Visual Studio Code*. Pro nasazení portálu jsem použil nástroj Docker, který popisuji níže.

3.2 Verzování kódu

Verzování kódu probíhalo pomocí nástroje Git na firemního úložiště GitLab. Jelikož se jedná o neveřejné úložiště, přiložil jsem v příloze otisk repositáře.

3.3 Docker

Docker je open-source nástroj pro kontejnerizaci aplikací. Základem jsou izolované kontejnery, které běží v Linuxovém prostředí. Základní jednotkou je Docker image, který definuje obsah kontejneru. Definuje operační systém, aplikační prostředí a potřebné závislosti pro funkčnost aplikace. Image slouží jako předpis pomocí kterého se vytváří spustitelné kontejnery. Pro vytvoření image složí *Dockerfile*. Při definování image můžeme uvést jiný image, na kterém nově vzniklý závisí. Pro sdílení Docker image slouží Dockerhub.

Spustitelná instance Docker image se označuje kontejner. Jedná se o izolovaný proces, který běží na hostitelském operačním systému a sdílí jádro s ostatními kontejnery. Každý kontejner má svůj vlastní souborový systém, síťové rozhraní a procesy.

Ke správě kontejnerů slouží Docker engine, který se skrze daemona stará o celý životní cyklus kontejneru.

Pokud potřebujeme nasazovat více na sobě závislých kontejnerů, použijeme *Docker compose*. Jedná se o nástroj pro orchestraci více kontejnerů. Skládá se z definic služeb (kontejnerů), úložiště (volume) a případně sítě (network).

Zde je přiložená ukázka *compose.yaml*, která definuje používaný stack v rámci této práce (celý compose je v příloze, jelikož bylo třeba zkrátit některé řádky, které přetékal).

```
services:  
  database:
```

```

build: ./db
container_name: database
user: ${DB_USER}
environment:
  - POSTGRES_PASSWORD=${DB_PASSWORD}
  - POSTGRES_DB=${DB_DATABASE}
ports:
  - ${DB_PORT}:5432
volumes:
  - pg_volume:/var/lib/postgresql/data
restart: ${RESTART_STRATEGY}
healthcheck:
  test: ["CMD", "pg_isready"]
  interval: 10s
  timeout: 5s
  retries: 5
pgadmin:
build: ./pgadmin4
container_name: pgadmin4
depends_on:
  - database
environment:
  - PGADMIN_DEFAULT_EMAIL=${PGADMIN_EMAIL}
  - PGADMIN_DEFAULT_PASSWORD=${PGADMIN_PASSWORD}
ports:
  - ${PGADMIN_PORT}:80
volumes:
  - pgadmin_operational_volume:/pgadmin4
  - pgadmin_storage_volume:/var/lib/pgadmin
restart: ${RESTART_STRATEGY}
portal:
image: liferay/portal:7.4.3.100-ga100
container_name: portal
depends_on:
  database:
    condition: service_healthy
environment:
  # Changes JDK version from 1.8 to 11.
  - JAVA_VERSION=zulu11
ports:
  - ${LR_PORT}:8080
volumes:
  - ${LR_DEPLOY}:/mnt/liferay/deploy
deploy:
  resources:
    limits:
      memory: 8g
  restart: ${RESTART_STRATEGY}
volumes:
  pg_volume:
  pgadmin_operational_volume:
  pgadmin_storage_volume:

```

Docker compose podporuje i proměnné prostředí, které lze definovat předem v operačním systému, nebo přiložením souboru s názvem *.env* vedle *compose.yaml*.

Pro spuštění slouží příkaz *Docker compose*, který má k dispozici přepínač *-d* pro spouštění určitých služeb. Jedinou podmínkou je spuštění všech kontejnerů, na kterých spouštěná služba závisí.

3.4 Vývoj portálu pomocí Liferay 7.4

V této sekci se zabývám popisem postupu vývoje jednotlivých modulů pro Liferay 7.4.

3.4.1 Liferay workspace

Prvním krokem je vytvoření Liferay workspace pomocí archetypu. K tomu jsem použil plugin Liferay pro IntelliJ Idea. Jedná se o společný projekt který seskupuje veškeré moduly, konfigurace a témata pro vyvíjené funkcionality pro jednu instanci portálu. Worksapce v sobě také obsahuje Docker image pro spuštění lokální instance pro vývoj.

Složka *configs* obsahuje veškeré konfigurace pro jednotlivá prostředí. Samotné moduly jsou vytvářeny ve složce *modules*.

Pro účely této práce jsem použil Gradle verzi prostředí, která je lépe optimalizovaná než Maven alternativa. Některé moduly totiž ignorovali nastavení a bylo nutné vše ručně dopravit. Při použití Gradle tyto nedostatky odpadly.

3.4.2 Tvorba servisní vrstvy

Pro tvorbu servisní vrstvy jsem použil archetyp *service builder*. Jedná se o template pro tvorbu servisních vrstev, kdy se nadefinují entity pomocí souboru *service.xml*. Zde je přiložená ukázka pro definování zprostředkovatele:

```
<?xml version="1.0"?>
<!DOCTYPE service-builder PUBLIC "-//Liferay//DTD Service Builder 7.4.0//EN" "http://www.liferay.com"
<service-builder dependency-injector="ds" package-path="com.cleverbee.cleleo.agent">
  <author>RSobeslav</author>
  <namespace>Cleleo</namespace>
  <entity data-source="cleleoDataSource" local-service="true" name="Agent" remote-service="false" t
    <!-- PK fields -->
    <column name="id" primary="true" type="long" db-name="id" />

    <!-- Other fields -->
    <column name="agentNumber" type="String" db-name="agent_number" />
    <column name="email" type="String" db-name="business_email" />
    <column name="phone" type="String" db-name="business_phone" />
    <column name="street" type="String" db-name="business_street" />
    <column name="apartment" type="String" db-name="business_apartment" />
    <column name="city" type="String" db-name="business_city" />
    <column name="region" type="String" db-name="business_region" />
    <column name="state" type="String" db-name="business_state" />
    <column name="zip" type="String" db-name="business_zip" />
    <column name="gender" type="String" db-name="gender" />
    <column name="birthDate" type="Date" db-name="birth_date" />
    <column name="provision" type="int" db-name="provision" />
```

```
<column name="username" type="String" db-name="username" />
<column name="firstName" type="String" db-name="first_name" />
<column name="lastName" type="String" db-name="last_name" />
<column name="companyName" type="String" db-name="company_name" />
<column name="visibility" type="String" db-name="visibility" />
<column name="active" type="boolean" db-name="active" />
<column name="parentId" type="long" db-name="insurance_agent_id" />
</entity>
</service-builder>
```

Zde konkrétně využívám vlastní datasource pro připojení k vlastní databázi. Spuštění úkolu *generate service* dojde k vygenerování implementace pro perzistentní vrstvu a komunikaci s databází. Dále je vytvořena lokální servisní vrstva, která definuje funkce volatelné přímo z OSGi prostředí.

Každá služba má svoji service část, která implementuje veškerou logiku a API část jakožto rozhraní.

U definice entity lze také zvolit, zda má mít remote service, která lze zavolat pomocí REST rozhraní.

Ve svém projektu jsem takto vytvořil službu pro práci s agenty, klienty a smlouvami.

3.4.3 Tvorba pravidelných úkolů

Pravidelné úkoly se vytváří jako Java třídy implementující rozhraní *BaseDispatchTaskExecutor*. Portál takovou třídu po nasazení zaregistruje a volá následně její implementaci funkce *doExecute*. Ve své práci jsem definoval dva úkoly – import agentů a mazání odebraných.

3.4.4 Tvorba prezenční vrstvy

Prezenční vrstvu jsem implementoval pomocí nástroje yeoman a liferay-js-generátor. Ten vytvoří základní Angular projekt, který lze sestavit a nasadit jako JAR.

Vzniklý projekt bohužel není plnohodnotný Angular projekt, jelikož implementuje upravené rozhraní pro dynamické zavádění za běhu. Nelze tak používat Angular CLI a spouštět vývojářský server.

Ve své práci jsem vytvořil modul pro listování osob, hledání, vytváření osob a zobrazování informací o osobě.



Kapitola 4

Testování

Tato kapitola shrnuje testování vzniklého kódu.

Jelikož se jedná o vývoj aplikace v OSGi prostředí, psaní jednotkových testů není tak jednoduchý, kvůli závislosti na nasazených modulech, které jsou odchyťvány za běhu z aplikačního prostředí. Pro psaní testů je tak třeba používat mockovacích frameworků. Alternativním řešením je provádění integrační end-to-end testů. Pro testování servisní vrstvy jsem používal Postman[25], přes který jsem si ji provolával a získával tak data z databáze. Jelikož se API dosti měnilo, nevytvářel jsem testovací scénáře, přestože je nástroj podporuje.

Testování uživatelského rozhraní probíhalo ručně.

Jelikož jsem v daném časovém rozmezí nedokončil vývoj portálu, neprovedl jsem uživatelské testování. Alternativou by bylo otestování vzniklého drátěného modelu. Osobně si myslím, že by výsledky byly nesměrodatné. Důvodem je vytvoření interaktivního modelu pomocí nástroje Figma, který však neumožňuje vkládat reálná data. Výsledkem by bylo jen zběžné proklikání, které by však neodhalilo slabiny návrhu.

Kapitola 5

Závěr

Cílem této práce je provést analýzu, návrh a implementaci portálu pro zprostředkovatele.

V první kapitole jsem se věnoval seznámení s problematikou sjednávání pojištění (převážně pojištění vozidel) na německém trhu. Provedl jsem analýzu sjednávání aplikací na německém trhu. Na základě těchto poznatků jsem navrhl byznys objektový model, který popisoval datový model extrahovaný z analytického výstupu.

Ve druhé kapitole jsem se věnoval návrhu uživatelského rozhraní pro byznys objektový model. Poté jsem zvolil technologie, ve kterých jsem následně implementoval své řešení. Následně jsem omezil rozsah projektu pro finální implementaci a navrhl jsem konkrétní logický datový model. Ten jsem následně konvertoval na fyzický datový model pro databázový systém PostgreSQL.

Ve třetí kapitole jsem se věnoval procesu vývoje jednotlivých částí na platformě Liferay 7.4. Popsal jsem jak se vytváří workspace, servisní vrstva a portlety.

Poslední čtvrtá kapitola se věnovala testování, které však z časových důvodů neproběhla. Nestihl jsem totiž implementovat navržený portál v dostatečném rozsahu, který by byl vhodný pro testování. Příložené zdrojové kódy patří třem aktivně vyvíjeným větvím, kdy větev *import agent* je již hotová a čeká na zaměření. Větev *person* obsahuje rozpracovanou větev pro práci s klienty, která je před dokončením. Poslední příložená větev *policy* obsahuje implementovanou servisní vrstvu pro obecnou práci s pojistnými smlouvami.

Důvodem nestihnutí dokončení implementace bylo špatné rozvržení časového plánu, kdy většina času připadla na analýzu a návrh obecného portálu pro zprostředkovatele, který bude sloužit jako základ pro implementaci řešení pro konkrétního klienta na německém trhu.

Co se týká dalších plánů, tak je dokončení portálu dle definovaného rozsahu této práce. Následně je nutné vytvořit téma, které sjednotí design aplikace. Následně dojde k uživatelskému testování a zahrnutí do balíčku Cleeo.

..... Příloha A

Administrátorská příručka

Kompletní administrátorská příručka je součástí každé větve ve formě README.md souboru.

Bibliografie

1. PROFESSIONAL SERVICES CORPORATION PTY LTD. *Insurance Explained: The Difference Between ‘Named Insured’, ‘Additional Insured’ and ‘Named Additional Insured’* [online]. [B.r.]. Dostupné také z: <https://www.pscinsurance.com.au/insurance-explained-difference-named-insured-additional-insured-named-additional-insured/>.
2. STEFAN TIRSCHLER. *Quote* [online]. 2023. Dostupné také z: <https://www.squareone.ca/resource-centres/insurance-glossary/quote>.
3. ČESKÁ ASOCIACE POJIŠŤOVEN. *POJISTNÁ SMLOUVA* [online]. [B.r.]. Dostupné také z: <https://www.cap.cz/pojistna-smlouva>.
4. NICOLAS BOULIANE. *How to choose car insurance in Germany* [online]. 2023. Dostupné také z: <https://allaboutberlin.com/guides/car-insurance-germany>.
5. GARY BUSWELL. *Car insurance in Germany* [online]. 2023. Dostupné také z: <https://www.expatica.com/de/finance/insurance/car-insurance-in-germany-246345/>.
6. ALLIANZ DIRECT VERSICHERUNGS-AG. *Allianz Direct* [online]. [B.r.]. Dostupné také z: <https://www.allianzdirect.de/>.
7. DA DIREKT. *DA direkt* [online]. [B.r.]. Dostupné také z: <https://www.da-direkt.de/>.
8. HDI AG. *HDI* [online]. [B.r.]. Dostupné také z: <https://www.hdi.de/>.
9. HELVETIA INSURANCE. *Homepage* [online]. [B.r.]. Dostupné také z: <https://www.helvetia.com/de/web/de/privatkunden.html>.
10. AXA GROUP AG. *Homepage* [online]. [B.r.]. Dostupné také z: <https://www.axa.de/>.
11. VERIVOX. *Homepage* [online]. [B.r.]. Dostupné také z: <https://www.verivox.de/>.
12. TARIFCHECK24 GMBH. *Kfz-Versicherung Vergleich — so finden Sie die passende Autoversicherung* [online]. [B.r.]. Dostupné také z: <https://www.tarifcheck.de/kfz-versicherung/>.
13. BUNDESMINISTERIUM DER JUSTIZ. *Gesetz über die Pflichtversicherung für Kraftfahrzeughalter (Pflichtversicherungsgesetz) § 4* [online]. [B.r.]. Dostupné také z: https://www.gesetze-im-internet.de/pflvg/_4.html.
14. NICOLAS BOULIANE. *What is an eVB number?* [Online]. 2023. Dostupné také z: <https://allaboutberlin.com/glossary/eVB-Nummer>.
15. AXA INSURANCE LTD. *Key points at a glance* [online]. [B.r.]. Dostupné také z: <https://www.axa.ch/en/private-customers/offers/vehicle-travel/car-insurance.html>.

16. THOMAS WACHSMUTH. *Seasonal license plates: everything you need to know!* [Online]. 2022. Dostupné také z: <https://www.tuningblog.eu/en/kategorien/tuev-gesetze-verbote/saisonkennzeichen-infos-411206/>.
17. INFO-BEIHILFE.DE. *Der umfassende PKV Vergleich* [online]. [B.r.]. Dostupné také z: <https://info-beihilfe.de/beamte-und-beamtenaerter/beamtenlexikon/beamter-auf-lebenszeit/>.
18. NICOLAS BOULIANE. *The tax number (Steuernummer)* [online]. 2023. Dostupné také z: <https://allaboutberlin.com/guides/german-tax-id-steuernummer#the-vat-number-umsatzsteuer-id>.
19. GOOGLE LLC. *Material Design* [online]. [B.r.]. Dostupné také z: <https://m3.material.io/>.
20. LIFERAY INC. *Homepage* [online]. [B.r.]. Dostupné také z: <https://www.liferay.com/>.
21. DEEPAK GOTHE. *Understanding the Java Portlet Specification 2.0 (JSR 286): Part 1, Overview and Coordination Between Portlets* [online]. 2010. Dostupné také z: <https://www.oracle.com/java/technologies/portlet-specification1-v20.html>.
22. ORACLE CORPORATION. *What is a Portlet?* [Online]. 2010. Dostupné také z: <https://docs.oracle.com/cd/E19636-01/819-2580/adger/index.html>.
23. MICROSOFT. *SharePoint* [online]. [B.r.]. Dostupné také z: <https://www.microsoft.com/cs-cz/microsoft-365/sharepoint/collaboration>.
24. MATTHIAS GELBMANN, PAUL ANDLINGER. *PostgreSQL is the DBMS of the Year 2023* [online]. 2024. Dostupné také z: https://db-engines.com/en/blog_post/106.
25. POSTMAN, INC. *Homepage* [online]. [B.r.]. Dostupné také z: <https://www.postman.com/>.

Obsah příloh

README.MD.....	stručný popis obsahu média
obrázky	adresář s grafickým výstupem práce
diagramy	adresář s grafickým výstupem práce
BOM	adresář s byznys objektovým modelem
LDM	adresář s logickým datovým modelem
PDM	adresář s fyzickým datovým modelem
SA	adresář s návrhem architektury
wireframe	adresář se vzniklým drátěným modelem
Cleeo - Insurance agent portal.pdf ..	kompletní drátěný model ve formátu PDF
Components	adresář s jednotlivými částmi drátěného modelu
src	
branches	větvě se zdrojovými kódy implementace
Components	implementace importu agentů
Components	implementace pro práci s osobami
Components	implementace servisní vrstvy pro práci se smlouvami
thesis	zdrojová forma práce ve formátu L ^A T _E X
text	text práce
thesis.pdf	text práce ve formátu PDF