

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Univerzální modul pro řešení dopravy pro e-shop

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd
Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Lucie TAUCHENOVÁ**
Osobní číslo: **A16B0151P**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informatika**
Téma práce: **Univerzální modul pro řešení dopravy pro e-shop**
Zadávající katedra: **Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Seznamte se s problematikou dopravy zboží v ČR a prozkoumejte způsoby komunikace mezi e-shopy a nejvyužívanějšími přepravními společnostmi.
2. Proveďte analýzu jednotlivých možností dopravy v ČR a proveďte rešerši vybraných dopravců s ohledem na závislost ceny dopravy na rychlosti doručení.
3. Navrhněte modul, který zajistí komunikaci s různými systémy dopravních společností a který jim bude předávat data o zásilkách.
4. Navržený modul implementujte ve vhodném programovacím jazyce, jehož volbu zdůvodněte.
5. Realizovaný modul důkladně otestujte s využitím unit testů, testovacích scénářů a s vybraným testovacím e-shopem.
6. Kriticky zhodnoťte klady a zápory vytvořeného softwaru.

Rozsah bakalářské práce: **doporuč. 30 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce.

Vedoucí bakalářské práce: **Ing. Martin Dostal, Ph.D.**
Katedra informatiky a výpočetní techniky

Datum zadání bakalářské práce: **7. října 2019**
Termín odevzdání bakalářské práce: **7. května 2020**

Radová

Doc. Dr. Ing. Vlasta Radová
děkanka



Brada

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů.

V bakalářské práci jsou použity názvy programových produktů, firem apod., které mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

V Plzni dne 7. května 2020

Lucie Tauchenová

Poděkování

Děkuji panu Ing. Martinovi Dostalovi, Ph.D. za cenné připomínky a odborné vedení práce. Děkuji zadavateli tématu firmě RTsoft s.r.o. za ochotu a zkušenosti nabyté během realizace práce. Dále děkuji panu Ing. Tomáši Hudovi za poskytnutí technických konzultací.

Abstract

The subject of this thesis called is a design and implementation of a universal module that enables to select the appropriate services for sending a shipment from an e-shop and provides communication between the e-shop information system and the carrier. This communication consists in obtaining data from the carrier, and in processing orders and their subsequent transmission to the carrier. The prerequisite for the implementation of the module was a comparison of shipment options in the Czech Republic and an analysis of the methods of communication between e-shops and selected carriers. The outputs of both analyzes were then applied in the resultant module.

Abstrakt

Tato práce se zabývá návrhem a implementací univerzálního modulu, který umožní vybrat vhodné služby pro odeslání zásilky z e-shopu a zajistí komunikaci mezi informačním systémem e-shopu a dopravcem. Tato komunikace spočívá v získání dat od dopravce, a ve zpracování objednávek a jejich následném předání dopravci. Předpokladem pro realizaci modulu bylo provedení srovnání možností přepravy zásilek v České republice a analýza způsobů komunikace mezi e-shopy a vybranými dopravci. Výstupy obou analýz byly potom aplikovány ve výsledném modulu.

Obsah

1	Úvod	1
2	Použité technologie a přístupy	2
2.1	Programovací jazyk PHP	2
2.2	Nette framework	3
2.2.1	Definice frameworku	3
2.3	Nette a MVC architektura	4
2.3.1	Výhody a nevýhody používání frameworku Nette . .	6
2.4	Protokoly pro přístup k API	9
2.4.1	Protokol SOAP	9
2.4.2	REST API	9
2.4.3	Srovnání protokolu SOAP a REST API	10
3	Analýza a srovnání možností dopravy zásilek v ČR	11
3.1	Česká pošta s.p.	13
3.2	Zásilkovna	15
3.3	DPD	17
3.4	PPL	18
3.5	Srovnání služeb	21
4	Analýza a srovnání způsobů komunikace mezi e-shopy a vybranými dopravci	27
4.1	Webové služby České pošty	28
4.2	Webové služby Zásilkovny	31
4.3	Webové služby DPD	34
4.4	Webové služby PPL	37
4.5	Srovnání webových služeb	37
5	Návrh a implementace modulu	40
5.1	Návrh funkcionality	40
5.1.1	Konfigurace modulu	40
5.1.2	Modul v procesu objednávky	42
5.2	Návrh rozhraní	44
5.2.1	Návrh uživatelského rozhraní	44
5.2.2	Návrh aplikačního rozhraní	44
5.3	Návrh databázové struktury	46

5.4	Implementace modulu	47
5.4.1	Rozvržení vrstvy modelu	48
5.4.2	Formulářové komponenty	51
5.4.3	Integrační část modulu	53
6	Testování	54
6.1	Testování pomocí jednotkových testů	54
6.2	Ověření pomocí testovacího e-shopu	55
6.3	Použití testovacích scénářů	57
6.4	Shrnutí testování	57
6.5	Návrhy na zlepšení	58
7	Závěr	59
	Literatura	65
A	Obsah CD	68
B	Uživatelská a programátorská příručka	69
C	Diagram tříd komponent	72
D	Případy užití modulu	73
E	Obrazovky testovacího e-shopu	82

1 Úvod

Cílem této práce je navrhnout a implementovat univerzální modul, který bude řešit problematiku dopravy pro e-shopy. Jeho úkolem bude vybrat služby, které jsou vhodné pro odeslání zásilky z e-shopu. Dalším úkolem modulu bude zajistit komunikaci mezi informačním systémem e-shopu a systémy dopravců. Základ této komunikace bude spočívat především v získání dat od dopravců a zpracování dat o zásilkách z e-shopu.

Práce je rozdělena do dvou částí. První, teoretickou část práce tvoří druhá kapitola nazvaná Použité technologie a přístupy. V této kapitole budou prezentovány hlavní technologie, které budou použity pro realizaci modulu. Ty tvoří programovací jazyk PHP a framework Nette. V rámci popisu bude rovněž vysvětlen princip architektury MVC a její hlavní rozdíl oproti architektuře MVP, kterou využívá právě vybraný framework Nette. Tato kapitola bude uzavřena charakteristikou dvou nejběžnějších přístupů k API – protokolu SOAP a REST API.

Druhá, již empirická část práce bude otevřena kapitolou třetí s názvem Analýza a srovnání možností dopravy zásilek v ČR. V této kapitole budou představeny jednotlivé přepravní společnosti a jejich služby. Služby zde budou stručně popsány a vzájemně porovnány, přičemž mezi srovnávací kritéria budou patřit zejména rozmanitost a cena služeb a rychlost doručení. Pro účely této analýzy byly vybrány čtyři přepravní společnosti využívané českými e-shopy nejčastěji. Informace potřebné k analýze služeb budou čerpány převážně z jejich oficiálních webových prezentací.

Čtvrtá kapitola s názvem Analýza a srovnání způsobů komunikace mezi e-shopy a vybranými dopravci se zaměří na webové služby a jejich rozhraní, pomocí nichž e-shopy získávají informace týkající se výdejních míst, dostupných služeb v regionu, apod. Analýza se mimo jiné zaměří na možnost předání dat o zásilce z e-shopu přímo dopravci. Informace o webových službách budou čerpány taktéž z elektronických zdrojů dopravců.

Téma páté kapitoly je shrnuto v jejím názvu Návrh a implementace modulu. Návrh modulu bude rozdělen na uživatelské rozhraní, aplikační rozhraní a databázovou strukturu. Druhá polovina kapitoly se bude věnovat rozvržení vrstev v implementovaném modulu, dále představí nejdůležitější komponenty a popíše integrační část modulu.

Poslední, šestá kapitola přiblíží techniky testování modulu, mezi něž patří jednotkové testy a testovací scénáře. V průběhu kapitoly bude také ukázán přínos testování modulu pomocí demo aplikace.

2 Použité technologie a přístupy

Cílem této kapitoly je popsat technologie, které budou použity pro realizaci modulu řešícího dopravu. Primárně použitými technologiemi budou programovací jazyk PHP a Nette framework. Neméně důležitým úkolem kapitoly bude přiblížit přístupy, kterými bude realizace řízena. Mezi tyto přístupy patří především MVC architektura aplikace.

2.1 Programovací jazyk PHP

Jazyk PHP je interpretovaným skriptovacím programovacím jazykem, který se používá zejména pro tvorbu webových aplikací. Skripty psané v PHP jsou prováděny na straně serveru a uživateli je odeslán výsledek jejich činnosti v podobě HTML výstupu, jehož zobrazení pak řeší webový prohlížeč.

První verzi PHP zveřejnil v roce 1995 Rasmus Lerdorf. Název PHP byl původně zkratkou pro Personal Home Page, až později se význam změnil na PHP: Hypertext Preprocessor. Cílem PHP bylo minimalizovat množství kódu, kterého bylo zapotřebí k dosažení požadovaného výsledku. Tato minimalizace vedla k tomu, že se z PHP stal jazyk, který byl tzv. HTML-centric. To znamená, že se PHP kód začal vnořovat do HTML kódu. Vnoření PHP kódu do HTML je možné dodnes. Časem navíc přibyla možnost extrahovat PHP skript do samostatného souboru. V obou případech užití platí, že PHP kód musí být uvozen startovací a ukončovací značkou ve tvaru: `<?php ?>`. Zkrácená verze zápisu značek se zapisuje ve tvaru: `<? ?>`. Zkrácená verze zápisu musí být ale nejdříve povolena na straně serveru. [9]

V kódu č. 2.1 je zobrazena ukázka vnoření PHP skriptu do HTML kódu.

```
1 | <html>
2 |   <body>
3 |     <p><?php print $jmeno; ?></p>
4 |   </body>
5 | </html>
```

Kód 2.1: Vnoření PHP skriptu do HTML kódu

Verze PHP 2 byla první verzí PHP, která se široce rozšířila mezi programátory. Od verze PHP 3 se tento jazyk stal navíc objektově orientovaným.

V současnosti se v softwarech využívá verze PHP 5, která je postupně aktualizována na PHP 7.

Výhodami PHP jsou v první řadě jeho rychlost a nezávislost na platformě. PHP funguje na operačních systémech Windows, Unix i Linux. PHP je jedním z nejrychlejších skriptovacích jazyků, kterému dnes konkuruje zejména jazyk ASP. Další výhodou je široké využití jazyka. Pomocí PHP lze pracovat se soubory, emaily, napojit se na různé typy databází, vytvářet PDF či upravovat grafiku. PHP také podporuje nejdůležitější internetové protokoly jako např. HTTP, HTTPS, a další. PHP tvoří tzv. triádu společně s webovým serverem Apache a databází MySQL. Triáda je trojice nejvyužívanějších programů pro tvorbu webových stránek. [9]

Nevýhodou PHP může být slabá typovost jazyka, kdy u proměnných není definován žádný datový typ. To znamená, že neprobíhá typová kontrola hodnot, které jsou přiřazovány do proměnných. Další nevýhodou spojenou s PHP představuje závislost výkonu serveru na počtu připojených uživatelů. Zatížení serveru, na kterém je spuštěna webová aplikace, stoupá s počtem připojených uživatelů. Výkon serveru se tak při velkém počtu připojených uživatelů může rapidně snížit.

I přes značné nevýhody jazyka PHP je tento jazyk v současné době využíván v 79 % webových aplikací a patří tak k nejoblíbenějším programovacím jazykům v této oblasti. Daleko za ním je s necelými 11 % ASP.NET a se 4 % jazyk Java. [16]

2.2 Nette framework

Nette framework je český open-source projekt, který v roce 2008 založil český vývojář David Grudl. Nette framework je v ČR velmi rozšířený. Fungují na něm velké projekty jako např. Zásilkovna, Knihy Dobrovský, Slevomat, ČSFD a také velké množství menších webů a e-shopů. Celosvětově patří společně s frameworky Symfony a Laravel k nejoblíbenějším PHP frameworkům. [18]

2.2.1 Definice frameworku

Framework - česky programový rámec webové aplikace - představuje komplexní nástroj pro rychlejší a bezpečnější vytváření webové aplikace. Framework obsahuje různé knihovny, připravené komponenty, ale i užitečné nástroje, které kontrolují metodiku vývoje softwaru a usnadňují ladění jeho chyb. Knihovny, které obsahuje, slouží pro práci s prezentační i datovou logikou aplikace, pro zpracování a ukládání dat nebo pro různé typy služeb,

např. autorizační služby. [9]

Právě knihovny tvoří primární důvod, proč je výhodné při vývoji rozsáhlého softwaru využít nějaký framework. Vezme-li se v úvahu programovací jazyk PHP, tak navzdory velkému počtu vestavěných funkcí se při vývoji webové aplikace bude muset vývojář potýkat s mezerami ve standardní knihovně jazyka. Pro běžné činnosti, jako jsou např. obsluha formulářů, vytváření databázové třídy či stránkování objemné tabulky, neexistují žádné knihovní funkce. Obsluhu musí znovu a znovu řešit vývojář aplikace sám, což zabírá zbytečný čas navíc. Má-li framework tyto rutinní záležitosti ošetřené v podobě jednoduchých knihoven, vývojář nemusí stále dokola tak říkajíc vynalézat kolo a může se o to důkladněji věnovat funkčnosti softwaru. Aplikace bude s frameworkem navíc mnohem udržitelnější.

I přes to, že framework je hlavně soubor knihoven, od knihoven jako takových se liší. Hlavním rozdílem mezi frameworkem a knihovnou je ten, že knihovna je volána z kódu, zatímco framework volá kód. Jinými slovy, framework je kostrou aplikace, do které jsou postupně přidány vlastnosti (funkce). Framework může sloužit jako platforma pro vyvíjení modulů. Oproti tomu knihovna poskytuje již funkční moduly připojitelné k vytvořené platformě. [14]

Architektura aplikace daná frameworkem se často obecně nazývá inverzí kontroly. Směr toku dat je totiž opačný než u procedurálního programování. Jiný název hovoří o principu Hollywoodu: Nevolejte nám, my zavoláme. Tento princip uplatňovaný ve filmovém průmyslu odkazuje na to, že vývojářův kód je volaný třetí stranou. Hlavním důvodem na pozadí je vytváření vysoko-úrovňových komponent, které jsou méně závislé na svých podsystémech. Tyto vyšší komponenty pak mohou předat kontrolu těm nižším, které si samy za sebe rozhodnou, jak a kdy budou odpovídat. Dobrým příkladem může být rozdíl mezi programem běžícím v příkazovém řádku, který se pozastaví a následně požádá o vstup od uživatele, a programem běžícím v okně s grafickým uživatelským rozhraním, ve kterém uživatel klikne na libovolné tlačítko a vyvolá reakci okenního manažera, jenž sám zavolá obsluhu programu. [14]

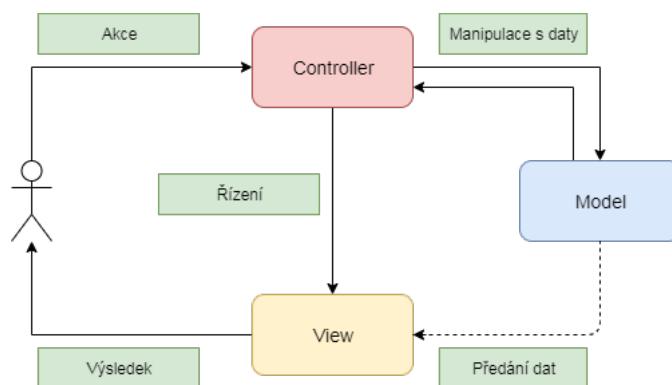
2.3 Nette a MVC architektura

Nette patří do skupiny frameworků postavených na architektuře s MVC přístupem. MVC je zkratkou sestávající se z počátečních písmen vrstev, ze kterých se systém s touto architekturou skládá. Vrstvami jsou Model, View a Controller. Pojem MVC byl poprvé zmíněn v práci z roku 1979, která

pojednává o webových aplikacích v programovacím jazyce SmallTalk. [11]

Hlavní ideou MVC je rozdělit aplikaci do tří základních vrstev, z nichž každá má na starosti jiný typ úloh. Těmi jsou logika (Model), výstup (View) a řízení (Controller). Oddělení kódu obsluhy od aplikační logiky a kódu vykreslujícího data aplikaci výrazně zpřehledňuje a ulehčuje její budoucí vývoj. Rozdělení dále umožňuje jednoduchou výměnu celých částí kódu programu a zvyšuje jeho znovupoužitelnost.

Princip návrhového vzoru MVC ilustruje obrázek č. 2.1.



Obrázek 2.1: Diagram popisující architekturu MVC

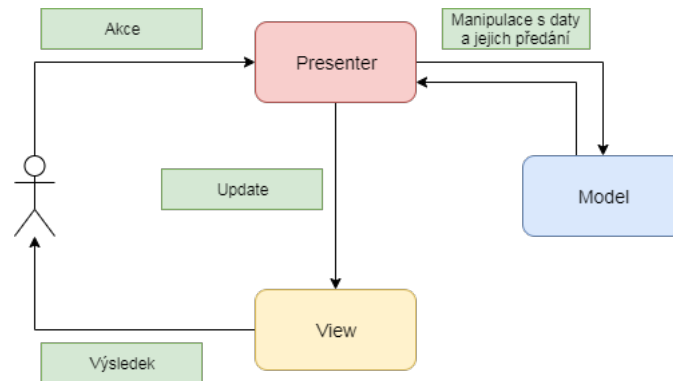
Model (datová vrstva) představuje hlavní funkční vrstvu aplikace. Měl by obsahovat veškerou aplikační logiku a manipulaci s daty. Model musí reprezentovat strukturu dat se všemi vztahy a závislostmi. Je také jedinou z vrstev, která využívá trvalého úložiště dat a obsluhuje veškerá spojení s databází. Model by měl upozornit View ve chvíli, kdy dojde ke změně vnitřních dat na data, která jsou potřeba ve View překreslit. [11]

View (prezentační vrstva) je pohled. Je to vrstva, která se stará o vykreslování výsledku požadavku. Nejdůležitějším pravidlem je, že View nikdy nemění data aplikace, pouze je zobrazuje. K zobrazování obvykle využívá některý ze šablonovacích systémů.

Controller (řídící vrstva) je vstupní branou aplikace. Controller zajišťuje interakci s uživatelem a stará se o další funkce. Controller by měl být jednoduchý a řídit pouze tu část, která používá funkce Modelu a View.

V Nette se vrstva Controlleru nazývá Presenter. Architektura aplikace, či návrhový vzor, kterým se aplikace řídí, se tedy mění z MVC na MVP. MVP je odvozenina od MVC, ve které Presenter slouží jako prostředník mezi Modelem a View. Presenter se liší od Controlleru tím, že načítá data z Modelu a posílá je do View, zatímco v MVC můžou jít data z Modelu do View přímo. [11]

Rozdíl architektury MVP je vidět na obrázku č. 2.2. Oproti diagramu MVC (viz obr. č. 2.1) na tomto diagramu chybí spojení mezi Modelem a View. Mezi Modelem a Presenterem pak neprobíhá pouze manipulace s daty, ale i jejich předání. Poslední změna je vidět mezi Presenterem a View, kdy v MVC Controller řídí, která data se mají zobrazit, zatímco v MVP Presenter pouze zasílá informaci o tom, která data se změnila a která tedy musejí být vrstvou View překreslena.



Obrázek 2.2: Diagram popisující architekturu MVP

Dále budou popsány výhody a nevýhody používání frameworku Nette.

2.3.1 Výhody a nevýhody používání frameworku Nette

Nette má jako každý framework své výhody i nevýhody. Mezi hlavní výhody patří:

- Dependency Injection kontejner,
- špičkový ladící nástroj Tracy,
- zabezpečení proti známým typům útoků (XSS, CSRF, SQL Injection),
- využití paměti Cache,
- šablonovací systém Latte,
- podpora technologií HTML 5, AJAX, SEO a další.

Dependency Injection (dále DI) se týká vytváření závislostí mezi jednotlivými komponentami. Závislost lze definovat přes konstruktor, metodou (obvykle setter metodou), nebo proměnnou. V rozsáhlých systémech je obtížné tyto závislosti spravovat. Nette řeší DI automaticky a to tak, že si generuje DI kontejner, ve kterém se závislosti uchovávají. Objekty, které se

instancují, se v souvislosti s DI nazývají službami. DI kontejner se stejně jako služby konfiguruje v konfiguračním souboru. [6]

Konfigurační soubory se v Nette obvykle zapisují v syntaxi NEON, která je velmi podobná syntaxi YAML. Konfigurační soubor je rozdělen na sekce. Parametrům je určena sekce *parameters*, službám sekce *services*. Definice parametrů je vidět v kódu č. 2.2.

```
1 | parameters:
2 |     dsn: 'mysql:host=127.0.0.1;dbname=test'
3 |     user: root
4 |     password: secret
```

Kód 2.2: Definice parametrů v NEON souboru

Parametry se mohou použít v následné definici služeb, např. v registraci služby pojmenované *database*, která je reálně instancí PDO (viz kód č. 2.3). Nejsou-li parametry služby nastavené předem, musí se definovat uvnitř zápisu služby (viz kód č. 2.4). Takto registrovaná služba automaticky vygeneruje metodu v DI kontejneru (viz kód č. 2.5).

```
1 | services:
2 |     database: PDO(%dsn%, %user%, %password%)
```

Kód 2.3: Definice služby v NEON souboru (1. způsob)

```
1 | services:
2 |     database: PDO('mysql:host=127.0.0.1;dbname=test',
3 |                 root, password)
```

Kód 2.4: Definice služeb v NEON souboru (2. způsob)

```
1 | function createServiceDatabase(): PDO {
2 |     $service = new PDO('mysql:host=127.0.0.1;dbname=test'
3 |                       , 'root', 'secret');
4 |     return $service;
5 | }
6 | $database = $container->getService('database');
```

Kód 2.5: Funkce pro vygenerování služby z DI kontejneru

Další významnou výhodou Nette je důraz na eliminaci bezpečnostních rizik v aplikaci. Nette poskytuje ochranu proti nejznámějším typům útoků, jako je např. SQL Injection, XSS nebo CSRF. Nette automaticky escapuje proměnné v šablonách, nastavuje bezpečnostní hlavičky, ošetřuje vstupy do

aplikace nebo automaticky konfiguruje a zabezpečuje cookies. Typickým příkladem eliminace bezpečnostních rizik je automatické ošetření řetězců, které jdou na výstup.

Zabezpečení jednoduché šablony (viz kód č. 2.6) je naznačeno v kódu č. 2.6. *\$message* je proměnná, kterou je potřeba ošetřit. Pokud by se tak nestalo, útočník by mohl do stránky prostřednictvím tohoto neošetřeného výstupu podstrčit vlastní kód, kterým by mohl stránku změnit, nebo třeba získat data o návštěvnicích. Takovému útoku se říká Cross-site scripting (XSS). Nette takovéto výstupy ošetřuje za pomoci technologie Context-Aware Escaping automaticky.

```
1 | <p onclick="alert ({$message}) ">{$message}</p>
2 | <script>
3 | document.title = {$message};
4 | </script>
```

Kód 2.6: Nezabezpečená šablona

```
1 | <p onclick="alert (&quot;Text&quot;; &quot;) "Text&quot;; </p>
2 | <script>
3 | document.title = "Text";
4 | </script>
```

Kód 2.7: Automaticky zabezpečená šablona

Velkým pomocníkem při tvorbě aplikace je ladicí nástroj Tracy, který usnadňuje vyhledávání a opravování chyb. Jeho pomocí je možné logovat chyby, zasílat upozornění na chybu emailem, vypisovat proměnné, hlídat časové a paměťové nároky aplikace a vizualizovat nalezené chyby a výjimky. [7]

Výhoda a zároveň nevýhoda Nette spočívá ve způsobu jeho vývoje a licencování. Nette patří mezi open-source projekty. Využívat a dále vyvíjet ho může každý a zdarma. To vytváří dvousečnou zbraň, neboť nikde není garantována správnost vyvinutého kódu ani jeho budoucí vývoj.

I přes značné výhody frameworku se nedoporučuje Nette použít pokaždé, když se začne s vývojem nové webové aplikace, neboť systém Nette je velmi robustní. Jeho použití se nehodí, je-li potřeba vytvořit jednoduchou webovou stránku v PHP. Několika testy bylo prokázáno, že použití frameworku v takovém případě vývoj naopak zpomaluje. [9]

2.4 Protokoly pro přístup k API

API je zkratka pro Application Programming Interface, česky aplikační programové rozhraní. API je část softwaru, která spojuje aplikaci s daty a službami jiné aplikace tím, že jí přidělí přístup do specifických částí serveru. Komunikace s API může probíhat dvěma různými přístupy, kterými jsou protokol SOAP a REST API.

2.4.1 Protokol SOAP

SOAP (Simple Object Access Protocol) je protokol pro výměnu XML zpráv přes protokol HTTP. SOAP tvoří základní komunikační vrstvu mezi webovými službami. Mluví-li se o webových službách, je potřeba zmínit tři hlavní technologie, které s nimi souvisejí. Jde o samotný SOAP a dále WSDL (Web Services Description Language) a UUDI (Universal Description, Discovery and Integration). [10]

SOAP jako protokol definuje způsob kódování dat pro komunikaci se službou, WSDL pak specifikuje formát pro popis rozhraní webové služby. Pro každou webovou službu existuje popis jejího rozhraní v tomto formátu. Součástí takového popisu je definice vstupní a výstupní XML zprávy (Extensible Markup Language) a informace, jakým protokolem a na jakou adresu se má zpráva přenášet. UUDI představuje standardní mechanismus pro registraci a vyhledávání webových služeb. Lze říci, že UUDI představuje databázi webových služeb popsanych pomocí WSDL. [10]

2.4.2 REST API

REST API (Representational State Transfer) je API k webovým službám. REST API je postavené na URI (Uniform Resource Identifier, URL je specifickým typem URI). Charakteristickými rysy RESTu jsou nižší bezpečnostní požadavky, kompatibilita s prohlížečem klienta, škálovatelnost a jednoduchost díky HTTP protokolu, na jehož metody GET, POST, PUT a DELETE jsou mapovány operace RESTu. Data požadavku jsou typicky přenášena ve formě XML dokumentu. Není ale překážkou data poslat jako plain text, HTML nebo JSON. V jednodušších případech je k přenosu dat využít parametr GET. Ve složitějších případech, kdy je nutné kromě samotného obsahu požadavku přenést i nějaké metainformace, se k přenosu používá parametr POST. [10]

2.4.3 Srovnání protokolu SOAP a REST API

Často dochází ke srovnávání výše popsaných přístupů. Srovnávat tato dvě paradigmaty ale z podstaty věci není možné, neboť SOAP je protokol, zatímco REST je architektonický styl. Mezi těmito přístupy existuje několik klíčových rozdílů, jež jsou shrnuty v následujících několika tvrzeních:

- REST je data-driven architektura, zatímco SOAP function-driven.
- REST povoluje různé datové formáty (XML, HTML, JSON), SOAP umožňuje práci pouze s XML.
- Volání REST může být cachované, což šetří zdroje i čas. U SOAPu nastavení příznaku pro cachování možné není.
- REST využívá zabezpečenou verzi HTTP, SOAP podporuje zabezpečení WS-Security. Oba přístupy pak na přenosové úrovni používají SSL (Secure Sockets Layer).
- REST není úzce svázán se serverem. Klient komunikující přes REST nepotřebuje znát API. SOAP je velmi těsně spojen se serverem, s nímž má striktně danou dohodu o komunikaci. Proto SOAP klient potřebuje znát API, se kterým komunikuje. [21]

Tyto rozdíly bývají často ilustrovány na příkladu pohledu a obálky. Používání RESTu je přirovnáváno k posílání pohledu. Pohled obsahuje různé typy dat, je jednoduché ho odeslat a každý si může snadno přečíst, co je na něm napsáno. Používání protokolu SOAP je srovnáváno s odesláním pohledu v obálce. To vyžaduje některé kroky navíc a způsob nadepsání obálky je striktně daný. K přečtení textu na pohledu je ale nutné nejdříve obálku rozbalit. Při výběru REST nebo SOAP by si měl programátor vždy uvědomit, jaké vlastnosti od komunikace s API bude vyžadovat, a podle toho si vybrat jeden z přístupů. [20]

V této kapitole byly čtenáři nastíněny techniky použité při tvorbě modulu. V následující kapitole se práce zaměří na možnosti dopravy zásilek v ČR. Čtenáři budou přiblíženy služby jednotlivých dopravních společností, které budou z různých hledisek porovnány v závěru kapitoly.

3 Analýza a srovnání možností dopravy zásilek v ČR

V této kapitole bude provedena analýza služeb, které je možné využít pro dopravu zásilek v ČR. V ČR působí v současné době devět velkých přepravních společností zajišťujících přepravu zásilek z e-shopů. Těmito společnostmi jsou:

- Česká pošta
- Zásilkovna
- DPD
- PPL
- Geis
- GLS
- Uloženka
- InTime
- TopTrans

Jak bylo řečeno v úvodní kapitole, tato práce se bude zabývat nejčastěji využívanými společnostmi působícími na českém trhu. Společnosti byly pro tuto práci vybrány na základě výsledků průzkumu trhu, který v rámci projektu Česká-ecommerce.cz pravidelně ve spolupráci s internetovým srovnávačem Zboží.cz provádí společnost Shoptet. Účelem tohoto projektu je každoročně analyzovat český trh, jeho velikost a specifika. [17]

Z výsledků průzkumu trhu vyplývá, že nejvyužívanějšími společnostmi v roce 2019 byly: Česká pošta, Zásilkovna, PPL a DPD. Ty dohromady zajistily téměř 80 % přepravních služeb v ČR. Největší podíl na trhu přitom měly Česká pošta a Zásilkovna. [17]

Průzkum trhu ukazuje, že popularita tří ze čtyř výše uvedených vybraných společností meziročně klesá. Jde o Českou poštu, PPL a DPD. Čím

dál tím častěji jsou naopak využívány služby Zásilkovny. Poslední zmíněná společnost – Zásilkovna – zvolila zcela odlišnou strategii než její konkurenti. Svůj obchodní model postavila na franchisingovém způsobu zakládání výdejních míst, kterými začala disponovat hned od samého vzniku společnosti. Vysokým počtem výdejních míst získala výhodu oproti konkurenci, která se zaměřovala především na doručování zásilek na konkrétní adresu.

Doručování na výdejní místo nabývá na oblíbenosti konstantně již několik let. Tento nárůst zaregistrovaly i ostatní přepravní společnosti a začaly taktéž zřizovat svá vlastní výdejní místa, aby mohly tuto službu zařadit do své nabídky. Zásilkovna v tu chvíli měla už poměrně silnou pozici na trhu, kterou si drží dodnes.

Doručení na výdejní místo je velmi dobrou alternativou pro zákazníky, kteří by využili možnost osobního odběru, ale vybrali si zboží z e-shopu, který nemá kamennou prodejnu v jejich místě bydliště. Možnost doručení na výdejní místo je stejně jako osobní odběr zboží výhodné v tom, že příjemce není vázán časem ani místem doručení, pouze otevírací dobou výdejního místa.

Druhou variantou je doručení na adresu. Zákazník v objednávce uvede adresu, na kterou chce zásilku doručit. Dopravce zpravidla při převzetí zboží od e-shopu informuje zákazníka o předpokládaném datu doručení zásilky. V den doručení je zákazník znovu kontaktován kurýrem, který mu sdělí konkrétní časové rozmezí, kdy mu bude zásilka doručena. Tato služba bývá oproti variantě doručení na výdejní místo dražší a preferují ji zákazníci, kteří jsou flexibilní a dokáží se v den doručení přizpůsobit kurýrovi. Doručení na adresu se dále často využívá u firem s recepcí. Zásilku za příjemce – firmu či zaměstnance dané firmy – převezme třetí osoba. Příjemce se jejím převzetím nemusí vůbec zabývat

Tato kapitola se bude dále věnovat představení a vzájemnému srovnání služeb, které dopravní společnosti nabízejí. Pro názornější porovnání služeb bude napříč touto kapitolou využít reálný příklad poslání balíku o určitých rozměrech v rámci ČR, a to prostřednictvím všech vybraných přepravních společností. Důvodem pro vytvoření reálného příkladu je široká nabídka služeb, která se obecně špatně porovnává. Příklad bude použit proto, aby bylo možné nabídku konkretizovat natolik, aby výsledky srovnání byly relevantní a směrodatné.

V rámci této práce budou představeny pouze nejběžnější balíkové služby. Paletové služby, které dopravci nabízejí, zde uvedeny nebudou. Zaměření pouze na balíkové služby bylo zvoleno proto, že nabídka paletových služeb je ještě variabilnější než nabídka u těch balíkových. Velmi často podmínky služeb záleží hlavně na individuální domluvě mezi e-shopem a dopravcem.

Tyto služby tak nelze jednoduše porovnat. Ze stejného důvodu nebudou paletové služby integrovány do modulu řešícího dopravu.

Co se týče cen uvedených u jednotlivých služeb přepravců, platí, že ceny jsou pouze orientační a jsou převzaty z oficiálních ceníků dopravců aktuálních v době psaní této práce. Do výsledné ceny je nutné dále započítat DPH, příplatek za mýto a palivový poplatek. Dopravci si vyhrazují právo ceníky kdykoliv měnit. V případě e-shopových služeb je cena daná smluvními podmínkami mezi e-shopem a dopravcem. Podmínky se domlouvají na základě kritérií, kterými jsou např. počet expandovaných zásilek, typ příjemce, průměrná váha zásilek, obsah zásilek (typ zboží), využívání doplňkových služeb apod. V závislosti na všech těchto parametrech si poté dopravce stanoví cenu za doručení zásilky. Výsledná cena se může výrazně lišit od té základní, která je uvedena v ceníku vybrané služby.

V následujícím oddíle bude uveden přehled vybraných přepravních společností a jejich služeb. Dopravci budou představeni od největšího po nejmenšího co do zajištění služeb na českém trhu. Tzn. dopravci budou uvedeni v pořadí Česká pošta, Zásilkovna, DPD a PPL. Na závěr kapitoly budou srovnány jejich služby.

3.1 Česká pošta s.p.

Česká pošta je od roku 1993 samostatným státním podnikem, který je jediným a povinným držitelem poštovní licence v ČR. Povinností spojenou s touto licencí je např. provozovat 3200 pošt, doručovat každý pracovní den na všechny adresy v ČR nebo provozovat dostatečně hustou síť schránek a vybírat je každý pracovní den.

Česká pošta má v současnosti více než 3 800 poboček. Toto číslo ukazuje, že Česká pošta má ve srovnání s ostatními přepravními společnostmi nejhustší pobočkovou síť na území ČR. Ročně přepraví Česká pošta zhruba 14 mil. balíkových zásilek. [3]

Služby České pošty se dělí do dvou základních skupin – psaní a balíky. Nabídka obou skupin je velmi rozsáhlá. Jak již bylo uvedeno v úvodu, práce se bude věnovat pouze druhé kategorii – kategorii balíkových služeb a to těm, které se objevují v českých e-shopech. [1] Těmito službami jsou:

- Balík Do ruky
- Balík Na poštu
- Balík Do balíkovny

Balík Do ruky spadá do kategorie doručení na adresu, Balík Na poštu a Balík Do balíkovny do kategorie doručení do výdejního místa, tj. na pobočku České pošty. Balíkovna je část pošty nebo depa. Obvykle se jedná o speciálně označenou přepážku, na které probíhá zrychlený výdej balíků. Česká pošta doručuje zásilky u všech zmíněných služeb následující pracovní den od podání zásilky.

Způsob výpočtu cen za dopravu balíků se u České pošty v minulých letech výrazně změnil. Zatímco v minulosti se cena dopravy zásilky určovala podle její hmotnosti, v současné době cena závisí primárně na její velikosti. Velikostí se rozumí délka nejdelší strany zásilky. Podle této délky se zvolí jedna ze čtyř nově zavedených tabulkových velikostí (S, M, X, XL), podle kterých se stanovuje cena dopravy. Cena je vždy pro všechny zásilky z jedné kategorie stejná. Hmotnostní kategorie zůstaly zachovány pouze u služby Balík Do balíkovny, u které naopak přímo nezáleží na velikosti zásilky, ale na její hmotnosti. Zásilky poslané tímto způsobem se dělí na zásilky do 5 kg a do 20 kg. Balík Do ruky je omezen hmotností 50 kg a Balík Na poštu hmotností 30 kg. [2]

Následující dvě tabulky (tabulka č. 3.1 a tabulka č. 3.2) zobrazují ceny zmíněných služeb, tj. Balík Do ruky, Balík Na poštu a Balík Do balíkovny. Cena dobírky je u České pošty plošně nastavena na 17 Kč.

Tabulka 3.1: Ceník služeb Balík Do ruky a Na poštu

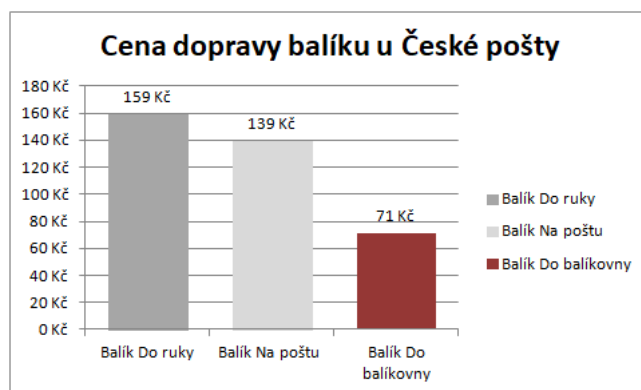
Balík	S (35 cm)	M (50 cm)	X (100 cm)	XL (240 cm)
Do ruky (do 50 kg)	129 Kč	159 Kč	209 Kč	359 Kč
Na poštu (do 30 kg)	109 Kč	139 Kč	189 Kč	339 Kč

Tabulka 3.2: Ceník služby Balík Do balíkovny

Balík	do 5 kg	do 20 kg
Do balíkovny	54 Kč	109 Kč

Aby bylo možné služby vybraných společností snáze porovnat, bude v této kapitole použita modelová situace, kdy chce zákazník odeslat balík o rozměrech 30x40x50 cm a hmotnosti 4 kg na dobírku za cenu 1 000 Kč.

U České pošty by posláním balíku vyšlo nejvýhodněji v případě využití služby Balík Do balíkovny, kdy by cena dopravy byla 71 Kč, což je dva krát méně než při využití služby Balík Do ruky. Ceny za dopravu balíku ukazuje graf č. 3.1.



Obrázek 3.1: Cenové porovnání modelové situace u České pošty

3.2 Zásilkovna

Společnost Zásilkovna.cz je česká přepravní společnost, která byla založena v roce 2010 a je tak jednou z nejmladších přepravních společností v ČR. Kromě ČR v současnosti funguje také na Slovensku a expanduje i do zemí východní Evropy. I přes relativně krátkou dobu působení v tuzemsku si získala silnou pozici na trhu.

Na rozdíl od ostatních dopravců, kteří byli vybráni pro účely této práce, Zásilkovna se až do roku 2019 orientovala výhradně na poskytování služeb pro e-shopy, tzn. prostřednictvím Zásilkovny si nebylo možné nechat poslat zásilku jinak než z e-shopu. Od dubna 2019 začala Zásilkovna nabízet i variantu přepravy zásilek pro fyzické osoby. Tato služba zatím funguje pod názvem Mezi Námi, ale pouze pro doručení na výdejní místa. [23]

Služby, které Zásilkovna nabízí e-shopům, jsou stejně jako u České pošty dvojího typu. Jedná se o doručování na výdejní místo a doručování na adresu, přičemž v českých e-shopech je integrována zejména první zmíněná služba.

Specifickou službou Zásilkovny je výše zmíněné doručování na adresu. Zásilkovna má několik smluvních partnerů, se kterými při doručování zásilek spolupracuje. Veškerou administrativu spojenou s podáním zásilky je možné provést v systému Zásilkovny, ale samotná doprava zásilky na adresu je realizována jedním z partnerů, kterými jsou DPD, Česká pošta, a pak také InTime a Zásilkovna – Expresní kurýr. Komunikaci s externími dopravci zajistí Zásilkovna sama. Využití Zásilkovny v kombinaci s externím dopravcem se liší nejen v ceně, ale i v délce trvání doručení zásilky. Doba doručení je upřesněna v tabulkách č. 3.3 a č. 3.4. U každého dopravce je v závorce označení $D+\text{číslo}$. Tento údaj značí, do kolika dnů ode dne podání je zásilka doručena. Např.: D+1 znamená následující pracovní den.

Při doručování na výdejní místo rozděluje Zásilkovna zásilky do dvou sku-

pin podle jejich hmotnosti a velikosti. Standardní zásilky váží do 5 kg, délka nejdelší strany nesmí přesáhnout 70 cm a součet všech tří stran pak 120 cm. Nadměrné zásilky váží do 10 kg, délka nejdelší strany nesmí přesáhnout 120 cm a součet všech tří stran pak 150 cm. Ceny služby doručení na výdejní místo jsou uvedeny v tabulce č. 3.3. Při doručení na adresu Zásilkovna rozlišuje zásilky pouze podle hmotnosti. Velikostně pak počítá s parametry pro nadměrné zásilky. Zásilkovna nabízí i možnost přepravy pro nadlimitní zásilky, tj. zásilky s rozměry většími než u nadměrných zásilek. Tuto dopravu je však nutné domluvit si předem a Zásilkovna si vyhrazuje právo na to zásilku odmítnout. Tato služba proto nebude zahrnuta do analýzy.

Dalším zajímavým faktorem, který ovlivňuje výslednou cenu dopravy, je místo podání zásilky. Je-li zásilka podána v jednom ze tří dep, která se nacházejí ve třech největších městech v ČR (konkrétně jsou to depa Praha Satalice, Brno Vídeňská a Ostrava Polanecká), je cena dopravy nižší než při podání na ostatních podacích místech Zásilkovny.

Srovnání cen služeb Zásilkovny je vidět v tabulce č. 3.4. Cena dopravy při podání ve speciálních depech v tabulce uvedena není. Oproti standardní ceně je vždy o 10 Kč nižší. Do tabulky je tentokrát v samostatném sloupci zaznamenána i cena dobírky, která se u jednotlivých dopravců liší. [22]

Tabulka 3.3: Ceník služeb doručení na výdejní místo u Zásilkovny

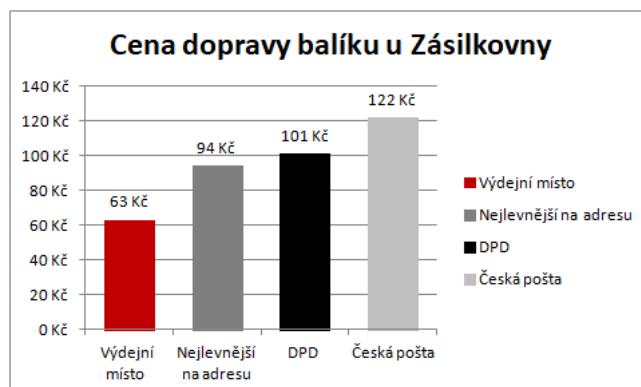
Služba	Standardní	Nadměrná zásilka	Dobírka
Doručení na výdejní místo (D+1)	51 Kč	101 Kč	12 Kč

Tabulka 3.4: Ceník služeb doručení na adresu u Zásilkovny

Služba	Do 5 kg	Do 10 kg	Dobírka
Večerní na adresu (D+0)	120 Kč	150 Kč	12 Kč
Nejlevnější na adresu (D+2)	75 Kč	99 Kč	19 Kč
DPD - na adresu (D+2)	79 Kč	109 Kč	22 Kč
Česká pošta - na adresu (D+2)	93 Kč	120 Kč	29 Kč

Srovnání cen Zásilkovny zde bude provedeno opět pomocí modelové situace (viz kapitola 3.1). Cenu dopravy za odeslání balíku na dobírku 1 000 Kč o rozměrech 30x40x50 cm a o hmotnosti 4 kg ukazuje graf č. 3.2. Nejlevnější by bylo balík doručit na výdejní místo, kdy by se cena dopravy včetně dobírky rovnala 63 Kč. Při doručení na adresu by bylo nejvýhodnější využít Zásilkovnu, která externího dopravce vybere na základě aktuální situace

v regionu, což umožňuje varianta Nejlevnější doručení na adresu. To vyjde zákazníka na 94 Kč včetně poplatku za dobírku.



Obrázek 3.2: Cenové porovnání modelové situace u Zásilkovny

3.3 DPD

Společnost DPD působí na českém trhu od roku 1994 a je členem mezinárodní přepravní sítě DPDgroup. DPDgroup opanuje v oblasti zásilkových a expresních služeb v Evropě 2. příčku. Mezinárodní síť tohoto dopravce tvoří 830 dep, která se nacházejí ve více než 40 zemích světa. V ČR zajišťuje DPD svými službami 100% pokrytí ČR. Tuzemskou síť tvoří 13 regionálních dep, které řídí jedno centrální celorepublikové překladiště se sídlem v Praze. [5]

DPDgroup odkoupila v listopadu 2019 společnost Geis Parcel CZ a Geis Parcel SK, které taktéž poskytují přepravu balíků. Tyto společnosti mají do budoucna v balíkové přepravě vzájemně spolupracovat. V průběhu roku 2020 bude proveden zápis o odkoupení společnosti a dojde přejmenování z Geis Parcel CZ na DPD. V tom okamžiku společnost Geis Parcel CZ fakticky přestane existovat. [15]

DPD nabízí v ČR tři typy služeb – Classic, Express a řešení pro e-shopy. Služby Classic a Express jsou si velmi podobné. U obou kurýr vyzvedne zásilku u klienta a doručí ji na adresu příjemce do druhého pracovního dne od vyzvednutí zásilky. U služby Classic však termín doručení není garantován, zatímco u služby Express ano. Klient si navíc u služby Express může vybrat ze tří variant, do kdy má být zásilka doručena (do 10:00, do 12:00 nebo do 18:00 hod. následujícího pracovního dne). Služby se liší i z pohledu hmotnostního a velikostního limitu. Službou Classic je možné přepravit zásilku do hmotnosti 50 kg s délkou nejdelší strany do 175 cm a obvodem

do 300 cm. U služby Express je maximální povolená hmotnost nižší, pouze 31,5 kg. Velikostní rozměry zůstávají stejné.

Řešení pro e-shopy tvoří dvě služby – služba Pickup (dříve DPD ParcelShop) a DPD Private. DPD Private odpovídá službě Classic s tím rozdílem, že odesílatelem je přímo e-shop. Zásilka je doručena příjemci na adresu. U služby Pickup zásilka není doručena příjemci na zadanou adresu, nýbrž na jedno z výdejních míst, které si příjemce vybere již v procesu objednávky. Co se týče limitů zásilek, DPD Private má stejná hmotnostní i velikostní omezení jako služba Classic. V případě služby Pickup je hmotnost omezena pouze na 20 kg, délka nejdelší strany na 100 cm a obvodová délka na 250 cm. Cena dopravy zásilky (viz tabulka č. 3.5) se odvíjí přímo od její hmotnosti. Na její velikosti tedy nezáleží.

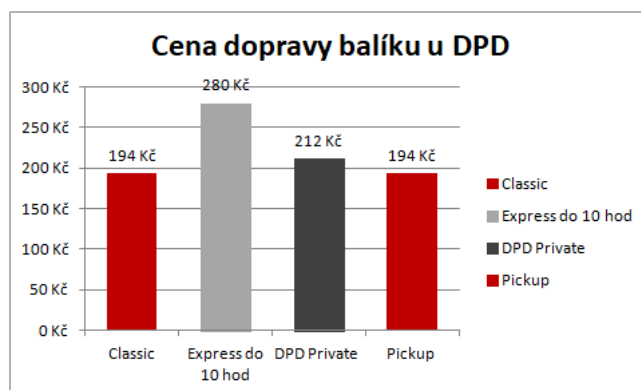
Tabulka 3.5: Ceník služeb doručení u DPD

Hmotnost do	Classic	Express 10:00	Private	Pickup
1 kg	116 Kč	193 Kč	133 Kč	113 Kč
2 kg	138 Kč	204 Kč	155 Kč	127 Kč
3 kg	141 Kč	217 Kč	157 Kč	137 Kč
4 kg	143 Kč	229 Kč	161 Kč	143 Kč
5 kg	148 Kč	233 Kč	163 Kč	148 Kč
10 kg	183 Kč	273 Kč	199 Kč	180 Kč
20 kg	216 Kč	337 Kč	231 Kč	193 Kč
50 kg	566 Kč	X	584 Kč	X

V modelové situaci, která je v této kapitole použita pro lepší srovnání cen dopravců, je posílána zásilka na dobírku 1 000 Kč o rozměrech 30x40x50 cm a o hmotnosti 4 kg. Graf č. 3.3 ukazuje, že nejlevněji by vyšlo odeslání zásilky službou Classic, které by stálo 194 Kč vč. poplatku za dobírku. Cena za dobírku se u DPD liší podle hodnoty dobírky. V kategorii do 5 000 Kč, do které spadá většina zásilek, si DPD účtuje poplatek 51 Kč. Stejná cena za dopravu by byla i v případě, že by zásilka byla posílána z e-shopu. Zákazník by si ji ale musel vyzvednout na jednom z výdejních míst (služba Pickup).

3.4 PPL

Systém PPL byl založen v roce 1995 a zpočátku reprezentoval sedm kooperujících společností. Sloučením těchto společností vznikla v roce 2004 jedna jediná: PPL CZ s.r.o, která patří k nejvýznamnějším společnostem přepravujících zásilky na českém trhu. Od počátku svého působení se specializuje primárně na balíkovou vnitrostátní přepravu, a to jak na firemní, tak



Obrázek 3.3: Cenové porovnání modelové situace u DPD

na soukromé adresy. Kromě třinácti regionálních dep a jednoho centrálního přecladiště disponuje PPL i hustou sítí nezávislých výdejních a podacích míst, tzv. PPL ParcelShopů. Tato místa mají stejnou funkci jako Balíkovny od České pošty, výdejní místa DPD Pickup či pobočky Zásilkovny. [19]

Služby, které nabízí PPL pro přepravu balíčků, se dělí na dvě kategorie. Do první z nich spadají služby, které jsou smluvně vázané: PPL Parcel CZ Business, PPL Parcel CZ Private a PPL Parcel CZ Dopolodní balík. Tyto služby mají stejné přepravní podmínky, ale liší se buď v typu příjemce (Business – firma, Private a Dopolodní balík – soukromá osoba) nebo času doručení (Dopolodní balík – do 10:00 hod). Doručování probíhá obecně do druhého pracovního dne. Druhou kategorií tvoří nově zavedená služba Balík pro Tebe. Ta je určena lidem, kteří chtějí využívat pro přepravu balíčků PPL, ale nechťejí se kvůli tomu smluvně vázat. PPL stejně jako ostatní dopravci nabízí možnost vyzvednutí zásilky na výdejním místě – v ParcelShopu.

Ceny služeb PPL jsou uvedeny v tabulce č. 3.6. Zásilka poslaná přes PPL nesmí překročit hmotnost 31,5 kg. Maximální rozměr zásilky může být 120x60x60 cm a zároveň součet obvodu a délky nejdelší strany zásilky může být max. 360 cm. U služby Balík pro Tebe jsou tyto limity nižší. Zásilka je omezena rozměry 100x50x50 cm. Cena dobírky u PPL se stejně jako u DPD rozlišuje dle výše její hodnoty. Pro zásilky s dobírkou do 1 000 Kč je poplatek 35 Kč. Pro zásilky s dobírkou do 5 000 Kč je poplatek jen o 10 Kč vyšší, tj. 45 Kč. Služba Balík pro Tebe zásilky na dobírku neumožňuje. Ceny služby Balík pro Tebe jsou uvedeny v tabulce č. 3.7.

Zaslání balíku, který je použit v modelové situaci napříč touto kapitolou, jehož rozměry jsou 30x40x50 cm a hmotnost 4 kg, by v případě PPL vyšlo nejméně na 182 Kč, vč. poplatku za dobírku. Pro zaslání by byla využita služba PPL Private. Ceny za dopravu balíku v modelové situaci jsou uvedeny v grafu č. 3.4. Pokud by byl příjemcem zásilky právnický subjekt, bylo by

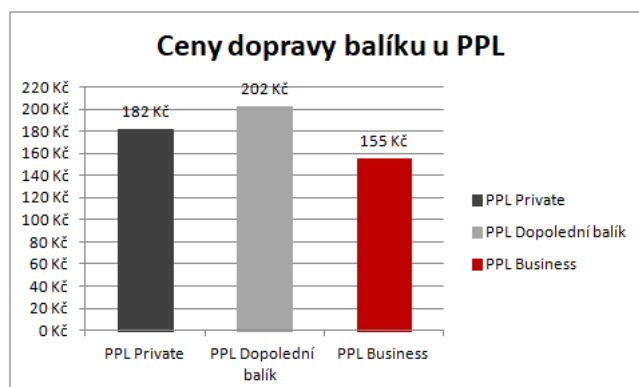
Tabulka 3.6: Ceník služeb doručení u PPL

Hm. do (kg)	Business	Private	Dopolední balík
2 kg	95 Kč	125 Kč	154 Kč
5 kg	110 Kč	147 Kč	167 Kč
10 kg	154 Kč	184 Kč	202 Kč
20 kg	187 Kč	230 Kč	271 Kč
31,5 kg	240 Kč	303 Kč	331 Kč

Tabulka 3.7: Ceník služby Balík Pro Tebe od PPL

Balík pro Tebe	Cena
S (30x30x30cm)	99 Kč
M (60x40x30cm)	119 Kč
L (100x50x50cm)	159 Kč

možné k odeslání využít službu PPL Business. Cena dopravy by pak byla pouze 155 Kč, vč. poplatku za dobírku. Služba PPL Balík pro Tebe není v této situaci brána v úvahu, neboť tato služba neumožňuje poslání zásilky na dobírku.



Obrázek 3.4: Cenové porovnání modelové situace u PPL

V předchozích podkapitolách byly vybrané společnosti vnímány jako samostatné celky a tak byly také analyzovány. Jejich služby byly rozebrány a následně využity v modelové situaci nezávisle na službách ostatních společností. V následující části této kapitoly budou služby dány do souvislostí. Srovnání služeb pak poskytne celkový pohled na situaci u vybraných dopravců.

3.5 Srovnání služeb

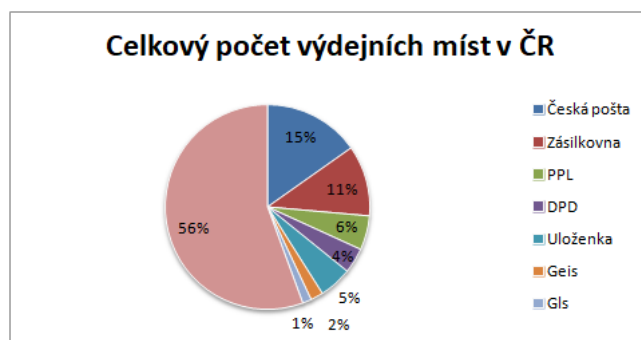
V následujícím oddíle budou služby všech vybraných společností porovnány z hlediska počtu výdejních míst, doby doručení zásilky a hmotnostních a velikostních limitů zásilek. Mimo jiné zde budou shrnuty výhody a nevýhody jednotlivých dopravců. Na závěr budou zhodnoceny služby, které by připadaly v úvahu v rámci modelové situace, která slouží pro transparentnější porovnání služeb společností a která se objevuje napříč touto kapitolou.

Podle internetového srovnávače Heureka.cz se v ČR nachází přibližně 25 000 výdejních míst, na kterých si zákazníci mohou vyzvednout své zboží objednané z internetových obchodů. Všechny čtyři porovnávané dopravní společnosti, tj. Česká pošta, Zásilkovna, DPD i PPL nabízejí varianty služeb pro doručení na adresu i na výdejní místo. Nej hustší síť výdejních míst má v ČR Česká pošta, u které se za výdejní místo považuje každá její pobočka, tj. pošta. Těch je v současné době více než 3 800. To znamená, že pobočky České pošty tvoří zhruba 15 % všech výdejních míst internetových obchodů v ČR. [8]

Českou poštu co do počtu výdejních míst rychle dohání Zásilkovna, u které se počet výdejních míst pohybuje kolem 2 800. To je zhruba 11 % z celkového počtu výdejních míst v ČR. Ve velkých městech se navíc výdejních míst Zásilkovny nachází obvykle výrazně více než poboček České pošty. Pro koncové zákazníky pohybující se ve městech je tak Zásilkovna dostupnější. Zásilkovna začala v posledních letech expandovat z velkých měst i do těch menších. Nyní se tak nachází ve všech městech s počtem obyvatel vyšším než 1 500. Touto expanzí, která zvýšila celkovou dostupnost Zásilkovny, výrazně omezila hlavní konkurenční výhodu České pošty, kterou byla právě dostupnost služeb v malých městech a vesnicích.

Zbývající dvě společnosti – PPL a DPD – provozují výdejních míst výrazně méně. PPL jich v současné době má kolem 1 350, což tvoří 5 % z jejich celkové počtu v ČR. Do konce roku 2019 jich ale původně bylo plánováno otevřít 1 900. Tento cíl se ukázal jako nepřiměřený. Síť výdejních míst rozšiřuje i společnost DPD, která na konci roku 2019 ve spolupráci se společností Sazka zvýšila počet míst asi 5x. DPD provozuje kolem 1 000 výdejních míst, což jsou 4 % z celkového počtu výdejních míst v ČR.

Poměr počtu výdejních míst jednotlivých společností k jejich celkovému počtu ukazuje graf č. 3.5. Kromě vybraných dopravních společností jsou v něm zaneseny údaje i od dalších přepravců s velkým počtem výdejních míst (Geis, GlS, Uloženko). Výdejní místa označená v grafu jako *Ostatní* značí ta místa, která provozují buď dopravci s minimálním počtem výdejních míst (např. TopTrans – 25) nebo přímo internetové obchody (např. Alza, atd.).



Obrázek 3.5: Celkový počet výdejních míst v ČR

Pokud by se na vybrané dopravní společnosti nahlíželo z pohledu e-shopů, bylo by celkem jasné, že jednou z priorit při výběru dopravců je právě dostupnost výdejních míst pro koncového zákazníka. Čím větší výběr míst má zákazník k dispozici, tím vyšší je pravděpodobnost, že si najde výdejní místo nacházející se v jeho blízkosti. Tím pohodlnější se pro něj stává nákup ve vybraném e-shopu, což přispívá k celkové spokojenosti s e-shopem. Úměrně tomu roste šance, že zákazník nákup v e-shopu zopakuje. Z tohoto pohledu je pro e-shopy nejvýhodnější Česká pošta a Zásilkovna, které provozují největší množství výdejních míst a svými službami pokrývají téměř celou ČR.

Dostupnost dopravce hraje roli nejen pro koncového zákazníka, ale i pro e-shop. Nejde jen o počet výdejních míst, ale v tomhle případě především o charakteristiku sítě distributora, vzdálenost e-shopu od dopravce a celkovou logistiku zajištění služeb. To se týká např. přijímání zásilek k doručení. Česká pošta, jakožto státní podnik, nemá oprávnění k tomu odmítat zásilky v době extrémního zájmu, např. před Vánoci. Soukromí dopravci toto právo mají a ve chvíli, kdy přestávají stíhat zásilky doručovat, mohou příjem nových zásilek pozastavit. Díky tomu se nestává, že by u soukromých dopravců docházelo v tomto čase k výraznému zpoždění doručování zásilek. Na druhou stranu toto pozastavení znamená komplikaci pro e-shopy, které s nimi spolupracují. Z toho důvodu naprostá většina českých e-shopů spolupracuje s Českou poštou i přes to, že spokojenost ze strany koncových zákazníků s jejími službami, zejména s dobou a způsobem doručení, je diskutabilní.

Doba doručení zásilek je oficiálně u všech společností srovnatelná. Všichni dopravci uvádějí doručení zásilky do druhého pracovního dne od podání zásilky. Jedinou výjimku tvoří Zásilkovna, která tento termín uvádí pouze u služby doručení na výdejní místo. Ke službě doručení na adresu využívá většinou externí partnerské dopravce. Využitím externího dopravce se doba dodání o jeden den prodlouží, tzn. zásilka bude doručena do dvou pracovních dnů ode dne podání.

Přestože společnosti slibují doručení do jednoho či dvou pracovních dnů od podání zásilky, tento termín nijak negarantují. V obchodních podmínkách se proti tomu vždy chrání větou, že termíny slouží pro informativní účely, a tudíž že nejsou závazné ani vymahatelné.

Jediný případ, kdy společnosti dobu doručení přímo garantují, je při využití některé z jejich expresních služeb. To jsou služby, u kterých doručení probíhá zrychleně a příjemci může být zásilka doručena již ten samý pracovní den večer nebo do určité hodiny následujícího pracovního dne po podání zásilky. Nevýhodou kromě vyšší ceny je, že tyto expresní služby bývají často dostupné pouze v krajských městech. Jedinou plošně dostupnou expresní službou je DPD 12:00, kdy je balíček doručen do 12:00 hod. následujícího pracovního dne.

Pokud by se do srovnání zařadily garantované expresní služby vybraných dopravců, kterými je možné doručit balík, nejrychleji ze všech reagují Česká pošta se službou Balík Expres, kdy je balík doručen v den dodání, a Zásilkovna se svou službou Večerní doručení (doručení taktéž tentýž den). Následuje DPD se službou DPD 10:00 a PPL se službou Dopolodní balík, kdy obě služby garantují dodání následující pracovní den do 10:00 hod. dopoledne.

Na posledních místech v pořadí by pak byly služby DPD 12:00 a DPD 18:00. Pořadí služeb je také vidět v tabulce č. 3.8. U všech těchto služeb ale musejí být splněny určité podmínky týkající se času a místa podání balíku (např. viz výše podání balíku pouze na určitých místech).

Tabulka 3.8: Srovnání služeb z hlediska rychlosti doručení

Pořadí	Společnost	Služba
1.	Česká pošta Zásilkovna	Balík Expres Večerní doručení
2.	DPD PPL	Express 10:00 Dopolodní balík
3.	DPD	Express 12:00
4.	DPD	Express 18:00

Dostupnost a rychlost doručení jsou tedy společnými faktory ovlivňujícími výběr způsobu dopravy (u zákazníka i u e-shopu). Mezi další kritéria, která mohou být klíčová při volbě dopravce u e-shopů, patří smluvní a reklamační podmínky, kvalita zákaznického servisu, přehled objednávkového systému, spolehlivost nebo také poměr mezi cenou a kvalitou služeb pro zákazníky.

U každého dopravce bývá situace rozdílná, a tak si při výběru dopravce

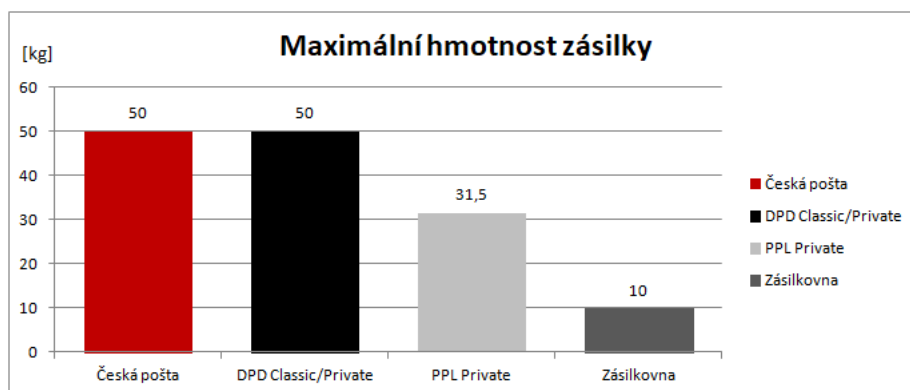
e-shop musí zmíněná kritéria prioritizovat a konečný výběr uskutečnit podle toho, jaké výhody a zároveň nevýhody by plynuly z navázání spolupráce s tím, či oním dopravcem. Hlavní výhody a nevýhody jednotlivých dopravců z pohledu e-shopů ale i koncového zákazníka přehledně shrnuje tabulka č. 3.9 [13]

Tabulka 3.9: Výhody a nevýhody dopravců

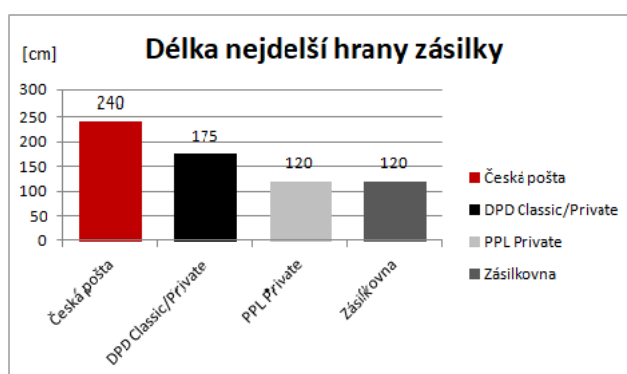
Dopravce	Výhody	Nevýhody
Česká pošta	Velké množství poboček, dostupnost i v malých městech, automatické svozy zásilek, standardní dopravce, neomezený příjem zásilek, automatické přesměrování zásilky na pobočku	Nízká míra komunikace, zdlouhavé řešení problémů, delší reálná doba doručení
Zásilkovna	Nízká cena služeb, velké množství poboček, automatické denní svozy, rychlost doručení, flexibilita	Nejnižší hmotnostní limit zásilek
DPD	Spolehlivost	Užší síť výdejních míst
PPL	Rychlost komunikace, přístup k řešení problémů, přeprava velkých spotřebičů	Častá nespolehlivost, dražší doprava

Níže uvedené grafy limitů srovnávají hmotnostní a velikostní limity zásilek u balíkových služeb. Z grafu hmotnostních limitů (viz graf č. 3.6) vyplývá, že největší rozměry zásilek povoluje ze srovnávaných dopravců Česká pošta. Prostřednictvím služby Balík Do ruky lze poslat zásilku až do hmotnosti 50 kg a délky nejdelší strany až do 240 cm. Graf také ukazuje, že stejný hmotnostní limit má také společnost DPD u služeb Classic a Private (alternativa Classic pro e-shopové řešení). V grafu velikostních limitů (viz graf č. 3.7) je však zřetelně vidět, že délka nejdelší strany balíku u DPD může být max. 175 cm. To je o 65 cm méně než u České pošty. Nejhuře ze srovnání vychází Zásilkovna, již je možné zaslat zásilky vážící max. 10 kg. Takto nastavený limit nemusí být nutně špatně. Zásilkovna se tímto způsobem vymezuje oproti své konkurenci, která cílí i na velké zásilky. Zásilkovna cílí na menší, lehčí balíčky, což je v podstatě jiná cílová skupina. Je to určitý typ strategie, kterou se Zásilkovna řídí a která jí zatím velmi dobře vychází.

Nejdůležitějším měřítkem při výběru doručovací služby je u většiny zá-



Obrázek 3.6: Srovnání hmotnostních limitů dopravců



Obrázek 3.7: Srovnání velikostních limitů dopravců

kazníků vedle doby doručení hlavně cena. Porovnat mezi sebou společnosti z hlediska ceny je ale samo o sobě obtížné, neboť každá z nich nabízí jiné služby a ceny nastavuje podle jiných parametrů. U DPD se např. cena zásilky vypočítává pouze podle hmotnosti, u České pošty naopak na cenu dopravy hmotnost vliv nemá a cena se určuje podle délky nejdelší strany zásilky. Proto byla na začátku této kapitoly vytvořena modelová situace, kdy je z Plzně do Prahy posílána zásilka na dobírku 1 000 Kč. Zásilka váží 4 kg a její rozměry jsou 30x40x50 cm. U každé přepravní společnosti pak byly propočítány ceny služeb, kterými lze zásilku odeslat. Ceny byly dále zobrazeny jak v tabulce, tak v odpovídajícím grafu.

V této části podkapitoly bude z modelové situace od každé společnosti vybrána cenově nejvýhodnější služba. Tyto služby budou následně porovnány. Tabulka č. 3.10 ukazuje výsledky tohoto srovnání pro doručení na adresu, zatímco tabulka č. 3.1 pro doručení na výdejní místo (pokud společnost takovou službu nabízí a cenově rozlišuje mezi doručením na adresu a na výdejní místo).

Tabulka 3.10: Srovnání nejvýhodnějších služeb doručení na adresu

Společnost	Služba	Cena
Česká Pošta	Balík Do ruky	159 Kč
Zásilkovna	Nejlevnější na adresu	94 Kč
DPD	Classic	194 Kč
PPL	Private	182 Kč

Tabulka 3.11: Srovnání nejvýhodnějších služeb doručení na výdejní místo

Společnost	Služba	Cena
Česká Pošta	Balík Do balíkovny	71 Kč
Zásilkovna	Na výdejní místo	63 Kč
DPD	DPD Pickup	194 Kč

Obě výše uvedené tabulky ukazují, že cenově nejdostupnější jsou služby nabízené Zásilkovnou. Odeslání zásilky vč. poplatku za dobírku vyjde při doručení na adresu na 94 Kč. To je výrazně nižší cena než u ostatních dopravců. Při doručení na výdejní místo vyjde cena dopravy na 63 Kč. Tomu může konkurovat pouze služba od České pošty Balík Do balíkovny. Balíkoven je ale v ČR zhruba 7x méně než výdejních míst Zásilkovny.

V tabulce č. 3.3 není vůbec uvedena cena služby PPL doručení na výdejní místo (do ParcelShopu). Je to proto, že PPL cenově nerozlišuje mezi doručením na adresu a na výdejní místo. DPD mezi službami cenově rozlišuje, ale pouze v rámci e-shopového řešení (Private a Pickup), jinak také ne.

Přepravní společnosti ke svým službám nabízejí pestrou nabídku doplňkových služeb. Mezi nejvyužívanější patří placení na dobírku či připojištění zásilky. Doplňkové služby jsou u každé společnosti jiné, jinak definované a jinak zpoplatněné. Do srovnání služeb byla zahrnuta pouze dobírka jakožto reprezentant těchto služeb. Další porovnání s využitím doplňkových služeb je nad rámec rozsahu a cílů této práce.

4 Analýza a srovnání způsobů komunikace mezi e-shopy a vybranými dopravci

Zatímco v předchozí kapitole byly jednotlivé přepravní společnosti porovnány z hlediska nabízených služeb, v této kapitole budou společnosti porovnány s ohledem na způsoby komunikace, kterými lze zajistit spojení mezi e-shopy a vybranými dopravci.

Nejčastějším způsobem, kterým lze navázat takovou komunikaci, je využití API či jiných webových služeb, které dopravci e-shopům poskytují. Cílem této kapitoly bude tyto možnosti představit a porovnat je z hlediska:

- typu a množství poskytovaných služeb
- možnosti integrace s e-shopem
- typu formátů dat potřebných pro přenos informací
- způsobu samotného přenosu dat o zásilce
- použitelnosti a uživatelské přívětivosti

Komunikace bude v rámci této analýzy rozdělena na využití webových služeb v průběhu nákupu a při předání zásilky dopravci. V průběhu nákupu získává e-shop od vybraného dopravce data o pobočkách a dostupných službách (např. aktuální informace o výdejních místech). Po dokončení objednávky pak probíhá komunikace ohledně podání zásilky z e-shopu dopravci.

Podání balíku je krok, který následuje po objednání zboží z e-shopu koncovým zákazníkem. V rámci podání balíku je nutné zpracovat data z objednávky, připravit je do požadovaného formátu a následně je předat dopravci, který bude zásilku doručovat. Předání dat jako takové může probíhat ručně, např. importem souboru do systému, nebo automatizovaně pomocí API. Druhý způsob přichází v úvahu pouze v případě, pokud dopravce tento typ přenosu dat podporuje a API e-shopům poskytuje.

Dále budou představeny různé webové služby, API a další způsoby komunikace vybraných dopravních společností ve stejném pořadí, jako tomu bylo v rámci analýzy služeb (viz kapitola 3). Tzn. v pořadí Česká pošta, Zásilkovna, DPD a PPL. Na závěr kapitoly bude provedeno srovnání uvedených způsobů komunikace.

4.1 Webové služby České pošty

Česká pošta disponuje několika typy webových rozhraní. První z nich se nazývá B2B¹ Brána. Jde o API k webové aplikaci POL², která zajišťuje veškeré služby spojené s online podáním zásilek. Aplikace POL je dostupná pouze smluvním zákazníkům. Dokumentace k API pro tuto aplikaci bohužel není veřejně dostupná a Česká pošta ji pro účely této práce neposkytlá. API B2B Brána proto nebude podrobněji rozebráno. O aplikaci POL bude více napsáno dále v textu.

Kromě B2B Brány nabízí Česká pošta také B2C³ Bránu. Tu tvoří několik webových služeb, z nichž každá se skládá vždy ze dvou operací, které se liší pouze formátem výstupních dat. Jedna operace vrací výstup ve formátu JSON, druhá ve formátu XML.

B2C Bránu tvoří služby, kterými e-shop získává informace od dopravce. Získané informace využívá e-shop především v procesu objednávky, tj. před samotným podáním balíku. Těmito službami jsou:

- *PostCode*: Služba, která vrací PSČ dotazované lokality a informace o dostupnosti služby Časová pásma v dané lokalitě.
- *PostOfficeInformation*: Služba, která vrací informace o pobočkách České pošty.
- *AfternoonParcelDelivery*: Služba, která vrací informace o dostupnosti služby odpoledního doručování na danou adresu.
- *Address*: Služba, která vrací identifikátor adresy potřebný pro službu *AfternoonParcelDelivery*.
- *ParcelHistory*: Služba, která vrací historii stavů požadované zásilky.
- *ParcelRating*: Služba, přes kterou se zadává spokojenost s doručením zásilky.

Třetí typ služeb nabízených Českou poštou je v e-shopech pravděpodobně využíván nejčastěji. Jde o webové služby, pomocí kterých e-shop získá informace o výdejních místech České pošty. Tyto informace potřebuje mít e-shop ve chvíli, kdy si jeho koncový zákazník vybere jako způsob dopravy služby

¹B2B – Business to Business – obchodní vztah mezi dvěma společnostmi

²Aplikace POL – aplikace pro podání balíku online

³B2C – Business to Customer – obchodní vztah mezi společností a koncovým zákazníkem.

Balík Do ruky nebo Balík Do balíkovny. Chce-li e-shop nabízet zákazníkům obě služby, musí mít zajištěné oba seznamy, neboť seznamy se vůči sobě liší.

Seznam pro službu Balík Na poštu se nachází na adrese:

`http://napostu.cpost.cz/vystupy/napostu.xml`

Seznam pro službu Balík Do balíkovny se nachází analogicky na adrese:

`http://napostu.cpost.cz/vystupy/dobalíkovny.xml`

Jak je vidět z výše uvedených URL adres, seznamy jsou dodány ve formátu XML. Data týkající se poboček mají jasně danou strukturu. Tato struktura je vidět níže v kódu č. 4.1.

Při používání této webové služby je důležité zajistit pravidelnou aktualizaci dat od dopravce. Seznam poboček tedy nestačí jednou stáhnout, zpracovat a pak už jen používat. Data o pobočkách se totiž mohou kdykoli změnit. V zájmu e-shopu je pak poskytovat svým zákazníkům v každém okamžiku platné informace, aby nedošlo k tomu, že si zákazník vybere výdejní místo, které již neexistuje. Každý e-shop by měl proto systémově synchronizovat data ze své databáze s daty od dopravce. K tomu se často využívá CRON, systémový nástroj pro automatizované spouštění příkazů.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <zv xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='https://www.cpost.cz/schema/aict/zv' xsi:
   schemaLocation=' '>
3   <row>
4     <PSC></PSC>
5     <NAZ_PROV></NAZ_PROV>
6     <OKRES></OKRES>
7     <ADRESA></ADRESA>
8     <V_PROVOZU></V_PROVOZU>
9     <PRODL_DOBA></PRODL_DOBA>
10    <OTV_DOBA></OTV_DOBA>
11  </row>
12 </zv>
```

Kód 4.1: Struktura XML souboru popisujícího pobočku České pošty

Dále budou rozebrány způsoby komunikace mezi e-shopy a Českou poštou během podání balíků.

Česká pošta nabízí čtyři způsoby podání balíků. Podání je možné uskutečnit pomocí:

- webového rozhraní k aplikaci POL - API B2B Brána
- webové aplikace POL
- elektronického podacího archu ePA

- datové věty

Z výše uvedených způsobů se pouze první z nich (API B2B Brána) řadí do kategorie automatizovaného způsobu podání balíku, kdy se data o zásilce posílají z e-shopu přímo do systému České pošty. Ostatní způsoby spadají do kategorie ručního předání dat dopravci ať už nahráním souboru do aplikace POL, či reálným podáním zásilek na pobočce České pošty.

Značnou nevýhodou používání webové aplikace POL (přímo, či přes rozhraní) je podmínka smluvního vztahu s Českou poštou. E-shop si tak před tím, než začne využívat tuto aplikaci, musí nejprve sjednat smlouvu a zařídit si tak přístup do klientské části aplikace. Výhodou podání balíků online přes aplikaci, popř. přes rozhraní, na druhou stranu je, že se zákazník nemusí starat o generování adresních štítků. Štítky aplikace generuje podle dat zadaných do systému. Zákazník si je pak musí už jen vytisknout a nalepit na zásilky.

Využívání aplikace POL se zdá být dobrým způsobem pro podání balíků online. Její ovládání však není příliš intuitivní ani uživatelsky přívětivé. Jedním z důvodů může být rozsáhlá funkcionality aplikace. Aplikace zajišťuje veškeré služby spojené s podáním balíku. Uživatel si v systému může nastavit různá podací místa, vytvářet seznamy adresátů, konfigurovat formát a strukturu souborů (formát TXT nebo CSV), ze kterých importuje data o zásilkách, nebo zásilky v systému ručně vytvářet. Dále může exportovat seznamy podaných zásilek včetně informací o jejich stavu a v neposlední řadě také generovat a tisknout adresní štítky.

Třetím způsobem podání balíku je pomocí elektronického archu ePa. Arch si zákazník vyplní sám a pak ho osobně předá dopravci. Předávaný soubor musí být ve formátu CSV s pevně definovanou strukturou týkající se pořadí sloupců, typu oddělovače (;), přípony souboru (.csv) nebo kódování (čeština). Soubor ePa musí obsahovat alespoň dva řádky. První řádek souboru se týká odesílatele. Druhý až n-tý řádek souboru pak popisuje jednotlivé zásilky. Výhodou tohoto způsobu je, že podání balíku není na rozdíl od prvních dvou popsanych způsobů podmíněno smluvním vztahem e-shopu s Českou poštou.

Pořadí sloupců v prvním řádku souboru je následující:
ID číslo podací pošty odesílatele; Příjmení a jméno; PSČ; Obec; Část obce; Ulice; Č.p.; Č.o.; ISO kód země; Telefon; E-mail; Číslo zákaznické karty České pošty

Pořadí hlavních sloupců v druhém až n-tém řádku souboru je následující:
ID. číslo zásilky; Příjmení a jméno adresáta (název firmy); PSČ; Obec; Část obce; Ulice; Č.p.; Č.o.; ISO kód země; Telefonní číslo; E-mail; Hmotnost zá-

sílky v kg; Doplnkové služby; Udaná cena v Kč; Částka dobírky v Kč;...

Posledním, čtvrtým způsobem podání balíku je podání datovou větou. Tento způsob slouží pro e-shopy, které chtějí využívat systému hromadného podávání zásilek. Aby bylo možné tento způsob využívat, musí být v e-shopu implementována příslušná funkcionality pro přípravu dat a náležitostí k zásilkám. Veškerá data o zásilkách se poté zpracují automaticky v e-shopu a na konci se dopravci předají již zpracovaná data osobně, např. na flash disk. Datová věta je podmíněna vlastním generováním adresních štítků s čárovým kódem. Generování štítků je nutné implementovat podle velmi rozsáhlé specifikace, což je velkou nevýhodou tohoto způsobu podání.

4.2 Webové služby Zásilkovny

Zásilkovna poskytuje stejně jako Česká pošta webové služby pro obě varianty komunikace s e-shopy, tzn. jednak služby pro získání dat od dopravce – informace o výdejních místech, jednak pro předání dat o zásilkách dopravci – pro podání balíku.

První varianta komunikace je podobná u Zásilkovny nejen co do získaných informací, ale i co do způsobu získávání těchto informací. Zásilkovna taktéž poskytuje seznam svých výdejních míst. Na rozdíl od České pošty si e-shop při implementaci této služby může vybrat ze dvou možných formátů souboru, v němž se nachází informace o výdejních místech. Těmito formáty jsou XML a JSON. Oba formáty mají opět jasně danou strukturu.

Kód č. 4.2 popisuje zkrácenou strukturu souboru v JSON formátu. Stejně jako u České pošty tu platí, že seznam je potřeba pravidelně aktualizovat, aby nedošlo k situaci, kdy si zákazník e-shopu vybere výdejní místo, které již neexistuje. Kromě toho je potřeba ošetřit výpadky při synchronizaci dat (např. aplikováním rollbacku v databázi).

```
1 | { "data": { "id": "", "name": "", "nameStreet": "", "place": "", "street": "", "city": "", "zip": "", "country": "", "currency": "", "maxWeight": "", "region": "", "district": "", "openingHours": "tableLong": "<table class='packetery-hours'><tr><th>Po</th><td></td></tr>\n</table>" }
```

Kód 4.2: Struktura JSON souboru popisujícího pobočku Zásilkovny

Pokud se e-shop nechce starat o pravidelnou aktualizaci seznamu výdejních míst, může využít alternativní způsob implementace služby. Zásilkovna nabízí kromě seznamu výdejních míst widget, který se integruje do e-shopu a data se v něm aktualizují automaticky. Výhodou tohoto widgetu je také

uživatelská přívětivost. Widget má intuitivní ovládání, přehledné grafické rozhraní s mapou a všemi užitečnými informacemi o výdejním místě. Kromě toho je v něm integrované filtrování a funkce autocomplete, která uživateli napovídá při vyhledávání místa.

Zásilkovna se k České poště připodobňuje i ve způsobech podání balíku. To u Zásilkovny může proběhnout dvěma způsoby. Prvním z nich je online podání prostřednictvím webové aplikace, kdy se zásilky podávají jednotlivě, nebo hromadně. Podání zásilek probíhá v aplikaci importem dat o zásilkách z předem připraveného souboru, který musí být ve formátu CSV a nebo XML. Po registraci zásilek v systému se automaticky vygenerují čárové kódy pro označení zásilek a volitelně i dodací list.

Soubor ve formátu CSV musí být v kódování UTF-8, Windows-1250 nebo ISO-8859-2. Struktura souboru je opět pevně daná. Na začátku souboru se nacházejí dva hlavičkové řádky. První z nich obsahuje řetězec verze 5, druhý řádek se ignoruje. Od třetího řádku dál se pak nacházejí data o zásilce, přičemž platí, že co řádek, to jeden balík. První sloupec v každém dalším řádku je rezervovaný a musí být prázdný. Oddělovačem v souboru může být buď symbol čárky (,) nebo středníku (;).

Pořadí sloupců v souboru je následující:

Rezervovaný prázdný sloupec; číslo objednávky; jméno příjemce; příjmení; společnost; email; telefon; částka dobírky; měna; hodnota zásilky; hmotnost v kg; ID výdejního místa; štítek odesílatele;

Druhou variantou podání balíku je použitím Packetery API, kterým je možné posílat data o zásilkách Zásilkovně z e-shopu přímo. Packetery API je jednotné pro celou logistickou skupinu Packeta. Packeta dále vlastní všechny ostatní regionální Zásilkovny (HU, DE, PL, SK, RU). Toto API tvoří jednu z hlavních výhod Zásilkovny, neboť integrací jednoho systému získá e-shop možnost spolupráce s desítkami dopravců, kteří využívají různé typy plateb i doručení, a to ve všech zemích EU a navíc ve Švýcarsku, na Ukrajině a v Rusku. API funguje na základě protokolu SOAP, nebo pomocí REST API. Funkcionalita je u obou řešení stejná.

Packetery API umožňuje:

- posílat balíčky na výdejní místo nebo prostřednictvím partnerských dopravců na adresu
- exportovat objednávky do systému
- generovat PDF štítky na označování balíků

Tuto funkcionalitu zajišťují především následující metody:

- *packetAttributesValid()*: Zkontroluje správnost atributů balíku.
- *createPacket()*: Vytvoří balík s atributy.
- *packetStatus()*: Vrátí informaci o stavu balíku.
- *packetTracking()*: Vrátí historii sledování balíku.
- *packetGetStoredUntil()*: Vrátí datum, do kdy je balík uskladněn.
- *packetSetStoredUntil()*: Nastaví datum, do kdy bude balík uskladněn.
- *packetLabelPdf()*: Vrátí pdf soubor se štítkem na označení balíku ve vybraném formátu (A6, A7, ...)

Všechny funkce API přijímají jako první parametr API key. API key je identifikátor k účtu Packeta. Každý tento klíč je unikátní a je úzce spojený s e-shopem. E-shop ho získá po registraci do systému Packeta.

Použití Packetery API přes protokol SOAP je demonstrováno na funkci *PacketIdDetail createPacket(PacketAttributes attributes)*, která na základě ověřených atributů v systému vytvoří balík (viz kód č. 4.3).

```

1 <?php
2 $client = new
3 SoapClient("http://www.zasilkovna.cz/api/soap-php-bugfix.wsdl
   ");
4 $pw = "1234567890abcdef1234567890abcdef";
5 $attributes = [
6     "number" => "00020",
7     "name" => "Lucie",
8     "surname" => "Tauchenova",
9     "email" => "lucie@tauchenova.com",
10    "addressId" => 95,
11    "value" => 100.00,
12    "eshop" => "muj-eshop.cz",
13 ];
14 try {
15     $client->createPacket($pw, $attributes);
16 }
17 catch (SoapFault $e) {
18 } ?>

```

Kód 4.3: Vytvoření balíku v Packetery API (protokol SOAP)

Jak by vypadalo vytvoření balíku při použití REST API, ukazuje kód č. 4.4. Požadavek na vytvoření balíku by měl být odeslaný metodou POST,

kteřá má v těle XML dokument. Požadovaná metoda API je uvedena v kořenovém názvu elementu, zatímco subelementy tvoří argumenty vybrané metody. Odpověď serveru dodržuje stejná pravidla jako požadavek.

```
1 <createPacket>
2   <apiPassword>...</apiPassword>
3   <packetAttributes>
4     <number>123abc</number>
5     <name>Lucie</name>
6     <surname>Tauchenova</surname>
7     <email>lucie@tauchenova.com</email>
8     <addressId>95</addressId>
9     <value>100.00</value>
10    <eshop>my-eshop</eshop>
11  </packetAttributes>
12 </createPacket>
```

Kód 4.4: Vytvoření balíku v Packetery API (REST API)

Detaily k dalším funkcím lze najít v referenční příručce, která je dostupná online v anglickém jazyce. [12]

4.3 Webové služby DPD

Společnost DPD, stejně jako dvě předchozí společnosti, nabízí službu vyzvednutí zásilek na výdejním místě. DPD k tomu e-shopům poskytuje seznam dostupných výdejních míst. Způsobů získání těchto dat existuje hned několik. Varianta, která se vyskytla zatím u všech zkoumaných společností, je stažení seznamu od dopravce. Tato varianta je vhodná pro účely vlastního řešení integrace služby. [4]

DPD poskytuje data ve formátu XML a CSV. Aktuální seznamy se nacházejí na adresách:

<https://pickup.dpd.cz/export/csv?country=203>

<https://pickup.dpd.cz/export/xml?country=203>

Na rozdíl od předchozích dopravců je u DPD možné si před stažením seznamů od dopravce výsledná data pomocí GET parametrů (*country*, *charset*, *output=file*) upravit. Těmito parametry je možné nastavit zemi, ze které se budou výdejní místa načítat. DPD je totiž nadnárodní společnost operující napříč evropskými zeměmi. Dále lze změnit kódování výstupu, a dokonce i určit místo (soubor), do kterého bude výsledek volání rovnou uložený.

Druhým způsobem implementace služby je integrace seznamu přímo do formuláře v e-shopu pomocí JavaScriptu. V kódu je pak nutné mít kromě

samotného kódu skriptu i odkaz na skript frameworku jQuery a DPD služby (viz kód č. 4.5).

```
1 <script src="https://ajax.googleapis.com/ajax/libs/jquery
  /3.4.1/jquery.min.js"></script>
2 <script src="https://pickup.dpd.cz/api/formIntegration"></
  script>
3 <script>
4 $(document).ready(function() {
5   dpdShipping.setBox($('select[name="odberne_misto"]'));
6   dpdShipping.addField($('input[name="ulice"]'));
7   dpdShipping.addField($('input[name="mesto"]'));
8   dpdShipping.addField($('input[name="psc"]'));
9   dpdShipping.setCrossBoarder(0);
10 });
11 </script>
```

Kód 4.5: Integrace DPD služby výdejních míst

Nechce-li e-shop implementovat ani tuto variantu, může být použita varianta přesměrování, kdy je zákazník v momentu výběru výdejního místa přesměrován na web DPD, konkrétně na adresu:

<https://pickup.dpd.cz/?address=Město,ulice,PSČ&return=http://example.com/objednavka>

Zde si zákazník vybere výdejní místo, které je nejbližší adrese předané dopravci v GET parametrech. Zákazník je po uskutečnění výběru přesměrován zpět do e-shopu. Vybrané výdejní místo je poté e-shopu předané v parametrech POST. Jedná o parametry:

- parcelshop_id – ID výdejního místa
- parcelshop_name – název
- parcelshop_address – adresa
- parcelshop_cod_allowed – povolení / zakázání dobírky

Společnost DPD poskytuje webové služby nejen pro získání dat od dopravce, ale i pro podání zásilky. k tomu slouží webová aplikaci Moje DPD a webové služby DPD GeoAPI. Moje DPD zajišťuje kompletní správu zásilek, zatímco DPD GeoAPI informuje o existujících zásilkách. Nevýhodou těchto služeb je, že je mohou využívat pouze smluvní zákazníci, kteří si služby nechají na vyžádání aktivovat. K jejich používání reálně potřebují uživatelské jméno (WSName) a heslo (WSPassword), identifikaci plátce (payerId) a identifikaci adresy svozu (addressId). Jméno a heslo vyžaduje každá metoda jako část vstupních parametrů.

Jak bylo zmíněno výše, aplikace Moje DPD se stará o správu zásilek. Správou je myšleno vytvoření zásilky, její případná editace, tisk štítků, objednávka svozu a správa seznamu všech zásilek. K aplikaci je možné využít i API a volat tím služby DPD přímo z e-shopu. Hlavními metodami služeb API jsou:

- CalculatePrice(): Vrací výpočet ceny dopravy.
- CreateShipment(): Vytvoří zásilku (viz kód č. 4.6).
- PrintLabelforShipment(): Vytiskne štítek zásilky.
- GetShipmentStatus(): Vrací aktuální stav zásilky.
- SearchShipment(): Vrací seznam zásilek dle vstupních parametrů (datum, zákazník, příjemce, ...).
- CloseManifest(): Vrací seznam zásilek během dokončení expedice zásilek (volitelně se štítky zásilek).

```
1 <ship:createShipment>
2   <shipmentList>
3     <shipmentId></shipmentId>
4     <!-- Identifikace zasilky, platce a mista odeslani-->
5     <shipmentReferenceNumber>WS123</
      shipmentReferenceNumber>
6     <payerId>1234567</payerId>
7     <senderAddressId>2005887</senderAddressId>
8     <receiverName>Lucie Tauchenova</receiverName>
9     <!-- Kod sluzby, 1 = Classic -->
10    <mainServiceCode>1</mainServiceCode>
11    <additionalServices></additionalServices>
12    <parcels>
13      <parcelReferenceNumber>WS333</
        parcelReferenceNumber>
14      <weight>10.0</weight>
15    </parcels>
16  </shipmentList>
17  <priceOption>WithoutPrice</priceOption>
18 </ship:createShipment>
```

Kód 4.6: Volání funkce CreateShipment()

Webové služby GeoAPI poskytují na rozdíl od výše popsanych služeb Moje DPD pouze dvě metody. Ty slouží k získání informací o zásilkách a případných omezeních, která se týkají některých služeb DPD.

Ke všem webovým službám i aplikačním rozhraním od DPD je k dispozici velmi dobrá online dokumentace v českém jazyce, ve které jsou všechny dostupné služby podrobně popsány.

4.4 Webové služby PPL

Také poslední zkoumaný dopravce – PPL – nabízí službu vyzvednutí zásilky na výdejním místě. E-shopům k tomu rovněž poskytuje seznam těchto výdejních míst. Na rozdíl od předchozích dopravců však tento seznam poskytuje online pouze prostřednictvím webové služby. Ta funguje na protokolu SOAP.

Webová služba obsahuje několik metod. Těmi hlavními jsou:

- `GetKTMList()` - Vrátí seznam všech ParcelShopů pro vybranou zemi, vč. informací o nich.
- `GetKTMDetail()` - Vrátí detail vybraného ParcelShopu dle zadaných parametrů.

Vstupní parametry, se kterými lze služby volat, jsou parametry *customerID* (ID ParcelShopu), *customerDepID* (ID vstupního depa ParcelShopu), *KTMID* (kombinace ID ParcelShopu a PSČ) a *couCode* (kód země). Vstupní parametry jsou volitelné. V případě metody `GetKTMDetail()` se zadává buď *KTMID*, nebo kombinace *customerID* + *customerDepID*. Nejsou-li parametry používány, doporučuje se je nastavit na hodnotu null.

Společnost PPL nemá další veřejně dostupné webové služby ani dokumentaci k nim, jako je tomu např. u DPD nebo Zásilkovny. Stejně jako Česká pošta ani společnost PPL pro účely této práce žádné další materiály, které by se této problematice věnovaly, neposkytla.

4.5 Srovnání webových služeb

V této části kapitoly bude po vzoru 3. kapitoly, Analýza a srovnání možností dopravy zásilek v ČR, ve které byly porovnány přepravní služby společností, provedeno srovnání služeb webových. Do tohoto srovnání budou zahrnuty pouze ty webové služby, které jsou volně přístupné nebo k nim existuje alespoň veřejně dostupná dokumentace, tzn. že byly zmíněny výše v přehledu u jednotlivých dopravců.

V rámci analýzy bylo zjištěno, že všechny čtyři vybrané společnosti poskytují e-shopům webové služby, které slouží pro získání dat o výdejních místech dopravců. Všichni dopravci poskytují data ve formátu XML, Zásilkovna navíc ve formátu JSON a DPD, navíc ve formátu CSV. Struktura dat

se dopravce od dopravce mírně liší. Obsah těchto dat je ale podobný – vždy je u výdejního místa uveden název, adresa a otevírací doba.

U České pošty si musí dát e-shop pozor na to, že seznam výdejních míst pro službu Balík Na poštu je jiný než ten pro službu Balík Do balíkovny. Chce-li e-shop nabízet obě služby, musí mít k dispozici oba seznamy.

Zásilkovna a DPD nabízí e-shopům uživatelsky přívětivější varianty. U Zásilkovny je možné službu integrovat ve formě widgetu, ve kterém je kromě informací o výdejních místech zobrazena také mapa míst a při vyhledávání konkrétního místa uživateli pomáhá funkce našeptávače. U DPD toto chování není implementováno ve formě widgetu v e-shopu, ale přímo na stránkách DPD, kam je uživatel z e-shopu při výběru výdejního místa přeměrován.

Co se týče webových služeb pro podání balíků, provedená analýza ukázala, že nejširší nabídku poskytují opět společnosti Zásilkovna a DPD. Oba dopravci disponují rozsáhlými webovými službami, ke kterým navíc existuje detailní, veřejně dostupná dokumentace. Základní funkcionalita, kterou webové služby zajišťují, je u obou dopravců stejná. Tu tvoří zejména funkce pro vytvoření zásilky, vytištění štítků a získání informací o zásilce. Obě společnosti k těmto klíčovým funkcím poskytují množství doplňkových. U Zásilkovny je možné data z e-shopu předat nejen přes API, ale také importem dat ze souboru ve formátu CSV či XML.

U České pošty nebyly nalezeny dostupné API či webové služby pro podání balíku, příp. dokumentace k nim. Jediným chytrým způsobem, kterým je možné komunikovat s Českou poštou během podání balíku, je prostřednictvím webové aplikace POL, která zajišťuje správu zásilek. V aplikaci je však nutné zásilky buď ručně vytvořit nebo je tam importovat z předem připraveného souboru ve formátu CSV či TXT. V aplikaci je před importováním dat nutné udělat zdlouhavou počáteční konfiguraci odesílatele, adresátů i samotného importu dat. Aplikace je navíc z uživatelského hlediska velmi nepřehledná a neintuitivní.

Společnost PPL nemá kromě výše uvedených webových služeb pro výběr výdejního místa další veřejně dostupné webové služby či aplikační rozhraní. Ve srovnání by tedy skončilo PPL na posledním místě.

V předchozích kapitolách byly analyzovány jednotlivé přepravní společnosti. U každé z nich byly rozebrány jednak služby balíkové, které slouží pro dopravování zásilek, jednak služby webové, jejichž prostřednictvím probíhá komunikace mezi e-shopem a dopravcem. Provedení analýzy bylo předpokladem pro pochopení situace na českém trhu, a tedy i prvním krokem na cestě k návrhu modulu, ve kterém budou využity právě znalosti získané při rozboru společností. V následujících kapitolách se bude práce zabývat již

samotným návrhem, realizací a testování zmíněného modulu pro výběr dopravy.

5 Návrh a implementace modulu

Tato kapitola se věnuje návrhu a implementaci modulu. Modul bude navržen zejména za účelem demonstrace komunikace mezi e-shopem a vybraným dopravcem. V první části kapitoly bude představena funkcionalita modulu, která se skládá z funkcí pro konfiguraci a funkcí pro používání modulu v procesu objednávky. Funkcionalita bude pro snadnější porozumění zobrazena také v UML diagramu případů užití. Jednotlivé případy užití budou detailně popsány v příloze D práce. V druhé části se kapitola zaměří na samotnou implementaci modulu. Budou zde popsány důležité funkce pro frontend a backend a také funkce některých komponent. Pomocí datového modelu bude dále přiblížena struktura databáze modulu. Struktura klíčových částí programu bude v závěru kapitoly ilustrována pomocí UML diagramu tříd.

5.1 Návrh funkcionality

Funkcionalita modulu bude rozdělena na dvě části. První částí je konfigurace, ve které bude nutné modulu nejprve nastavit způsob dopravy, které bude e-shop nabízet svým zákazníkům. Druhou částí je pak funkcionalita spojená se samotným procesem objednávky, ve kterém bude modul zákazníkovi asistovat při výběru vhodné služby pro dopravu zásilky, poskytne mu aktuální seznam výdejních míst vybraného dopravce a nakonec zajistí zpracování objednávky zboží po jejím odeslání.

5.1.1 Konfigurace modulu

Konfigurací modulu je myšleno nastavení služeb dopravců, ze kterých si pak zákazník bude moci vybrat tu, jíž bude chtít doručit zboží z e-shopu. Nastavení dopravců bude dvoufázové. V první fázi si e-shop nadefinuje parametry, podle nichž se zúží výběr dostupných služeb, kterých mohou být na počátku konfigurace až desítky.

Parametry, které si e-shop bude určovat, budou rozdělené do tří sekcí. První sekce se bude týkat hmotnosti a velikosti zásilek. V této sekci bude muset e-shop zvolit kategorii zboží, které nabízí. Na výběr bude mít ze čtyř možností. Nabízí-li e-shop různorodé zboží, pak zaškrtně více možností. Vždy však bude muset zaškrtnout nejméně jednu z nich.

Kategorie zboží budou následující:

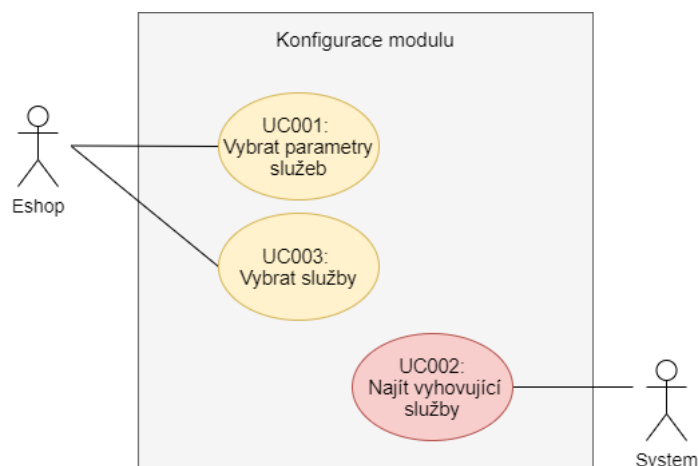
- Hmotnost zásilky do 20 kg, nejdelší strana do 175 cm,
- hmotnost zásilky do 31,5 kg, nejdelší strana do 240 cm,
- hmotnost zásilky do 50 kg, nejdelší strana do 240 cm,
- nadlimitní zásilky (rozměry neomezeny).

Reálný příklad fungování by tedy měl být takový, že e-shop nabízející menší, lehčí sortiment, jako je např. kosmetika, medikamenty, oblečení atd., by měl zaškrtnout primárně variantu č. 1 – položky do 20 kg. Sekundárně by pak mohl zvolit ještě variantu č. 2, aby měl nějakou rezervu. K objednávkám s hmotností nad 30 kg by za běžných okolností docházet nemělo, tudíž volba zbývajících variant je zbytečná. Naopak e-shop nabízející elektroniku, domácí spotřebiče či nábytek by měl zvolit primárně poslední variantu, protože se může stát, že zásilky překročí limit třetí varianty (50 kg, 240 cm).

Druhá a třetí sekce konfigurace je nepovinná. E-shop si ve druhé sekci volí způsob doručení zásilek, tzn. zda chce zákazníkům nabídnout doručování na adresu, na výdejní místa, nebo obojí. Tato volba se může zdát zbytečná, nicméně může se stát, že e-shop bude vyžadovat doručování na výdejní místo, ale některá ze služeb toto nabízet nebude. Proto se musí ze seznamu služeb vyřadit. V poslední, třetí sekci si e-shop volí už jen doplňkové služby – zda by mělo být možné zboží odeslat na dobírku a být možné jej doručit v určitém čase (o víkendu, večer, či zrychleně).

V druhé fázi konfigurace modul vybere z interní databáze všech služeb služby, které vyhovují zadaným parametrům. Modul na závěr uloží vybrané služby. Od této chvíle bude modul pracovat pouze s těmito službami. Popsanou funkcionalitu lze ilustrovat UML diagramem případů užití, který je na obrázku č. 5.1. Na obrázku je vidět, že funkcionalita modulu se skládá ze tří případů užití, z čehož výběr parametrů a výběr služeb provádí e-shop, zatímco nalezení vyhovujících služeb provádí na základě interakce s e-shopem modul sám.

Zde je nutné poznamenat, že modul neřeší právní vztahy mezi e-shopem a přepravní společností. V reálném případě by bylo nutné se po dokončení konfigurace spojit s obchodními zástupci přepravních společností, jejichž služby si e-shop v rámci konfigurace vybere, a domluvit si s nimi smluvní podmínky, příp. si zajistit přístupové údaje do jejich webových služeb. U České pošty by to byly přístupy do aplikace pro online podávání zásilek (aplikace POL), u Zásilkovny pak autentizační údaje pro používání jejího API. Tyto údaje by si následně e-shop uložil v konfiguračním souboru modulu ručně.



Obrázek 5.1: Diagram případů užití konfigurace modulu

5.1.2 Modul v procesu objednávky

V procesu objednávky zboží v e-shopu bude modul využit hned na několika místech. Podle nastavených služeb bude modul kontrolovat maximální povolenou váhu košíku. Podle váhy košíku bude vybírat vhodné služby pro způsob dopravy. Zákazníkovi bude poskytovat možnost výběru doplňkových služeb a aktuální seznamy výdejních míst dopravců. Po odeslání objednávky se také postará o její zpracování.

Maximální povolená váha košíku se bude určovat podle limitu, který bude uveden u vybraných služeb v databázi modulu. Modul vyhledá službu s nejvyšším hmotnostním limitem a ten nastaví jako maximální povolenou váhu košíku. Zákazník e-shopu si poté v rámci jedné objednávky bude moci vložit do košíku pouze zboží do této hmotnosti. To znamená, že jedna objednávka bude představovat jednu fyzickou zásilku. Někteří dopravci sice umožňují posílat vícekusové zásilky, modul ale v rámci snahy o univerzální řešení nebude moci rozlišovat, zda e-shop posílá knihy, které lze rozdělit do více zásilek, nebo nábytek, který rozdělit většinou nelze.

Podle aktuální váhy košíku bude modul vyhodnocovat, které vybrané služby je možné použít pro odeslání této zásilky. Pokud by měl e-shop nastaveno, že jako způsoby dopravy bude nabízet službu A s limitem 10 kg na zásilku, službu B s limitem 20 kg na zásilku a službu C s limitem 31,5 kg na zásilku, a zákazník by měl v košíku zboží vážící 15 kg, modul mu jako způsoby dopravy nabídne službu B a C. Službu A modul vyhodnotí jako nevhodnou pro tuto objednávku.

Vybere-li si zákazník službu, která vyžaduje výběr výdejního místa, modul zákazníkovi poskytne seznam výdejních míst vybraného dopravce. Jsou-li

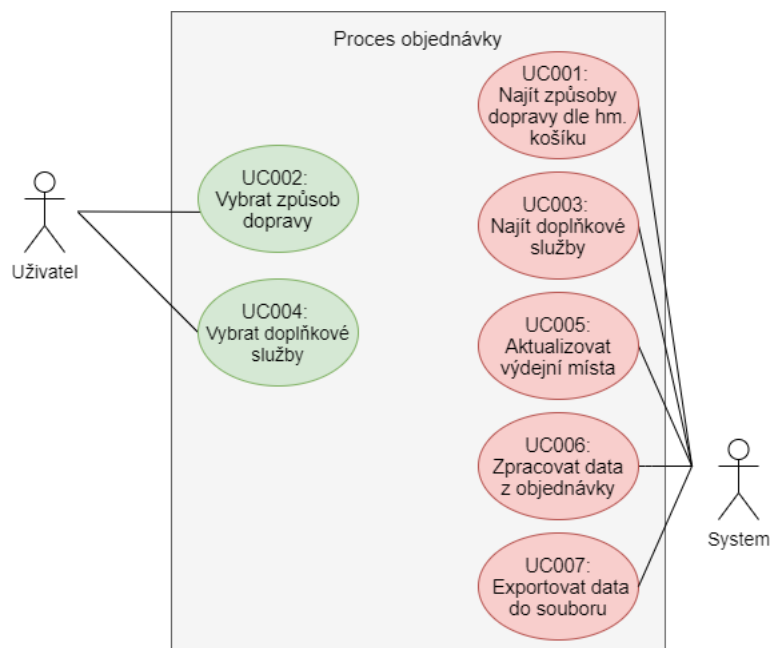
ke zvolené službě navíc dostupné nějaké doplňkové služby, zákazník na to bude upozorněn a bude si z nich moci některou vybrat.

Poté, co zákazník odešle objednávku, modul ji zpracuje. Zpracování objednávky obnáší následující činnosti:

- Validace dat,
- rozpoznání zvoleného dopravce a služby,
- zpracování dat z objednávky způsobem definovaným dopravcem.

Výchozí způsob zpracování dat z objednávky bude export dat do souboru ve formátu CSV s jasně danou strukturou. Tento soubor pak e-shop použije při podání balíku dopravci. Dalším způsobem bude odeslání dat dopravci přes API. Vyskytne-li se během zpracování neočekávaná chyba, nebo nebudou-li data z objednávky validní, objednávka bude stornována. Zákazník bude vždy po zpracování objednávky informován o výsledku operace.

Popsaná funkcionalita je opět zobrazena v UML diagramu případů užití (viz obr. č. 5.2). Z diagramu i z popisu výše vyplývá, že zatímco koncovému zákazníkovi modul příliš funkcností nenabízí, e-shopu, ve kterém je modul integrovaný, poskytuje podporu během celého procesu objednávky, potažmo už i během výběru zboží.



Obrázek 5.2: Diagram případů užití modulu po dokončení konfigurace

5.2 Návrh rozhraní

5.2.1 Návrh uživatelského rozhraní

Ne všechny funkcionality spojené s modulem budou vyžadovat uživatelské rozhraní. Uživatelským rozhraním se bude logicky nutné zabývat pouze u těch případů užití, které jsou v UML diagramech výše navázány na e-shop, resp. na uživatele (viz obr. č. 5.2). Uživatelské rozhraní tedy bude nutné řešit v následujících případech:

- Výběr parametrů a služeb v konfiguraci,
- výběr způsobu dopravy, doplňkových služeb a výdejního místa v objednávce.

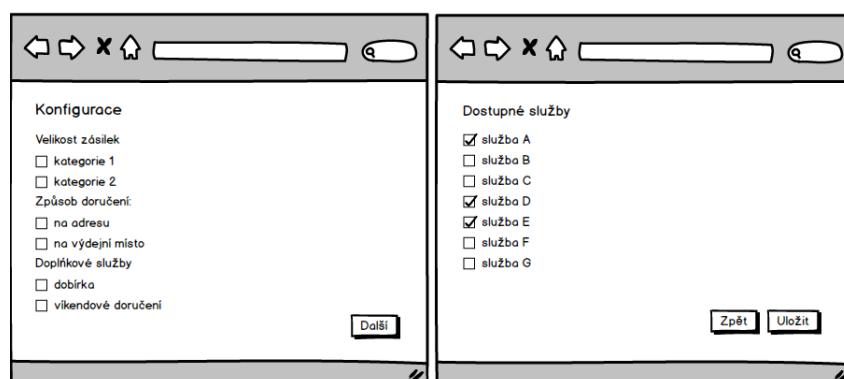
Tyto případy budou řešené pomocí formulářových komponent. Rozložení prvků v jednotlivých formulářích ukazují obrázky (viz obr. č. 5.3 a č. 5.4). Vyobrazené formuláře obsahují typické HTML prvky, kterými jsou:

- *Checkboxlist*:
Checkboxlist bude použit pro výběr parametrů a dostupných služeb v konfiguraci a dále pro výběr doplňkových služeb v objednávce.
- *Radiobuttonlist*:
Radiobuttonlist bude použit pro výběr způsobu dopravy. Checkboxlist je pro tento krok nevhodný, neboť je nutné vybrat pouze jeden způsob, ne více jako tomu bylo v předchozím případě.
- *Text input*:
Text input bude použit pro výběr výdejního místa.
- *Button*:
Buttons budou použity ve všech formulářích pro postup vpřed nebo vrácení se o krok vzad.

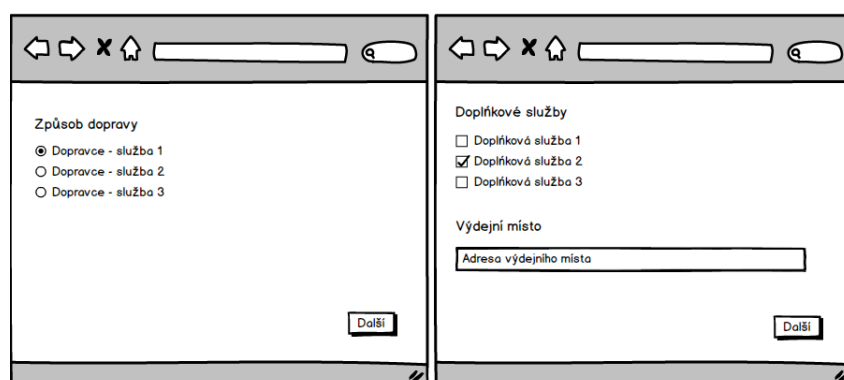
Výsledná podoba GUI bude vždy záviset na e-shopu, který navržené formuláře vzhledově přizpůsobí ostatním částem e-shopu tak, aby vzhled konfigurace i objednávky vypadal konzistentně (např. využitím připravených css tříd).

5.2.2 Návrh aplikačního rozhraní

Nebudou-li data z objednávky zpracována defaultním způsobem – exportem do souboru, budou odeslána dopravci přes aplikační rozhraní. Komunikace



Obrázek 5.3: Návrh uživatelského rozhraní konfigurace modulu

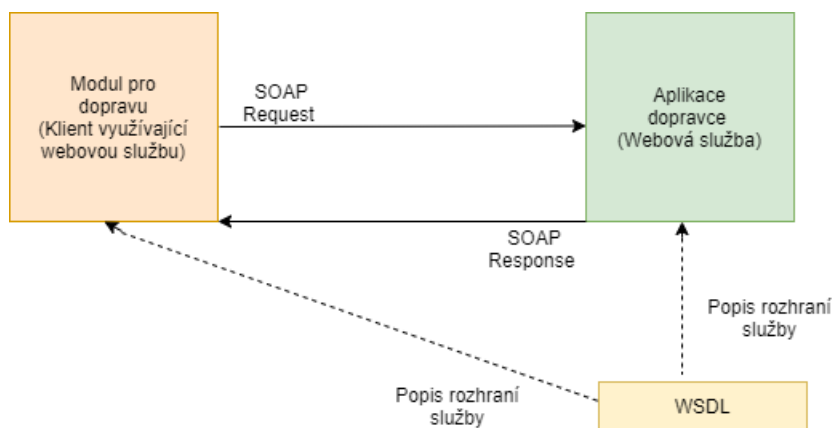


Obrázek 5.4: Návrh uživatelského rozhraní modulu v procesu objednávky

bude probíhat přes protokol SOAP. Vstupy a výstupy funkcí webových služeb budou popsány typicky v jazyce WSDL, který je standardním formátem pro popis rozhraní webové služby. Zprávy (dotazy a odpovědi) mezi aplikacemi se budou přenášet v XML formátu.

Způsob této komunikace ilustruje obrázek č. 5.5. Aby bylo možné tento typ komunikace použít, je důležité, aby každá webová služba byla formálně zapsána v jazyce WSDL. Z popisu poté bude možné vygenerovat SOAP požadavek. Bude-li vybraná webová služba (např. služba pro vytvoření balíku v systému dopravce) zavolána webovým serverem, který obdrží přes HTTP protokol SOAP požadavek od klienta (od modulu integrovaného v e-shopu), webový server spustí webovou službu a požadavek jí předá. Služba požadavek zpracuje a výsledek pak přes server pošle zpět volajícímu klientovi. Výsledkem volání může být platná odpověď (např. ID vytvořeného balíku), ale také výjimka, pokud by požadavek byl ve špatném formátu, nebo data nebyla validní (např. neplatný klíč pro komunikaci s dopravcem).

Při návrhu modulu je kromě struktury komunikace zcela zásadní také



Obrázek 5.5: SOAP komunikace modulu a webové služby dopravy

struktura databáze, ze které bude modul získávat data jak během samotné konfigurace, tak během objednávky i po jejím odeslání.

5.3 Návrh databázové struktury

Modul bude pracovat s relační databází, která bude obsahovat několik tabulek. Modul bude primárně využívat následující dvě tabulky:

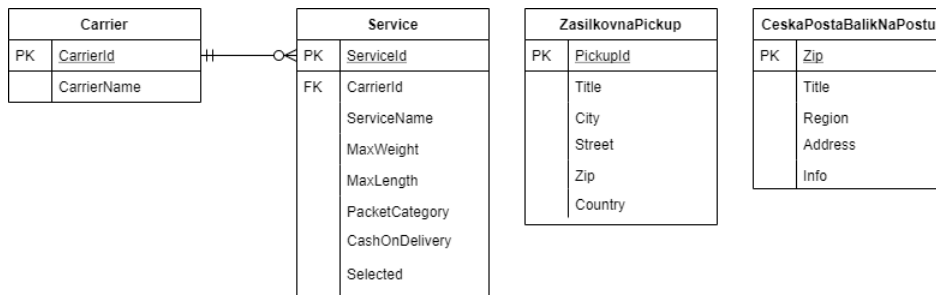
- Tabulka CARRIER
- Tabulka SERVICE

Tabulka Carrier bude reprezentovat seznam dopravců, se kterými lze v ČR spolupracovat při přepravě zboží z e-shopu. Tabulka bude mít pouze dva sloupce: identifikační číslo dopravce (ID) a název dopravce. Primárním klíčem tabulky bude sloupec ID.

Tabulka Service bude obsahovat seznam a informace o dostupných službách dopravců, kteří budou uvedeni v tabulce Carrier. Spojení mezi tabulkami Carrier a Service bude zajištěno relací s kardinalitou 1:N. V této tabulce bude primárním klíčem ID služby, cizím klíčem pak ID dopravce. V tabulce budou dále uložena data jako např. velikostní a hmotnostní limity služeb, dostupnost doplňkových služeb, typ doručení nebo základní cena služby. U každé služby bude mimo jiné uvedeno, které kategorie zásilek lze touto cestou zaslat. Do tabulky se propíše i výsledek konfigurace, tzn. které služby budou v e-shopu nabízeny.

Do dalších tabulek modulu se budou ukládat data o výdejních místech dopravců, kteří poskytují službu doručení na výdejní místo. Struktura těchto tabulek bude záviset na specifikaci určené jednotlivými dopravci. Obecně lze

řící, že tyto tabulky by měly obsahovat informace jako je ID místa, název, adresa, atd. E-shop může tuto strukturu ovlivnit během ukládání dat například tím, že nepoužije všechna poskytnutá data, ale vybere si pouze některá z nich. V diagramu databázového modelu (viz obr. č. 5.6) bude uveden příklad těchto tabulek pro Zásilkovnu a Českou poštu Balík Na poštu (tabulka ZasilkovnaPickup a tabulka CeskaPostaBalikNaPostu). Tyto tabulky jsou na ostatních tabulkách v databázi nezávislé.



Obrázek 5.6: Zjednodušený databázový model interní databáze modulu

Kapitola se bude dále věnovat implementaci modulu, v rámci které budou popsány jednotlivé vrstvy modulu, důležité komponenty a rozhraní.

5.4 Implementace modulu

Jak bylo popsáno dříve v kapitole č. 2.2, modul je implementován v jazyce PHP a ve frameworku Nette. Databáze je řešena pomocí open source databázového systému MySQL, který pro popis dotazů využívá jazyk SQL. Dalšími technologiemi, které byly během vývoje použity, byly zejména JavaScript a AJAX. Kaskádové styly byly řešeny pouze v obecné rovině. To znamená, že zde bylo provedeno pouze rozložení jednotlivých prvků v komponentách dle návrhu. K tomuto účelu byl použit Bootstrap, open source nástroj pro vývoj HTML, CSS a JS.

Ačkoli se architektura modulu drží návrhového vzoru MVC, modul tvoří primárně vrstva Modelu. Vrstvy Controlleru a View jsou v modulu zastoupeny pouze ve formě formulářových komponent, které se řadí na pomezí těchto vrstev. Komponenty jako takové mohou představovat MVC architekturu samy o sobě, protože mají skoro vždy šablonu (tj. část View), funkčně se chovají jako Controllery (Presentery), protože mohou vyhodnocovat vstupy, a navíc v určitých případech mohou obsahovat i nějakou logiku (tj. část Modelu).

5.4.1 Rozvržení vrstvy modelu

Vrstva Modelu tvoří podstatnou část modulu. Model definuje strukturu dat a zajišťuje veškerou business logiku pro práci s nimi. V Modelu se nachází funkce pro práci s databází, s webovými službami dopravců i s e-shopem, do kterého je modul integrovaný.

Zdrojový kód vrstvy Modelu je členěný do následujících jmenných prostorů:

- *Carrier*:
Složka Carrier obsahuje třídy reprezentující jednotlivé dopravce z databáze. Ve třídě každého dopravce se nacházejí funkce pro zpracování dat, které modul obdrží z e-shopu. Konkrétní způsob zpracování se dopravce od dopravce liší. Obecně se ale zpracování řeší vždy ve funkci *processOrder()*, která je vynucena rozhraním *ICarrier*, které každý dopravce implementuje.
- *Exception*:
Složka Exception obsahuje definici výjimek, které mohou v průběhu používání v modulu nastat.
- *Factory*:
Složka Factory obsahuje zdrojové kódy pro vytvoření instancí dopravců. Pro implementaci funkcí byl použit návrhový vzor Továrna (Tovární metoda).
- *PacketItem*:
Složka PacketItem obsahuje strukturu dat. Konkrétně se jedná o definici objektů odesílatele (třída *Sender*), adresáta (*Recipient*), velikosti balíku (*Size*), dobírky (*Cod*) a celého balíku (*Packet*). E-shop by měl tyto struktury využívat pro ukládání dat a následně modulu poslat data z objednávky v tomto formátu.
- *PickUpPoint*:
Složka PickupPoint obsahuje zdrojový kód pro práci s webovými službami dopravců, které slouží pro získání dat o jejich výdejních místech. Kromě funkcí pro získání dat se v kódu nachází i ty pro jejich ukládání do lokální databáze a jejich aktualizaci.
- *Repository*:
Složka Repository obsahuje zdrojové kódy s funkcemi pro práci s databází a podpůrnými funkcemi. z pohledu modulu je to složka, ve které

se nachází významná část aplikační logiky, která je využívána formulá-
řovými komponentami, ale také přímo e-shopem. Nejdůležitější třídou
této složky, resp. celého modulu, je třída *CarrierModel*.

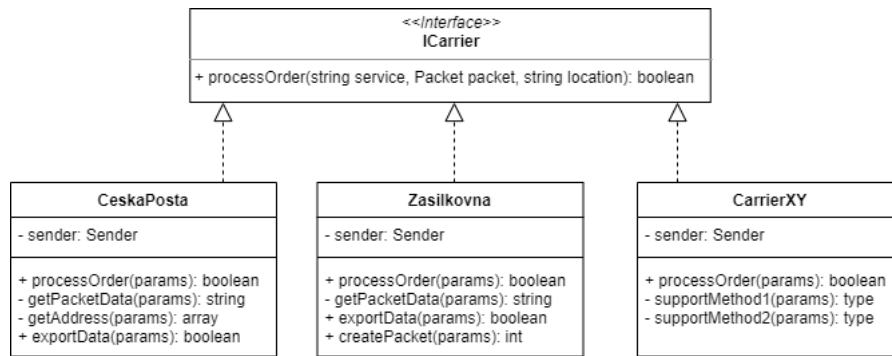
Ve třídě *CarrierModel* se nacházejí pouze funkce pro práci s databází.
Většina z nich je používána buď v konfiguraci modulu nebo v procesu objed-
návky. Typickým příkladem funkce z této třídy je funkce *getServicesByIds()*
(viz kód č. 5.1). Parametrem této služby je pole ID služeb. Podle ID se v da-
tabázi vyhledá, o které služby se jedná a ke kterým dopravcům z tabulky
Carrier služby patří. Funkce jako výsledek vrátí seznam služeb, jehož po-
ložky budou ve tvaru: název dopravce – název služby. Každá položka bude
navíc obsahovat i informaci o ceně služby.

```
1  /**
2  * @param $serviceIds
3  * @return \Nette\Database\ResultSet
4  */
5  public function getServicesByIds($serviceIds) {
6      $sql = 'SELECT service_id, CONCAT(carrier_name," - ",
7          service_name) AS complete_name, price FROM carrier
8          INNER JOIN service ON service.carrier_id = carrier.
9          carrier_id WHERE service_id IN (?)';
10     $params = [$serviceIds];
11     return $this->database->queryArgs($sql, $params);
12 }
```

Kód 5.1: Metoda třídy *CarrierModel* *getServicesByIds()*

V Modelu se nachází jedna třída, která nebyla zařazena do žádného jmen-
ného prostoru. Jde o třídu *ICarrier*. Třída *ICarrier* je rozhraní, které imple-
mentují třídy reprezentující dopravce. Jakým způsobem byla implementace
vyřešena, ilustruje uvedený UML diagram (viz obr. č. 5.7). V diagramu je
zobrazeno reálné provedení implementace rozhraní třídou dopravce *Ceska-
Posta* a třídou dopravce *Zasilkovna*. Třetí třída nazvaná *CarrierXY* je vzor,
podle kterého se implementace řídila. Podle vzoru každý dopravce implemen-
toval svým způsobem jednak funkci *processOrder()*, jednak další podpůrné
funkce, které se starají např. o úpravu instance balíku, generování ID balíku,
formátování dat pro export, následný export dat a volání webových služeb
doprovce.

Třídou, která souvisí s modelem, ale přímo do modelu nepatří, je třída
OrderAcceptance, která představuje vstupní bránu modulu. Dalo by se říci,
že tato třída vytváří jediný Controller aplikace. Zde se nachází funkce *ac-
ceptOrderFromEshop()* (viz kód č. 5.2). Z názvu funkce je patrné, že tato
funkce vytváří spojení mezi modulem a e-shopem. Zde dochází ke zpracování



Obrázek 5.7: UML diagram tříd popisující implementaci rozhraní ICarrier

objednávky z e-shopu a zároveň prostřednictvím návratové hodnoty uvedené funkce je e-shop a následně zákazník informován o tom, zda zpracování objednávky proběhlo v pořádku.

```

1  /**
2  * @param Packet $packet
3  * @param array $serviceData
4  * @return bool order processed successfully
5  */
6  public function acceptOrderFromEshop(Packet $packet, array
   $serviceData) {
7  $this->processServiceData($serviceData);
8  try {
9  $this->carrier = $this->carrierFactory->createCarrier(
   $this->carrierName);
10 } catch (\NonExistingCarrierException $e) {
11 }
12
13 return $this->carrier->processOrder($this->serviceName,
   $this->packet, $location);
14 }
  
```

Kód 5.2: Funkce acceptOrderFromEshop()

I přes to, že jde o relativně významnou funkci, její implementace je velmi jednoduchá. Zavoláním funkce *processServiceData()* zde dojde k rozparsování služby v parametru funkce (zvoleného způsobu dopravy). Služba se rozdělí na název dopravce a název služby. Pomocí návrhového vzoru Továrna se podle jména dopravce vytvoří instance jeho třídy (viz kód č. 5.3), nad kterou je následně zavolána funkce *processOrder()*, které se jako argumenty pošlou název služby a data balíku a která se následně postará o zpracování těchto dat.

```

1  /**
2  * @param $name
3  * @return CeskaPosta|Dpd|Ppl|Zasilkovna
4  * @throws \NonExistingCarrierException
5  */
6  public function createCarrier($name) {
7      switch ($name) {
8          case "Ceska posta":
9              return new CeskaPosta($this->sender);
10         case "Zasilkovna":
11             return new Zasilkovna();
12         case "DPD":
13             return new Dpd();
14         case "PPL":
15             return new Ppl();
16         default:
17             throw new \NonExistingCarrierException();
18     }
19 }

```

Kód 5.3: Zkrácená verze tovární metody createCarrier()

5.4.2 Formulářové komponenty

Druhou podstatnou část modulu tvoří formulářové komponenty. Formulářové komponenty jsou oddělené od vrstvy Modelu, jejich zdrojový kód se nachází ve složce Forms. Všechny formuláře v modulu fungují na stejném principu. Každý formulář je vytvářený svojí továrnou a zobrazovaný svojí šablonou. Třída formuláře obsahuje vždy nejméně dvě funkce. První funkce definuje strukturu formuláře, druhá funkce definuje způsob jeho zpracování po jeho odeslání. K těmto základním funkcím může třída formuláře obsahovat další funkce, které např. zpracují GET parametry, nastaví třídní proměnné atd.

Modul obsahuje čtyři formulářové komponenty:

- ConfigFormFirst
- ConfigFormSecond
- ShipmentFormFirst
- ShipmentFormSecond

První dva formuláře se starají o konfiguraci modulu, druhé dva formuláře zajišťují výběr způsobu dopravy, doplňkových služeb a výdejního místa

v procesu objednávky v e-shopu. Závislosti mezi třídami a způsob implementace komponent ilustruje UML diagram tříd (viz příloha C). Z diagramu je zřejmé, že všechny formulářové komponenty jsou oddělené od abstraktní třídy nazvané *BaseComponent*. Tato komponenta definuje způsob vykreslení formuláře. Pro jeho vykreslení se vždy použije šablona, jejíž název se odvodí z názvu třídy formuláře. Uvedený UML diagram také potvrzuje výše zmíněný fakt, že pro každý formulář existuje továrna, která ho vytváří. Formulář by bylo možné vytvořit i prostřednictvím Nette služby (viz kapitola 2.2). Použití továrny je ale čitelnější a pochopitelnější pro ty, kdo budou formuláře dále využívat. Typicky tedy pro programátora, který bude modul integrovat do e-shopu.

Za zmínku stojí funkce *handleAutocomplete()*, kterou obsahuje pouze formulář *ShipmentFormSecond*. Tato funkce zajišťuje získání správného seznamu výdejních míst dopravce z databáze. Ten se dále zpracuje pomocí JavaScriptu a AJAXového volání přímo ve formuláři v e-shopu. Funguje to tak, že zákazník e-shopu začne do textového pole ve formuláři psát adresu výdejního místa (město, PSČ, ulici, ..). Funkce *autocomplete* podle jeho vstupu vyfiltruje originální seznam všech výdejních míst daného dopravce a zobrazí mu pouze ta výdejní místa, jejichž adresa obsahuje zadaný vstup od uživatele. Funkce *handleAutocomplete()* sama o sobě pouze vyhodnotí, nad kterým seznamem výdejních míst se má zavolat filtrovací funkce *filterByAddress()*, která se nachází vždy ve třídě daného dopravce ve jmenném prostoru *PickupPoint*. Interakci s uživatelem zajišťuje funkce *autocomplete*, která se nachází v javascriptovém souboru *autocomplete.js* (složka *www/js*). Jádro této javascriptové funkce ukazuje kód č. 5.4.

```

1 | $( "#pickup-place" ).autocomplete({
2 |     source: function( request, response ) {
3 |         $.ajax( {
4 |             method: "POST",
5 |             url: autocompleteURL,
6 |             data: { "userInput": request.term },
7 |             success: function( data ) { response( $.map(
8 |                 data, function( item ) {
9 |                     return {
10 |                         label: item.adresa,
11 |                         value: item.adresa,
12 |                         data: item } } ) );
13 |         } } );

```

Kód 5.4: Funkce *autocomplete* pro výběr výdejního místa

5.4.3 Integrační část modulu

Vzhledem k vybrané hlavní technologii - Nette frameworku – nemusely být po celou dobu vývoje modulu řešeny závislosti tříd. Tzv. Dependency Injection (DI), které může vzniknout při nesprávném zacházení s instancemi tříd, je v Nette automaticky ošetřené DI kontejnerem, který byl představený v kapitole č. 2.2. Kromě povinné registrace služeb v konfiguračním souboru *common.neon* bylo během vývoje modulu potřeba vyřešit ještě to, jakým způsobem se budou přenášet závislosti do e-shopů, které budou modul integrovat. Po analýze možností byla nakonec zvolena varianta přenosu závislostí pomocí rozšíření pro DI kontejner. Díky tomu bude možné do e-shopu zaregistrovat stávající *common.neon* soubor a závislosti se tak přenesou s ním. Tento přenos zajistí funkce *loadConfiguration()* (viz kód č. 5.5), která se používá k validaci konfiguračních parametrů a přidání služeb do kontejneru. V konfiguračním souboru e-shopu pak bude potřeba pouze zaregistrovat toto rozšíření (stejně jako služby). K tomu slouží klíčové slovo *extensions*.

```
1 public function loadConfiguration() {
2     // vytvoreni kontejneru
3     $builder = $this->getContainerBuilder();
4     // validace parametru
5     $this->setUpParams();
6     // pridani sluzeb z existujiciho konfiguracniho souboru
7     // common.neon
8     $this->compiler->loadConfig(__DIR__ . '/../config/common.
    neon');
```

Kód 5.5: Rozšíření DI kontejneru

Tato kapitola se věnovala návrhu a implementaci modulu. V následující kapitole se práce zaměří na testování implementované funkcionality. V druhé části kapitoly bude představena demo aplikace, která byla vytvořena pro důkladnější otestování modulu.

6 Testování

V předchozích kapitolách byl popsán návrh a následná implementace modulu. V této kapitole bude popsáno ověření funkčnosti výsledného modulu s využitím unit testů, testovacích scénářů a s vybraným testovacím e-shopem. Testování probíhalo již od samého začátku vývoje modulu. Workflow testování bylo takové, že pro implementovaný kód modulu byly napsány unit testy, které ověřovaly správnost kódu. Modul byl dále pravidelně integrován do vytvořeného testovacího e-shopu. Funkčnost modulu byla na závěr potvrzena pomocí testovacích scénářů.

6.1 Testování pomocí jednotkových testů

Jednotkové testy byly napsané pomocí testovacího nástroje Nette Tester. Nette Tester na rozdíl od třídy PHPUnit, která se běžně využívá pro testování PHP kódu, dovoluje hromadné spouštění všech testovacích skriptů a poskytuje další užitečné třídy, které při selhání čitelně vypíší, v čem byl problém, co měla funkce vypsat a co skutečně vypsala.

Unit testy se nacházejí ve složce *tests*. Testy byly psané formou testovacích případů (test cases). Jeden testovací případ je jinak řečeno testovací třída, která obsahuje testovací skripty (metody) vztahující se k jedné třídě programu. Kromě samotných testovacích metod se mohou v takové třídě nacházet i metody *setUp()* *tearDown()*, které se volají před, resp. po každé testovací metodě. Jejich úkolem je příprava testovacích dat a jejich následné vymazání (úklid po testu). Příklad činnosti těchto metod ukazuje kód č. 6.1 a č. 6.2. V metodě *setUp()* dojde k inicializaci proměnné *\$configFormFirst*, která představuje formulář používaný při konfiguraci modulu. V metodě *tearDown()* se této proměnné po dobehnutí testovací metody nastaví zpátky hodnota null. Tyto dvě pomocné metody se používají proto, aby nedošlo k situaci, kdy bude hodnota proměnné během testu změněna a následně použita v další metodě, což by zapříčinilo pád testu.

```
1 public function setUp() {  
2     $this->configFormFirst = new ConfigFormFirst($this->  
3         carrierModel);  
}
```

Kód 6.1: Pomocná testovací metoda setUp()

```

1 public function tearDown() {
2     parent::tearDown();
3     $this->configFormFirst = null;
4 }

```

Kód 6.2: Pomocná testovací metoda `tearDown()`

Co se týče ostatních testovacích metod, testovací metody se typicky nazývají podle metody, kterou testují. Pro jednu metodu může existovat více testovacích metod (viz kód č. 6.3). Na začátku názvu testovací metody je přidáno klíčové slovo `test`, které je důležité pro hromadné spouštění testů, při kterém se právě podle tohoto slova vyhledají testovací metody, které se mají spustit. V rámci testovacích metod jsou v modulu používány zejména metody `aserci`. Pomocí jednotkových testů bylo odhaleno několik zásadních logických chyb.

```

1 public function testCreateExistingCarrier() {
2     $carrier = "Ceska Posta";
3     $res = $this->carrierFactory->createCarrier($carrier);
4     Assert::type('LuTauch\App\Model\Carrier\CeskaPosta', $res)
5     ;
6 }
7 public function testCreateNonExistingCarrier() {
8     $carrier = "NonExistingCarrier";
9     Assert::exception($this->carrierFactory->createCarrier(
10     $carrier), \NonExistingCarrierException::class);
11 }

```

Kód 6.3: Testovací metody pro tovární metodu `createCarrier()`

V kódu č. 6.3 je zdrojový kód dvou testovacích metod, které prověřují správnost tovární metody uvedené v kódu č. 5.3 (viz kapitola č. 5.4). První metoda `testCreateExistingCarrier()` testuje chování továrny, pokud jí přijde validní název dopravce, v tomto případě `Ceska Posta`. Druhá testovací metoda pak kontroluje vyhození výjimky, pokud továrně přijde nevalidní název (např. `NonExistingCarrier`). Obě metody používají k vyhodnocení testu zmíněnou `aserci` ze třídy `Tester/Assert`.

6.2 Ověření pomocí testovacího e-shopu

Pro potřeby této práce byl vytvořen testovací e-shop, ve kterém bylo možné odzkoušet veškerou funkcionální modulu. V e-shopu lze modul nakonfigu-

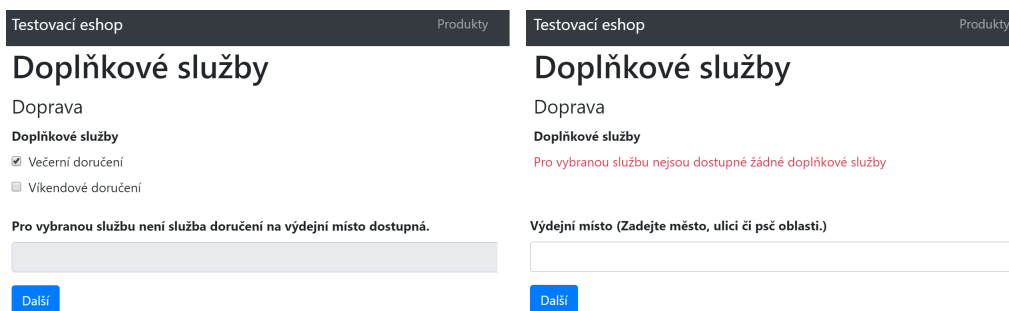
rovat a poté již běžně používat. Z pohledu zákazníka je v e-shopu možné provádět následující činnosti:

- Přidat zboží do košíku / odebrat zboží z košíku.
- Vyplnit kontaktní formulář.
- Vybrat způsob platby.
- Vybrat způsob dopravy.
- Vybrat doplňkové služby, příp. výdejní místo.
- Zobrazit shrnutí objednávky a objednávku odeslat.

Vývoj e-shopu probíhal iterativně a navíc paralelně s vývojem modulu. Integrace modulu do e-shopu tedy byla možná již od začátku vývoje a probíhala po každém větším přírůstku zdrojového kódu modulu. Tento přístup se ukázal jako velmi přínosný, neboť tak bylo možné v rychlosti nalézt nejen syntaktické chyby, ale i chyby v návrhu modulu. Původní návrh modulu se lišil od toho, který byl popsán v kapitole č. 5.1.

Změna v návrhu nastala v části objednávkového procesu. Původně si zákazník nejdříve zvolil preference dopravy (tj. doplňkové služby a způsob doručení zboží – na adresu/na výdejní místo). Podle těchto parametrů byly v databázi služeb vyhledány ty, které nabízejí vybraný způsob doručení a alespoň některou z požadovaných doplňkových služeb. Ty byly v dalším kroku zákazníkovi zobrazeny. Zákazník si z těchto způsobů jeden zvolil a pokud vybraná služba vyžadovala navíc výběr výdejního místa, zobrazil se mu vstup pro zadání adresy místa odběru zboží. Doplňkové služby se už ale dále neřešily. Pokud si v preferencích zákazník vybral, že by chtěl doručit zboží večer nebo o víkendu, neměl už možnost si tuto doplňkovou službu k vybranému způsobu dopravy nikde přidat. Zákazník navíc neměl zaručeno, zda vybraný způsob dopravy obě tyto služby vůbec nabízí.

Během testování v e-shopu se na tuto koncepční chybu přišlo a návrh aplikace byl upraven do stávající podoby, ve které se zákazníkovi zobrazí způsoby dopravy pouze na základě splnění hmotnostního limitu služeb. Doplňkové služby a výdejní místo si zákazník vybírá až v dalším kroku již pro konkrétní službu. Pokud pro vybranou službu nejsou k dispozici žádné doplňkové služby nebo služba nevyžaduje výběr výdejního místa, zákazník je na to upozorněn (viz obr. č. 6.1). Pokud by si zákazník vybral způsob dopravy, který nemá žádné doplňkové služby, ani u něj není potřeba vybrat výdejní místo, tento krok se automaticky přeskočí a zákazník bude přesměrován rovnou na shrnutí objednávky.



Obrázek 6.1: Upozornění zákazníka na nedostupné služby

Náhledy dalších obrazovek e-shopu se nacházejí v příloze E. Další testování v e-shopu probíhalo manuálně s pomocí testovacích scénářů.

6.3 Použití testovacích scénářů

Manuální testování bylo provedeno ze zmíněných testovacích technik až jako poslední. Účelem manuálního testování bylo ověřit robustnost aplikace a odhalit neočekávané pády způsobené dosud nevyzkoušenou kombinací vstupních parametrů aplikace. Pro zvýšení efektivity testování byla vytvořena sada testovacích scénářů, která měla tento typ testování usnadnit. Testovací scénáře vycházely z případů užití, které byly uvedeny v rámci návrhu modulu v první části kapitoly č. 5.1 a jejichž detailní popis se nachází v příloze D.

6.4 Shrnutí testování

I přes to, že průběžná integrace do testovacího e-shopu výrazně přispěla ke kvalitě modulu, ať už pohledu nižšího počtu chybových stavů či vyšší uživatelské přívětivosti a logičnosti systému, nemohly být některé části modulu plně otestovány. Tyto části se týkají přenosu dat z objednávky do systému dopravce. Přenos dat je realizován prostřednictvím API, která někteří dopravci e-shopům nabízejí. Jak správně metody API u jednotlivých dopravců volat typicky popisuje dokumentace, kterou dopravci e-shopům poskytnou. K reálnému použití jejich webových služeb je ale ve většině případů potřeba mít přístupy (ID, uživatelské jméno, heslo, klíč, ...).

Tyto přístupy e-shop získá až po uzavření smlouvy s dopravcem. Bez nich bohužel není možné služby používat a ani testovat jejich funkčnost. Převážně společnosti, které takovými webovými službami disponují, byly v této věci opakovaně kontaktovány a byly požádány o vytvoření testovacích

přístupů do jejich systémů. Kromě DPD společnosti na tuto žádost neodpověděly. Společnost DPD odpověděla s tím, že testovací prostředí bohužel nemá k dispozici a přístup do něj tedy není možný.

V modulu je implementován přenos dat z objednávky přes API od Zásilkovny. Tato implementace sice odpovídá dokumentaci k API, reálné volání služby ale nebylo možné prověřit, protože v modulu není nastavený API Key, který podmiňuje její využití. Ve chvíli, kdy e-shop získá přístupové údaje k webovým službám a v modulu je nastaví, by mělo vše fungovat dle očekávání. API se nicméně nepřetržitě vyvíjí, aplikace je však vytvořena takovým způsobem, aby na tyto změny bylo snadné a rychlé reagovat.

U Zásilkovny a České pošty byl dále implementován export dat z objednávky do souboru, který kromě těchto dvou zmíněných společností využívají i některé další. Export dat je plně funkční.

6.5 Návrhy na zlepšení

Modul poskytuje základní řešení pro dopravu pro e-shop. Pokud by měl být ale nasazen do reálného prostředí e-shopu, bylo by třeba modul rozšířit o další prvky. Mezi ně patří především expedice zboží. V testovacím e-shopu probíhala pouze simulace nákupu zboží a nebylo tak potřeba se zabývat změnami stavů zboží (např. objednáno, zapláceno, připraveno pro expedici, expedováno). V reálném e-shopu by bylo nutné s těmito stavy začít pracovat a upravit podle nich způsob zpracování dat z objednávky.

Objednávka by musela být zpracována ve dvou fázích. V první fázi by došlo pouze k validaci dat a k jejich uložení do databáze. Ve druhé fázi by pak došlo k dokončení procesu, tzn. k exportu dat do souboru či předání dat dopravci. Druhá fáze by byla spuštěna poté, co by zboží přešlo do stavu připraveno pro expedici. Podobným způsobem by bylo nutné rozlišovat mezi způsoby platby. Pokud by si zákazník vybral platbu převodem, muselo by se před přípravou zboží na expedici čekat ještě na příjem platby. Pokud by ale zboží bylo doručováno na dobírku, objednávka by přešla ze stavu objednáno do stavu připraveno pro expedici rovnou. Druhá fáze by mohla být řešena i automaticky, např. s využitím CRONu, který by jednou za čas vybral z databáze objednávky, které by měly nastavený určený stav, a ty by poslal na dokončení zpracování. CRONem by bylo možné řešit také automatickou synchronizaci seznamů výdejních míst dopravců.

Další zlepšení, která by se dala v modulu udělat, by se týkala např. trackování zásilek či rozšíření implementace předávání dat dopravcům, kteří jsou k tomu přizpůsobeni.

7 Závěr

Cílem této práce bylo vytvořit univerzální modul pro řešení dopravy pro e-shop, pro jehož implementaci byl zvolen framework Nette, který je jedním z nejoblíbenějších frameworků programovacího jazyka PHP v ČR. Dalšími technologiemi, které byly vybrány pro vývoj modulu, jsou JavaScript, AJAX nebo protokol SOAP. Vybrané technologie a přístupy, podle kterých se během vývoje postupovalo, byly shrnuty a stručně popsány v kapitole 2.

Předpokladem pro implementaci modulu bylo provedení analýzy možností posílání zásilek v ČR a způsobů komunikace mezi e-shopy a dopravci. Této analýze se věnovaly 3. a 4. kapitola práce. Do analýzy byly zahrnuty čtyři nejčastěji využívané přepravní společnosti na českém trhu, kterými jsou Česká pošta, DPD, PPL a Zásilkovna.

Ze srovnání možností posílání zásilek v ČR, které bylo provedeno v kapitole č. 3, vyplynulo, že cenově nejvýhodnější služby nabízí společnost Zásilkovna, která jako jediný ze soukromých dopravců provozuje také největší množství výdejních míst. Vysokou mírou dostupnosti Zásilkovna konkuruje České poště, která momentálně provozuje výdejních míst nejvíce a která je jakožto státní podnik a držitel poštovní licence povinna udržovat své pobočky i v malých městech a vesnicích, ve kterých ostatní dopravci své zastoupení většinou nemají.

Ve 4. kapitole proběhla u vybraných dopravců také analýza způsobů komunikace s e-shopy. Do způsobů komunikace byly zahrnuty webové služby a rozhraní, které e-shopy využívají jak k získávání dat od dopravce během objednávky (např. platný seznam výdejních míst), tak pro zpracování dat z objednávky a jejich případné předání dopravci (tj. podání balíku). V provedeném srovnání byly nejlépe ohodnoceny společnosti Zásilkovna a DPD, které e-shopům poskytují široké spektrum webových služeb a rozhraní.

Druhá část práce byla věnována již samotnému modulu. V 5. kapitole byl představen návrh uživatelského a aplikačního rozhraní. Návrhy byly podpořeny diagramy případů užití a návrhy grafického uživatelského rozhraní. V neposlední řadě zde byl popsán návrh databázové struktury, který byl ilustrován UML diagramem databázového modelu. Poté byla pozornost přesunuta k implementaci modulu, v rámci které bylo přiblíženo rozvržení vrstvy modelu, která tvoří hlavní vrstvu v jinak třívrstvé architektuře. Následně byly představeny nejdůležitější komponenty, jež byly v modulu využity. Na závěr kapitoly byl objasněn způsob ošetření tzv. Dependency Injection, ke kterému by mohlo dojít po integraci modulu do e-shopu.

Poslední, 6 kapitola, byla zaměřena na testování výsledného modulu. Testování probíhalo s využitím nástroje Nette Tester, s jehož pomocí byly napsané jednotkové testy. Pro účely testování byl dále vytvořen testovací e-shop, ve kterém bylo možné otestovat veškerou funkcionalitu danou dříve popsanými případy užití. Mimo jiné v něm byl prověřen způsob integrace modulu, který je realizován prostřednictvím Composeru. Testování bylo uzavřeno aplikací několika testovacích scénářů, jejichž účelem bylo odladit drobné nedokonalosti v UX.

Všechny body zadání práce tak byly splněny.

Seznam zkratek

API	Application Programming Interface
B2B	Business To Business
B2C	Business To Customer
ČP	Česká Pošta
ČR	Česká republika
CRON	Command Run On
CSV	Comma-separated Values
DPD	Direct Parcel Distribution
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
JS	JavaScript
JSON	JavaScript Object Notation
MVC	Model-View-Controller
MVP	Model-View-Presenter
PHP	Hypertext Preprocessor
PPL	Professional Parcel Service
PSČ	Poštovní směrovací číslo
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
SMS	Short Message Service
XML	Simple Object Access Protocol

Seznam obrázků

2.1	Diagram popisující architekturu MVC	5
2.2	Diagram popisující architekturu MVP	6
3.1	Cenové porovnání modelové situace u České pošty	15
3.2	Cenové porovnání modelové situace u Zásilkovny	17
3.3	Cenové porovnání modelové situace u DPD	19
3.4	Cenové porovnání modelové situace u PPL	20
3.5	Celkový počet výdejních míst v ČR	22
3.6	Srovnání hmotnostních limitů dopravců	25
3.7	Srovnání velikostních limitů dopravců	25
5.1	Diagram případů užití konfigurace modulu	42
5.2	Diagram případů užití modulu po dokončení konfigurace . .	43
5.3	Návrh uživatelského rozhraní konfigurace modulu	45
5.4	Návrh uživatelského rozhraní modulu v procesu objednávky	45
5.5	SOAP komunikace modulu a webové služby dopravce	46
5.6	Zjednodušený databázový model interní databáze modulu . .	47
5.7	UML diagram tříd popisující implementaci rozhraní ICarrier	50
6.1	Upozornění zákazníka na nedostupné služby	57
B.1	Stránka modulu na portále Packagist	70
B.2	Rozdělení kódu modulu	71
C.1	UML diagram tříd formulářových komponent v modulu . . .	72
E.1	Výběr produktů	82
E.2	Košík	82
E.3	Kontaktní formulář v objednávce	83
E.4	Výběr způsobu platby	83
E.5	Výběr způsobu dopravy	84
E.6	Výběr doplňkových služeb	84
E.7	Shrnutí objednávky	85
E.8	Potvrzení odeslání objednávky	85

Seznam tabulek

3.1	Ceník služeb Balík Do ruky a Na poštu	14
3.2	Ceník služby Balík Do balíkovny	14
3.3	Ceník služeb doručení na výdejní místo u Zásilkovny	16
3.4	Ceník služeb doručení na adresu u Zásilkovny	16
3.5	Ceník služeb doručení u DPD	18
3.6	Ceník služeb doručení u PPL	20
3.7	Ceník služby Balík Pro Tebe od PPL	20
3.8	Srovnání služeb z hlediska rychlosti doručení	23
3.9	Výhody a nevýhody dopravců	24
3.10	Srovnání nejvýhodnějších služeb doručení na adresu	26
3.11	Srovnání nejvýhodnějších služeb doručení na výdejní místo	26
D.1	UC001: Vybrat parametry služeb	73
D.2	UC002: Najít vyhovující služby	74
D.3	UC003: Vybrat služby	74
D.4	UC001: Najít dostupné služby dle hmotnosti zboží v košíku	75
D.5	UC002: Vybrat způsob dopravy	76
D.6	UC003: Najít dostupné doplňkové služby, příp. seznam výdejních míst	77
D.7	UC004: Vybrat doplňkové služby, příp. výdejních míst	78
D.8	UC005: Aktualizovat výdejní místa	79
D.9	UC006: Zpracovat data z objednávky	80
D.10	UC007: Exportovat data z objednávky do souboru	80
D.11	UC008: Poslat data dopravci	81
D.12	Popis aktérů v případech užití	81

Seznam ukázek kódu

2.1	Vnoření PHP skriptu do HTML kódu	2
2.2	Definice parametrů v NEON souboru	7
2.3	Definice služby v NEON souboru (1. způsob)	7
2.4	Definice služeb v NEON souboru (2. způsob)	7
2.5	Funkce pro vygenerování služby z DI kontejneru	7
2.6	Nezabezpečená šablona	8
2.7	Automaticky zabezpečená šablona	8
4.1	Struktura XML souboru popisujícího pobočku České pošty .	29
4.2	Struktura JSON souboru popisujícího pobočku Zásilkovny .	31
4.3	Vytvoření balíku v Packeteru API (protokol SOAP)	33
4.4	Vytvoření balíku v Packeteru API (REST API)	34
4.5	Integrace DPD služby výdejních míst	35
4.6	Volání funkce CreateShipment()	36
5.1	Metoda třídy CarrierModel getServicesByIds()	49
5.2	Funkce acceptOrderFromEshop()	50
5.3	Zkrácená verze tovární metody createCarrier()	51
5.4	Funkce autocomplete pro výběr výdejního místa	52
5.5	Rozšíření DI kontejneru	53
6.1	Pomocná testovací metoda setUp()	54
6.2	Pomocná testovací metoda tearDown()	55
6.3	Testovací metody pro tovární metodu createCarrier()	55

Literatura

- [1] *Základní informace - Česká pošta*. [online]. [cit. 02.05.2020]. Dostupné z: <https://www.ceskaposta.cz/o-ceske-poste/profil/zakladni-informace>.
- [2] *Zjednodušení portfolia balíkových služeb - Česká pošta* [online]. [cit. 02.05.2020]. Dostupné z: <https://www.ceskaposta.cz/sluzby/baliky/cr/zjednoduseni-portfolia-balikovych-sluzeb>.
- [3] *Výroční zpráva České pošty 2018* [online]. 2019. [cit. 02.05.2020]. Dostupné z: <https://www.ceskaposta.cz/documents/10180/282479/V%C3%BDro%C4%8Dn%C3%AD+zpr%C3%A1va+2018+podepsan%C3%A1.pdf/9e2d3e9e-7537-00dc-b67d-3f9c32075cc3>.
- [4] CZ, D. *Výdejní místa Pickup* [online]. 2020. [cit. 02.05.2020]. Dostupné z: <https://pickup.dpd.cz/integrace/>.
- [5] *Kdo jsme / DPD CZ* [online]. 2019. [cit. 02.05.2020]. Dostupné z: https://www.dpd.com/cz/business_customers/poznejte_nas/kdo_jsme.
- [6] FOUNDATION, N. *Dependency Injection / Nette Docs* [online]. 2008. [cit. 02.05.2020]. Dostupné z: <https://doc.nette.org/cs/3.0/dependency-injection>.
- [7] FOUNDATION, N. *Průvodce / Tracy debugging tool. Tracy – A Must-Have Debugging Tool for All PHP Developers* [online]. 2008. [cit. 02.05.2020]. Dostupné z: <https://tracy.nette.org/cs/guide>.
- [8] HEUREKA.CZ. *Online nákupy v roce 2019: Češi utratili 155 miliard korun, z výdejních míst se stal fenomén* [online]. 2020. [cit. 02.05.2020]. Dostupné z: <https://onas.heureka.cz/online-nakupy-v-roce-2019-cesi-utratili-155-miliard-korun-z-vydejnich-mist-se-stal-fenomen>.
- [9] HUDSON, P. *PHP in a Nutshell: A Desktop Quick Reference*. O'Reilly Media, 2006. ISBN 978-0596100674.
- [10] KOSEK, J. *PHP a XML*. Grada Publishing a.s., 2009. ISBN 978-8024767123.

- [11] MUNRO, J. *20 Recipes for Programming MVC 3 Faster, Smarter Web Development*. O'Reilly Media, 2011. ISBN 978-1449309862.
- [12] PACKETEA. *Packetery API reference* [online]. 2020. [cit. 02.05.2020]. Dostupné z: <https://docs.packetery.com/03-creating-packets/06-packetery-api-reference.html>.
- [13] PODNIKATEL.CZ. *Pošta, nebo kurýr? Kdo je horší doručovatel balíků z e-shopu?* [online]. 2020. [cit. 02.05.2020]. Dostupné z: <https://www.podnikatel.cz/clanky/posta-nebo-kuryr-kdo-je-horsi-dorucovatel-baliku-z-eshopu/>.
- [14] POREBSKI BARTOSZ, N. L. P. K. *Building PHP Applications with Symfony, CakePHP, and Zend Framework*. Wrox Press Ltd., GBR, 2011. ISBN 978-0470887349.
- [15] *Skupina DPD koupila balíkové služby Geis v Česku a na Slovensku* [online]. 2019. [cit. 02.05.2020]. Dostupné z: <https://byznys.ihned.cz/c1-66671420-dpdgroup-koupila-balikove-sluzby-geis-v-cesku-a-na-slovensku-cenu-transakce-firmy-nezverejnily>.
- [16] *Historical trends in the usage of server-side programming languages for websites* [online]. 2019. [cit. 02.05.2020]. Dostupné z: https://w3techs.com/technologies/history_overview/programming_language.
- [17] *Stav e-commerce v ČR v roce 2019* [online]. [cit. 02.05.2020]. Dostupné z: <https://www.ceska-ecommerce.cz/#dopravy-a-platby>.
- [18] SITEPOINT – LEARN HTML, J. P. R. . R. D. C. *The Best PHP Framework for 2015: SitePoint Survey Results — SitePoint* [online]. 2000. [cit. 02.05.2020]. Dostupné z: <https://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>.
- [19] s.r.o., P. *PPL s.r.o. / O nás* [online]. 2019. [cit. 02.05.2020]. Dostupné z: https://www.ppl.cz/main.aspx?cls=art&tre_id=45&art_id=1.
- [20] TRIVEDI, V. *How to Speak Tech: The Non-Techie's Guide to Key Technology Concepts*. Apress, 2019. ISBN 978-1484243237.
- [21] WODEHOUSE, C. *SOAP vs. REST: A Look at Two Different API Styles* [online]. 2017. [cit. 02.05.2020]. Dostupné z: <https://www.business2community.com/brandviews/upwork/soap-vs-rest-look-two-different-api-styles-01827446>.

- [22] ZÁSILKOVNA. *Kompletní ceník služby Mezi námi* [online]. 2019. [cit. 02.05.2020]. Dostupné z: https://files.packeta.com/web/files/Mezi_nami_kompletni_cenik.pdf.
- [23] ZÁSILKOVNA. *Služba Mezi Námi* [online]. 2019. [cit. 02.05.2020]. Dostupné z: <https://www.zasilkovna.cz/obcane/vyzvednuti-zasilky>.

A Obsah CD

Přílohou této práce je CD obsahující elektronickou formu práce ve formátu PDF včetně zadání a kompletní zdrojové kódy aplikace a kompletní zdrojové kódy testovacího eshopu. Adresářová struktura na disku je znázorněna níže.

- 01_Dokumentace - Adresář se zadáním a elektronickou formou písemné části bakalářské práce
- 02_Zdrojove_kody - Adresář se zdrojovými kódy modulu
- 03_Testovaci_eshop - Adresář se zdrojovými kódy testovacího eshopu

B Uživatelská a programátorská příručka

Uživatelská a programátorská příručka je zde spojena, neboť výsledný modul pro řešení dopravy je určený především pro programátory, nikoliv pro koncové uživatele. V této příručce bude popsána instalace modulu, struktura kódu a nutné úpravy kódu aplikace, do které je modul integrován.

Instalace modulu

Modul byl publikován v největším PHP repositáři pro Composer nazvaném Packagist. Balíčky, které jsou publikovány na portále Packagist, je možné instalovat přímo do projektu. K instalaci je potřeba mít zprovozněný Composer, což je nástroj pro správu závislostí. Tyto závislosti jsou řešeny přes autoloader, který Composer obsahuje.

Samotná instalace modulu je velmi jednoduchá. Stačí si otevřít příkazový řádek z kořenové složky projektu aplikace, do kterého bude modul integrován. Pro instalaci modulu je třeba zadat příkaz:

```
composer require lu-tauch/shipping-module
```

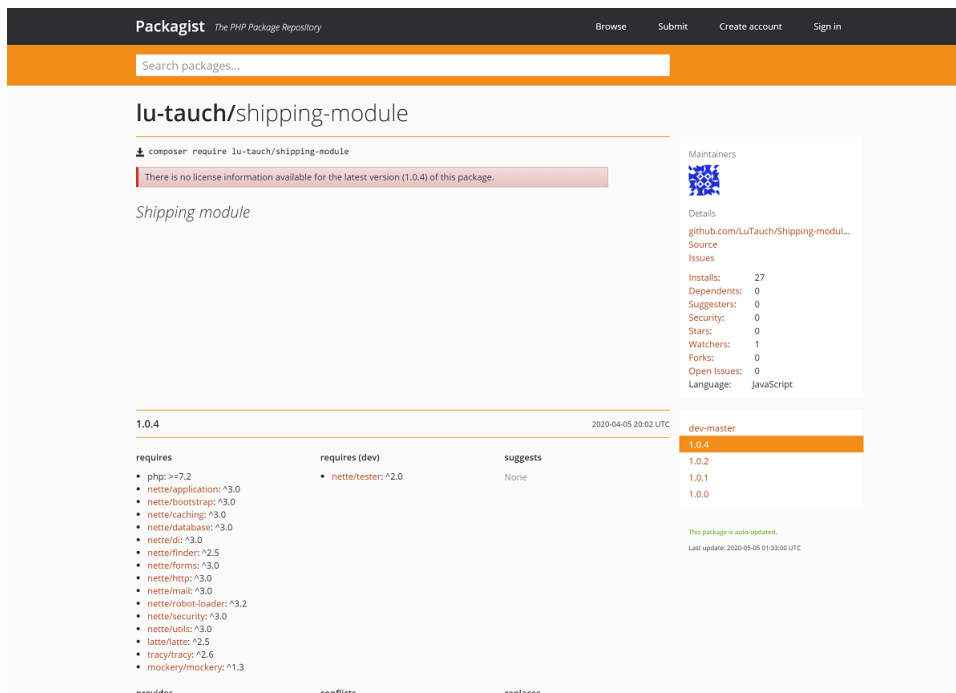
Tímto příkazem dojde k importu modulu včetně všech jeho závislostí, které jsou potřeba (např. Nette Tester, Nette Mockery, atd.).

Seznam všech závislostí a informací o aktuální verzi modulu je možné nalézt na portále Packagist na adrese: <https://packagist.org/packages/lu-tauch/shipping-module>. Stránka modulu je zobrazena na obr. č. B.1.

Portál Packagist automaticky stahuje aktualizace modulu z jeho repositáře na Githubu. K aktualizaci modulu v projektu tak stačí v příkazové řádce zadat příkaz: `composer update lu-tauch/shipping-module`. Tím se zkontrolují a nainstalují dostupné aktualizace.

Adresářová struktura modulu

Po instalaci modulu do aplikace se zdrojový kód modulu bude nacházet na stejné úrovni jako kód ostatních modulů (pluginů), které byly v minulosti do aplikace integrované, tj. `app/vendor`. Složka s kódem modulu je pak ve složce `app/vendor/lu-tauch/shipping-module`.



Obrázek B.1: Stránka modulu na portále Packagist

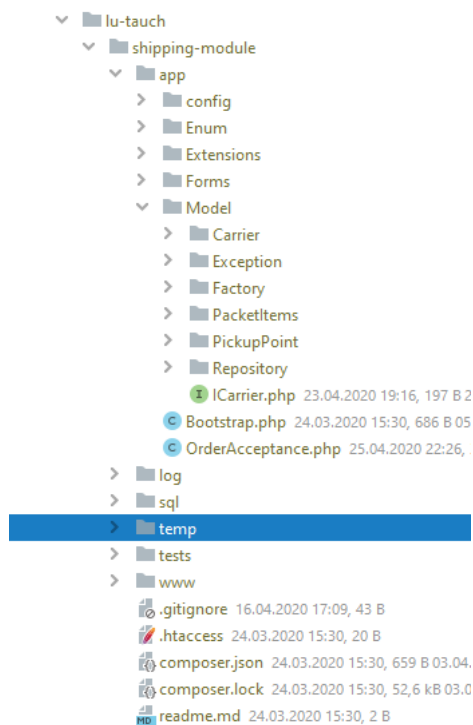
Kód modulu je rozdělený do několika složek (viz obr. č. B.2). Těmi hlavními jsou:

- *app*: Tato složka obsahuje veškerý funkční kód aplikace.
- *sql*: Tato složka obsahuje SQL skripty pro vygenerování databáze modulu.
- *tests*: Tato složka obsahuje testovací skripty.
- *www*: Tato složka obsahuje soubory s kaskádovými styly a javascriptovými soubory modulu.

Struktura složky *app* je blíže popsána v částech této práce nazvaných Rozvržení vrstvy modulu a Formulářové komponenty. Oba tyto oddíly se nacházejí v kapitole č. 5. V případě potřeby je dále možné vygenerovat PHP dokumentaci z kódu modulu (např. s pomocí některého z externích nástrojů v prostředí PHPStorm).

Nutné úpravy v aplikaci

V aplikaci, do které byl modul integrovaný, je nutné upravit některé části kódu. Primárně jde o nastavení dat o e-shopu poté, co proběhne konfigurace



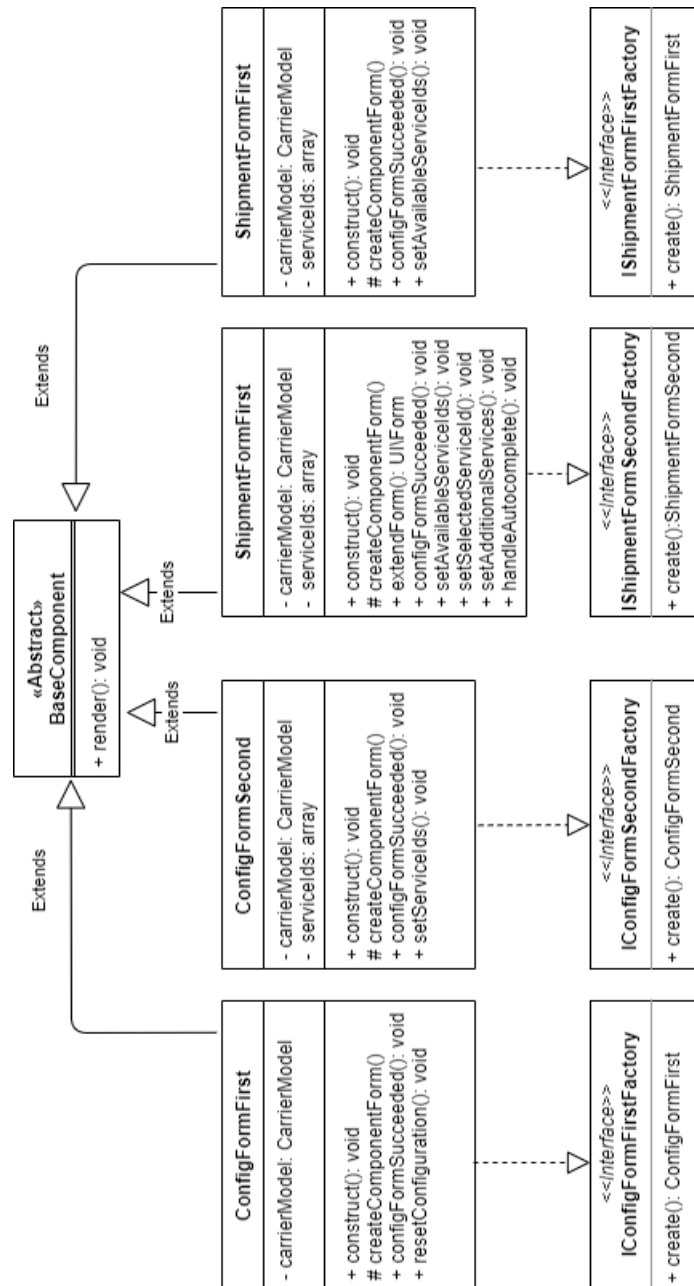
Obrázek B.2: Rozdělení kódu modulu

modulu. Těmito daty se myslí nastavení odesílatele, přístupových údajů pro spojení s API dopravců, apod. Tyto údaje je možné nastavit v konfiguračním souboru *common.neon*, který se nachází ve složce *app/vendor/lu-tauch/shipping-module/app/config*. V něm se také nastavuje cesta k rozšíření (tzv. extensions), které řeší předávání konstant a závislostí z modulu do aplikace.

V aplikaci bude třeba implementovat zobrazení formulářů, které neřeší e-shop, ale modul, tzn. konfigurační formuláře, výběr způsobu dopravy a výběr doplňkových služeb. Pro tyto formuláře je potřeba vytvořit vrstvu Presenteru. Veškeré komponenty v modulu jsou vytvářeny pomocí továren, které se nazývají stejně jako dané komponenty. V tu chvíli nad instancemi těchto komponent stačí zavolat metodu *create()*, která se postará o vytvoření komponenty.

Dále je třeba vyřešit předávání dat z objednávky v e-shopu do modulu, aby mohlo dojít k jejich zpracování. Vstupní branou modulu je třída *OrderAcceptance*, která je na nejvyšší úrovni ve složce *app* v modulu (tzn. není zařazena do žádného dalšího jmenného prostoru). V této třídě řeší příjem dat metoda *acceptOrderFromEshop()*. Té je tedy nutné předání dat z objednávky přizpůsobit.

C Diagram tříd komponent



Obrázek C.1: UML diagram tříd formulářových komponent v modulu

D Případy užití modulu

Seznam případů užití – konfigurace

- UC001: Vybrat parametry služeb
- UC002: Najít vyhovující služby
- UC003: Vybrat služby

Tabulka D.1: UC001: Vybrat parametry služeb

Popis UC	Umožňuje e-shopu definovat parametry služeb dopravců, které chce nabízet svým zákazníkům.
Akteři	E-shop, systém
Vstupní podmínky	Modul musí být integrován do e-shopu.
Standardní průběh	<ol style="list-style-type: none">1. E-shop - administrátor e-shopu - klikne v menu na možnost Nastavení.2. Systém e-shopu zobrazí formulář pro volbu parametrů.3. E-shop musí zvolit alespoň jednu z nabízených možností v sekci Velikost zásilek. Další sekce jsou volitelné. <alt: e-shop nechá sekci Velikost zásilek nevyplněnou4. E-shop po vyplnění formuláře formulář odešle stisknutím tlačítka Další.5. Systém zpracuje jeho odpovědi.
Alternativní průběh	<alt: e-shop nechá sekci Velikost zásilek nevyplněnou>: Systém e-shop vyzve k výběru alespoň jedné možnosti v této sekci.
Výstupní podmínky	E-shop (administrátor) je přesměrován do dalšího kroku konfigurace.

Tabulka D.2: UC002: Najít vyhovující služby

Popis UC	Umožňuje systému najít služby dopravců, které vyhovují zadaným parametrům.
Aktéři	Systém
Vstupní podmínky	Je možné navázat spojení s databází systému.
Standardní průběh	<ol style="list-style-type: none"> 1. Systém obdrží data z formuláře pro výběr parametrů služeb dopravy. 2. Systém naváže spojení s databází a položí dotaz na služby. 3. Databáze vrátí seznam vyhovujících služeb. <alt: Databáze vrátí prázdný seznam> 4. Systém seznam služeb pošle na výstup do druhého kroku konfigurace.
Alternativní průběh	<alt: Databáze vrátí prázdný seznam>: Proběhne přeměrování zpět na první krok konfigurace. E-shop musí zvolit jinou kombinaci parametrů.
Výstupní podmínky	Neprázdný seznam dopravců

Tabulka D.3: UC003: Vybrat služby

Popis UC	Umožňuje e-shopu zvolit si služby, které bude nabízet jako způsob dopravy svým zákazníkům.
Aktéři	E-shop, systém
Vstupní podmínky	E-shop zadal parametry služeb.
Standardní průběh	<ol style="list-style-type: none"> 1. Systém zobrazí uživateli seznam služeb, které může nabízet svým koncovým zákazníkům. 2. E-shop si vybere 1-n z nabízených služeb a formulář odešle stisknutím tlačítka Další. <alt: Uživatel si nevybere žádnou službu> 3. Systém volbu uloží.
Alternativní průběh	<alt: Uživatel si nevybere žádnou službu>: Výběr služeb je povinný. E-shop je znovu vyzván k výběru.
Výstupní podmínky	Zákazníkům e-shopu jsou v kroku výběru dopravy nabízeny uložené služby.

Seznam případů užití – proces objednávky

- UC001: Najít dostupné služby dle hmotnosti zboží v košíku
- UC002: Vybrat způsob dopravy
- UC003: Najít dostupné doplňkové služby, příp. seznam výdejních míst
- UC004: Vybrat doplňkové služby, příp. seznam výdejních míst
- UC005: Aktualizovat výdejní místa
- UC006: Zpracovat data z objednávky
- UC007: Exportovat data z objednávky do souboru
- UC008: Předat data dopravci

Tabulka D.4: UC001: Najít dostupné služby dle hmotnosti zboží v košíku

Popis UC	Umožňuje systému nalézt dostupné služby pro dopravu zásilky.
Aktéři	System
Vstupní podmínky	Uživatel má v košíku alespoň jednu položku
Standardní průběh	<ol style="list-style-type: none"> 1. Uživatel je v rámci objednávkového procesu přesměrován na výběr dopravy. 2. Systém obdrží informace o celkové váze zboží v košíku a po navázání spojení s databází položí dotaz na dostupné služby. 3. Databáze vrátí seznam služeb, jejichž hmotnostní limit odpovídá váze zboží v košíku. 4. Systém odešle seznam služeb na výstup do kroku výběru dopravy objednávkového procesu.
Alternativní průběh	Žádný
Výstupní podmínky	Neprázdný seznam služeb.

Tabulka D.5: UC002: Vybrat způsob dopravy

Popis UC	Umožňuje uživateli zvolit si způsob dopravy.
Aktéři	Uživatel, systém
Vstupní podmínky	Uživatel zatím vyplnil všechny povinné údaje v objednávkovém formuláři.
Standardní průběh	<ol style="list-style-type: none"> 1. Systém zobrazí uživateli seznam služeb, které může využít pro dopravu zboží z e-shopu. 2. Uživatel si vybere jednu z nabízených služeb a formulář odešle stisknutím tlačítka Další. <p><alt: Uživatel si nevybere žádnou službu></p> <ol style="list-style-type: none"> 3. Systém volbu uloží.
Alternativní průběh	<alt: Uživatel si nevybere žádnou službu>: Výběr dopravy je povinný. Uživatel je znovu vyzván k výběru služby.
Výstupní podmínky	Uživatel je přesměrován na výběr doplňkových služeb, příp. výdejního místa.

Tabulka D.6: UC003: Najít dostupné doplňkové služby, příp. seznam výdejních míst

Popis UC	Umožňuje systému nalézt dostupné doplňkové služby pro vybraný způsob dopravy, popř. seznam výdejních míst dopravce.
Aktéři	Systém
Vstupní podmínky	Uživatel má vybraný způsob dopravy.
Standardní průběh	<ol style="list-style-type: none"> 1. Systém obdrží informaci o vybraném způsobu dopravy (službě). 2. Systém naváže spojení s databází a dotáže se na doplňkové služby, které jsou dostupné pro vybraný způsob dopravy. 3. Databáze vrátí seznam služeb. <alt: Databáze vrátí prázdný seznam> 4. Systém pošle seznam služeb na výstup. 5. Pokud si uživatel vybere způsob dopravy, u kterého je nutný výběr výdejního místa, systém následně vyhledá odpovídající seznam míst a pošle ho taktéž na výstup.
Alternativní průběh	<alt: Databáze vrátí prázdný seznam>: Uživatel bude upozorněn, že pro danou službu nejsou dostupné žádné služby. Není-li potřeba ani vybrat výdejní místo, je uživatel rovnou přeměrován na shrnutí objednávky.
Výstupní podmínky	Neprázdný seznam služeb, popř. odpovídající seznam výdejních míst dopravce.

Tabulka D.7: UC004: Vybrat doplňkové služby, příp. výdejních míst

Popis UC	Umožňuje uživateli vybrat si doplňkové služby ke zvolenému způsobu dopravy.
Aktéři	Uživatel, systém
Vstupní podmínky	Uživatel má vybraný způsob dopravy.
Standardní průběh	<p>1. Systém uživateli zobrazí seznam dostupných doplňkových služeb ke zvolenému způsobu dopravy, příp. seznam výdejních míst dopravce, vyžaduje-li to vybraná služba.</p> <p>2. Uživatel si vybere 0-n doplňkových služeb a jedno z výdejních míst dopravce a formulář odešle stisknutím tlačítka Další.</p> <p><alt: Uživatel si nevybere žádné výdejní místo></p> <p>3. Systém jeho volbu uloží.</p>
Alternativní průběh	<p><alt: Uživatel si nevybere žádné výdejní místo>: Vyžaduje-li vybraná služba výběr výdejního místa, uživatel je opakovaně vyzván k jeho výběru.</p>
Výstupní podmínky	Uživatel je přesměrován na shrnutí objednávky.

Tabulka D.8: UC005: Aktualizovat výdejní místa

Popis UC	Umožňuje systému aktualizovat seznam výdejních míst dopravce.
Aktéři	Systém
Vstupní podmínky	Systém obdrží signál k provedení aktualizace seznamu.
Standardní průběh	<ol style="list-style-type: none"> 1. Systém po obdržení signálu k provedení aktualizace naváže spojení s databází a položí dotaz na vymazání obsahu dat o výdejních místech. 2. Databáze vymaže záznamy. 3. Systém naváže spojení s dopravcem a prostřednictvím jeho webových služeb získá aktuální data o výdejních místech. <p><alt: Nastane chyba během stahování nových dat></p> <ol style="list-style-type: none"> 4. Systém nová data zpracuje a položí dotaz k uložení dat do databáze. 5. Databáze data uloží.
Alternativní průběh	<p><alt: Nastane chyba během stahování nových dat>:</p> <p>Dojde-li během přenosu dat k neočekávané chybě, systém provede dotaz na rollback do databáze. Databáze následně obnoví smazaná data.</p>
Výstupní podmínky	Aktualizovaný seznam výdejních míst

Tabulka D.9: UC006: Zpracovat data z objednávky

Popis UC	Umožňuje systému zpracovat data o zásilce, která obdrží z e-shopu.
Aktéři	Systém, uživatel
Vstupní podmínky	Uživatel dokončí proces objednávky stisknutím tlačítka Odeslat.
Standardní průběh	<ol style="list-style-type: none"> 1. Systém obdrží data z objednávky. 2. Podle vybrané služby systém data zpracuje způsobem definovaným dopravcem. <alt: Nastane chyba během zpracování dat> 3. Po zpracování dat odešle zprávu s o dokončení zpracování objednávky zpět do e-shopu.
Alternativní průběh	<alt: Nastane chyba během zpracování dat>: Nastane-li neočekávaná chyba během zpracování dat, uživatel je informován o chybě a o stornování objednávky.
Výstupní podmínky	Uživatel je informován o zpracování objednávky a je přeměrován na stránku s produkty.

Tabulka D.10: UC007: Exportovat data z objednávky do souboru

Popis UC	Umožňuje systému exportovat zpracovaná data z objednávky do souboru .csv.
Aktéři	Systém
Vstupní podmínky	Data z objednávky jsou zpracována.
Standardní průběh	<ol style="list-style-type: none"> 1. Systém upraví strukturu dat do formátu daného dopravcem. 2. Systém upravená data zapíše do požadovaného souboru. <alt: Soubor neexistuje>
Alternativní průběh	<alt: Soubor neexistuje>: Pokud požadovaný soubor neexistuje, systém soubor nejprve vytvoří a až poté do něj zapíše data.
Výstupní podmínky	Soubor .csv s daty z objednávky.

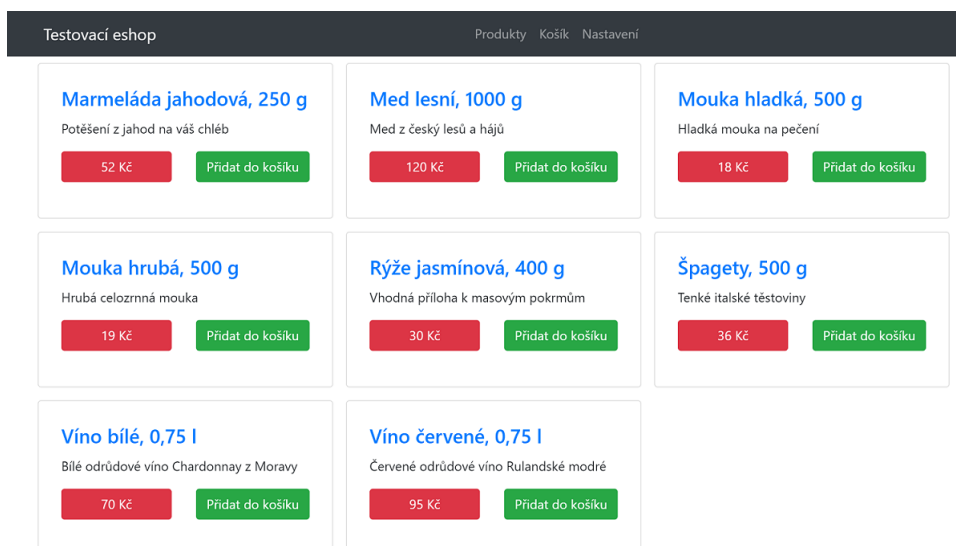
Tabulka D.11: UC008:Poslat data dopravci

Popis UC	Umožňuje předat data ze systému dopravci.
Akteři	Systém
Vstupní podmínky	Data z objednávky jsou zpracována.
Standardní průběh	<p>1. E-shop obdrží signál na zavolání metody API dopravce pro vytvoření zásilky.</p> <p>2. Systém v rámci volání metody předá zpracovaná data z objednávky. V odpovědi získá výsledek volání API dopravce.</p> <p><alt: Nastane chyba během spojení s dopravcem></p>
Alternativní průběh	<p><alt: Nastane chyba během spojení s dopravcem>: Nastane-li chyba během spojení s API dopravce (např. chyba při autorizaci), systém zalogue chybu a upozorní e-shop na vzniklý problém.</p>
Výstupní podmínky	Návratová hodnota v závislosti na API dopravce (např. ID zásilky).

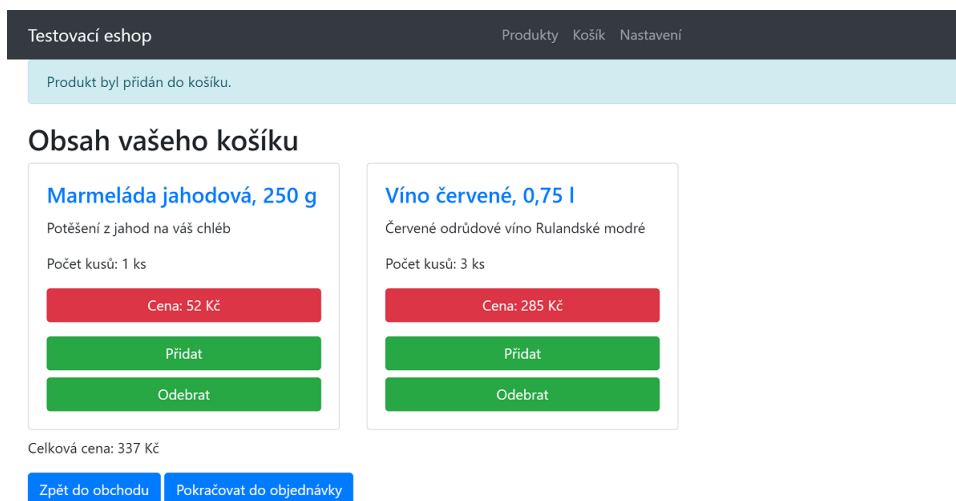
Tabulka D.12: Popis aktérů v případech užití

E-shop	V e-shopu je integrován modul pro dopravu. Úkolem e-shopu je v modulu po integraci udělat počáteční konfiguraci.
Uživatel	Uživatel je zákazník e-shopu, který si chce objednat zboží.
Systém	Systém reprezentuje integrovaný modul pro dopravu.

E Obrazovky testovacího e-shopu



Obrázek E.1: Výběr produktů



Obrázek E.2: Košík

Testovací eshop Produkty Košík Nastavení

Objednávka

Kontakt

Jméno *	Příjmení *	
<input type="text" value="Lucie"/>	<input type="text" value="Tauchenová"/>	
Email *	Předvolba	Telefon *
<input type="text" value="tauchenl@students.zcu.cz"/>	<input type="text" value="+420"/>	<input type="text" value="123456789"/>
Ulice *	Číslo popisné *	Část města
<input type="text" value="Solní"/>	<input type="text" value="20"/>	<input type="text" value="Vnitřní město"/>
Město *	PSČ: *	Stát
<input type="text" value="Plzeň"/>	<input type="text" value="30100"/>	<input type="text" value="Česká republika"/>

Obrázek E.3: Kontaktní formulář v objednávce

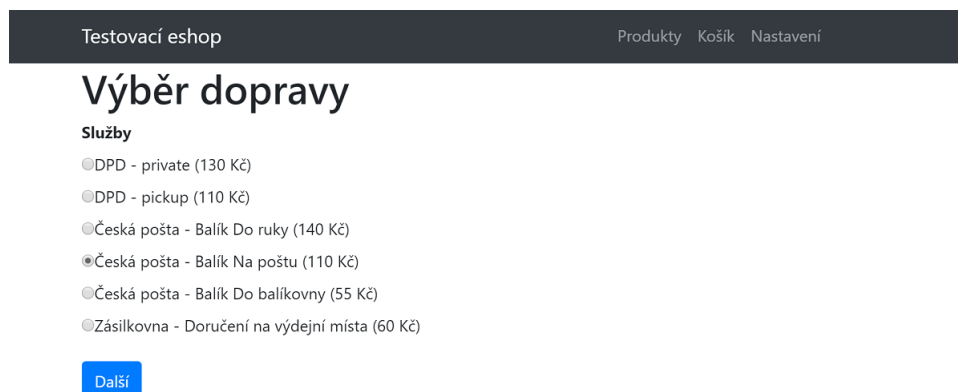
Testovací eshop Produkty Košík Nastavení

Platba

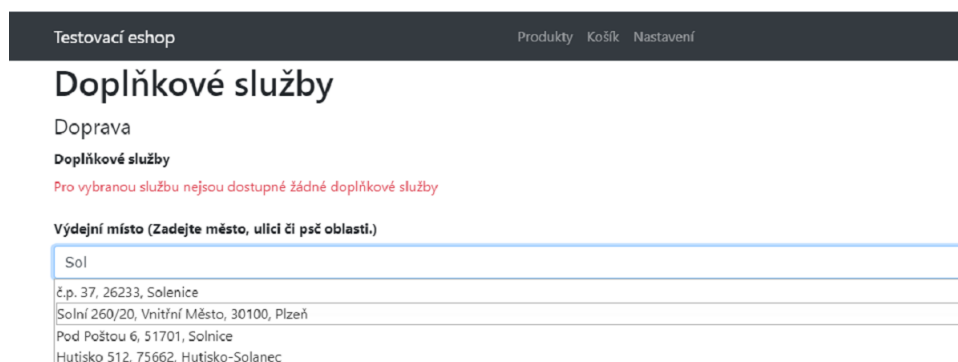
Vyberte prosím způsob platby:

- Bankovním převodem (0 Kč)
- Dobírkou (39 Kč)

Obrázek E.4: Výběr způsobu platby



Obrázek E.5: Výběr způsobu dopravy



Obrázek E.6: Výběr doplňkových služeb

Testovací eshop		Produkty	Košík	Nastavení
Shrnutí objednávky				
Jméno:	Lucie			
Příjmení:	Tauchenová			
Email:	tauchenl@students.zcu.cz			
Telefon:	+420123456789			
Adresa:	Solní 20, Plzeň, Vnitřní město, 30100, Česká republika			
Obsah košíku	Marmeláda jahodová, 250 g - 1 ks Víno červené, 0,75 l - 3 ks			
Cena zboží	337 Kč			
Způsob platby:	Dobírkou (39 Kč)			
Doprava:	Česká pošta - Balík Na poštu (80 Kč); Solní 260/20, Vnitřní Město, 30100, Plzeň			
Celková cena:	456 Kč			
Odeslat objednávku				

Obrázek E.7: Shrnutí objednávky

Testovací eshop		Produkty	Košík	Nastavení
Vaše objednávka byla odeslána. ×				
<p>Marmeláda jahodová, 250 g</p> <p>Potěšení z jahod na váš chléb</p> <p>52 Kč Přidat do košíku</p>	<p>Med lesní, 1000 g</p> <p>Med z český lesů a hájů</p> <p>120 Kč Přidat do košíku</p>	<p>Mouka hladká, 500 g</p> <p>Hladká mouka na pečení</p> <p>18 Kč Přidat do košíku</p>		
<p>Mouka hrubá, 500 g</p> <p>Hrubá celozrnná mouka</p> <p>19 Kč Přidat do košíku</p>	<p>Rýže jasmínová, 400 g</p> <p>Vhodná příloha k masovým pokrmům</p> <p>30 Kč Přidat do košíku</p>	<p>Špagety, 500 g</p> <p>Tenké italské těstoviny</p> <p>36 Kč Přidat do košíku</p>		
<p>Víno bílé, 0,75 l</p> <p>Bílé odrůdové víno Chardonnay z Moravy</p> <p>70 Kč Přidat do košíku</p>	<p>Víno červené, 0,75 l</p> <p>Červené odrůdové víno Rulandské modré</p> <p>95 Kč Přidat do košíku</p>			

Obrázek E.8: Potvrzení odeslání objednávky