



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Optilynx frontend - Informační systém pro optiky
Student:	Bc. Jaroslav Hrách
Vedoucí:	Ing. Jiří Hunka
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2018/19

Pokyny pro vypracování

Navrhňte, realizujte a optimalizujte frontend informačního systému pro středně velkou oční optiku, který bude umožňovat především komplexní správu zakázek v podobě výroby brýlí dle specifikace očního lékaře a požadavků zákazníka, správu zákazníků, skladu, faktur a dodacích listů či kompletní pokladnu pro prodejnu.

Seznamte se s potřebami oční optiky, jako zdroj informací využijte společnost OČNÍ OPTIKA BENEŠOV s.r.o., spolupracujte s Bc. Filipem Glazarem, který řeší backend informačního systému.

Na základě již existujícího mockupu a zjištěných potřeb oční optiky navrhňte a realizujte první verzi frontendu.

Zvolte vhodné testování první verze. Minimálně realizujte uživatelské testování u zaměstnanců optiky.

Na základě testů navrhňte vhodné uživatelské i funkční optimalizace frontendu.

Optimalizace implementujte do druhé verze a ověřte jejich kladný přínos v reálném nasazení.

Zhodnoťte celkový stav aplikace, navrhňte budoucí vylepšení se zaměřením na uživatelskou část aplikace.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 15. února 2018



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

Diplomová práce

OptiLynx frontend – Informační systém pro optiky

Bc. Jaroslav Hrách

Katedra softwarového inženýrství

Vedoucí práce: Ing. Jiří Hunka

9. května 2018

Poděkování

Rád bych poděkoval všem, kteří mi pomáhali při vzniku této práce. V první řadě Ing. Jiřímu Hunkovi, vedoucímu mé diplomové práce, za poskytnuté náměty, cenné rady a spolupráci při závěrečném testování. Dále Bc. Filipu Glazarovi za jeho ochotu a velmi intenzivní spolupráci. Děkuji také všem ostatním, kteří se na tomto projektu podíleli. Závěrem bych chtěl poděkovat svým nejbližším, rodině a přátelům, kteří mne podporovali během celého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, avšak pouze k nevýdělečným účelům. Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 9. května 2018

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2018 Jaroslav Hrách. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Hrách, Jaroslav. *OptiLynx frontend – Informační systém pro optiky*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Tato diplomová práce si klade za cíl vytvořit frontend informačního systému pro středně velkou oční optiku. Nejprve je provedena analýza současného stavu. Na základě analýzy jsou zvoleny vhodné technologie pro implementaci a tvorbu uživatelského rozhraní. Poté je detailně popsán vývoj aplikace společně s popisem uživatelského rozhraní využívajícího principů Material Design. Závěrem je provedeno uživatelské testování se skutečnými uživateli přímo v očních optikách.

Klíčová slova Informační systém, web, návrh, uživatelské rozhraní, Angular, Material Design.

Abstract

The aim of this master's thesis is a creation of a frontend of an informative system for a medium-sized eye optics. At first, an analysis of a current state is made. Based on the analysis, an appropriate technologies for an implementation and creation of a user interface are chosen. It is then followed by detailed description of the application development including user interface, which follows Material Design principles. In conclusion, a user testing with real users right in the eyes optics is made.

Keywords Information system, web, design, user interface, Angular, Material Design.

Obsah

Úvod	1
1 Seznámení s požadavky oční optiky a provedení analýzy	3
1.1 Seznámení s doménou očních optik	3
1.2 Analýza společnosti OČNÍ OPTIKA BENEŠOV s.r.o.	4
1.3 Uživatelské rozhraní	6
1.4 Charakteristika person	7
1.5 Obecné nařízení o ochraně osobních údajů	10
1.6 Analýza konkurenčních systémů pro optiky	11
1.7 Analýza drátěného modelu systému OptiLynx	13
2 Návrh frontendu informačního systému pro optiky	17
2.1 Volba typu aplikace	17
2.2 Volba přístupu	18
2.3 Zvolené technologie	20
2.4 Komunikační rozhraní	27
2.5 Uživatelské rozhraní	30
3 Realizace frontendu informačního systému OptiLynx	33
3.1 Časový průběh realizace	33
3.2 Implementace ve frameworku Angular	34
3.3 Tvorba uživatelského rozhraní	40
4 Testování informačního systému OptiLynx	53
4.1 Počáteční testování při reálném provozu	53
4.2 Uživatelské testování nové verze	54
4.3 Oční optika Sedlčany	56
4.4 Oční optika Benešov	58
4.5 Oční optika Vlašim	60
4.6 Závěr testování	61

Závěr	63
Výhled do budoucnosti	63
Literatura	65
A Seznam použitých zkratk	71
B Finální verze informačního systému OptiLynx	73
C Obsah příloženého média	83

Seznam obrázků

1.1	Newton	11
1.2	OpticEvidence	12
1.3	Tiss Optic	13
1.4	Drátěný model – Sklad	15
1.5	Drátěný model – Zakázka	15
2.1	MPA vs. SPA	19
2.2	Schéma frameworku Angular	26
2.3	Webpack – ilustrace funkcionality	27
3.1	OptiLynx – Pokladna	41
3.2	OptiLynx – Zakázka	42
3.3	OptiLynx – Pokladna	44
3.4	OptiLynx – Rozbalovací nabídka	45
3.5	OptiLynx – Platební karta	45
3.6	OptiLynx – Zákazník	46
3.7	OptiLynx – Čočky	47
3.8	OptiLynx – Poukaz	48
3.9	OptiLynx – Rekapitulace zakázky	49
3.10	OptiLynx – Dialogové okno	49
3.11	OptiLynx – Snackbar	50
3.12	OptiLynx – Kalendář u data	51
4.1	Poukaz na brýle a optické pomůcky	55
B.1	OptiLynx – Přihlášení	73
B.2	OptiLynx – Odhlášení	73
B.3	OptiLynx – Pokladna	74
B.4	OptiLynx – Pokladna – rozbalovací nabídka	74
B.5	OptiLynx – Pokladna – platební karta	75
B.6	OptiLynx – Zákazníci	75

B.7 OptiLynx – Detail zákazníka	76
B.8 OptiLynx – Zakázky	76
B.9 OptiLynx – Detail zakázky	77
B.10 OptiLynx – Sklad	78
B.11 OptiLynx – Detail položky	78
B.12 OptiLynx – Sklad – Přesun	79
B.13 OptiLynx – Sklad – Naskladnění	79
B.14 OptiLynx – Sklad – Párování faktur	80
B.15 OptiLynx – Sklad – Detail faktury	81

Seznam zdrojových kódů

2.1	Struktura zakázky ve formátu JSON	29
3.1	Observable pro získání všech zakázek z API zdroje	36
3.2	Metoda pozorující Observable	37
3.3	Ukázka datového modelu zakázky v Angularu	37
3.4	Reactive form u zákazníka	39
3.5	Angular Pipe – filtr převádějící číslo na českou měnu	40
3.6	Univerzální potvrzovací modální okno	49

Seznam tabulek

2.1 Srovnání MPA a SPA	21
3.1 Srovnání Reactive Forms a Template-driven Forms	39

Úvod

„Dejte brýle Woodymu Harrelsonovi v Neslušném návrhu a bude vypadat jako architekt. Nebo je dejte Denise Richardsové a stane se z ní paleontoložka,“ [1] vtipkuje americká komička Tina Feyová na téma brýlí. Brýle mají zvláštní moc a v dnešní době je nosí mnoho lidí. Ať už z prostého důvodu, že bez nich prostě nic nevidí, nebo jen jako módní doplněk. Takové brýle si člověk musí nejprve vyzkoušet, což ho zavede pravděpodobně do nějaké oční optiky. Téměř každá oční optika v dnešní době používá nějaký informační systém – pro evidenci zákazníků, správu zakázek či třeba pro práci se skladem. A právě touto problematikou se bude zabývat následující práce.

Ambicí této diplomové práce je vytvořit informační systém pro středně velkou oční optiku, který bude moderní, přehledný a intuitivní. Pro dosažení takového cíle bude nejprve provedena analýza, při které bude zjištěno, jaký je současný stav a jakým směrem se ubírat. Vzhledem ke komplexnosti dané problematiky se tato práce zaměří výhradně na část, kterou uvidí koncový uživatel – tedy na frontend informačního systému.

Aby byl informační systém použitelný a konkurenceschopný, bude připraven pečlivý návrh zahrnující výběr vhodných technologií společně s žádoucím uživatelským rozhraním. Poté bude informační systém implementován. Pro dokončení uživatelského rozhraní proběhne uživatelské testování se skutečnými uživateli přímo v očních optikách.

Seznámení s požadavky oční optiky a provedení analýzy

Ještě před samotným návrhem a následnou implementací informačního systému pro oční optiky bylo potřeba danou oblast prozkoumat a vyvodit z ní patřičné závěry. Tím se bude zabývat tato kapitola. Nejprve se zaměří na oční optiky z obecného hlediska. Zjistí, jaká je současná situace na trhu očních optik a zda se jedná o perspektivní oblast pro vývoj informačního systému. Zanalyzuje společnost, pro kterou bude informační systém vyvíjen, čímž se vyzkoumají požadavky a potřeby uživatelů, již by ho v budoucnu používali. Poté definuje typické uživatele systému, čímž usnadní proces následného návrhu. Následně zanalyzuje už hotová řešení konkurence. Závěr kapitoly se věnuje již existujícím drátěným modelům, které společně vytvořili studenti Bc. Filip Glazar a Bc. Oldřich Malec a které poskytli k této práci jako podklady k tvorbě uživatelského rozhraní.

1.1 Seznámení s doménou očních optik

Oční optikou se rozumí specializované oddělení, které se zaměřuje na veškeré služby spojené s lidským zrakem. Zpravidla se jedná o maloobchodní prodejnu brýlí a dalších optických pomůcek a přístrojů. Podle Ivana Vymyslického [2], viceprezidenta Společenstva českých optiků a optometristů, bylo k roku 2016 na českém trhu kolem 1600 očních optik. Český trh očních optik je až na francouzský řetězec GrandOptical doménou výhradně tuzemských společností. Dominují mu velké řetězce FOKUS optik a Eiffel optic, které mají po celé republice desítky poboček. Stále však existuje mnoho menších či středně velkých očních optik, které nabízejí své služby na lokální úrovni ve vybraných městech, kde mají zpravidla jednu či více provozoven, ale svým počtem se rozhodně nemohou rovnat velkým řetězcům.

V posledních letech lidé velmi často nakupují brýle i kontaktní čočky

prostřednictvím internetu, v drogeriích, nebo dokonce u benzinových pump. Nicméně řada z nich stejně nakonec skončí u optika, protože potřebuje odbornou péči a vhodné doporučení od očního specialisty při výběru nových brýlí či jiných optických pomůcek. Vzhledem k zvyšující se délce dožití a stárnutí populace přibývají lidé, kteří by měli nosit brýle. Vymyslický se domnívá, že v roce 2020 bude brýle potřebovat zhruba 60 procent lidí. To je prý důsledkem stále častějšího používání elektroniky, například počítačů a mobilních telefonů, které má neblahé účinky na lidský zrak. Pokud se naplní Vymyslického predikce a vezme se v úvahu rostoucí poptávka po brýlích, dá se považovat vývoj informačního systému pro oční optiky za perspektivní.

1.2 Analýza společnosti OČNÍ OPTIKA BENEŠOV s.r.o.

Jako zdroj informací jsem si vybral společnost OČNÍ OPTIKA BENEŠOV s.r.o. [3], poněvadž pro ni byl vyvíjen informační systém OptiLynx. Společnost funguje přes dvacet pět let a dá se zařadit do skupiny středně velkých očních optik.

Firma začínala v roce 1991 ještě pod názvem **Oční optik Šafanda**, kdy majitel František Šafanda vydražil státní provozovnu oční optiky ve Vlašimi. Po roce následovala dražba státní provozovny v Benešově u Prahy. Poté došlo k otevření další optiky v Sedlčanech a provozování ambulantních služeb v Týnci nad Sázavou a Voticích. Až teprve v roce 1999 došlo k přeměně firmy v právnický subjekt pojmenovaný **OČNÍ OPTIKA BENEŠOV s.r.o.**

Oční optika v současnosti disponuje pěti pobočkami a zaměstnává 15 pracovníků, kteří splňují nejvyšší odborná kritéria pro práci v oboru oční optiky a mají potřebné praktické i teoretické znalosti. Hlavními nabízenými službami je poradenství a pomoc při výběru odpovídající korekční pomůcky a měření zraku zaměstnanci. Oční optika dále nabízí produkty:

- **Brýlové čočky** od všech výrobců čoček.
- **Brýlové obruby** od renomovaných značek až po domácí výrobce.
- **Sluneční brýle** pro sport i volný čas. Lze nechat vyhotovit dioptrické sluneční brýle, polarizační a zrcadlové vrstvy.
- **Kontaktní čočky** a roztoky pro péči o čočky.

1.2.1 Základní procesy oční optiky

Proces při obsluze zákazníka, který si chce koupit nějaké zboží:

1. Přidat zboží na pokladnu

2. Provést platbu
3. Vytisknout účtenku
4. Předat zaplacené zboží zákazníkovi

Proces při obsluze zákazníka, který si chce pořídit nové brýle:

1. Vybrat zákazníka (pokud zákazník v systému neexistuje, vytvořit nového zákazníka)
2. Vytvořit zakázky pro zákazníka
3. Pokud je potřeba, přidat poukaz od lékaře k zakázce a uplatnit všechny slevy
4. Pokud to zakázka vyžaduje, vzít si od zákazníka zálohu
5. Naskladnit zboží ze zakázky
6. Přidat zakázku na pokladnu
7. Provést platbu / doplatit zakázku
8. Vytisknout účtenku
9. Předat zaplacené zboží zákazníkovi

Proces naskladňování nového zboží:

1. Vytvořit v systému všechny položky, které jsou nové
2. Přidat položky, které je potřeba naskladnit, do seznamu naskladnění
3. Pokud je potřeba, spárovat konkrétní naskladnění s fakturou

1.2.2 Potřeby oční optiky

- Informační systém by měl bez problémů provádět všechny výše uvedené procesy.
- Informační systém by měl být přehledný a jednoduchý. Neměl by obsahovat žádné zbytečné prvky a měl by uživateli, který ho bude používat, usnadnit práci.
- Informační systém by neměl být závislý na platformě.
- Informační systém by pro svůj chod neměl vyžadovat externí knihovny.
- Informační systém by měl být dostupný odkudkoliv a vždy by měl disponovat aktuálními daty.

- Vzhledem k tomu, že jsou pracovníci zvyklí na starý informační systém Newton (viz 1.6.1), bylo by dobré to při návrhu zohlednit.
- Společnost OČNÍ OPTIKA BENEŠOV s.r.o. byla roku 2008 zařazena do programu **Hoya partner**, což znamená, že má v sortimentu nabízeného zboží mnoho čoček od společnosti Hoya. To je potřeba vzít na vědomí a uzpůsobit informační systém tak, aby upřednostňoval zboží od tohoto dodavatele.

1.3 Uživatelské rozhraní

Uživatelské rozhraní (anglicky user interface, zkráceně UI) se dá definovat jako prostor, ve kterém dochází k interakci mezi uživatelem a strojem [4]. Problematika uživatelského rozhraní má široký záběr a dá se aplikovat na jakákoliv zařízení vytvořená člověkem. V případě této diplomové práce bude za stroj považován počítač. Princip je následující – uživatel se snaží přes nějaký vstupní kanál (např. na stole pomocí klávesnice či myši, přímo v zařízení za pomoci internetového prohlížeče) komunikovat s počítačem a sdělovat mu své požadavky a potřeby; počítač je musí správně vyhodnotit a na daný vstup vrátit uživateli vhodný výstup, nejlépe správnou odpověď v takové řeči, které daný uživatel porozumí.

Uživatelské rozhraní by mělo být navrženo tak, aby umožňovalo uživateli, který s rozhraním komunikuje, efektivní a bezstarostnou práci. Uživatel by vůbec neměl přemýšlet o tom, jak s daným rozhraním komunikovat, vše by pro něj mělo být intuitivní a zřejmé, což je gró této diplomové práce.

1.3.1 Heuristická analýza

Heuristická analýza je jednou ze základních a velmi oblíbených metod testování použitelnosti interaktivních rozhraní [5]. Používá se k odhalování slabých míst a chyb daného rozhraní. Metoda spočívá v tom, že odborníci na použitelnost procházejí dané rozhraní a snaží se ho vyhodnotit na základě nějakých předem definovaných pravidel. V kontextu použitelnosti se nejčastěji aplikuje deset pravidel, které vydefinoval dánský profesor, informatik a odborník na použitelnost Jakob Nielsen.

Testování by mělo provádět více odborníků, aby se podařilo odhalit všechny chyby a problematická místa rozhraní. Počet odborníků se liší v závislosti na požadavcích systému a také očekávaném zisku, obvykle se však počet testujících odborníků pohybuje kolem tří až pěti [6].

Všech deset pravidel použitelnosti od Jakoba Nielsena, která budou zohledněna při návrhu informačního systému OptiLynx, zní následovně:

Viditelnost stavu systému Systém by měl vždy průběžně informovat uživatele o tom, co se děje. Neměl by zůstat zamrzlý a nereagovat na uživatelské vstupy.

Shoda mezi systémem a realitou Systém by měl zachovat konvence reálného světa, mluvit jazykem uživatelů a používat fráze a slova, kterým člověk rozumí.

Minimální zodpovědnost Uživatelé při práci chybují, a proto by vždy měli mít možnost vrátit se zpět do předchozího stavu.

Konzistence a standardizace Měly by se dodržovat obecné konvence, to, na co jsou uživatelé zvyklí. Uživatelé by totiž neměli přemýšlet o tom, zda různé pojmy, situace či akce znamenají to samé.

Prevence chyb Pokusit se zamezit, nebo alespoň minimalizovat počet chybových hlášení, která by se uživateli mohla zobrazit.

Rozpoznání namísto vzpomínání Uživatel by neměl být nucen vzpomínat si na provádění operací v systému, návod k použití systému by měl být vždy viditelný nebo snadno dosažitelný, kdykoliv je potřeba.

Flexibilita a efektivita Pokročilí uživatelé by měli mít možnost zrychleného a efektivního používání systému.

Estetický a minimalistický vzhled Systém by neměl obsahovat žádné zbytečné informace. Čím méně možností uživatel má, tím rychleji koná.

Smysluplné chybové hlášky Nejlépe se vyvarovat chybovým hláškám, ale když už nastanou, měly by být uváděny v přirozeném jazyce a navrhovat konstruktivní řešení.

Nápověda a dokumentace Ideálním stavem je, když se dá systém používat bez nápovědy. I přesto je však nezbytné kdykoliv poskytnout uživateli pomoc a nápovědu, když ji bude potřebovat. Nápověda by měla obsahovat postupy v krocích a podporovat funkci vyhledávání.

1.4 Charakteristika person

Při navrhování uživatelského rozhraní by mělo být snahou přiblížit se co nejvíc lidským potřebám namísto toho, aby se uživatelé učili, jak se daný produkt

používá. Návrh by se měl orientovat přímo na uživatele (anglicky user-centered design), nikoliv na technologii.

Jednou z praktik, která je při návrhu nápomocná, je zaměřit se na potřeby konkrétních uživatelů, což může v budoucnu usnadnit uživatelské testování. Takové praktice se říká tvorba person. Personou se rozumí detailní popis fiktivní osoby, která by měla být typickým uživatelem daného produktu či služby. Persona odpovídá archetypu, nicméně její popis by měl být velmi věrný skutečnému člověku.

„Persona musí být založena na uživatelském výzkumu“ [7], aby byla přesná a reprezentativní. Rozhodně by však neměla odpovídat uživatelské skupině nebo segmentu na trhu. Taková identifikace je příliš neosobní a těžko se pro ni navrhuje konkrétní řešení. Ideálním přístupem je vytvořit jedinečného člověka se silným životním příběhem, který se vryje do paměti, a zvýší tak pravděpodobnost, že se na něj bude při navrhování produktu či systému myslet.

Jaké náležitosti by persona měla mít, záleží na oblasti, pro kterou se persona vytváří. Platí, že čím víc informací o osobě uvedeme, tím lépe si ji představíme jako reálného uživatele. Nicméně je dobré vyvarovat se až příliš vtípným popiskům, které mohou odvádět pozornost.

Nelze brát následující seznam za dogma, nicméně u běžné osoby se většinou uvádí následující vlastnosti:

- jméno,
- pohlaví,
- věk,
- vzdělání,
- zkušenosti,
- práce,
- příjem,
- zájmy,
- kontext, jak persona může interagovat s produktem,
- citáty/postoje osoby.

V případě této diplomové práce jsou ideálními osobami osoby, které pracují v optických optikách a přišly už do styku s nějakým informačním systémem pro správu zákazníků a zakázek.

1.4.1 Igor Štěpánek

Igor je sice stále mlád (29 let), ale ideály o životě ho už opustily. V mládí chtěl být kunsthistorikem, ale štěstí mu moc nepřálo. Po vystudovaném gymnáziu se zaměřením na historii se snažil dva roky studovat na vysoké škole, ale nezvládal to.

Pracovních pozic vystřídal hned několik. Při studiu gymnázia si nepravdělně přivydělával v místním fastfoodu. Při studiu vysoké školy si vyzkoušel práci prodavače v supermarketu, kde vydržel jen půl roku. Potom nějakou dobu pracoval coby hlídač parkoviště. Po zanechání studia se mu naskytla příležitost pracovat v nedaleké oční optice. Kývl na ni, doučil se vše potřebné a pracuje tam doteď.

Igor rád chodí do místní hospody, kde s přáteli sleduje různé sportovní zápasy (velmi ho baví závody Nascar) a diskutuje o politice. Politika je jeho vášní stejně jako hledání konspiračních teorií, které pak rozebírá nad pivem s přáteli, kteří ho vtipně nazývají „ten náš intelektuál“. Ne, že by konspiračním teoriím věřil, spíš naopak je rád vyvrací a dělá si z nich legraci, nicméně na internetu bývá poněkud paranoidní. Účet na Facebooku má anonymní, občas při surfování používá Tor a komunikuje výhradně přes zašifrované služby.

1.4.2 Jiřina Kučerová

Paní Jiřině je padesát čtyři let, má o rok mladšího manžela a dvě děti – dceru Marii a syna Karla. Po revoluci začala pracovat v oční optice, nejprve na poloviční úvazek, a potom, co se jí odstěhovaly obě děti z bytu, začala pracovat na plný úvazek. Za tu dobu se vypracovala na hlavní vedoucí pobočky a stará se o všechny záležitosti, ke kterým není potřeba podpis šéfa. Jen s prací na počítači měla dlouhou dobu problémy. Inu, informační technologie Jiřině k srdci prostě nikdy nepřirostly a trvalo jí pár let, než se naplno sžila s programem, který slouží k evidenci zakázek. Nemá ráda změny a zastává názor „proč měnit něco, co funguje“.

Mezi její záliby patří turistika, sledování televizních pořadů (obzvláště ráda má britské detektivky), křížovky, historie a v neposlední řadě pletení. Těší ji dělat radost svým přátelům a jednou za čas někomu něco uplete. Každoročně jezdí se svým manželem na týdenní dovolenou do rekreačního střediska Stará Živohošť.

1.4.3 Lumír Hrubý

Lumír (38 let) byl vždycky ambiciózní. Mohl za to jeho otec Libor, úspěšný majitel velké sítě poboček očních optik, který se svého syna snažil vést směrem k podnikání. Bohužel to byl i vášnivý kuřák a ve svých padesáti letech zemřel na rakovinu. Lumír po něm zdědil obří síť očních optik. Otec ho našťastí

naučil vše potřebné, takže netrvalo moc dlouho a Lumír se role majitele zhostil bez problémů.

Lumír je velmi často na cestách, komunikuje s dodavateli a řeší byznys. Má rád, když *věci fungují tak, jak mají*. Občas musí kontrolovat stav svých poboček a vše by nejraději řešil online. Preferoval by nějaký jednoduchý a stručný systém, protože nemá čas učit se něco nového. Otec mu vřdycky říkával, že „čas jsou peníze“, což si Lumír vzal k srdci.

Se svým dobrým kamarádem Milanem chodívá hrát každý pátek tenis. O víkendech jezdí se svou výrazně mladší přítelkyní Dominikou odpočívat do chaty v Krkonoších. Dovolenu rád tráví v zahraničí, zpravidla v nějaké exotické zemi. Každoročně se těší na zimu, protože velmi rád lyžuje.

1.5 Obecné nařízení o ochraně osobních údajů

Obecné nařízení o ochraně osobních údajů (anglicky General Data Protection Regulation neboli GDPR) je nová revoluční legislativa Evropské unie, která má výrazně zvýšit ochranu osobních dat občanů [8]. Týká se všech, kteří shromažďují nebo zpracovávají osobní údaje Evropanů, včetně společností a institucí mimo území Evropské unie, které působí na evropském trhu, a klade si za cíl chránit digitální práva občanů Evropské unie. Ačkoliv je řada mechanismů, se kterými GDPR přichází, již nyní v dosavadní právní úpravě, přináší s nimi i nové povinnosti pro zpracovatele údajů, kteří byli doposud kryti subjektem správce údajů. Nařízení nabývá platnosti 25. května 2018 a platí jednotně pro celou Evropskou unii.

Občané díky GDPR získávají nová práva. Právo na přístup dává občanům možnost ověřit si zákonnost zpracování jejich údajů. Právo na opravu jim umožňuje změnit již používané údaje. Právo na výmaz rozšiřuje právo být zapomenut a spočívá v provedení přiměřených kroků, které slouží k vymazání veškerých odkazů na osobní údaje žadatele a jejich kopie, nicméně v tomto případě uvádí GDPR řadu výjimek. Pokud by se jednalo o výjimku, může občan využít alespoň právo vznést námitku a tím donutit společnost k omezenému zpracování těchto údajů.

Zmínka o GDPR je v této diplomové práci z prostého důvodu – informační systém OptiLynx umožňuje vytvářet zakázky konkrétním zákazníkům. V informačním systému není možné vytvořit zakázku bez přiřazení ke konkrétnímu člověku, vždy je potřeba vědět alespoň jeho jméno. Proces vytváření nového zákazníka umožňuje vyplnit jeho adresu, kontaktní údaje (telefon a e-mail) a další informace. Kvůli GDPR je ve formuláři pro vytváření nového zákazníka povinné vyplnit pouze jeho jméno. Na některé položky ze zakázky lze uplatnit kromě slevy přímo od prodejce také slevu přes pojišťovnu. V takovém případě je zákazník povinen dané optice sdělit své rodné číslo a pojišťovnu, se kterou má uzavřenou smlouvu o pojištění. Tyto informace jsou uvedené na poukazu od lékaře, který zákazník musí předložit při vytváření nové zakázky [9]. V ta-

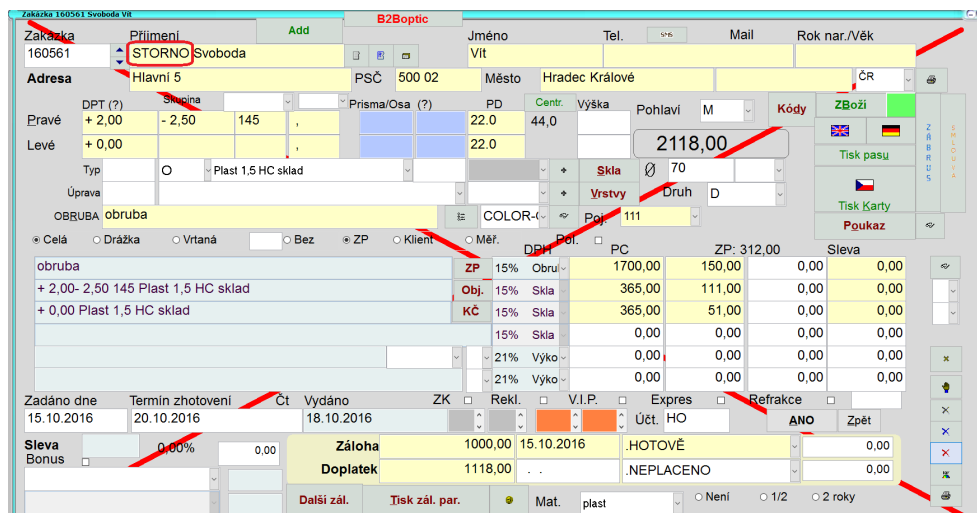
kovém případě systém musí při vkládání slevového poukazu do zakázky hlídat, zda jsou tyto informace uloženy v databázi, jinak není možné poukaz a slevu z něj na danou zakázku použít.

1.6 Analýza konkurenčních systémů pro optiky

Analýza konkurence je užitečnou metodou pro získání lepšího přehledu o zkoumané oblasti [10]. Pokud se analýza provede pořádně, může být přínosná nejen při návrhu vlastního systému, ale také při následné implementaci. Výstupem by měl být seznam chyb a závažných problémů, kterým by bylo dobré se vyvarovat. Dobrým přístupem je také volba konkurence, které se v dané oblasti daří, protože může sloužit jako zdroj inspirace.

Na českém trhu existuje pár společností, které vyvíjejí informační systémy zaměřené na oční optiky. Řada z nich je ovšem závislá na vybrané platformě, nevypadá moderně a nesplňuje základní pravidla správného uživatelského rozhraní. K analýze konkurence jsem si vybral tři informační systémy, které v současnosti mnoho středně velkých optik využívá – **Newton**, **OpticEvidence** a **Tiss Optic**.

1.6.1 Newton – specializovaný software pro oční optiku



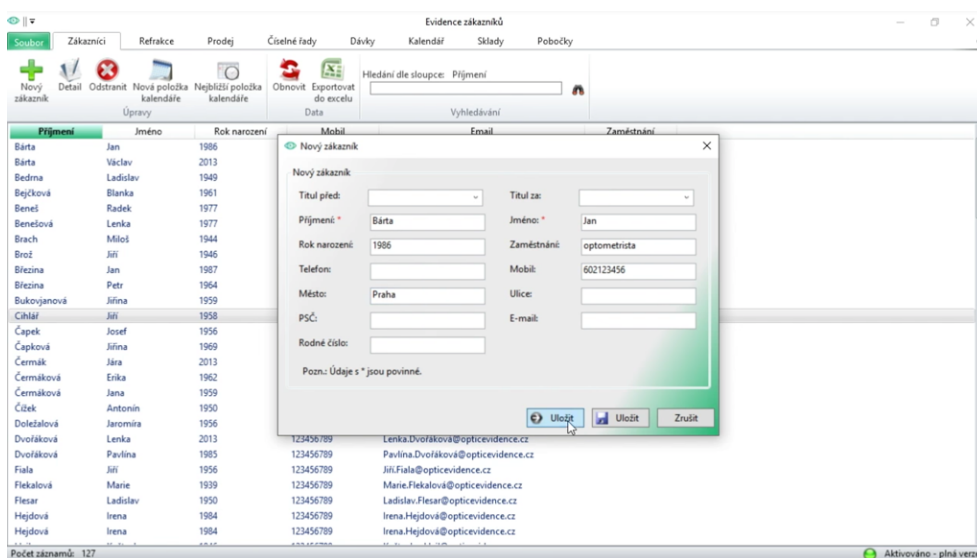
Obrázek 1.1: Newton – ukázka stornované zakázky [11]

Newton (obrázek 1.1) je aplikace vyvíjená společností MAXOFT, s.r.o., kterou do konce minulého roku využívala společnost OČNÍ OPTIKA BENEŠOV s.r.o. Kvůli legislativním změnám však Newton přestal vyhovovat, což byl také důvod, proč se začal vyvíjet informační systém OptiLynx. Při pohledu

1. SEZNÁMENÍ S POŽADAVKY OČNÍ OPTIKY A PROVEDENÍ ANALÝZY

na uživatelské rozhraní je jasné, že jde o aplikaci s velmi bohatou minulostí. Starý a nepřehledný vzhled působí zmatečně a nedá se v něm vyznat. Pro nového uživatele je téměř nemožné, aby v systému provedl nějakou akci bez použití nápovědy. Newton je dostupný pouze na Windows.

1.6.2 OpticEvidence – Systém pro komplexní správu optiky



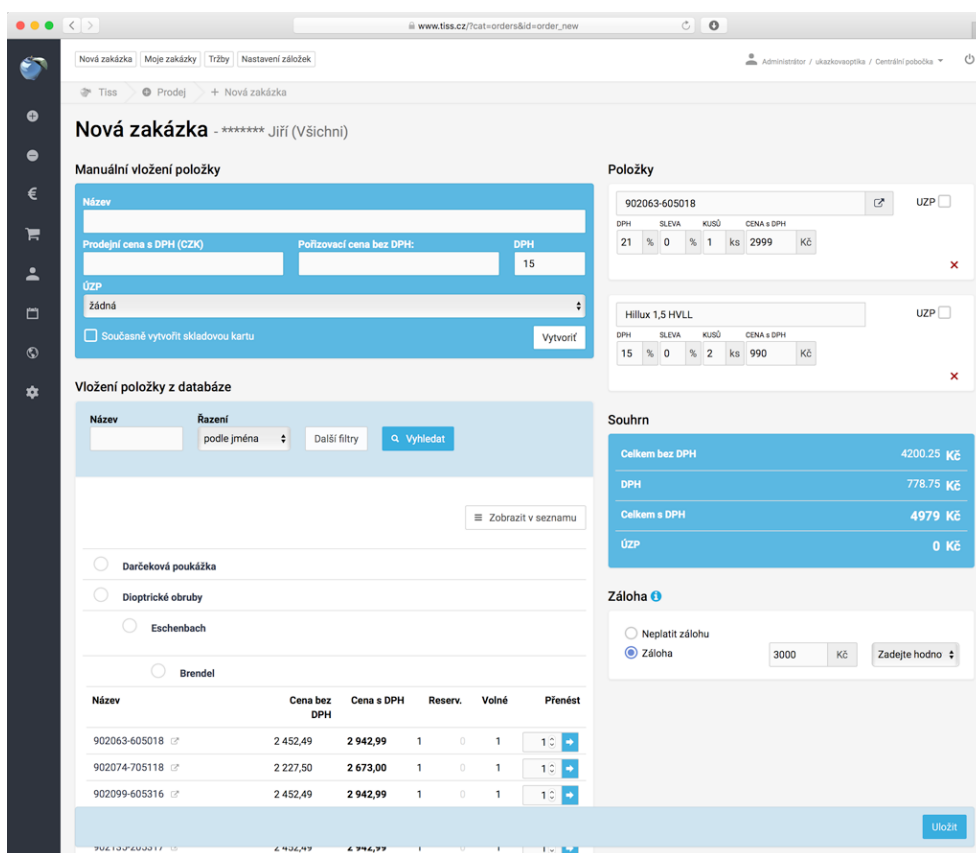
Obrázek 1.2: OpticEvidence – ukázka vytvoření zákazníka [12]

Informační systém OpticEvidence (obrázek 1.2) má velmi podobné uživatelské rozhraní s kancelářskými nástroji Microsoft Office. Funguje jen na operačním systému Windows a při instalaci vyžaduje Microsoft SQL Server. Vyznat se v něm dá celkem snadno, jen vzhled působí příliš technokraticky – všude jsou samá tlačítka, formuláře vypadají nelibě a při různých procesech se otevírá mnoho oken. Záloha dat je možná pouze na flash disk.

1.6.3 Tiss Optic – online informační systém pro optiky

Tiss Optic (obrázek 1.3) vypadá z trojice analyzovaných systémů nejmoderněji. Jde o klasickou webovou aplikaci, která funguje ve webovém prohlížeči, tudíž je nezávislá na platformě. Práce v systému je jednoduchá a intuitivní. Pro nového uživatele není problém se zorientovat v systému. Jelikož systém funguje online, jsou data dostupná odkudkoliv, kde je přístup k internetu.

1.7. Analýza drátěného modelu systému OptiLynx



Obrázek 1.3: Tiss Optic – ukázka vytvoření zakázky [13]

1.7 Analýza drátěného modelu systému OptiLynx

Před tvorbou samotného návrhu jsem dostal od vedoucího práce Ing. Jiřího Hunky a od studentů Bc. Filipa Glazara a Bc. Oldřicha Malce již připravený drátěný model neboli wireframe ve formě klikatelného prototypu. V kontextu webových stránek definuje webdesignér Jan Řezáč wireframy následovně: „Wireframy jsou sadou stránek webu z analytického pohledu. Znárodnují obsah webu, jeho přibližnou polohu na stránce a v případě detailnějších wireframů i jeho vizuální prioritu. Drátěné modely stránek tvoří předěl mezi grafickou a funkční stránkou webu.“ [14]

Klikatelný prototyp drátěného modelu pro informační systém OptiLynx byl vytvořen pomocí prototypovacího softwaru Axure a vznikl ve spolupráci se zaměstnanci společnosti OČNÍ OPTIKA BENEŠOV s.r.o. 1.2, kteří poskytli informace o tom, co v systému musí být. Snahou bylo jednak vytvořit snadnou a přehlednou informační architekturu neboli hierarchii a provázanost jednotlivých částí, jednak na jednotlivých obrazovkách systému rozmístit všechny důležité prvky a komponenty.

V drátěném modelu byly přichystány tyto obrazovky:

- pokladna,
- otevření/zavření pokladny,
- přehled zákazníků,
- detail a úprava zákazníka,
- vytvoření nového zákazníka,
- přehled zakázek,
- detail zakázky (obrázek 1.5),
- vytvoření nové zakázky,
- přehled skladových položek (obrázek 1.4),
- detail skladové položky,
- přehled zakázek,
- naskladnění a přeskladnění zboží,
- párování faktur s naskladněným zbožím,
- historie pohybu skladových položek (naskladnění, přeskladnění).

Jednotlivé obrazovky spojuje horní navigace společně s patičkou. Většina obrazovek obsahuje vstupní pole určené pro hledání a filtrování výsledků. Pokud se hledá ve více skupinách (např. na pokladně se hledá ve všech produktech, službách, zakázkách a zákaznících), zobrazí pod vstupním polem okno s výsledky (obrázek 1.4).

Zakázka představuje nejkompexnější proces celého systému (obrázek 1.5), proto jsem se rozhodl podrobně rozebrat její drátěný model. V horní části zakázky jsou uvedeny informace o zákazníkovi. Měly by jít kdykoliv během tvorby zakázky změnit a změny by měly být aplikovány globálně. Poté následuje vkládání čoček se všemi parametry (dioptrie, cylindr, prisma atd.), které se u nich uvádějí. Do zakázky by mělo jít přidat víc čoček. Každá čočka může mít jinou cenu a jinou slevu. Zakázka může obsahovat produkty a služby. Poukaz od lékaře (jehož veškeré náležitosti byly popsány v 1.5) se otevírá v modálním okně (viz 3.3.13), avšak nemyslí na situaci, kdy bude chtít zákazník uplatnit dva a více poukazů. Závěrem lze zakázku uložit, uložit a vydat nebo stornovat.

1.7. Analýza drátěného modelu systému OptiLynx

The screenshot shows the 'Sklad' (Warehouse) section of the OptiLynx system. At the top, there are navigation tabs: Pokladna, Zákazníci, Zakázky, Sklad, Exporty, and Volby. Below these are input fields for 'Číslo dodacího listu', 'Dodavatel', 'Datum vystavení', and 'Datum přijetí'. A search bar contains the text 'Pouzdro' and a 'Hledat' button. The search results are displayed in a table with columns for 'Položka', 'Kusů', and 'Detail'. The first item is 'Pouzdro Armor' with 1 unit. Below the table is a 'Vytvořit novou položku' button. A section titled 'Přesun mezi sklady' shows a table with columns for 'ID', 'Datum', 'Zdroj', 'Cíl', and 'Detail'. The first entry is '13547861273', '25. 11. 2017', 'Benešov', 'Karel'. There are several yellow callout boxes with text: 'Pokud byla stránka otevřena přímo z dokončení naskladnění, budou automaticky zvoleny věci z daného naskladňování', 'Pokud je zvoleno naskladnění předchozího vyskladnění, nelze již přidávat další položky, dodavatel se nastaví na zdrojový sklad a nelze vyplnit další pole v horní části stránky', 'Dokončit naskladnění', and 'Dokončit naskladnění a přidat fakturu'. At the bottom, there is a footer with '© 2017 Jagu s. r. o.' and a 'Nápověda' link.

Obrázek 1.4: Drátěný model skladu – naskladnění

The screenshot shows the 'Zakázky' (Orders) section of the OptiLynx system. At the top, there are navigation tabs: Pokladna, Zákazníci, Zakázky, Sklad, and Volby. The main heading is 'Číslo zakázky 45123 Karel Ostrovid'. Below this are fields for '780217/0123', '+420 124 456 789', 'Týnec na Sázavou', and 'VIP: ano'. There are two buttons: 'Upravit údaje zákazníka' and 'Změnit zvoleného zákazníka'. Below this is a 'Skla' section with a grid of input fields for 'Levé' and 'Pravé' lenses, including 'Sféra', 'Cylindr', 'Osa', 'Addice', 'Prisma', 'Osa prisma', 'PD', and 'Výška'. There are also radio buttons for 'Zisk dat' (ZP, Klient, Naměřeno). Below the grid are fields for 'Výrobce', 'L: Kód levé', 'P: Kód pravé', 'Výrobce', and 'Typ', with a 'Přidat do zakázky' button. There is also an 'Úprava' section with 'Kód' and 'Typ' fields. The 'Obsah zakázky' section shows a table with columns for 'Hledat ve skladu a službách', 'Sko L: +1.5, 10, Hilux, HC, QSP', 'Sko P: +1.0, 0, Hilux, HC, QSP', and 'Obruba HoldEm, 452 XSP'. The table shows prices in Kč and buttons for 'Upravit' and 'Odstranit'. Below the table is a summary: 'Souhrn obsahu zakázky: Sleva celkem (%)', 'Cena celkem: 903 Kč', 'Zaplaceno: 0 Kč', and 'Zbývá doplatit: 903 Kč'. At the bottom, there is a 'Tisk' section with buttons for 'Zábrus', 'Pas', 'Poukaz', 'Uložit', 'Karta (CZ)', 'EN', 'DE', 'Reklamační protokol', 'Stornovat zakázku', and 'Uložit a vydat'. There are also checkboxes for 'Reklamační protokol', 'VIP', 'Měřena refrakce', 'Obruby', 'Celá', 'Drážka', and 'Vrtaná'.

Obrázek 1.5: Drátěný model zakázky

Návrh frontendu informačního systému pro optiky

Vzhledem ke komplexitě informačního systému OptiLynx bylo potřeba před samotnou implementací vypracovat podrobný návrh. Ten se na úrovni front-endu skládá ze dvou částí – technické řešení a uživatelské rozhraní. První část kapitoly se zabývá přístupem k vývoji. Zvažuje dostupné možnosti a vybere tu, která nejlépe splní požadavky získané z předchozí analýzy. S tím souvisí také volba technologií a metodologií, které se použijí při implementaci informačního systému.

Druhá část kapitoly se na informační systém podívá z uživatelského hlediska. Zvaží, jaká hotová řešení by se dala použít, a pokusí se navrhnout uživatelské rozhraní tak, aby bylo moderní a použitelné pro všechny uživatele, kteří s ním přijdou do styku.

2.1 Volba typu aplikace

Od této fáze vývoje jsem úzce spolupracoval s Bc. Filipem Glazarem, který měl na starost backend informačního systému. S ním jsem diskutoval možné postupy, kterými se při vývoji vydat.

Brali jsme v potaz dva hlavní požadavky na informační systém. Prvním požadavkem byla nezávislost na platformě. Nechtěli jsme se upínat na konkrétní zařízení ani na konkrétní operační systém, jako to dělají některé konkurenční systémy (viz 1.6), naším záměrem bylo vytvořit informační systém pro každého, kdo disponuje moderním elektronickým zařízením, které se dá využít k práci, tedy především stolní počítač, notebook, tablet a mobilní telefon. Druhým požadavkem byla dostupnost informačního systému a jeho aktuálních dat odkudkoliv, tedy možnost přihlásit se do systému z jakéhokoliv zařízení a vidět v něm stejný obsah.

Ideálním kandidátem, který splní požadavek na nezávislost i dostupnost, je

webová aplikace. Webovou aplikaci lze spouštět ve webovém prohlížeči, který obsahuje již v základním nastavení většina běžně dostupných elektronických zařízení. I přesto, že možnosti webových prohlížečů nejsou neomezené a je potřeba brát v potaz některé jejich restriktce, lze pro ně vyvinout plnohodnotné a použitelné aplikace, které nemusejí být závislé na žádných externích knihovnách.

Pro uspokojení požadavku na dostupnost informačního systému je zapotřebí přístup k internetu. V dnešní době, kdy je téměř všude po České republice dostupný vysokorychlostní internet, to není problém. Je však potřeba počítat se situací, kdy internet nebude plně funkční (např. při výpadku sítě) a zařídit zpětnou synchronizaci.

Při analýze všech požadavků jsme se dále shodli na tom, že bychom chtěli vytvořit multiplatformní desktopovou variantu dostupnou pro Windows, Linux a Mac, aby se informační systém OptiLynx netvářil jako klasická webová stránka a uživatelé při jeho používání měli pocit, že využívají plnohodnotnou aplikaci. Pro tento případ jsme zvolili framework Electron, který slouží k vytváření nativních aplikací pomocí webových technologií. Electron zjednodušeně funguje jako tzv. obálka, do které se vloží webová aplikace bez klasických prvků webového prohlížeče a tváří se jako nativní aplikace v daném operačním systému.

2.2 Volba přístupu

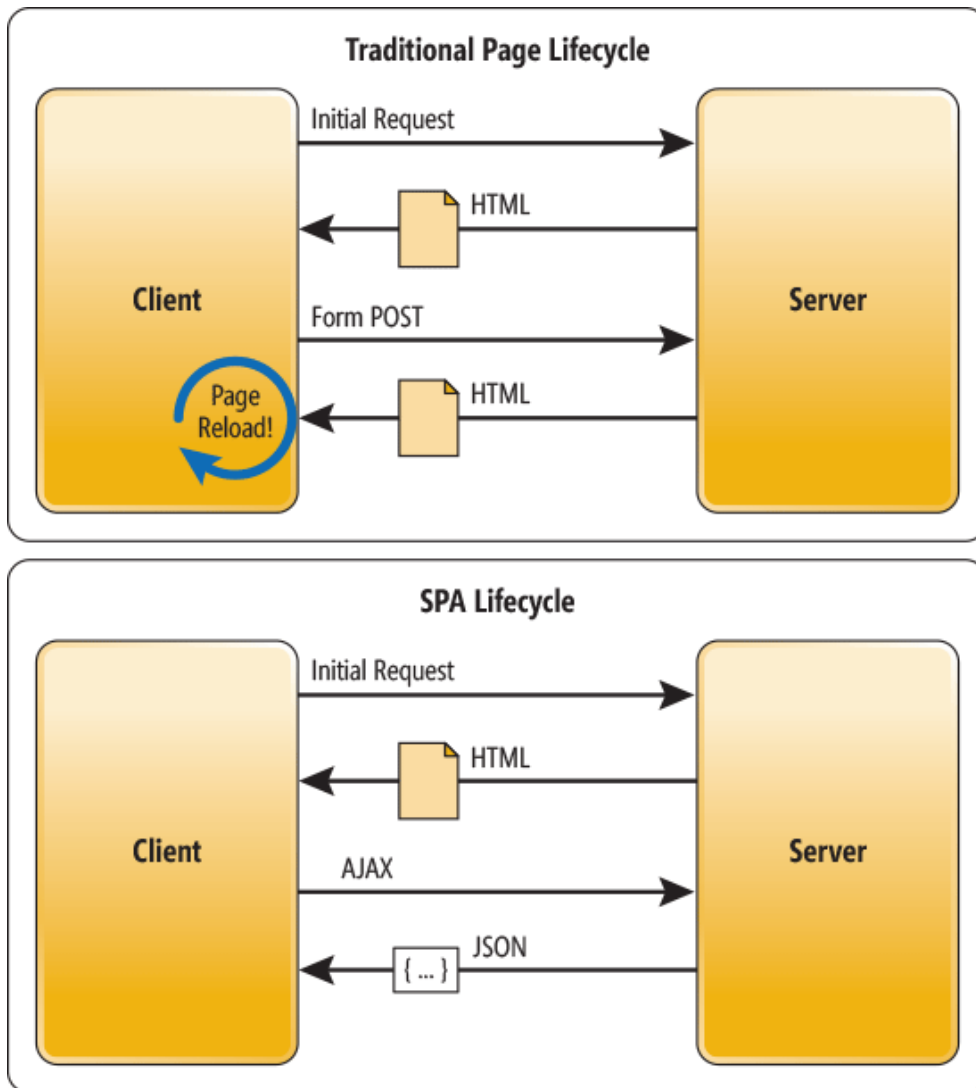
Volba přístupu, jakým bude webová aplikace fungovat, pro nás byla mnohem složitější. Ve světě webového vývoje existují dva přístupy, jak se dá aplikace pojmout: Single Page Application (SPA) a Multiple Page Application (MPA). Podstatným rozdílem v těchto přístupech je komunikace mezi klientem (webový prohlížeč, Electron aplikace) a serverem. Obě varianty mají své výhody a nevýhody a pro každý projekt se může hodit něco jiného. [16]

2.2.1 Multiple Page Application

Multiple Page Application představuje tradiční přístup, jakým se webové aplikace, potažmo webové stránky, vyvíjejí od svého vzniku. Komunikace klienta se serverem je na sobě při každém požadavku přímo závislá. Tento druh komunikace je příliš zdlouhavý a narušuje plynulost používání aplikace.

Komunikace mezi klientem a serverem probíhá následovně (viz obrázek 2.1):

1. Klient pošle inicializační požadavek na server.
2. Server požadavek zpracuje a vytvoří odpověď.
3. Server pošle odpověď klientovi.



Obrázek 2.1: Srovnání životního cyklu tradičních stránek (MPA) a SPA [15]

4. Klient odpověď přijme.
5. Klient překreslí obsah stránky na základě odpovědi a zobrazí ji uživateli.
6. Klient při každém novém požadavku vytvoří dotaz, pošle ho serveru a opět se pokračuje bodem 2.

2.2.2 Single Page Application

Single Page Application (někdy také one page application) je poměrně nový a progresivní přístup, ke kterému se uchyluje stále víc vývojářů. V současnosti

ho využívá řada velkých společností jako Google či Facebook. Komunikace klienta se serverem je největší při prvním spuštění – klient si stáhne celou aplikaci a pak v ní komunikuje se serverem pouze, když chce pracovat s daty. Server v tomto případě slouží pouze jako zdroj a úložiště dat. Z uživatelského hlediska jde o lepší variantu, protože je práce v aplikaci plynulá. [17]

Komunikace mezi klientem a serverem probíhá následovně (také viz 2.1):

1. Klient pošle inicializační požadavek na server.
2. Server požadavek zpracuje a vytvoří odpověď, která obsahuje celou aplikaci.
3. Server pošle odpověď klientovi.
4. Klient odpověď přijme a spustí aplikaci.
5. Klient při každém novém požadavku vytvoří dotaz a pošle ho asynchronně na server.
6. Server požadavek zpracuje a odpoví ve formě dat (nejčastěji ve formátu JSON).
7. Klient přijme data, zpracuje je a překreslí potřebné elementy na stránce.

2.2.3 MPA vs. SPA – srovnání a zvolený přístup

Obecné srovnání přístupů Multiple Page Application a Single Page Application je zpracováno v tabulce 2.1.

Informační systém OptiLynx by měl mít rychlou odezvu a z uživatelského hlediska reagovat na veškeré akce přirozeně (tedy nepřekreslovat po každé akci celou stránku). Optimalizace pro vyhledávače není v rámci interního systému vůbec relevantní. Problém se závislostí na JavaScriptu a s podporou moderních webových prohlížečů nehrozí, poněvadž bude informační systém nakonfigurován v Electronu jako nativní aplikace, která bude tyto požadavky splňovat.

Po společném zvážení všech výhod a nevýhod jednotlivých přístupů jsme se nakonec s Bc. Filipem Glazarem rozhodli zvolit cestu **Single Page Application**.

2.3 Zvolené technologie

V následující části kapitoly budou popsány konkrétní technologie, které byly vybrány pro realizaci informačního systému OptiLynx.

Tabulka 2.1: Srovnání MPA a SPA [18]

Multiple Page Application	Single Page Application
Pomalé. Při každém dotazu na server se musí načíst celá stránka znovu.	Rychlé, reaktivní chování. Pomalejší načítání jen při prvotním spuštění. Při každém dotazu na server se načtou a překreslí pouze nová data.
Frontend a backend jsou na sobě přímo závislé, jedno nefunguje bez druhého.	Frontend a backend jsou jasně odděleny, každý plní svou roli.
Optimalizované pro vyhledávače, internetový robot nemá problém přečíst a procházet obsah jednotlivých stránek.	Optimalizace pro vyhledávače je náročnější. Existují nástroje, které přichystají jednotlivé stránky v přijatelné podobě pro internetového robota, ale nejde o triviální záležitost.
Bez závislosti na JavaScriptu.	Vyžaduje JavaScript. Pokud je JavaScript v prohlížeči vypnutý, aplikace nebude fungovat správně.
Podporuje starší webové prohlížeče.	Funguje především v moderních webových prohlížečích.
Při vývoji na mobilní zařízení je třeba přizpůsobit či optimalizovat backend aplikace.	Podporuje vývoj pro mobilní zařízení, lze použít backend aplikace bez jakékoliv změny.
Prohlížeč informuje o stavu načítání při každém dotazu.	Informace o stavu načítání je potřeba implementovat do aplikace.

2.3.1 HyperText Markup Language

Jednou z klíčových technologií, které se používají při tvorbě webových stránek a webových aplikací, je značkovací jazyk nazvaný HyperText Markup Language (HTML). HTML jednak popisuje strukturu dokumentu ze sémantického hlediska a jednak mezi více dokumenty vytváří vazby a skrz hypertext je propojuje. K sémantickému popisu se používají elementy, které jsou definovány speciálními tagy. Jejich definice vycházejí ze standardů, které od roku 1994 vydává mezinárodní konsorcium World Wide Web Consortium (W3C). Tyto standardy dodržují všechny moderní webové prohlížeče.

Od vzniku W3C vyšlo mnoho verzí HTML. Pro tuto diplomovou práci jsem si zvolil poslední verzi **HTML5**.

2.3.2 Cascading Style Sheets

Druhou klíčovou technologií při tvorbě webových stránek a webových aplikací je Cascading Style Sheets (CSS) neboli česky kaskádové styly. CSS definuje

vzhled dokumentu za pomoci pravidel, která se aplikují na HTML elementy. Kaskádový přístup umožňuje vrstvit definice stylu, nicméně platí pravidlo, že se aplikuje pouze poslední definice stylu. CSS umožňuje stylovat a formátovat texty, upravovat formulářové prvky, vytvářet různé grafické objekty, následně je pozicovat a v neposlední řadě také definovat layout neboli rozvržení celého dokumentu. Vlastnosti se zapisují do tříd, kterými se identifikují HTML tagy.

Standardy pro CSS opět vydává mezinárodní konsorcium W3C. Neustále pracuje na nových specifikacích, které se průběžně implementují do vybraných webových prohlížečů. Poslední verzí CSS je CSS3, které přineslo mnoho revolučních změn. Největším přínosem jsou tzv. media queries, které podporují styly pro různé druhy zařízení. Lze pro jeden element napsat víc CSS pravidel, která se na něj aplikují v závislosti na zařízení, ve kterém se dokument otevře. [19]

2.3.3 CSS preprocessory

CSS preprocessor je nástroj, který převede zdrojový kód ve vlastní syntaxi do CSS syntaxe podporované webovými prohlížeči. Přináší lepší syntaxi, umožňuje zkrácený zápis a snazší zápis dědičnosti vlastností, dokáže rozdělit CSS kód na víc částí, rozšiřuje kód o proměnné, početní operace, funkce a mixiny (předefinované bloky kódu, které se často opakují).

Nejčastěji se používají tyto CSS preprocessory:

- Leaner Style Sheets (LESS),
- Stylus,
- Syntactically Awesome Style Sheets (SASS).

Jednotlivé preprocessory se mezi sebou výrazně neliší. LESS respektuje deklarativní povahu CSS, dá se snadno naučit, ale nenabízí elegantní zápis pro pokročilejší konstrukce jako podmínky a cykly. Naproti tomu SASS a Stylus mají imperativní charakter a se svými syntaxemi se přibližují klasickým programovacím jazykům [19].

Pro práci jsem si vybral **SASS**, protože ho používá frontendová knihovna Bootstrap ve své poslední verzi.

2.3.4 Flexbox Layout

CSS 2.1 definovalo čtyři režimy rozložení, jejichž algoritmy určují velikost a pozici elementů v dokumentu v závislosti na okolních elementech:

- **Blokový layout** – určený k rozvržení dokumentu.
- **Inline layout** – určený k rozložení textu.

- **Tabulkový layout** – určený pro zobrazení 2D dat v tabulkovém formátu.
- **Pozicovaný layout** – určený pro velmi explicitní pozicování bez ohledu na ostatní elementy v dokumentu.

Žádná z výše vypsanych metod však nevznikla pro tvorbu layoutů. Až teprve CSS3 přineslo způsob flexibilního stylování za pomoci vybraných vlastností, kterým se souhrnně říká **flexbox layout**. Dlouhou dobu se tento přístup nevyužíval, což bylo především z důvodu slabé podpory webových prohlížečů. Nicméně nyní se situace změnila a majoritní webové prohlížeče vlastnosti flexboxu podporují [20].

Flexbox představuje moderní způsob tvorby layoutu. Jeho princip spočívá v tom, že elementům v dokumentu přiřadí vlastnosti, díky kterým se stanou pružné a přizpůsobivé okolí. Pomocí vlastností se dá nastavovat zarovnání, rozmístění a pořadí elementů. Mezi hlavní výhodu flexboxu patří schopnost vyplňovat zbylý prostor bez nutnosti přepočítávání Javascriptem [19].

Alternativou pro flexbox je CSS Grid, což je relativně nový webový layout. Jeho síla spočívá v tom, že na rozdíl od flexboxu, který pracuje s layoutem pouze v jedné dimenzi, používá typografickou mřížku, která umožňuje pozicovat elementy rovnou ve dvou dimenzích – v řádcích i sloupcích. Tento přístup je mnohem lepší pro celostránkové a komplexnější layouty. Podpora u webových prohlížečů je však stále žalostná, tudíž jsem si pro tuto diplomovou práci zvolil flexbox layout.

2.3.5 Organizace kódu

U velkých webových projektů se velmi často stává, že obsahují mnoho elementů, které mají spoustu CSS pravidel. Poměrně rychle se stane takový kód neudržitelným, nepřehledným a naprosto neefektivním, pokud s ním pracuje víc vývojářů. Proto se vyplatí dodržovat určité konvence, jak takový kód psát, případně zvolit již existující metodiku pro organizaci CSS.

Nejznámější existující přístupy organizace CSS jsou:

- Atomic CSS,
- BEM – Block, Element, Modifier,
- Object Oriented CSS (OOCSS),
- Scalable and Modular Architecture for CSS (SMACSS).

Pro práci jsem si vybral **přístup BEM**, poněvadž s ním mám nejvíc zkušeností a také protože jsem se s ním setkával nejčastěji. Na BEM se dá nahlížet dvěma způsoby – buď jako na plnohodnotnou metodiku pro organizaci CSS, nebo jako

na konvenci pro pojmenovávání tříd. Oba způsoby mají na internetu dostupnou dokumentaci. Já jsem pro potřeby informačního systému OptiLynx zvolil druhý způsob, protože plnohodnotná metodika je příliš komplexní a svazující.

BEM k zápisu tříd přistupuje na základě typu použití [21]:

Blok Odpovídá komponentě uživatelského rozhraní. Jde o nezávislý prvek uživatelského rozhraní, který je znovupoužitelný.

Element Prvek uvnitř bloku, který se styluje. Nelze ho v rozhraní použít samostatně, jeho existence má smysl pouze v rámci bloku.

Modifikátor Upravuje komponentu na základě vizuální vlastnosti, stavu nebo chování. Jeho název by neměl být závislý na vzhledu.

BEM v CSS lze zapsat následovně:

- **Blok** – .block
- **Element** – .block__element
- **Modifikátor** – .block--modifier
- **Modifikátor elementu** – .block--modifier

2.3.6 JavaScript

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, který vznikl v roce 1995 a od té doby se používá na webu jako interpretovaný programovací jazyk [22]. Na webu slouží k ovládní různých interaktivních prvků (tlačítka, formuláře), tvoří se s ním animace či jiné efekty a v neposlední řadě dokáže manipulovat s HTML elementy. K tomu potřebuje **Document Object Model (DOM)**, což je objektově orientovaná reprezentace XML nebo HTML dokumentu. DOM vznikne při načtení HTML dokumentu a z jednotlivých elementů vytvoří strom objektů, ke kterému má JavaScript přístup.

JavaScript na webu umožňuje vytvářet a posílat asynchronní dotazy webovému serveru, který je zpracuje a v případě potřeby na ně pošle odpovědi. Přijaté odpovědi poté JavaScript zpracuje a modifikuje HTML DOM bez nutnosti znovunačítání celé webové stránky. Těmto technikám se souhrnně říká **Asynchronous JavaScript and XML (AJAX)** a využívají se u dříve zmiňovaného přístupu Single Page Application (viz 2.2.2).

Syntaxe a sémantika jazyka JavaScript vychází ze standardu ECMAScript 5, který podporují všechny moderní webové prohlížeče. JavaScript využívá řada moderních webových frameworků, mezi něž patří Angular, který jsem si vybral pro realizaci informačního systému OptiLynx.

2.3.7 TypeScript

TypeScript je programovací jazyk vytvořený a spravovaný společností Microsoft, který slouží jako nadstavba nad jazykem JavaScript. TypeScript umožňuje psát moderní javascriptový kód rozšířený o typovou kontrolu, třídy, rozhraní, moduly a mnoho dalších vlastností. TypeScript nerozšiřuje JavaScript o moderní konstrukty, které vycházejí z nových verzí skriptovacího jazyka ECMAScript, TypeScript, pouze rozšiřuje používaný standard tak, aby nedošlo k nekompatibilitě ve webových prohlížečích [23].

Jeho překlad funguje na podobném principu jako CSS preprocesory. Překladáč vezme všechny zdrojové soubory psané v TypeScriptu a převede je do jednoho velkého souboru psaného v čistém JavaScriptu.

2.3.8 Webový framework

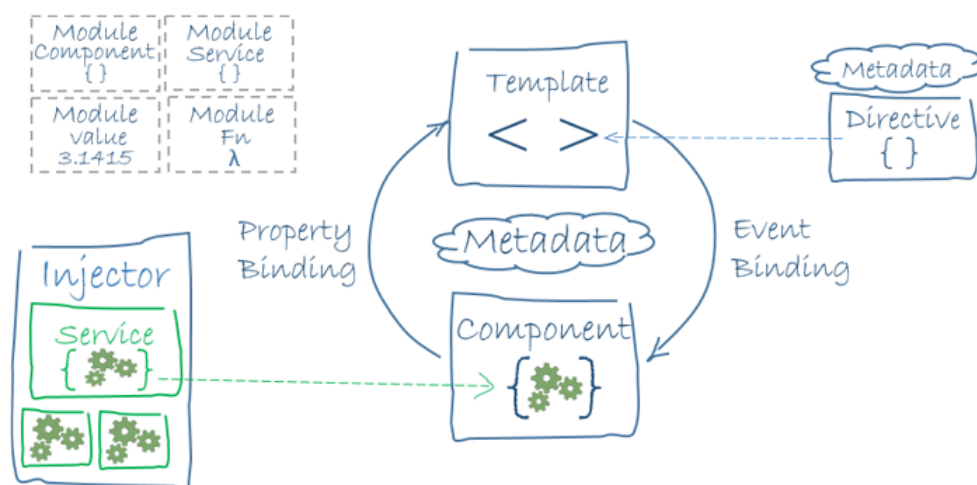
Webový framework představuje ucelený soubor postupů a tematicky zaměřených knihoven. Slouží k efektivnímu vývoji; odstiňuje vývojáře od typických problémů dané oblasti a umožňuje mu soustředit se na konkrétní zadání.

Ve světě moderních webových frameworků, které fungují na principu Single Page Application, je momentálně nejpoužívanější trojice **Angular** (vyvíjí Google), **React** (vyvíjí Facebook) a **Vue.js** (vyvíjí Evan You, bývalý zaměstnanec Googlu). Každý z těchto frameworků má své výhody a nevýhody a nedá se říct, který z nich je nejlepší. Volba webového frameworku pro SPA byla nakonec čistě pragmatická – podle průzkumu vývojářů po celém světě podle komunity Stack Overflow [24] vyšel nejlépe **Angular**, takže jsme se rozhodli pro něj.

2.3.9 Angular

Angular je platforma a framework určený k vytváření klientských aplikací využívajících webovou technologii HTML a programovací jazyk TypeScript. Angular je celý napsaný v jazyce TypeScript a ve svém základu obsahuje hlavní funkcionality pro vytvoření jednoduché webové aplikace. Lze ho snadno rozšířit o řadu dalších funkcionalit pomocí knihoven, které jsou také napsány v TypeScriptu.

Základem aplikace napsané v Angularu jsou **moduly** (modules). Moduly poskytují kontext pro komponenty a udávají závislosti na jiné moduly. **Komponenty** (components) jsou stavební prvky, ze kterých se staví výsledná aplikace. Vždy musí existovat kořenová komponenta, do které se vkládají další komponenty, čímž vznikne strom komponent. Každá komponenta je oddělená část kódu, která obsahuje vlastní logiku, data a šablonu definující zobrazení. **Šablona** kombinuje klasické HTML s direktivami Angularu, které slouží například k vykreslení dat. Komponenty dále používají **služby**, které definují speciální funkce nesouvisející s pohledy. Služby umožňují vkládat závislosti



Obrázek 2.2: Schéma propojenosti základních stavebních bloků aplikace v Angularu [25]

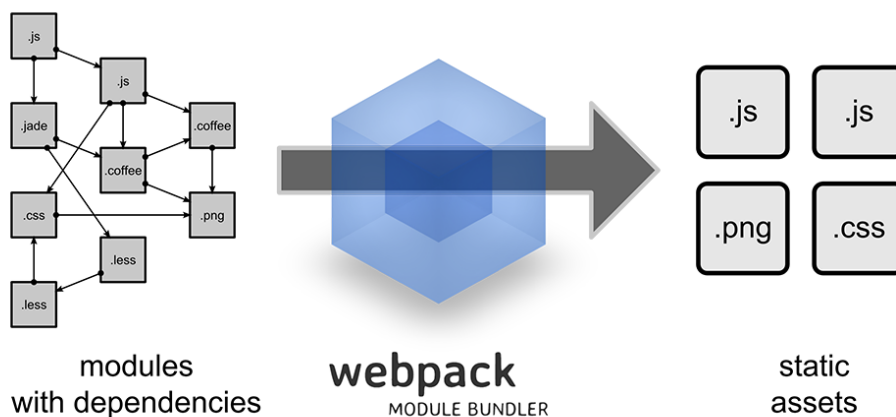
mezi jednotlivými komponentami aplikace (Dependency Injection). [25] Provázanost základních stavebních bloků aplikace napsané v Angularu znázorňuje schéma na obrázku 2.2.

Pro spuštění aplikace napsané v Angularu přímo ve webovém prohlížeči je zapotřebí ji přeložit do jazyka JavaScript. K tomu slouží buď jednoduché rozšíření příkazové řádky v operačním systému pomocí nástroje Angular CLI, nebo komplexnější řešení ve formě balíčkovacího systému Webpack. Pro vývoj jsem si vybral Webpack jednak z důvodu větší konfigurovatelnosti a jednak z důvodu flexibility při psaní odlišných konfigurací pro různá prostředí (vývojové prostředí, produkční prostředí).

2.3.10 Webpack

Webpack je javascriptový balíčkovací systém, který si při zpracování aplikace vytvoří graf závislostí všech vyžadovaných modulů a vygeneruje jeden či více statických balíčků. Obsahem balíčků pak jsou přeložené a plně kompatibilní kódy pro požadované prostředí. Názorně to ilustruje obrázek 2.3.

Aplikace v Angularu je díky členění na komponenty pěkně modulární, což znamená, že každá komponenta je nezávislá a snadno zaměnitelná, protože obsahuje vše nezbytné pro svou vlastní funkcionalitu. V praxi to znamená, že každá komponenta má zdrojový kód ve svém vlastním souboru. Stejně modulární mohou být i kaskádové styly, poněvadž je možné psát styl pro každou komponentu zvlášť (případně při použití nějakého CSS preprocesoru lze rozdělit styly do více souborů a pak je mezi sebou provázat). Webpack je schopný projít všechny soubory, zanalyzovat závislosti, zpracovat obrázky (převést je



Obrázek 2.3: Webpack – ilustrace funkcionality [26]

do úspornějšího formátu, přejmenovat a přesunout do hlavního adresáře), přeložit zdrojové kódy a vygenerovat finální spustitelnou aplikaci.

Webpack je mocný nástroj. Informační systém OptiLynx bylo potřeba nakonfigurovat pro produkční a vývojový režim, k čemuž je Webpack ideální volba, a proto jsem si ho vybral k Angularu.

2.4 Komunikační rozhraní

Následující část kapitoly se zabývá komunikačním rozhraním mezi frontendem a backendem informačního systému OptiLynx. Popisuje vybranou architekturu a zabývá se návrhem komunikačních zdrojů, které měl na starosti Bc. Filip Glazar při vývoji backendu. Během návrhu komunikačních zdrojů se nám osvědčilo spolupracovat – předešli jsme případným chybám a urychlili celkový proces vývoje.

2.4.1 Hypertext Transfer Protocol

Hypertext Transfer Protocol (HTTP) je aplikační protokol pro distribuované, kolaborativní a hypermediální informační systémy [27]. Používá se na webu pro datovou komunikaci a nejčastěji se používá k výměně hypertextových dokumentů ve formátu HTML. HTTP funguje na principu žádost–odpověď, kdy mezi sebou komunikuje klient se serverem. Klientem se většinou rozumí webový prohlížeč a serverem počítač, na kterém je uložena webová aplikace. Jde o formu komunikace, která je klíčová u Single Page Application (viz 2.2.2).

2.4.2 Representational State Transfer

Representational State Transfer (REST) je architektura rozhraní, navržená pro distribuované prostředí, se kterou poprvé přišel Roy Fielding, jeden ze spoluautorů protokolu HTTP, ve své disertační práci z roku 2000 [28]. REST je orientován datově, nikoli procedurálně, a určuje, jak se přistupuje k datům.

„Rozhraní REST je použitelné pro jednotný a snadný přístup ke zdrojům (resources). Zdrojem mohou být data, stejně jako stavy aplikace (pokud je lze popsat konkrétními daty). Všechny zdroje mají vlastní identifikátor URI a REST definuje čtyři základní metody pro přístup k nim.“ [29]

GET (Retrieve) Pro přístup ke zdrojům slouží základní metoda GET. Při opakovaném volání vrátí vždy stejný výsledek.

POST (Create) Pro vytvoření nových dat slouží metoda POST, která se často používá v HTML formulářích. V době volání není známý přesný identifikátor zdroje, protože ještě neexistuje.

PUT (Update) Pro změnu dat slouží metoda PUT. Funguje podobně jako metoda POST, jen je v této fázi už známa URI zdroje, jejíž data se mají změnit. Metoda se tedy musí volat na konkrétní zdroj.

DELETE Zdroj se dá snadno smazat metodou DELETE.

HTTP definuje návratové kódy, které používá server, když odpovídá na klientský požadavek. Návratové kódy vycházejí ze standardu [30] a jsou nezbytné pro rozhraní REST. Při provedení výše uvedených metod by server měl v hlavičce odpovědi pro klienta posílat správný návratový kód, aby ho klient mohl správně zpracovat.

2.4.3 Identifikátory zdrojů

Identifikátory zdrojů (URI) a definování všech přírodních metod měl na starosti Bc. Filip Glazar. Mou prací bylo dostupné zdroje využít a při konkrétních akcích aplikovat správné zdroje a metody. Jelikož je OptiLynx velmi komplexní informační systém, obsahuje mnoho zdrojů.

2.4.4 Struktura JSON

Pro reprezentaci dat jsme společně s Bc. Filipem Glazarem vybrali formát JSON (javascriptový objektový zápis), protože se s ním dobře pracuje v JavaScriptu. Zjednodušená ukázka struktury zakázky ve formátu JSON, která se posílá při použití metody GET na zdroj se zakázkami, vypadá následovně:

```
1 {
2   "id": 1371,
3   "customer": {
4     "name": "Karel Novotný"
5     "insurance": {
6       "shortcut": "VoZP",
7       "name": "Vojenská zdravotní pojišťovna"
8     }
9   },
10  "entered_at": "2018-04-24 00:00:00",
11  "deadline_at": "2018-05-04 00:00:00",
12  "state": "New",
13  "price": "1330",
14  "deposit": "958",
15  "measured": false,
16  "complaint": false,
17  "vip": false,
18  "kind": "full",
19  "glasses": [
20    {
21      "type": "hilux 1,5 hc",
22      "diameter": null,
23      "producer": "Hoya",
24      "priceLeft": "240",
25      "priceRight": "240",
26      "kind": "to_close",
27      "left_lens": {
28        "id": 757,
29        "name": "Hoya hilux 1,5 hc",
30      },
31      "right_lens": { ... },
32      "frame": { ... }
33    }
34  ],
35  "products": [],
36  "services": [
37    {
38      "code": 10,
39      "name": "zábrus do nové obruby kov,plast",
40      "price": "200",
41    }
42  ],
43  "vouchers": [
```

```
44  {
45    "dName": "Farionová Lenka",
46    "diagnosis": "h524",
47    "date": "2018-04-10",
48    "items": [
49      {
50        "code": "0091375",
51      }
52    ]
53  }
54 ]
55 }
```

Zdrojový kód 2.1: Struktura zakázky ve formátu JSON

2.5 Uživatelské rozhraní

Stejně jako se při programování volí již předpřipravené knihovny ve formě frameworků, vyplatí se i v případě navrhování uživatelského rozhraní sáhnout po existujících knihovnách. Ty většinou vznikají na základě konceptů, u kterých se jejich autoři snaží dodržovat základní pravidla použitelnosti a správného návrhu. Čím častěji se budou tyto koncepty používat ve veřejné sféře, tím spíše si na ně navyknou koncoví uživatelé.

2.5.1 Bootstrap

Bootstrap je knihovna neboli sada nástrojů určená pro vývoj frontend aplikace. Využívá HTML, CSS a JavaScript a obsahuje řadu hotových řešení – předpřipravený layout, typografii, obrázky, ikony, tlačítka, formuláře a další komponenty rozhraní, které se běžně používají na webu [31].

V době, kdy jsem začínal s diplomovou prací, vyšla nová verze frameworku Bootstrap. Jelikož jsem měl zkušenost se starší verzí Bootstrapu, lákala mě nová verze a diplomová práce byla skvělou příležitostí k vyzkoušení. Bootstrap 4 začal podporovat flexbox a původní CSS preprocesor LESS vystřídal za SASS; zdokonalil práci s layoutem, upravil některé komponenty a zapracoval na responsivitě tabulek a obrázků [32].

Pro první fázi vývoje jsem zvolil **Bootstrap 4**.

2.5.2 Material Design

Material Design (kódovým označením Quantum Paper) je vizuální jazyk vyvinutý společností Google. Poprvé byl představen na vývojářské konferenci Google I/O v roce 2014. Vedoucí designér Matías Duarte na konferenci [33] mluví o Material designu jako o papíru, zmiňuje, že na rozdíl od skutečného papíru

může digitální papír měnit tvar a chovat se inteligentně. Fyzikální vlastnosti papíru evokují v uživateli pocit, že se ho může dotknout.

Vágně lze Material Desing popsat jako jednotný systém kombinující teorii, zdroje a nástroje pro vytváření digitálních zážitků. Jeho cílem je sjednotit uživatelská rozhraní napříč všemi platformami (počítač, tablet, mobilní telefon). Google využívá principy a přesně definované uživatelské komponenty Material Designu v mobilním operačním systému Android a také ve všech svých webových aplikacích, díky čemuž s ním přišlo do styku již mnoho lidí. Rozsáhlá příručka Material Designu [34] důkladně popisuje všechny principy a specifikace. Kromě toho poskytuje již hotové a otestované uživatelské komponenty, které z tohoto konceptu vycházejí.

Material Design jsem v začátcích diplomové práce znal pouze okrajově. Až teprve v průběhu práce jsem zjistil, že pro Angular existuje knihovna s předpřipravenými komponentami, kterou má na svědomí přímo Google. Po mírné schizofrenii ohledně zvoleného uživatelského rozhraní, která mě doprovázela při finišování první fáze vývoje, jsem se nakonec rozhodl použít ve druhé fázi vývoje Material Design namísto Bootstrapu. Zpětně tento krok vyhodnocuji jako správný, protože i přes náročnější úpravy a předělovky v začátcích mi ušetřil pozdější práci při zdokonalování jednotlivých komponent. Také při uživatelském testování sklídl lepší ohlasy.

Realizace frontendu informačního systému OptiLynx

V této kapitole bude nejprve popsán časový plán realizace. Poté bude následovat detailní popis implementace informačního systému OptiLynx za pomoci webového frameworku Angular společně s přehledem všech komponent, které byly vytvořeny. Dále bude popsána práce se zdroji a následným používáním dat. Poté budou porovnány dvě klíčové technologie tvorby formulářů v Angularu, protože formuláře jsou podstatnou složku informačního systému.

V druhé části kapitoly bude podrobně popsán vznik a tvorba jednotlivých komponent informačního systému OptiLynx z hlediska uživatelského rozhraní. Budou rozebrány jednotlivé obrazovky systému na všechny klíčové složky a zhodnotí se, jaký přínos skýtají pro budoucího uživatele.

3.1 Časový průběh realizace

K vývoji informačního systému OptiLynx jsem se dostal poprvé v říjnu. Nejprve jsem byl seznámen s požadavky na systém a poté jsem dostal k dispozici drátěný model. Drátěný model jsem patřičně zanalyzoval (viz 1.7) a použil ho jako odrazový můstek při návrhu první verze.

Během listopadu jsem začal s kódováním šablon v HTML. Použil jsem frontend knihovnu Bootstrap 4.

V prosinci jsem se seznámil s webovým frameworkem Angular a začal s vývojem webové aplikace. Vytvořil jsem základní kostru aplikace.

Během ledna jsem na vývoji spolupracoval s Bc. Filipem Glazarem, který pracoval na backendové části aplikace a připravoval mi komunikační zdroje. Já mezitím využil zdrojů a začal pracovat s daty ze serveru v jednotlivých komponentách aplikace.

Koncem ledna byla připravená funkční aplikace obsahující pokladnu, zákazníky, zakázky a sklad. První verze postrádala některé funkcionality a ob-

sahovala chyby. Testování aplikace měl na starosti Bc. Oldřich Malec, který zdárně reportoval nalezené chyby.

V únoru jsem se z osobních důvodů nemohl věnovat vývoji na informačním systému OptiLynx, tudíž se vývoje frontend části ujal Bc. Filip Glazar, za což mu děkuji.

V březnu jsem se k vývoji opět vrátil. Průběžně jsem opravoval chyby a přidal do aplikace funkci párování faktur s dodacími listy a ostatními produkty.

Koncem března jsem začal předělávat uživatelské rozhraní z Bootstrapu do Material Designu, čímž jsem strávil celý následující měsíc.

18. dubna jsem s vedoucím práce Ing. Jiřím Hunkou vyrazil na uživatelské testování do oční optiky v Sedlčanech.

Poté jsem zpracoval připomínky z uživatelského testování a opravil nalezené chyby.

4. května jsem s vedoucím práce Ing. Jiřím Hunkou vyrazil na druhé testování do oční optiky v Benešově a do oční optiky ve Vlašimi.

3.2 Implementace ve frameworku Angular

V této části kapitoly bude detailně popsána implementace ve frameworku Angular společně s ukázkami některých zdrojových kódu.

3.2.1 Struktura informačního systému

Výsledná aplikace v Angularu se skládá z komponent, které jsou na sobě závislé v rámci stromové struktury. Informační systém OptiLynx je složený z následujících komponent:

- **auth** – Komponenta zabývající se autentifikací uživatele.
 - **login** – Komponenta s přihlašovacím formulářem a volbou pobočky a pokladny.
 - **logout** – Komponenta s odhlašovacím formulářem.
 - **auth-guard service** – Služba, která kontroluje, zda je daný uživatel přihlášený.
 - **auth service** – Služba, která ověřuje přihlášení uživatele.
- **customers** – Komponenta přehledu zákazníků.
 - **customer-data** – Komponenta obsahující kartu zákazníka. Slouží k vytváření nového zákazníka nebo upravování stávajícího zákazníka.
 - **customer-detail** – Komponenta zobrazující kartu zákazníka a seznam všech jeho zakázek.

- **customer service** – Služba, která obsahuje metody pro manipulaci s daty zákazníků.
- **exports** – Komponenta určená k exportu dat.
 - **export-insurance** – Komponenta provádějící export dat pro pojišťovny.
- **changelog** – Komponenta zobrazující seznam změn. Využívá **Markdown** syntaxi, což je odlehčený značkovací jazyk určený pro úpravu prostého textu [35].
- **navigation** – Komponenta hlavní navigace.
- **not-found** – Komponenta určená pro zobrazování chybové hlášky, která se zobrazí při dotazu na zobrazení neexistující stránky.
- **orders** – Komponenta zakázky.
 - **order-detail** – Komponenta detailu zakázky.
 - **order-modals** – Komponenty modálních oken. Zakázka používá modální okno pro poukaz a pro vytvoření nové úpravy čocky.
 - **order-products** – Komponenta obsahující hledání produktů a služeb.
 - **order-list** – Komponenta se seznamem všech zakázek společně s filtrováním.
 - **order service** – Služba, která obsahuje metody pro manipulaci s daty zakázek.
- **product** – Komponenta s detailem produktu. Lze v ní produkt vytvořit a také upravovat.
 - **product-history** – Komponenta zobrazující detail pohybu produktu mezi sklady.
- **terminal** – Komponenta pokladny.
 - **terminal-cashregister** – Komponenta pokladního deníku.
 - **terminal-close** – Komponenta obsahující formulář pro zavření pokladny.
 - **terminal-history** – Komponenta zobrazující historii pokladny.
 - **terminal-modals** – Komponenty modálních oken. Pokladna používá modální okna pro vydání zakázky, pro provedení platby, pro přidání položky na pokladnu a pro přidání výdaje.

- **terminal-open** – Komponenta obsahující formulář pro otevření pokladny.
- **terminal-summary** – Komponenta obsahující přehled pokladny.
- **terminal service** – Služba, která obsahuje metody pro manipulaci s daty na pokladně.
- **warehouse** – Komponenta skladu. Obsahuje přehled všech produktů.
 - **warehouse-insert** – Komponenta určená pro naskladňování produktů.
 - **warehouse-insert-history** – Komponenta zobrazující historii naskladnění.
 - **warehouse-invoice** – Komponenta přehledu faktur.
 - **warehouse-invoice-detail** – Komponenta detailu faktury.
 - **warehouse-invoice-pairing** – Komponenta určená pro párování faktur.
 - **warehouse-move** – Komponenta určená pro přesun produktů na jinou pobočku.
 - **warehouse-move-detail** – Komponenta zobrazující detail přesunu vybraných produktů.
 - **warehouse-move-history** – Komponenta zobrazující historii všech přesunů.
 - **warehouse-nav** – Komponenta skladové navigace.
 - **warehouse service** – Služba, která obsahuje metody pro manipulaci s daty na skladě.

3.2.2 Práce se zdroji

Pro získávání a ukládání dat je zapotřebí backend informačního systému. Ke snadnému přístupu ke zdrojům se používá rozhraní REST, ke kterému uživatel potřebuje znát identifikátory potřebných zdrojů. V Angularu k tomu slouží objekt typu **Observable**, který posílá notifikace [36].

V případě zakázek funguje manipulace s daty následovně – v zakázkové komponentě se zavolá metoda, která se připojí ke konkrétnímu zdroji skrz správný identifikátor zdroje (URI). Objekt typu Observable data postupně stahuje k sobě. Metoda pro stažení všech zakázek se nachází ve službách v souboru *orders.service.ts*, který obsahuje všechny metody komunikující se zdroji, které souvisejí se zakázkou.

```
1 private branchesUrl = process.env.API_URL + '/branches';  
2
```

```
3 getOrders(branchId: number): Observable<any> {
4   const url = `${this.branchesUrl}/${branchId}/orders`;
5   return this.http.get<any>(url);
6 }
```

Zdrojový kód 3.1: Observable pro získání všech zakázek z API zdroje

Pro získání dat je zapotřebí metoda, která bude objekt typu Observable poslouchat. Pro odposlouchávání streamu dat se používá metoda **subscribe**. Ta má definované tři manipulátory zpětného volání (callback function) – **next** pro průběžné zpracování přicházejících dat, **error** pro případnou chybu při přenosu a **complete** pro úspěšné ukončení odposlechu všech dat.

```
1 getOrders() {
2   let branchId = this.authService.user.branch_id;
3   this.authService.progressBarIncrement();
4
5   this.ordersService.getOrders(branchId).subscribe(
6     (orders) => {
7       this.orders = orders.data;
8     },
9     (error) => {
10      this.authService.progressBarIncrement();
11    },
12    () => {
13      this.authService.progressBarDecrement();
14    }
15  );
16 }
```

Zdrojový kód 3.2: Metoda pozorující Observable

3.2.3 Datové modely

Pro snadné používání dat, která jsou ve formátu JSON posílána ze serveru do webové aplikace pomocí Observable, lze vytvořit vlastní datové modely a k datům je přiřadit [37].

```
1 export class Order {
2   id: number;
3   state: string;
4   customer: Customer;
5
6   entered_at: Date;
```

```
7   deadline_at: Date;
8   issued_at: Date;
9   release: boolean;
10
11  glasses: OrderGlass[];
12  products: Product[];
13  services: ProductService[];
14
15  vouchers: Voucher[];
16  price: number;
17  sale: number;
18  finalPrice: number;
19  deposit: number;
20
21  kind: string;
22  vip: any;
23  complaint: boolean;
24  measured: any;
25  frames: number;
26
27  keywords: string;
28  filterType: string;
29 }
```

Zdrojový kód 3.3: Ukázka datového modelu zakázky v Angularu

3.2.4 Angular Forms

Jakýkoliv informační systém se neobejde bez formulářů, které po odeslání zpracují vyplněné údaje. Angular nabízí dva přístupy, jak formuláře vytvořit – první je svým přístupem založený na práci s modelem, druhý využívá reaktivní princip [38]. Nedá se říct, který je lepší. Nedá se ani říct, který používat, protože každý najde uplatnění v jiných situacích. Obecné srovnání Reactive Forms a Template-driven Forms je zpracováno v tabulce 3.1.

Template-driven Forms Formulář založený na šablonách a vytváří se především za pomoci HTML elementů [40]. Data formuláře jsou úzce svázána s datovým modelem. Pro jednoduché formuláře je tento způsob naprosto vyhovující a na implementaci rychlý.

Reactive Forms Formulář umožňující reaktivní styl programování, při kterém nejsou jeho data závislá na datovém modelu [40]. Pro vytvoření formuláře v šabloně stačí minimální HTML kód, protože se veškerá logika řeší na úrovni

Tabulka 3.1: Srovnání Reactive Forms a Template-driven Forms [39]

Reactive Forms	Template-driven Forms
Mnohem flexibilnější, ale náročnější na použití.	Snadný na použití.
Vhodný pro komplexní scénáře.	Vhodný pro jednoduché scénáře. U složitějších scénářů selhává.
Není závislý na datovém modelu v komponentě.	Váže se na data v komponentě.
Formulář je potřeba definovat a pracovat s ním přímo v komponentě. Nepotřebuje žádné speciální HTML značky.	Formulář stojí na HTML značkách, v komponentě stačí jen deklarace proměnné.
Data se ukládají do speciálního objektu.	Data jsou závislá na vytvořeném objektu v komponentě.
Snadné použití validace a snadná tvorba vlastních validačních pravidel.	Základní validace skrz HTML, jinak je potřeba pro speciální validace psát vlastní metody.
Synchronní.	Asynchronní.

komponenty. Reactive Forms mají speciální strukturu, která v sobě nese informace o veškerých validacích a testech. Ve výsledku je reaktivní formulář lepší z programátorského hlediska, ale jeho vytvoření je ze začátku náročnější.

Ze začátku jsem formuláře vytvářel pomocí Template-driven Forms, ale jakmile jsem objevil kouzlo Reactive Forms, rozhodl jsem se původní předělat. Reactive Forms mají smysl u komplexních formulářů. Proto jsem je použil u zakázky, jejíž vyplnění odpovídá velmi složitému scénáři. Výhodné bylo i použití u tvorby zákazníka, kde jsem si musel napsat funkci pro validaci rodného čísla (`ValidateBirthNumber`).

```

1 this.customerForm = new FormGroup({
2   name: new FormControl(null, [Validators.required]),
3   pin: new FormControl(null, [ValidateBirthNumber]),
4   insurance: new FormGroup({
5     code: new FormControl(null)
6   }),
7   email: new FormControl(null),
8   phoneNumber: new FormControl(null),
9   discount: new FormControl(null),
10  vip: new FormControl(false),
11  address: new FormGroup({
12    street: new FormControl(null),

```

```
13   houseNumber: new FormControl(null),
14   city: new FormControl(null),
15   zip: new FormControl(null),
16   country: new FormControl('Česka_republika')
17 }
18 });
```

Zdrojový kód 3.4: Reactive form u zákazníka

3.2.5 Angular Pipe – transformační filtry

Syntaxe pro vkládání obsahu z proměnných do šablon používá zápis dvou kudrnatých závorek – name [41]. Někdy je však potřeba data transformovat do přijatelnějšího formátu, například při zobrazování datumů, měny. K tomu lze v Angularu využít tzv. Pipes [42]. Pro tyto potřeby jsem si napsal pár vlastních filtrů, které jen mění výpis, přidávají měnu, přepisují anglické texty na české či třeba převádějí čas do jiného formátu. Zkoušel jsem i náročnější filtry pro řazení nebo hledání, ale při práci s větším množstvím dat jsem zjistil, že nejde o elegantní řešení a že je lepší i z hlediska výkonu filtrovat rovnou v komponentě.

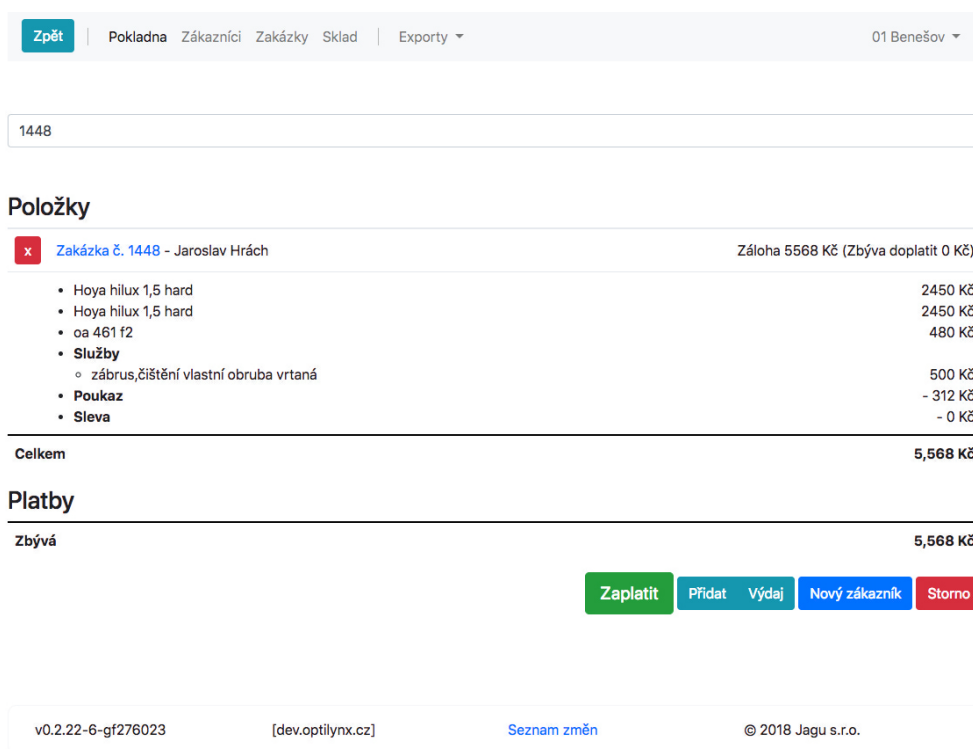
```
1 @Pipe({
2   name: 'currencyCzech',
3   pure: false
4 })
5 export class CurrencyCzechFilterPipe implements PipeTransform {
6   public transform(value: string, vat?: string): string {
7     let finalVat = vat ? Number(vat) + 1 : 1;
8     let result = Number(finalVat) * Number(value);
9     result = result || 0;
10    return +(result).toFixed(12) + '₴Kč';
11  }
12 }
```

Zdrojový kód 3.5: Angular Pipe – filtr převádějící číslo na českou měnu

3.3 Tvorba uživatelského rozhraní

V této části kapitoly bude nejprve zmíněno uživatelské rozhraní v první verzi, kdy využívalo knihovnu Bootstrap. Poté bude detailně rozebrána verze využívající Material Design.

3.3. Tvorba uživatelského rozhraní



Obrázek 3.1: OptiLynx – Pokladna (verze používající knihovnu Bootstrap 4)

Během návrhu jednotlivých komponent uživatelského rozhraní jsem se řídil výrokem Steva Kruga, jednoho z předních odborníků v oblasti použitelnosti: „*Nenuťte mě přemýšlet.*“ [43]

3.3.1 Bootstrap

Při první fázi vývoje jsem použil Bootstrap, poněvadž jsem s ním měl již nějakou zkušenost. Provázat Bootstrap s Angularem se ukázalo jako poměrně komplikovaný proces. Problém Bootstrapu je v tom, že jde pouze o sadu nástrojů a nastavených prvků, nikoli o designérský koncept s filosofií. Výsledek se mi po estetické stránce nezdál příliš uspokojivý a připomínal mi spíše propracovaný prototyp než hotovou verzi. Vzhled by se dal samozřejmě upravit, nicméně by to vyžadovalo velké zásahy do knihovny Bootstrap a stálo by to mnoho času.

Ukázka informačního systému OptiLynx při použití knihovny Bootstrap je na obrázku 3.1 (pokladna) a 3.2 (detail zakázky).

3. REALIZACE FRONTENDU INFORMAČNÍHO SYSTÉMU OPTILYNX

[Zpět](#) | Pokladna Zákazníci Zakázky Sklad | Exportsy ▾ 01 Benešov ▾

Zakázka č. 1448

Jaroslav Hrách

[Upravit údaje zákazníka](#)

Údaje o zákazníkovi Rodné číslo: 9303252288 Pojišťovna: VoZP (201) VIP: Ne Trvalá sleva:	Kontakt na zákazníka E-mail: me@jaroslavhrach.cz Telefonní číslo: +420 604 994 558	Adresa zákazníka Pešlova 361/22 19000 Praha Česká republika
---	---	---

Čočky

Pravé	Sféra	Cylindr	Osa	Adice	Prisma	Osa prisma	PD	Výška	Zisk dat
Levé	Sféra	Cylindr		Adice					<input type="radio"/> ZP <input type="radio"/> Klient <input type="radio"/> Naměřeno

Výrobce: Typ: Kód P: Cena P: Kód L: Cena L:

Průměr: Průměr Druh:

[Nová úprava](#)

Vložte úpravu

[Přidat obsah do zakázky](#)

Obsah zakázky

Čočka P: Hoya hilux 1,5 hard	2450 Kč	<input type="text"/>	Po slevě: 2450 Kč	Upravit	Odebrat ze zakázky
Čočka L: Hoya hilux 1,5 hard	2450 Kč	<input type="text"/>	Po slevě: 2450 Kč		
Obruba: oa 461 f2	480 Kč	<input type="text"/>	Po slevě: 480 Kč		Změnit obrubu

[Přidat další obsah do zakázky](#)

Přidat produkt nebo službu

Hledat ve skladu a službách

Produkty				
Služby				
zábrus,čištění vlastní obruba vrtaná	500 Kč	<input type="text"/>	Po slevě: 500 Kč	Odstranit

Cena: 5880 Kč **Poukazy: 312 Kč** **Sleva: 0 Kč** **Po slevě/poukazu: 5568 Kč** **Zaplaceno: 0 Kč** **Doplatek: 5568 Kč**

Zadáno	<input type="text"/>	Vlastnosti	Obruby	Poukazy	Uložit
Termín	<input type="text"/>	<input type="checkbox"/> Reklamační <input type="checkbox"/> VIP <input type="checkbox"/> Měření refrakce	<input checked="" type="radio"/> Celá <input type="radio"/> Drážka <input type="radio"/> Vrtaná	Tisk	Uložit a zaplatit
Vydáno	<input type="text"/>				

v0.2.22-6-gf276023 [dev.optilynx.cz] [Seznam změn](#) © 2018 Jagu s.r.o.

Obrázek 3.2: OptiLynx – Zakázka (verze používající knihovnu Bootstrap 4)

3.3.2 Angular Material

Pro využití Material Designu v Angularu jsem musel použít knihovnu Angular Material. Ta v sobě obsahuje řadu komponent připravených k použití. [44]

3.3.3 Angular Flex-Layout

Jelikož knihovna Angular Material obsahuje pouze komponenty, je potřeba řešit layout. Velmi často se v takových situacích používá knihovna Angular Flex-Layout [45], která do HTML elementů implementuje speciální atributy pro práci s flexboxem.

3.3.4 Barvy uživatelského rozhraní

Barvy v Material Designu jsou definovány na celé škále jejich odstínů – od tmavých a tlumených po jasné až velmi výrazné. Koncepčně se v designu vyskytují dvě barvy – hlavní a akcentní [46]. Pro speciální případy se využívá varující barva. Oficiální dokumentace uvádí paletu odstínů těchto základních barev, které lze použít. Barvy mají následující smysl:

- **hlavní barva (primary color)** – udává styl vzhledu a má nejčastější výskyt,
- **akcentní barva (accent color)** – používá se u interaktivních prvků, které vykonávají nějakou akci,
- **varující barva (warn color)** – používá se u chybových stavů/hlášek.

Angular Material umožňuje snadnou tvorbu vlastního barevného tématu [47]. V souboru s kaskádovými styly lze předefinovat proměnné všech typů barev a lze také pomocí speciálních funkcí upravovat jejich odstíny. Při změně dvou hodnot je možné změnit celé barevné téma.

Pro informační systém OptiLynx jsem zvolil fialovou jako hlavní barvu a růžovou jako akcentní barvu. U varující barvy jsem dodržel normu ISO 22324 [48], která se zabývá výstražnými barvami, a použil červenou barvu.

3.3.5 Pokladna

Pokladna (obrázek 3.3) je koncipovaná tak, aby na ní uživatelé trávili co nejméně času a nemuseli přecházet do jiných sekcí. Pole pro hledání prohledává všechny produkty, služby, zakázky a zákazníky. Uživatel by měl při scénáři obsluhy zákazníka nejprve použít hledání. U prodeje produktů stačí vyhledat dané produkty. Pokud si chce zákazník koupit brýle, uživatel by měl nejprve zadat jméno zákazníka, aby zjistil, zda je uživatel v databázi. Teprve potom buď vytvoří nového zákazníka, nebo přejde rovnou na tvorbu zakázky.

3. REALIZACE FRONTENDU INFORMAČNÍHO SYSTÉMU OPTILYNX

The screenshot displays the 'Pokladna' (POS) interface. At the top, there is a navigation bar with 'OptiLynx', 'Pokladna', 'Zákazníci', 'Zakázky', 'Sklad', and 'Export'. A search bar contains '1448'. Below this, a section titled 'Položky v pokladně' (Items in the till) shows a receipt for order #1448 by Jaroslav Hrách. The total amount is 3000 Kč, with 4158 Kč remaining to be paid. The receipt lists the following items and prices:

Item	Price (Kč)
Pravé: Hoya hilux 1,5 hard	2450
Levé: Hoya hilux 1,5 hard	2450
Obruba: oa 461 f2	480
Produkt: mr. gain Shrek 160	1590
Služba: zábrus,čištění vlastní obruba vrtaná	500
Poukaz	-312,0
Sleva	0
Celkem zaplatit	3000

Below the receipt, the 'Platby' (Payments) section shows:

Payment Method	Amount (Kč)
Hotovost	2000
Kartou	500
Zbývá doplatit	500

At the bottom of the receipt area, there are buttons for 'Storno pokladny' and 'Zaplatit'. The footer contains version information (v0.2.16-94-g8e0954b), the website (beta.optilynx.cz), a 'Seznam změn' link, and the copyright notice (© 2018 Jagu s.r.o.).

Obrázek 3.3: OptiLynx – Pokladna

Pokladna dále disponuje dvěma modálními okny pro přidání speciální položky na pokladnu a přidání výdaje. Při přidání zakázky na pokladnu se zobrazí kompletní obsah zakázky i s cenami. Při placení je možné zvolit víc typů, které se také zobrazují na pokladně v části s platbami.

3.3.6 Rozbalovací nabídka

Napříč systémem je u hledání zobrazována rozbalovací nabídka (obrázek 3.4). Zobrazí se v okamžiku, kdy uživatel označí pole pro hledání. V rozbalovací nabídce se dá pohybovat a vybírat jednotlivé položky klávesnicí.

3.3.7 Platební karta

Při placení na pokladně se otevírá modální okno s volbou typu platby (obrázek 3.5). Nejčastější platby (hotovost, karta, stravenky) obsahují názornou

3.3. Tvorba uživatelského rozhraní

Hledat položku
144

Produkty + Vytvořit novou položku

coloroptic 3f1443	Skladem: 1 ks	320 Kč	Detail
gafas boom 1443/2	Skladem: 1 ks	2350 Kč	Detail
dominic mod.144 c1	Skladem: 1 ks	1680 Kč	Detail
UNIOPTIK 2-11442-5017	Skladem: 0 ks Komise	1490 Kč	Detail
s 14444of lichtenberg	Skladem: 0 ks	1450 Kč	Detail
UNIOPTIK 5-20144-5420	Skladem: 0 ks Komise	1360 Kč	Detail

Zakázky

#1448	Jaroslav Hrách	Nová	7470 Kč	Detail
#1445	Filip Glazar	Nová	1160 Kč	Detail

Obrázek 3.4: OptiLynx – Rozbalovací nabídka

Zbývá zaplatit 500 Kč

Vyberte typ placení

Hotově Kartou Stravenkami

Kolik zaplatíte výše uvedeným způsobem? *
500 Kč

Obrázek 3.5: OptiLynx – Platební karta

ikonu. Dále je možné zaplatit jiným způsobem, který není tak častý. K tomu slouží výběrové pole obsahující další možnosti.

3.3.8 Zákazník

Detail zákazníka (obrázek 3.6) se skládá z jeho osobních údajů, které se dají snadno editovat kliknutím na ikonu tužky. Při editaci se rozmístění všech údajů zachová, jen se holý text vymění za formulářové prvky. Pomocí uživatelského testování se rozmístění jednotlivých položek upravilo tak, aby bylo pro vyplňující

3. REALIZACE FRONTENDU INFORMAČNÍHO SYSTÉMU OPTILYNX

The screenshot displays the OptiLynx web interface. At the top, a navigation bar includes the logo, currency (CZK), user profile (Zákazníci), menu (Zakázky), warehouse (Sklád), and export (Export) options. The main content area is titled 'Detail zákazníka' and features a red back arrow and a '+ Vytvořit novou zakázku' button. The customer's name 'Jaroslav Hrách' is prominently displayed. Below this, three columns of information are provided: 'Údaje o zákazníkovi' (personal details), 'Adresa zákazníka' (address), and 'Ostatní informace' (additional info). A section titled 'Zákazníkovy zakázky' (Customer orders) follows, showing a table with columns for order ID, status, date, price, and contact info. The first order, #1448, is highlighted in yellow. Below the table, there are two sub-tables: 'Produkty' (Products) and 'Služby' (Services).

#1448 Jaroslav Hrách	Stav	Zadáno	Cena	Doplatek	Kontaktovat	Detail	Výdej
	Nová	08. 05. 2018	7158 Kč	7158 Kč	SMS		

Jiné	Sféra	Cylindr	Osa	Adice	Prisma	Osa Prisma	PD	Výška	Cena
Pravé: Hoya hilux 1,5 hard	-3.00	-	-	-	-	-	30	-	2450 Kč
Levé: Hoya hilux 1,5 hard	-2.75	-	-	-	-	-	29	-	2450 Kč
Obruba: oa 461 f2									480 Kč

Produkty	Cena	Služby	Cena
mr. gain Shrek 160	1590 Kč	zábrus,čištění vlastní obru...	500 Kč

Obrázek 3.6: OptiLynx – Detail zákazníka s přehledem jeho zakázek

ciho uživatele nejvýhodnější.

Dále se na detailu zákazníka zobrazuje přehled všech jeho zakázek.

3.3.9 Zakázka – přehledová karta

Přehledová karta zakázky (dolní část obrázku 3.6) obsahuje základní informace o zakázce společně s jejím obsahem. Při uživatelském testování jsem zjistil, jaké informace na přehledu zobrazovat, aby je uživatelé využili. Zobrazuje seznam všech čoček společně s obrubami, dále také seznam přidáných produktů a služeb.

Horní pruh má variabilní barvu v závislosti na stavu zakázky:

- nová zakázka – žlutá,
- připravená zakázka – zelená,
- vydaná zakázka – šedá,
- zrušená zakázka – červená,

Čočky Smazat čočky ze zakázky

DD	Sféra	Cylindr	Osa	Adice	Prisma	Osa Prisma	PD	Výška	
Pravé	-3						30		Upřesnění ▾
Levé	-2,75						29		Upřesnění ▾
Obruba	oa 461 f2								Odebrat obrubu

Výrobce: Hoya | Typ: hilux 1,5 hard | Průměr: Průměr | Druh: Do dálky

Vložitě úpravu Vytvořit novou úpravu

Typ obruby **Zisk dat**

Celá ZP
 Drážka Klient
 Vrtaná Naměřeno

Cena			
Pravá čočka: Hoya hilux 1,5 hard	Cena: 2450 Kč	1225	Po slevě: 1225 Kč
Levá čočka: Hoya hilux 1,5 hard	Cena: 2450 Kč	1225	Po slevě: 1225 Kč
Obruba: oa 461 f2	Cena: 480 Kč	100	Po slevě: 380 Kč
Celková cena za čočky a obrubu:			2830 Kč

+ Přidat další čočky

Obrázek 3.7: OptiLynx – Formulář pro vkládání čoček

- reklamovaná zakázka – oranžová.

3.3.10 Zakázka – Čočky

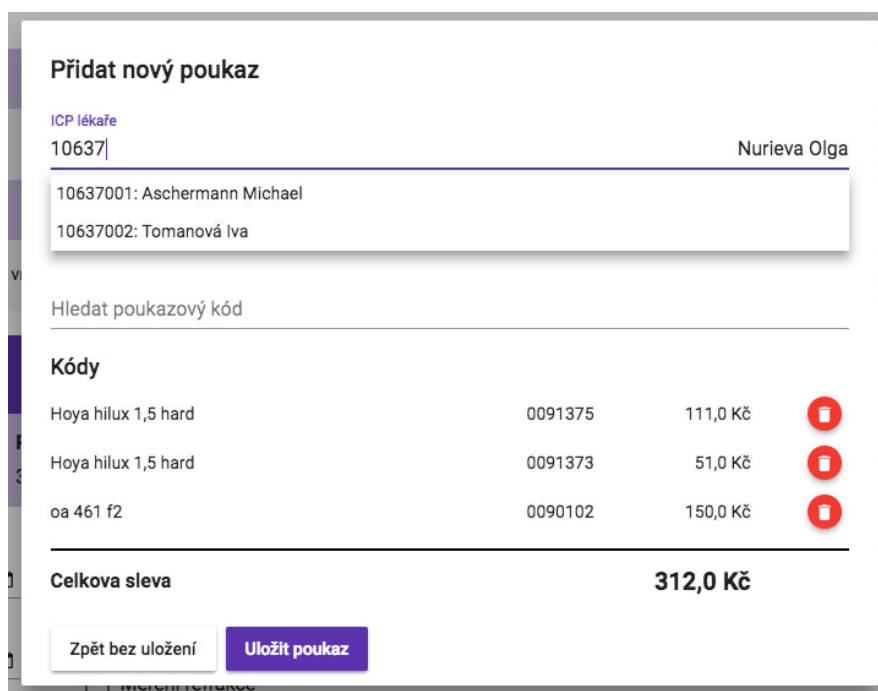
Při uživatelském testování se ukázalo, že uživatelé neradi klikají, když nemusejí. Na základě jejich připomínek jsem kompletně předělal formulář pro přidávání čoček (obrázek 3.7). Dalším požadavkem bylo speciální přecházení mezi poli formuláře. Toho jsem dosáhl použitím HTML atributu **tabindex**, kterým se dá nastavit pořadí procházených prvků formuláře.

Jedním z požadavků na vyplňování vlastností čoček bylo, aby se u některých vlastností zobrazovala znamínka. Bohužel jsem narazil na omezení prohlížeče, nejde zobrazovat znamínko + v případě použití vstupního pole s typem číslo (type number), které ale umožňuje nastavit rozmezí a vkládání čísel pomocí šipek klávesnice po skocích.

3.3.11 Zakázka – Poukaz

Poukaz (obrázek 3.8) se skládá z identifikačního čísla lékaře, diagnózy, data vydání poukazu a poukázkových kódů, které obsahují slevy na vybrané položky.

3. REALIZACE FRONTENDU INFORMAČNÍHO SYSTÉMU OPTILYNX



Obrázek 3.8: OptiLynx – Modální okno sloužící pro přidání nového poukazu

V první verzi informačního systému OptiLynx byli uživatelé nuceni psát ceny ručně, což se ukázalo jako špatný způsob, který akorát vedl k chybám. Proto jsem poukaz předělal do interaktivnější podoby. U všech vstupních polí se při psaní zobrazí rozbalovací nabídka, která v závislosti na psaném textu asynchronně hledá v databázi a napovídá výsledek.

3.3.12 Zakázka – Rekapitulace

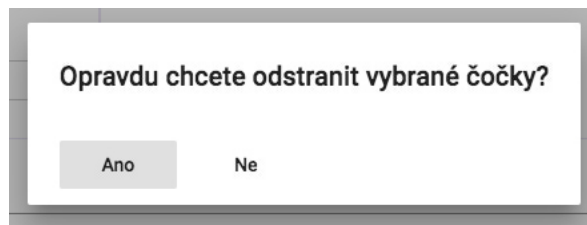
Rekapitulace zakázky (obrázek 3.9) obsahuje všechny podstatné informace o ceně zakázky – celková cena zakázky, hodnota poukazu, sleva, výsledná cena po odečtení všech slev a hodnoty poukazu, dosud zaplacená částka a doplatek k zakázce. Cena po odečtení všech slev a hodnoty poukazu je nejdůležitější, protože udává, kolik má zákazník zaplatit, proto je zvýrazněná.

Datum zadání zakázky odpovídá aktuálnímu datu a termín odpovídá třem pracovním dnům. U tlačítka s poukazem se zobrazuje číslo, které indikuje, kolik poukazů zakázka obsahuje (lze přidat na jednu zakázku víc poukazů).

3.3.13 Modální dialogové okno

Modální okno je grafický řídicí prvek podřízený hlavnímu oknu aplikace. Uživatel je nucen zaměřit svou pozornost na nově vzniklé okno. Modální dialogová okna se používají zpravidla u operací, které budou mít výrazný vliv

Obrázek 3.9: OptiLynx – Rekapitulace zakázky



Obrázek 3.10: OptiLynx – Dialogové okno

na následující stav aplikace a vyžadují po uživateli potvrzení nebo zamítnutí akce. [49]

V rámci zachování jednotného stylu bylo nativní modální dialogové okno webového prohlížeče nahrazeno univerzálním modálním oknem, které používá Material Design (obrázek 3.10).

```

1 @Component({
2   selector: 'app-alert-modal',
3   template:
4     '<h1 mat-dialog-title>{{ passedData.title }}</h1>
5     <mat-dialog-actions>
6       <button mat-button [mat-dialog-close]="true">
7         Ano
8       </button>
9       <button mat-button [mat-dialog-close]="false">
10        Ne
11      </button>
12    </mat-dialog-actions>'
13 })
14 export class AlertModalComponent {
15   public itemForm: FormGroup;
16
17   constructor(

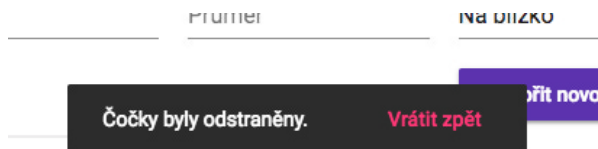
```

```
18   @Inject(MAT_DIALOG_DATA) public passedData: any
19   ) { }
20 }
```

Zdrojový kód 3.6: Univerzální potvrzovací modální okno

3.3.14 Snackbar

Uživatel musí být informován o každé akci, kterou provede. V nejhorším případě by měl mít možnost vrátit danou akci zpět. K tomu jsem použil tzv. **snackbar** (obrázek 3.11) neboli řádek popisující provedenou operaci, který se zobrazuje ve spodní části obrazovky po provedení operace. Když uživatel například omylem smaže nějakou položku a vzápětí si uvědomí, že by ji chtěl vrátit zpět, stále může kliknout na tlačítko **vrátit zpět** dostat se do původního stavu aplikace.



Obrázek 3.11: OptiLynx – Snackbar

3.3.15 Tlačítko zpět

Uživatel by měl mít vždy možnost vrátit se do předchozího stavu. K tomu slouží šipka v levém horním rohu.

3.3.16 Kalendář u datumů

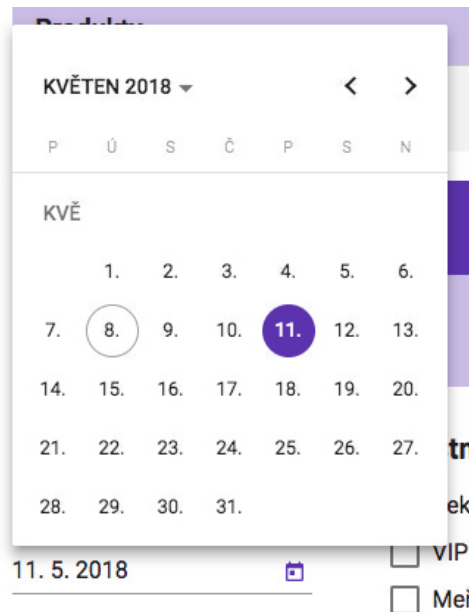
Pro snadnou volbu datumu byl použit přehledný kalendář (obrázek 3.12) dostupný v knihovně Angular Material (datepicker). Umožňuje snadné listování mezi měsíci i roky a dá se v něm pohybovat pomocí klávesnice.

3.3.17 Slevové pole

V praxi se ukázalo, že uživatelé často používali slevy v procentech a pokaždé si výslednou hodnotu slevy museli dopočítávat. Proto jsme společně s Bc. Filipem Glazarem a Ing. Jiřím Hunkou vymysleli a vytvořili vstupní pole, do kterého se dá zadávat jak číselná sleva, tak procentuální sleva.

3.3.18 Progress bar

Dalším z důležitých prvků aplikace je indikátor stavu. Jedná se o jeden bod z Nielsova desatera (viz 1.3.1). Při optimalizaci výkonu webových aplikací



Obrázek 3.12: OptiLynx – Kalendář u datumů

je třeba mít na paměti 3 hlavní časové limity (které jsou určeny vnímacími schopnostmi člověka). [50]

Základní doporučení ohledně doby odezvy je přibližně padesát let stejné [51]:

- **0.1 sekundy** je limit, u kterého uživatel pociťuje, že systém reaguje okamžitě,
- **1.0 sekunda** je limit, u kterého se nepřerušuje myšlenkový tok uživatele,
- **10 sekund** je limit pro udržení pozornosti uživatele.

Angular Material nabízí progress bar ve čtyřech variantách:

- **determinate** – operace, u kterých jsou známa procenta v jejich průběhu,
- **indeterminate** – operace, u kterých je uživatel požádán, aby počkal, dokud se něco nedokončí a není nutné uvádět, jak dlouho daná operace bude trvat,
- **buffer** – operace, u kterých se chce uživateli naznačit nějaká aktivita,
- **query** – slouží pro situace, kdy se chce uživateli zobrazit předběžné načítání (až do skutečného načtení).

3. REALIZACE FRONTENDU INFORMAČNÍHO SYSTÉMU OPTILYNX

Pro účely informačního systému OptiLynx byla vybrána varianta **query**, protože se vždy stahují data ze serveru a dopředu se neví, jak dlouho to bude trvat. Jelikož se v aplikaci vytváří víc nezávislých dotazů, které se zpracovávají asynchronně, bylo potřeba implementovat slučování jednotlivých dotazů, aby se progress bar zobrazil jen jednou.

Testování informačního systému OptiLynx

Tato kapitola se bude zabývat jednotlivými kroky testování informačního systému OptiLynx. Ukáže se, zda prvotní návrh uživatelského rozhraní, který odpovídal spíš propracovanějšímu prototypu nežli finálnímu produktu, splňuje všechny náležitosti, které byly na informační systém kladeny. Popíše, jak s ním byli uživatelé spokojeni při reálném provozu.

Poté bude následovat detailní popis několika uživatelských testování, která proběhla přímo se zaměstnanci očních optik na vybraných pobočkách. V této fázi testování bude použita nová verze informačního systému OptiLynx disponující Material Designem, což přinese úplně nové poznatky.

Kapitolu zakončí podrobné vyhodnocení, při kterém budou vypíchnuty největší přínosy testování.

4.1 Počáteční testování při reálném provozu

Začátkem února byla dokončena první verze informačního systému OptiLynx. Ve stejnou dobu byla nasazena rovnou do tří poboček společnosti OČNÍ OPTIKA BENEŠOV s.r.o., konkrétně do očních optik v Benešově, ve Vlašimi a v Sedlčanech. Byla to skvělá příležitost pro získání zpětné vazby od uživatelů.

Počáteční fázi testování nejlépe vystihuje přísloví „*zvyk je železná košile*“. Zaměstnanci jednotlivých poboček to měli zpočátku těžké; byli zvyklí na jiný systém a vzápětí se museli naučit pracovat s novým. Často se dožadovali starých funkcí, aby dosáhli výsledku, ke kterému se dá v novém systému dospět mnohem elegantnější cestou.

Jako velký problém se ukázal i způsob, jakým zaměstnanci ovládají počítač, potažmo klávesnici. Ze starých systémů si vytvořili zlozvyk u vyplňování formulářů – při stisknutí klávesy Enter na pravém okraji klávesnice očekávají skok na další prvek formuláře. Tuto funkcionalitu přitom v dnešní době plní

speciální klávesa tabulátor. Při zmáčknutí klávesy tabulátoru skočí výběr na následující prvek formuláře a v případě držení klávesy shift a zmáčknutí klávesy tabulátoru se výběr vrátí na předchozí prvek formuláře. S tím souvisí celková nespokojenost s procházením pomocí klávesnice, na které zaměstnanci dost lpěli. Hodně si stěžovali na to, „že v tomhle novém systému musí pořád jen klíkat myš“ a že je to zdržuje při vyplňování potřebných údajů. Nejhuře z toho vyšla zakázka, která obsahovala přibližně čtyřicet prvků k vyplnění a svým rozsahem a rozmanitostí odpovídala velmi komplexnímu formuláři. U zakázky se uživatelé také dožadovali netradičního procházení mezi jednotlivými prvky formuláře. Svou logiku to mělo, ale tu jsme se dozvěděli až při prvním uživatelském testování.

Informační systém OptiLynx se ve své rané verzi snažil uspokojit především ty nejdůležitější potřeby uživatelů a kladl důraz na běžné procesy, které jsou nutné při obsluze zákazníka. V praxi se ukázalo, že systém u některých procesů postrádá potřebné funkcionality, o kterých jsme i přes patřičnou analýzu vůbec nevěděli. Nezbytné prvky a drobné problémy jsme se s Filipem Glazarem snažili doimplementovat ještě do staré verze, aby byla práce pro zaměstnance optik méně frustrující. Řešili jsme:

- lepší řazení a filtrování u zákazníků a skladových položek,
- více typů plateb na pokladně (především možnost platit stravenkami),
- párování faktur k dodacím listům či samostatným produktům,
- pokladní deník,
- víc poukazů v zakázce,
- slevy na položky v zakázkách,
- absence tlačítka zpět – podstatný bod z heuristické analýzy (viz 1.3.1),
- validátor rodného čísla,
- export pro pojišťovny.

4.2 Uživatelské testování nové verze

Zpětná vazba z první fáze testování informačního systému při provozu na pobočkách mi přinesla mnoho nových informací a poznatků. Ty jsem využil při návrhu a realizaci nové verze, kdy jsem vyměnil Bootstrap (viz 2.5.1) za Material Design (viz 2.5.2). Novou verzi jsem testoval na třech pobočkách ve dvou etapách.

V polovině dubna jsme se společně s vedoucím práce Ing. Jiřím Hunkou vydali na pobočku v Sedlčanech, kde jsme zjistili chyby a nedostatky nové

4.2. Uživatelské testování nové verze

verze. V následujících čtrnácti dnech jsem chyby a nedostatky z první etapy zpracoval a udělal pár dalších vylepšení. Začátkem května jsme opět s vedoucím práce vyrazili na uživatelské testování, tentokrát však na pobočky do Benešova a do Vlašimi.

K testování jsem využil svůj starý nevyužitý poukaz na brýle a optické pomůcky (obrázek 4.1), který jsem dostal při návštěvě svého očního lékaře koncem roku 2017 a nevyužil jsem ho. Díky němu bylo možné simulovat naprosto přesně celý proces, od vytvoření nového zákazníka přes tvorbu nové zakázky se všemi jejími náležitostmi až po naskladnění všech položek zakázky, a finální prodej zboží na pokladně.

Kód pojišťovny: 201

POUKAZ NA BRÝLE A OPTICKÉ POMŮCKY

poř. č. _____

Skupina pomůcky **09**

Přijmení	Předpis	Sféra dioptrie	Cylindr Dp	Osa	Prisma Dp	Basis	Kód	Cena
Jméno: HRÁCH	pravé oko							
Číslo pojistnice: JAROSLAV	DO DÁLKY	-3.0						
Bydliště (adresa): 9303252288	levé oko	2.75						
IVANČICKÁ 509 PRAHA 9 LETŇANY 19900	NA BLÍZKO (addice)							
<input type="checkbox"/> hradí pojišťovna	Jiná optická pomůcka:							
<input type="checkbox"/> spoluúčast pacienta	Bifokální zatavené							
<input type="checkbox"/> hradí pacient	Bifokální Franklin							
	Výkony							
	Obruba							
	Výměna skel							
	Tvrzení ano							
	Absorpční vrstva %							
								Celkem

razítko poskytovatele, jmenovka a podpis lékaře

Dne: 18. října 2017

Obrázek 4.1: Poukaz na brýle a optické pomůcky, který byl použit při uživatelském testování

Při každém uživatelském testování byla nahrávána obrazovka počítače, aby zaznamenala, jak testující subjekty používají novou verzi informačního systému OptiLynx. Během uživatelského testování a i při následných diskusích s uživateli byl nahráván zvuk pomocí diktafonu. Videozáznamy a audiozáznamy jsem využil při vyhodnocení testování.

4.2.1 Scénář uživatelského testování

Uživatelské testování probíhalo podle běžného procesu, kdy přijde zákazník do oční optiky a chce si pořídit nové brýle. Pro takový případ jsem si přichystal následující scénář:

V poslední době jsem začal pociťovat mírné zhoršení zraku do dálky. Objednal jsem se tedy na kontrolu zraku do své oční ordinace. Ošetřující lékař mi při vyšetření zraku zjistil zhoršené vidění do dálky o tři čtvrtě dioptrie na pravém oku a o půl dioptrie na levém oku. Předepsal mi poukaz na nové brýle. Vydal jsem se tedy do oční optiky s tím, že si chci pořídit nové brýle

a nejraději bych využil nějaké akce na čočky. K brýlím by se mi hodilo také nové pouzdro. Nemám s sebou žádnou hotovost, takže bych případnou zálohu na brýle platil kartou. Až by byly brýle naskladněné, přišel bych si pro ně na pobočku, tentokrát však i s hotovostí, protože bych věděl, kolik mám doplatit.

4.3 Oční optika Sedlčany

První uživatelské testování proběhlo ve středu 18. dubna 2018 na pobočce v Sedlčanech. Společně s vedoucím práce Ing. Jiřím Hunkou jsme provedli testování s oční technikou a odpovědnou vedoucí sedlčanské optiky. Pobočka v Sedlčanech se stará především o vytváření zakázek pro klienty a prodej zboží, které se musí naskladňovat. Žádné složitější operace se skladem však zaměstnankyně nedělají.

4.3.1 Jana Blandová

Prvním testovaným uživatelem byla technička Jana Blandová. Po přihlášení do systému a obdržení mého poukazu od doktora se vrhla na vytvoření nového zákazníka. Při vyplňování adresy napsala chybně město do políčka určeného k vyplnění ulice, což bylo způsobeno spíš nepozorností, než že by měla problém najít správné pole.

Poté se Jana Blandová pustila do tvorby zakázky. Během vyplňování parametrů z poukazu mi důsledně změřila hodnotu sférických dioptrií (DP). Zasekla se u vyplňování ceny s odůvodněním, že v tento moment částku ještě neví, protože dosud nevybrala žádný typ čoček. K tomu využila papírový katalog a vybrala odpovídající typ. Vyplnila cenu a teprve potom se dostala k políčkům výrobce a typu čočky.

Vyplňování poukazu proběhlo bez problémů, Jana Blandová ocenila nový způsob, kdy informační systém při psaní napovídá ze seznamu doktorů, diagnóz a poukázkových kódů. Trochu zaskočená byla po potvrzení formuláře, který ji přesměřoval na přehled všech vyplněných poukazů. Zpočátku si nebyla jistá, jak modální okno zavřít.

Po zaplacení zálohy nám oční technička předvedla, jak vytváří zakázku v systému **Hoya iLog**, čímž přesně simulovala proces, který provádí při objednávání čoček u dodavatele. Následně bez problémů naskladnila v informačním systému OptiLynx a zakázku na pokladně vydala.

4.3.2 Lenka Mašková

Druhým testovaným uživatelem byla Lenka Mašková, odpovědná vedoucí sedlčanské pobočky. Na začátku uživatelského testování se správně podívala do seznamu zákazníků, zda už zákazník existuje, nicméně pro účely testování jsme ji požádali o založení nového zákazníka. Při vyplňování používala pro skok

na další políčka tabulátor. U pojišťovny začala psát číslo pojišťovny, nikoli její zkratku. Telefonní číslo psala nejprve do políčka pro e-mail, ale hned se opravila. Následně se zeptala na způsob kontaktování a po zjištění odpovědi dopsala vedle telefonního čísla SMS.

Tvorba zakázky probíhala odlišně od Jany Blandové. Vedoucí pobočky nejprve vyplnila parametry čoček, následně zvolila výrobce a typ čočky, k zakázce přidala zábrus do nové obruby a pak zmateně hledala formulář pro vyplnění ceny. Když se to povedlo, přešla k vložení poukazu. Po vyplnění všech náležitostí poukazu a vložení poukázkových kódů byla překvapená stejně jako její kolegyně, když ji formulář přeměroval na seznam všech poukazů.

Zaplacení zálohy, následné naskladnění a prodej na pokladně společně s přidáním pouzdra proběhl bez obtíží.

4.3.3 Vyhodnocení uživatelského testování

Jana Blandová a Lenka Mašková se shodly na tom, že by bylo vhodné reorganizovat položky u zákazníka a do prvního sloupce dát kontaktní údaje namísto trvalé slevy a volby VIP. Telefonní číslo je častější než e-mail, tudíž by bylo dobré ho dát jako první. Zjistilo se, že ve formuláři chybí volba způsobu kontaktování zákazníka.

Míchat parametry čočky společně s její cenou v jedné tabulce se ukázalo jako špatný nápad. Při posloupnosti vyplňování se cena řeší až po zadání výrobce, typu a druhu čočky. Při následné diskusi se přišlo na to, že existují případy, kdy si někdo chce nechat zhotovit pouze jednu čočku, případně že sklo čočky může mít nějakou speciální vlastnost (vývažné či matové sklo).

Vkládání poukázkových kódů z databáze sklidilo úspěch, jen by nemělo zbytečně zobrazovat přehled všech poukazů, když se vyplňuje jen jeden. Dále došlo k upřesnění popisků u tlačítek pro uložení, tisk a vydání.

Přehled zakázek se Janě i Lence líbil, ocenily přehlednost. Jen by v něm chtěly zobrazovat i produkty a služby a někde do rohu by umístily informaci o druhu čoček.

Lenka Mašková přišla s dobrým poznatkem – při naskladňování nových produktů, které nejsou v systému, je potřeba je vytvořit. Na jeden dodací list se naskladňují produkty od jednoho výrobce, který se vyplní ve formuláři. Dobrou funkcí by bylo, kdyby se výrobce (pokud je vyplněný) automaticky předvyplňoval při vytváření nového produktu.

Testování v Sedlčanech bylo velmi přínosné. Obě zaměstnankyně oční optiky byly ochotné a shovívavé k chybám testovací verze. Všechny jejich připomínky jsem zapracoval do nové verze určené k testování v Benešově a Vlašimi.

4.4 Oční optika Benešov

Druhé uživatelské testování proběhlo během pátečního dopoledne 4. května 2018 na pobočce v Benešově. Společně s vedoucím práce Ing. Jirím Hunkou jsme nejprve otestovali majitele oční optiky, který se systémem nepracuje, tudíž to byla ideální příležitost ke zjištění, zda se v něm zorientuje někdo, kdo nikdy nedělal typické procesy oční optiky. Poté se testu podrobily odpovědná vedoucí pobočky a diplomovaná oční technička. S vedoucí pobočky ještě před uživatelským testováním proběhla diskuse ohledně skladů a párování faktur.

Testovací verze informačního systému OptiLynx měla zapracované všechny připomínky z předchozího testování plus pár nových vylepšení jako třeba interaktivnější rozbalovací nabídku, indikátor stavu na všech obrazovkách či hledání cen čoček z ceníků od výrobce Hoya.

4.4.1 Roman Šafanda

Roman Šafanda je majitel všech očních optik společnosti OČNÍ OPTIKA BENEŠOV s.r.o. S informačním systémem OptiLynx nemá velké zkušenosti, protože ho ke své práci potřebuje minimálně. Jeho účast v uživatelském testování byla velmi přínosná, poněvadž šlo o člověka, který nezná detailně proces vyplňování zakázky.

S vytvořením nového zákazníka neměl majitel žádný problém. Zakázku vyplnil téměř bez problémů, nevěděl si rady jen u vyplňování poukazu, což ovšem není problém špatného uživatelského rozhraní informačního systému, nýbrž optické neznalosti při čtení údajů na poukazu (úpravu by si zasloužil spíš samotný papírový poukaz od lékaře). Bez předchozích znalostí vložil do políčka se slevou hodnotu v procentech (50% slevu na obrubu) a vůbec se nepozastavil nad tím, zda políčko pro slevu přijímá hodnotu v procentech. Majitel byl z vyplňování zakázky opravdu nadšený: „*Je to luxusní!*“

Když došlo na placení, Roman Šafanda začal vyplňovat zálohu, což způsobilo, že ji připsal na již předgenerovanou cenu celé zakázky. Díky tomuto postřehu jsme usoudili, že by bylo lepší při otevření dialogu cenu označit tak, aby se při psaní nové ceny přepsala. Nakonec se tento problém vyskytl i u dalšího testování.

Při závěrečném placení došlo v informačním systému k chybě s cenou – na pokladnu se vkládala špatná cena. Majitel totiž vkládal zakázku do pokladny úplně jiným způsobem, který jsme vůbec nepředpokládali, avšak způsob to byl korektní. Nešlo o podstatný problém, nicméně před dalším uživatelským testováním jsem chybu opravil, aby zbytečně nerozptylovala.

4.4.2 Šárka Martinková, DiS.

První z testovaných zaměstnankyň byla Šárka Martinková – diplomovaná oční technička.

Vytvoření nového zákazníka jí nečinilo žádné problémy. Informace vyplňovala na přeskáčku pomocí myši. Tvorba zakázky byla taktéž bezproblémová a oční technička se nezasekla u ničeho. Naskladnění a následný prodej zboží proběhl také hladce.

Šárka Martinková ocenila jako výrazné zlepšení od první verze funkcionality automatického vyplňování cen čoček a práci s poukazem, který umožňuje hledat poukazkové kódy. Na přehledu zakázek jí nechyběla žádná informace a s detailem zakázky byla také spokojená.

4.4.3 Věra Chmúrová

Věra Chmúrová zastává na benešovské pobočce roli odpovědné vedoucí. Od nasazení informačního systému OptiLynx je však nespokojená a dává to patřičně najevo, proto jsme se na toto testování pořádně připravili.

Před uživatelským testováním proběhla diskuse na téma práce se skladem, jelikož s ním Věra Chmúrová jako jediná pracuje. Po vyjevení své frustrace z komplikovaného procesu naskladňování a párování faktur nám vedoucí pobočky nakonec sdělila potřebné informace. Od postoje „*je to celý špatně*“ jsme se dostali ke konstruktivnímu řešení, které vyžaduje jen drobné úpravy u párování faktur. Klíčovou změnou bude zobrazování položek podle identifikátoru zakázky, nikoli podle dodacího listu.

Poté následovalo uživatelské testování. Věra Chmúrová postupovala opravdu důsledně. Nevytvářela nového zákazníka, ale našla si mě v seznamu zákazníků podle rodného čísla. U tvorby zakázky při vyplňování parametrů z poukazu mi nasadila brýle, na kterých mi změřila správné hodnoty pro vyplnění sférických dioptrií (DP). Taktéž ocenila automatické vyplňování cen z ceníku.

Zaplacení zálohy a naskladnění nečinilo žádný problém. Menší zaškobrnutí nastalo jen při vydání zakázky na pokladně. Vedoucí pobočky byla ze starého systému zvyklá na to, že je u objednávky klikatelný celý řádek, v novém systému je klikatelné pouze tlačítko pro vydání zakázky.

Při závěrečné diskusi nám sdělila, že se jí nová verze líbila a vyzdvihla dobrou práci se zakázkou. Nejlepší moment testování, který si nemohu odpustit nezmínit, nastal při placení na pokladně pomocí dialogového okna, ve kterém se volí typ placení a částka. V ten moment Věra Chmúrová s patřičným entusiasmem zahlásila: „*Jé, to je hezký!*“

4.4.4 Vyhodnocení uživatelského testování

Původně obávané testování nakonec dopadlo velmi dobře. Majitel Roman Šafanda se v informačním systému velmi rychle zorientoval a práce v něm mu nečinila žádné problémy. Stejně hladký průběh uživatelského testování následoval u Šárky Martinkové a Věry Chmúrové. Nebyly zjištěny žádné výrazné nedostatky, pouze drobnosti.

Dále byly upřesněny problémy kolem skladu. Zároveň bylo navrženo a schváleno řešení, které bude zapracováno v následující verzi.

4.5 Oční optika Vlašim

Třetí uživatelské testování proběhlo během pátečního odpoledne 4. května 2018 na pobočce ve Vlašimi. Společně s vedoucím práce Ing. Jiřím Hunkou jsme otestovali nejprve oční techničku a potom odpovědného vedoucího vlašimské pobočky. Nutno podotknout, že toto testování bylo, co se výsledků týče, nejrozporupnější.

4.5.1 Jana Exnerová

Nejprve byla uživatelskému testování podrobena diplomovaná oční technička Jana Exnerová. Po přihlášení se z pokladny přesunula na seznam zákazníků a v něm klikla na tlačítko pro vytvoření nového zákazníka. Do políčka určeného k vyplnění ulice vložila ulici společně s číslem popisným, čímž vůbec nevyužila následující políčko určené pro vyplnění čísla.

Jana Exnerová nám při vyplňování zakázky sdělila, že některé náležitosti zakázky nevyplňuje hned se zákazníkem, ale že se k nim vrací až po odchodu zákazníka z pobočky. Ze všech testovaných uživatelů byla první, kdo postupuje tímto způsobem. Také poznamenala, že většinou nevyplňuje typ obruby a zisk dat a že se jí to zdá zbytečné.

Zaplacení zálohy, naskladnění i vydání zakázky proběhlo bez problémů. Při uživatelském testování zaznělo mnoho superlativů, kterými oční technička vůbec nešetřila, z čehož jsme zpočátku vydedukovali, že je s aktuální verzí informačního systému spokojená.

4.5.2 Michal Novák, DiS.

Druhým testovaným uživatelem byl Michal Novák, odpovědný vedoucí vlašimské pobočky. Od nasazení informačního systému OptiLynx patří mezi jeho největší kritiky. Několikrát nás častoval slovy jako „*Tenhle software není konkurenceschopný*“ či „*Máte hotovo tak 20 %!*“. Očekávali jsme, že si na testování připraví konstruktivní kritiku a řekne, co mu nevyhovuje a co bychom měli změnit. Proto jsme se na toto testování velmi těšili a očekávali společnou kreativní diskuzi nad jeho návrhy.

Michal Novák při testování nejprve správně vytvořil zákazníka a pak se vrhl na tvorbu zakázky. Základní náležitosti zakázky vyplnil bez problémů. Při prosbě, zda by mohl přidat do zakázky pouzdro na brýle, to pravděpodobně z důvodu své zbrklosti nedokázal. Při vkládání slevy na čočky ho vůbec nenapadlo zadávat hodnotu v procentech, což mu nakonec napověděla kolegyně. Když to vyzkoušel, byl překvapený, že to funguje. S vyplňováním poukazu problém neměl, dokonce ocenil, že je intuitivní. Problém nastal při na-

skladňování, které údajně nikdy nedělal. Nejprve se snažil vytvořit nový produkt a teprve po upozornění, aby si pořádně přečetl, co je na stránce s naskladněním napsané, vyhledal a naskladnil danou zakázku. Vydání zakázky probíhalo ve stejně zmateném duchu.

Michal Novák při testování postupoval velmi rychle a nepozorně, nic si pořádně nepřečetl a neustále přecházel mezi obrazovkami aplikace. Prý aby správně simuloval stresovou situaci při obsluze zákazníka, což se nám zdálo jako lichý argument. Po pár minutách se z uživatelského testování stala hotová fraška. Michal Novák neustále vtipkoval, takže nebylo jasné, co myslí vážně a co ne. Často si protířčil, jen aby vše zkritizoval. „*Když jsem ve starém systému vytvářel zakázku, mohl jsem se zároveň věnovat zákazníkovi,*“ pronesl u tvorby zakázky, načež vše po pár vteřinách vyvrátil sexistickou poznámkou: „*Víte, že chlapi umí dělat jenom jednu věc... ženský, tady Jana je ještě schopná se bavit s lidma!*“

Dobrý pocit z tohoto testování jsem neměl ani já, ani můj vedoucí práce Ing. Jiří Hunka.

4.5.3 Vyhodnocení uživatelského testování

Závěr z uživatelského testování v oční optice Vlašim se nám dělal poměrně těžce. Jana Exnerová prošla testováním v pořádku a neměla s ničím výrazný problém. Zato Michal Novák byl nespokojený už z principu a nebyl ochotný nijak pomoci. Domníváme se, že jeho neochota a laxní přístup pramení ze zesilující frustrace, která vznikla po nasazení první verze informačního systému OptiLynx, kterou byl nucen používat při svém zaměstnání. Bohužel jsme mu nedokázali správným způsobem vysvětlit, že uživatelské testování smysl má a že je přínosné pro vývoj další verze, se kterou bude v budoucnu pracovat.

Když jsme se oprostili od všech rádooby vtipných poznámek, které padaly z úst Michala Nováka, zarazila nás snad jen připomínka k vyplňování zakázky. Prý mu vyhovuje tvorba zakázky v optickém softwaru Newton, protože na ní vidí všechny náležitosti zakázky na jedné obrazovce a nemusí se v ní pohybovat. Na to mu souhlasně přitakala kolegyně Jana Exnerová, ačkoliv nám po svém uživatelském testování sdělila, že byla s novou tvorbou zakázky spokojená.

4.6 Závěr testování

Počáteční testování informačního systému OptiLynx při reálném provozu ukázalo nejzávažnější chyby, které sloužily jako podklady pro tvorbu nové verze v Material Designu. Díky tomu byla přepracovaná celá zakázka a tvorba poukazu.

Následně byly naplánované dvě etapy uživatelského testování s čtrnáctidenní prodlevou. Při první etapě v oční optice Sedlčany jsem se ujistil v tom, že

nově zvolený směr s Material Designem je správný. Diskuse se zaměstnankyněmi přinesla nové poznatky a odhalila závažné chyby. Vše bylo do další etapy zapracováno a opraveno.

Druhá etapa uživatelského testování sloužila spíš k objevování menších chyb, které se přehlídly. Bylo zapracováno pár nových funkcionalit, které někdo využil (hledání cen z ceníku) a které nikdo nepoužil, ale určitě budou mít smysl v budoucnu (zlepšená práce v systému pouze s klávesnicí).

Uživatelské testování bylo jednoznačně přínosné. Jeho nevýhodou je sice velká časová investice, ale bez něj bych nevěděl, co uživatele trápí a co chtějí změnit.

4.6.1 Finální seznam neopravených věcí

Přehled neopravených věcí, na které se přišlo během testování:

- označovat text ve formuláři tam, kde je to potřeba (při placení, při hledání),
- při editaci zákazníka u zakázky uložit upravená data, když se klikne mimo nebo když se dokončuje zakázka,
- informovat, že při volbě výrobce Hoya je potřeba začít psát typ, aby se mohlo vyhledávat v ceníku,
- zobrazovat u poukazu částku bez zaokrouhlení,
- seznam cen z ceníku Hoya by chtělo načítat asynchronně a informovat uživatele, že musí psát
- po vydání zakázky zneaktivnit všechny formulářové prvky, aby se zakázka nedala zpětně upravovat,
- pokud zakázka neobsahuje žádné čočky, zeptat se při uložení uživatele, zda jde opravdu o zakázku bez čoček.
- upravit naskladňování a párování faktur

Závěr

Cílem této práce bylo navrhnout, zrealizovat a otestovat frontend informačního systému pro středně velkou oční optiku. Během návrhu byly zvoleny vhodné technologie a dvě uživatelská rozhraní – Bootstrap a Material Design. Následně proběhla implementace informačního systému OptiLynx za pomoci webového frameworku Angular.

Poté byla nasazena první verze informačního systému na pobočky očních optik společnosti OČNÍ OPTIKA BENEŠOV s.r.o. Během testovacího provozu byla objevena řada chyb a problémů, které byly vyhodnoceny a následně zapracovány do druhé verze informačního systému OptiLynx. Ve druhé verzi se použil Material Design.

Následovaly dvě etapy uživatelského testování. První etapa uživatelského testování proběhla v polovině dubna na pobočce v Sedlčanech. Po čtrnácti dnech následovala druhá etapa, při které proběhlo testování na pobočce v Benešově a na pobočce ve Vlašimi. Testování v Benešově dopadlo velmi dobře, nicméně výsledky z testování vlašimské pobočky byly poměrně rozpačité.

Výhled do budoucnosti

Vzhledem k tomu, že je možné používat informační systém i na jiných zařízeních (mobilní telefon a tablet) než na počítači, bylo by dobré doladit zobrazování všech obrazovek aplikace a některé lépe uzpůsobit pro menší rozlišení.

Pro lepší udržitelnost kódu a znovupoužitelnost by se hodilo rozdělit některé komponenty na víc menších komponent.

V rámci optimalizace rychlosti informačního systému by bylo vhodně upravit některé API zdroje, aby složité minimalizoval počet operací na frontendu. Dále by se mohlo víc využívat asynchronní načítání dat ze serveru.

Také by stálo za zvážení použití **ngrx/store**, což je stavový kontejner, který dokáže držet stav aplikace v jednotné struktuře, ze které je možné získávat příslušná data [52].

Literatura

- [1] Butler, J. G.: *Television: Visual Storytelling and Screen Culture*. Routledge, 2018.
- [2] ČTK: *Optikům se daří, do karet jim hraje vyšší věk dožití i užívání elektroniky [online]*. [cit. 2018-04-23]. Dostupné z: <http://zpravy.e15.cz/byznys/obchod-a-sluzby/optikum-se-dari-do-karet-jim-hraje-vyssi-vek-doziti-i-uzivani-elektroniky-1281008>
- [3] OČNÍ OPTIKA BENEŠOV s.r.o.: *OČNÍ OPTIKA BENEŠOV s.r.o. [online]*. [cit. 2018-04-23]. Dostupné z: <http://www.optikabenesov.cz/>
- [4] Galitz, W. O.: *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*. Wiley Computer Publishing, druhé vydání, ISBN 9780470146224.
- [5] Nielsen, J.: Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, ACM, 1994, s. 152–158.
- [6] Martina, S.: *Heuristická analýza [online]*. [cit. 2018-04-24]. Dostupné z: <http://www.inflow.cz/heuristicka-analyza>
- [7] Harley, A.: *Personas Make Users Memorable for Product Team Members [online]*. [cit. 2018-04-22]. Dostupné z: <https://www.nngroup.com/articles/persona/>
- [8] Škorníčková, E.: *Co je GDPR a jak bude aplikováno v Česku [online]*. [cit. 2018-04-20]. Dostupné z: <https://www.gdpr.cz/gdpr/co-je-gdpr/>
- [9] Zákon č. 268/2014 Sb., o zdravotnických prostředcích a o změně zákona č. 634/2004 Sb., o správních poplatcích, ve znění pozdějších předpisů. *Sbírka zákonů České republiky*, 2014-11-24: s. 3146–3187, ISSN 1211-1244.

- [10] Ilinčev, O.: *Jak analyzovat konkurenci [online]*. [cit. 2018-05-03]. Dostupné z: <http://www.ilincev.com/analyza-konkurence>
- [11] MAXOFT, s.r.o.: *Newton -- specializovaný software pro oční optiku [online]*. [cit. 2018-04-28]. Dostupné z: <http://www.maxoft.cz/newton-spec-software-pro-ocni-optiku/>
- [12] OpticEvidence: *OpticEvidence – Systém pro komplexní správu optiky [online]*. [cit. 2018-04-28]. Dostupné z: <https://www.opticevidence.cz/>
- [13] TISS Optic CZ s.r.o.: *Tiss Optic – online informační systém pro optiky [online]*. [cit. 2018-04-28]. Dostupné z: <http://www.tissoptic.cz/optic/>
- [14] Řezáč, J.: *Drátěné modely webu [online]*. [cit. 2018-04-28]. Dostupné z: <http://blog.filosof.biz/dratene-modely-webu/>
- [15] Wasson, M.: Single-Page Applications: Build Modern, Responsive Web Apps with ASP.NET. *MSDN magazine*, 2013-11: s. 40–50.
- [16] Fink, G.; Flatow, I.; Group, S.: *Pro Single Page Application Development: Using Backbone.js and ASP.NET*. EBL-Schweitzer, Apress, 2014, ISBN 978-1430266730. Dostupné z: <https://books.google.cz/books?id=ayuLAWAAQBAJ>
- [17] Jahoda, B.: *Single page application [online]*. [cit. 2018-04-20]. Dostupné z: <http://jecas.cz/spa>
- [18] Jain, S.: *Ultimate DEATH Match: SPA Vs. MPA [online]*. [cit. 2018-04-24]. Dostupné z: <https://medium.com/@jainshilpa1993/ultimate-death-match-spa-vs-mpa-82e0b79ae6b6>
- [19] Michálek, M.: *Vzhůru do CSS3*. Michálek Martin – Vzhůru dolů, 2017, ISBN 978-80-260-8440-2.
- [20] Deveria, A.: *Can I use... CSS Flexible Box Layout Module [online]*. [cit. 2018-05-02]. Dostupné z: <https://caniuse.com/#feat=flexbox>
- [21] Michálek, M.: *BEM: Pojmenovací konvence pro třídy v CSS [online]*. [cit. 2018-04-30]. Dostupné z: <https://www.vzhurudolu.cz/prirucka/bem>
- [22] Haverbeke, M.: *Eloquent JavaScript, 3rd Edition: A Modern Introduction to Programming*. No Starch Press, 2018, ISBN 1593279507.
- [23] Bierman, G.; Abadi, M.; Torgersen, M.: Understanding TypeScript. In *European Conference on Object-Oriented Programming*, Springer, 2014, s. 257–281.

-
- [24] Stack Overflow: *Developer Survey 2018 [online]*. [cit. 2018-04-28]. Dostupné z: <https://insights.stackoverflow.com/survey/2018/>
- [25] Google LLC: *Angular – Documentation [online]*. [cit. 2018-04-23]. Dostupné z: <https://angular.io/docs>
- [26] Tobias Koppers, S. T. L., Johannes Ewald: *Webpack – Documentation [online]*. [cit. 2018-04-24]. Dostupné z: <https://webpack.js.org/>
- [27] Fielding, R.; Gettys, J.; Mogul, J.; aj.: Hypertext transfer protocol–HTTP/1.1. Technická zpráva, 1999.
- [28] Fielding, R. T.; Taylor, R. N.: *Architectural styles and the design of network-based software architectures*, ročník 7. University of California, Irvine Doctoral dissertation, 2000.
- [29] Malý, M.: *REST: architektura pro webové API [online]*. [cit. 2018-05-05]. Dostupné z: <https://www.zdrojak.cz/clanky/rest-architektura-pro-webove-api/>
- [30] Fielding, R.; Reschke, J.: Hypertext transfer protocol (HTTP/1.1): Semantics and content. Technická zpráva, 2014.
- [31] Otto, M.; Thornton, J.; contributors, B.: *Bootstrap – Documentation [online]*. [cit. 2018-04-24]. Dostupné z: <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
- [32] Lambert, M.; Jobsen, B.; Cochran, D.; aj.: *Complete Bootstrap: Responsive Web Development with Bootstrap 4: Learn all the new features and build a set of example applications for your portfolio with the latest version of Bootstrap*. Packt Publishing, 2017, ISBN 978-1788833400.
- [33] Brian, M.: *Google’s new ‘Material Design’ UI coming to Android, Chrome OS and the web [online]*. [cit. 2018-05-05]. Dostupné z: <https://www.engadget.com/2014/06/25/googles-new-design-language-is-called-material-design/>
- [34] Google LLC: *Material Design – Guidelines [online]*. [cit. 2018-04-24]. Dostupné z: <https://material.io/guidelines/>
- [35] The Daring Fireball Company LLC.: *Daring Fireball: Markdown [online]*. [cit. 2018-05-07]. Dostupné z: <https://daringfireball.net/projects/markdown/>
- [36] Haták, M.: *Observables jednoduše – co to je a k čemu je to dobré [online]*. [cit. 2018-05-07]. Dostupné z: <https://www.inizio.cz/blog/observables/>

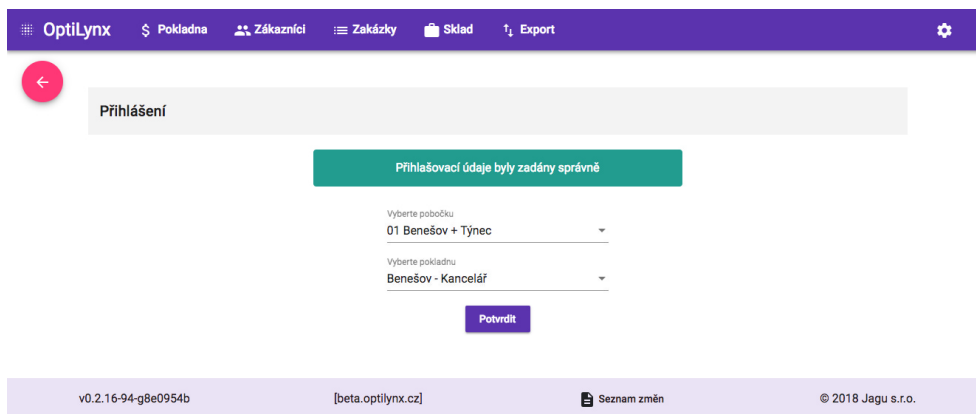
- [37] Hirczy, K.: *Working with models in Angular [online]*. [cit. 2018-05-07]. Dostupné z: <https://nehalist.io/working-with-models-in-angular/>
- [38] Silveira, R.: *Angular 2 Forms – Template Driven and Model Driven Approaches [online]*. [cit. 2018-04-29]. Dostupné z: <https://www.codeproject.com/Tips/1166952/Angular-Forms-Template-driven-and-Model-driven-a>
- [39] Aravind: *What are the practical differences between template-driven and reactive forms? - Stack Overflow [online]*. [cit. 2018-05-08]. Dostupné z: <https://stackoverflow.com/questions/39142616/what-are-the-practical-differences-between-template-driven-and-reactive-forms>
- [40] Google LLC: *Angular: Forms – Documentation [online]*. [cit. 2018-05-08]. Dostupné z: <https://angular.io/guide/forms>
- [41] Google LLC: *Angular: Pipes – Documentation [online]*. [cit. 2018-05-08]. Dostupné z: <https://angular.io/guide/template-syntax>
- [42] Google LLC: *Angular: Pipes – Documentation [online]*. [cit. 2018-05-08]. Dostupné z: <https://angular.io/guide/pipes>
- [43] Krug, S.: *Nenuřte uživatelé přemýřlet!* Brno: Computer Press, druhé vydání, ISBN 80-251-1291-8.
- [44] Google LLC: *Angular Material – Documentation [online]*. [cit. 2018-04-24]. Dostupné z: <https://material.angular.io/>
- [45] Laurens, F.: *Quick start with Angular Material and Flex-Layout [online]*. [cit. 2018-05-08]. Dostupné z: <https://medium.com/letsboot/quick-start-with-angular-material-and-flex-layout-1b065aa1476c>
- [46] Google LLC: *Material Design – Color [online]*. [cit. 2018-05-08]. Dostupné z: <https://material.io/guidelines/style/color.html>
- [47] Google LLC: *Angular Material – Theming your Angular Material app [online]*. [cit. 2018-05-08]. Dostupné z: <https://material.angular.io/guide/theming>
- [48] International Organization for Standardization: *ISO 22324:2015 [online]*. [cit. 2018-05-08]. Dostupné z: <https://www.iso.org/standard/50061.html>
- [49] Raskin, J.: *The Humane Interface: New Directions for Designing Interactive Systems*. Addison-Wesley Professional, 2000, ISBN 0201379376.

- [50] Nielsen, J.: *Usability Engineering*. Academic Press Inc, 1993, ISBN 0-12-518405-0. Dostupné z: <https://www.amazon.com/Usability-Engineering-Jakob-Nielsen/dp/0125184050>
- [51] Card, S. K.; Robertson, G. G.; Mackinlay, J. D.: The information visualizer, an information workspace. In *Proceedings of the SIGCHI Conference on Human factors in computing systems*, ACM, 1991, s. 181–186.
- [52] Navrátil, P.: Aplikace pro vedení čtenářského deníku. 2017.

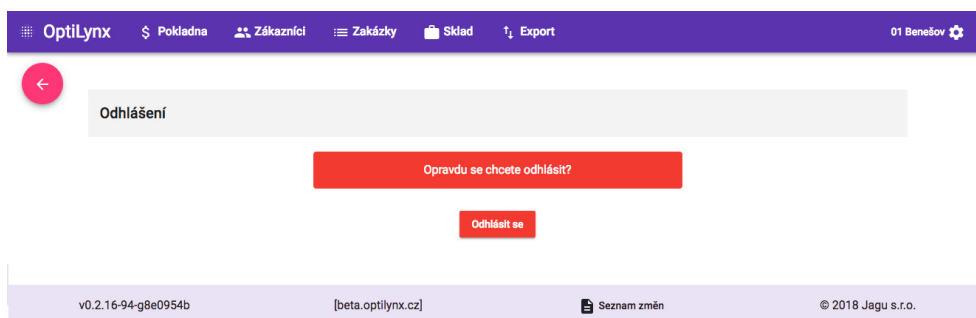
Seznam použitých zkratk

- AJAX** Asynchronous JavaScript and XML (český asynchronní JavaScript a XML)
- CLI** Command Line Interface (český příkazový řádek)
- CSS** Cascading Style Sheets (český kaskádové styly)
- DOM** Document Object Model (český objektový model dokumentu)
- GDPR** General Data Protection Regulation (česky obecné nařízení o ochraně osobních údajů)
- HTML** HyperText Markup Language (česky hypertextový značkovací jazyk)
- HTTP** Hypertext Transfer Protocol
- MPA** Multiple Page Application
- REST** Representational State Transfer
- SPA** Single Page Application
- UCD** User-centered design (česky uživatelsky přívětivý design)
- JSON** JavaScript Object Notation (česky javascriptový objektový zápis)
- UI** User interface (český uživatelské rozhraní)
- URI** Uniform Resource Identifier (český jednotný identifikátor zdroje)
- UX** User experience (český uživatelský prožitek)
- XML** eXtensible Markup Language (český rozšiřitelný značkovací jazyk)

Finální verze informačního systému OptiLynx



Obrázek B.1: OptiLynx – Přihlášení



Obrázek B.2: OptiLynx – Odhlášení

B. FINÁLNÍ VERZE INFORMAČNÍHO SYSTÉMU OPTILYNX

OptiLynx Pokladna Zákazníci Zakázky Sklad Export 01 Benešov

Pokladna + Přidat nového zákazníka

Hledat
Hledá se mezi produkty, zakázkami, službami a zákazníky

Položky v pokladně

Zakázka #1448 - Jaroslav Hrách Platit 4608 Kč a vydat (celková cena zakázky 4608 Kč)

Pravé: Hoya hilux 1,5 hard	2450 Kč
Levé: Hoya hilux 1,5 hard	2450 Kč
Obruba: oa 461 f2	480 Kč
Produkt: mr. gain Shrek 160	1590 Kč
Služba: zábrus,čištění vlastní obruba vrtaná	500 Kč
Ploukaz	-312 Kč
Sieva	-2550 Kč
Celkem zaplatit	0 Kč

Platby

Zatím nebyla provedena žádná platba.

Zbývá doplatit 0 Kč

Storno pokladny Zaplatit

v0.2.16-94-g8e0954b [beta.optilynx.cz] Seznam změn © 2018 Jagu s.r.o.

Obrázek B.3: OptiLynx – Pokladna

OptiLynx Pokladna Zákazníci Zakázky Sklad Export 01 Benešov

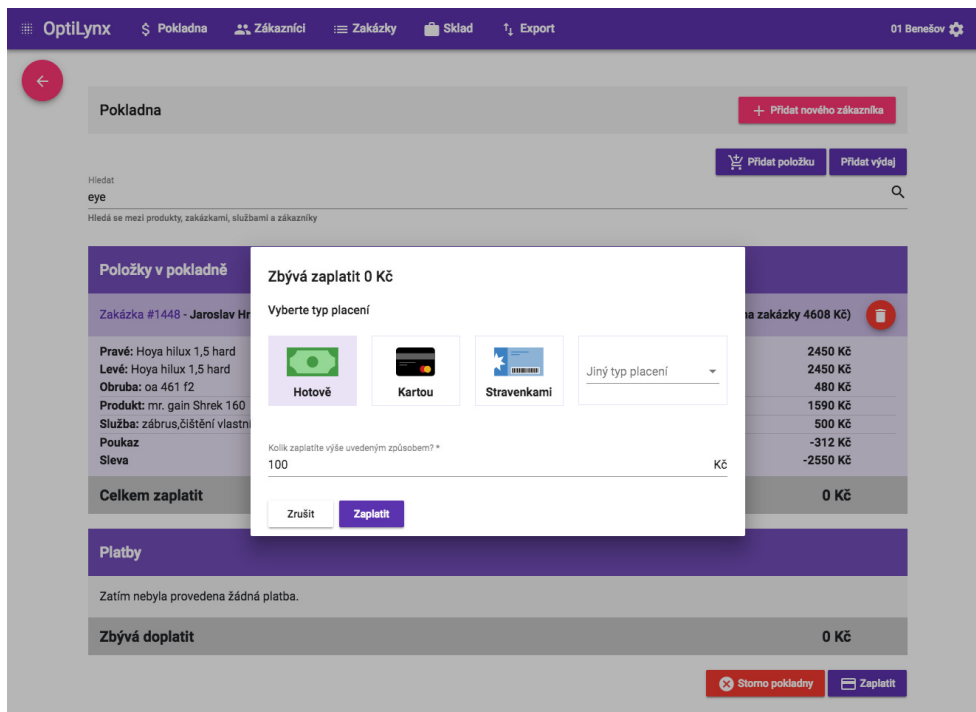
Pokladna + Přidat nového zákazníka

Hledat
eyej

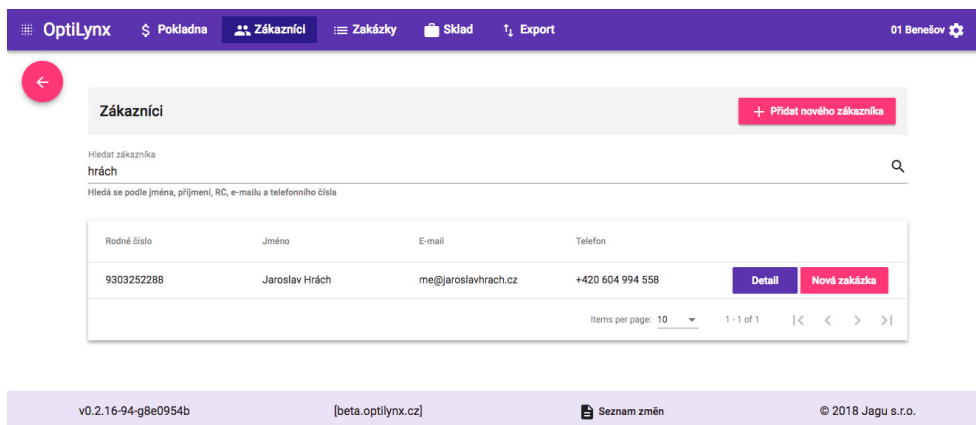
Produkty + Vytvořit novou položku

Eye 2000 850	Skladem: 3 ks	850 Kč	Detail
Eye 2000 zen typ 112	Skladem: 1 ks	5800 Kč	Detail
Eye 2000 zen typ 110	Skladem: 1 ks	5800 Kč	Detail
Eye 2000 minima m502/a206	Skladem: 1 ks	9490 Kč	Detail
mr. gain Eyefunc 509	Skladem: 1 ks	2990 Kč	Detail
eye dance 021/3	Skladem: 1 ks	750 Kč	Detail
alpina Cheyenne 1180	Skladem: 1 ks	1180 Kč	Detail
eye dance e051/3	Skladem: 1 ks	890 Kč	Detail
obruba	Skladem: 0 ks	750 Kč	Detail

Obrázek B.4: OptiLynx – Pokladna – rozbalovací nabídka



Obrázek B.5: OptiLynx – Pokladna – platební karta



Obrázek B.6: OptiLynx – Zákazníci

B. FINÁLNÍ VERZE INFORMAČNÍHO SYSTÉMU OPTILYNX

Detail zákazníka + Vytvořit novou zakázku

Jaroslav Hrách

Údaje o zákazníkovi	Adresa zákazníka	Ostatní informace
Rodné číslo: 9303252288	Pešlova 361/22	Trvalá sleva: 0%
Pojišťovna: VoZP (201)	Praha 19000	VIP: Ne
Telefonní číslo: +420 604 994 558	Česká republika	Kontaktovat o stavu zakázky: SMS
E-mail: me@jaroslavhrach.cz		

Zákazníkovy zakázky

#1448 Jaroslav Hrách	Stav	Zadáno	Cena	Doplatek	Kontaktovat	Detail	Výdej
	Nová	08. 05. 2018	4608 Kč	4608 Kč	SMS		

DD	Sféra	Cylindr	Osa	Adice	Prisma	Osa Prisma	PD	Výška	Cena
Pravé: Hoya hilux 1,5 hard	-3.00	-	-	-	-	-	30	-	2450 Kč
Levé: Hoya hilux 1,5 hard	-2.75	-	-	-	-	-	29	-	2450 Kč
Obruba: oa 461 f2									480 Kč

Produkty	Cena	Služby	Cena
mr. gain Shrek 160	1590 Kč	zábrus,čištění vlastní obru...	500 Kč

v0.2.16-94-g8e0954b [beta.optilynx.cz] [Seznam změn](#) © 2018 Jagu s.r.o.

Obrázek B.7: OptiLynx – Detail zákazníka

Zakázky + Vytvořit novou zakázku

Hledat zakázku

hrách

Hledá se podle čísla zakázky a jména zákazníka

#1448 Jaroslav Hrách	Stav	Zadáno	Cena	Doplatek	Kontaktovat	Detail	Výdej
	Nová	08. 05. 2018	4608 Kč	4608 Kč	SMS		

DD	Sféra	Cylindr	Osa	Adice	Prisma	Osa Prisma	PD	Výška	Cena
Pravé: Hoya hilux 1,5 hard	-3.00	-	-	-	-	-	30	-	2450 Kč
Levé: Hoya hilux 1,5 hard	-2.75	-	-	-	-	-	29	-	2450 Kč
Obruba: oa 461 f2									480 Kč

Produkty	Cena	Služby	Cena
mr. gain Shrek 160	1590 Kč	zábrus,čištění vlastní obru...	500 Kč

v0.2.16-94-g8e0954b [beta.optilynx.cz] [Seznam změn](#) © 2018 Jagu s.r.o.

Obrázek B.8: OptiLynx – Zakázky



Detail zakázky #1448

Jaroslav Hrách

Údaje o zákazníkovi
 Rodné číslo: 9303252288
 Pojišťovna: VoZP (201)
 Telefonní číslo: +420 604 994 558
 E-mail: me@jaroslavhrach.cz

Adresa zákazníka
 Pešlova 361/22
 Praha 19000
 Česká republika

Ostatní informace
 Trvalá sleva: 0%
 VIP: Ne
 Kontaktovat o stavu zakázky: SMS

Čočky Smazat čočky ze zakázky

DD	Sféra	Cylindr	Osa	Adice	Prisma	Osa Prisma	PD	Výška	
Pravé	-3						30		Upřesnění
Levé	-2,75						29		Upřesnění
Obruba	oa 461 f2								Odebrat obrubu

Vyroba: Hoya Typ: hilux 1,5 hard Průměr: Druh: Do dálky

Vložit opravu Vytvořit novou opravu

Cena			
Pravé čočka: Hoya hilux 1,5 hard	Cena: 2450 Kč	1225	Po slevě: 1225 Kč
Levé čočka: Hoya hilux 1,5 hard	Cena: 2450 Kč	1225	Po slevě: 1225 Kč
Obruba: oa 461 f2	Cena: 480 Kč	100	Po slevě: 380 Kč
Celková cena za čočky a obrubu:			2830 Kč

[+ Přidat další čočky](#)

Produkty a služby

Hledat 🔍
 Hledejte se mezi produkty a službami

Produkty			
mr. gain Shrek 160	1590 Kč	0	Po slevě: 1590 Kč 🗑️
Služby			
zábrus,čistění vlastní obruba vrtaná	500 Kč	0	Po slevě: 500 Kč 🗑️

Obrázek B.9: OptiLynx – Detail zakázky

B. FINÁLNÍ VERZE INFORMAČNÍHO SYSTÉMU OPTILYNX

The screenshot shows the 'Sklad' (Warehouse) section of the OptiLynx application. The top navigation bar includes 'OptiLynx', 'Pokladna', 'Zákazníci', 'Zakázky', 'Sklad', and 'Export'. The user is logged in as '01 Benešův'. The main content area is titled 'Skladové položky' and features a search bar and a '+ Přidat nový produkt' button. Below the search bar is a table listing various products with columns for 'Název', 'Typ', 'Skladem', 'Skladem ext.', and 'Komise'. Each row includes 'Detail' and 'Přidat' buttons. The table contains 10 items, with pagination showing '1 - 10 of 2789' items.

Název	Typ	Skladem	Skladem ext.	Komise	Detail	Přidat
gafae aghata ruiz al63115575	obruby	1	0	---	Detail	Přidat
Eye 2000 850	hotový výrobek	3	0	---	Detail	Přidat
optiplast Of 536	obruby	1	0	---	Detail	Přidat
unioptik 2-23086	obruby	0	1	ANO	Detail	Přidat
unioptik 2-7149	obruby	1	0	ANO	Detail	Přidat
unioptik 1-1131	obruby	1	0	ANO	Detail	Přidat
unioptik 1-1135	obruby	1	0	ANO	Detail	Přidat
mr. gain Shrek 160	obruby	1	1	---	Detail	Přidat
icode ic-6171/1	obruby	0	1	---	Detail	Přidat
lks L160/3	obruby	1	0	---	Detail	Přidat

Obrázek B.10: OptiLynx – Sklad

The screenshot shows the 'Detail položky' (Product Detail) page for 'GAIN SHREK 156 BURGUNDY'. The top navigation bar is identical to the previous screenshot. The main content area is titled 'Detail položky "GAIN SHREK 156 BURGUNDY"' and includes a '+ Zobrazit pohyb zboží' button. The product details are displayed in a structured layout:

- Product Name:** GAIN SHREK 156 BURGUNDY (ID produktu: 1401)
- EAN:** R1401
- Popis:**
 - Typ:** OBRUBY
 - Výrobce:** GAIN
 - Model:** SHREK 156 BURGUNDY
 - Barva:** FIALOVÁ
 - Zkratka:**
- Prodejní cena:** 1390 Kč
- DPH:** 15 %
- Ceník nákup:** 390 Kč
- Maloobchodní cena:** 0 Kč
- Na skladě:** 0
- Na skladě ext.:** 1
- Poptávka:**
- Komise:** Ne

A 'Naskladnit tento výrobek' button is located at the bottom of the product details section.

Obrázek B.11: OptiLynx – Detail položky

OptiLynx Pokladna Zákazníci Zakázky Sklad Export 01 Benešov

Skladové položky Přesun mezi sklady Naskladnit Párování faktur

Přesun mezi sklady Historie přesunů

Cílový sklad *
02 Vlašim + Votice

Hledat položku
Hledá se podle názvu, zkratky, modelu, typu a výrobce

Položky k přesunu			
mr. gain čibr 30ml	EAN: R332	Kusů: 1 (max. 45)	Detail
feba P 18	EAN: R435	Kusů: 1 (max. 65)	Detail
feba šňůrka tenká 15	EAN: R380	Kusů: 1 (max. 364)	Detail

Vyskladnit

v0.2.16-94-g8e0954b [beta.optilynx.cz] Seznam změn © 2018 Jagu s.r.o.

Obrázek B.12: OptiLynx – Sklad – Přesun

OptiLynx Pokladna Zákazníci Zakázky Sklad Export 01 Benešov

Skladové položky Přesun mezi sklady Naskladnit Párování faktur

Naskladnění Historie naskladnění

Číslo dodacího listu: 122
Dodavatel: Hoya
Datum vystavení: 4. 5. 2018
Datum přijetí: 4. 5. 2018

Hledat položku
Hledá se podle názvu, zkratky, modelu, typu, výrobce a čísla zakázky

Položky k naskladnění			
Hoya hilux 1,5 hard		Kusů: 1 ks	Detail
Hoya hilux 1,5 hard		Kusů: 1 ks	Detail
GAIN SHREK 156 BURGUNDY	EAN: R1401	Kusů: 1 ks	Detail

Dokončit naskladnění | Dokončit naskladnění a přidat fakturu

v0.2.16-94-g8e0954b [beta.optilynx.cz] Seznam změn © 2018 Jagu s.r.o.

Obrázek B.13: OptiLynx – Sklad – Naskladnění

B. FINÁLNÍ VERZE INFORMAČNÍHO SYSTÉMU OPTILYNX

OptiLynx Pokladna Zákazníci Zakázky Sklad Export 01 Benešov

Skladové položky Přesun mezi sklady Naskladnit Párování faktur

Párování faktur Seznam faktur

Číslo faktury * Platební metoda * Datum vystavení * Datum splatnosti * Datum zdanitelného plnění *

Hotovost 4. 5. 2018 18. 5. 2018 18. 5. 2018

Hledat položku

Hleď se podle názvu a EANu produktu, dále podle čísla dodacího listu

Položky s dodacím listem

Dodací list č. 180100040

str bg6216 t01 levá	EAN: R2026	Kusů: 1 z 1	Cena za kus: 240	240 Kč
---------------------	------------	-------------	------------------	--------

Dodací list č. 1740218

fresco f 827/2	EAN: R2045	Kusů: 1 z 1	Cena za kus: 295	295 Kč
fresco f 681	EAN: R2043	Kusů: 1 z 1	Cena za kus: 295	295 Kč

Položky bez dodacího listu

ENZO p 827/2	EAN: R1496	Kusů: 1 z 1	Cena za kus: 600	600 Kč
--------------	------------	-------------	------------------	--------

Celková cena faktury: 1430 Kč

Přidat položku Spárovat fakturu

v0.2.16-94-g8e0954b [beta.optilynx.cz] Seznam změn © 2018 Jagu s.r.o.

Obrázek B.14: OptiLynx – Sklad – Párování faktur

OptiLynx
\$ Pokladna
👤 Zákazníci
☰ Zakázky
📦 Sklad
📄 Export
01 Benešov ⚙️

←
Skladové položky
Přesun mezi sklady
Naskladnit
Párování faktur

Detail faktury
✎ Upravit

Faktura č. vř-1802620

Platební metoda: Převedem

Datum vystavení: 24. 04. 2018

Datum splatnosti: 11. 05. 2018

Datum zdanitelného plnění: 24. 04. 2018

Produkty na faktuře

Jméno	EAN	Cena	Kusů	Celkem
OA 411 F10	R3495	110 Kč	1	110 Kč
OA 411 F11	R3496	110 Kč	1	110 Kč
OA 414 f11	R3497	110 Kč	1	110 Kč
oa 459 f1	R3498	110 Kč	1	110 Kč
oa 461 f2	R3499	110 Kč	1	110 Kč
oa 462 f3	R3500	110 Kč	1	110 Kč
str flex kov	R3501	91 Kč	1	91 Kč

Další položky

Jméno	Cena	DPH
rež.mat	100 Kč	0.21

Souhrn faktury

Celkový počet produktů: 7

Celková cena: **851 Kč**

v0.2.16-94-g8e0954b
[beta.optilynx.cz]
📄 Seznam změn
© 2018 Jagu s.r.o.

Obrázek B.15: OptiLynx – Sklad – Detail faktury

Obsah přiloženého média

readme.txt	stručný popis obsahu média
recordings	záznamy z uživatelského testování
screens	obrazovky z informačního systému OptiLynx
├─ bootstrap	OptiLynx – Bootstrap
├─ material-design	OptiLynx – Material Design
src		
├─ thesis	zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
text	text práce
├─ DP_Hrach_Jaroslav_2018.pdf	text práce ve formátu PDF