

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Objednávkový a rezervační systém



2021

Vedoucí práce: Mgr. Jiří Zecpal,
Ph.D.

Radek Pilař

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Radek Pilař
Název práce: Objednávkový a rezervační systém
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2021
Studijní obor: Aplikovaná informatika, prezenční forma
Vedoucí práce: Mgr. Jiří Zaccpal, Ph.D.
Počet stran: 46
Přílohy: 1 CD/DVD
Jazyk práce: český

Bibliographic info

Author: Radek Pilař
Title: Order and booking system
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2021
Study field: Applied Computer Science, full-time form
Supervisor: Mgr. Jiří Zaccpal, Ph.D.
Page count: 46
Supplements: 1 CD/DVD
Thesis language: Czech

Anotace

Práce pojednává o naprogramované webové aplikaci, která bude sloužit jako objednávkový a rezervační systém (pro klienty firmy společnosti ELITEC software, s. r. o.). Vytvořená aplikace bude umožňovat objednávku, správu a řízení služeb-/produktů firmy. V systému může klient sledovat informaci o platbách či historii svého nákupu. Aplikace bude naprogramována v jazycích .NET a Angular8+.

Synopsis

The thesis deals with a programmed web application that will serve as an ordering and reservation system (for clients of the company ELITEC software, p. r. o.). The created application will enable ordering, administration and management services/products of the company. In the system, the client can monitor information about payments or the history of their purchase. The application will be programmed in .NET and Angular8+.

Klíčová slova: C#; .NET; Angular; Typescript

Keywords: C#; .NET; Angular; Typescript

Děkuji Mgr. Jiřímu Zaccalovi, Ph.D. za rady a připomínky při vedení této práce a firmě Elitec Software s. r. o., která mi umožnila zpracovat praktický projekt, jež bude mít reálně využití v praxi

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Úvod	9
1.1	Motivace k vytvoření systému	9
1.2	Cíle práce	9
1.3	Požadavky	10
1.4	Průzkum trhu	10
2	Použité technologie	11
2.1	Back-end	11
2.1.1	C#	11
2.1.1.1	.NET	11
2.1.1.2	LINQ	11
2.1.2	NoSQL	11
2.1.3	MongoDB	12
2.1.3.1	Atlas	13
2.2	Middleware	13
2.2.1	HTTPS	13
2.2.2	REST API	13
2.2.3	Postman	13
2.2.4	JWT Bearer Token	13
2.2.5	Cookie	13
2.3	Front-end	14
2.3.1	Typescript	14
2.3.2	Angular	14
2.3.3	HTML a SASS	14
2.3.4	MDBBootstrap	15
2.3.5	NG-ZORRO	15
2.4	Automatizace vývoje	15
2.4.1	GIT	15
2.4.2	DevOps	15
2.4.2.1	Automatický build	15
2.4.2.2	Automatický deploy	16
2.4.3	Azure	16
3	Architektura Serveru	17
3.1	Core	18
3.1.1	Model	18
3.1.2	Constant	18
3.1.3	Extension	18
3.2	DAL - Data Access Layer	19
3.2.1	Repository pattern	19
3.2.2	Unit Of Work	19
3.3	BL - Business Logic	20
3.3.1	Servisní architektura	20

3.3.1.1	Request	20
3.3.1.2	Response	20
3.3.1.3	Příklad použití	21
3.4	Web	22
3.4.1	Controllers	23
3.4.2	Attributes	23
3.5	Libraries	23
4	Klient	24
4.1	Presentation	25
4.1.1	Components	25
4.1.2	Pages	25
4.1.3	Theme	25
4.2	Internal	26
4.3	Services	26
4.3.1	API	26
4.3.2	Storage	26
4.3.3	Guards	26
4.4	Models	27
4.5	Utils	27
4.6	Assets a Styles	27
5	Uživatelský Web	28
5.1	Homepage	28
5.2	Přihlášení a Registrace	29
5.3	Produkty	29
5.4	Detail Produktu	30
5.4.1	Produkt a služba	30
5.5	Košík	31
5.6	Objednávka	32
5.6.1	Košík	32
5.6.2	Doprava	32
5.6.3	Platba	33
5.6.4	Osobní údaje	34
5.6.4.1	Nový vs registrovaný uživatel	34
5.6.4.2	Firma	34
5.7	Detail objednávky	35
5.7.1	Stavy	35
5.7.2	Mailing	35
5.8	Seznam objednávek	36
5.9	Přiřazené služby	36

6	Uživatelská Administrace	37
6.1	Přihlášení	37
6.2	Dashboard	37
6.3	Uživatelé	38
6.4	Kategorie	38
6.5	Doprava a Platba	38
6.6	Produkty	39
6.6.1	Přidání	39
6.6.2	Editace	40
6.6.3	Odstranění	40
6.7	Objednávky	41
6.8	Nastavení	41
7	Plánovaná rozšíření	42
	Závěr	43
	Conclusions	44
8	Přístup k testování	45
9	Obsah přiloženého CD/DVD	45
	Literatura	46

Seznam obrázků

1	Seznam dokumentů	12
2	Dokument produktu	12
3	Microsoft DevOps Release detail	16
4	Architektura serveru	17
5	Princip zpracování požadavku	22
6	Struktura Angular aplikace	24
7	Struktura komponenty	25
8	Vzhled stránky Homepage	28
9	Vzhled stránky seznamu produktů	29
10	Vzhled stránky detail produktu	30
11	Vzhled stránky košík	31
12	Vzhled stránky košík	32
13	Vzhled stránky doprava	33
14	Vzhled stránky platba	33
15	Vzhled stránky osobní údaje	34
16	Vzhled stránky detail objednávky	35
17	Vzhled stránky seznamu objednávek	36
18	Vzhled stránky seznamu objednávek	36
19	Vzhled administrace dashboard	37
20	Vzhled administrace seznamu uživatelů	38
21	Vzhled administrace seznamu produktů	39
22	Vzhled administrace editace produktu	40
23	Vzhled administrace odstranění produktů	40
24	Vzhled administrace seznamu objednávek	41

Seznam zdrojových kódů

1	Implementace boolean metody IsEmpty	18
2	Použití extension metody IsEmpty	18
3	Metoda Search v Product Repository	19
4	Metoda Update ve třídě CategoryService	21
5	Endpoint Search v OrderController	23
6	Metoda pro přidání kategorie	26
7	Část modelu objednávky	27

1 Úvod

Každá společnost potřebuje pro svůj růst a budování značky kvalitní a reprezentativní web. První kontakt klienta s firmou bývá zprostředkován nejčastěji pomocí firemních webových stránek. Uživatel si již z prvního načtení webové stránky udělá vnitřní obrázek o tom, jestli se mu prezentace dané společnosti líbí a zda využije jejich služeb.

1.1 Motivace k vytvoření systému

Motivací k vytvoření systému bylo především nahradit starý prezenční web firmy. Zpočátku se mělo jednat o jednoduchý prezenční web, avšak po narůstajících požadavcích na systém jsem se rozhodl pro vytvoření komplexnějšího objednávkového a rezervačního systému v rámci mé bakalářské práce. Líbilo se mi spojení bakalářské práce a systému, jež bude mít reálné využití v praxi.

1.2 Cíle práce

Cílem práce je vytvořit objednávkový a rezervační systém, který nahradí již starý dosluhující web. Nový systém bude zpracován jako webová aplikace a oproti starému webu bude přidávat mnoho nových funkcionalit. Vytvořená aplikace bude umožňovat objednávku, správu a řízení služeb/produktů firmy. V systému bude taktéž možné sledovat informace o platbách či historii nákupu. Systém bude využíván především klienty, administrátory a agenty firmy. Agenti zde mohou vystupovat jako osoby vyřizující zakoupené služby klienta. Systém bude zahrnovat rovněž statistiky prodeje či obratu společnosti.

Důležitým cílem je navrhnout kvalitní architekturu projektu tak, aby bylo možné později systém bez větších problémů rozšiřovat a navazovat na něj další funkcionalitu. V tomto ohledu firma kladla velký důraz na rozvržení kódu a jeho kvalitu. Dalším důležitým cílem je vyvinout zabezpečené API, které bude přístupné i pro další části systému, např. plánovaná mobilní aplikace.

1.3 Požadavky

Na práci jsou kladeny následující požadavky:

1. Vytvoření webové aplikace sloužící pro objednávku a rezervaci produktů/služeb firmy.
2. Navrhnout kvalitní architekturu systému, která bude zajišťovat udržitelnost a přehlednost kódu.
3. Možnost systém dále rozvíjet po funkční stránce a navazovat na další systémy.
4. Rychlé načítání dat pomocí návrhu REST API pro přenos dat mezi klientem a serverem.
5. Kvalitní zabezpečení serverová API s použitím JWT Bearer Tokenu.
6. Pro každého uživatele by měla být aplikace přehledná a intuitivní.
7. Klientská část musí fungovat rychle a svižně, bude tedy zpracována jako SPA – Single Page Application.

1.4 Průzkum trhu

Po částečném průzkumu trhu podobných aplikací jsem se rozhodl tuto kapitolu více nerozvádět a pouze vypsát důležitá fakta, na která jsem během průzkumu narazil.

Jako podobné aplikace je možné uvést veškeré veřejné systémy umožňující objednávku produktů či služeb. Hlavními příklady mohou být systémy jako [Alza.cz](https://www.alza.cz), [Datart.cz](https://www.datart.cz), či [Mall.cz](https://www.mall.cz). Všechny tyto společnosti dosahují velkých rozměrů a cílí na různé skupiny zákazníků. Náš systém je převážně zaměřen na klienty typu menších technických společností, které ke svému produktu často potřebují také servis či zaškolení. Systém nebude zahrnovat takové množství produktů, proto není potřeba klást velký důraz na přehnané řazení či filtraci zboží.

2 Použité technologie

Při výběru technologií byl kladen důraz na volbu moderních jazyků, jejich udržitelnost, reálně využití a údržbu ve firmě. Jelikož se společnost, pro kterou byl systém vyvinut, zabývá dlouhodobě vývojem softwaru v jazyku C#, jednalo se o jasnou volbu pro vývoj serverové části aplikace na platformě .NET. Pro vývoj klientské části SPA aplikace byl zvolen jazyk Typescript a framework Angular. Hlavním uložištěm dat byla zvolena NoSQL databáze MongoDB hostovaná v cloudu Atlas.

2.1 Back-end

2.1.1 C#

C# je open-source objektově orientovaný jazyk vyvinutý společností Microsoft. Jedná se o typově bezpečný jazyk využívaný pro vývoj desktopových, webových i mobilních aplikací. K vývoji webových aplikací slouží rozhraní .NET [1].

2.1.1.1 .NET

Z velké škály komponent .NET jsem se rozhodl zvolit rozhraní .NET Framework. Jedná se o základní komponentu sloužící pro vývoj webových aplikací nahrazující ASP.NET framework. Dalším nástupcem pro vývoj webů v C# je kompletně přepsané rozhraní .NET Core, které není závislé na konkrétní platformě na rozdíl od .NET Framework. Toto rozhraní jsem se po konzultaci s firmou rozhodl nepoužít, jelikož knihovny, které v systému budou v dalším vývoji použity, nejsou v této verzi momentálně podporovány.

2.1.1.2 LINQ

Linq je integrovaný jazyk v .NET Frameworku sloužící pro dotazování dat s jednotnou syntaxí, u kterého nezáleží na zdroji dat. Tímto způsobem je možné jednoduše přistupovat k datům pomocí stejné syntaxe, což vede k větší přehlednosti kódu [2].

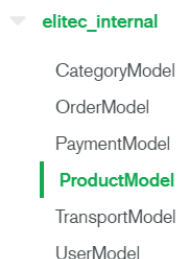
2.1.2 NoSQL

Schéma NoSQL je nejčastěji používaným způsobem v dokumentových databázích, kde je kladen důraz na jednoduchost a vysokou optimalizaci i na velkých objemech dat.

Jednotlivé dokumenty databáze jsou definovány jako třídy v kódu. Jsou mapovány na strukturu dokumentu pomocí systému `key-value`, což umožňuje velmi rychlý vývoj a zároveň vysokou flexibilitu vůči příchozím změnám.

2.1.3 MongoDB

MongoDB je konkrétní implementace NoSQL dokumentové databáze. Jednotlivé dokumenty jsou podobné formátu typu JSON. Dokument je možné přirovnat v relačních databázích k pojmu tabulka. V dokumentech je možné vkládat jednotlivé tabulky do vnořených struktur. Klademe důraz na správné rozdělení kódu do samostatných tříd tak, aby nevznikaly v kódu duplicitní třídy, jež vykonávají stejnou činnost. Na obrázku 1 je možné vidět jednoduchost struktury databáze.



Obrázek 1: Seznam dokumentů

Na dalším obrázku 2 je možné vidět strukturu dokumentu - `Produkt model`, jež obsahuje několik zanořených struktur.



Obrázek 2: Dokument produktu

Řada programovacích jazyků poskytuje rozhraní pro dotazování nad touto databází. Konkrétně v jazyku C# je možné využít rozhraní `MongoDB Driver` poskytující funkcionalitu pro správu databáze [3].

2.1.3.1 Atlas

Atlas je cloudová služba od společnosti MongoDB, Inc. Umožňuje online správu databáze a řeší problémy s uložením a výpadky serverů. K databázi je přístupováno pomocí vygenerovaného tokenu [4].

2.2 Middleware

2.2.1 HTTPS

HyperText Transfer Protokol Secured slouží pro zabezpečenou komunikaci a přenos dat mezi klientem a serverem. Komunikace probíhá pomocí `Requests` požadavků a `Response` odpovědí. Nejčastější architekturou pro komunikaci pomocí HTTP je REST API a zabezpečení pomocí protokolu SSL či TLS [5].

2.2.2 REST API

Representational state transfer je architektura umožňující číst či upravovat data na serveru. Největší výhodou je jednotný přístup k datům, který umožňuje definovat jeden server pro více prostředí. Tímto způsobem je využita stejná funkcionality na serveru pro další navazující systémy [6]

2.2.3 Postman

Postman je nástroj sloužící pro testování Web API. Umožňuje odesílat `Request` na server bez použití klientské části. Nástroj umožňuje rovněž vygenerovat kompletní dokumentaci k REST API.

2.2.4 JWT Bearer Token

JWT token je využíván pro zabezpečení přístupu k datům na serveru. Uživateli je při ověření na serveru vygenerován token, který je odeslán zpět ke klientovi a uložen do `Cookie`. Následně je pomocí tohoto tokenu uživatel identifikován na serveru. Ověření probíhá rozšifrováním tokenu, ze kterého jsou získány základní informace. Nejčastěji se jedná o unikátní identifikátor uživatele či expiraci tokenu. Token není potřeba ukládat do žádného úložiště. Pracujeme pouze s posloupností znaků, kterou obdržíme od klienta.

2.2.5 Cookie

`Cookie` je využívána pro ukládání malých dat na straně uživatele. Hlavní výhodou této paměti je, že informace zůstane u klienta uložena i po obnovení či opuštění stránky.

2.3 Front-end

2.3.1 Typescript

Typescript je open-source rozšíření jazyka Javascript od firmy Microsoft. Přidává některé prvky z objektově orientovaných jazyků jako jsou třídy, rozhraní či statické typování proměnných. Tímto způsobem umožňuje zpřehlednit kód a vyvarovat se zbytečným chybám. [7].

2.3.2 Angular

Angular je framework, který umožňuje psát v jazyce Typescript webové, mobilní či desktopové aplikace. Pro vývoj webových aplikací je použit klasický Typescript. K vývoji multiplatformních mobilních aplikací se používá nadstavba Cordova či Native script umožňující kompilovat aplikaci pro Android i iOS. K vývoji desktopových aplikací je možné použít nadstavbu Electron.

Angular představuje nový směr umožňující pomocí jednoho jazyku psát většinu systémů. Jedná se o velkou výhodu, jež dokáže programátorovi ušetřit při vývoji spoustu času [8].

Kód psaný v jazyce Typescript a frameworku Angular je překládán pomocí kompilátoru do jazyka Javascript. Nejčastěji používanými způsoby překladu jsou metody AOT a JIT.

Metoda JIT - *Just In Time* je způsob překladu, kde překladač vyhodnocuje kód při běhu aplikace. Výhodou tohoto přístupu je, že překladač vyhodnocuje pouze část kódu, která bude v daném momentě použita, což razantně urychluje překlad kódu. Tento způsob je využíván především při vývoji a testování systému, kde není potřeba čekat při každé změně na překlad celé aplikace.

Naopak metoda AOT - *Ahead Of Time* nejprve přeloží celý kód do Javascriptu, a poté jej spustí. Tento přístup je vhodný při nasazení aplikace na produkci. [9].

Angular podporuje oba výše zmíněné způsoby. Při vývoji používá metodu JIT, při nasazení na produkci automaticky přepne do režimu AOT a spustí již předkompilovaný Javascript.

2.3.3 HTML a SASS

Značkový jazyk HTML je nedílnou součástí každé webové stránky. Při vývoji je kladen důraz na validní kód, který má vliv na SEO.

SASS je CSS preprocessor usnadňující práci se stylováním a celkovým vzhledem aplikace. Oproti CSS umožňuje definovat vnořené struktury, podmínky či cykly.

2.3.4 MDBBootstrap

Material design for bootstrap je knihovna kombinující vzhled Material designu a knihovny Bootstrap. Knihovna je využita především v klietkové části webu, kde usnadňuje práci s definováním vzhledu uživatelského obsahu [10].

2.3.5 NG-ZORRO

NgZorro poskytuje komponenty pro práci s formuláři, tabulky, texty, tlačítka a dalšími součástmi pro přípravu webu. Knihovna je použita především v administraci. Stejně komponenty jsou dostupné rovněž pro vývoj ve frameworku React, které je možné nalézt pod názvem Ant Design [11].

2.4 Automatizace vývoje

Automatizace vývoje je nedílnou součástí programování složitějších systémů, na kterých pracuje více programátorů zároveň. K tomuto procesu je využívána implementace systému GIT. K automatizaci patří rovněž automatická správa procesů jako jsou build, deploy či testování aplikace.

2.4.1 GIT

GIT je distribuovaný systém určený ke správě verzí kódu. K hostingu vývojářských souborů je použit GitHub Cloud. Jedná se o online službu umožňující uchovávat potřebné soubory v cloudu. Jednou z možností připojení ke cloudu může být pomocí systému GitHub Desktop.

2.4.2 DevOps

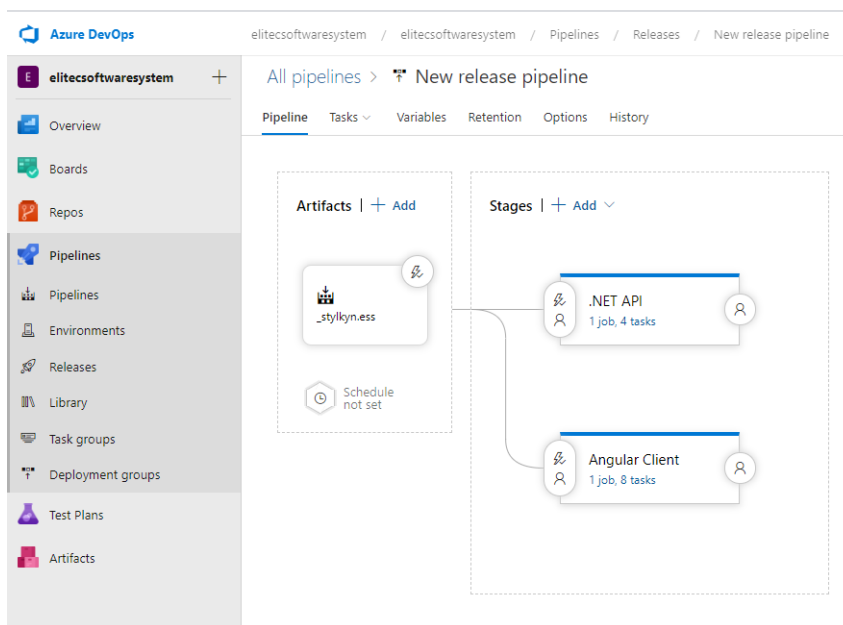
Spojením slov Development a Operations vznikla zkratka DevOps, jež slouží ke sjednocení vývoje softwaru a jeho správy. Jedná se dnes o velmi častý způsob programování využitý při agilním vývoji, ve kterém je potřeba průběžně nasazovat jednotlivé části aplikace. Výhodou je velmi rychlá zpětná vazba od klienta, kterou je možné využít při návrhu další části aplikace. V systému je využita konkrétní implementaci DevOps od společnosti Microsoft.

2.4.2.1 Automatický build

Automatizace kompilace produkční verze aplikace je důležitým krokem k indikaci chyb. V prostředí DevOps slouží k tomuto účelu sekce `Pipeline`. Zde je definována posloupnost příkazů provádějící kompilaci kódu do produkční podoby. Build je spuštěn po každém nahrání libovolných změn do cloudu.

2.4.2.2 Automatický deploy

Deploy, nebo-li nasazení systému na produkci, je další důležitá část procesu, jež lze efektivně automatizovat. K tomuto účelu slouží v DevOps sekce Release. V této části je sestrojena posloupnost příkazů pracující s kompilovanou produkční verzí aplikace, která je přemístěna na produkční server. Na obrázku 3 je vidět, že klientská a serverová část jsou nasazovány na samostatné servery.



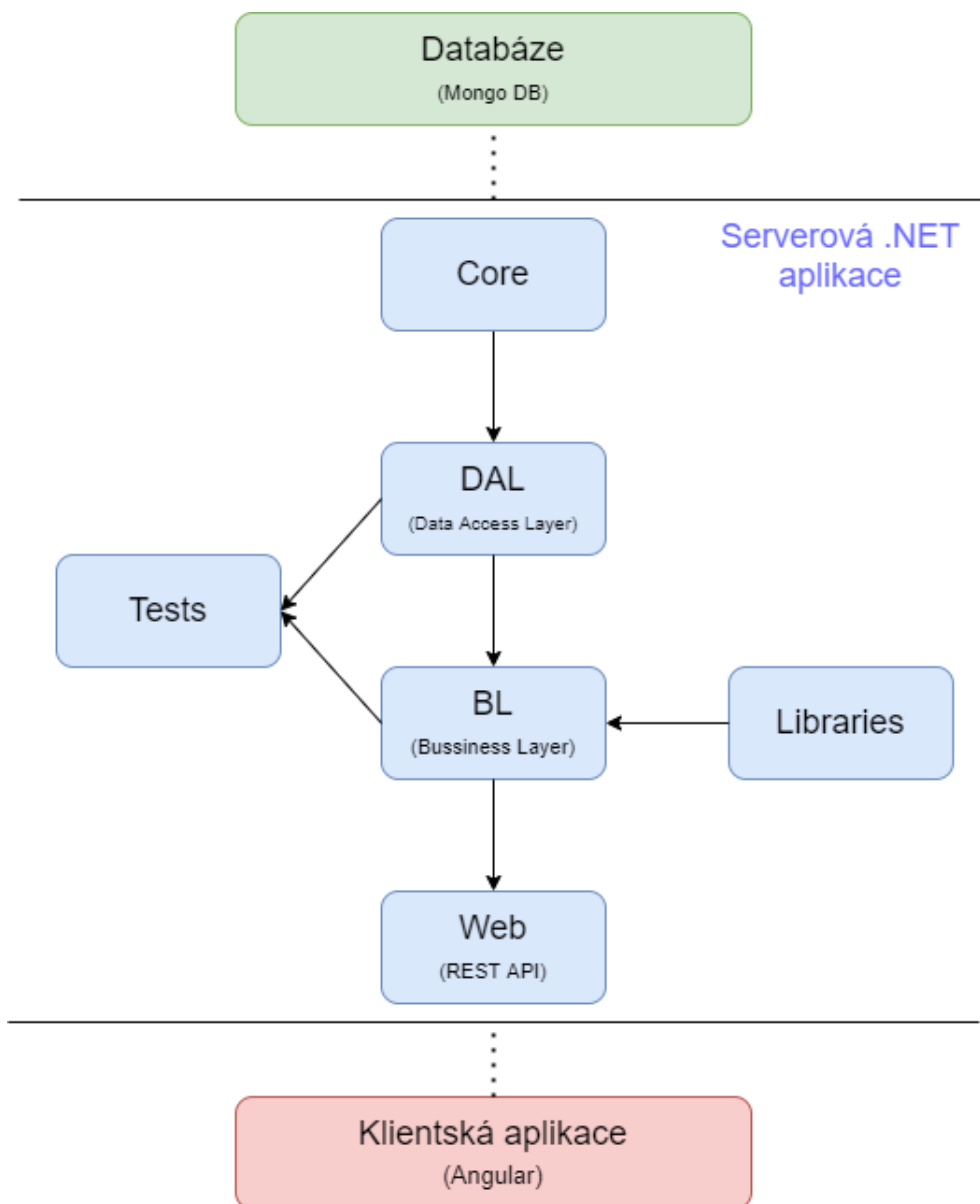
Obrázek 3: Microsoft DevOps Release detail

2.4.3 Azure

Azure poskytuje cloudové služby po celém světě. Podporuje širokou sadu služeb, které jsou neustále rozšiřovány společností Microsoft. Ke správě a přístupu do Azure slouží systém Azure Portal. Webová aplikace je rozdělena na dvě části - server a klient. Každá z těchto částí je ukládána na samostatný server, díky čemuž je možné při nárůstu systému jednotlivé části samostatně škálovat.

3 Architektura Serveru

Architektura je jedním z nejdůležitějších částí při rozvržení a implementaci systému. Pomáhá udržet systém stabilní vůči změnám či jeho růstu. Častou implementací bývá rozdělení projektu do jednotlivých vrstev, kde každá vrstva obsluhuje určitou část systému. Tato sekce bude věnována především představením jednotlivých částí architektury.



Obrázek 4: Architektura serveru

3.1 Core

První vrstvou v architektuře je projekt `Core`. Do této vrstvy jsou psány veškeré sdílené definice, metody, výčtové typy a pomocné metody.

3.1.1 Model

Složka `Model` obsahuje definici modelů a strukturu databázových objektů. Třídy jsou vytvářeny samostatně dle logického uvážení a objektově orientovaných principů. V případě, že existuje objekt, který bude obsažen ve více `Models`, je potřeba jej přidat do složky `Shared`.

3.1.2 Constant

Obsahuje všechny sdílené konstanty využívané v projektu. Nejčastěji se jedná o konfigurační či pomocné řetězce využívané napříč celým projektem.

3.1.3 Extension

Rozšiřující metody jsou přidávány do složky `Extension`. Jedná se o metody rozšiřující určitou funkcionalitu datových typů. Ve zdrojovém kódu [1](#) můžeme vidět implementaci metody `IsEmpty`, která přijímá list a vrací pravdivostní hodnotu.

```
1 public static class ConditionalExtension
2     {
3         public static bool IsEmpty<T>(this List<T> any)
4         {
5             return any == null || any.Count() == 0;
6         }
7     }
```

Zdrojový kód 1: Implementace boolean metody `IsEmpty`

Pomocí použití klíčového slova `this` v parametru metody je možné použít metodu bez parametrů následujícím způsobem.

```
1
2 if (products.IsEmpty())
3     ...
```

Zdrojový kód 2: Použití extension metody `IsEmpty`

3.2 DAL - Data Access Layer

Vrstva Data Access Layer slouží k abstrakci datového modelu. Její důležitou výhodou je poskytnutí ochrany vůči změně úložiště dat. Funguje na jednoduchém principu samostatnosti jednotlivých `Repository`. Každá `Repository` slouží pro přístup k jednomu datovému modelu.

Druhou, podstatnější výhodou, je ochrana vůči redundanci kódu v dotazech na databázi. Všechny dotazy jsou zaobaleny do metod umístěných v jednotlivých `Repositories`.

3.2.1 Repository pattern

K přispění udržitelnosti tohoto principu je definována hlavní třída `Repository` obsahující obecné dotazy na databázi. Všechny ostatní `Repositories` jsou potomkem této třídy.

Ve zdrojovém kódu 3 je možné vidět metodu `Search` ve třídě `ProductRepository`. Funkce využívá v jejím těle další metodu ze třídy `Repository.cs` s názvem `FindManyAsync`, jež přijímá jako parametr lambda funkci vracející pravdivostní hodnotu.

```
1
2 public async Task<List<ProductModel>> Search(List<string>
   categories)
3 {
4     return await FindManyAsync(x =>
5         categories.Contains(x.CategoryId)
6         || categories.Count == 0);
7 }
```

Zdrojový kód 3: Metoda `Search` v `ProductRepository`

3.2.2 Unit Of Work

Způsob ukládání dat do databáze pomocí třídy `Unit Of Work` je možné přirovnat v relačních databázích k principu fungování transakcí. Data jsou provizorně ukládána do paměti. Poté, co daný proces proběhne úspěšně, jsou uloženy všechny informace najednou, aniž by došlo k nekonzistenci dat.

NoSQL dokumentová databáze funguje na principu `Thread Safe`, což umožňuje zadávat dotazy ve více vláknech zároveň. Tento princip zásadním způsobem urychluje běh celé serverové části aplikace.

3.3 BL - Business Logic

Nejdůležitější vrstvou v systému je Business Logic, která obsahuje hlavní logiku celé aplikace a všechny důležité algoritmy, které jsou potřebné k fungování systému jako celku. V této části projektu je využívána především vrstva DAL pro přístup k datům, a také vrstva Libraries sloužící pro přístup k externím cloudovým službám a pomocným knihovnám.

3.3.1 Servisní architektura

Vrstva BL využívá servisní architekturu rozdělující jednotlivé logické celky do vlastních tříd. Každá třída, nebo-li *Service*, přijímá v parametru metody objekt typu *Request*, ve kterém jsou definovány všechny potřebné parametry pro výpočet. Následně vrací jednotný objekt typu *Response*, jež je jeho výstupním objektem. Tento přístup umožňuje zachovat principy REST API a flexibilitu vůči případným změnám.

3.3.1.1 Request

Každá veřejná metoda obsahuje nejčastěji jeden parametr typu objekt s postfixem - *Request*. V tomto objektu jsou uloženy všechny potřebné parametry, které jsou rovněž vyžadovány na straně klienta.

3.3.1.2 Response

Response, nebo-li odpověď, je využívána pro systémové vrácení dat. Z různých metod jsou uživateli vráceny jednotné objekty, čímž je udržována konzistentní struktura dat. Nejčastějšími výsledky bývá chybová hláška, data či celkový počet vrácených výsledků.

Třída *Response* přijímá data pouze pokud se jedná o objekt typu *ResponseData*. Každá nová třída musí dědit z této hlavní třídy. Poté může být použita jako návratový typ metody.

3.3.1.3 Příklad použití

Na následujícím příkladu je možné vidět metodu `Update` přijímající jeden parametr typu objekt - `CategoryRequest`. Metoda vyhledá kategorii, upraví její data a uloží změny do databáze. Následně vrátí zpět upravený objekt pomocí metody `MapCategory` ze třídy `SharedMapService`, která převádí interní databázový objekt na objekt typu `Response`. Důvodem mapování objektů je určitá ochrana uložených dat. Klientovi by neměly být nikdy vrácena všechna data z databáze. Vždy je potřeba kvůli zabezpečení provést určitý výběr dat.

```
1 public async Task<Response<CategoryResponse>>
   Update(CategoryRequest request)
2     {
3         var category = await
           _uow.Categories.FindAsync(new
           Guid(request.Id));
4         if (category == null)
5             return new Response<CategoryResponse>(
6                 ResponseStatus.NotFound,
7                 null,
8                 ResponseMessagesConstans.NotFound);
9
10        category.Name = request.Name;
11        category.UrlName =
           WebUtility.UrlEncode(request.UrlName);
12        category.ParentCategoryId =
           request.ParentCategoryId;
13
14        await _uow.Categories.ReplaceAsync(
15            category.Id,
16            category);
17        return new Response<CategoryResponse>(
18            ResponseStatus.Ok,
19            _mapService.MapCategory(category));
20    }
```

Zdrojový kód 4: Metoda `Update` ve třídě `CategoryService`

3.4.1 Controllers

Controllers jsou třídy obsahující metody použitelné klientem. Pojmem Endpoint je možné rozumně každou veřejnou metodu v těchto třídách. Třídy by měly být rozděleny do logických celků zastupujících danou funkcionalitu. V následující ukázce kódu 5 je možné vidět Controller s názvem OrderController zastřešující přístup k objednávce. V těchto metodách je možné explicitně uvést pomocí Attributes typ metody, název či zabezpečení. Všechny třídy dědí z hlavního controlleru BaseApiController, jež je potomkem hlavního ApiController z frameworku .NET.

Ve zdrojovém kódu 5 je přidán atribut [JwtAuthenticationAdmin], jež umožňuje přístup uživatelům pouze s daným oprávněním.

```
1 [HttpGet]
2 [JwtAuthenticationAdmin]
3 [Route("Search")]
4 public async Task<IHttpActionResult> Search([FromUri]
5     OrderSearchRequest request)
6 {
7     var response = await _orderService.Search(request);
8     return new CreateResult(response);
9 }
```

Zdrojový kód 5: Endpoint Search v OrderController

3.4.2 Attributes

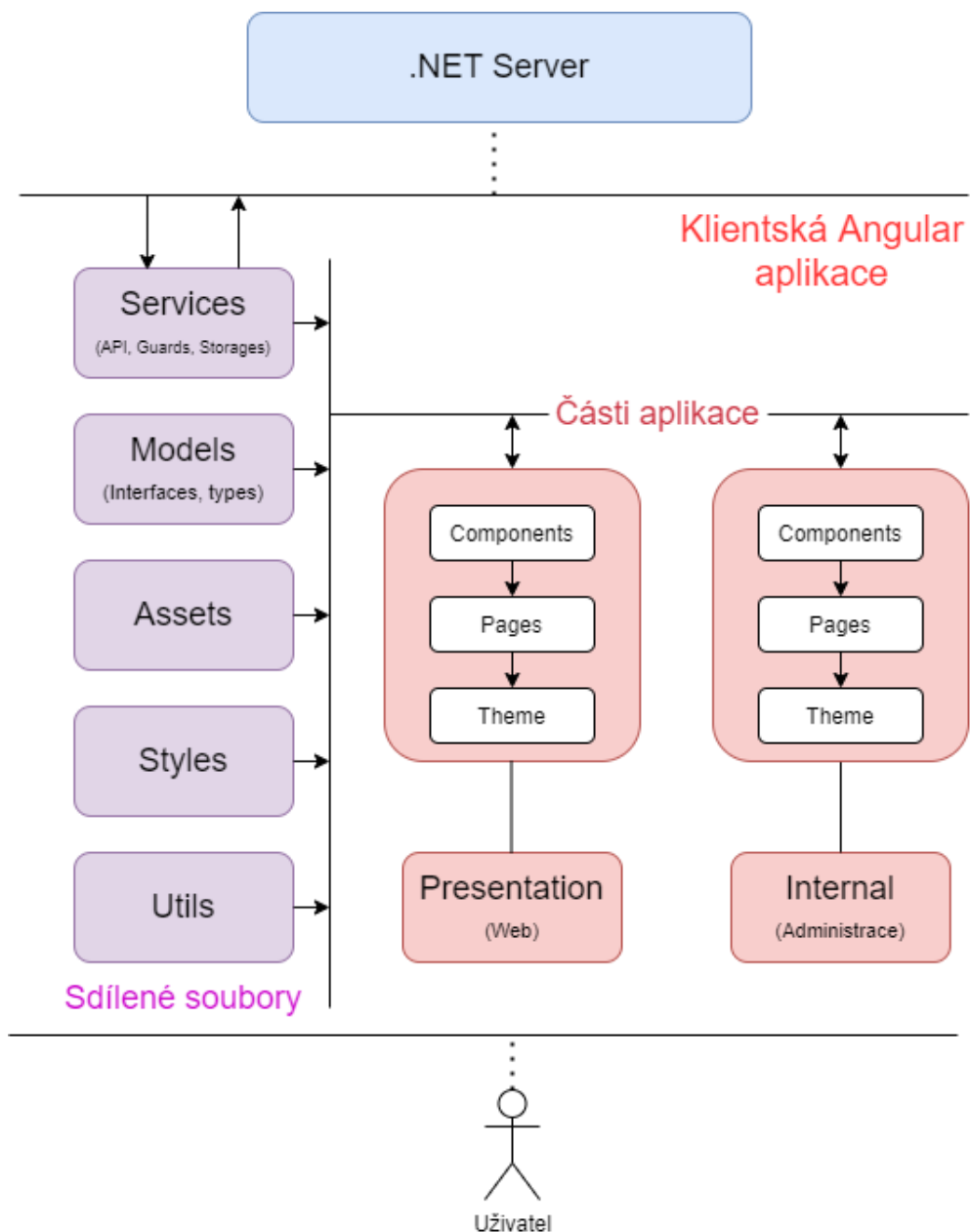
Attribute umožňují přidávat funkcionalitu, která je provedena před zavoláním metody, nad kterou je atribut uveden. Nejčastěji se jedná o metody pro autorizaci uživatele.

3.5 Libraries

Další pomocnou vrstvou je Libraries. Tato část obsahuje veškeré externí knihovny využívané v hlavní bussiness logice. Například mailing, šifrování či generování JWT tokenů. Pokud je možné oddělit danou část systému od hlavní logiky aplikace, měla by být zde vytvořena jako pomocná knihovna. Umístění použití externích knihoven do této části aplikace umožňuje flexibilně reagovat na změny, které mohou přijít zvenčí. V případě, že by došlo ke zrušení mailingové služby SendGrid, jež je v systému využívána, bude potřeba přepsat funkcionalitu pouze ve třídě MailingLibrary.

4 Klient

Druhou částí systému je Angular SPA aplikace napsaná v Typescriptu. V této části je popsána technická struktura klientského webu. Na obrázku 6 je zobrazena základní struktura projektu.



Obrázek 6: Struktura Angular aplikace

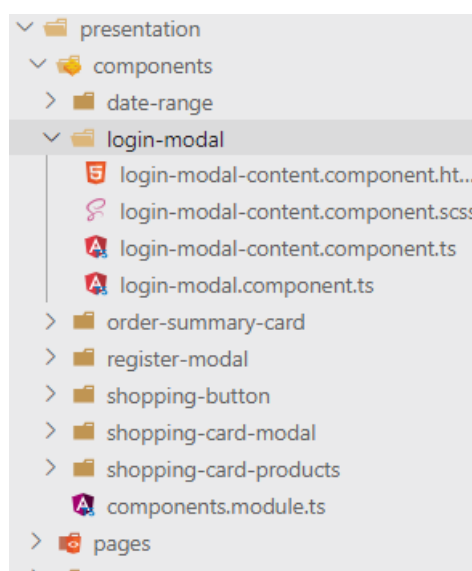
4.1 Presentation

Složka `presentation` obsahuje kompletní část prezenčního webu - komponenty, stránky, routing a šablony.

4.1.1 Components

V adresáři `components` jsou obsaženy sdílené komponenty používané napříč celou prezentační vrstvou. Všechny složky jsou tvořeny jako samostatné komponenty nahrávané do modulu `components.module.ts`. Základní modul je poté vložen do ostatních částí webu.

Na obrázku 7 je zobrazena struktura Angular komponenty. Většinou obsahuje 3 základní typy souborů: HTML, styly, soubory v jazyce Typescript obsahující logiku komponenty.



Obrázek 7: Struktura komponenty

4.1.2 Pages

Složka `pages` obsahuje konkrétní stránky zobrazené na webu. Každá `page` je vytvářena jako samostatný modul. Toto rozdělení umožňuje efektivně využívat `Lazy loading` aplikace. `Lazy loading` načítá pouze aktuálně používané moduly, což urychluje načítání celé aplikace.

4.1.3 Theme

Poslední částí `presentation` adresáře je složka `theme`, nebo-li šablona. Zde je definován obsah zobrazený ve všech částech aplikace jako hlavička či patička stránky.

4.2 Internal

Adresář `Internal` obsahuje administraci pro správu webu. Architektura této části je navržena stejně jako část `presentation`. Je tedy rozdělena do adresářů `components`, `pages` a `theme`.

4.3 Services

`Services`, nebo-li služby, jsou v Angularu využívají pro přístup k datům z různých externích úložišť. Příkladem mohou být vrácená data z `Request` požadavku či z lokálního úložiště nebo cookie. `Services` mohou být také umístěny uvnitř modulů.

4.3.1 API

Všechny `Services`, které slouží pro práci s daty na API, jsou uloženy ve složce `service/API`. Každá služba musí dědit z hlavní `API.service`, která poskytuje veškeré CRUD metody pro práci s API. Ve zdrojovém kódu 6 je možné vidět konkrétní příklad metody pro přidání kategorie.

```
1 public add(request: ICategoryCreateRequest):  
    Observable<ICategory> {  
2     this.categories = [];  
3     return this._API.post(`${this.className}/Add`, request);  
4 }
```

Zdrojový kód 6: Metoda pro přidání kategorie

4.3.2 Storage

Adresář `storage` obsahuje všechny `Services` pracující s pamětí `LocalStorage`. Úložiště je využíváno pro uložení části dat rozpracované objednávky u nepřihlášeného uživatele.

4.3.3 Guards

`Guards` jsou speciální `Services` implementující rozhraní `CanActive`, které umožňuje zavolat metodu před načtením modulu či komponenty. Používají se převážně pro zamezení přístupu určitým rolím do různých částí systému. Metody pro ověření jsou volány ještě před načtením modulu, což zvyšuje bezpečnost celé aplikace.

4.4 Models

Další částí projektu je adresář `models`, ve kterém jsou uloženy rozhraní definující datovou strukturu aplikace. Ve zdrojovém kódu 7 je zobrazeno základního rozhraní objednávky.

```
1 export interface IOrder {
2     id: string;
3     state: OrderState;
4     orderNumber: number;
5     orderNumberFormatted: string;
6     service: IOrderService;
7     customer: IOrderCustomer;
8     transport: IOrderTransport;
9     payment: IOrderPayment;
10    calculatedData: ICalculatedOrder;
11 }
```

Zdrojový kód 7: Část modelu objednávky

4.5 Utils

Složka `utils` obsahuje sdílené metody, které jsou používány napříč celým projektem. Nejčastěji se jedná o obecné metody pro transformaci řetězců do potřebné podoby.

4.6 Assets a Styles

Poslední částí tohoto projektu jsou složky `assets` a `styles`. Veškeré soubory z adresáře `assets` jsou kopírovány nezměněné do buildu aplikace. Zde jsou ukládány veškeré ikony či logo aplikace. Dynamický obsah obrázků je automaticky nahráván do cloudové služby [Cloudinary](#).

Složka `styles` obsahuje sdílené styly, jež definují celkový vzhled aplikace. Ostatní styly jsou umístěny uvnitř jednotlivých modulů.

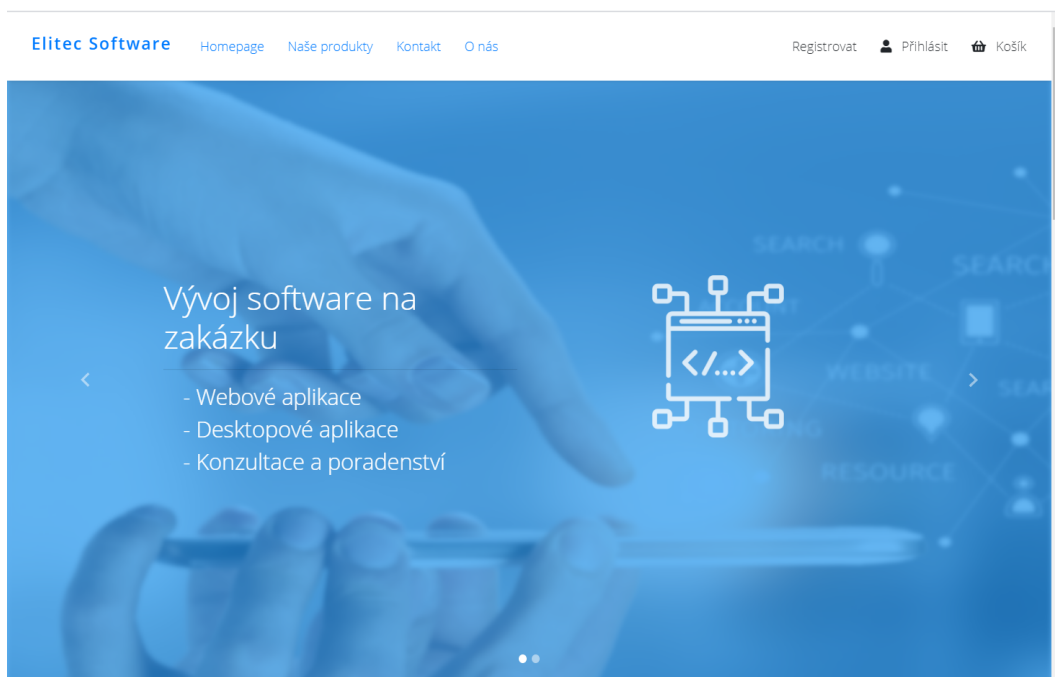
5 Uživatelský Web

V následující sekci je popsán prezenční web z pohledu klienta. Hlavním tématem je popis funkcionality a ovládání jednotlivých prvků webu. Kostra webu je složena z fixní hlavičky a patičky. Hlavní obsah je poté zobrazován na základě vybrané stránky.

5.1 Homepage

Stránka Homepage, nebo-li domovská obrazovka, slouží k souhrnu informací z celého webu. Nalezneme zde prvky poskytující klientovi základní informace pro orientaci na webu. V úvodu je zobrazen slider se základní nabídkou a karty odkazující na seznam produktu. Další částí jsou základní informace o firmě a její reference.

V horní části obrázku 8 je zobrazena fixní hlavička webu. Na levé straně je umístěn název společnosti a navigace na základní stránky webu. V pravé části jsou odkazy pro registraci či přihlášení a odkaz na košík. Pod hlavičkou se nachází již zmiňovaný slider.



Obrázek 8: Vzhled stránky Homepage

5.2 Přihlášení a Registrace

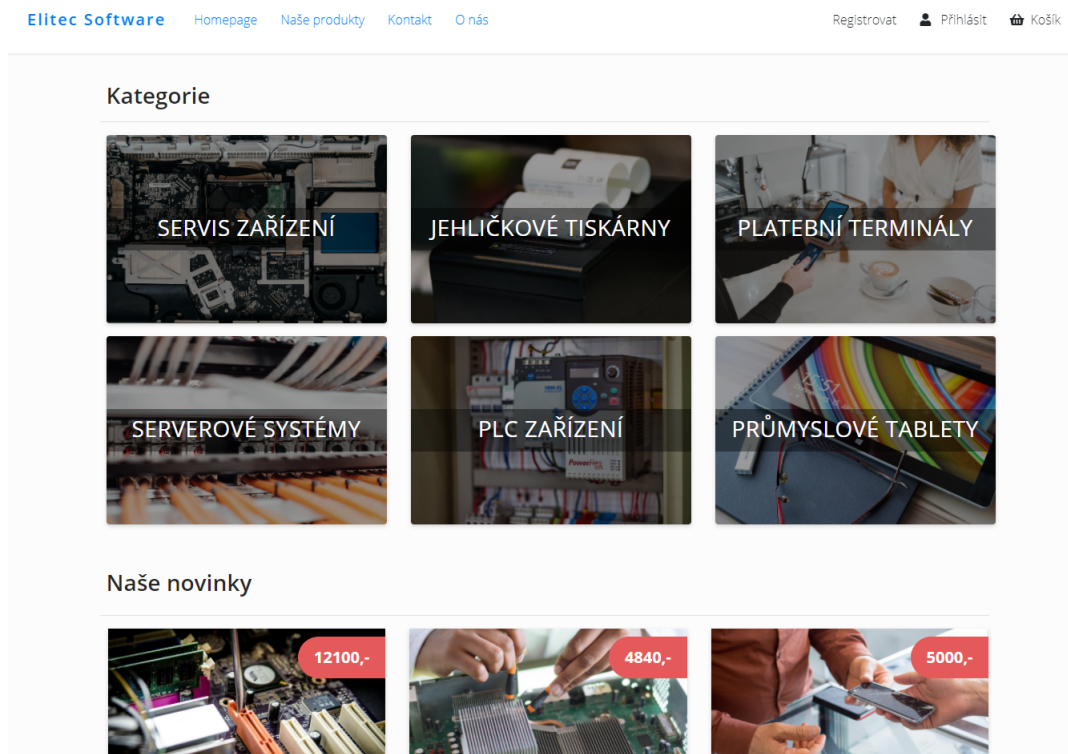
K přihlášení do systému je možné využít vyskakovací okno, které je zobrazeno po stisknutí tlačítka `Přihlásit` umístěného v pravé části hlavičky webu viz. obrázek 8. Registrace uživatelů funguje na stejném principu a je zobrazena po stisknutí tlačítka `Registrovat`.

5.3 Produkty

Stránka `Produkty` zobrazuje aktuální seznam všech dostupných produktů. Produkty jsou dostupné po kliknutí na odkaz `Naše produkty`.

V horní části stránky je umístěno navigační menu třídící produkty do jednotlivých kategorií. Kategorie jsou zobrazovány v podobě karet. Po kliknutí na konkrétní kategorii je zobrazen seznam produktů vázajících se pouze k dané kategorii.

Níže nalezneme seznam všech produktů či seznam produktů k dané kategorii. Produkty jsou na této straně zobrazovány jako samostatné karty. Karta produktu obsahuje název, popis, obrázek a cenu s DPH. Všechny tyto části kromě ceny jsou ukládány samostatně tak, aby administrátor mohl volit různá data pro přehled produktu a jeho detail. Po kliknutí na kartu produktu je uživatel přeměrován na jeho detail.




Obrázek 9: Vzhled stránky seznamu produktů

5.4 Detail Produktu

Detail Produktu obsahuje detailní informace o produktu. Jedná se o hlavní obrázek, galerii, název, popisky, cenu či skladovou informaci. Po přidání/odebrání produktu je vyvolána příslušná notifikace.

Elitec Software Homepage Naše produkty Kontakt O nás Registrat

Naše produkty / Průmyslové tablety / XSLATE D10 Turbo



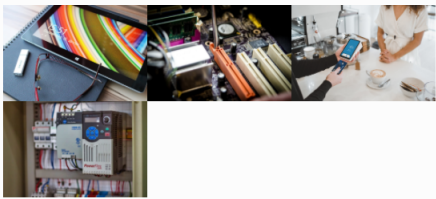
XSLATE D10

Tablet - 6,1" IPS 1792 × 828, procesor Apple A13 Bionic 6jádrový, RAM 4 GB, Interní paměť 128 GB, zadní fotoaparát s optickým zoomem 12 Mpx (f/1,8) + 12 Mpx (f/2,4), přední fotoaparát 12 Mpx, optická stabilizace, GPS, Glonass, NFC, LTE, Lightning port, voděodolný dle IP68, single SIM + eSIM, neblokováný, rychlé nabíjení 18W, bezdrátové nabíjení, baterie 3110 mAh, iOS 13

Skladem

12100 Kč | s DPH 21% **10000 Kč** | bez DPH

PŘIDAT DO KOŠÍKU



XSLATE D10 Turbo

Tablet - 6,1" IPS 1792 × 828, procesor Apple A13 Bionic 6jádrový, RAM 4 GB, Interní paměť 128 GB, zadní fotoaparát s optickým zoomem 12 Mpx (f/1,8) + 12 Mpx (f/2,4), přední fotoaparát 12 Mpx, optická stabilizace, GPS, Glonass, NFC, LTE, Lightning port, voděodolný dle IP68, single SIM + eSIM.

Obrázek 10: Vzhled stránky detail produktu

5.4.1 Produkt a služba

U produktů jsou rozlišovány 2 základní typy - fyzický produkt a služba k produktu. Typ produktu je definován v administraci. Při výběru služby je zobrazeno vyskakovací okno s výběrem konkrétního termínu provedení. V případě produktu je zadáno množství objednaných kusů omezené jeho vrchní hranicí dle množství na skladě, které je možné nastavit v administraci.

5.5 Košík

Košík je zobrazen po kliknutí v hlavičce na ikonku s popiskem Košík. Po kliknutí vyjede vyskakovací okno se seznamem objednaných produktů či služeb. Položky je možné z košíku buďto úplně odstranit či změnit počet objednaných kusů u produktů.

Pokud košík obsahuje alespoň jednu položku, je možné pokračovat v procesu objednávky dál pomocí tlačítka Pokračovat k objednávce. V případě, že nákup není dokončen, lze se vrátit zpět pomocí tlačítka Chci ještě nakupovat.

Produkt	Cena / ks	Počet	Celková cena	Odstranit
LESAK ITER7	9680 Kč / ks	1	9680 Kč	X
Electric TM221ME16R	21780 Kč / ks	2	43560 Kč	X
XSLATE D10 Turbo	12100 Kč / ks	1	12100 Kč	X
Oprava HDD SSD	4840 Kč / ks 22.01.2021	1 ks	4840 Kč	X
Cena celkem			70180 Kč s DPH 58000 Kč bez DPH	

[POKRAČOVAT K OBJEDNÁVCE](#)

[CHCI JEŠTĚ NAKUPOVAT](#)

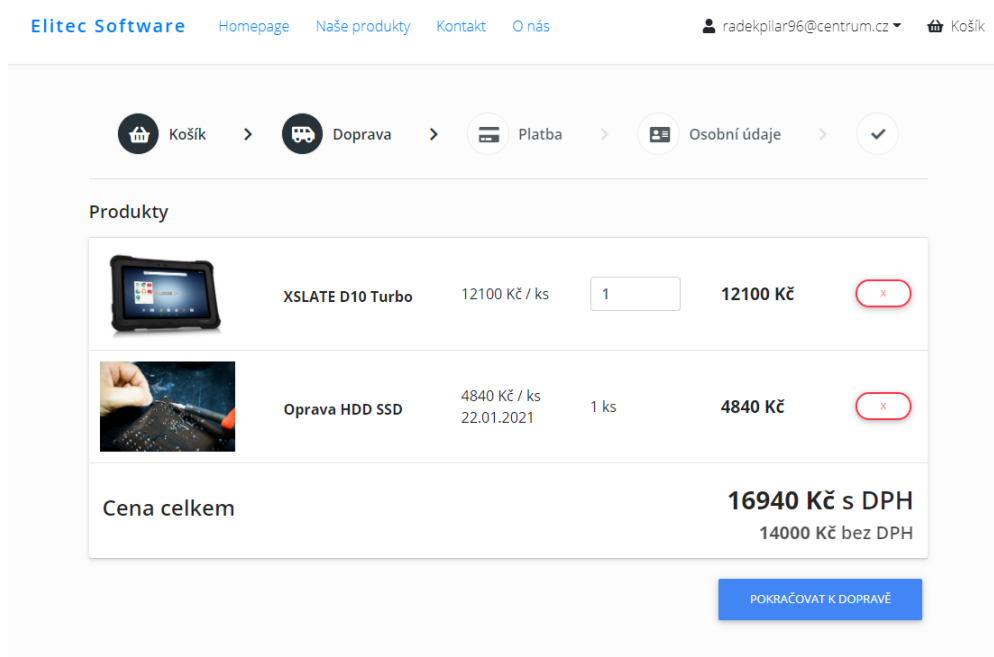
Obrázek 11: Vzhled stránky košík

5.6 Objednávka

V následující sekci je popsán proces objednávky, který je rozdělen do čtyř částí - Košík, Doprava, Platba a Osobní údaje.

5.6.1 Košík

Jako první jsou rekapitulovány uživateli jeho vybrané položky v podobě košíku, kde může provést poslední změny v objednávce. Základním předpokladem k vytvoření objednávky je vybrat alespoň jeden produkt/službu. Bez splnění tohoto předpokladu není možné pokračovat v objednávce dál.

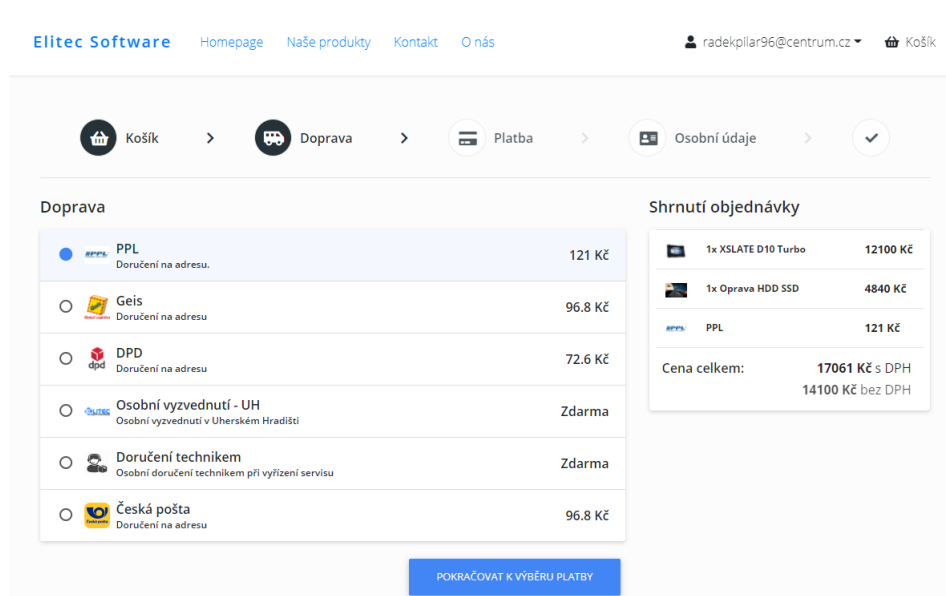


Obrázek 12: Vzhled stránky košík

5.6.2 Doprava

Na následující stránce je možné vybrat způsob dopravy zboží ke klientovi. Tento seznam je kompletně nastavitelný v administraci, tudíž zobrazené způsoby dopravy i jejich cena můžou být proměnlivé.

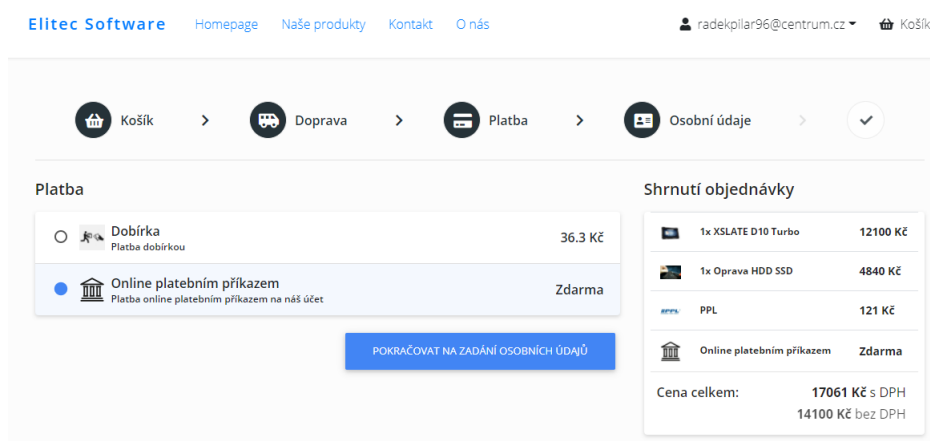
Na obrázku 13 je zobrazeno rozdělení panelů na 2 části. V levá část obsahuje údaje a formuláře k zadání objednávky a pravá část souhrn těchto informací.



Obrázek 13: Vzhled stránky doprava

5.6.3 Platba

Další sekci je `Platba`, která slouží pro výběr způsobu platby. Stejně jako u dopravy se jedná o generovaný seznam, který je možné rovněž nastavit v administraci. Každá platba obsahuje svůj typ dle kterého je zobrazována na základě vybraného typu dopravy. Pro pokračování k další sekci `Osobní údaje` musí uživatel zvolit jeden typ platby.



Obrázek 14: Vzhled stránky platba

5.6.4 Osobní údaje

Na stránce Osobní údaje uživatel zadává všechna potřebná data pro dokončení objednávky. Jedná se o osobní, kontaktní a doručovací údaje.

The screenshot shows a web interface for entering personal data. At the top, there is a navigation bar with icons for 'Košik', 'Doprava', 'Platba', and 'Osobní údaje'. The main content area is titled 'Osobní údaje' and contains several sections:

- Osobní údaje:** Fields for 'Jméno' (Radek) and 'Příjmení' (Pilař), 'Email' (radek.pilar96@centrum.cz), and 'Telefon' (420 747 727 772).
- Adresa pro doručení:** Fields for 'Město' (Vysoka), 'PSČ' (155), and 'Stрана' (Praha, Česká Republika).
- Firemní údaje:** A checkbox labeled 'Fakturovat na firmu'.
- Shrnutí objednávky:** A table listing items and their prices:

	1x XSLATE D10 Turbo	12100 Kč
	1x Oprava HDD SSD	4840 Kč
	PPL	121 Kč
	Online platebním příkazem	Zdarma
Cena celkem:		17061 Kč s DPH 14100 Kč bez DPH

At the bottom, there are two checked checkboxes: 'Souhlasím s Obchodními podmínkami' and 'Souhlasím s GDPR podmínkami', followed by a blue button labeled 'Potvrdit objednávku'.

Obrázek 15: Vzhled stránky osobní údaje

5.6.4.1 Nový vs registrovaný uživatel

Objednávka vytvořená novým uživatelem je automaticky přiřazena k nově vytvořenému účtu. Pokud objednávku vytváří již registrovaný uživatel, je objednávka svázána automaticky s tímto účtem.

5.6.4.2 Firma

Objednávky je možné vytvářet a fakturovat na právnické osoby. V tomto případě je potřeba zvolit pole Fakturovat na firmu a vyplnit všechny příslušné údaje. Po uzavření objednávky je uživateli zobrazeno upozornění o jejím úspěšném dokončení. Následně je odeslán email o potvrzení objednávky obsahující odkaz na detail objednávky.

5.7 Detail objednávky

Detail objednávky je možné zobrazit po přihlášení po kliknutí na odkaz *Moje objednávky*, více je tato část popsána v sekci 5.8. Druhou možností je rozkliknutí adresy z potvrzujícího emailu. Na této stránce je možné vidět souhrn všech informací k objednávce. Na obrázku 16 je zobrazen příklad objednávky, na kterém uživatel sleduje její stav, seznam položek, ceny, výběr dopravy, platby či jeho doručovací a kontaktní údaje.

Objednávka č. 00000021

Potvrzená > Zaplacená > Připravena k odeslání > Odeslaná > **Doručená**

Shrnutí objednávky

	Zebra XC6	18150 Kč / ks	1 ks	18150 Kč
	Zebra ET50	16940 Kč / ks	1 ks	16940 Kč
	XSLATE D10 Turbo	12100 Kč / ks	1 ks	12100 Kč

Doprava **Platba**

DPD **72.6 Kč** **Online platebním příkazem** **Zdarma**

Celková cena za objednávku **47262.6 Kč s DPH**
39060 Kč bez DPH

Doručovací údaje **Kontaktní údaje**

Toufarova 2954/5
15500 Mesto
Radek Pilař

radekpilar96@centrum.cz
+420730939334

Obrázek 16: Vzhled stránky detail objednávky

5.7.1 Stav

V horní sekci stránky je vidět aktuální stav objednávky. Pro přehlednost jsou zobrazeny i předešlé a následující stavy objednávky. Aktuální a předešlé stavy jsou vyznačeny tmavou barvou, a naopak následující stavy neoznačeně.

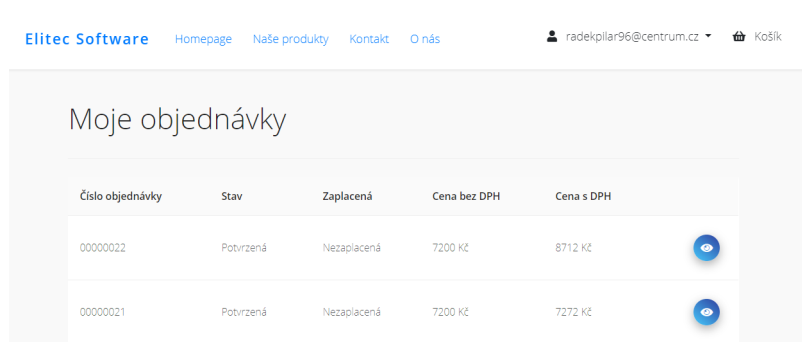
5.7.2 Mailing

Poté, co uživatel odešle objednávku, obdrží email s potvrzením této objednávky. Email obsahuje také odkaz na detail objednávky. Pro zajímavost v odkazu ob-

jednávky je umístěn token, díky kterému je uživatel přímo po načtení stránky přihlášen, tudíž má práva k zobrazení jeho objednávky.

5.8 Seznam objednávek

Na stránce *Moje objednávky* uživatel vidí přehled svých objednávek. Na pravé straně má možnost zobrazit detail dané objednávky pomocí kliknutí na modrou ikonu oka.



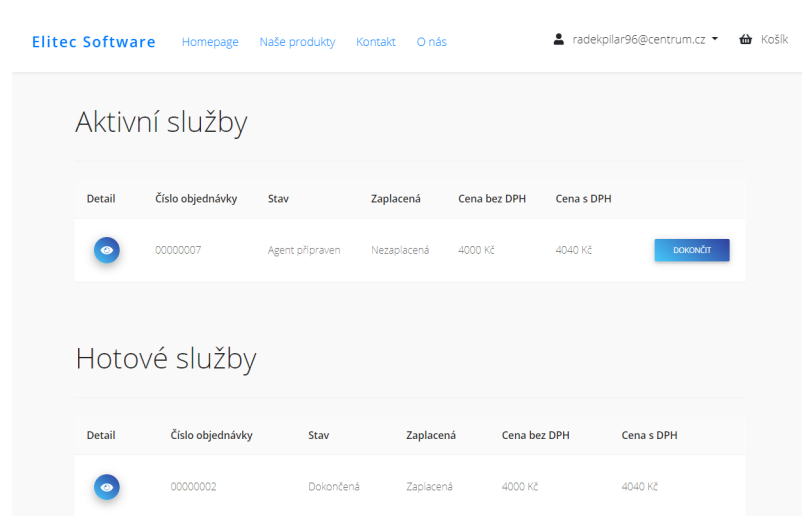
The screenshot shows the 'Moje objednávky' page with a table of orders. The table has columns for 'Číslo objednávky', 'Stav', 'Zaplacená', 'Cena bez DPH', and 'Cena s DPH'. There are two rows of data, each with a blue eye icon on the right side.

Číslo objednávky	Stav	Zaplacená	Cena bez DPH	Cena s DPH	
00000022	Potvrzená	Nezaplacená	7200 Kč	8712 Kč	
00000021	Potvrzená	Nezaplacená	7200 Kč	7272 Kč	

Obrázek 17: Vzhled stránky seznamu objednávek

5.9 Přiřazené služby

Stránka *Přiřazené služby* je viditelná pouze pro agenty a administrátory. V této sekci agenti sledují své aktivní služby, které jsou přiřazeny administrátorem. Jednotlivé služby agent obsluhuje a po jejich dokončení potvrdí každou službu pomocí tlačítka *Dokončit*. Níže může rovněž sledovat historii všech provedených služeb v sekci *Hotové služby*.



The screenshot shows the 'Aktivní služby' page with a table of active services. The table has columns for 'Detail', 'Číslo objednávky', 'Stav', 'Zaplacená', 'Cena bez DPH', and 'Cena s DPH'. There is one row of data with a blue eye icon on the left and a 'DOKONČIT' button on the right.

Detail	Číslo objednávky	Stav	Zaplacená	Cena bez DPH	Cena s DPH	
	00000007	Agent připraven	Nezaplacená	4000 Kč	4040 Kč	<input type="button" value="DOKONČIT"/>

Below the active services table, there is a section for 'Hotové služby' with a similar table structure. It shows one row of data with a blue eye icon on the left.

Detail	Číslo objednávky	Stav	Zaplacená	Cena bez DPH	Cena s DPH	
	00000002	Dokončená	Zaplacená	4000 Kč	4040 Kč	

Obrázek 18: Vzhled stránky seznamu objednávek

6 Uživatelská Administrace

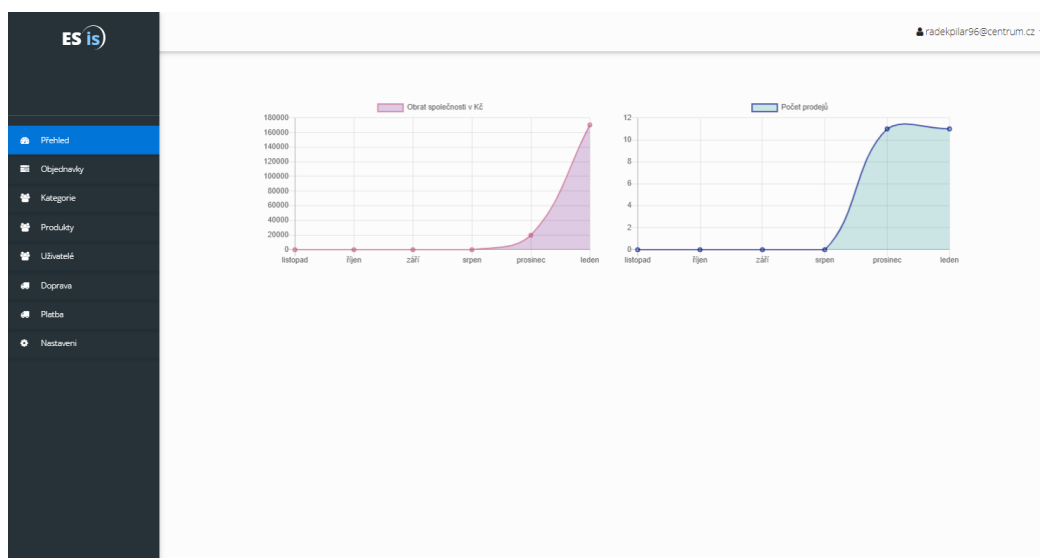
V následující sekci je popsána administrace systému. Administrace umožňuje správu objednávek, produktů, uživatelů a dalších důležitých částí systému. Do této sekce má přístup pouze uživatel s rolí administrátora. Uživatelům i agentům je zde přístup zakázán. Struktura administrace je rozdělena na 2 části - levé menu, které slouží pro přístup ke všem stránkám a hlavní variabilní obsah.

6.1 Přihlášení

Přihlášení do administrace je možné zobrazit přidáním `is` za hlavní url webu. Pro přihlášení musí administrátor zadat svůj přístupový email a heslo.

6.2 Dashboard

Úvodní stránka administrace slouží k zobrazení statistik obrátů a prodejů společnosti. Administrátor zde může sledovat aktuální finanční vývoj za posledních 6 měsíců.

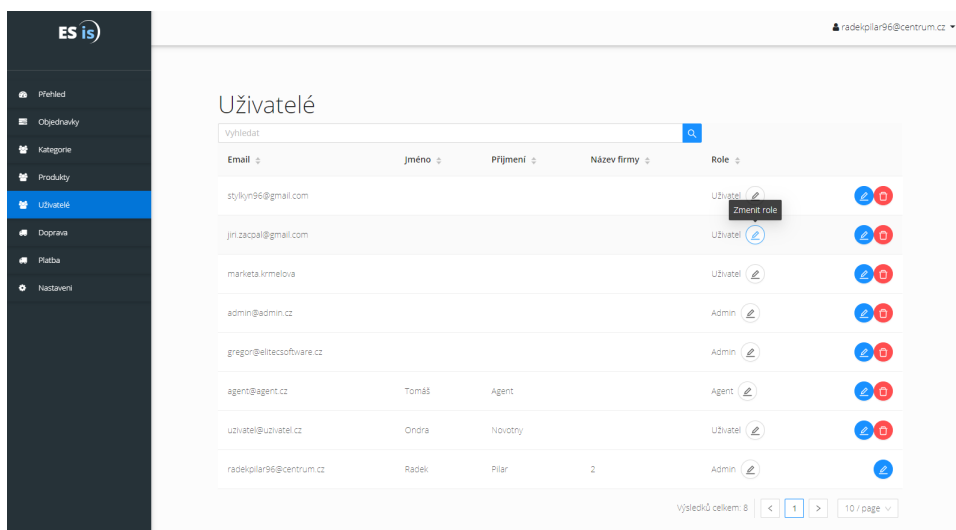


Obrázek 19: Vzhled administrace dashboard

6.3 Uživatelé

Další částí administrace je sekce pro správu uživatelů. Na této stránce je možné spravovat údaje jednotlivých uživatelů či měnit jejich role. V systému evidujeme 3 základní role:

1. **Uživatel** má přístup pouze na prezenční web.
2. **Agent** má všechna oprávnění uživatele, a navíc přístup do sekce **Moje služby**.
3. **Administrátor** má přístup do všech částí webu, jako jediný může upravovat role ostatních uživatelů a přiřazovat služby jednotlivým agentům.



Obrázek 20: Vzhled administrace seznamu uživatelů

6.4 Kategorie

Záložka **Kategorie** slouží pro základní správu kategorií, do kterých jsou rozdělovány jednotlivé produkty. Pokud je potřeba dočasně smazat kategorii, stačí nastavit její stav na **Neaktivní**. V případě, že dojde k úplnému smazání kategorie, jsou všechny produkty navázané na tuto kategorii automaticky zneaktivněny.

6.5 Doprava a Platba

Sekce **Doprava** slouží k vytváření jednotlivých způsobů dopravy zobrazovaných v procesu objednávky. Následně navazuje sekce **Platba**, která je použita rovněž v procesu objednávky. Jednotlivé platby jsou zobrazovány na základě vybrané dopravy. U každé dopravy i platby je možné nastavit její aktivitu na prezenčním webu.

6.6 Produkty

Stránka Produkty slouží pro kompletní správu produktů a služeb. Produkty lze fulltextově vyhledávat pomocí jejich názvů, url slugu, kategorie či jejich typu. V tabulce jsou zobrazeny základní informace, jako jsou hlavní fotografie produktu, název, url slug, kategorie, typ či ceny. V pravé části stránky nalezneme ovládací prvky pro správu produktů.

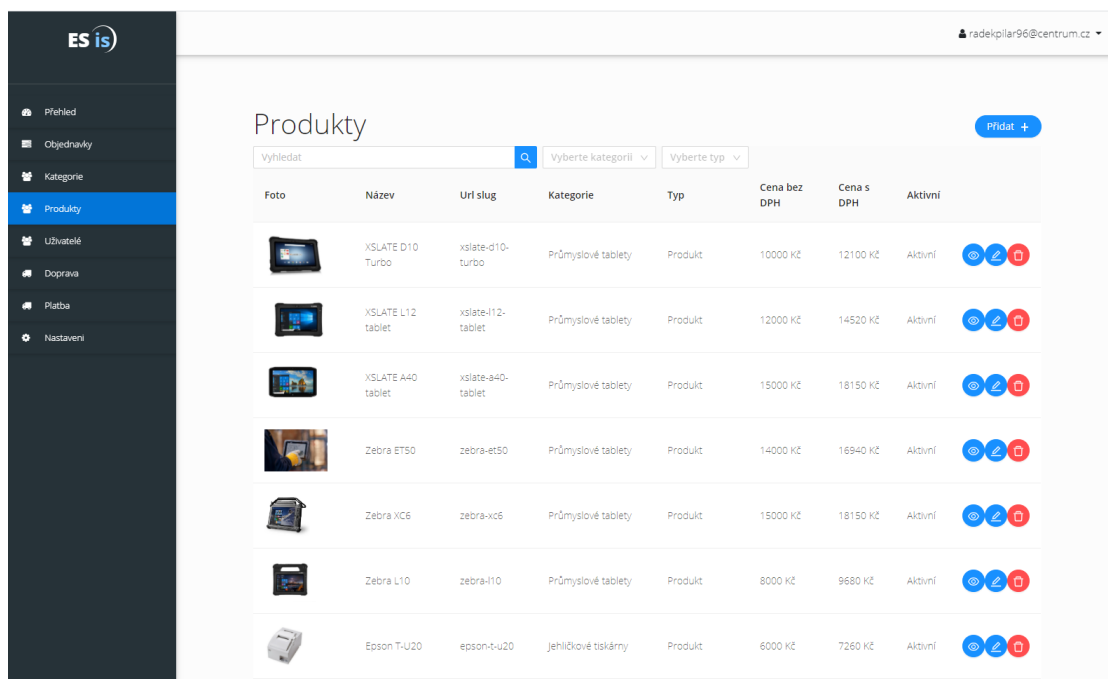









Foto	Název	Url slug	Kategorie	Typ	Cena bez DPH	Cena s DPH	Aktivní
	XSLATE D10 Turbo	xslate-d10-turbo	Průmyslové tablety	Produkt	10000 Kč	12100 Kč	Aktivní
	XSLATE L12 tablet	xslate-l12-tablet	Průmyslové tablety	Produkt	12000 Kč	14520 Kč	Aktivní
	XSLATE A40 tablet	xslate-a40-tablet	Průmyslové tablety	Produkt	15000 Kč	18150 Kč	Aktivní
	Zebra ET50	zebra-et50	Průmyslové tablety	Produkt	14000 Kč	16940 Kč	Aktivní
	Zebra XC6	zebra-xc6	Průmyslové tablety	Produkt	15000 Kč	18150 Kč	Aktivní
	Zebra L10	zebra-l10	Průmyslové tablety	Produkt	8000 Kč	9680 Kč	Aktivní
	Epson T-U20	epson-t-u20	Jehličkové tiskárny	Produkt	6000 Kč	7260 Kč	Aktivní

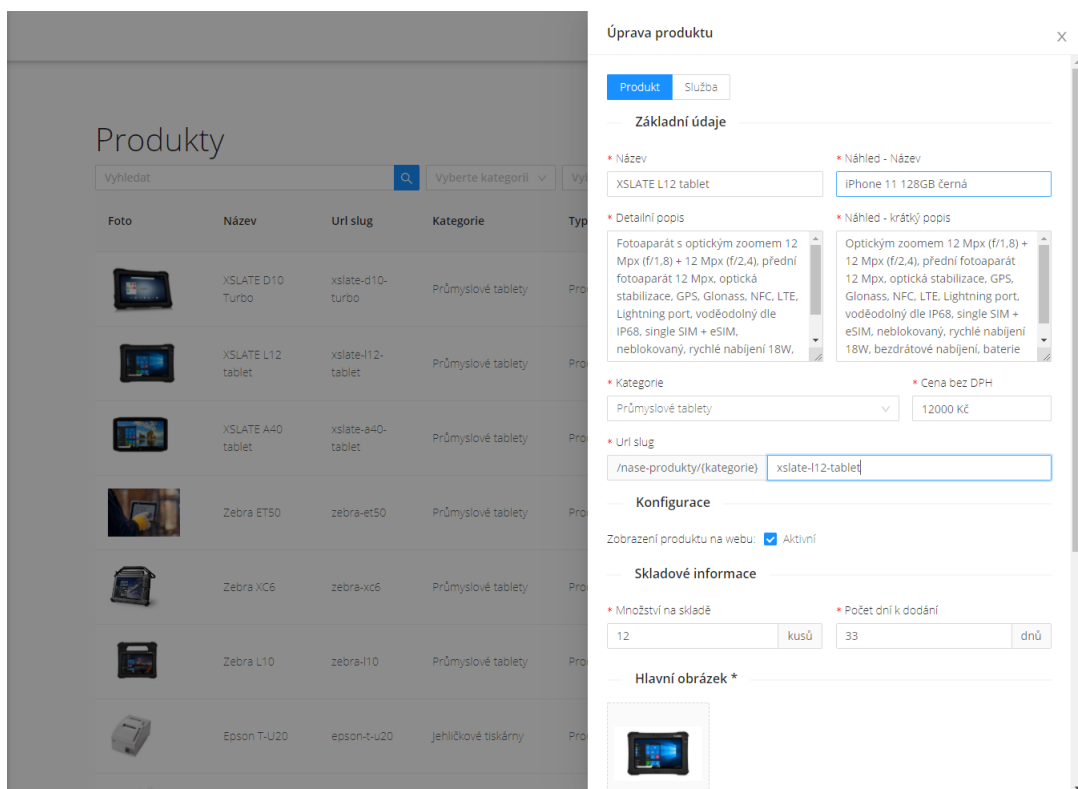
Obrázek 21: Vzhled administrace seznamu produktů

6.6.1 Přidání

Produkt je možné přidat pomocí tlačítka Přidat produkt v pravé horní části obrazovky. Pro vytvoření produktu je potřeba zadat jeho typ, název, popisky, kategorii, cenu bez DPH, hlavní obrázek, minimálně jednu fotografii pro galerii a url slug obrázku. Url slug slouží jako část url adresy pro efektivní vyhledávání SEO.

6.6.2 Editace

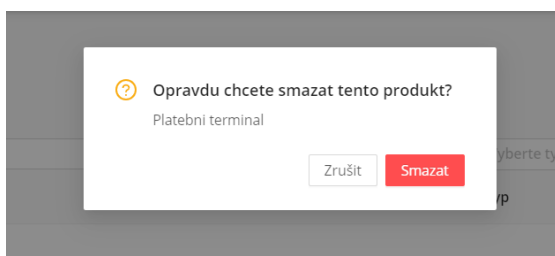
Pro editaci produktu je použit stejný formulář jako pro jeho přidání. Před editací jsou přednačtena všechna data do jednotlivých polí formuláře.



Obrázek 22: Vzhled administrace editace produktu

6.6.3 Odstranění

Odstranit produkt lze pomocí červeného tlačítka s ikonou koše. Po najetí na tlačítko je zobrazen popisek Odstranit produkt. Po stisknutí tlačítka je zobrazeno vyskakovací okno s potvrzením smazání vybraného produktu.

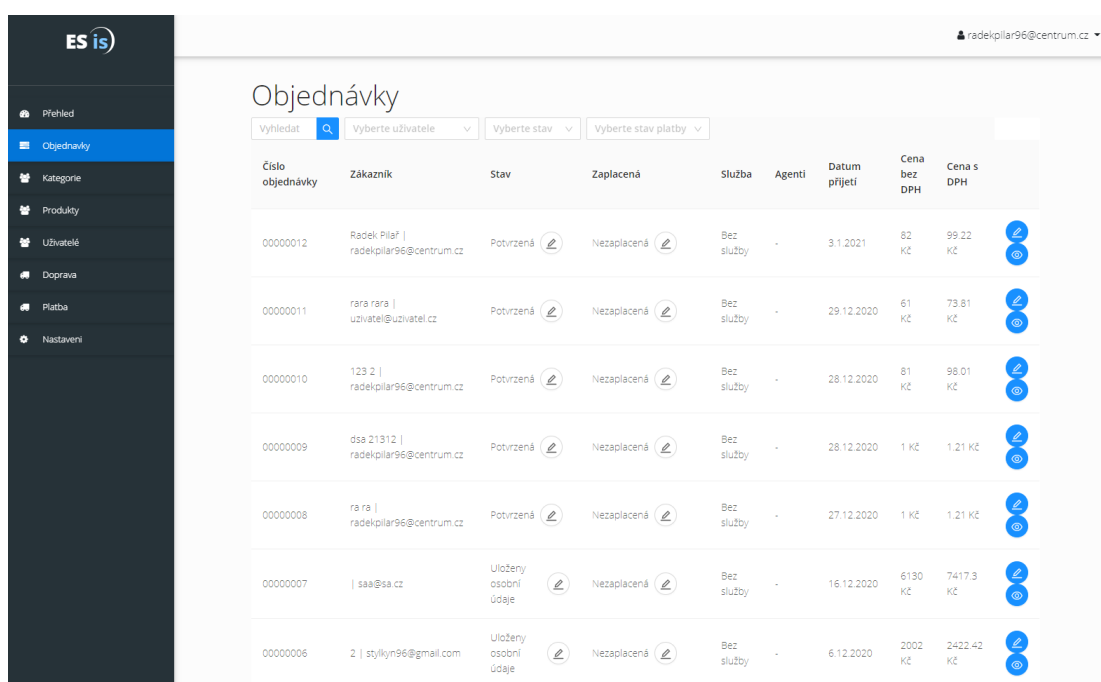
















Obrázek 23: Vzhled administrace odstranění produktů

6.7 Objednávky

Stránka Objednávky slouží pro kompletní správu všech objednávek. Na této stránce může administrátor sledovat stavy objednávek, upravovat a přiřazovat objednávky zahrnující službu jednotlivým agentům. Tyto přiřazené objednávky vidí každý agent v sekci *Moje služby* na prezenčním webu. Objednávky je možné filtrovat dle jejich čísla, zákazníka, stavu objednávky či stavu platby. V pravé části je možné zobrazit detail objednávky po kliknutí na modré tlačítko oka.

Stav objednávky je možné změnit pomocí šedého tlačítka tužky, které je umístěno u každého názvu stavu viz. obrázek 24.



Číslo objednávky	Zákazník	Stav	Zaplacená	Služba	Agenti	Datum přijetí	Cena bez DPH	Cena s DPH	
00000012	Radek Pilař radekpiler96@centrum.cz	Potvrzená	Nezaplacená	Bez služby	-	3.1.2021	82 Kč	99.22 Kč	 
00000011	rara rara uzivatel@uzivatel.cz	Potvrzená	Nezaplacená	Bez služby	-	29.12.2020	61 Kč	73.81 Kč	 
00000010	123 z radekpiler96@centrum.cz	Potvrzená	Nezaplacená	Bez služby	-	28.12.2020	81 Kč	98.01 Kč	 
00000009	dsa 21312 radekpiler96@centrum.cz	Potvrzená	Nezaplacená	Bez služby	-	28.12.2020	1 Kč	1.21 Kč	 
00000008	rara radekpiler96@centrum.cz	Potvrzená	Nezaplacená	Bez služby	-	27.12.2020	1 Kč	1.21 Kč	 
00000007	saa@saa.cz	Uloženy osobní údaje	Nezaplacená	Bez služby	-	16.12.2020	6130 Kč	7417.9 Kč	 
00000006	2 styklyn96@gmail.com	Uloženy osobní údaje	Nezaplacená	Bez služby	-	6.12.2020	2002 Kč	2422.42 Kč	 

Obrázek 24: Vzhled administrace seznamu objednávek

6.8 Nastavení

Poslední sekci je *Nastavení*, ve kterém je aktuálně možné nastavit maximální počet služeb v jeden den a počet dní, ke kterému je možné nejpozději servis objednat. Obě tyto nastavení jsou použity při výběru termínu služby na prezenčním webu.

7 Plánovaná rozšíření

K webové aplikaci jsou plánovaná následující úpravy a rozšíření

1. K webu je plánováno připojení k mobilní aplikaci, která bude sloužit agentům pro jejich vyřizování servisů a služebních cest a bude využívat stejné serverové API jako naše webová aplikace.
2. Dalším rozšířením je zapojení platební brány do procesu objednávky.
3. Pro větší přehlednost plateb bude napojena Fio banka a její API rozhraní, díky kterému přibude sekce Banka, ve které bude administrátor mít možnost sledovat finanční stránku systému.
4. V případě, že bude vysoký prodej produktů a naroste i jejich počet, budou přidány nové podkategorie, fulltextové vyhledávání a jejich filtrace.
5. V budoucnu je plánována změna systému plánování agentů, jejich dostupnost a přiřazování služeb. Do systému bude přidáno plánování agentů na jednotlivé dny a automatické přiřazování dle jejich dostupnosti.
6. Knowledge base, nebo-li nová sekce strukturovaného obsahu pro uživatele.

Závěr

Výsledkem této práce je objednávkový a rezervační systém, který umožňuje provést všechny základní procesy objednávky, registraci uživatelů, správu produktů a mnoho dalších funkcionalit. Při vývoji práce jsem využil své zkušenosti s vývojem podobných systému, které jsem získal v posledních pěti letech ve spolupracující firmě. Vzhledem ke své zkušenosti vyvíjení systému ve velmi krátkém čase, kdy nebyl prostor pro navržení kvalitní architektury, jsem se rozhodl u této práce dbát na rozvržení a kvalitu kódu tak, aby bylo možné systém dále bez problémů rozšiřovat. V projektu jsem se snažil používat nejnovější technologie a posouvat své znalosti dále. Jeden z příkladů může být přechod z relační databáze na dokumentovou či využití cloudových služeb, jež mi usnadnily mnoho nutné práce.

Při vývoji byl kladen vysoký důraz na kvalitní architekturu systému. V práci se podařilo navíc implementovat kompletní automatizace systému, se kterou se v původním návrhu nepočítalo a v textu je jí věnována sekce [2.4](#). Automatizace přinesla velkou kontrolu nad vývojem a umožnila zrychlit nasazování a detekci chyb v jejím procesu.

Na projektu je samozřejmě možné vymyslet mnoho dalších vylepšení, ale vzhledem k určitému časovému rozsahu, který bakalářská práce nabízí, se podařilo systém vyladit po funkční stránce a jeho první verze je připravena pro nasazení do produkce. Před finálním nasazením je plánovaná ještě mírná korekce designu a naplnění aplikace reálným obsahem.

Conclusions

The result of this work is an ordering and booking system that allows all basic processes of order, user registration, product management and next important functionality. During programming the whole system I took advantage my experience in developing of similar systems, which I had the opportunity to get in the last five years in a cooperating company. With my experience developing systems in very short time, where there was no time to design quality architecture, I decided wrote quality code with good architecture due to future expansion of the system. I tried to use latest technology in the project and advance my knowledge. One of examples can be the transition from a relational database to a document database or using cloud services that helped me wit my work.

During the development, I was focused on quality system architecture. In addition, was created a complete automation of the system, which was not in the original design. In the bachelor thesis is described in section 2.4.. Automation has improved control over development, speed up the deployment and detection of errors.

Of course, it is possible to invent many other improvements on the project, but due to the time range that the bachelor thesis offers, the functionality of system was successfully repaired and his first version is ready to deployment into production. Before deployment is planned slight adjustment of design and filling the application with real content.

8 Přístup k testování

K testovacím účelům doporučuji využít online verzi bez nutnosti instalace.

1. Uživatelský web <https://elitecsoftware.azurewebsites.net>
2. Uživatelská administrace <https://elitecsoftware.azurewebsites.net/is>

K testovacím účelům byly vytvořeny následující účty. Přístupové údaje jsou také dostupné v souboru `ess/readme.txt` na CD/DVD.

- **Admin**

- Přihlášení: **admin@admin.cz**
- Heslo: adminadmin

- **Agent**

- Přihlášení: **agent@agent.cz**
- Heslo: agentagent

- **Uživatel**

- Přihlášení: **uzivatel@uzivatel.cz**
- Heslo: uzivatel

9 Obsah příloženého CD/DVD

Níže uvádím popis obsahu CD/DVD, které je přiloženo k mé práci.

ess

- Složka obsahuje kompletní zdrojový kód projektu.

ess/essclient

- Zdrojový kód klientské části - Angular aplikace

ess/ESS_API

- Zdrojový kód serverové části - .NET aplikace

ess/readme.txt

- Soubor obsahuje přístupové údaje, návod na instalaci a spuštění webu.

doc

- Text práce v PDF a zdrojové soubory k jeho vygenerování.

Literatura

- [1] Introduction to the C# language and the .NET Framework [online] Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>
- [2] Úvod do dotazů LINQ [online] Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/csharp/programming-guide/concepts/linq/introduction-to-linq-queries>
- [3] What is MongoDB? Introduction, Architecture, Features & Example [online] Dostupné z: <https://www.guru99.com/what-is-mongodb.html>
- [4] MongoDB Atlas [online] Dostupné z: <https://docs.atlas.mongodb.com>
- [5] Co je to HTTPS? [online] Dostupné z: <https://topranker.cz/slovník/co-je-to-https/>
- [6] REST: architektura pro webové API [online] Dostupné z: <https://www.zdrojak.cz/clanky/rest-architektura-pro-webove-api/>
- [7] Úvod do Typescriptu [online] Dostupné z: <https://www.itnetwork.cz/javascript/typescript/uvod-do-typescriptu>
- [8] Angular official [online] Dostupné z: <https://angular.io/>
- [9] Úvod do Typescriptu [online] Dostupné z: <https://dev.to/imbilal1/what-are-aot-jit-compiler-in-angular-2-488h>
- [10] Material design for Bootstrap [online] Dostupné z: <https://mdblbootstrap.com/>
- [11] Ant Design of Angular [online] Dostupné z: <https://ng.ant.design/docs/introduce/en>
- [12] DevOps in a Scaling Environment [online foto] Dostupné z: <https://medium.com/tech-tajawal/devops-in-a-scaling-environment-9d5416ecb928>
- [13] GIT - Základy systému GIT [online] Dostupné z: <https://git-scm.com/book/cs/v2/%C3%A9vod-Z%C3%A1klady-syst%C3%A9mu-Git>
- [14] GIT - Co je Azure - Microsoft Cloud Service [online] Dostupné z: <https://azure.microsoft.com/cs-cz/overview/what-is-azure>