



MATEMATICKO-FYZIKÁLNÍ FAKULTA Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Milan Truchan

ServIS – webový systém pro firmy zabývající se opravami bagrů

Katedra distribuovaných a spolehlivých systémů

Vedoucí bakalářské práce: Mgr. Pavel Ježek, Ph.D.

Studijní program: Informatika

Studijní obor: Programování a vývoj software

Praha 2023

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne
Podpis autora

Rád by som sa podakoval vedúcemu práce Mgr. Pavlovi Ježkovi, Ph.D. za jeho vedenie, rady, čas a trpezlivosť, ktorú mi venoval. Taktiež by som sa chcel podakovať svojej rodine, ktorá ma motivovala a podporovala počas vypracovania tejto práce, ale aj počas celej doby štúdia.

Název práce: ServIS – webový systém pro firmy zabývající se opravami bagrů

Autor: Milan Truchan

Katedra: Katedra distribuovaných a spolehlivých systémů

Vedoucí bakalářské práce: Mgr. Pavel Ježek, Ph.D., Katedra distribuovaných a spolehlivých systémů

Abstrakt: Cieľom tejto práce bolo vytvoriť softvérové dielo pre malé firmy zaberajúce sa zemnými a výkopovými prácami, opravou a predajom bagrov, ktoré nemajú prístup k vhodnému softvérovému riešeniu pre svoju činnosť.

Potreba a správanie funkcionálít bola prekonzultovaná s majiteľom jednej z týchto firiem.

Vzniknutý softvér predstavuje riešenie problému, je schopný zobraziť ponuku (stroje, prídavné zariadenia) a umožňuje užívateľom dopyt (v podobe emailov) na tieto ponuky. V aplikácii tiež existuje aukcia, kde sa dražia opravené bagre. Užívatelia si tiež môžu v aplikácii vytvoriť účet. Vymenované funkcionality môžu využívať prihlásení aj neprihlásení užívatelia. Bežní prihlásení užívatelia nemusia vyplňovať informácie o sebe vo formulároch pri dopytovaní sa na ponuku. Prihlásení admini majú možnosť spravovať stránku. Tj. pridávať nové, editovať a mazať existujúce ponuky, odpovedať na správy atď.

Title: ServIS – a web system for companies dealing with excavator repairs
Klíčová slova: Informačný systém C# .NET Blazor Server MySQL

Author: Milan Truchan

Department: Department of Distributed and Dependable Systems

Supervisor: Mgr. Pavel Ježek, Ph.D., Department of Distributed and Dependable Systems

Abstract: Abstract.

Keywords: Information system C# .NET Blazor Server MySQL

Obsah

1 Úvod	3
Úvod	3
1.1 Požiadavky na systém	3
1.2 Cieľ práce	7
2 Návrh užívateľského rozhrania	8
2.1 Hlavné rozloženie aplikácie	8
2.2 Modálne okná	9
2.3 Splnenie P2 a P3	9
2.3.1 Hlavná ponuka	10
2.3.2 Vytvorenie novej a úprava existujúcej hlavnej ponuky	11
2.3.3 Ponuka (nových) bagrov	12
2.3.4 Ponuka prídavných zariadení	13
2.3.5 Detail bagra, prídavného zariadenia	14
2.3.6 Dopyt	15
2.3.7 Vytvorenie nového a úprava existujúceho bagra	16
2.3.8 Úprava bagrov, ktoré nie sú v ponuke (nových) bagrov	17
2.3.9 Vytvorenie nového a úprava existujúceho prídavného zariadenia	17
2.4 Splnenie P4	18
2.4.1 Aukčné ponuky	19
2.4.2 Detail aukčnej ponuky	20
2.4.3 Ponúkanie sumy do dražby	21
2.4.4 Vytvorenie novej a úprava existujúcej aukčnej ponuky	22
2.5 Splnenie P8	23
2.6 Splnenie P9	25
2.7 Splnenie P5, P7 a správa predmetov	27
2.7.1 Správy – schránka	28
2.7.2 Správy – konverzácia	30
2.7.3 Správy – nastavenia automaticky generovaných správ	31
2.7.4 Náhradné diely	32
2.7.5 Bagre	32
2.8 Splnenie P6	32
2.8.1 Prihlásovací panel	32
2.8.2 Prihlásenie	33
2.8.3 Registrácia	33
2.8.4 Profil	33
3 Analýza	36
3.1 Výber typu aplikácie	36
3.2 Výber jazyka a frameworku	36
3.2.1 Výber backend jazyka	36
3.2.2 SPA	36
3.2.3 Klasické webové aplikácie	36

3.2.4	Blazor Server	37
3.3	Volba databázy	37
3.3.1	Návrh relačného modelu databázy	37
3.3.2	Volba typu databázy a databázového servera	37
3.3.3	ORM	37
3.4	Aukcia- odpočet a vyhodnocovanie	37
3.5	Volba typu aplikácie, jazyka a frameworku	39
3.6	Návrh systému	39
3.7	Volba databázy	40
3.7.1	Návrh relačného modelu databázy	40
3.7.2	Volba typu databázy a databázového servera	42
3.7.3	ORM	43
3.8	Aukcia- odpočet a vyhodnocovanie	43
3.9	Posielanie a prijímanie správ	44
3.9.1	Automaticky generované správy	45
4	Vývojová dokumentácia	46
4.1	ServISData	46
4.1.1	Koreňový priečinok projektu	46
4.1.2	Attributes	46
4.1.3	DataOperations	46
4.1.4	Interfaces	47
4.1.5	Migrations	47
4.1.6	Models	47
4.2	ServISWebApp	47
4.2.1	Koreňový priečinok projektu	47
4.2.2	Auth	48
4.2.3	BackgroundServices	48
4.2.4	Components	48
4.2.5	CssProviders	49
4.2.6	Pages	49
4.2.7	Resources	49
4.2.8	Shared	49
5	Užívateľská dokumentácia	51
Záver		52
5.1	GDPR	53
5.2	Možné vylepšenia	53
A	Prílohy	55
A.1	Implementácia	55
A.2	Videonávod	55

1. Úvod

V súčasnosti existujú malé firmy, ktoré fungujú ako dodávateľia rôznych drahých produktov. Tieto firmy môžu ponúkať na predaj okrem nových produktov aj staré produkty, ktoré prešli nejakou opravou. Takisto stojí za zmienku, že keďže ide o dodávateľov drahých produktov, tak zákazníci sa najprv s firmou musia dohodnúť na detailoch obchodu, a až potom je možné dodanie produktu.

Spoločným problémom takýchto firiem býva, že ich ľudia nepoznajú. Preto by sa spomínaným firmám hodilo riešenie, ktoré by im umožnilo zviditeľniť ich ponuku produktov. Jednoduché riešenie v podobe statických stránok v tomto prípade nestačí, pretože by neumožnilo dynamicky meniť ponuku danej firmy. Použitie nejakého CMS systému (z ang. content management system), napr. WordPress, takisto nie je optimálnym riešením, pretože vyžaduje znalosť platformy, ktorá nie je samozrejmesťou.

My sme dostali ponuku na tvorbu riešenia od jednej z takýchto firiem. Konkrétnie ide o firmu, ktorá sa zaoberá predajom a opravou bagrov (ďalej už len klientská firma). Po konzultácii s majiteľom (ďalej už len klient) sme zistili, že klientská firma ponúka služby v podobe výkopových prác, predaja a opravy bagrov, a takisto predaja prídavných zariadení pre bagre. Taktiež sme zistili, že doteraz fungovala komunikácia medzi klientskou firmou a zákazníkmi prostredníctvom telefonátov, emailov alebo sa strany fyzicky stretli a dohodli obchod. Klient od nás vyžaduje riešenie, ktoré by splňalo požiadavky uvedené v následujúcej podkapitole.

1.1 Požiadavky na systém

V priebehu niekoľkých stretnutí sme s klientom prebrali a vypracovali následujúce požiadavky, ktoré softvér musí splňať:

- **P1 Roly užívateľa**

Jednou z požiadaviek je, že softvér má rozlišovať zamestnancov firmy spravujúcich systém (ďalej už len administrátori, resp. administrátor) a bežných zákazníkov. Obom rolám sa bude zobrazovať len obsah podľa funkcionality, ktoré majú k dispozícii. Čiže napr. zákazník si bude môcť zobraziť detail bagra a vyjadriť oň nejakým spôsobom záujem, ale nezobrazí sa mu možnosť na jeho vymazanie. V prípade administrátora bude možné bager napr. vymazať, ale nedáva zmysel, aby mohol administrátor vyjadrovať záujem o bager.

- **P2 Predstavenie ponuky zákazníkom**

Ako už bolo spomenuté, klientská firma predáva bagre a prídavné zariadenia pre bagre. Klient preto chce, aby bol softvér schopný prezentovať ponuku firmy (bagre a prídavné zariadenia), pričom hlavná ponuka je tvorená bagrami. Klient vyžaduje, aby po príchode užívateľa na domovskú (úvodnú) stránku sa zobrazila hlavná ponuka.

- **P2.1 Hlavná ponuka**

Hlavná ponuka predstavuje bagre určitého typu (t. j. určitej kombinácie značky a kategórie). Hlavná ponuka obsahuje opis typu bagrov a fotku reprezentujúcu daný typ bagrov. Po rozkliknutí nejakej hlavnej ponuky sa zobrazia bagre typu asociovaného s vybranou ponukou.

– **P2.2 Bager**

Každý bager má obsahovať informácie: názov, značku, kategóriu, opis, fotky a vlastnosti. Tieto informácie majú byť viditeľné pre každého užívateľa, t. j. ako pre bežného zákazníka, tak aj pre administrátora. Navyše má ešte stroj obsahovať informáciu o náhradných dieloch – táto informácia má byť viditeľná iba pre administrátorov. Navyše môžu existovať bagre, ktoré nepatria do ponuky a môžu byť využité výhradne iba v aukcii (viac o aukcii v P4).

– **P2.3 Prídavné zariadenie**

Každé prídavné zariadenie má obsahovať informácie: názov, značka, kategória, pre akú kategóriu strojov je zariadenie určené, opis a fotky. Tieto informácie majú byť viditeľné rovnako pre každého užívateľa.

– **P2.4 Správa bagrov, prídavných zariadení a hlavných ponúk**

Aby mohol administrátor spravovať bagre, prídavné zariadenia ale takisto aj hlavné ponuky podľa potreby, tak je tiež nutné vytvoriť miesto, ktoré mu ich umožní pridať, odstraňovať a editovať.

• **P3 Posielanie dopytu**

Takisto klient od softvéru vyžaduje, aby umožnil zákazníkom objednať si daný produkt alebo službu prostredníctvom emailových správ. Bežnou praxou v tomto odvetví je, že cena strojov sa dopredu neudáva. Zákazník najprv vyjadrí záujem (dopyt), prekonzultujú sa detaily medzi potenciálnym kupcom a firmou, a až potom prebehne obchod. Z tohto dôvodu systém nebude fungovať na princípe ako bežné internetové obchody (tým sa myslí pridávanie do košíka s následnou platbou), ale bude fungovať na princípe posielania správ (dopytov).

– **P3.1 Dopyt**

Dopyt by mal v sebe obsahovať informácie o žiadnom predmete, údaje o užívateľovi, a tiež správu užívateľa. Uživatelskými údajmi sa myslí meno, priezvisko, email – tie sú povinné údaje. A takisto telefónne číslo, mesto – tie sú nepovinné údaje.

• **P4 Aukcia**

Ked sme v predošlých podmienkach spomínali ponuku strojov a prídavných zariadení, tak išlo o nové produkty. No ako už bolo skôr spomenuté, klientská firma sa špecializuje aj na opravu bagrov. Klient vyžaduje, aby mohol administrátor v systéme vytvoriť aukčnú ponuku, do ktorej by okrem špecifikovania jej konca a počiatočnej sumy (vyvolávacej ceny), vedel zaradiť opravený bager, a aby systém umožnil zákazníkom ponúkať sumy (prvá ponúknutá suma môže byť rovná počiatočnej sume, nasledujúce ponúkané sumy musia mať medzi sebou rozdiel aspoň 100 eur), pričom po skončení dražby zákazník s najvyššou ponúknutou sumou vyhráva dražený bager.

– **P4.1 Správanie aukcie**

Kedž aukcia skončí, systém má upozorniť jej účastníkov (poslať email) na to, či vyhrali alebo prehrali dražbu. Taktiež má softvér upozorniť administrátora systému na to, že aukcia skončila a kto je jej víťazom. V prípade, že aukcia skončila bez víťaza (nikto sa jej nezúčastnil), tak má softvér dražbu automaticky reštartovať, posunúť termín konca dražby o týždeň a upozorniť o tom administrátora (poslať mu email).

– **P4.2 Odpočet a ďalšie údaje**

Okrem toho sa od nášho softvéru vyžaduje, aby bol pri každej aukčnej ponuke zobrazený odpočet do konca danej dražby, počet účastníkov, a taktiež aktuálna (najvyššia ponúknutá) suma.

• **P5 Správy**

Kedže posielanie dopytov a správanie aukcie zahŕňa posielanie emailov administrátorom, tak je tiež žiadúce, aby sa emails dali prečítať nielen z Gmailu (emailová služba používaná klientom), ale aj z nášho systému a rovnako aby systém administrátorom umožnil na ne odpovedať.

– **P5.1 Podobnosť s Gmailom**

Nakoľko je klient zvyknutý na prácu s Gmailom, tak sa má schránka podobať na Gmail. Teda aspoň funkcionálou, t. j. pri príchode do schránky sa zobrazia najnovšie správy pre každú konverzáciu (vlákno) zoradené zhora smerom dole od najnovšej po najstaršiu.

Po rozkliknutí nejakej zo správ sa zobrazí celá konverzácia (každá správa vo vybranom vlákne) zoradená zhora dole od najstaršej po najnovšiu.

Ďalej má schránka umožňovať označovanie správ, pričom označené správy budeme môcť hromadne vymazať alebo označiť za prečítané, resp. neprečítané. Ak sú označené správy neprečítané, zobrazí sa tlačidlo umožňujúce označenie vybraných správ ako prečítané, ak sú všetky označené správy prečítané, tak sa zobrazí tlačidlo umožňujúce označiť vybrané správy ako neprečítané, a ak označené správy obsahujú aj prečítané, aj neprečítané správy, tak sa zobrazí tlačidlo umožňujúce označiť vybrané správy ako prečítané.

Podobne po rozkliknutí nejakej zo správ sa nám zobrazí celá konverzácia a administrátor bude môcť celú konverzáciu vymazať alebo označiť za neprečítanú, a taktiež bude môcť odoslať novú správu do konverzácie (odpovedať na správy).

Čo sa týka mazania správ, tak po kliknutí na tlačidlo vymazania správy (resp. správ) stačí ak sa zobrazí potvrdzovacie okno, nie je nutné vytvárať osobitné miesto pre vymazané správy (kôš).

– **P5.2 Prepojenie správy s predmetom**

Okrem toho budeme ešte od softvéru vyžadovať, aby v správach, ktoré boli odoslané z nášho systému, ako napr. dopyt alebo správy z aukcie, tak aby v sebe obsahovali okno, ktoré prepojí správu a vec, ktorej sa

daná správa týka. Teda napríklad ak zákazník odošle dopyt na stroj X, tak po otvorení správy nájde administrátor okrem predmetu a tela správy, takisto nejaký odkaz (prepojenie) odkazujúci na stroj X, ktorým sa dá jednoducho dostať k údajom o stroji X.

- **P5.3 Automaticky generované správy**

Okrem toho je tiež žiadúce, aby systém umožnil administrátorom upravovať formát automaticky odosielaných (generovaných) správ týkajúcich sa aukcie.

- **P6 Registrácia a prihlásenie užívateľov**

Ďalšou požiadavkou je, aby softvér umožnil zákazníkom registrovať sa do systému a následne sa doň prihlásiť. Do systému sa môžu prihlasovať rovnakým spôsobom ako bežní zákazníci aj administrátori. Systém má rozlíšiť, či ide o účet bežného zákazníka alebo o administrátorský (dôležité pre P1). Každý užívateľ má mať po prihlásení výhodu v tom, že do formulárov nemusí zadávať svoje osobné údaje.

- **P6.1 Funkcionality pre neprihlásených užívateľov**

No požiadavkou je takisto aj to, aby aj neprihlásení užívatelia mohli posielat dopyty a účastniť sa aukčných dražieb.

- **P6.2 Registrácia a prihlasovanie užívateľa**

Pri registrácii si bude môcť užívateľ vybrať svoje užívateľské meno, heslo, takisto bude môcť zadat svoje meno, priezvisko a email. Všetky spomenuté údaje sú povinné. Nepovinnými údajmi, ktoré môže užívateľ ďalej zadať, sú telefónne číslo a mesto.

Pri prihlasovaní do systému má užívateľ zadat svoje prihlasovacie meno a heslo.

- **P6.3 Profil užívateľa**

Takisto je nutné vytvoriť profil, kde si užívateľ môže svoje údaje upravovať.

- **P7 Prístup k súčiastkam strojov**

Ked si zákazník zakúpi bager, tak po nejakom čase má firma vykonať kontrolu tohto bagra. Ale predtým než zamestnanci pôjdu vykonať kontrolu si musia zistíť, aké súčiastky obsahuje daný bager. A preto klient od softvéru vyžaduje, aby umožňoval administrátorom zobraziť aké náhradné diely obsahuje konkrétny bager.

- **P7.1 Náhradný diel**

Jeden bager môže obsahovať viacero náhradných dielov (počet nás nezaujíma) a jeden náhradný diel sa môže nachádzať vo viacerých bagroch. Náhradný diel obsahuje informácie: katalógové číslo, názov.

- **P7.2 Správa náhradných dielov**

Taktiež je potrebné vytvoriť časť aplikácie, ktorá administrátorovi umožní náhradné diely pridávať, editovať, mazať a upravovať ich vzťahy s bagrami.

- **P8 Objednávanie výkopových prác**

Klient taktiež vyžaduje časť aplikácie, kde budú opísané služby (presnejšie výkopové práce), ktoré firma poskytuje, a aby užívateľ mohol odtiaľ o dané služby požiadať (odoslať email, v ktorom opíše svoje požiadavky).

- **P9 Sekcie O nás a Kontakt**

Navyše klient žiada časť aplikácie, kde bude opísaná firma a jej história, a takisto časť, kde bude zobrazený kontakt (email, telefónne číslo) na klientskú firmu (príp. jej pridružené firmy). V oboch prípadoch pôjde len o statický text, príp. fotky.

- **P10 Dostupnosť**

Nakoľko klientovi ide o to, aby dostal ponuku viac do povedomia (i potenciálne nových) zákazníkov, je žiadúce, aby bol softvér jednoducho dostupný každému užívateľovi – tým sa myslí, že užívateľ si nemusí stahovať, inštalovať žiadnen softvér, a takisto v prípade administrátorov sa chceme vyhnúť problémom s kompatibilitou (s pozorovaním autor vie, že firmy častokrát používajú staré počítače s potenciálne starým softvérom, čo by mohlo spôsobať problémy s prevádzkou nášho systému).

- **P11 Náklady**

Kedže ide o malú firmu, tak chceme, aby náklady spojené s tvorbou a vedením softvéru boli minimálne alebo v ideálnom prípade žiadne. Konkrétnie sa myslia náklady spojené s potenciálnym využitím softvéru, balíčkov tretích strán alebo databázových serverov atď.

1.2 Cieľ práce

Po hľadaní alternatívnych riešení sa nám nepodarilo nájsť žiadnen už existujúci systém, ktorý by splňal predchádzajúce požiadavky. Podobný systém by si mohol vytvoriť majiteľ firmy sám napr. pomocou WordPressu, ale to by vyžadovalo pokročilejšie znalosti platformy.

Preto je cieľom tejto práce implementovať systém splňajúci požiadavky P1 až P11, určený pre firmy, ktoré sa zaoberajú predajom a opravou bagrov, ktorý by majiteľom firiem umožnil sústrediť sa len na ich doménu (t. j. napr. pridávanie bagrov do ponuky) a neriesiť detaľy implementácie funkcionálít a vzhľadu systému (ako by tomu bolo v prípade WordPressu).

2. Návrh užívateľského rozhrania

V tejto kapitole si prejdeme koncept užívateľského rozhrania aplikácie.

2.1 Hlavné rozloženie aplikácie

Hlavné rozloženie aplikácie má vyzerat tak, že v hornej lavej časti má byť zobrazený názov firmy (má fungovať ako odkaz na domovskú stránku) a vpravo hore má byť zobrazené telefónne číslo firmy (má fungovať ako odkaz na sekciu Kontakt).

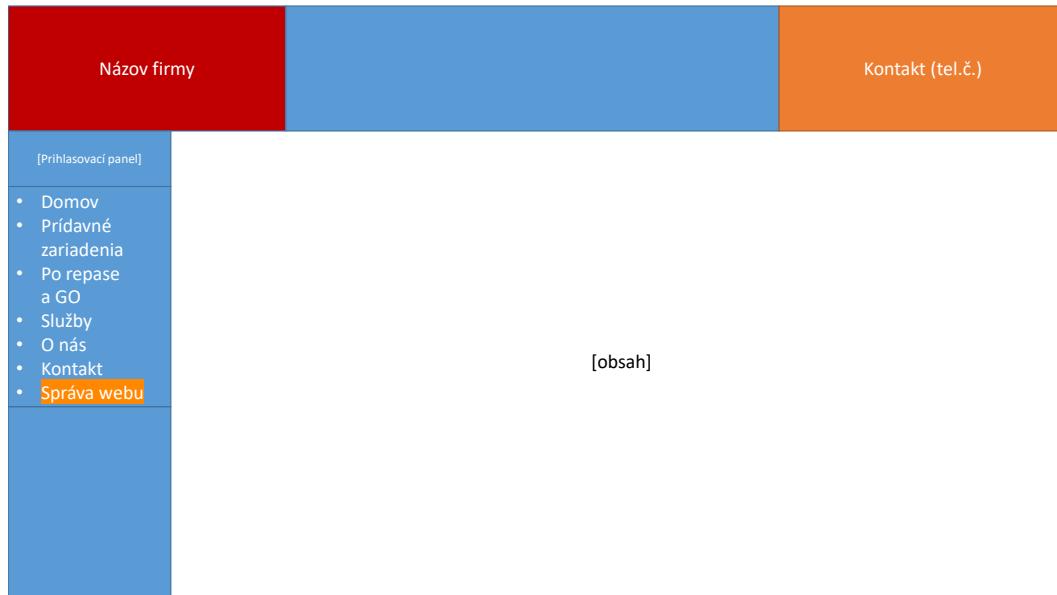
Pod prihlásovacím panelom má byť zobrazená navigácia. V navigácii sa majú nachádzať odkazy „Domov“, „Prídavné zariadenia“, „Po repase a GO“, „Služby“, „O nás“, „Kontakt“ a „Kontakt“. Okrem týchto odkazov sa v navigácii pre administrátorov zobrazuje navyše aj odkaz „Správa webu“.

Podobné správanie, t. j. obmedzovanie alebo pridávanie obsahu na základe roly užívateľa, bude opísané aj v následujúcich podkapitolách a súvisí s požiadavkou P1.

Nad navigáciou sa má nachádzať prihlásovací panel, o ktorom si viac povieme v podkapitole 2.8.

Ďalej v centre aplikácie má byť obsah, ktorý sa má meniť podľa toho v akej časti aplikácie sa užívateľ nachádza.

Pre lepšiu predstavu rozloženia aplikácie vid obrázok 2.1.



Obr. 2.1: Návrh rozloženia aplikácie

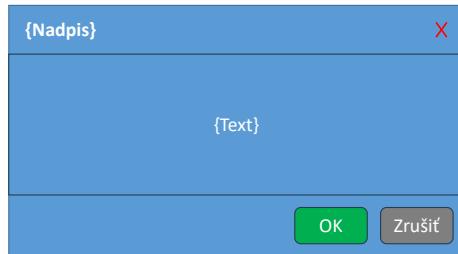
Všetky obrázky v následujúcich podkapitolách (s výnimkou grafov prechádzania medzi časťami aplikácie a modálnym oknom) budú predstavovať obsahovú časť ak sa explicitne nepovie inak.

2.2 Modálne okná

V následujúcich podkapitolách bude v spojení s mazaním položiek viackrát spomenuté modálne potvrzovacie okno.

Okno má obsahovať vo vrchnej časti nadpis a v centre text oznamujúci akciu. V spodnej časti sa majú nachádzať tlačidlá „OK“ (pre potvrdenie vymazania) a „Zrušiť“ (pre zrušenie mazania).

Pre lepšiu predstavu vid' obr. 2.2.



Obr. 2.2: Potvrzovacie modálne okno.

2.3 Splnenie P2 a P3

V tejto podkapitole si prejdeme časti aplikácie splňujúce požiadavky P2 a P3, t. j. predstavenie hlavnej ponuky, bagrov, prídavných zariadení, a takisto si z časti ukážeme ako môže administrátor jednotlivé položky vytvárať, upravovať a vymazovať (z časti preto, lebo v prípade bagrov bude existovať aj iný spôsob, o ktorom si povieme neskôr).

Po príchode na stránku (alebo po kliknutí na odkaz „Domov“ v navigácii) má byť užívateľovi zobrazená sekcia Domov s vylistovanými kartami hlavnej ponuky.

Administrátorovi sa na každej z kariet má zobrazovať tlačidlo, ktoré ho po kliknutí presmeruje do časti aplikácie, kde môže danú ponuku upravovať. Takisto sa v sekcií Domov nachádza nad kartami odkaz, ktorý má po kliknutí administrátora presmerovať do časti aplikácie, kde bude môcť vytvoriť novú hlavnú ponuku. Na každej karte hlavnej ponuky má byť tlačidlo „Zobraziť“, ktoré má užívateľa presmerovať k vylistovaným kartám (nových) bagrov.

Kliknutím na nejakú z kariet bagrov má byť užívateľ presmerovaný do časti aplikácie zobrazujúcej detail vybraného bagra – odtiaľ sa má byť (tentokrát už len) administrátor schopný dostať kliknutím na tlačidlo „Upraviť“ do časti aplikácie umožňujúcej upravovanie daného bagra.

Podobne má byť užívateľ schopný kliknúť na odkaz „Prídavné zariadenia“ v navigácii, ktorý ho presmeruje na vylistované karty prídavných zariadení. Ak užívateľ klikne na nejakú z kariet, má byť presmerovaný na detail vybraného prídavného zariadenia. Odtiaľ sa má byť (tentokrát už len) administrátor schopný dostať kliknutím na tlačidlo „Upraviť“ do časti aplikácie umožňujúcej upravovanie daného prídavného zariadenia.

Pre lepšie pochopenie prechádzania medzi jednotlivými časťami programu vid obr. 2.3.



Obr. 2.3: Prechádzanie medzi časťami aplikácie spĺňajúcimi požiadavky P2 a P3.

2.3.1 Hlavná ponuka

Po príchode na stránku sa užívateľ ocitne na domovskej stránke, kde sú zoobrazené karty hlavných ponúk (vid obr. 2.4).



Obr. 2.4: Hlavná ponuka (úvodná stránka)

Každá z kariet má zobrazovať fotku reprezentujúcu danú ponuku (resp. typ bagra). Ak užívateľ prejde kurzorom na kartu, tak sa má namiesto fotky zobraziť text opisujúci danú ponuku.

Ďalej majú byť len pre administrátorov zobrazené tlačidlá „E“ a „X“ vo vrchných rohoch kariet, a takisto nad vylistovanými kartami odkaz „Pridať novú ponuku“.

Po kliknutí na tlačidlo „E“ nejakej z hlavných ponúk má byť administrátor presmerovaný na formulár vyplnený údajmi danej hlavnej ponuky, kde bude môcť túto ponuku upravovať.

Tlačidlo „X“ má slúžiť na vymazanie danej hlavnej ponuky, po kliknutí naň sa má zobraziť modálne potvrzovacie okno s nadpisom „Vymazať ponuku natrvalo“ a textom „Na ozaj chcete túto ponuku vymazať natrvalo?“.

Ďalej po kliknutí na odkaz „Pridať novú ponuku“ má byť administrátor presmerovaný na prázdný formulár, kde bude môcť vytvoriť novú hlavnú ponuku.

Ďalej každá z kariet má obsahovať tlačidlo „Zobrazit“, ktoré má po kliknutí užívateľa presmerovať k ponuke (nových) bagrov typu, ktorý prezentuje daná hlavná ponuka.

2.3.2 Vytvorenie novej a úprava existujúcej hlavnej ponuky

Ako bolo v predošej časti spomenuté, administrátor má byť schopný z domovskej stránky (časť aplikácie kde majú byť vylistované hlavné ponuky) kliknutím na odkaz „Pridať novú ponuku“ vytvoriť novú hlavnú ponuku a kliknutím na tlačidlo „E“ existujúcu hlavnú ponuku upravovať.

Či už v prípade vytvárania novej, alebo upravovania existujúcej hlavnej ponuky, má byť administrátor presmerovaný na časť aplikácie s formulárom umožňujúcim vložiť jednu fotku, vybrať z možností typov bagrov, a takisto napísat popis hlavnej ponuky. Všetky údaje až na popis sú povinné. Navyše po vložení sa majú fotka a jej názov zobraziť vo formulári.

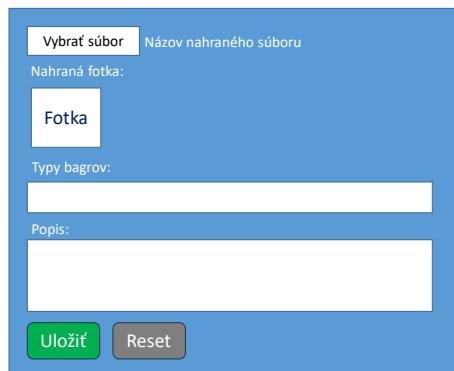
Formulár má obsahovať tlačidlá „Uložiť“ (na uloženie hlavnej ponuky) a tlačidlo „Reset“ (na vyprázdenie formulára).

Ak sú po kliknutí na tlačidlo „Uložiť“ povinné údaje nevyplnené, tak má na to systém administrátora upozorniť prostredníctvom chybových správ pri jednotlivých poliach.

Navyše nad formulárom má byť v prípade vytvárania novej hlavnej ponuky nadpis „Hlavné ponuky – vytvorenie nového záznamu“ a v prípade úpravy existujúcej hlavnej ponuky má byť nadpis „Hlavné ponuky – úprava existujúceho záznamu“.

Pre lepšiu predstavu viď obr. 2.5.

Hlavné ponuky- vytvorenie nového záznamu



Obr. 2.5: Časť aplikácie pre vytvorenie hlavnej ponuky.

2.3.3 Ponuka (nových) bagrov

Po kliknutí na tlačidlo „Zobrazit“ nejakej z hlavných ponúk sa užívateľovi majú vylistovať karty strojov typu, ktorý prezentovala vybraná hlavná ponuka (viď obr. 2.6). Medzi vylistovanými bagrami nemajú byť bagre určené iba pre aukciu.

[Pridať nový bager tohto typu](#)

Kategória bagrov- značka bagrov



Obr. 2.6: Ponuka bagrov.

Nad kartami bagrov sa má nachádzať nadpis v tvare „{Kategória vybraného typu bagrov} – {Značka vybraného typu bagrov}“ (typ bagra určuje jeho kategóriu a značku).

Okrem nadpisu sa má vo vrchnej časti nachádzať aj odkaz „Pridať nový bager“, ktorý má užívateľa po kliknutí presmerovať k formuláru pre vytvorenie nového stroja. Ak sa užívateľ dostal k formuláru touto cestou, má mať vo formulári predvyplnený typ stroja podľa typu vylistovaných strojov.

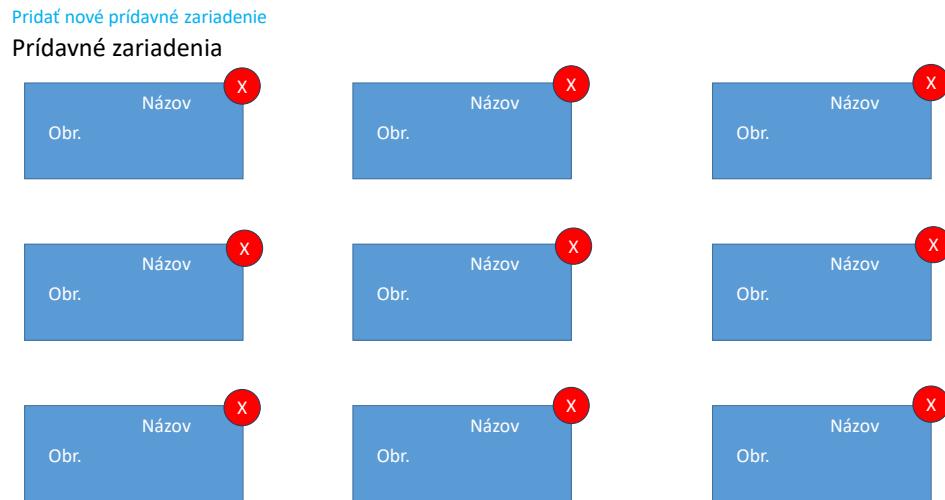
Každá z kariet má mať v pravom hornom rohu tlačidlo „X“, ktoré umožní daný stroj vymazať. Po kliknutí naň sa má zobraziť modálne potvrzovacie okno s nadpisom „Vymazať bager natrvalo“ a textom „Na ozaj chcete tento bager vymazať natrvalo?“.

Tlačidlo „X“ aj odkaz „Pridať nový bager“ majú byť viditeľné iba pre administrátorov.

Po kliknutí na nejakú z kariet má byť užívateľ presmerovaný na detail bagra.

2.3.4 Ponuka prídavných zariadení

Po kliknutí na „Prídavné zariadenia“ v navigácii (viď vľavo na obr. 2.1) sa užívateľovi majú vylistovať karty prídavných zariadení (viď obr. 2.7).



Obr. 2.7: Ponuka prídavných zariadení

Nad kartami prídavných zariadení sa má nachádzať nadpis „Prídavné zariadenia“.

Okrem nadpisu sa má vo vrchnej časti nachádzať aj odkaz „Pridať nové prídavné zariadenie“, viditeľný iba pre administrátora, ktorý ho má po kliknutí presmerovať k formuláru pre vytvorenie nového prídavného zariadenia.

Každá z kariet má mať v pravom hornom rohu tlačidlo „X“, takisto viditeľné iba pre administrátora, ktoré má umožniť dané prídavné zariadenie vymazať. Po kliknutí na tlačidlo „X“ sa má zobraziť modálne potvrzovacie okno s nadpisom „Vymazať prídavné zariadenie natrvalo“ a textom „Na ozaj chcete toto prídavné zariadenie vymazať natrvalo?“.

Po kliknutí na nejakú z kariet má byť užívateľ presmerovaný na detail prídavného zariadenia.

2.3.5 Detail bagra, prídavného zariadenia

Po kliknutí na nejakú z kariet bagra alebo prídavného zariadenia sa má užívateľovi zobraziť stránka s detailom vybraného predmetu (viď obr. 2.8).

Obr. 2.8: Detail bagra/prídavného zariadenia.

V tejto časti sa majú nachádzať fotky predmetu (klikaním na tlačidlá šipiek vpravo, vľavo sa má prechádzať medzi fotkami predmetu). Ďalej sa má na stránke nachádzať opis predmetu, a takisto tabuľka s dodatočnými informáciami – v prípade bagra to majú byť jeho vlastnosti (napr. výška, šírka atď.), v prípade prídavného zariadenia to majú byť údaje o tom, akej značky, kategórie, a taktiež pre akú kategóriu bagrov je dané prídavné zariadenie.

Okrem toho sa má v tejto časti ešte nachádzať tlačidlo „Dopyt“. Po kliknutí naň sa má zobraziť formulár, ktorým bude môcť užívateľ odoslať dopyt na daný predmet. Po opäťovnom kliknutí na tlačidlo sa má formulár zatvoriť. Tlačidlo (a ani formulár) sa nemajú zobrazovať administrátorom.

V hornej časti sa má nachádzať tlačidlo „Upraviť“, ktoré má byť viditeľné iba pre administrátorov. Po kliknutí naň má byť administrátor presmerovaný na formulár vybraného bagra (resp. prídavného zariadenia). Pomocou formulára bude môcť administrátor daný bager (resp. prídavné zariadenie) upravovať.

2.3.6 Dopyt

V predošej časti textu bolo spomenuté, že v detaile bagra alebo prídavného zariadenia sa má nachádzať tlačidlo „Dopyt“, ktorým si bude môcť zákazník (nie administrátor) zobraziť (a opäťovným kliknutím schovať) formulár pre odosielanie dopytu na daný predmet.

Formulár má pre neprihláseného zákazníka obsahovať povinné polia meno, priezvisko, email, správa, a takisto nepovinné polia telefón, mesto. Pre prihláseného zákazníka má formulár obsahovať iba povinné pole správa. Povinné polia

majú byť označené hviezdičkou.

Okrem polí má v sebe formulár obsahovať pre neprihlásených zákazníkov správu upozorňujúcu na to, že prihlásený užívateľ nemusí vyplňať svoje osobné údaje. Takisto by mal formulár pre neprihláseného užívateľa obsahovať upozornenie, že s odoslaním správy súhlasí s podmienkami a zásadami aplikácie.

Dalej má formulár obsahovať tlačidlo „Odoslať“, ktoré má zákazníkovi umožniť odoslanie dopytu.

Ak sú po kliknutí na tlačidlo „Odoslať“ povinné údaje nevyplnené, tak má na to aplikácia užívateľa upozorniť chybovými správami pri jednotlivých poliach.

Pre lepšiu predstavu formulára viď pravú časť obrázka 2.8.

2.3.7 Vytvorenie nového a úprava existujúceho bagra

Kliknutím na odkaz „Pridať nový bager“, ktorý sa má nachádzať v časti s vylistovanými (novými) bagrami, má byť administrátor presmerovaný do časti aplikácie, kde bude môcť vytvoriť nový bager.

Podobne kliknutím na tlačidlo „Upraviť“, ktoré sa má nachádzať v časti s detailom bagra, má byť administrátor presmerovaný do časti aplikácie, kde bude môcť existujúci bager upravovať.

Obe časti majú obsahovať formulár umožňujúci vloženie viacerých fotiek, názvu bagra, zadanie jeho popisu, uvedenie či je stroj určený iba pre aukciu, tiež bude môcť administrátor vybrať, ktoré náhradné diely patria do daného bagra, a takisto vybrať z možností typov bagrov. Podľa vybraného typu bagra sa majú zobraziť vlastnosti ním určené a administrátor bude môcť vyplniť ich hodnoty. Fotky, názov bagra a jeho typ sú povinné údaje.

Naďalej po vložení sa majú fotky a ich počet zobraziť vo formulári. V pravom hornom rohu každej zobrazenej fotky sa má nachádzať tlačidlo „X“, ktorým môže administrátor fotku odstrániť.

Formulár taktiež obsahuje tlačidlá „Uložiť“ (na uloženie bagra) a „Reset“ (na vyprázdenie formulára).

Ak sú po kliknutí na tlačidlo „Uložiť“ povinné údaje nevyplnené, tak má na to systém administrátora upozorniť prostredníctvom chybových správ pri jednotlivých poliach.

Naďalej nad formulárom má byť v prípade vytvárania nového bagra nadpis „Bagre – vytvorenie nového záznamu“ a v prípade úpravy existujúceho bagra má byť nadpis „Bagre – úprava existujúceho záznamu“.

Pre lepšiu predstavu viď obr. 2.9.

Bagre- vytvorenie nového záznamu

Výbrať súbory	Počet súborov: 2	
Nahráne fotky:		
Fotka	Fotka	
Názov:		
Popis:		
<input type="checkbox"/> Je IBA pre aukciu		
Náhradné diely v bagri:		
Vybrať	Katalógové číslo	Názov
<input type="checkbox"/>
Typ bagrov:		
Špecifické vlastnosti:		
Názov vlastnosti	Hodnota	
...		
...		
Uložiť Reset		

Obr. 2.9: Časť aplikácie pre vytvorenie bagra.

2.3.8 Úprava bagrov, ktoré nie sú v ponuke (nových) bagrov

V predošлом texte bolo napísané, že administrátor sa bude môcť cez karty hlavných ponúk dostať k ponuke (nových) bagrov a cez nich k detailu bagra, kde môže kliknutím na tlačidlo „Upraviť“ upravovať daný bager. Ale ak by administrátor vytvoril bager typu, pre ktorý neexistuje hlavná ponuka alebo bager určený iba pre aukciu, tak by sa bager nezobrazoval medzi ponukou (nových) bagrov.

Preto bude pre takéto bagre (a tiež pre položky s podobným problémom) vyhradená osobitná časť aplikácie, kde ich bude môcť administrátor spravovať (t. j. vytvárať, upravovať a mazať). Viac o tejto časti aplikácie v podkapitole 2.7.

2.3.9 Vytvorenie nového a úprava existujúceho prídavného zariadenia

Kliknutím na odkaz „Pridať nové prídavné zariadenie“, ktorý sa má nachádzať v časti s vylistovanými prídavnými zariadeniami, má byť administrátor presmerovaný do časti aplikácie, kde bude môcť vytvoriť nové prídavné zariadenie.

Podobne kliknutím na tlačidlo „Upraviť“, ktoré sa má nachádzať v časti s detailom prídavného zariadenia, má byť administrátor presmerovaný do časti aplikácie, kde bude môcť existujúce prídavné zariadenie upravovať.

Obe časti majú obsahovať formulár umožňujúci vloženie viacerých fotiek, vybranie kategórie bagrov (pre ktoré je prídavné zariadenie určené), vybranie jeho

kategórie, vybranie značky, a takisto zadanie názvu a popisu prídavného zariadenia. Okrem popisu sú všetky údaje povinné.

Navyše po vložení sa majú fotky a ich počet zobrazí vo formulári. V pravom hornom rohu každej zobrazenej fotky sa má nachádzať tlačidlo „X“, ktorým môže administrátor fotku odstrániť.

Formulár taktiež obsahuje tlačidlá „Uložiť“ (na uloženie prídavného zariadenia) a „Reset“ (na vyprázdenie formulára).

Ak sú po kliknutí na tlačidlo „Uložiť“ povinné údaje nevyplnené, tak má na to systém administrátora upozorniť prostredníctvom chybových správ pri jednotlivých poliach.

Navyše nad formulárom má byť v prípade vytvárania nového bagra nadpis „Prídavné zariadenia – vytvorenie nového záznamu“ a v prípade úpravy existujúceho prídavného zariadenia má byť nadpis „Prídavné zariadenia – úprava existujúceho záznamu“.

Pre lepšiu predstavu vid' obr. 2.10.



Obr. 2.10: Časť aplikácie pre vytvorenie prídavného zariadenia.

2.4 Splnenie P4

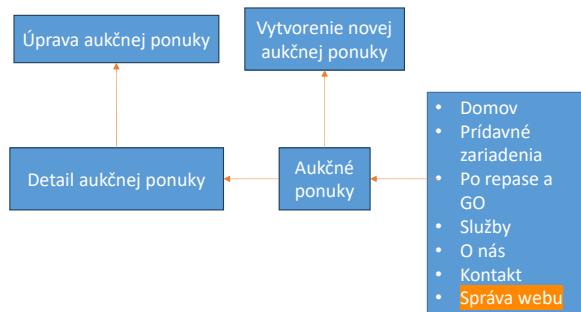
V tejto podkapitole si prejdeme časti aplikácie splňujúce požiadavku P4, t. j. predstavenie aukčnej ponuky, časti umožňujúcej administrátorovi vytvárať (a upravovať) aukčné ponuky, a takisto časti umožňujúce zákazníkom ponúkať sumy do dražby.

Po kliknutí na odkaz „Po repase a GO“ v navigácii má byť užívateľovi zobrazená časť aplikácie s vylistovanými kartami aukčnej ponuky. Nad kartami

(tentokrát už iba) administrátorovi sa má zobrazovať odkaz „Pridať novú aukčnú ponuku“, ktorým sa bude môcť dostať do časti aplikácie s prázdnym formulárom pre vytvorenie novej aukčnej ponuky.

Po kliknutí na nejakú z kariet sa má užívateľovi zobraziť detail aukčnej ponuky. V časti detailu sa má iba administrátorovi zobrazovať tlačidlo „Upraviť“, ktorým sa bude môcť dostať do časti aplikácie, kde môže danú ponuku upravovať a iba zákazníkovi tlačidlo „Mám záujem“, ktorým si môže zobraziť formulár pre ponúknutie sumy do dražby.

Pre lepšie pochopenie prechádzania medzi jednotlivými časťami programu vidí obr. 2.11.



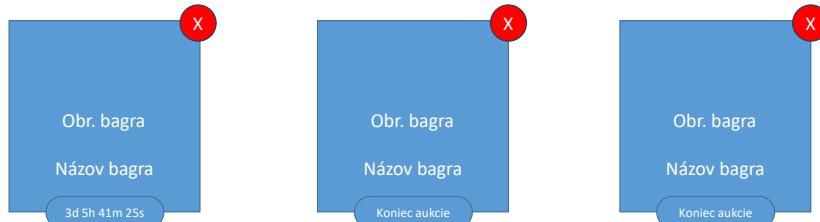
Obr. 2.11: Prechádzanie medzi časťami aplikácie splňajúcimi požiadavku P4.

2.4.1 Aukčné ponuky

Kliknutím na odkaz „Po repase a GO“ v navigácii má byť užívateľ presmerovaný do časti aplikácie s vylistovanými kartami aukčných ponúk (viď obr. 2.12).

[Pridať novú aukčnú ponuku](#)

Aukcia bagrov po repase a generálnej oprave



Obr. 2.12: Aukčné ponuky.

Karta aukčnej ponuky má obsahovať obrázok a názov draženého bagra. V spodnej časti sa má nachádzať odpočet do konca dražby. Ďalej v pravej hornej časti karty má byť tlačidlo „X“, viditeľné iba pre administrátora, ktorým môže ponuku vymazať. Po kliknutí naň sa má zobraziť modálne potvrzovacie okno s nadpisom „Vymazať aukčnú ponuku natrvalo“ a textom „Na ozaj chcete túto aukčnú ponuku vymazať natrvalo?“.

Nad kartami sa má nachádzať nadpis „Aukcia bagrov po repase a generálnej oprave“, a takisto sa má vo vrchnej časti nachádzať aj odkaz „Pridať novú aukčnú ponuku“, viditeľný iba pre administrátorov, ktorý po kliknutí presmeruje administrátora do časti aplikácie, kde môže vytvoriť novú aukčnú ponuku.

Po kliknutí na nejakú z kariet má byť užívateľ presmerovaný na detail vybranej aukčnej ponuky.

2.4.2 Detail aukčnej ponuky

Po kliknutí na nejakú z kariet aukčnej ponuky má byť užívateľ presmerovaný do časti detailu aukčnej ponuky (vid. obr. 2.13).

The screenshot shows a user interface for managing an auction item. On the left, there's a panel for editing the auction. It includes a section for photos ('Fotky/y') with navigation arrows, a text area ('TEXT (zákl. info o aukčnej ponuke)'), and a button ('Mám záujem!'). Below these is a general description ('Opis bagra'). At the bottom is a table ('Tabuľka s dodatočnými informáciami o bagri (vlastnosti bagra)'). A yellow arrow points from the 'Dopyt' button in the table back to the 'Mám záujem!' button. On the right, there's a detailed bidding form with fields for name, surname, phone number, email, city, and amount. It also includes a note about accepting terms and conditions, and a 'Ponúknut' button.

Obr. 2.13: Detail aukčnej ponuky.

V tejto časti sa majú nachádzať fotky draženého bagra (klikaním na tlačidlá šípiek vpravo, vľavo sa má prechádzať medzi fotkami predmetu). Ďalej sa majú v tejto časti aplikácie nachádzať základné informácie o aukčnej ponuke bagra (napr. aký je bager starý, čo na ňom bolo opravované atď.), okrem toho by mal byť v tejto časti aj všeobecný popis bagra (všeobecné informácie o bagri, ktoré platia nezávislý od toho, že bol opravovaný), ďalej by sa v tejto časti mala nachádzať tabuľka s vlastnosťami bagra (napr. výška, šírka atď.).

Okrem toho sa má v tejto časti ešte nachádzať tlačidlo „Mám záujem“. Po kliknutí naň sa má zobraziť formulár, ktorým bude môcť užívateľ ponúknutú sumu do dražby. Po opäťovnom kliknutí na tlačidlo sa má formulár zatvoriť.

Tlačidlo (a ani formulár) sa nemajú zobrazovať administrátorom.

V hornej časti sa má nachádzať tlačidlo „Upraviť“, ktoré má byť viditeľné iba pre administrátorov. Po kliknutí naň má byť administrátor presmerovaný na formulár danej aukčnej ponuky, ktorým ju môže upravovať. Okrem tlačidla „Upraviť“ sa v hornej časti má nachádzať aj tlačidlo „Zmazat“, takisto viditeľné iba pre administrátora, ktorým vie danú aukčnú ponuku zmazať.

2.4.3 Ponúkanie sumy do dražby

V predošej časti textu bolo spomenuté, že v detaile bagra alebo prídavného zariadenia sa má nachádzať tlačidlo „Mám záujem“, ktorým si bude môcť zákazník (nie administrátor) zobraziť (a opäťovným kliknutím schovať) formulár pre ponúknutie sumy do dražby daného bagra.

Formulár má pre neprihláseného zákazníka obsahovať povinné polia meno,

priezvisko, email, ponúkaná suma, a takisto nepovinné polia telefón, mesto. Pre prihláseného zákazníka má formulár obsahovať iba povinné pole ponúkaná suma. Povinné polia majú byť označené hviezdičkou. Navýše má platiť, že prvá zákazníkom ponúknutá suma môže byť rovná počiatočnej sume, ale nasledujúce ponúkané sumy musia mať medzi sebou rozdiel aspoň 100 eur.

Okrem polí má v sebe formulár obsahovať pre neprihlásených zákazníkov správu upozorňujúcu na to, že prihlásený užívateľ nemusí vyplňať svoje osobné údaje. Takisto by mal formulár pre neprihláseného užívateľa obsahovať upozornenie, že s odoslaním správy súhlasy s podmienkami a zásadami aplikácie.

Dalej má formulár obsahovať tlačidlo „Ponúknut“, ktoré má zákazníkovi umožniť ponúknutie sumy do dražby o daný bager.

Ak sú po kliknutí na tlačidlo „Ponúknut“ povinné údaje nevyplnené alebo sú ich hodnoty neplatné, tak má na to aplikácia užívateľa upozorniť chybovými správami pri jednotlivých poliach.

Pre lepšiu predstavu formulára viď pravú časť obrázka 2.13.

2.4.4 Vytvorenie novej a úprava existujúcej aukčnej ponuky

Kliknutím na odkaz „Pridať novú aukčnú ponuku“, ktorý sa má nachádzať v časti s vylistovanými aukčnými ponukami, má byť administrátor presmerovaný do časti aplikácie, kde bude môcť vytvoriť novú aukčnú ponuku.

Podobne kliknutím na tlačidlo „Upraviť“, ktoré sa má nachádzať v časti s detailom aukčnej ponuky, má byť administrátor presmerovaný do časti aplikácie, kde bude môcť danú aukčnú ponuku upravovať.

Časť aplikácie pre vytváranie novej aukčnej ponuky, ale takisto aj časť pre upravovanie existujúcej aukčnej ponuky, majú obsahovať formulár umožňujúci vloženie počiatočnej (vyvolávacej) ceny draženého bagra, dátumu konca dražby, popisu (napr. aké časti boli v bagri opravované), a takisto by mal formulár obsahovať možnosť výberu stroja do dražby. Údaj o type bagra je povinný, dátum konca dražby nesmie obsahovať dátum v minulosti a počiatočná cena nesmie byť záporná.

Formulár má taktiež obsahovať tlačidlo „Reset“ na vyprázdenie formulára.

Okrem tlačidla „Reset“ má obsahovať aj tlačidlo „Uložit“ (na uloženie aukčnej ponuky) ak sa vytvára nová aukčná ponuka alebo upravuje existujúca aukčná ponuka, ktorá je ešte stále aktívna (t. j. stále prebieha dražba). V prípade, že administrátor upravuje existujúcu aukčnú ponuku, ktorá už nie je aktívna, pretože stroj bol vydražený, tak sa má zobraziť tlačidlo „Uložit a spustiť odznova“. Toto správanie je vyžadované z toho dôvodu, že ak výherca dražby z nejakého dôvodu nebude súhlasiť s prevzatím bagra, tak administrátor bude môcť celú dražbu odštartovať znova kliknutím na tlačidlo „Uložit a spustiť odznova“.

Ak sú pri ukladaní povinné údaje nevyplnené alebo sú ich hodnoty neplatné, tak má na to systém administrátora upozorniť prostredníctvom chybových správ pri jednotlivých poliach.

Navýše nad formulárom má byť v prípade vytvárania novej aukčnej ponuky nadpis „Aukčné ponuky – vytvorenie nového záznamu“ a v prípade úpravy existujúcej aukčnej ponuky má byť nadpis „Aukčné ponuky – úprava existujúceho záznamu“.

Pre lepšiu predstavu vidľ obr. 2.14.

Aukčné ponuky- vytvorenie nového záznamu

Počiatočná cena:	Koniec akcie:
Popis:	
Stroje:	
<input type="button" value="Uložit"/>	<input type="button" value="Reset"/>

Obr. 2.14: Časť aplikácie pre vytvorenie aukčnej ponuky.

2.5 Splnenie P8

Po kliknutí na odkaz „Služby“ v navigácii má byť užívateľ presmerovaný do časti aplikácie, kde budú opísané výkopové služby, ktoré firma ponúka, a tiež sa má v tejto časti nachádzať tlačidlo „Mám záujem o službu“, ktoré nemá byť viditeľné pre administrátorov. Pre lepšiu predstavu prechodu medzi časťami aplikácie vidľ obr. 2.15.



Obr. 2.15: Prechádzanie medzi časťami aplikácie splňajúcimi požiadavku P8.

Kliknutím na tlačidlo „Mám záujem o službu!“ sa má zobrazit formulár, ktorým bude môcť užívateľ požiadať o výkopovú službu. Opäťovným kliknutím na tlačidlo sa má formulár schovať.

Formulár má pre neprihláseného zákazníka obsahovať povinné polia meno, priezvisko, email, správa, a takisto nepovinné polia telefón, mesto. Pre prihláseného zákazníka má formulár obsahovať iba povinné pole správa. Povinné polia majú byť označené hviezdičkou.

Okrem polí má v sebe formulár obsahovať pre neprihlásených zákazníkov správu upozorňujúcu na to, že prihlásený užívateľ nemusí vyplňať svoje osobné údaje. Takisto by mal formulár pre neprihláseného užívateľa obsahovať upozornenie, že s odoslaním správy súhlasí s podmienkami a zásadami aplikácie.

Ďalej má formulár obsahovať tlačidlo „Odoslať“, ktoré má zákazníkovi umožniť odoslanie žiadosti o službu.

Ak sú po kliknutí na tlačidlo „Odoslať“ povinné údaje nevyplnené, tak má na to aplikácia užívateľa upozorniť chybovými správami pri jednotlivých poliach.

Pre lepšiu predstavu opisovanej časti aplikácie a formulára viď obr. 2.16.

The diagram illustrates a user interface flow. On the left, a large white box labeled "Služby" contains a blue rectangular area with the word "TEXT". Below this is a blue button with the text "Mám záujem o službu!". A yellow arrow points from this button to the right, leading to a second, smaller blue box. This second box contains a message "Po kliknutí na tlačidlo „Mám záujem o službu!“" above a form. The form includes fields for "Meno*" and "Priezvisko*", both with input boxes. It also has fields for "Telefónne číslo:" and "E-mail*", each with an input box. There is a field for "Mesto:" with an input box, and a larger field for "Správa*" with an input box. At the bottom of the form is a note: "Odoslaním správy súhlasíte s podmienkami a zásadami systému." followed by a blue "Odoslať" button.

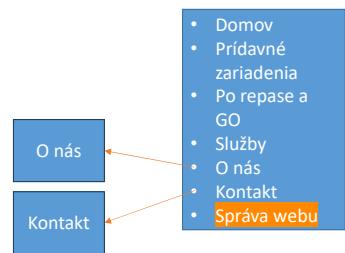
Obr. 2.16: Sekcia Služby.

2.6 Splnenie P9

Po kliknutí na odkaz „O nás“ v navigácii má byť užívateľ presmerovaný do časti aplikácie, ktorá obsahuje text o firme (napr. jej história), príp. nejaké fotky.

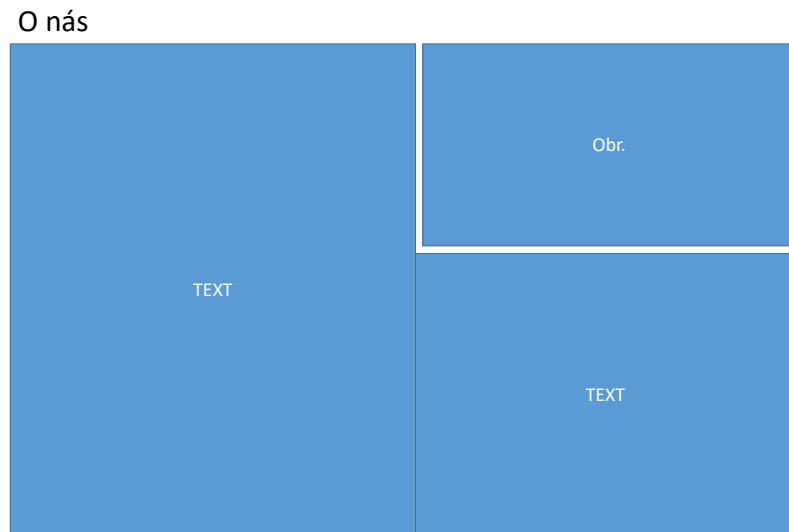
Po kliknutí na odkaz „Kontakt“ v navigácii alebo po kliknutí na telefónne číslo v hornej časti aplikácie (vid vpravo hore na obr. 2.1) má byť užívateľ presmerovaný do časti aplikácie, ktorá obsahuje tabuľku s kontaktom (telefónne číslo a emailová adresa) firmy, príp. aj iných pridružených firiem.

Pre lepšiu predstavu prechodu medzi jednotlivými časťami aplikácie viď obr. 2.17.■



Obr. 2.17: Prechádzanie medzi časťami aplikácie splňajúcimi požiadavku P9.

Pre lepšiu predstavu sekcie „O nás“ vid obr. 2.18.



Obr. 2.18: Sekcia O nás.

Pre lepšiu predstavu sekcie „Kontakt“ vid obr. 2.19.

Kontakt		
Firma	Telefón	Email
Klientská firma	telefón	email
Pridružená firma	telefón	email
...

Obr. 2.19: Sekcia Kontakt.

2.7 Splnenie P5, P7 a správa predmetov

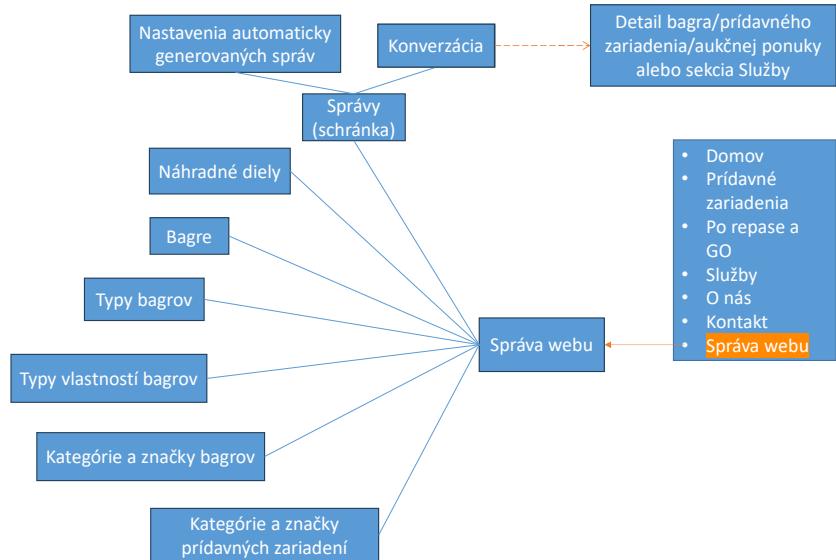
Po kliknutí na odkaz „Správa webu“ v navigácii, ktorý je viditeľný iba pre administrátorov, má byť administrátor presmerovaný do časti aplikácie, ktorá vyzierá následovne: v hornej časti sa má nachádzať panel s kartami, ktorými sa bude môcť administrátor preklikávať, a pod ním sa nachádza obsah podľa toho, na akej karte sa administrátor nachádza.

Po príchode do tejto časti sa má administrátorovi zobraziť prvá karta – „Správy“. Okrem tejto karty sa bude môcť administrátor prekliknúť na karty: „Náhradné diely“, „Bagre“, „Typy bagrov“, „Typy vlastností bagrov“, „Kategórie a značky bagrov“, „Kategórie a značky prídavných zariadení“.

Na karte „Správy“ majú byť vylistované konverzácie. Po kliknutí na nejakú z konverzácií sa má administrátorovi zobraziť konverzácia. Ak bola konverzácia odoslaná z našeho systému, má byť v tejto konverzáции vložené prepojenie na časť aplikácie z ktorej bola správa odoslaná. Môže to byť budď detail bagra, detail prídavného zariadenia, detail aukčnej ponuky alebo sekcia Služby.

Okrem toho sa má byť administrátor schopný dostať z karty „Správy“ (pomocou tlačidla „Nastavenia“, ktoré sa má na karte nachádzať) do časti aplikácie s nastaveniami automaticky generovaných správ.

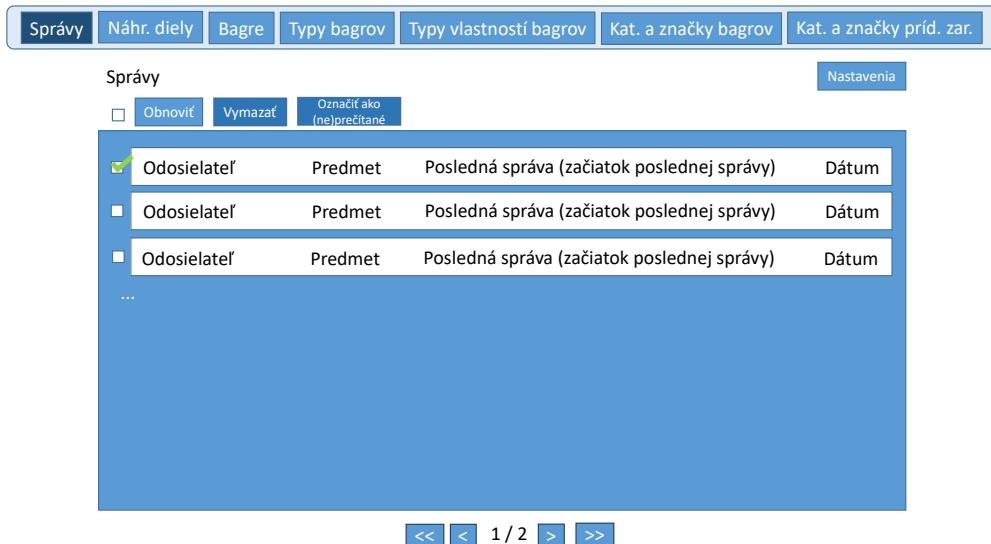
Pre lepšiu predstavu prechodu medzi stránkami a kartami vid obr. 2.20.



Obr. 2.20: Prechádzanie medzi časťami aplikácie splňajúcimi požiadavku P5 (orientovaná hrana značí prechod do novej časti aplikácie, neorientovaná prepínanie medzi časťami, prerušovaná nepovinný prechod do novej časti).

2.7.1 Správy – schránka

Po príchode do sekcie „Správa webu“ alebo po kliknutí na kartu „Správy“ v sekcií „Správa webu“ sa má administrátorovi zobraziť emailová schránka podobná Gmailu (viď obr. 2.21).



Obr. 2.21: Emailová schránka

V schránke majú byť vylistované konverzácie (vlákna správ) zhora dole od najnovšej po+najstrašiu. Každá takáto konverzácia má byť reprezentovaná riadkom, v ktorom je uvedené meno odosielateľa, predmet, a text poslednej správy (resp. jeho začiatok v prípade dlhšieho textu) a dátum poslednej správy (emailu v konverzácií).

Vylistované konverzácie by mali byť stránkované (max. osem konverzácií na stránku). Pod vylistovanými konverzáciami by malo byť tlačidlo „«“, ktoré presunie administrátora na prvú stranu vylistovaných konverzácií, tlačidlo „<“, ktoré presunie administrátora o jednu stranu dopredu (ak ešte nie je na prvej stran) a tlačidlá „>“, „»“, ktoré majú fungovať analogicky (lenže na opačnú stranu).

Veľa každého riadku konverzácie má byť začiarkovacie políčko, ktorým vieme konverzáciu označiť.

Nad vylistovanými konverzáciami má byť nadpis „Správy“ a tlačidlo „Nastavenia“, ktorým sa bude môcť administrátor dostať do časti aplikácie s nastaveniami automaticky generovaých správ.

Ďalej pod nadpisom má byť začiarkovacie políčko, ktorým bude môcť administrátor začiarknúť (označiť) všetky konverzácie na aktuálnej strane. Okrem tohto začiarkovacieho políčka, tam majú byť tlačidlo „Obnoviť“ a ak je označená aspoň jedna konverzácia, tak potom sú viditeľné aj tlačidlá „Vymazať“ a „Označiť ako prečítané“ alebo „Označiť ako neprečítané“.

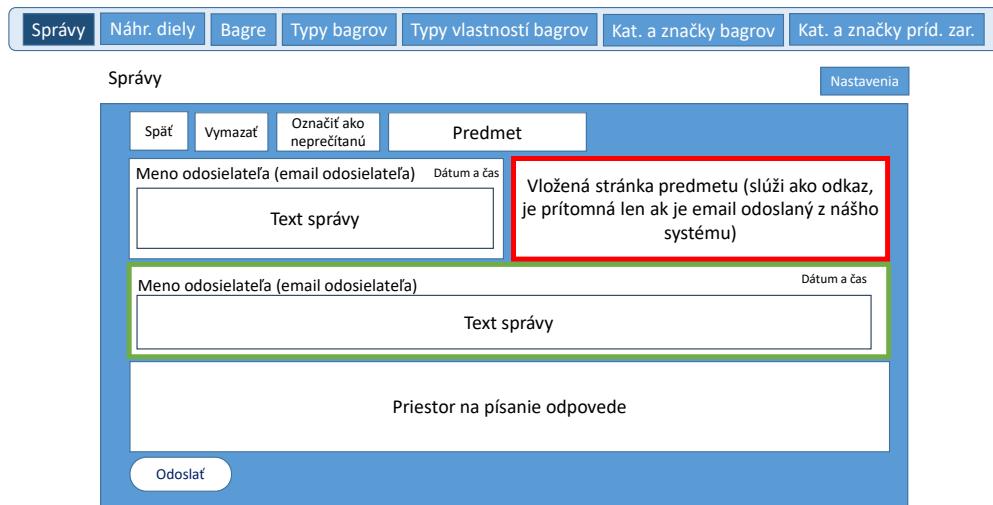
Tlačidlo „Označiť ako neprečítané“ sa zobrazí vtedy, keď sú všetky označené správy prečítané. Tlačidlo „Označiť ako prečítané“ sa zobrazí vtedy, keď sú všetky označené správy neprečítané alebo sú medzi správami aj prečítané, aj neprečítané správy.

Tlačidlom „Vymazat“ bude môcť administrátor vymazať označené konverzácie. Po kliknutí naň sa má zobraziť potvrzovacie modálne okno s nadpisom „Vymazať vybrané konverzácie“ a textom „Naozaj chcete vymazať vybrané konverzácie natrvalo?“.

Po kliknutí na nejaký z riadkov sa má administrátorovi zobraziť vybraná konverzácia.

2.7.2 Správy – konverzácia

Po kliknutí na nejakú z konverzácií sa má administrátorovi zobraziť daná konverzácia (viď obr. 2.22).



Obr. 2.22: Konverzácia

V hornej časti sa má nacházať nadpis „Správy“ a tlačidlo „Nastavenia“ (pre otvorenie nastavenia automaticky generovaných správ). Pod nimi sa majú nachádzať tlačidlá „Späť“ (pre návrat do schránky), „Vymazat“ a tlačidlo „Označiť ako neprečitanú“ (pre označenie správy za neprečitanú).

Tlačidlom „Vymazat“ bude môcť administrátor vymazať otvorenú konverzáciu. Po kliknutí sa má zobraziť potvrzovacie modálne okno s nadpisom „Vymazať konverzáciu“ a textom „Naozaj chcete vymazať túto konverzáciu natrvalo?“.

Napravo od tlačidiel má byť zobrazený predmet (poslednej správy v konverzáции).

Pod tlačidlami a predmetom majú byť zobrazené jednotlivé správy (emaily) konverzácie zoradené zhora dole od najstaršej po najnovšiu.

Ak bola (prvá) správa odoslaná z nášho systému, tak vedľa nej má byť vložená časť aplikácie, odkiaľ bola správa odoslaná (na obr. 2.22 viď časť označenú

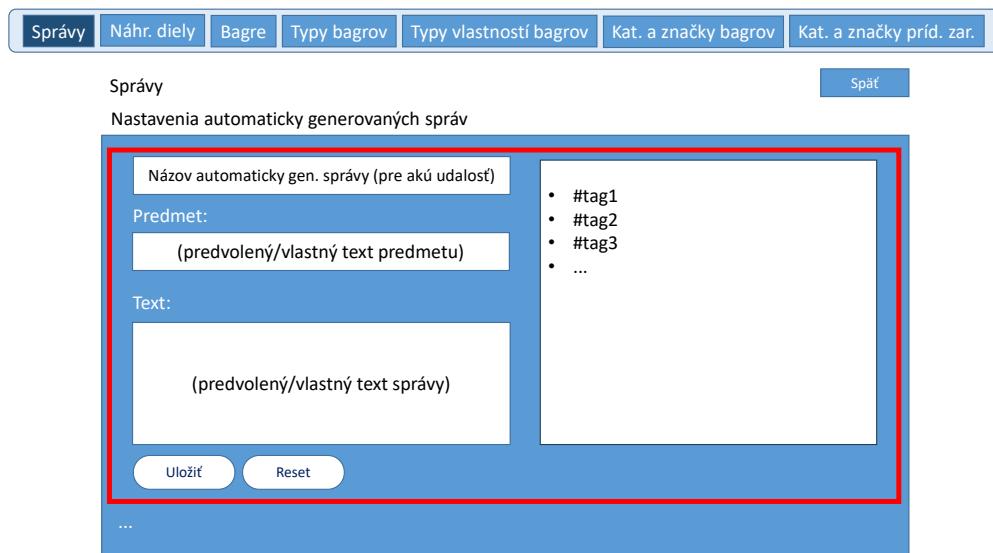
červeným rámom). Po kliknutí na túto časť má byť administrátor presmerovaný do časti aplikácie, odkiaľ bola správa odoslaná (má sa otvoriť v novom okne). V prípade, že správa nebola odoslaná z nášho systému, má sa správa roztiahnuť na šírku tejto časti aplikácie (rovnako ako je druhá správa, označená zeleným rámom, na obr. 2.22).

Pre každú správu v konverzácií sa má zobraziť meno a email odosielateľa, dátum a čas správy, a takisto text správy.

Pod vylistovanými správami má byť priestor pre písanie odpovede a tlačidlo „Odoslať“, ktorým môže administrátor odosielat správy do konverzácie.

2.7.3 Správy – nastavenia automaticky generovaných správ

Po kliknutí na tlačidlo „Nastavenia“ sa administrátorovi zobrazia nastavenia automaticky generovaných správ (viď obr. 2.23).



Obr. 2.23: Nastavenia automaticky generovaných správ

Vo vrchnej časti sa má nachádzať nadpis „Správy“ a tlačidlo „Späť“, ktoré schová/zatvorí nastavenia. Pod nimi sa má nachádzať nadpis „Nastavenia automaticky generovaných správ“ a sekcia, kde sa pre každú automaticky generovanú správu zobrazí časť, kde sa dá nastavovať jej tvar (jedna zo spomínaných častí je vyznačená červeným rámom na obr. 2.23).

Každá z takýchto častí má obsahovať názov automaticky generovanej správy (resp. pre koho, akú udalosť je určená, napr. „Pre víťaza aukcie“). Ďalej má obsahovať šablónu predmetu, šablónu textu. A v pravej časti zoznam tagov povolených v danej automaticky generovanej správe.

Tag má tvar „#{názov tagu}“, kde názov tagu môže obsahovať iba písmená a znak „_“. Pomocou tagov bude vedieť administrátor špecifikovať kde sa má v texte (reps. predmete) automaticky generovanej správy nachádzať konkrétna hondota špecifikovaná tagom. Napr. ak by sme použili vetu „Gratulujeme, vyhrali ste #nazov_stroja“ ako šablónu predmetu, tak email, ktorý by prišiel užívateľovi by mohol vyzerat takto: „Gratulujeme, vyhrali ste Stroj123“ (ak by bol názov výherného stroja „Stroj123“).

Každá z automaticky generovaných správ má mať predvolený text, ktorý bude môcť byť pomocou šablón prepísaný. Ak nebudú definované vlastné šablóny, majú sa využívať predvolené, a preto sa majú v spomínaných poliach pre predmet a text zobrazovať predvolené šablóny. Ak by boli predvolené šablóny prepísané a neskôr sa vymazú, tak polia sa majú znova napíniť predvolenými hodnotami.

Dalej pod poliami sa majú nachádzať tlačidlá „Uložiť“ (pre uloženie šablón) a „Reset“ (pre navrátenie k predvoleným hodnotám).

2.7.4 Náhradné diely

TODO

2.7.5 Bagre

TODO?

2.8 Splnenie P6

V tejto podkapitole si prejdeme časti aplikácie splňajúce požiadavku P6, t. j. umožnenie užívateľom registrovať a prihlásiť sa do systému (pričom administrátori sa majú prihlasovať rovnakým spôsobom ako bežní zákazníci), umožnenie zmeny údajov v profile, zjednodušenie prihláseným užívateľom vypĺňanie formulárov (aby nemuseli zadávať svoje osobné údaje), umožnenie neprihláseným užívateľom posielat dopyty a účastniť sa aukčných dražieb.

V častaiach 2.3.6, 2.4.3 a 2.5 už bolo spomenuté, že aj prihlásení, aj neprihlásení užívateelia sa môžu dopytovať na bager, prídavné zariadenie, výkopovú službu alebo takisto môžu ponúkať sumy do dražby (teda zapájať sa do aukcie). Pričom prihláseným užívateľom sa nemajú zobrazovať polia s osobnými údajmi, takže ich nemusia vypĺňať. Tým pádom sú požiadavky zjednodušenie vypĺňania formulárov prihlásenými užívateľmi, umožnenie posielania dopytov neprihláseným užívateľom a umožnenie účasti v aukcii neprihláseným užívateľom splnené.

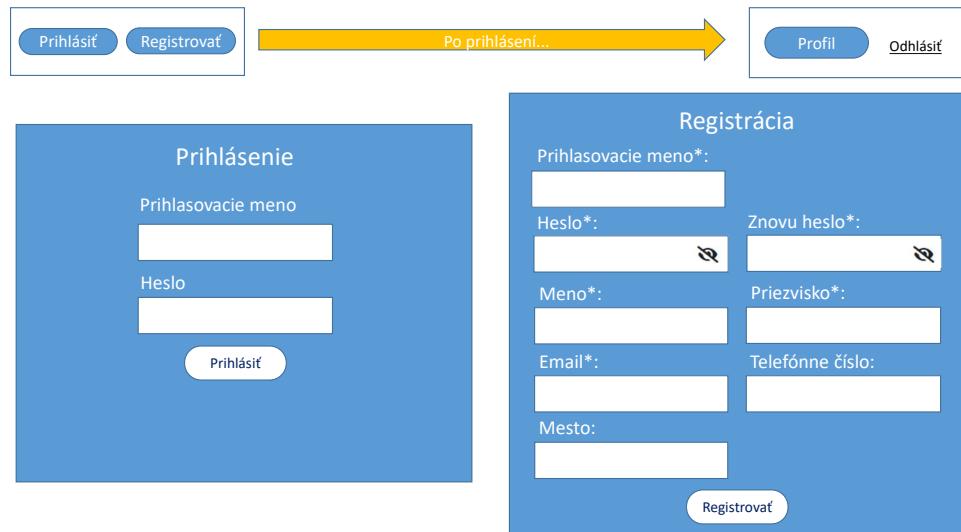
2.8.1 Prihlasovací panel

V ľavej časti aplikácie nad navigáciou sa má nachádzať prihlasovací panel (pre lepšiu predstavu umiestnenia panela vidieť obr. 2.1)).

Ak je užívateľ neprihlásený, tak sa majú v prihlasovacom paneli nachádzať tlačidlá „Prihlásiť“ a „Registrovať“ (vidieť vľavo hore na obr. 2.24).

Po kliknutí na tlačidlo „Prihlásiť“ má byť užívateľ presmerovaný do časti aplikácie s prihlasovacím formulárom (vidieť ľavú časť obr. 2.24).

Po kliknutí na tlačidlo „Registrovať“ má byť užívateľ presmerovaný do časti aplikácie s regisračným formulárom (viď pravú časť obr. 2.24).



Obr. 2.24: Prihlasovací panel (hore), prihlasovací formulár (vľavo), regisračný formulár (vpravo).

Ak je užívateľ prihlásený, tak sa má v prihlasovacom paneli nachádzať tlačidlo „Profil“ a odkaz „Odhlásiť“ (viď vpravo hore na obr. 2.24).

Po kliknutí na tlačidlo „Profil“ má byť užívateľ presmerovaný do časti aplikácie s jeho údajmi.

Po kliknutí na odkaz „Odhlásiť“ má byť užívateľ odhlásený.

2.8.2 Prihlásenie

TODO

2.8.3 Registrácia

TODO

2.8.4 Profil

Po kliknutí na tlačidlo „Profil“, ktoré sa má zobrazovať prihláseným užívateľom v prihlasovacom paneli, má byť užívateľ presmerovaný do časti aplikácie s formulárom, ktorý má obsahovať jeho údaje. Užívateľ si bude môcť pomocou

tohto formulára svoje údaje upravovať. Navyše nad formulárom sa má nachádzať nadpis „Profil“. Pre lepšiu predstavu tejto časti aplikácie vid ď obr. 2.25.

The screenshot shows a user profile edit screen. At the top left is the word "Profil". Below it is a blue header bar with the title "Moje údaje". Underneath are several input fields:

- "Prihlasovacie meno*": An input field with three dots below it.
- "Nové heslo*": An input field with a password strength icon and three dots below it.
- "Znova heslo*": An input field with a password strength icon and three dots below it.
- "Meno*": An input field with three dots below it.
- "Priezvisko*": An input field with three dots below it.
- "Email*": An input field with three dots below it.
- "Telefónne číslo": An input field with three dots below it.
- "Mesto": An input field with three dots below it.

At the bottom right of the form area is a blue "Uložiť" (Save) button.

Obr. 2.25: Profil užívateľa

V hornej časti formulára sa má nachádzať nadpis „Moje údaje“. Formulár má obsahovať polia: prihlásenie meno, heslo, znova heslo, meno, priezvisko, email, telefónne číslo a mesto. Pričom všetky polia, okrem polí telefónne číslo a mesto, sú povinné. Povinné polia majú byť označené hviezdičkou.

Všetky polia, až na polia heslo a znova heslo, majú byť vyplnené užívateľovými údajmi.

Ak užívateľ začne písat do pola heslo alebo znova heslo, namiesto písmen by sa mali zobrazovať hviezdičky. Pri týchto poliach by mali byť ikonky, ktoré kliknutím umožnia užívateľovi meniť medzi hviezdičkami a písmenami v poliach heslo a znova heslo. Teda ide o funkciu zobrazovať a schovať heslo.

Pod formulárom by sa malo nachádzať tlačidlo „Uložiť“ pre uloženie zmien.

Ak neboli vykonané žiadne zmeny, tak po kliknutí na tlačidlo „Uložiť“ sa má zobrazovať vyskakovacie okno s textom "Neboli vykonané žiadne zmeny." a tlačidlom „OK“.

Ak užívateľ zinvalídnuje nejaké z polí (napr. zanechá prázdne povinné pole), má sa zobrazovať chybová správa pri danom poli. Výnimkou sú polia heslo a znova heslo, tie môžu zostať prázdne.

Ak by si chcel užívateľ zmeniť heslo, bude musieť vyplniť obe polia, a ak by obsah polí neboli rovnaký, tak sa má zobrazovať vyskakovacie okno s textom "Obsah políčok Nové heslo a Žnova heslo nie sú rovnaké." a tlačidlom „OK“. Alebo ak by si chcel užívateľ zmeniť heslo a zadal by staré heslo, tak znova sa má zobrazovať vyskakovacie okno s textom "Nové heslo nemôže byť staré heslo." a tlačidlom „OK“.

Po úspešnej zmene údajov sa má zobraziť vyskakovacie okno s textom "Údaje úspešne zmenené." a tlačidlom „OK“.

3. Analýza

TBA: Uvodna rec o tom ako si ideme rozmysliet naplenenie poziadavok z uvedu a zaroven z casti Navrh UI (predosla kapitola).

3.1 Výber typu aplikácie

Kvôli požiadavke P10 (1.1) dáva dobrý zmysel vytvoriť náš systém ako webovú aplikáciu – vyriešime tým problém s distribúciou softvéru k (aj potenciálne novým) zákazníkom, a takisto vyriešime problém s potenciálne zastaralými počítačmi vo firme. Statická stránka by nám nestačila nakoľko chceme administrátorm umožniť dynamickú zmenu obsahu.

3.2 Výber jazyka a frameworku

Teraz keď už vieme, že ideme vytvárať webovú aplikáciu si musíme rozmysliet v akom jazyku a pomocou akého frameworku ju budeme využívať.

3.2.1 Výber backend jazyka

Na tvorbu webovej aplikacie s bohatym UI sa hodí high level jazyk, napr Java alebo C#. Autor ovlada C#, preto volime C# a platformu .NET, ktorá je s ním spojená.

3.2.2 SPA

SPA by nam pomohli s interaktivitou nasho UI (tým sa myslí napr zobrazovanie/schovavanie formov alebo obsah zorbazovany/schovavany podla toho aky užívateľ je prihlásený a HLAVNE odpocet aukčných ponuk), kedže kod je vykonavany rychlejsie lebo je vykonavany priamo na klientovi.

Aplikacia by mohla bezat aj offline, ale to pre nas není podstatné.

Nevyhodou je že by sme museli vytvárať nejaké bezpečné API, napr pre pracu s databazou.

Takisto je nevyhodou dlhsí prvotny nacitavaci cas (lebo cakame kým sa zdrojaky stiahnu), ktorý by mohol odraditi novych potencialnych zakaznikov.

Prikladom tohto by boli napr JS frameworky ako React, Angular atď alebo C#ový Blazor WebAssembly.

3.2.3 Klasické webové aplikácie

Zdá sa že toto by bolo dobre pretože kód je vykonavany na serveri, preto možeme priamo pracovať s databazou, nemusíme vytvárať api.

Užívateľovi sa serviruje len HTML, CSS, JS, takže nemali by mať starsie pocitace problem.

Navyše kvôli tomu sú CEO friendly, co znamena že search-enginy ako napr Google ich vie ľahšie najst, a ich prvotny nacitavaci cas je rýchly, co je tiež dobre pre ziskavanie novych zakaznikov.

ALE vracaju sa zo serveru cele stranku... No a ked sa pozrieme na nase UI, tak vidime ze je celkom interaktivne a castokrat meni len cast aplikacie, nie celu (znovu moze uviesť príklady ako v SPA).

Tento problem by sa možno dal vyriesiť napr. pomocou nejakých featur v ASP.NET MVC, ale ako si možme v časti Navrh UI vsimnut, tak viacero časti aplikacie vyzerá rovnako, a preto by sa nam hodilo, ak by sme tieto rovnake/podobne časti mohli zadefinovať raz a potom využívať na viacerých miestach. ASP.NET MVC nam umožňuje definovať celú stranku ale nie len časť stránky – komponentu. Existuje ešte jedna možnosť ktorá nam prinesie výhody SPA, klasických webových aplikácií a zarovne nam umožní definovať spomínané komponenty.

3.2.4 Blazor Server

Existuje však ešte jedna možnosť – Blazor Server. Tento framework je niečo „medzi“. Poskytuje výhody klasickej webovej aplikácie v tom zmysle, že má rýchle prvotné nacitanie stránky, serviruje užívateľom iba HTML, CSS, JS. Ďalšou výhodou je že kód je iba na serveri preto možme komunikovať s databázou priamo a nemusíme vytvárať separačné API. Ale zároveň nam poskytuje výhody SPA interaktivity preto, pretože pri zmene UI sa neposiela zo serveru celá stránka ale iba zmeny. Nie je to súčasťou tak rýchle ako keby kód bezal u klienta ale keďže my netvoríme nejakú náročnú reálnu aplikáciu, tka nam to stačí. Zarovne by bolo nevyhodou to že sa vytvára zátaž na server, ale keďže program je určený pre male firmy, ktoré nemajú až tak vysoký počet užívateľov tak nam to nevadí, plus dalsou nevyhodou by možlo byť že nemožné aplikácia bezat offline, ale to nam takisto nevadí, to sme ani nechceli.

Naďalej kvôli výberu tohto frameworku budeme môcť písat aj frontend, aj backend v jazyku C#.

3.3 Volba databázy

TODO

3.3.1 Návrh relačného modelu databázy

TODO

3.3.2 Volba typu databázy a databázového servera

TODO

3.3.3 ORM

TODO

3.4 Aukcia- odpočet a výhodnocovanie

TODO

PO TADE JE NOVY TEXT

TBA: Ako vidíme tak potrebujeme bohaty frontend, na to sa hodi napr JavaScriptové frameworky ako su React, Vue atd., a takisto nam nestaci Backend as a Service (který poskytuje napr Amazon...), lebo napr kvôli vyhodnocovaniu aukcie, takže preto vlastný backend. Na tvorbu webovej aplikacie s bohatym UI sa hodi high level jazyk, napr Java alebo C#. Autor ovlada C#, preto volime C# a platformu .NET, ktorá je s ním spojená. Backend by sme mohli implementovať pomocou frameworku ASP.NET MVC. No este existuje iná varianta a tou je Blazor. Ten nam umožní využívať komponenty (to sa nam hodi napr. aj kvôli implementácii odpočtu) a takisto nám umožní písat frontend aj backend v rovnakom jazyku – C#.

Blazor poskytuje viacej hosting modelov a v dobe vyberu technológií existovali dva – Blazor WebAssembly a Blazor Server. Blazor WebAssembly funguje skôr na spôsob SPA, kde logika aplikacie beží na klientskom počítači vo webovom prehliadaci, a s ním spojené nejaké problémky [vymenujem, napr. to že search engingy možu mať problem s ním, P10 aby to užívateľom rozbehol pc, aby neboli zastaralé prehliadac, prvotné nacitanie trva dlho (lebo stahuje zdrojáky) čo by mohlo potencialných zakazníkov odraďať, a tiež nejaké WebAPI na server by sme museli vytvárať]. Napäť Blazor Server sa podoba skôr tradičnému web app prístupu a hodi sa nam viac lebo je CEO-friendly, kde beží na servery a užívateľ dostane len HTML, CSS, JS, preto aj starsie pc by nemali mať problem s rozbehnutím, a takisto nemať preto problem s dlhým prvotným nacitaním, a navyše nemusíme vytvárať WebAPI.

Preto si volime C# (.NET) a Blazor Server.

V tejto kapitole sa zamyslíme nad tým, ako splniť požiadavky definované v Úvode (??).

Pre splnenie požiadavky P1 dáva veľmi dobrý zmysel vytvoriť naše riešenie ako webovú aplikáciu. Týmto spôsobom sa nemusíme staráť o distribúciu programu k užívateľom. Stačí ak má zákazník (resp. admin) pripojenie na internet.

Je sice pravda, že voľba webovej aplikácie zahrňa i voľbu hostingu. A ten nemusí byť lacný. To by mohlo byť v rozpore s P2. Ale je potrebné dodať, že ak by sme zvolili klasickú desktopovú aplikáciu, tak by sme ju museli nejakým spôsobom dodať zákazníkovi. A to by bolo nepraktické, prípadne by mohlo stať takisto nejaké peniaze. Navyše práve webová aplikácia má potenciál pomôcť firme tak, že ju zákazník objaví pri surfovania internetu.

Pre splnenie P4, P5 a P8 je jasné, že budeme potrebovať databázu. A to na to, aby si firmy vedeli samé tvoriť ponuku, ktorú si do databázy uložia. Po príchode zákazníka bude možné ponuku z databázy načítať a zobraziť. Podobne v prípade P8. Keď sa užívateľ zaregistruje, jeho údaje sa uložia v databáze a pri prihlásení sa z nej prečítajú a môžu použiť pre vyplnenie formulárov podľa potreby.

Znova sa vrátim k P5. Kedže ide o aukciu, budeme potrebovať nejaký mechanizmus, ktorý by vedel zabezpečiť odpočet, a takisto vyhodnotenie aukcie na pozadí. Taktiež si musíme rozmyslieť, ako sa má aukcia správať v rôznych situáciách.

Na to, aby sme splnili P6, musí byť náš softvér schopný posielat správy. Z pod-

mienky P3 usudzujeme, že nikto nebude pri softvéri sedieť, a teda posielanie dopytov by nemalo mať povahu četu. Posielanie správ bude prebiehať prostredníctvom emailov. To nám vytvára novú požiadavku na softvér. Aby administrátor nemusel preklikávať medzi svoju emailovou schránku a naším systémom, bolo by dobre integrovať jeho schránku priamo do systému.

3.5 Volba typu aplikácie, jazyka a framework

Po prejdení požiadaviek vieme, že chceme vytvoriť webovú aplikáciu s bohatým uživatelským rozhraním, ktorá by bola schopná posielat a prijímať správy, pracovať s databázou, umožnila nám autentikáciu a autorizáciu, a taktiež vykonávať prácu na pozadí. Pre túto úlohu sa hodia vysokoúrovňové jazyky, ako sú napríklad C# alebo Java... Na základe autorových skúseností si volíme jazyk C# a platformu .NET, ktorá je s ním spojená.

Platforma .NET nám pre vývoj webových aplikácií poskytuje framework ASP.NET alebo Blazor. Obe frameworky sú si podobné. Rozdiel nájdeme v tom, že Blazor umožňuje vytváranie komponentov. Komponent si môžeme predstaviť ako logickú časť stránky (napr. tabuľka, tlačidlo...). Po zadefinovaní komponentu ho vieme „recyklovať“. Tým myslím to, že ho môžeme použiť na viacerých miestach na webe. Na každom mieste sa bude správať a vyzeráť rovnako (príp. vieme jeho správanie meniť pomocou parametrov). Táto myšlienka komponentov sa autorovi páči, dobre sa s ňou pracuje a neskôr si ukážeme ako nám pomôže vyriešiť problém s odpočtom.

Blazor poskytuje viacero hosting modelov. V čase rozhodovania existovali dva: Blazor WebAssembly a Blazor Server. Výber WebAssembly by zahrínať niekoľko problémov. Pri prvotnej návštive stránky sa musia klientovi stiahnuť zdrojové kódy aplikácie. To môže chvíľu trvať a mohlo by to odradiť nových potenciálnych zákazníkov. V prípade Blazor Server tento problém nemáme, pretože kód beží na serveri a užívateľovi sa servíruje už len prerenderovaný HTML, CSS, JavaScript kód stránky. Z rovnakého dôvodu sú weby vytvorené Blazor Serverom SEO-friendly (čo znamená, že sú dohľadateľné vyhľadávačmi, akým je napríklad Google). V prípade WebAssembly môžu mať vyhľadávače problém. Na to, aby sa dostali k obsahu stránky si musia obsah vygenerovať zo stiahnutých zdrojových kódov. Ale to môžu mať zakázané z bezpečnostných dôvodov alebo toho nemusia byť schopné. V súčasnosti webové prehliadače (Google Chrome, Mozilla Firefox, ...) podporujú WebAssembly. Ale autor pozorovaním zistil, že vo firmách sa zvyknú využívať staré počítače s potenciálne starým softvérom. Takže hrozí, že by sme mali problém WebAssembly rozbehnúť. Kvôli spomenutým dôvodom si volíme Blazor Server.

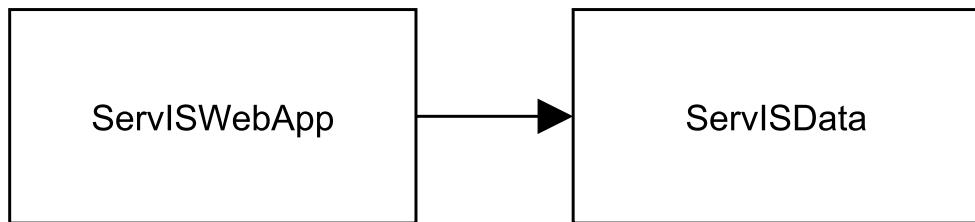
3.6 Návrh systému

V Úvode sme rozhodli, že náš systém je webová aplikácia, ktorá pracuje s databázou. Webová aplikácia funguje ako rozhranie pre interakciu s užívateľmi. Databázu potrebujeme kvôli perzistencia dát. Ako vidíme, obe časti poskytujú vlastnú funkcionality. Ak tieto časti oddelíme, pomôžeme rozšíritelnosti systému.

Ked' už vidíme, že systém je zložený z dvoch častí, podme si rozmyslíme ako

budú spolu interagovať. Webová aplikácia potrebuje pre svoje fungovanie dátu. Tejto závislosti sa nezbavíme. Lenže dátova časť nepotrebuje webovú aplikáciu pre svoje fungovanie. Závislosť z tejto strany neexistuje.

Majme preto dva projekty: ServISData a ServISWebApp. ServISData slúži ako dátová časť, ktorá je nezávislá a ServISWebApp poslúži ako webová aplikácia, ktorá je závislá na dátach z projektu ServISData. Architektúru demonštruje obrázok 3.1.



Obr. 3.1: Architektúra systému (predstavuje závislosť ServISWebApp od ServIS-Data).

3.7 Volba databázy

Vieme, že náš systém potrebuje pre splnenie podmienok P4, P5 a P8 databázu. V tejto kapitole si vyberieme typ databázy, databazový server a rozoberieme si aké entity potrebujeme.

3.7.1 Návrh relačného modelu databázy

Z P4 vieme, že potrebujeme entity pre bagre a prídavné zariadenia. Niekoľko by mohlo napadnúť spojiť obe entity do jednej, ale to my nespravíme. Ide o rozdielne entity, ktoré môžu uchovávať rozdielne informácie (a ako neskôr v texte uvidíme, skutočne budú uchovávať rozdielne dátu).

Kedže ide o ponuku, ktorú chceme prezentovať zákazníkom, chceme okrem textových údajov prezentovať položku, či už stroj alebo prídavné zariadenie, pomocou fotky. A nie jednej (predpokladám, že položku budú chcieť firmy predviesť zákazníkom z viacerých uhlov). Znovu by niekomu mohlo napadnúť, že by bol dobrý nápad zlúčiť entitu fotky stroja s entitou fotky prídavného zariadenia. Ale tieto veci nie sú totožné. Ak by sme entity zlúčili (a mali teda iba 1 entitu pre fotku všeobecne), tak by existovala možnosť priradiť fotku pridaného zariadenia stroju (a naopak). Ale to je nesprávne. Preto znova vytvoríme dve entity. Jedna bude fotka stroja, druhá bude fotka prídavného zariadenia. Pre fotku stroja platí, že patrí práve jednému stroju. Stroj môže mať viacero fotiek. Analogicky platí pre prídavné zariadenia a ich fotky.

Pri strojoch sa ešte zastavíme. Existujú rôzne značky strojov (napr. Locust, Eurocomach,...), a takisto rôzne kategórie strojov (napr. šmykľom riadené nakladače, pásové bagre,...). Ďalej v texte, keď budem používať spojenie typ stroja, tak tým myslím kombináciu značky stroja a kategórie stroja. Každý typ stroja sa môže lísiť druhom a formou údajov. Napríklad typ A má hmotnosť ako vlastnosť,

ktorú chceme spolu so zbytkom údajov zobraziť užívateľovi. Ďalej typ B má namiesto hmotnosti údaj o výške stroja. Existujú údaje (ako sú napr. meno a opis stroja), ktoré existujú pre každý stroj. Ale takisto existujú údaje, ktoré sa líšia v závislosti od typu stroja. Takýmito údajmi sú vlastnosti stroja. Ako budeme tieto premenlivé údaje ukladať? Jedno z riešení, ktoré by nás mohlo napadnúť je vytvoriť rodičovskú entitu, ktorá by obsahovala údaje spoločné pre všetky typy strojov. A v entitách, ktoré by dedili od rodičovskej triedy by sme dodefinovali premenlivé vlastnosti. Toto riešenie by pravdepodobne fungovalo, lenže má zásadnú nevýhodu. Zakaždým keď si firma zmyslí, že potrebuje nový typ stroja, by entita musela byť manuálne doprogramovaná. Ale keďže my chceme systém navrhnuť všeobecne tak, aby si každá firma vedela zadefinovať vlastnú ponuku strojov, volíme inú alternatívu. Vytvoríme si entitu pre typ stroja. Každý stroj bude nejakého typu. Každý typ obsahuje údaj o značke a kategórii. Značka a kategória sú tiež ďalšími entitami. Typ stroja určuje, akého typu budú vlastnosti konkrétneho stroja. Takže budeme potrebovať entitu typ vlastnosti stoja. Tá obsahuje údaje: názov vlastnosti (napr. hmotnosť, výška,...) a typ hodnoty vlastnosti (napr. číslo, text,...). Teda admin bude môcť priradiť typu stroja akého typu bude mať konkrétny stroj vlastnosti.

Stroj vie svoj typ, a ten vie aké (akého typu) má konkrétny stroj vlastnosti. To, čo ešte nevieme, sú konkrétné hodnoty vlastností stroja. Dovolím si vysvetliť na príklade. Momentálne máme informáciu o tom, že konkrétny stroj S, typu T, má vlasnosť hmotnosť, ale stále nevieme konkrétnu hodnotu, teda stále nevieme kolko váži. Preto vytvoríme entitu reprezentujúcu vlastnosť stroja. Táto entita vie, akého je typu. A takisto v sebe uchováva konkrétnu hodnotu vlastnosti (v kontexte príkladu, uchováva v sebe váhu stroja). Každý stroj má v sebe toľko vlastností, kolko ich je zadefinovaných v jeho type.

Teraz sa vrátme k prídavným zariadeniam. Každé prídavné zariadenie má, podobne ako stroj, takisto svoju značku a patrí donejakej kategórie. Navyše má oproti strojom aj údaj o tom, pre akú kategóriu strojov je zariadenie vytvorené. Avšak na rozdiel od predošlého prípadu so strojmi, sa tento prípad lísi v tom, že každé prídavné zariadenie, bez ohľadu na kombináciu typu, kategórie a kategórie stroja, má druh a formu údajov rovnakú. Takže stačí ak vytvoríme entitu pre značku a kategóriu prídavného zariadenia (entitu pre kategóriu stroja už máme) a entita reprezentujúca prídavné zariadenie si bude v sebe držať informáciu o tom, akej je značky, kategórie a pre akú kategóriu strojov je vytvorená.

Aby sme splnili P5, budeme potrebovať entitu reprezentujúcu aukčnú ponuku. Aukčná ponuka bude držať informáciu o tom, aký stroj je v dražbe. A takisto údaje o ponuke (napr. vyvolávacia cena, koniec aukcie,...).

No okrem udržania údajov o aukčnej ponuke a draženom stroji, si musíme zapamätať i údaje o ponukách užívateľov, ktorí sa do aukcie zapojili. Preto si vytvoríme entitu reprezentujúcu ponuku užívateľa. Bude v sebe niesť údaje o tom, ktorý užívateľ ponúkol sumu, v akej výške a pre ktorú aukčnú ponuku.

Entitu užívateľa som načal už v prechádzajúcim odstavci. Túto entitu skutočne budeme potrebovať a to aj kvôli splneniu P8. Údaje užívateľov musíme pri registrácii uchovať, aby sme nimi vedeli predvypĺniť formuláre, a takisto aby sme vedeli vytvoriť prihlásovanie.

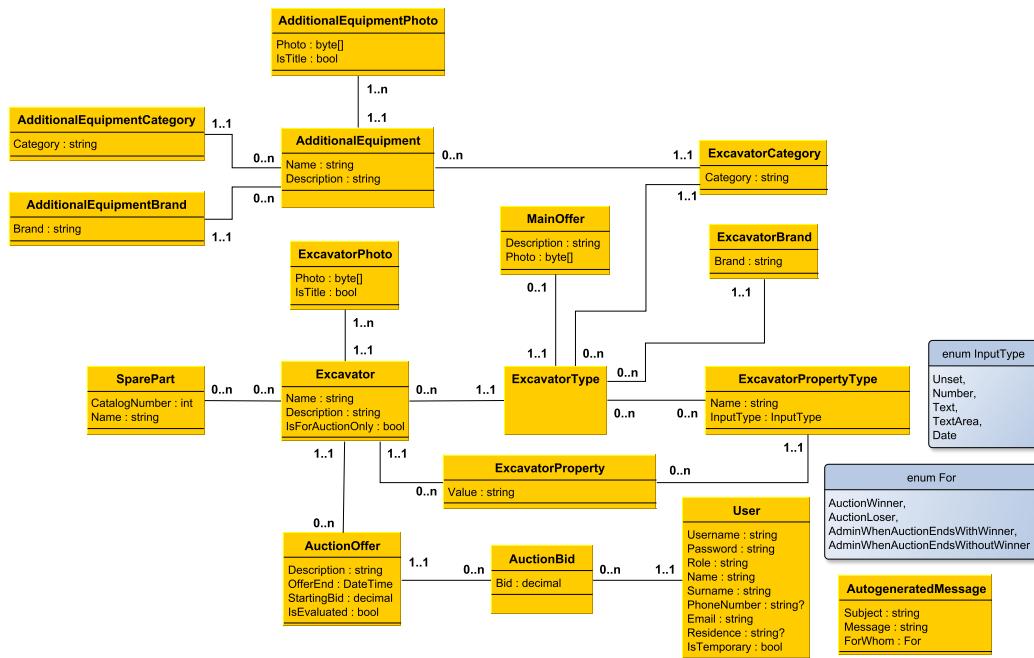
Čítateľ by mohol navrhnúť, že pre splnenie P6 budeme potrebovať entitu reprezentujúcu správy. Toto riešenie by mohlo fungovať, ale ako si neskôr ukážeme,

existuje aj iné riešenie. Také, ktoré nám (okrem iného, ale detailnejšie rozobranie príde neskôr) ušetrí úložisko v databáze. Preto entitu pre správy nevytvárame.

Pre splnenie P7 si vytvoríme entitu, ktorá bude reprezentovať náhradné diely stroja. Každý náhradný diel bude niesť informáciu o tom, v ktorých strojoch sa nachádza. A každý stroj bude vedieť aké diely obsahuje. Každý náhradný diel obsahuje katalógové číslo. Toto číslo je unikátne medzi strojmi, a preto by mohlo byť použité ako primárny klúč entity. Ale autor sa z opatrnosti a kvôli konzistencii (každá entita má id) rozhodol využiť ako primárny klúč id i pri náhradných dieloch.

Hlavným predmetom predaja sú stroje. A preto chceme aby prvé, čo zákazník po príchode na stránku uvidí bola ponuka strojov. Takže ešte budeme potrebovať entitu na reprezentáciu hlavnej ponuky. Táto entita vie aký typ strojov ponúka, a zároveň obsahuje reprezentatívnu fotku a opis strojov daného typu.

Pre detailnejšiu predstavu môžeme návrh vidieť na obrázku 3.2.



Obr. 3.2: Relačný model databázy.

Pozorný čítač si mohol všimnúť, že na obrázku existuje entita, ktorú sme ešte nespomenuli. Ide o entitu reprezentujúcu automaticky generované správy. Na dôvod jej existencie sa pozrieme neskôr. Zatiaľ len prezradím, že táto entita sa využije pri fungovaní aukcie.

3.7.2 Volba typu databázy a databázového servera

Kvôli tomu, že naše dátá majú presne definovanú štruktúru, si volíme relačnú databázu. Tá nám poskytne rýchly prístup k dátam, a takisto zaručí ich neporušenosť. Ďalším plusom je, že autor má s relačnými databázami trochu viac skúseností, než s NoSQL databázami.

Čo sa týka databázového servera, tak si najprv podme rozmyslieť, čo od neho budeme vyžadovať. Kvôli P2 chceme, aby bol server (v najlepšom prípade) bez-

platný. Taktiež musí byť kompatibilný s platformou .NET. Tieto požiadavky spĺňajú servery: Microsoft SQL Server, Oracle Database a MySQL. V prípade prvých dvoch serverov je síce pravda, že poskytujú aj bezplatnú verziu, ale tie sú obmedzené na kapacitu. Ale aj napriek obmedzeniu je kapacita dosť veľká. V prípade Community verzie MySql takéto obmedzenie kapacity nemáme žiadne. V budúcnosti, ak by sa firme darilo a získala by viacej zákazníkov, prípadne rozšíria ponuku, by sa nám mohla neobmedzená kapacita servera hodit. Preto si volíme MySql.

3.7.3 ORM

Ked' už máme vybraný databázový server, podme si rozmyslieť ako s ním budeme pracovať. Potrebujeme vytvoriť databázu s tabuľkami, vkladať, editovať, čítať a mazať dátu. Všetky tieto úkony by sme mohli vykonať tak, že budeme písat kód s SQL syntaxou. Ale keďže autor nemá veľa skúseností s jazykom SQL, prichádza do úvahy využiť nejaký z ORM frameworkov (z ang. object relational mapping). Asi najznámejšími na platforme .NET sú: Dapper a Entity Framework Core. Využívanie frameworku Dapper zahŕňa písanie SQL kódu. Pretože sa snažíme obmedziť písanie SQL kódu, nedáva zmysel si vybrať Dapper. Našou voľbou je preto Entity Framework Core. Konkrétnie s prístupom založeným na kóde (anglicky code first approach). To znamená, že entity z relačného modelu najprv napíšeme do C# kódu ako triedy a z nich Entity Framework Core vytvorí tabuľky databázy.

3.8 Aukcia- odpočet a vyhodnocovanie

Ako už bolo skôr v texte spomenuté, budeme potrebovať nejaký mechanizmus, ktorý zvládne vykonávať odpočet do konca každej aukčnej ponuky, a takisto aby ju po skončení vedel vyhodnotiť.

Odpočet by sme mohli implementovať tak, že by sme si vytvorili inštanciu triedy `Timer`¹ pre každú aukčnú ponuku. Každá z inštancií by každú sekundu odpálila event, pri ktorom by došlo k prerenderovaniu komponentu s časom do skončenia ponuky. Toto riešenie by síce fungovalo, ale je to mrhanie združmi. Stačí nám jeden `Timer`. Na jeho event `Elapsed` si každý komponent s odpočtom zaregistruje metódu na prerenderovanie. Týmto spôsobom sa nám podarí ušetriť zdroje, ale vyskytá sa otázka. Kde bude inštancia triedy `Timer` uložená? Prirodzenou odpovedou by bolo, že ju uložíme do rodičovského komponentu. Limitácia tohto riešenia spočíva v tom, že ak budeme potrebovať vykonať nejakú operáciu periodicky každú sekundu, ale v komponente, ktorá sa nenachádza v rodičovi s triedou `Timer`, tak nemáme prístup k triede `Timer`. Lepším riešením bude presunúť `Timer` do nového vlákna, ktoré bude bežať na pozadí. Takýmto spôsobom si môžeme kedykoľvek a odkialkoľvek registrovať periodické operácie.

Tu by som sa ešte rád zastavil a vrátil k poznámke z úvodu, kedy som spomnul, že komponenty nám pomôžu vyriešiť problém s odpočtom. Vieme, že kvôli odpočtu dôjde každú sekundu k prerenderovaniu stránky. No určite nechceme,

¹<https://learn.microsoft.com/en-us/dotnet/api/system.timers.timer?view=net-8.0>

aby sa nám každú sekundu prerenderovala celá stránka (resp. hľadali zmeny na celej stránke). Ak izolujeme odpočet do samostatného komponentu, prerenderovanie sa vykoná iba v ňom. Zbytok stránky to neovplyvní.

Ďalej si potrebujeme rozmyslieť ako bude prebiehať vyhodnocovanie aukcie. Náš systém bude fungovať tak, že užívateľ (ak má záujem o dražený stroj) odošle svoju ponúkanú sumu. Tá sa uloží v databáze. Potrebujeme mechanizmus, ktorý by sledoval aukčné ponuky. Ak by uvidel, že nejaká z ponúk už skončila, tak ju vyhodnotí. Periodické sledovanie aukčných ponúk je dlhodobo bežiaca operácia, a preto sa hodí ju takisto vykonávať vo vedľajšom vlákne na pozadí.

Podme sa ešte pozrieť na to, čo presne zahŕňa vyhodnocovanie jednej aukčnej ponuky. Prejdeme všetky ponúknuté sumy, nájdeme najvyššiu sumu a užívateľ, ktorý ju ponúkol bude vyhlásený za víťaza. Vítaz aukcie bude informovaný o skutočnosti, že vyhral stroj v aukcii. Porazení budú takisto oboznámení o svojej prehre. Okrem tohto scenára existuje ešte jeden. Čo ak sa nikto nezapojil do aukcie? Má sa aukčná ponuka zmazať? Alebo iba označiť za ukončenú a schovať sa pred bežnými zákazníkmi? Alebo sa má koniec aukcie presunúť na neskôr? Ako vidíme možnosti je viacero. Prvé dve možnosti vyžadujú zásah admina, ktorý by musel nanovo vložiť ponuku do systému (prvý prípad) alebo upraviť ponuku a znova ju zviditeľniť zákazníkom (druhý prípad). Všimnime si, že posledná z možností nevyžaduje akciu admina. Administrátor môže ponuku upraviť, ale nemusí. Systém sa postará sám o seba a funguje aj bez zásahu admina. A to je presne to, čo od nás vyžuje P3. Preto volíme tretí prístup. V prípade, že aukcia skončí bez víťaza, tak systém o tom upozorní administrátora a koniec ponuky posunie na neskôr (napr. o týždeň).

V oboch prípadoch (odpočet i vyhodnocovanie) sme sa rozhodli využiť vlákna na pozadí. Vo frameworku Blazor existuje trieda `BackgroundService`². Tá nám umožní rozbehnúť kód na pozadí. Takže logiku behu vlákienn na pozadí si nemusíme implementovať sami.

3.9 Posielanie a prijímanie správ

Z pozorovania autor vie, že zákazníci týchto firiem, a rovnako i samotné firmy, sú zvyknuté na komunikáciu pomocou emailov.

V predchádzajúcim texte sme už viackrát spomínali, že budeme posielat správy. Jednak kvôli spĺneniu P6, ale takisto pre upozorňovanie účastníkov a administrátorov na priebeh aukcií. Tiež sme si povedali, že správy nebudú reprezentované entitami v databáze. Ako teda vyriešime túto úlohu? Ak bude posielanie správ (dopytov, notifikácií z aukcie) implementované prostredníctvom posielania emailov do emailovej schránky firmy, tak ušetríme miesto v databáze. To nám môže potenciálne znížiť náklady, čo je v prospech P2. Nevýhodou by však bolo, že administrátor by musel chodiť kontrolovať emaily mimo našu aplikáciu. Tejto nevýhody sa dokážeme zbaviť. A to tak, že vytvoríme zjednodušenú emailovú schránku priamo v našej aplikácii. Týmto spôsobom docielime toho, že užívateľ vie odoslať správu z nášho systému, náš systém vie odoslať správu o stave aukcií,

²<https://learn.microsoft.com/en-us/aspnet/core/fundamentals/host/hosted-services?view=aspnetcore-7.0&tabs=visual-studio#backgroundservice-base-class>

admin si ich v našej aplikácii vie prečítať a reagovať na ne. Zároveň sa nestaráme o ukladanie správ v databáze, takže šetríme úložisko. A čo viac, zákazníci vedia kontaktovať firmu i mimo nášho systému (ked zákazník pošle firme email z miesta mimo nášho systému, admin si ho vie aj napriek tomu v našej aplikácii prečítať).

Dospeli sme k tomu, že nepotrebujeme správy len odosielat ale potrebujeme ich aj čítať. Ešte predtým, než si zvolíme spôsob akým budeme správy prijímať a posielat, si podľame vybrať akú emailovú službu budeme využívať. Od služby vyžadujeme, aby podporovala protokoly IMAP (pre prijímanie správ) a SMTP (pre posielanie správ). Jednou zo služieb ktorá spĺňa podmienku je Gmail od spoločnosti Google. Ide o veľkú spoločnosť, ktorú pozná snáď každý a autor dôveruje ich zabezpečeniu. Ďalším plusom je to, že k tejto službe existuje dokumentácia. Navyše služba poskytuje API (z ang. application programming interface), ktoré by sme mohli využiť v prípade potreby. Čitateľ by mohol poznamenať, že API by sme mohli využiť ku kompletnej implementácii našej schránky. Dôvodom prečo nevyužijeme API služby pre kompletnú implementáciu je ten, že to čo ideme vytvoriť sa skôr podobá emailovému klientovi. Na jeho implemenáciu je vhodnejšie použiť protokoly IMAP a SMTP (rovnako to spomína i dokumentácia API³). Ako vidíme, Gmail sa zdá byť dobrým kandidátom, a preto si ho zvolíme.

Ďalej potrebujeme zistiť ako môžeme využívať protokoly IMAP a SMTP z nášho programu. Zdá sa, že najznámejšími balíčkami pre prácu s týmito protokolmi sú MailKit a AE.Net.Mail. Obe umožňujú prácu s IMAP i SMTP, ale AE.Net.Mail neboli už dlhšiu dobu aktualizovaný. A preto, ak sa z tohto systému stane dlhodobejší projekt, sa zdá byť lepšou možnosťou MailKit.

3.9.1 Automaticky generované správy

Spomínali sme, že systém bude sám automaticky odosielat správy, či už víťazom aukcie, porazeným alebo adminom, o stave aukcie. Ak by náš systém využívali viaceré firmy, je prirodzené, že by si chceli tvar automaticky generovaných správ upraviť. Náš systém firmám túto funkcialitu umožní. Preto sme si v časti Návrh relačného modelu datábazy (3.7.1) dopredu pripravili entitu reprezentujúcu automaticky generované správy.

³<https://developers.google.com/gmail/api/guides>

4. Vývojová dokumentácia

V tejto kapitole sa pozrieme na implementáciu nášho systému. Zameráme sa na organizáciu a implementáciu dôležitých častí aplikácie. Cieľom tohto textu nie je opísat správanie každého riadku kódu. Na to slúži kód samotný (prípadne komentáre, ktoré autor pridal na miesta, kde to uznal za vhodné). Odkaz k zdrojovým kódom aplikácie sa nachádza v prílohe (A.1).

Celá aplikácia je rozdelená do dvoch projektov- ServISData a ServISWebApp.

4.1 ServISData

Zmyslom projektu ServISData je správa databázových entít a komunikácia s databázou. Teraz si prejdeme jednotlivé priečinky tohto projektu a opíšeme si ich obsah.

4.1.1 Koreňový priečinok projektu

Nachádza sa tu `ServISDbContext` a spolu s ním i `ServISDbContextFactory`. Tieto triedy sú zodpovedné za konfiguráciu databázy, a tiež samotné pripojenie aplikácie k databáze. Viac o týchto triedach a spôsobu ich použitia v projekte nájdeme tu.

V priečinku sa tiež nachádzajú triedy `AutogeneratedMessageForExtensions` a `InputTypeExtensions`. V oboch prípadoch ide o extension metódy umožňujúce získať metadáta z atribútov, ktoré sme použili v enumoch `AutogeneratedMessage.For` a `InputType`.

Takisto sa v priečinku nachádza už spomenutý `InputType`. Ide o enum, ktorého hodnoty predstavujú možné typy hodnôt vlastností stroja.

Ďalej sa tu nachádza trieda `ServISApi`. Tá slúži pre komunikáciu s databázou. Respektívne umožnuje iným projektom (v našom prípade ServISWebapp) modely ukladať, čítať, editovať a mazať z databázy.

4.1.2 Attributes

Priečinok obsahuje atribúty¹. Konkrétnie ide o `AutogeneratedMessageDataAttribute` a `InputLabelAttribute`.

Prvý slúži na nastavenie predvoleného predmetu a textu automaticky generovaných správ, ale takisto aj na nastavenie ich podporovaných tagov.

Druhý slúži pre uloženie užívateľsky prívetivejšieho názvu typu hodnoty vlastnosti stroja. Tieto názvy sa zobrazujú napríklad pri vytváraní typu vlastnosti stroja.

4.1.3 DataOperations

Priečinok obsahuje triedy, ktoré ich užívateľom umožnia vykonávať rôzne operácie nad dátami (filtranie, stránkovanie, ...).

¹<https://learn.microsoft.com/en-us/dotnet/csharp/advanced-topics/reflection-and-attributes/>

4.1.4 Interfaces

Priečinok obsahuje rozhrania (ang. interfaces).

`IServISApi` nás zbaví potreby upravovať kód využívajúci API projektu `ServISData` v prípade, ak sa rozhodneme vymeniť `ServISApi` za nejakú inú implementáciu.

`IPhoto` nám umožní všeobecne pracovať s fotkami aj napriek tomu, že pôjde o fotky rôznych entít.

`IItem` nám dovolí písat všeobecný kód pre prácu s modelmi entít (využíva sa napr. pri mazaní entít z databázy; vid' metódu `DeleteItem` v triede `ServISApi`).

4.1.5 Migrations

Tento priečinok bol vygenerovaný frameworkom Entity Framework Core a obsahuje vygenerovaný kód. Ide o migrácie². Nad migráciou môžeme rozmýšľať ako nad commitom v gitu. Po aplikovaní migrácie dôjde k zmene v databáze (napr. sa pridá nový stĺpec do nejakej z tabuľiek).

4.1.6 Models

Priečinok obsahuje modely- triedy reprezentujúce entity uložené v databáze.

Špeciálne sa zastavíme pri triede `AutogeneratedMessage` a enume `AutogeneratedMessage.For`. Platí, že každá hodnota enumu predstavuje druh automaticky generovanej správy. Napríklad `AutogeneratedMessage.For.AuctionWinner` predstavuje automaticky generovanú správu pre víťaza aukcie. A tiež platí, že pre každú hodnotu tohto enumu existuje inštancia triedy `AutogeneratedMessage` uložená v databáze. Predvolené hodnoty správ sa nachádzajú v atribútoch hodnôt enumu a administrátormi definované hodnoty sa ukladajú do databázy.

4.2 ServISWebApp

Projekt ServISWebApp predstavuje rozhranie pre užívateľov a zároveň obsahuje logiku nášho systému. Znova, ako v predošej podkapitole, si prejdeme jednotlivé priečinky tohto projektu a opíšeme si ich obsah.

4.2.1 Koreňový priečinok projektu

Priečinok obsahuje súbory `_Imports.razor` (nachádzajú sa v ňom `using` direktívy aplikované pre každý komponent v projekte), `App.razor` (koreňový komponent), `appsettings.json` (ide o konfiguráciu aplikácie, v ktorej okrem iného vieme nastaviť emailovú adresu, ktorú systém využíva pre odosielanie a prijímanie emailov) a `Program.cs` (obsahuje kód zodpovedný za spustenie celej aplikácie). Pre viac informácií o štruktúre Blazor projektu môže čitateľ kliknúť tu.

V časti Models 4.1.6 bolo spomenuté, že pre každú hodnotu enumu `AutogeneratedMessage.For` existuje inštancia triedy

²<https://learn.microsoft.com/en-us/ef/core/managing-schemas/migrations/?tabs=dotnet-core-cli>

`AutogeneratedMessage` uložená v databáze. To zabezpečuje metóda `StoreNewAutogeneratedMessagesAsync`, ktorá sa nachádza v `Program.cs`.

4.2.2 Auth

Priečinko obsahuje triedy zodpovedné za prihlásovanie užívateľa a hashovanie hesla.

4.2.3 BackgroundServices

V tomto priečinku sa nachádzajú triedy predstavujúce služby bežiace na pozadí. Konkrétnie ide o službu zodpovednú za vyhodnocovanie aukcie a službu, ktorá každú sekundu vykoná akciu, ktorú si užívateľ triedy v službe zaregistroval (v našom prípade ide o prerenderovanie komponentov pri odpočte).

4.2.4 Components

V tomto priečinku sa nachádzajú komponenty využité v aplikácii.

Komponenty ktorých meno končí na slovo “Lister”, slúžia na zobrazenie (vylistovanie) nejakého obsahu. Väčšinou sú tým obsahom komponenty končiace na slovo “Card”.

Ďalej sa tu nachádzajú komponenty začínajúce na slovo “Input”. Tieto komponenty spolu s komponentom `ChecklistTable` slúžia na získavanie vstupu užívateľa.

Komponenty `Message`, `Messages`, `ThreadRow`, `ThreadView` a `ThreadsView` tvoria miesto pre manažovanie správ. Umožňujú adminovi čítať správy, a takisto na nich odpovedať. Navyše `MessageSettings` umožňuje adminovi meniť tvar automaticky generovaných správ.

Komponent `CountdownDisplayer` sa využíva na zobrazenie odpočtu do konca aukcie.

`CustomDataAdapter` sa využíva v komponente `ItemsManagement` na čítanie dát.

`LoginPanel` umožňuje neprihlásenému užívateľovi prístup k prihlásovaciemu a regisitračnému formuláru. Alebo v prípade prihláseného užívateľa poskytuje možnosti prejsť na profil a odhlásiť sa.

`PhotoSlider` sa využíva na zobrazovanie obrázkov (môžeme ho vidieť napríklad na stránke detailu ľubovoľného stroja).

Komponent `TabControl` v sebe obsahuje komponenty `TabPage`. Umožňujú užívateľom (najmä adminom, pretože bežní zákazníci majú iba jeden tab) preklikávať sa medzi rôznymi časťami profilu.

Ďalej sa tu nachádzajú priečinky `Buttons`, `Forms` a `Managements`.

V priečinku `Buttons` sa nachádzajú komponenty reprezentujúce rôzne tlačidlá.

Vo `Forms` sa nachádzajú rôzne formuláre ktorými dokážeme vytvárať a do databázy ukladať inštancie modelov (napr. stroje, prídavné zariadenia, ...). Ale takisto sa tam nachádza i komponent `DemandForm`, ktorý zákazníkom umožňuje odosielat dopyt.

A v priečinku `Managements` sa nachádza komponent `ItemsManagement`. Tento komponent tvorí rozhranie pre administrátorov na správu entít (napr. strojov,

náhradných dielov,...). Umožňuje adminom entity vytvárať, editovať a mazať. Takisto im pri prezeraní existujúcich entít umožňuje ich vyhľadávanie a triedenie.

4.2.5 CssProviders

Triedy v tomto priečinku sú zodpovedné za zmenu css štýlov vo formulároch pri modifikovaní a validácii políčok formuláru. O triede **FieldCssClassProvider** sa čitateľ môže dozvedieť viac tu.

4.2.6 Pages

V tomto priečinku sa nachádzajú stránky a priečinok Admin, ktorý obsahuje stránky exkluzívne pre administrátorov.

4.2.7 Resources

Tento priečinok obsahuje “.resx” súbory so slovenskými prekladmi slov a viet. Využívajú ich komponenty z balíčku od spoločnosti Syncfusion.

4.2.8 Shared

V tomto priečinku sa nachádzajú triedy zdieľané (potenciálne) celým projektom.

Triedy **Email** a **Thread** slúžia ako dátosné alternatívy k triedam balíčka MailKit. Môžeme nad nimi rozmýšľať ako nad fasádami, ktoré zjednodušujú prístup k dátam.

EmailManager pokrýva funkcionality práce s emailami, napr. ich posielanie a prijímanie.

MainLayout a **NavMenu** sú štandardné Blazor komponenty. Prvý definuje rozloženie stránky a druhý navigáciu.

SyncfusionLocalizer a **SyncfusionDataOperations** sú triedy, ktoré využívajú kód z balíčka od spoločnosti Syncfusion. Prvá sa stará o nastavenie lokalizácie užívateľského rozhrania Syncfusion komponentov. Druhá vykonáva operácie nad dátami (napr. vyhľadávanie, triedenie,...). Čitateľ môže byť v tento moment mierne zmätený, pretože raz sme tu už podobnú triedu mali. Konkrétnie v časti DataOperations (4.1.3). Rozdiel je v tom, že trieda spomenutá v predošej časti využíva na konfiguráciu operácií našu vlastnú triedu

MyDataOperations.Configuration, kdežto **SyncfusionDataOperations** využíva triedu **DataManagerRequest**, ktorá je takisto z balíčka spoločnosti Syncfusion.

FileTools združuje metódy pre prácu so súbormi. Konkrétnie v našom prípade ho využívame pre prácu s fotkami.

Ostali nám už len trieda **AuctionSummary** a trieda **AutogeneratedMessageExtensions**, ktorá sa nachádza v priečinku Extensions. Obe triedy spolu úzko súvisia.

AutogeneratedMessageExtensions obsahuje extension metódy pre triedu **AutogeneratedMessage**. **AutogeneratedMessage** obsahuje text (predmet a telo správy). V texte sa nachádzajú tagy a slúži ako šablóna. Metódy triedy

`AutogeneratedMessageExtensions` fungujú tak, že vezmú túto šablónu a na-
miesto tagov dosadia skutočné dátá vzaté z inštancií
tryedy (príp. tried) `AuctionSummary`.

5. Užívateľská dokumentácia

Užívateľská dokumentácia existuje v podobe videonávodu pre administrátorov na obsluhu systému. Odkaz na video sa nachádza v prílohe (A.2).

Záver

V závere zhodnotíme ako sa nám podarilo naplniť požiadavky definované v Úvode, konkrétnie v podkapitole Požiadavky na softvér (??).

- **P1 Dostupnosť**

Náš systém je webovou aplikáciou, ktorá je dostupná pre užívateľov odkiaľkoľvek (samozrejme za predpokladu, že majú prístup k internetu).

- **P2 Náklady**

Všetky balíčky využívané aplikáciou sú buď úplne zdarma alebo využívame ich bezplatné verzie. Čo sa týka databázového servera, tak ten takisto využívame v jeho bezplatnej verzii.

- **P3 Minimálna obsluha softvéru**

Táto požiadavka sa odzrkadluje v správaní aukcie. V prípade, že aukčná ponuka skončí bez toho, aby sa jej niekto účastnil, posunie sa jej termín ukončenia aj bez zásahu administrátora.

- **P4 Predstavenie ponuky zákazníkom**

Systém je schopný načítať z databázy dát strojov aj prídavných zariadení a zobraziť ich užívateľom.

- **P5 Aukcia**

V aplikácii existuje aukcia strojov. Administrátorom je umožnené vytvárať aukčné ponuky, v ktorých sa draží (adminom vybraný) stroj. Užívateľom je v prípade záujmu umožnené ponúkať sumy do dražby. Po skončení odpočtu prebehne vyhodnocovanie aukčných ponúk, kde sa rozhodne kto je ich víťazom. Vítaz je oboznámený prostredníctvom emailu, že vyhral. Porazení sú informovaní o skutočnosti, že sa im aukciu nepodarilo vyhrať. Admin je upozornený, že aukčná ponuka skončila a kto je jej víťazom.

- **P6 Dopyt**

Systém umožňuje užívateľom podávať dopyt prostredníctvom formulára, ktorý odošle email administrátorovi s potrebnými informáciami o užívateľovi a jeho záujme o danú položku.

- **P7 Prístup k súčiastkam strojov**

V aplikácii si vie administrátor po kliknutí na profil v správe strojov rozkliknúť detail nejakého konkrétného stroja. Okrem iných údajov o spomínanom stroji sa mu zobrazí aj zoznam náhradných dielov, ktoré stroj obsahuje.

- **P8 Registrácia a prihlásenie užívateľov**

Systém umožňuje bežným užívateľom vytvoriť si účet v systéme, a takisto sa doň prihlásiť (možnosť odhlásenia je samozrejmostou). A po prihlásení už nemusia do formulárov (pri posielaní dopytu alebo pri zapájani sa do aukcie) zadávať svoje údaje.

5.1 GDPR

Kedže je náš systém webovou aplikáciou zhromažďujúcou užívateľské údaje, tak je potrebné riešiť zásady ochrany osobných údajov. Autor nie je právnikom, a teda nemôže oficiálne prehlásiť, že systém je v súlade s GDPR¹. Ale systém bol navrhnutý tak, aby zásady splnil. Užívateľa upozorňuje na podmienky a zásady používania systému. Takisto užívateľovi umožnuje vlastné dátá zmazať zo systému, zobraziť a exportovať ich na vyžiadanie. Dátá sice nie sú šifrované softvérovo, ale táto podmienka sa dá splniť pri výbere hostingu. Stačí si vybrať hosting, ktorý ponúka disky podporujúce šifrovanie dát.

5.2 Možné vylepšenia

Softvér sice spĺňa požiadavky a jeho funkčnosť je dostatočná, ale stále existuje priestor pre rôzne vylepšenia a pridanie nových funkcia. V tejto podkapitole si o niektorých povieme.

- **Výkonnosť listovania ponuky**

Ponuku firmy (napr. stroje) prečítame z databázy a vylistujeme jednotlivé položky. Nevyužívame žiadnu virtualizáciu, ani stránkovanie. Kedže systém je určený pre malé firmy a nepredpokladá sa veľká ponuka, tak táto skutočnosť nepredstavuje problem. No v budúcnosti, ak by sa firme používali nás systém darilo a rozhodla by sa rozšíriť ponuku, mohla by sa virtualizácia alebo stránkovanie hodit.

- **Vylepšenie manažovania správ**

Systém sice umožňuje administrátorovi prijímať emaily, a takisto na nich odpovedať, mazat ich, a tiež označovať správy ako prečítané/neprečítané. No je v tom trocha pomalý. Akcie sice netrvajú hodiny, a ani minúty, ale do budúcnosti je to určite niečo, na čom by bolo dobré popracovať.

Takisto by bolo dobré správam pridať stránkovanie, vyhľadávanie a triedenie podľa dátumu. Hlavným dôvodom existencie správ je umožniť administrátorovi reagovať na dopyt užívateľov, a takisto admina upozorňovať na stav aukčných ponúk. Z podstaty obsahu správ sa čas na ich vybavenie odhaduje na jednotky dní. A teda predpokladáme, že počet správ v schránke nebude príliš vysoký. V prípade, že by ich admin nemazal stále platí, že relevantné správy sa budú nachádzať na vrchu schránky. Ako vidíme, spoľané vlastnosti preto nie sú pre naše účely nevyhnutné. Ale takisto ide o funkcionality, ktoré by mohli oceniť firmy obzvlášť v budúcnosti, kedy by sa zvýšením počtu zákazníkov zvýšila aj frakvencia prichádzajúcich správ.

- **Upozornenie na prehliadku stroja**

Vo svete to funguje tak, že keď si užívateľ zakúpi stroj, tak po určitom čase by firma mala prísť na prehliadku a stroj skontrolovať. Preto by sme do nášho systému mohli pridať funkciu, ktorá by fungovala následovne. Ak užívateľ odošle dopyt na nejaký stroj a obchod prebehne úspešne,

¹https://en.wikipedia.org/wiki/General_Data_Protection_Regulation

admin by túto skutočnosť zaznačil v systéme, čím by užívateľovi priradil dopytovaný stroj. Potom by sa spustil časovač, ktorý by po uplynutí definovaného času upozornil administrátora na blížiacu sa prehliadku.

A. Prílohy

A.1 Implementácia

Implementácia aplikácie (Visual Studio solution) sa nachádza v priečinku ServIS na adrese: <https://github.com/milantru/ServIS>.

A.2 Videonávod

Odkaz k videonávodu pre administrátorov: .