

**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 poďakovať 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: Cílem tejto práce bolo vytvoriť softvérové dielo pre malé firmy zaoberajúce sa zemnými a výkopovými prácami, opravou a predajom bagrov, ktoré nemajú prístup k vhodnému softvérovému riešeniu pre svoju činnosť. Pri riešení práce sme vychádzali z konzultácií s majiteľom jednej z takýchto firiem.

Vzniknutá webová aplikácia, napísaná pomocou frameworku Blazor Server, predstavuje riešenie problému. Je schopná zobrazit ponuku (bagre, prídavné zariadenia) a umožňuje užívateľom odosielať dopyt (v podobe emailov) na tieto ponuky. V aplikácii tiež existuje aukcia, kde sa môžu dražiť opravené bagre. Užívateľia si tiež môžu v aplikácii vytvoriť účet. Vymenované funkcionality môžu využívať prihlásení aj neprihlásení užívateľia. Bežní prihlásení užívateľia nemusia vyplňovať informácie o sebe vo formulároch pri dopytovaní sa na ponuku. Prihlásení administrátori majú možnosť spravovať stránku, t. j. pridávať nové, editovať a mazať existujúce ponuky, odpovedať na správy atď.

Klíčová slova: Informačný systém C# .NET Blazor Server MySQL

Title: ServIS – a web system for companies dealing with excavator repairs

Author: Milan Truchan

Department: Department of Distributed and Dependable Systems

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

Abstract: The goal of this bachelor thesis was to create a software for small companies dealing with earthworks and excavation works, repairing and selling excavators, which do not have access to a suitable software solution for their activity. The solution to the work was based on consultations with the owner of one of these companies.

The resulting web application, written using the Blazor Server framework, represents a solution to the problem. It is able to display the offer (excavators, additional equipment) and allows users to send inquiries (in the form of emails) to these offers. There is also an auction in the application, where repaired excavators can be auctioned. Users can also create an account in the application. The listed functionalities can be used by both logged-in and non-logged-in users. Regular registered users do not have to fill in information about themselves in the forms when inquiring about the offer. Logged-in administrators can manage the page, i.e. add new, edit and delete existing offers, reply to messages, etc.

Keywords: Information system C# .NET Blazor Server MySQL

Obsah

1	Úvod	5
1.1	Požiadavky na systém	5
1.2	Cieľ práce	10
2	Návrh užívateľského rozhrania	11
2.1	Hlavné rozloženie aplikácie	11
2.2	Modálne potvrdzovacie okno	12
2.3	Sekcie Podmienky používania a Zásady ochrany osobných údajov .	13
2.4	Splnenie P2 a P3	14
2.4.1	Hlavná ponuka	16
2.4.2	Vytvorenie novej a úprava existujúcej hlavnej ponuky . . .	18
2.4.3	Ponuka (nových) bagrov	19
2.4.4	Ponuka prídavných zariadení	19
2.4.5	Detail bagra, prídavného zariadenia	21
2.4.6	Dopyt	22
2.4.7	Vytvorenie nového a úprava existujúceho bagra	23
2.4.8	Úprava bagrov, ktoré nie sú v ponuke (nových) bagrov . .	24
2.4.9	Vytvorenie nového a úprava existujúceho prídavného zaria- denia	25
2.5	Splnenie P4	26
2.5.1	Aukčné ponuky	27
2.5.2	Detail aukčnej ponuky	28
2.5.3	Ponúkание sumy do dražby	29
2.5.4	Vytvorenie novej a úprava existujúcej aukčnej ponuky . . .	30
2.6	Splnenie P8	31
2.7	Splnenie P9	33
2.8	Splnenie P5, P7 a správa predmetov	34
2.8.1	Karta Správy – schránka	36
2.8.2	Karta Správy – konverzácia	37
2.8.3	Karta Správy – nastavenia automaticky generovaných správ	38
2.8.4	Karta Náhradné diely	40
2.8.5	Karty Bagre, Typy bagrov, Typy vlastností bagrov	42
2.8.6	Karty Kategórie a značky bagrov, Kategórie a značky prí- davných zariadení	44
2.9	Splnenie P6	48
2.9.1	Prihlasovací panel	49
2.9.2	Prihlásenie	49
2.9.3	Registrácia	50
2.9.4	Profil	50
3	Analýza	53
3.1	Výber typu aplikácie	53
3.2	Rozdelenie na frontendovú a backendovú časť	53
3.3	Výber typu webovej aplikácie	54
3.3.1	Single-page application	54

3.3.2	Multi-page application	55
3.3.3	Blazor Server	56
3.4	Volba databázy	57
3.4.1	Návrh relačného modelu databázy	58
3.4.2	Volba databázového servera	61
3.4.3	Object Relational Mapping	62
3.5	Aukcia – odpočet a vyhodnocovanie	63
4	Vývojová dokumentácia	67
4.1	Adresárová štruktúra systému	68
4.2	Backendová časť	68
4.2.1	Trieda CustomAuthenticationStateProvider	69
4.2.2	Trieda PasswordHasher	69
4.2.3	Trieda ServISApi a rozhranie IServISApi	69
4.2.4	Trieda DataOperations	69
4.2.5	Trieda EmailManager	69
4.2.6	Trieda TimerService a jej potomkovia	69
4.3	Frontendová časť	70
4.3.1	Zobrazovanie obsahu podľa roly	70
4.3.2	Hlavné rozloženie stránky	70
4.3.3	Modálne potvrdzovacie okno	71
4.3.4	Stránky s podmienkami používania a zásadami ochrany osobných údajov	71
4.3.5	Stránka s hlavnou ponukou (domovská stránka)	71
4.3.6	Stránka pre vytvorenie a úpravu hlavnej ponuky	71
4.3.7	Stránka s ponukou (nových) bagrov	72
4.3.8	Stránka s ponukou prídavných zariadení	72
4.3.9	Stránka s detailom bagra, prídavného zariadenia	72
4.3.10	Stránka pre vytvorenie a úpravu bagra	72
4.3.11	Stránka pre vytvorenie a úpravu prídavného zariadenia	73
4.3.12	Stránka s aukčnými ponukami	73
4.3.13	Stránka s detailom aukčnej ponuky	73
4.3.14	Stránka pre vytvorenie a úpravu aukčnej ponuky	74
4.3.15	Stránky Služby, Kontakt a O nás	74
4.3.16	Stránka pre správu webu	74
4.3.17	Stránka s prihlasovacím formulárom	76
4.3.18	Stránka s registračným formulárom	76
4.3.19	Stránka s profilom užívateľa	76
5	Užívateľská dokumentácia	77
5.1	Databáza	77
5.2	Predpoklady pre lokálne spustenie	77
5.3	Návod na lokálne spustenie	77
5.4	Nasadenie do Azure	78
6	Záver	79
6.1	GDPR	80
6.2	Možné vylepšenia	81

Seznam použité literatury	83
A Přílohy	87
A.1 Implementácia	87
A.2 Videonávod pre zákazníkov	87
A.3 Videonávod pre administrátorov	87
A.4 Prehľad elektronických príloh	87

1. Úvod

V súčasnosti existujú malé firmy, ktoré fungujú ako dodávatelia 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 samozrejmosťou.

My sme dostali ponuku na tvorbu riešenia od jednej z takýchto firiem. Konkrétne 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 spĺňalo požiadavky uvedené v nasledujúcej podkapitole.

1.1 Požiadavky na systém

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

- **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 funkcionalít, 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. Všetky tieto údaje okrem opisu sú povinné. 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, pričom všetky okrem opisu sú povinné. Vlastnosti bagra sú určené kombináciou jeho značky a kategórie (teda typom bagra, každý typ bagra môže mať rôzne vlastnosti, napr. jeden typ bagra by mohol mať vlastnosť ťažná sila, iný typ bagra by mohol mať vlastnosť šírka lyžice). Informácie o bagri 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. Taktiež platí, že 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 bagrov je prídavné zariadenie určené, opis a fotky, pričom všetky okrem opisu sú povinné. 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 systém umožní pridávať, odstraňovať a editovať.

• **P3 Posielanie dopytu**

Takisto klient od softvéru vyžaduje, aby umožnil zákazníkovi 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 žiadanom predmete, údaje o užívateľovi, a tiež správu užívateľa. Užívateľskými údajmi sa myslí meno, priezvisko, email – tie sú povinné údaje. A takisto telefónne číslo, mesto – tie sú nepovinné údaje.

- **P4 Aukcia**

Keď sme v predošlých podmienkách 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, ktorej môže špecifikovať dátum jej konca, pričom administrátor nesmie byť schopný nastaviť túto hodnotu do minulosti, počiatočnú sumu (vyvolávaciu cenu), ktorá nesmie byť záporná, popis ponuky (v popise môže napísať napr. čo bolo v bagri opravované), a takisto aby mohol do ponuky vybrať (opravený) bager. Všetky tieto údaje okrem popisu sú povinné.

Ďalej klient vyžaduje aby systém umožnil zákazníkovi 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**

Keď aukcia skončí, systém má upozorniť jej účastníkov (poslať email s automaticky generovanou správou) 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 stále platí, že má na to softvér administrátora upozorniť, ale taktiež má 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**

Keďže posielanie dopytov a správanie aukcie zahŕňa posielanie emailov administrátorom, tak je tiež žiadúce, aby sa emaily 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 podobáť na Gmail. Teda aspoň funkcionalitou, 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 generovaných (odosielaných) správ týkajúcich sa aukcie.

• **P6 Registrácia a prihlásenie užívateľov**

Ďalšou požiadavkou je, aby softvér umožnil zákazníkovi 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 prihlasovacie (užívateľské) meno, heslo, takisto bude môcť zadať svoje (krstné) 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 zadať 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**

Keď si zákazník zakúpi bager, tak po nejakom čase má firma vykonať kontrolu tohto bagra. Ale predtým než pôjdu zamestnanci vykonať kontrolu si musia zistiť, 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 dielov rovnakého druhu v bagri nás nezaujíma) a jeden náhradný diel sa môže nachádzať vo viacerých bagroch. Náhradný diel obsahuje informácie: katalógové číslo, názov (obe sú povinné).

- **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 sťahovať, inštalovať žiaden softvér, a takisto v prípade administrátorov sa chceme vyhnúť problémom s kompatibilitou (z pozorovania autor vie, že firmy častokrát používajú staré počítače s potenciálne starým softvérom, čo by mohlo spôsobovať problémy s prevádzkou nášho systému, napr. sa využívajú počítače s operačným systémom XP).

- **P11 Náklady**

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

1.2 Cieľ práce

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

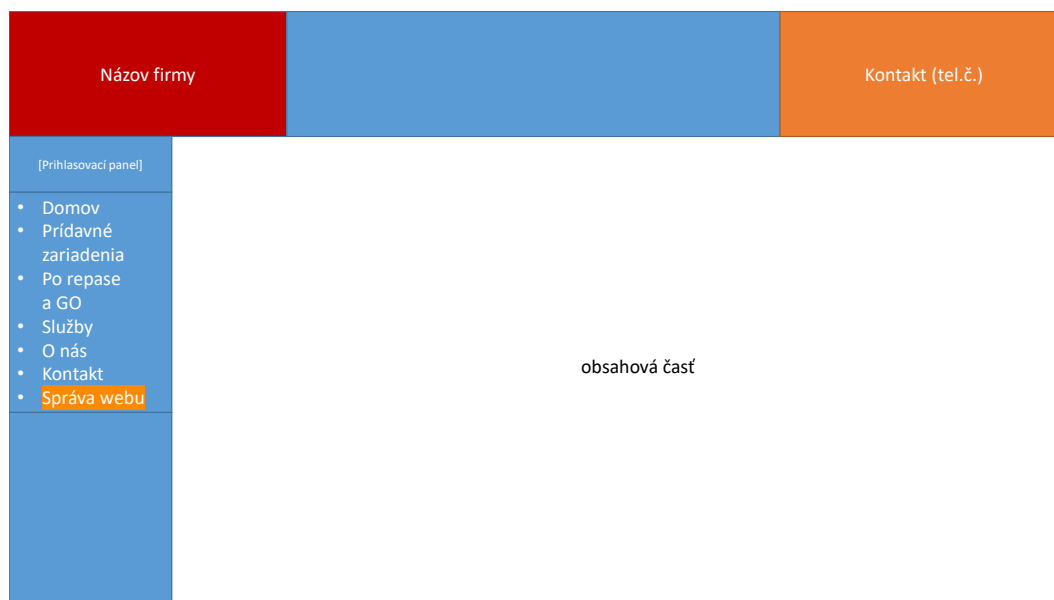
Preto je cieľom tejto práce implementovať systém spĺňajúci požiadavky P1 až P11, určený pre firmy, ktoré sa zaoberajú predajom a opravou bagrov, ktorý by majiteľom firiem umožnil sústrediť sa len na ich doménu (t. j. napr. pridávanie bagrov do ponuky) a neriešiť detaily implementácie funkcionalít a vzhľadu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

2. Návrh užívateľského rozhrania

Skôr než začneme aplikáciu programovať, tak si musíme stanoviť ako bude vyzeráť. Po prejení požiadaviek z podkap. 1.1 a prekonzultovaní s klientom bol vypracovaný návrh užívateľského rozhrania, ktorý si v tejto kapitole predstavíme. V požiadavke P1 (Roly užívateľa, viď 1.1) bolo spomenuté, že budeme chcieť zobrazovať odlišný obsah bežným zákazníkom a administrátorom. Preto bude následujúci text rozdelený na časti podľa pohľadov pre zákazníka, pre administrátora, príp. spoločný pohľad, ktorý je pre zákazníka a administrátora rovnaký (ak užívateľ nie je prihlásený, berieme ho ako neprihláseného zákazníka) alebo pohľad pre neprihlásených užívateľov.

2.1 Hlavné rozloženie aplikácie

Spoločný pohľad: Hlavné rozloženie aplikácie by malo vyzeráť tak, že v hornej ľavej časti by mal byť zobrazený názov firmy (mal by fungovať ako odkaz na domovskú stránku) a vpravo hore by malo byť zobrazené telefónne číslo firmy (malo by fungovať ako odkaz na sekciu Kontakt). Pre lepšiu predstavu hlavného rozloženia aplikácie viď obr. 2.1.



Obr. 2.1: Návrh rozloženia aplikácie.

V ľavej časti by mala byť zobrazená navigácia. V navigácii by sa mali nachádzať odkazy „Domov“, „Prídavné zariadenia“, „Po repase a GO“, „Služby“,

„O nás“ a „Kontakt“. Nad navigáciou by sa mal nachádzať prihlasovací panel, o ktorom si viac povieme v podkapitole 2.9.

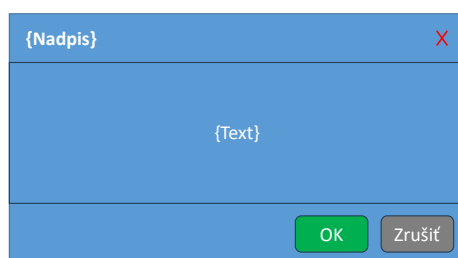
Ďalej v centre aplikácie by mal byť obsah (viď obsahovú časť v centre obr. 2.1), ktorý by sa mal meniť podľa toho, v akej časti aplikácie sa užívateľ nachádza. Všetky obrázky v nasledujúcich podkapitolách (s výnimkou grafov prechádzania medzi časťami aplikácie a modálnym oknom, o ktorom si povieme v nasledujúcej podkapitole) budú predstavovať obsahovú časť ak sa explicitne nepovie inak.

Pohľad pre administrátorov: Bolo spomenuté, že v navigácii by sa mali nachádzať odkazy „Domov“, „Prídavné zariadenia“, „Po repase a GO“, „Služby“, „O nás“ a „Kontakt“. Okrem týchto odkazov by mal mať administrátor zobrazený aj odkaz „Správa webu“.

2.2 Modálne potvrdzovacie okno

Spoločný pohľad: Modálne okno [1] je vyskakovacie dialógové okno, ktoré núti užívateľa k interakcii s ním. Kým užívateľ nevykoná interakciu s oknom, nemôže interagovať s rodičovkou aplikáciou (rodičovským oknom).

V nasledujúcich podkapitolách bude v spojení s mazaním položiek viackrát spomenuté modálne potvrdzovacie okno. Pre jeho lepšiu predstavu viď obr. 2.2.



Obr. 2.2: Potvrdzovacie modálne okno.

Okno by malo obsahovať vo vrchnej časti nadpis a v centre text oznamujúci akciu. V spodnej časti by sa mali nachádzať tlačidlá „OK“ (pre potvrdenie akcie)

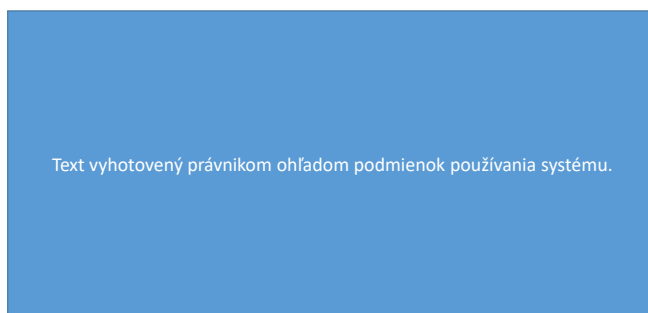
a „Zrušiť“ (pre zrušenie akcie).

2.3 Sekcie Podmienky používania a Zásady ochrany osobných údajov

Z požiadaviek P6 (užívateľ sa môže zaregistrovať do systému, a tým poskytnúť systému svoje údaje), P3 a P8 (užívateľ poskytuje osobné informácie pri odosielaní dopytu), P4 a P5 (užívateľ ponúka sumy do aukcie a následne po vyhodnotení aukcie môžu byť jeho údaje zverejnené v rozposlaných emailoch) vieme, že systém má zhromažďovať a pracovať s užívateľskými údajmi. Preto by bolo vhodné systém pripraviť na splnenie GDPR [2]. GDPR predstavuje regulácie na ochranu užívateľa pred zneužitím jeho údajov.

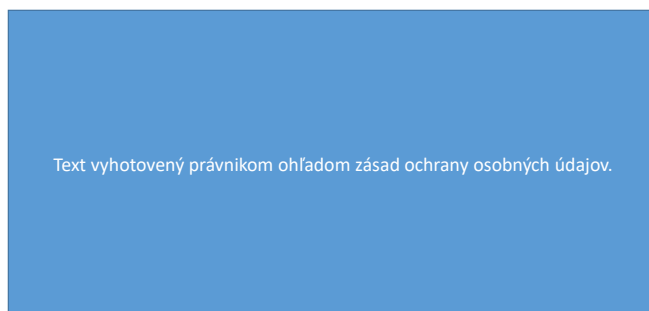
Pre splnenie GDPR by bolo dobré vyhradiť časti aplikácie, kde sa budú môcť vložiť texty ohľadom podmienok používania aplikácie a zásad ochrany osobných údajov vyhotovené právnickou osobou. Pre lepšiu predstavu týchto častí aplikácie viď obr. 2.3 a 2.4.

Podmienky používania



Obr. 2.3: Časť aplikácie s podmienkami používania.

Zásady ochrany osobných údajov



Obr. 2.4: Časť aplikácie so zásadami ochrany osobných údajov.

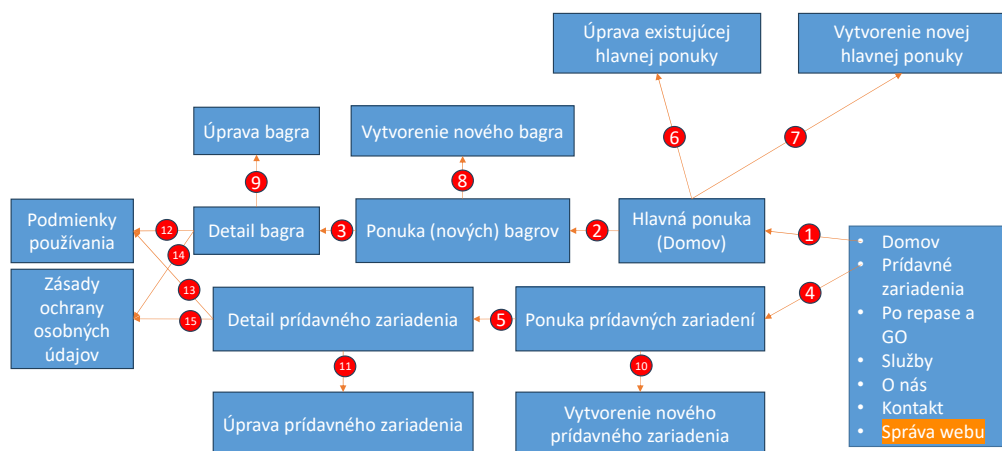
V časti aplikácie s podmienkami používania by mal byť nadpis „Podmienky používania“. V časti aplikácie so zásadami ochrany osobných údajov by mal byť nadpis „Zásady ochrany osobných údajov“. V prípade oboch častí aplikácie platí, že pod názvom by sa mal nachádzať text vyhotovený právnickou osobou na danú tému.

Na tieto časti aplikácie by sa malo odkazovať pri registrácii užívateľa, pri odosielaní dopytu z detaila bagra, detaila prídavného zariadenia alebo zo sekcie Služby, podobne pri ponúkaní sumy do aukcie z detailu aukčnej ponuky. Viac si o týchto častiach aplikácie povieme v nasledujúcom texte.

Okrem toho by mal systém umožniť užívateľovi vymazať svoje dáta zo systému, a taktiež ich na vyžiadanie exportovať. Toto sa splní v časti aplikácie s profilom užívateľa, ale o nej si povieme viac v časti Profil (viď 2.9.4).

2.4 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 vymazať (z časti preto, lebo v prípade bagrov si neskôr ukážeme aj iný spôsob). Každá časť aplikácie, ktorú si spomenieme v tejto časti textu, a ktorú môžeme vidieť na obr. 2.5, si opíšeme detailnejšie neskôr v nasledujúcich častiach textu.



Obr. 2.5: Prechádzanie medzi časťami aplikácie spĺňajúcimi požiadavky P2 a P3.

Spoločný pohľad: Pre lepšie pochopenie prechádzania medzi jednotlivými časťami programu viď obr. 2.5. Pri prvotnom príchode do aplikácie (alebo po kliknutí na odkaz „Domov“ v navigácii, viď hranu 1 na obr. 2.5) má byť užívateľovi zobrazená sekcia Hlavná ponuka s vylistovanými kartami hlavnej ponuky.

Na každej karte hlavnej ponuky by malo byť tlačidlo „Zobraziť“, ktoré by malo užívateľa presmerovať k vylistovaným kartám (nových) bagrov – Ponuka (nových) bagrov (viď hranu 2 na obr. 2.5). Kliknutím na nejakú z kariet bagrov by mal byť užívateľ presmerovaný do časti aplikácie zobrazujúcej detail vybraného bagra (viď hranu 3 na obr. 2.5).

Podobne by mal byť užívateľ schopný kliknúť na odkaz „Prídavné zariadenia“ v navigácii, ktorý by ho mal presmerovať na vylistované karty prídavných zariadení – Ponuka prídavných zariadení (viď hranu 4 na obr. 2.5). Ak užívateľ klikne na nejakú z kariet, mal by byť presmerovaný na detail vybraného prídavného zariadenia (viď hranu 5 na obr. 2.5).

Pohľad pre zákazníkov: V časti s detailom bagra by sa mal nachádzať dopytový formulár, ktorý by mal obsahovať odkaz do časti aplikácie s podmienkami používania (viď hranu 12 na obr. 2.5). Okrem neho by sa mal vo formulári nachádzať odkaz do časti so zásadami ochrany užívateľských údajov (viď hranu 13 na obr. 2.5).

V časti s detailom prídavného zariadenia by sa mal nachádzať dopytový formulár, ktorý by mal obsahovať odkaz do časti aplikácie s podmienkami používania (viď hranu 14 na obr. 2.5). Okrem neho by sa mal vo formulári nachádzať odkaz do časti so zásadami ochrany užívateľských údajov viď hranu 15 na obr. 2.5.

Pohľad pre administrátorov: Administrátorovi by sa na každej z kariet

hlavnej ponuky malo zobrazovať tlačidlo „E“, ktoré by ho malo po kliknutí presmerovať do časti aplikácie, kde by mohol danú ponuku upravovať (viď hranu 6 na obr. 2.5). Takisto by sa mal v sekcii Domov nachádzať nad kartami odkaz, ktorý by mal po kliknutí administrátora presmerovať do časti aplikácie, kde by mal byť schopný vytvoriť novú hlavnú ponuku (viď hranu 7 na obr. 2.5).

V časti s vylistovanými bagrami (Ponuka (nových) bagrov) by sa mal nachádzať odkaz, ktorý by po kliknutí presmeroval administrátora do časti aplikácie, kde by mohol vytvoriť nový bager (viď hranu 8 na obr. 2.5). Ďalej z časti aplikácie zobrazujúcej detail bagra by sa mal byť administrátor schopný dostať kliknutím na tlačidlo „Upraviť“ do časti aplikácie umožňujúcej upravovanie daného bagra (viď hranu 9 na obr. 2.5).

V časti s vylistovanými prídavnými zariadeniami (Ponuka prídavných zariadení) by sa mal nachádzať odkaz, ktorý by po kliknutí presmeroval administrátora do časti aplikácie, kde by mohol vytvoriť nové prídavné zariadenie (viď hranu 10 na obr. 2.5). Ďalej z časti aplikácie zobrazujúcej detail prídavného zariadenia by sa mal byť administrátor schopný dostať kliknutím na tlačidlo „Upraviť“ do časti aplikácie umožňujúcej upravovanie daného prídavného zariadenia (viď hranu 11 na obr. 2.5).

2.4.1 Hlavná ponuka

Spoločný pohľad: Po príchode do aplikácie by sa mal užívateľ ocitnúť v domovskej časti aplikácie, kde by mali byť zobrazené karty hlavných ponúk. Pre lepšiu predstavu viď obr. 2.6.



Obr. 2.6: Hlavná ponuka (domovská časť aplikácie).

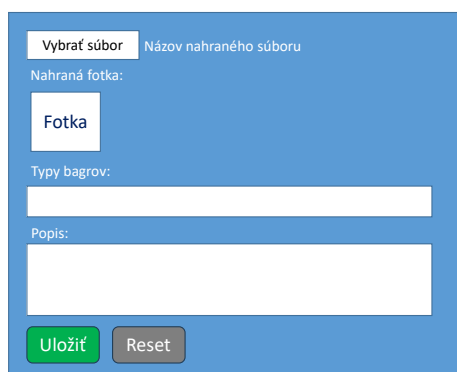
Každá z kariet by mala zobrazovať fotku reprezentujúcu danú ponuku (resp. typ bagra, ktorý je určený kombináciou kategórie a značky bagra ako bolo spomenuté v požiadavke P2.1., viď podkap. 1.1). Ak užívateľ prejde kurzorom na kartu, tak by sa mal namiesto fotky zobrazíť text opisujúci danú ponuku. Každá karta by mala obsahovať tlačidlo „Zobraziť“, ktoré by malo užívateľa po kliknutí presmerovať k ponuke (nových) bagrov typu, ktorý prezentuje daná hlavná ponuka.

Pohľad pre administrátorov: Na každej karte hlavnej ponuky by mali byť 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 by mal byť administrátor presmerovaný na formulár vyplnený údajmi danej hlavnej ponuky, kde by mohol túto ponuku upravovať. Tlačidlo „X“ by malo slúžiť na vymazanie danej hlavnej ponuky, po kliknutí naň by sa malo zobrazíť modálne potvrdzovacie okno s nadpisom „Vymazať ponuku natrvalo“ a textom „Naozaj chcete túto ponuku vymazať natrvalo?“. Ďalej po kliknutí na odkaz „Pridať novú ponuku“ by mal byť administrátor presmerovaný na prázdny formulár, kde bude môcť vytvoriť novú hlavnú ponuku.

2.4.2 Vytvorenie novej a úprava existujúcej hlavnej ponuky

Pohľad pre administrátorov: Ako bolo v predošlej časti spomenuté, administrátor by mal byť schopný z domovskej časti (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. Pre lepšiu predstavu tejto časti viď obr. 2.7.

Hlavné ponuky- vytvorenie nového záznamu



Obr. 2.7: Časť aplikácie pre vytvorenie hlavnej ponuky.

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

Formulár by mal obsahovať tlačidlá „Uložiť“ (na uloženie hlavnej ponuky) a tlačidlo „Reset“ (na vyprázdnenie formulára). Ak sú po kliknutí na tlačidlo „Uložiť“ povinné údaje nevyplnené, tak by mal na to systém administrátora upozorniť prostredníctvom chybových správ pri jednotlivých poliach.

Ďalej nad formulárom by mal byť v prípade vytvárania novej hlavnej ponuky nadpis „Hlavné ponuky – vytvorenie nového záznamu“. V prípade úpravy existujúcej hlavnej ponuky by tam mal byť nadpis „Hlavné ponuky – úprava existujúceho záznamu“.

2.4.3 Ponuka (nových) bagrov

Spoločný pohľad: Po kliknutí na tlačidlo „Zobraziť“ nejakej z hlavných ponúk by sa mali užívateľovi vylistovať karty strojov typu, ktorý prezentovala vybraná hlavná ponuka (viď obr. 2.8). Medzi vylistovanými bagrami nemajú byť bagre určené iba pre aukciu.



Obr. 2.8: Ponuka bagrov.

Nad kartami bagrov by sa mal nachádzať nadpis v tvare „{Kategória vybraného typu bagrov} – {Značka vybraného typu bagrov}“ (typ bagera určuje jeho kategóriu a značku). Po kliknutí na nejakú z kariet by mal byť užívateľ presmerovaný na detail bagera.

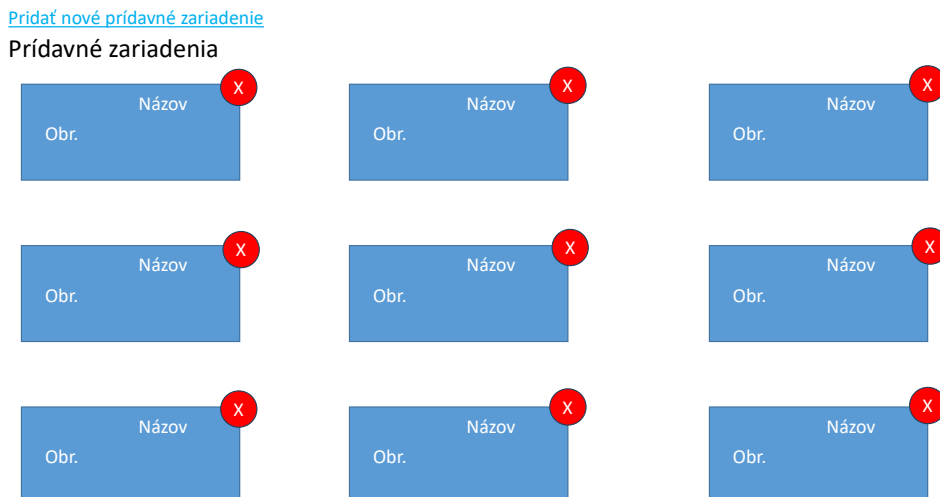
Pohľad pre administrátorov: Vo vrchnej časti by sa mal nachádzať odkaz „Pridať nový bager“, ktorý by mal užívateľa po kliknutí presmerovať k formuláru pre vytvorenie nového stroja. Ak sa administrátor dostal k formuláru touto cestou, mal by mať vo formulári predvyplnený typ stroja podľa typu vylistovaných strojov.

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

2.4.4 Ponuka prídavných zariadení

Spoločný pohľad: Po kliknutí na „Prídavné zariadenia“ v navigácii (viď vľavo na obr. 2.1) by sa mali užívateľovi vylistovať karty prídavných

zariadení. Pre lepšiu predstavu tejto časti aplikácie viď obr. 2.9.



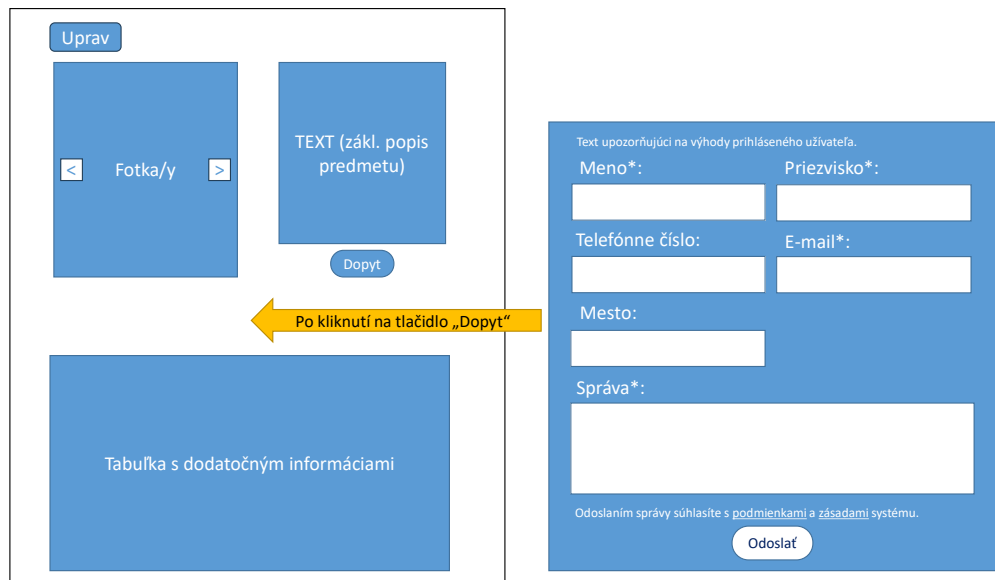
Obr. 2.9: Ponuka prídavných zariadení.

Nad kartami prídavných zariadení by sa mal nachádzať nadpis „Prídavné zariadenia“. Ďalej po kliknutí na nejakú z kariet by mal byť užívateľ presmerovaný na detail prídavného zariadenia.

Pohľad pre administrátorov: Vo vrchnej časti by sa mal nachádzať odkaz „Pridať nové prídavné zariadenie“, ktorý by mal administrátora po kliknutí presmerovať k formuláru pre vytvorenie nového prídavného zariadenia. Ďalej každá z kariet by mala mať v pravom hornom rohu tlačidlo „X“, ktoré by malo umožniť dané prídavné zariadenie vymazať. Po kliknutí na tlačidlo „X“ by sa malo zobraziť modálne potvrdzovacie okno s nadpisom „Vymazať prídavné zariadenie natrvalo“ a textom „Naozaj chcete toto prídavné zariadenie vymazať natrvalo?“.

2.4.5 Detail bagra, prídavného zariadenia

Spoločný pohľad: Po kliknutí na nejakú z kariet bagra alebo prídavného zariadenia by sa mala užívateľovi zobraziť časť aplikácie s detailom vybraného predmetu. Pre lepšiu predstavu tejto časti viď obr. 2.10.



Obr. 2.10: Detail bagra/prídavného zariadenia.

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

Pohľad pre zákazníkov: V tejto časti by sa malo nachádzať tlačidlo „Dopyt“. Po kliknutí naň by sa mal zobraziť formulár, ktorým bude môcť užívateľ odoslať dopyt na daný predmet. Po opätovnom kliknutí na tlačidlo by sa mal formulár zatvoriť.

Pohľad pre administrátorov: V hornej časti by sa malo nachádzať tlačidlo „Upraviť“. Po kliknutí naň by mal byť administrátor presmerovaný na formulár vybraného bagra (resp. prídavného zariadenia). Pomocou formulára by mal byť administrátor schopný daný bager (resp. prídavné zariadenie) upravovať.

2.4.6 Dopyt

Pohľad pre zákazníkov: V predošlej časti textu bolo spomenuté, že v detaile bagra alebo prídavného zariadenia by sa malo nachádzať tlačidlo „Dopyt“, ktorým by si mal byť zákazník schopný zobrazit' (a opätovným kliknutím schovať) formulár pre odosielanie dopytu na daný predmet. Pre lepšiu predstavu formulára viď pravú časť obrázka 2.10.

Formulár by mal obsahovať:

- *Pre neprihláseného zákazníka:* Povinné polia meno, priezvisko, email, správa, a takisto nepovinné polia telefón, mesto. Okrem polí by mal v sebe formulár obsahovať správu upozorňujúcu na to, že prihlásený užívateľ nemusí vyplňať svoje osobné údaje, a takisto upozornenie, že odoslaním správy súhlasí s podmienkami a zásadami aplikácie. Upozornenie by malo zároveň fungovať ako odkaz do časti aplikácie s podmienkami používania, a tiež ako odkaz do časti so zásadami ochrany osobných údajov (obe časti boli spomenuté v podkap. 2.3).
- *Pre prihláseného zákazníka:* Povinné pole správa.

Povinné polia by mali byť označené hviezdíčkou.

Ďalej by mal formulár obsahovať tlačidlo „Odoslať“, ktoré by malo zákazníkovi umožniť odoslanie dopytu. Ak by boli po kliknutí na tlačidlo „Odoslať“ povinné údaje nevyplnené, tak by mala na to aplikácia užívateľa upozorniť chybovými správami pri jednotlivých poliach.

2.4.7 Vytvorenie nového a úprava existujúceho bagra

Pohľad pre administrátorov: Kliknutím na odkaz „Pridať nový bager“, ktorý by sa mal nachádzať v časti s vylistovanými (novými) bagrami, by mal byť administrátor presmerovaný do časti aplikácie, kde bude môcť vytvoriť nový bager. Pre lepšiu predstavu tejto časti aplikácie viď obr. 2.11.

Bagre- vytvorenie nového záznamu

Vybrať súbory Počet súborov: 2

Nahrané fotky:

Fotka X Fotka X ...

Názov:

Popis:

Je IBA pre aukciu

Náhradné diely v bagri:

Vybrať	Katalógové číslo	Názov
<input type="checkbox"/>

Typ bagrov:

Špecifické vlastnosti:

Názov vlastnosti	Hodnota
...	<input type="text"/>

Obr. 2.11: Časť aplikácie pre vytvorenie bagra.

Podobne kliknutím na tlačidlo „Upraviť“, ktoré by sa malo nachádzať v časti s detailom bagra, by mal byť administrátor presmerovaný do časti aplikácie, kde by mal byť schopný upravovať daný bager. V nasledujúcom opise uvidíme, že obe časti aplikácie (časť pre vytváranie a časť pre upravovanie bagra) sú si veľmi podobné.

Obidve časti by mali obsahovať formulár umožňujúci vloženie viacerých fotiek, názvu bagra, zadanie jeho popisu, uvedenie, či je stroj určený iba pre aukciu, tiež by mal byť administrátor schopný vybrať, ktoré náhradné diely patria do daného bagra, a takisto vybrať z možností typov bagrov. Podľa vybraného typu bagra by sa mali zobrazíť vlastnosti ním určené a administrátor by mal byť schopný vyplniť ich hodnoty. Fotky, názov bagra a jeho typ patria medzi povinné údaje.

Navyše po vložení by sa mali fotky a ich počet zobrazíť vo formulári. V pravom hornom rohu každej zobrazenej fotky by sa malo nachádzať tlačidlo „X“, ktorým by mal byť administrátor schopný fotku odstrániť.

Formulár by mal taktiež obsahovať tlačidlá „Uložiť“ (na uloženie bagra) a „Reset“ (na vyprázdnenie formulára). Ak by boli po kliknutí na tlačidlo „Uložiť“ povinné údaje nevyplnené, tak by mal na to systém administrátora upozorniť prostredníctvom chybových správ pri jednotlivých poliach.

Navyše nad formulárom by mal byť v prípade vytvárania nového bagra nadpis „Bagre – vytvorenie nového záznamu“. V prípade úpravy existujúceho bagra by tam mal byť nadpis „Bagre – úprava existujúceho záznamu“.

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

Pohľad pre administrátorov: V predošlom texte bolo napísané, že administrátor by sa mal byť schopný dostať cez karty hlavných ponúk 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 by mala byť 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 2.8.

2.4.9 Vytvorenie nového a úprava existujúceho prídavného zariadenia

Pohľad pre administrátorov: Kliknutím na odkaz „Pridať nové prídavné zariadenie“, ktorý by sa mal nachádzať v časti s vylistovanými prídavnými zariadeniami, by mal byť administrátor presmerovaný do časti aplikácie, kde by mal byť schopný vytvoriť nové prídavné zariadenie. Pre lepšiu predstavu tejto časti viď obr. 2.12.

The screenshot shows a web form titled "Prídavné zariadenia- vytvorenie nového záznamu". At the top, there is a file selection area with a button "Vybrať súbory" and a status "Počet súborov: 2". Below this, under "Nahrané fotky:", two image thumbnails labeled "Fotka" are shown, each with a red 'X' icon in its top right corner. The form contains several input fields: "Pre akú kategóriu bagrov:" (empty), "Kategória:" (empty), "Značka:" (empty), "Názov:" (empty), and "Popis:" (empty). At the bottom, there are two buttons: a green "Uložiť" button and a grey "Reset" button.

Obr. 2.12: Časť aplikácie pre vytvorenie prídavného zariadenia.

Podobne kliknutím na tlačidlo „Upraviť“, ktoré by sa malo nachádzať v časti s detailom prídavného zariadenia, by mal byť administrátor presmerovaný do časti aplikácie, kde by mal byť schopný upravovať dané prídavné zariadenie. Ako uvidíme v nasledujúcom opise, obe časti aplikácie (časť pre vytváranie a časť pre upravovanie prídavného zariadenia) sú si veľmi podobné.

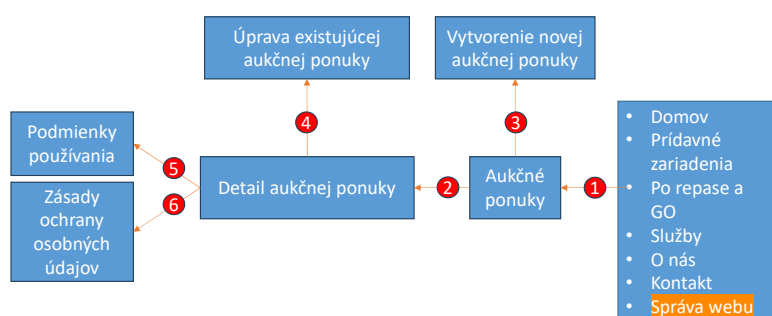
Obe časti by mali obsahovať formulár umožňujúci vloženie viacerých fotiek, vybranie kategórie bagrov (pre ktoré je prídavné zariadenie určené), vybranie jeho kategórie, vybranie značky, a takisto zadanie názvu a popisu prídavného zariadenia. Okrem popisu sú všetky údaje povinné. Navyše po vložení by sa mali fotky a ich počet zobrazíť vo formulári. V pravom hornom rohu každej zobrazenej fotky by sa malo nachádzať tlačidlo „X“, ktorým by mohol administrátor fotku odstrániť.

Formulár by mal taktiež obsahovať tlačidlá „Uložiť“ (na uloženie prídavného zariadenia) a „Reset“ (na vyprázdnenie formulára). Ak by boli po kliknutí na tlačidlo „Uložiť“ povinné údaje nevyplnené, tak by mal na to systém administrátora upozorniť prostredníctvom chybových správ pri jednotlivých poliach.

Navyše nad formulárom by mal byť v prípade vytvárania nového bagra nadpis „Prídavné zariadenia – vytvorenie nového záznamu“. V prípade úpravy existujúceho prídavného zariadenia by tam mal byť nadpis „Prídavné zariadenia – úprava existujúceho záznamu“.

2.5 Splnenie P4

V tejto podkapitole si prejdeme časti aplikácie splňujúce požiadavku P4, t. j. predstavenie aukčnej ponuky, časti umožňujúce administrátorovi vytvárať (a upravovať) aukčné ponuky, a takisto časti umožňujúce zákazníkom ponúkať sumy do dražby. Pre lepšie pochopenie prechádzania medzi jednotlivými časťami programu viď obr. 2.13.



Obr. 2.13: Prechádzanie medzi časťami aplikácie splňajúcimi požiadavku P4.

Spoločný pohľad: Po kliknutí na odkaz „Po repase a GO“ v navigácii by mala byť užívateľovi zobrazená časť aplikácie s vylistovanými kartami aukčnej ponuky – Aukčné ponuky (viď hranu 1 na obr. 2.13). Po kliknutí na nejakú z kariet by sa mal užívateľovi zobraziť detail aukčnej ponuky (viď hranu 2 na obr. 2.13).

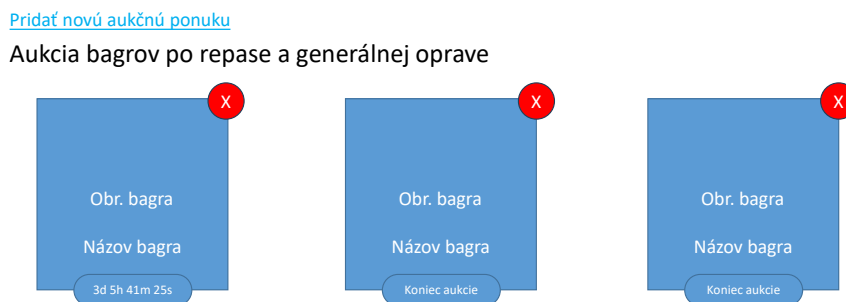
Pohľad pre zákazníkov: V časti aplikácie s detailom aukčnej ponuky by sa malo zobrazovať tlačidlo „Mám záujem!“, pomocou ktorého by si mal byť zákazník schopný zobraziť (a opätovným kliknutím schovať) formulár pre ponúknutie sumy do dražby. Vo formulári by sa mal tiež nachádzať odkaz do časti aplikácie s podmienkami používania (viď hranu 5 na obr. 2.13), a takisto aj odkaz do časti so zásadami ochrany užívateľských údajov (viď hranu 6 na obr. 2.5).

Pohľad pre administrátorov: Nad vylistovanými kartami by sa mal zobrazovať odkaz „Pridať novú aukčnú ponuku“. Pomocou tohto odkazu by sa mal byť administrátor schopný dostať do časti aplikácie s prázdny formulárom pre vytvorenie novej aukčnej ponuky (viď hranu 3 na obr. 2.13).

V hornej časti detailu aukčnej ponuky by sa malo zobrazovať tlačidlo „Upraviť“. Pomocou tohto tlačidla by sa mal byť administrátor schopný dostať do časti aplikácie, kde by mal byť schopný danú ponuku upravovať (viď hranu 4 na obr. 2.13).

2.5.1 Aukčné ponuky

Kliknutím na odkaz „Po repase a GO“ v navigácii by mal byť užívateľ presmerovaný do časti aplikácie s vylistovanými kartami aukčných ponúk. Pre lepšiu predstavu viď obr. 2.14.



Obr. 2.14: Aukčné ponuky.

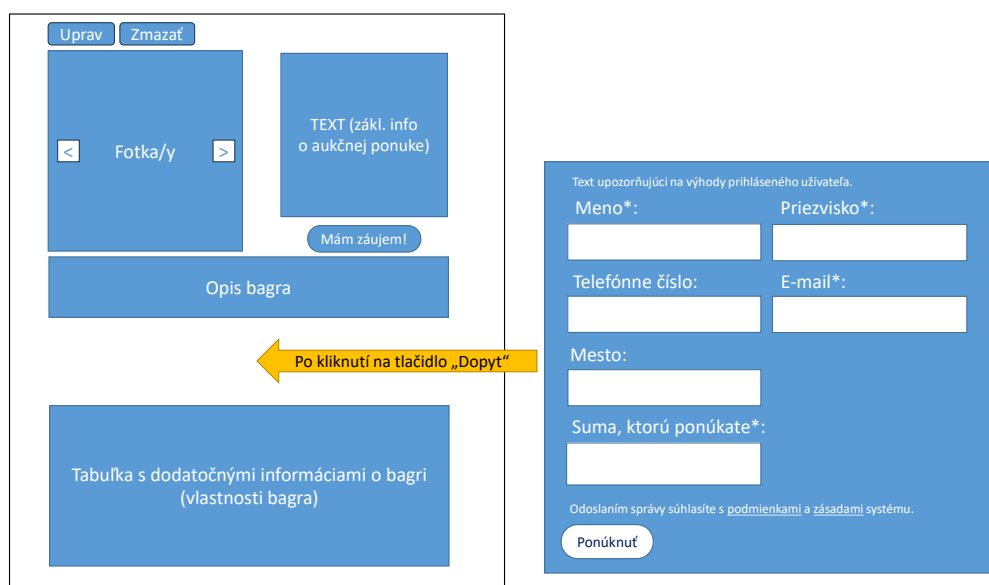
Spoločný pohľad: Karta aukčnej ponuky by mala obsahovať obrázok a názov draženého bagra. V spodnej časti by sa mal nachádzať odpočet do konca dražby. Po kliknutí na nejakú z kariet by mal byť užívateľ presmerovaný na detail vybranej aukčnej ponuky. Nad kartami by sa mal nachádzať nadpis „Aukcia bagrov po repase a generálnej oprave“.

Pohľad pre administrátorov: V pravej hornej časti karty aukčnej ponuky by malo byť tlačidlo „X“, ktorým by mal byť administrátor schopný ponuku vymazať. Po kliknutí naň by sa malo zobraziť modálne potvrdzovacie okno s nadpisom „Vymazať aukčnú ponuku natrvalo“ a textom „Naozaj chcete túto aukčnú ponuku vymazať natrvalo?“.

Vo vrhnej časti s vylisťovanými kartami aukčnej ponuky by sa mal nachádzať odkaz „Pridať novú aukčnú ponuku“. Po kliknutí na odkaz by mal byť administrátor presmerovaný do časti aplikácie, kde by mohol vytvoriť novú aukčnú ponuku.

2.5.2 Detail aukčnej ponuky

Po kliknutí na nejakú z kariet aukčnej ponuky by mal byť užívateľ presmerovaný do časti detailu aukčnej ponuky. Pre lepšiu predstavu viď obr. 2.15.



Obr. 2.15: Detail aukčnej ponuky.

Spoločný pohľad: V tejto časti by sa mali nachádzať fotky draženého bagra (klikaním na tlačidlá šípok vpravo, vľavo by sa malo dať prechádzať medzi fotkami predmetu). Ďalej by sa mali 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 nezávislé od toho, že bol opravený), ďalej by sa v tejto časti mala nachádzať tabuľka s vlastnosťami bagra (napr. výška, šírka atď.).

Pohľad pre zákazníkov: Okrem toho by sa malo v tejto časti ešte nachádzať tlačidlo „Mám záujem!“. Po kliknutí naň by sa mal zobrazíť formulár, ktorým by mal byť užívateľ schopný ponúknuť sumu do dražby. Po opätovnom kliknutí na tlačidlo sa má formulár zatvoriť.

Pohľad pre administrátorov: V hornej časti detailu aukčnej ponuky by sa malo nachádzať tlačidlo „Upraviť“. Po kliknutí naň by mal byť administrátor presmerovaný na formulár danej aukčnej ponuky, ktorým by ju mal byť schopný

upravovať. Okrem tlačidla „Upraviť“ by sa v hornej časti malo nachádzať aj tlačidlo „Zmazať“, ktorým by mal byť schopný danú aukčnú ponuku zmazať. Po kliknutí na tlačidlo „Zmazať“ by sa malo zobraziť modálne potvrdzovacie okno s nadpisom „Vymazať“ a textom „Naozaj chcete vymazať túto aukčnú ponuku?“.

2.5.3 Ponúkание sumy do dražby

Pohľad pre zákazníkov: V predošlej časti textu bolo spomenuté, že v detaile bagra alebo prídavného zariadenia by sa malo nachádzať tlačidlo „Mám záujem!“, ktorým by si mal byť zákazník (nie administrátor) schopný zobraziť (a opätovným kliknutím schovať) formulár pre ponúkание sumy do dražby daného bagra. Pre lepšiu predstavu formulára viď pravú časť obrázka 2.15.

Formulár by mal obsahovať:

- *Pre neprihláseného zákazníka:* Povinné polia meno, priezvisko, email, pole pre ponúkanú sumu, a takisto nepovinné polia telefón, mesto. Okrem polí by mal v sebe formulár obsahovať aj správu upozorňujúcu na to, že prihlásený užívateľ nemusí vyplňať svoje osobné údaje, a takisto aj upozornenie, že s odoslaním správy súhlasí s podmienkami a zásadami aplikácie. Upozornenie by malo zároveň fungovať ako odkaz do časti aplikácie s podmienkami používania, a tiež ako odkaz do časti so zásadami ochrany osobných údajov (obe časti boli spomenuté v podkap. 2.3).
- *Pre prihláseného zákazníka:* Povinné pole pre ponúkanú sumu.

Povinné polia by mali byť označené hviezdíčkou. Navyše má platiť, že prvá zákazníkovi ponúkaná suma môže byť rovná počiatočnej sume, ale nasledujúce ponúkané sumy musia mať medzi sebou rozdiel aspoň 100 eur.

Ďalej by mal formulár obsahovať tlačidlo „Ponúknuť“, ktoré by malo zákazníkovi umožniť ponúkание sumy do dražby daného bagra. Ak by boli po kliknutí na tlačidlo „Ponúknuť“ povinné údaje nevyplnené alebo ak by boli ich hodnoty neplatné, tak by mala na to aplikácia užívateľa upozorniť chybovými správami pri jednotlivých poliach.

2.5.4 Vytvorenie novej a úprava existujúcej aukčnej ponuky

Pohľad pre administrátorov: Kliknutím na odkaz „Pridať novú aukčnú ponuku“, ktorý by sa mal nachádzať v časti s vylistovanými aukčnými ponukami, by mal byť administrátor presmerovaný do časti aplikácie, kde by mal byť schopný vytvoriť novú aukčnú ponuku. Pre lepšiu predstavu tejto časti aplikácie viď obr. 2.16.

Aukčné ponuky- vytvorenie nového záznamu

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

Obr. 2.16: Časť aplikácie pre vytvorenie aukčnej ponuky.

Podobne kliknutím na tlačidlo „Upraviť“, ktoré by sa malo nachádzať v časti s detailom aukčnej ponuky, by mal byť administrátor presmerovaný do časti aplikácie, kde bude môcť danú aukčnú ponuku upravovať. Ako uvidíme v nasledujúcom opise, obe časti aplikácie (časť pre vytváranie a časť pre upravovanie aukčnej ponuky) sú si veľmi podobné.

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

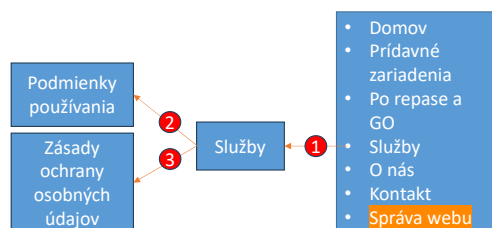
Formulár by mal taktiež obsahovať tlačidlo „Reset“ na vyprázdenie formulára. Okrem tlačidla „Reset“ by mal formulár obsahovať aj tlačidlo „Uložiť“ (na uloženie aukčnej ponuky) ak sa vytvára nová aukčná ponuka alebo upravuje existujúca aukčná ponuka, ktorá je ešte stále aktívna (t. j. stále prebieha dražba). V prípade,

že by administrátor upravoval existujúcu aukčnú ponuku, ktorá už nie je aktívna, pretože bager bol vydražený, tak by sa malo zobraziť tlačidlo „Uložiť a spustiť odznova“. Toto správanie je vyžadované preto, lebo ak by výherca dražby z nejakého dôvodu nesúhlasil s prevzatím bagra, administrátor by mohol kliknutím na tlačidlo „Uložiť a spustiť odznova“ celú dražbu odštartovať odznova (teda ako novo vytvorenú, bez účastníkov a ich ponúknutých súm). Ak by boli pri ukladaní povinné údaje nevyplnené alebo by boli ich hodnoty neplatné, tak by mal na to systém administrátora upozorniť prostredníctvom chybových správ pri jednotlivých poliach.

Navyše nad formulárom by mal byť v prípade vytvárania novej aukčnej ponuky nadpis „Aukčné ponuky – vytvorenie nového záznamu“. V prípade úpravy existujúcej aukčnej ponuky by mal byť nadpis „Aukčné ponuky – úprava existujúceho záznamu“.

2.6 Splnenie P8

Spoločný pohľad: Po kliknutí na odkaz „Služby“ v navigácii by mal byť užívateľ presmerovaný do časti aplikácie, kde by mali byť opísané výkopové služby, ktoré firma ponúka. Pre lepšiu predstavu prechodu medzi časťami aplikácie viď obr. 2.17.



Obr. 2.17: Prechádzanie medzi časťami aplikácie spĺňajúcimi požiadavku P8.

Pohľad pre zákazníkov: V tejto časti aplikácie by sa malo nachádzať tlačidlo „Mám záujem o službu!“. Kliknutím na tlačidlo „Mám záujem o službu!“

by sa mal zobrazit formulár, ktorým by mal byť užívateľ schopný požiadať o výkopovú službu. Pre lepšiu predstavu formulára viď pravú časť obr. 2.18.

The diagram illustrates the 'Služby' (Services) section of a user interface. It shows a transition from a simplified view to a detailed form. The simplified view on the left features a blue box with the title 'Služby', a 'TEXT' placeholder, and a button labeled 'Mám záujem o službu!'. A yellow arrow points from this button to the right, where the detailed form is displayed. This form includes a blue header with a warning text: 'Text upozorňujúci na výhody prihláseného užívateľa.' Below the header are several input fields: 'Meno*:' (Name), 'Priezvisko*:' (Surname), 'Telefónne číslo:' (Phone number), 'E-mail*:' (Email), 'Mesto:' (City), and 'Správa*:' (Address). At the bottom of the form, there is a disclaimer: 'Odoslaním správy súhlasíte s podmienkami a zásadami systému.' and a button labeled 'Odoslať' (Send).

Obr. 2.18: Sekcia Služby.

Opätovným kliknutím na tlačidlo by sa mal formulár schovať. Formulár by mal obsahovať:

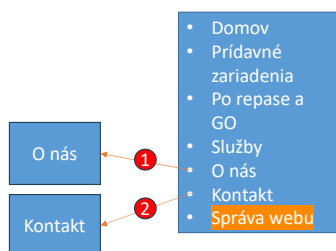
- *Pre neprihláseného zákazníka:* Povinné polia meno, priezvisko, email, správa, a takisto nepovinné polia telefón, mesto. Okrem polí by mal v sebe formulár obsahovať správu upozorňujúcu na to, že prihlásený užívateľ nemusí vyplňať svoje osobné údaje, a takisto by mal formulár obsahovať upozornenie, že odoslaním správy súhlasí s podmienkami a zásadami aplikácie. Upozornenie by malo zároveň fungovať ako odkaz do časti aplikácie s podmienkami používania (viď hranu 2 na obr. 2.17), a tiež ako odkaz do časti so zásadami ochrany osobných údajov (viď hranu 3 na obr. 2.17), obe časti boli spomenuté v podkap. 2.3.
- *Pre prihláseného zákazníka:* Povinné pole správa.

Povinné polia by mali byť označené hviezdíčkou.

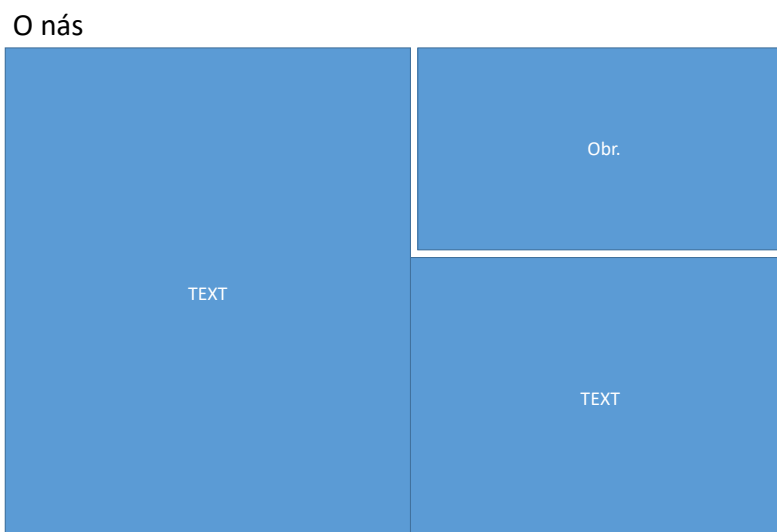
Ďalej by mal formulár obsahovať tlačidlo „Odoslať“, ktoré by malo zákazníkovi umožniť odoslanie žiadosti o službu. Ak by boli po kliknutí na tlačidlo „Odoslať“ povinné údaje nevyplnené, tak by na to mala aplikácia užívateľa upozorniť chybovými správami pri jednotlivých poliach.

2.7 Splnenie P9

Spoločný pohľad: Po kliknutí na odkaz „O nás“ v navigácii by mal byť užívateľ presmerovaný do časti aplikácie s informáciami o firme (viď hranu 1 na obr.2.19). Pre lepšiu predstavu sekcie „O nás“ viď obr. 2.20.



Obr. 2.19: Prechádzanie medzi časťami aplikácie spĺňajúcimi požiadavku P9.



Obr. 2.20: Sekcia O nás.

Sekcia „O nás“ by mala obsahovať text o firme (napr. jej histórii), príp. nejaké fotky týkajúce sa firmy. Ďalej po kliknutí na odkaz „Kontakt“ v navigácii alebo po kliknutí na telefónne číslo v hornej časti aplikácie (viď vpravo hore na obr. 2.1) by mal byť užívateľ presmerovaný do časti aplikácie s kontaktom na firmu, príp. pridružené firmy (viď hranu 2 na obr.2.19). Pre lepšiu predstavu sekcie „Kontakt“ viď obr. 2.21.

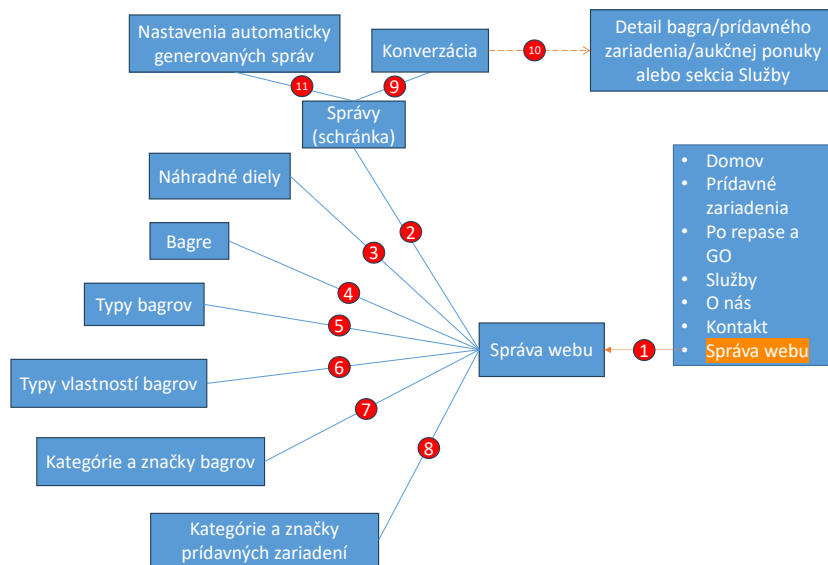


Obr. 2.21: Sekcia Kontakt.

Sekcia „Kontakt“ by mala obsahovať tabuľku s kontaktom (telefónne číslo a emailová adresa) firmy. Okrem kontaktu klientskej firmy by mohla tabuľka obsahovať aj kontakty na iné (pridružené) firmy.

2.8 Splnenie P5, P7 a správa predmetov

Pohľad pre administrátorov: Po kliknutí na odkaz „Správa webu“ v navigácii, ktorý by mal byť viditeľný iba pre administrátorov, by mal byť administrátor presmerovaný do sekcie Správa webu (viď hranu 1 na obr. 2.22). Tá by mala vyzeráť nasledovne: v hornej časti by sa mal nachádzať panel s kartami, ktorými by sa mal byť administrátor schopný preklikávať (viď hrany 2 až 9 na obr. 2.22), a pod panelom by sa mal nachádzať obsah podľa toho, na akej karte sa administrátor nachádza.



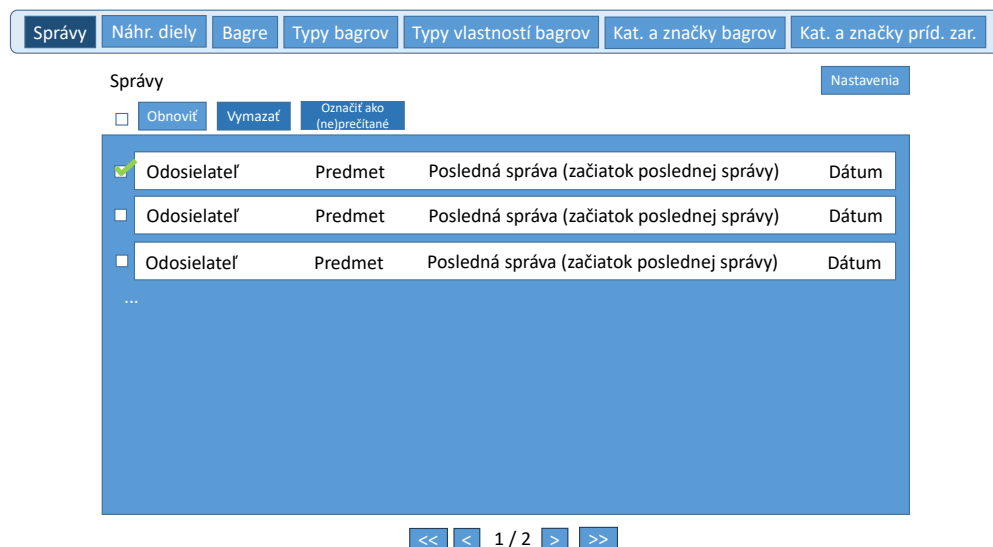
Obr. 2.22: Prechádzanie medzi časťami aplikácie spĺňajúcimi požiadavku P5 (orientovaná hrana značí prechod do novej časti aplikácie, neorientovaná prepínanie medzi časťami, prerušovaná nepovinný prechod do novej časti).

Po príchode do tejto časti by sa mala administrátorovi zobrazíť prvá karta – „Správy“ (viď hranu 2 na obr. 2.22). Okrem tejto karty by sa mal byť administrátor schopný prekliknúť na karty: „Náhradné diely“ (hrana 3 na obr. 2.22), „Bagre“ (hrana 4 na obr. 2.22), „Typy bagrov“ (hrana 5 na obr. 2.22), „Typy vlastností bagrov“ (hrana 6 na obr. 2.22), „Kategórie a značky bagrov“ (hrana 7 na obr. 2.22), „Kategórie a značky prídavných zariadení“ (hrana 8 na obr. 2.22).

Na karte „Správy“ by mali byť vylistované konverzácie. Po kliknutí na nejakú z konverzácií by sa mala administrátorovi zobrazíť konverzácia (viď hranu 9 na obr. 2.22). Ak by bola konverzácia odoslaná z nášeho systému, malo by byť v tejto konverzácii vložené prepojenie do časti aplikácie z ktorej bola správa odoslaná (viď hranu 10 na obr. 2.22). Mohla by to byť časť aplikácie s detailom bagra, detailom prídavného zariadenia, detailom aukčnej ponuky alebo sekcia Služby. Okrem toho by sa mal byť administrátor schopný dostať z karty „Správy“ (pomocou tlačidla „Nastavenia“, ktoré by sa malo na karte nachádzať) do časti aplikácie s nastaveniami automaticky generovaných správ (viď hranu 11 na obr. 2.22).

2.8.1 Karta Správy – schránka

Pohľad pre administrátorov: Po príchode do sekcie „Správa webu“ alebo po kliknutí na kartu „Správy“ v sekcii „Správa webu“ by sa mala administrátorovi zobraziť emailová schránka podobná Gmailu. Pre lepšiu predstavu tejto časti aplikácie viď obr. 2.23.



Obr. 2.23: Emailová schránka.

V schránke by mali byť vylistované konverzácie (vlákna správ) zhora dole od najnovšej po najstrašiu. Každá takáto konverzácia by mala byť reprezentovaná riadkom, v ktorom by malo byť uvedené meno odosielateľa, predmet a text poslednej správy (resp. jeho začiatok v prípade dlhšieho textu), a dátum poslednej správy (emailu v konverzácii). Vedľa každého riadka konverzácie by malo byť začiarkovacie políčko, ktorým vieme konverzáciu označiť. Po kliknutí na nejaký z riadkov by sa mala administrátorovi zobraziť vybraná konverzácia.

Vylistované konverzácie by mali byť stránkované (max. osem konverzácií na stránku). Pod vylistovanými konverzáciami by malo byť tlačidlo „«“, ktoré by malo byť schopné presunúť administrátora na prvú stranu vylistovaných konverzácií, tlačidlo „<“, ktoré by malo byť schopné presunúť administrátora o jednu stranu dopredu a tlačidlá „>“, „»“, ktoré by mali fungovať analogicky (lenže na opačnú stranu). Ak by sa administrátor nachádzal na prvej strane, tak tlačidlá „«“ a „<“ by nemali byť aktívne (nemalo by sa na ne dať kliknúť). Podobne ak by sa administrátor nachádzal na poslednej strane, tlačidlá „>“ a „»“ by nemali byť aktívne.

Nad vylistovanými konverzáciami by mal byť nadpis „Správy“ a tlačidlo „Nastavenia“, ktorým by sa mal byť administrátor schopný dostať do časti aplikácie

s nastaveniami automaticky generovaných správ. Ďalej pod nadpisom by malo byť začiarkovacie políčko, ktorým by mal byť administrátor schopný začiarknúť (označiť) všetky konverzácie na aktuálnej strane. Okrem tohto začiarkovacieho políčka, by tam malo byť tlačidlo „Obnoviť“ a ak by bola označená aspoň jedna konverzácia, tak potom by mali byť viditeľné aj tlačidlá „Vymazať“ a „Označiť ako prečítané“ alebo „Označiť ako neprečítané“.

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

Tlačidlom „Vymazať“ by mal byť administrátor schopný vymazať označené konverzácie. Po kliknutí naň sa má zobraziť potvrdzovacie modálne okno s nadpisom „Vymazať vybrané konverzácie“ a textom „Naozaj chcete vymazať vybrané konverzácie natrvalo?“.

2.8.2 Karta Správy – konverzácia

Pohľad pre administrátorov: Po kliknutí na nejakú z konverzácií by sa mala administrátorovi zobraziť daná konverzácia. Pre lepšiu predstavu tejto časti aplikácie viď obr. 2.24.

Obr. 2.24: Konverzácia.

V hornej časti by sa mal nachádzať nadpis „Správy“ a tlačidlo „Nastavenia“ (pre otvorenie nastavenia automaticky generovaných správ). Pod nimi by sa mali nachádzať tlačidlá „Späť“ (pre návrat do schránky), „Vymazať“ a tlačidlo „Označiť ako neprečítanú“ (pre označenie správy za neprečítanú). Tlačidlom „Vymazať“

by mal byť administrátor schopný vymazať otvorenú konverzáciu. Po kliknutí by sa malo zobrazíť potvrdzovacie modálne okno s nadpisom „Vymazať konverzáciu“ a textom „Naozaj chcete vymazať túto konverzáciu natrvalo?“.

Napravo od tlačidiel by mal byť zobrazený predmet (poslednej správy v konverzácii). Pod tlačidlami a predmetom by mali byť zobrazené jednotlivé správy (emaily) konverzácie zoradené zhora dole od najstaršej po najnovšiu. Ak by bola (prvá) správa odoslaná z nášho systému, tak vedľa nej by mala byť vložená časť aplikácie, odkiaľ bola správa odoslaná (na obr. 2.24 viď časť označenú červeným rámom). Po kliknutí na túto časť by mal byť administrátor presmerovaný do časti aplikácie, odkiaľ bola správa odoslaná (mala by sa otvoriť v novom okne). V prípade, že by správa nebola odoslaná z nášho systému, mala by sa správa rozťahnuť na šírku tejto časti aplikácie (rovnako ako je druhá správa, označená zeleným rámom, na obr. 2.24). Pre každú správu v konverzácii by sa malo zobrazíť meno a email odosielateľa, dátum a čas správy, a takisto text správy. Pod vylisťovanými správami by mal byť priestor pre písanie odpovede a tlačidlo „Odoslať“, ktorým by mal byť administrátor schopný odosielať správy do konverzácie.

2.8.3 Karta Správy – nastavenia automaticky generovaných správ

Pohľad pre administrátorov: Po kliknutí na tlačidlo „Nastavenia“ by sa mali administrátorovi zobrazíť nastavenia automaticky generovaných správ. Pre lepšiu predstavu tejto časti aplikácie viď obr. 2.25.

Správy Náhr. diely Bagre Typy bagrov Typy vlastností bagrov Kat. a značky bagrov Kat. a značky príd. zar.

Správy Späť

Nastavenia automaticky generovaných správ

Název automaticky gen. správy (pre akú udalosť)

Predmet:
(predvolený/vlastný text predmetu)

Text:
(predvolený/vlastný text správy)

- #tag1
- #tag2
- #tag3
- ...

Uložiť Reset

Obr. 2.25: Nastavenia automaticky generovaných správ.

Vo vrchnej časti by sa mal nachádzať nadpis „Správy“ a tlačidlo „Spät“, ktoré by malo byť schopné schovať (zatvoriť) nastavenia. Pod nimi by sa mal nachádzať nadpis „Nastavenia automaticky generovaných správ“ a sekcia, kde by sa pre každú automaticky generovanú správu mala zobrazovať časť, kde by sa dal nastavovať jej tvar (jedna zo spomínaných častí je vyznačená červeným rámom na obr. 2.25).

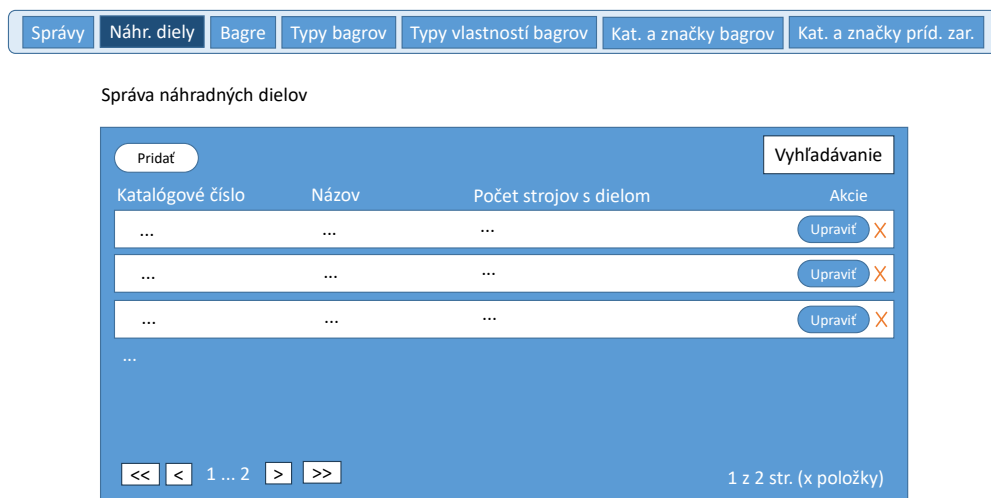
Každá z takýchto častí by mala obsahovať názov automaticky generovanej správy (resp. pre koho, akú udalosť je určená, napr. „Pre víťaza aukcie“). Ďalej by mala obsahovať šablónu predmetu, šablónu textu a v pravej časti zoznam tagov povolených v danej automaticky generovanej správe.

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

Každá z automaticky generovaných správ by mala mať predvolený text, ktorý by mal byť administrátor schopný prepísať pomocou šablón. Ak by neboli definované vlastné šablóny, mali by sa využívať predvolené, a preto by sa mali v spomínaných poliach pre predmet a text zobrazovať predvolené šablóny. Ak by boli predvolené šablóny prepísané a neskôr vymazané, tak polia by sa mali znova naplniť predvolenými hodnotami. Ďalej pod poliami by sa mali nachádzať tlačidlá „Uložiť“ (pre uloženie šablón) a „Reset“ (pre navrátenie k predvoleným hodnotám).

2.8.4 Karta Náhradné diely

Pohľad pre administrátorov: Po prekliknutí na kartu „Náhradné diely“ v sekcii Správa webu by sa mala administrátorovi zobrazíť časť aplikácie, kde by mal byť schopný spravovať (vytvárať, upravovať a mazať) náhradné diely. Pre lepšiu predstavu tejto časti viď obr. 2.26.



Obr. 2.26: Správa náhradných dielov.

Pod panelom s kartami by sa mal nachádzať nadpis „Správa náhradných dielov“, a pod ním by sa mala nachádzať tabuľka s vylistovanými záznamami náhradných dielov. Tabuľka by mala obsahovať stĺpce pre katalógové číslo, názov a počet strojov s daným náhradným dielom. Každý záznam (riadok) náhradného dielu v tabuľke by mal zobrazovať tieto údaje. Vedľa týchto stĺpcov by mal existovať aj stĺpec pre akcie, v ktorom by pre každý záznam boli zobrazené dve tlačidlá - „Upraviť“ a „X“.

Tlačidlo „X“ slúži na vymazanie záznamu. Po jeho stlačení by sa malo zobrazíť modálne potvrdzovacie okno s nadpisom „Vymazať záznam“ a textom „Naozaj chcete tento záznam vymazať natrvalo?“. Ďalej po kliknutí na tlačidlo „Upraviť“ alebo po dvojitém kliknutí na záznam (riadok v tabuľke) by sa mal zobrazíť formulár ako modálne okno s vyplnenými údajmi o vybranom náhradnom diele. Pomocou tohto formulára by mal byť administrátor schopný daný náhradný diel upravovať. Pre lepšiu predstavu formulára viď obr. 2.27.

Obr. 2.27: Formulár náhradného dielu.

Formulár náhradného dielu by mal obsahovať polia pre katalógové číslo a názov náhradného dielu, ktoré sú povinné. Taktiež by mal formulár obsahovať vylistované bagre. Vedľa názvu bagra by malo existovať začiarkovacie políčko. Ak stroj obsahuje daný náhradný diel, mal by administrátor políčko začiarknuť. Ďalej formulár obsahuje tlačidlá „Uložiť“ (pre uloženie zmien) a „Reset“ (pre vyprázdenie obsahu formulára). Ak by nejaké z polí po kliknutí na tlačidlo „Uložiť“ obsahovalo neplatné hodnoty, napr. by bolo nejaké z povinných polí prázdne, tak by sa pri danom poli mala zobraziť odpovedajúca chybová správa.

Podobne by sa mal po kliknutí na tlačidlo „Pridať“, ktoré by sa malo nachádzať v ľavej hornej časti tabuľky, zobrazíť prázdny formulár náhradného dielu ako modálne okno, kde by mal byť administrátor schopný vytvoriť nový záznam náhradného dielu. V pravej hornej časti tabuľky by sa malo nachádzať pole pre vyhľadávanie záznamov v tabuľke. Ak by administrátor napísal do tohto poľa nejakú hodnotu, tak by sa vyhľadávala vo všetkých stĺpcov tabuľky okrem stĺpca pre počet strojov s dielom. Ďalej kliknutím na názov stĺpca by mal byť administrátor schopný zoradiť záznamy vzostupne podľa hodnôt v danom stĺpci. Opätovným kliknutím na názov stĺpca by sa mali záznamy zoradiť zostupne podľa hodnôt v danom stĺpci. Ak by administrátor klikol tretíkrát na ten istý názov stĺpca, poradie záznamov by sa malo vrátiť do pôvodného stavu ako bolo pred prvým kliknutím. Ak by už boli záznamy zoradené podľa stĺpca X a administrátor by klikol na názov iného stĺpca Y, záznamy by sa mali zoradiť (iba) podľa stĺpca Y.

Navyše záznamy v tabuľke by mali byť stránkované. Maximálny povolený počet zobrazených záznamov na stránku by mal byť päť. V ľavej spodnej časti tabuľky by sa mal nachádzať panel s číslami strán, na ktoré by sa malo dať klik-

nút a presunúť tak na danú stranu. Okrem čísel strán by sa tam mali nachádzať tlačidlá „«“, „<“, „>“, „»“. Kliknutím na tlačidlo „«“ by mal byť administrátor presunutý na prvú stranu, kliknutím na tlačidlo „<“ by mal byť administrátor presunutý o jednu stranu dopredu. Tlačidlá „>“ a „»“ by mali fungovať analogicky (lenže na opačnú stranu). Ak by sa administrátor nachádzal na prvej strane, tak tlačidlá „«“ a „<“ by nemali byť aktívne (nemalo by sa na ne dať kliknúť). Podobne ak by sa administrátor nachádzal na poslednej strane, tlačidlá „>“ a „»“ by nemali byť aktívne. Okrem toho by mal byť v pravej spodnej časti tabuľky vypísaný údaj, že na akej strane sa administrátor nachádza, koľko je strán a koľko je vypísaných záznamov na aktuálnej strane.

2.8.5 Karty Bagre, Typy bagrov, Typy vlastností bagrov

Pohľad pre administrátorov: Tak ako by sa mal byť administrátor schopný prekliknúť na kartu Náhradné diely, tak by sa mal byť schopný prekliknúť aj na karty Bagre, Typy bagrov alebo Typy vlastností bagrov. Všetky tieto karty by mali byť skoro totožné s kartou Náhradné diely, ktorá bola opísaná v časti 2.8.4. Líšiť by sa mali len v nadpise (ktorý by sa mal nachádzať nad tabuľkou), v stĺpcoch tabuľky (pochopteľne aj stĺpcami, ktoré nie sú určené pre vyhľadávanie), pričom tabuľka na každej karte by mala obsahovať stĺpec pre akcie rovnako ako bolo spomenuté v opise karty Náhradné diely (viď v časti 2.8.4). Takisto by sa mali líšiť aj formulárom pre vytvorenie alebo úpravu záznamu. Teraz si prejdeme jednotlivé karty a detailnejšie opíšeme zmeny oproti karte Náhradné diely.

- **Karta Bagre**

Na tejto karte by mal byť zobrazený nadpis „Správa bagrov“ a tabuľka obsahujúca stĺpce pre názov, kategóriu a značku bagra, a takisto údaj, či je bager určený iba pre aukciu. Nemalo by sa dať vyhľadávať podľa stĺpca, ktorý udáva, či je bager určený iba pre aukciu. Formulár pre vytvorenie, úpravu bagra by mal byť totožný s formulárom spomenutým v časti 2.4.7 (formulár môžeme vidieť na obr. 2.11, pričom nadpis nad formulárom by nemal byť súčasťou formulára).

- **Karta Typy bagrov**

Táto karta by mala obsahovať nadpis „Správa typov bagrov“ a tabuľku, ktorá by mala obsahovať stĺpce pre značku a kategóriu bagra, a taktiež počet bagrov daného typu. Nemalo by sa dať vyhľadávať podľa stĺpca pre počet bagrov. Pre lepšiu predstavu formulára pre vytvorenie, úpravu typu bagra viď obr. 2.28.

The image shows a web form with a blue background. It contains the following elements:

- A label "Značka:" followed by a white input field.
- A label "Kategória:" followed by a white input field.
- A label "Typy vlastností:" above a table.
- The table has three columns: "Vybrať", "Názov vlastnosti", and "Typ vlastnosti". The first row shows a checkbox, an ellipsis "...", and another ellipsis "...".
- At the bottom, there are two buttons: a green "Uložiť" button and a grey "Reset" button.

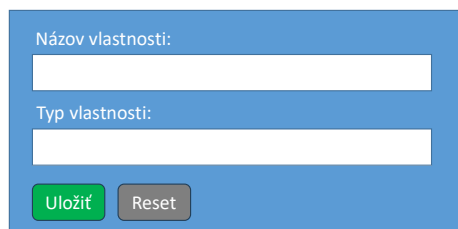
Obr. 2.28: Formulár typu bagra.

Formulár by mal obsahovať polia pre výber značky bagra a kategórie bagra (obe údaje sú povinné), a taktiež vylistované typy vlastností bagra. Pre každý vylistovaný typ vlastnosti bagra je zobrazený jeho názov (napr. hmotnosť) a typ (napr. číslo), pričom vedľa týchto údajov by malo byť začiarkovacie políčko. Pomocou týchto začiarkovacích políčok by mal byť administrátor schopný vybrať aké typy vlastností by mal mať vytváraný (resp. upravovaný) typ bagra. Ďalej by mal mať formulár v spodnej časti tlačidlá „Uložiť“ (pre uloženie záznamu) a „Reset“ (pre vyprázdenie formulára). Ak by po kliknutí na tlačidlo „Uložiť“ nebolo nejaké z polí validné, napr. by chýbala hodnota v nejakom z povinných polí, tak by sa pri danom poli mala zobrazit odpovedajúca chybová správa.

- **Karta Typy vlastností bagrov**

Na tejto karte by mal byť nadpis „Správa typov vlastností bagrov“ a tabuľka obsahujúca stĺpce pre názov a typ vlastnosti, ale takisto aj stĺpec pre údaj o počte typov bagrov s danou vlastnosťou. Nemalo by sa dať vyhľá-

dávať podľa počtu typov bagrov s danou vlastnosťou. Pre lepšiu predstavu formulára pre vytvorenie, úpravu typu vlastnosti bagra viď obr. 2.29.



The image shows a web form with a blue background. At the top, there is a label 'Názov vlastnosti:' followed by a white text input field. Below that is another label 'Typ vlastnosti:' followed by a white text input field. At the bottom of the form, there are two buttons: a green button labeled 'Uložiť' and a grey button labeled 'Reset'.

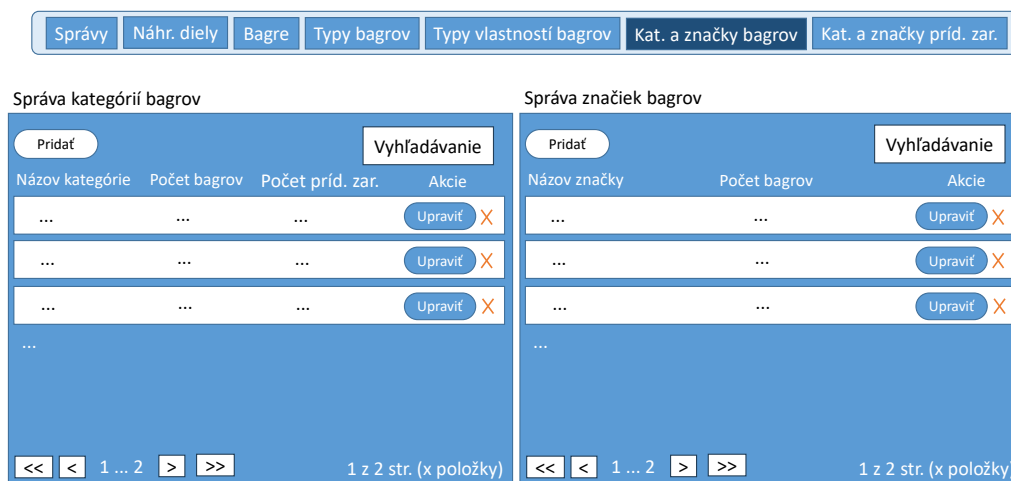
Obr. 2.29: Formulár typu vlastnosti bagra.

Formulár by mal obsahovať polia pre názov a výber typu vlastnosti bagra, pričom obe údaje sú povinné. V spodnej časti formulára by mali byť tlačidlá „Uložiť“ (pre uloženie záznamu) a „Reset“ (pre vyprázdenie formulára). Ak by po kliknutí na tlačidlo „Uložiť“ chýbala v nejakom z polí hodnota, tak by sa pri danom poli mala zobrazíť odpovedajúca chybová správa, ktorá by na to administrátora upozornila.

2.8.6 Karty Kategórie a značky bagrov, Kategórie a značky prídavných zariadení

Pohľad pre administrátorov: Rovnako ako by sa mal byť administrátor schopný prekliknúť na kartu Náhradné diely, tak by sa mal byť schopný prekliknúť aj na kartu Kategórie a značky bagrov alebo kartu Kategórie a značky prídavných zariadení. Tieto karty sa oproti karte Náhradné diely líšia, rovnako ako karty spomenuté v časti 2.8.5, v nadpise (ktorý sa nachádza nad tabuľkou), v stĺpcoch tabuľky (pochopiteľne aj stĺpcami, ktoré nie sú určené pre vyhľadávanie), pričom stále platí, že tabuľka na každej karte by mala obsahovať stĺpec pre akcie rovnako ako bolo spomenuté v opise karty Náhradné diely (viď v časti 2.8.4). Taktiež by sa mali líšiť aj formulárom pre vytvorenie alebo úpravu záznamu. No navyše platí, že tieto karty by mali obsahovať každá nie len jednu, ale dve tabuľky. Jednu pre ka-

tegérie bagrov a druhú pre značky bagrov (analogicky pre kartu s kategóriami a značkami prídavných zariadení). Pre lepšiu predstavu viď obr. 2.30.



Obr. 2.30: Správa kategórií a značiek bagrov.

Táto zmena by mala existovať kvôli tomu, aby sa zmenšil počet kariet (malo by to viesť ku krajšiemu, prehľadnejšiemu užívateľskému rozhraniu), ale takisto aj kvôli tomu, aby sa zamedzilo chybám spôsobenými zamenením tabuliek a formulárov. Kategórie a značky bagrov aj prídavných zariadení sú si koncepčne podobné a ako uvidíme ďalej v texte, ich tabuľky a formuláre sú si tiež veľmi podobné. Preto by mala byť tabuľka so značkami bagrov a tabuľka s kategóriami bagrov na jednej karte a tabuľka so značkami prídavných zariadení a tabuľka s kategóriami prídavných zariadení na inej karte. Teraz si prejdeme a detailnejšie opíšeme zmeny oproti karte Náhradné diely jednotlivé karty podobne ako sme to spravili v časti 2.8.5.

- **Karta Kategórie a značky bagrov**

Ako bolo spomenuté, na tejto karte sa nachádzajú dve tabuľky – jedna je pre správu kategórií bagrov a druhá pre správu značiek bagrov. Obe si teraz opíšeme (opisovanú kartu je možné vidieť na obr. 2.30):

- *Správa kategórií bagrov*

Nad tabuľkou by sa mal nachádzať nadpis „Správa kategórií bagrov“. Tabuľka by mala obsahovať stĺpce pre názov kategórie, počet bagrov danej kategórie, a takisto počet prídavných zariadení určených

pre danú kategóriu bagrov. Nemalo by sa dať vyhľadávať podľa počtu bagrov danej kategórie, a ani podľa počtu prídavných zariadení určených pre danú kategóriu. Pre lepšiu predstavu formulára pre vytvorenie, úpravu kategórie bagra viď obr. 2.31.

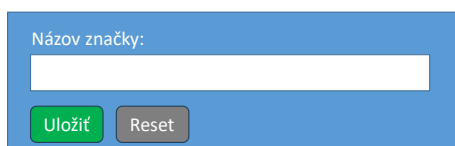
The image shows a screenshot of a web form. At the top, there is a label "Názov kategórie:" in a blue box. Below the label is a white text input field. At the bottom of the blue box, there are two buttons: a green button labeled "Uložiť" and a grey button labeled "Reset".

Obr. 2.31: Formulár kategórie bagra.

Formulár by mal obsahovať pole pre názov kategórie bagra (údaj je povinný). Ďalej by sa mali v spodnej časti formulára nachádzať tlačidlá „Uložiť“ (pre uloženie záznamu) a „Reset“ (pre vyprázdenie formulára). Ak by bolo po kliknutí na tlačidlo „Uložiť“ pole prázdne, tak by na to mala aplikácia upozorniť odpovedajúcou chybovou správou.

– *Správa značiek bagrov*

Nad tabuľkou by sa mal nachádzať nadpis „Správa značiek bagrov“. V tabuľke by mali byť stĺpce pre názov značky a počet bagrov danej značky. Nemalo by sa dať vyhľadávať podľa počtu bagrov danej značky. Pre lepšiu predstavu formulára pre vytvorenie, úpravu značky bagra viď obr. 2.32.



Obr. 2.32: Formulár značky bagra.

Formulár by mal obsahovať pole pre názov značky bagra (údaj je povinný). V spodnej časti formulára by sa mali nachádzať tlačidlá „Uložiť“ (pre uloženie záznamu) a „Reset“ (pre vyprázdenie formulára). Ak by bolo po kliknutí na tlačidlo „Uložiť“ pole prázdne, tak by sa mala pri poli zobrazíť odpovedajúca chybová správa.

- **Karta Kategórie a značky prídavných zariadení**

Na karte Kategórie a značky prídavných zariadení sa takisto nachádzajú dve tabuľky – jedna je pre správu kategórií prídavných zariadení a druhá pre správu značiek prídavných zariadení. Taktiež si ich teraz obe opíšeme:

– *Správa kategórií prídavných zariadení*

Nad tabuľkou by sa mal nachádzať nadpis „Správa kategórií prídavných zariadení“. Tabuľka by mala obsahovať stĺpce pre názov kategórie a počet prídavných zariadení danej kategórie. Nemalo by sa dať vyhľadávať podľa počtu prídavných zariadení danej kategórie. Formulár pre vytvorenie, úpravu kategórie prídavného zariadenia je totožný

s formulárom pre vytvorenie, úpravu bagra, a preto pre jeho lepšiu predstavu viď obr. 2.31.

– *Správa značiek prídavných zariadení*

Nad tabuľkou by sa mal nachádzať nadpis „Správa značiek prídavných zariadení“. V tabuľke by mali byť stĺpce pre názov značky a počet prídavných zariadení danej značky. Nemalo by sa dať vyhľadávať podľa počtu prídavných zariadení danej značky. Pre lepšiu predstavu formulára pre vytvorenie, úpravu značky bagra je možné pozrieť obr. 2.32, pretože tieto formuláre sú výzorovo a správaním totožné.

2.9 Splnenie P6

V tejto podkapitole si prejdeme časti aplikácie spĺňajúce 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 vyplňanie formulárov (aby nemuseli zadávať svoje osobné údaje), umožnenie neprihláseným užívateľom posielat dopyty a účastniť sa aukčných dražieb.

V častaiach 2.4.6, 2.5.3 a 2.6 už bolo spomenuté, že aj prihlásení, aj neprihlásení užívatelia by mali byť schopní dopytovať sa na bager, prídavné zariadenie, výkopovú službu alebo takisto ponúkať sumy do dražby (teda zapájať sa do aukcie). Pričom prihláseným užívateľom by sa nemali zobrazovať polia s osobnými údajmi (nemusia ich vyplňať). Tým pádom sú požiadavky zjednodušenie vyplňania formulárov prihlásenými užívateľmi, umožnenie posielania dopytov neprihláseným užívateľom a umožnenie účasti v aukcii neprihláseným užívateľom splnené.

2.9.1 Prihlasovací panel

Spoločný pohľad: V ľavej časti aplikácie nad navigáciou by sa mal nachádzať prihlasovací panel. Pre lepšiu predstavu umiestnenia panela viď obr. 2.1) a pre lepšiu predstavu samotného panela, spolu s prihlasovacím a registračným formulárom, viď obr. 2.33.



Obr. 2.33: Prihlasovací panel (hore), prihlasovací formulár (vľavo), registračný formulár (vpravo).

Prihlasovací panel by mal obsahovať:

- **Pre neprihláseného užívateľa:** Tlačidlá „Prihlásiť“ a „Registrovať“ (viď vľavo hore na obr. 2.33). Po kliknutí na tlačidlo „Prihlásiť“ by mal byť užívateľ presmerovaný do časti aplikácie s prihlasovacím formulárom (viď ľavú časť obr. 2.33). Po kliknutí na tlačidlo „Registrovať“ by mal byť užívateľ presmerovaný do časti aplikácie s registračným formulárom (viď pravú časť obr. 2.33).
- **Pre prihláseného užívateľa:** Tlačidlo „Profil“ a odkaz „Odhlásiť“ (viď vpravo hore na obr. 2.33). Po kliknutí na tlačidlo „Profil“ by mal byť užívateľ presmerovaný do časti aplikácie s jeho údajmi. Po kliknutí na odkaz „Odhlásiť“ by mal byť užívateľ odhlásený.

2.9.2 Prihlásenie

Pohľad pre neprihlásených užívateľov: Po kliknutí na tlačidlo „Prihlásiť“ (ktoré by sa malo nachádzať v prihlasovacom paneli) by sa mal užívateľ dostať

do časti aplikácie s formulárom pre prihlásenie. Pre lepšiu predstavu formulára viď ľavú časť obr. 2.33.

Formulár by mal vo vrchnej časti obsahovať nadpis „Prihlásenie“, pod ním polia pre prihlasovacie meno a heslo. Pod poliami by sa malo nachádzať tlačidlo „Prihlásiť“. Ak by užívateľ zadal prihlasovacie meno, ktoré neexistuje (nie je evidované v systéme), malo by sa mu po kliknutí na tlačidlo „Prihlásiť“ zobraziť vyskakovacie okno s textom „Užívateľ s takýmto menom neexistuje.“ a tlačidlom „OK“. Podobne ak by užívateľ zadal nesprávnu kombináciu prihlasovacieho mena a hesla, malo by sa mu po kliknutí na tlačidlo „Prihlásiť“ zobraziť vyskakovacie okno s textom „Nesprávne heslo.“ a tlačidlom „OK“. Po vyplnení (validných) údajov a kliknutí na tlačidlo „Prihlásiť“ by mal byť užívateľ prihlásený do systému a presmerovaný do sekcie Hlavná ponuka.

2.9.3 Registrácia

Pohľad pre neprihlásených užívateľov: Po kliknutí na tlačidlo „Registrovať“ by sa mal užívateľ dostať do časti aplikácie s formulárom pre registráciu. Pre lepšiu predstavu formulára viď pravú časť obr. 2.33.

V hornej časti formulára by sa malo nachádzať nadpis „Registrácia“. Formulár by mal obsahovať povinné polia prihlasovacie meno, heslo, znova heslo, meno, priezvisko, email a nepovinné polia telefónne číslo a mesto. Ak by užívateľ začal písať do poľa heslo alebo znova heslo, namiesto písmen by sa mu mali zobrazovať hviezdičky. Pri týchto poliach by mali byť ikonky pre zobrazenie/schovanie hesla. Ak by užívateľ klikol na ikonku, namiesto hviezdičiek by sa mu mali zobraziť písmená (opätovným kliknutím opäť hviezdičky). Teda išlo by o funkcionálnu zobrazenie a schovanie hesla. Formulár by mal takisto obsahovať upozornenie, že registráciou užívateľ súhlasí s podmienkami a zásadami systému. Upozornenie by malo zároveň odkazovať na časti aplikácie s podmienkami používania a so zásadami ochrany osobných údajov, ktoré boli spomenuté v podkap. 2.3. V spodnej časti formulára by sa malo nachádzať tlačidlo „Registrovať“, ktorým by sa užívateľ mal byť schopný zaregistrovať do systému. Po úspešnej registrácii by mal byť užívateľ presmerovaný do sekcie Hlavná ponuka.

Ak by po kliknutí na tlačidlo „Registrovať“ nebola hodnota nejakého z polí validná (napr. absencia hodnoty v povinnom poli), mala by sa pri takomto poli zobrazovať vhodná chybová správa. Zároveň obsah políček heslo a znova heslo by mal byť rovnaký. V prípade, že by rovnaký nebol a užívateľ by klikol na tlačidlo „Registrovať“, tak by sa malo zobraziť vyskakovacie okno s textom „Obsah políček 'Heslo' a 'Znova heslo' nie sú rovnaké.“ a tlačidlom „OK“.

2.9.4 Profil

Spoločný pohľad: Po kliknutí na tlačidlo „Profil“, ktoré by sa malo zobrazovať prihláseným užívateľom v prihlasovacom paneli, by mal byť užívateľ presmerovaný do časti aplikácie s formulárom, ktorý by mal obsahovať jeho údaje. Užívateľ by mal byť schopný pomocou tohto formulára svoje údaje upravovať. Navyše nad formulárom by sa malo nachádzať nadpis „Profil“. Pre lepšiu predstavu tejto časti aplikácie viď obr. 2.34.

Profil

The image shows a user profile form with a blue header and a white background. The title 'Moje údaje' is centered at the top. Below the title, there are several input fields arranged in a grid. The first row contains 'Prihlasovacie meno*' with a text input field. The second row contains 'Nové heslo*' and 'Znova heslo*', both with password input fields and eye icons. The third row contains 'Meno*' and 'Priezvisko*', both with text input fields. The fourth row contains 'Email*' and 'Telefónne číslo', both with text input fields. The fifth row contains 'Mesto:' with a text input field. At the bottom right, there are three buttons: 'Uložiť' (white), 'Export dát' (yellow), and 'Zmazať účet' (red).

Obr. 2.34: Profil užívateľa

V hornej časti formulára by sa mal nachádzať nadpis „Moje údaje“. Formulár by mal obsahovať polia prihlasovacie meno, heslo, znova heslo, meno, priezvisko, email, telefónne číslo a mesto. Pričom všetky polia, okrem polí telefónne číslo a mesto, sú povinné. Povinné polia by mali byť označené hviezdíčkou.

Všetky polia, až na polia heslo a znova heslo, by mali byť vyplnené užívateľovými údajmi. Ak by užívateľ začal písať do poľa heslo alebo znova heslo, namiesto písmen by sa mu mali zobrazovať hviezdíčky. Pri týchto poliach by mali byť ikony, ktoré by kliknutím umožnili užívateľovi meniť medzi hviezdíčkami a písmenami v poliach heslo a znova heslo. Teda išlo by o funkcionality zobraziť a schovať heslo.

Pod formulárom by sa malo nachádzať tlačidlo „Uložiť“ pre uloženie zmien. Ak by neboli vykonané žiadne zmeny, tak po kliknutí na tlačidlo „Uložiť“ by sa malo zobraziť vyskakovacie okno s textom „Neboli vykonané žiadne zmeny.“ a tlačidlom „OK“. Ak by užívateľ zinvalidoval nejaké z polí (napr. zanechá prázdne povinné pole), mala by sa zobraziť chybová správa pri danom poli. Výnimkou sú polia heslo a znova heslo – aj napriek tomu, že sú povinné, tak môžu ostať prázdne.

Ak by si chcel užívateľ zmeniť heslo, musel by vyplniť obe polia, a ak by obsah polí nebol rovnaký, tak by sa malo zobraziť vyskakovacie okno s textom „Obsah políček 'Nové heslo' a 'Znova heslo' nie sú rovnaké.“ a tlačidlom „OK“. Alebo ak by si chcel užívateľ zmeniť heslo a zadal by staré heslo, tak znova by sa malo zobraziť vyskakovacie okno s textom „Nové heslo nemôže byť staré heslo.“ a tlačidlom „OK“. Po úspešnej zmene údajov by sa malo zobraziť vyskakovacie okno s textom „Údaje úspešne zmenené.“ a tlačidlom „OK“.

Okrem toho by sa mali v spodnej ľavej časti formulára nachádzať tla-

čidlá „Export dát“ a „Zmazať účet“. Po kliknutí na tlačidlo „Export dát“ by sa mali užívateľovi stiahnuť údaje (v textovej forme), ktoré má o ňom systém uložené. Po kliknutí na tlačidlo „Zmazať účet“ by sa malo zobraziť modálne potvrdzovacie okno s nadpisom „Vymazať svoj účet natrvalo“ a textom „POZOR! Naozaj chcete vymazať svoj účet so všetkými svojimi dátami?“. Tieto obe tlačidlá by mali existovať z dôvodu prípravy na splnenie GDPR (spomenuté v podkap. 2.3).

3. Analýza

V tejto kapitole si na základe potrieb vyplývajúcich z návrhu užívateľského rozhrania (viď kap. 2) rozhodneme, aké technológie a prístupy sa nám hodia na implementáciu systému, a teda aké využijeme.

3.1 Výber typu aplikácie

Kvôli požiadavke P10 (Dostupnosť, viď v podkap. 1.1) dáva dobrý zmysel vytvoriť náš systém ako webovú aplikáciu – vyriešime tým problém s distribúciou softvéru k (aj potenciálne novým) zákazníkom. Statická stránka by nám nestačila nakoľko chceme administrátorom umožniť dynamickú zmenu obsahu (napr. pridávanie alebo mazanie bagrov, prídavných zariadení atď.).

3.2 Rozdelenie na frontendovú a backendovú časť

Kvôli lepšej organizácii kódu dáva pre našu aplikáciu dobrý zmysel rozdeliť ju na frontendovú a backendovú časť. Frontendová časť pokryje užívateľské rozhranie navrhnuté v kap. 2 a backendová časť pokryje zbytok logiky systému (napr. prácu s databázou, aukčný odpočet atď.).

V súčasnosti rôzne spoločnosti, napr. Amazon alebo Cloudflare, poskytujú backendové služby, tzv. Backend-as-a-Service (Baas) [3]. Ak by sme sa rozhodli využiť takúto službu, ušetrilo by nám to implementovanie užívateľskej autentifikácie (vyžadované požiadavkou P6) alebo práce s databázou (vyžadované napr. požiadavkou P2.4). Ale ak by sme chceli naimplementovať napr. vyhodnocovanie aukcie alebo funkcionality, ktorá umožní meniť administrátorovi formát automaticky generovaných správ týkajúcich sa aukcie (vyžadované v požiadavkách P4 a P5.3), tak BaaS by nám na to nestačil. Navyše ak spoločnosť neposkytuje bezplatnú možnosť využívania BaaS, využitie BaaS by bolo v rozpore s požiadavkou P11. Kvôli týmto dôvodom si budeme musieť naimplementovať vlastnú backendovú časť.

Pre implementáciu backendovej časti by sa mohli hodiť vysokoúrovňové jazyky, akými sú napr. Java alebo C#. Nakoľko ide o podobné jazyky a autor jazyk Java neovláda, ale C# áno, tak si volíme C# a platformu .NET, ktorá je s ním spojená.

V prípade frontendovej časti je zvyčajnou voľbou jazyk JavaScript (príp. TypeScript), a keďže s ním už má autor nejaké skúsenosti, tak predstavuje jednu z možností. No v súčasnosti je možné využiť aj vo frontendovej časti jazyk C#. Ako už bolo spomenuté, autor tento jazyk pozná. Navyše ak by sme si vybrali C#, tak by to znamenalo, že frontendová aj backendová časť, by boli obe napísané v rovnakom jazyku, čo by mohlo viesť k zjednodušeniu implementácie. A preto je C# ďalšou z možností jazykov, ktoré môžeme využiť vo frontendovej časti. Obe jazyky so sebou prinášajú svoje frontendové frameworky, ktoré nám dokážu uľahčiť implementáciu frontendovej časti. V nasledujúcej podkapitole sa

pozrieme na možné typy webovej aplikácie a frameworky, ktoré môžeme využiť na ich implementáciu. Podľa toho si vyberieme jazyk pre frontendovú časť.

3.3 Výber typu webovej aplikácie

V predchádzajúcej podkapitole sme si rozdelili aplikáciu na frontendovú a backendovú časť. Takisto sme si predstavili jazyky, ktoré by sme mohli v týchto častiach využiť. Každý jazyk so sebou nesie i svoje frameworky. V tejto podkapitole sa pozrieme na to, aký typ webovej aplikácie by mal byť náš systém a podľa toho si aj zvolíme jazyk s frameworkom.

3.3.1 Single-page application

Single-page application (SPA) [4] je druh webovej aplikácie, ktorá po príchode užívateľa na stránku načíta jeden webový dokument, a potom už iba mení jeho obsah. Väčšina logiky aplikácie sa vykonáva na strane klienta, preto je aplikácia veľmi rýchla z hľadiska interakcie s užívateľom.

Ako už bolo skôr spomenuté, naša aplikácia mení obsahovú časť podľa toho, v akej sekcii sa užívateľ nachádza (viď 2.1). Takisto v sekcii „Správa webu“ preklikávaním medzi kartami sa mení obsah, ktorý vidí užívateľ (spomenuté v 2.8). Podobne na stránkach s formulármi (detail bagra, prídavného zariadenia, aukčnej podnuky alebo sekcia Služby) chceme umožniť zobrazovanie a schovávanie formulára (spomenuté v 2.4.5, 2.5.2 a 2.6). Aj kvôli týmto dôvodom sa zdá, že voľba SPA by bola rozumná.

Jazyk JavaScript (resp. TypeScript) nám ponúka viacero frontendových frameworkov pre tvorbu SPA aplikácií, napr. AngularJS, ReactJS, Vue atď. Systém by sa dal naimplementovať ktorýmkoľvek z týchto frameworkov, no kvôli predošlej skúsenosti autora si ako zástupcu JavaScriptových frameworkov zvolíme ReactJS. Ďalej jazyk C# nám ponúka framework Blazor WebAssembly.

Obe frameworky, ReactJS aj Blazor WebAssembly, fungujú na princípe komponentov. Komponent si môžeme predstaviť ako nejakú ucelenú časť stránky, napr. tlačidlo, formulár, navigácia atď. Akonáhle je komponent zadaný, môže sa využívať vo viacerých častiach webu. Všimnime si, že v našom návrhu užívateľského rozhrania sa nachádza viacero častí, ktoré vyzerajú úplne rovnako (príp. sú rozdiely minimálne, ale to sa dá vyriešiť parametrizáciou komponentov), napr. formuláre v časti s detailami bagra, prídavného zariadenia alebo v sekcii Služby (viď pravú stranu obrázkov 2.4.5, 2.5.2, 2.6).

Zatiaľ sa zdá, že obe frameworky by boli dobrou voľbou. No použitie SPA prístupu by nám takisto prinieslo zopár nevýhod:

- **Prvotný čas načítania stránky**

Ako bolo spomenuté, väčšina logiky aplikácie sa vykonáva na strane klienta, kvôli tomu sa pri prvotnom načítaní stránky sťahujú zdrojové kódy aplikácie, a to spôsobuje zdržanie. Toto správanie by mohlo odradiť nových potenciálnych zákazníkov.

- **SPA nie sú „SEO-friendly“**

Search engine optimization (SEO) [5] je proces optimalizovania stránky na to, aby bola jednoducho dohľadateľná vyhľadávačmi (search engines). Vyhľadávače, akým je napr. Google, prechádzajú stránky na internete, hodnotia ich a na základe toho vedia usúdiť, či je stránka pre užívateľa relevantná.

V prípade SPA aplikácii nastáva problém, pretože obsah je užívateľom generovaný dynamicky a vyhľadávače majú pri prechádzaní s týmto správaním SPA aplikácií problém. Nie je síce nemožné vytvoriť „SEO-friendly“ SPA aplikáciu, ale ide o komplikovanejší proces. Viac informácií na túto tému môžeme nájsť na stránkach Novateus [6] a Cloudways [7].

- **Potenciálny problém so staršími verziami prehliadačov**

Tento bod sa vzťahuje skôr špeciálne k Blazor WebAssembly. Aktuálne verzie prehliadačov obsahujú všetky nástroje potrebné pre bezproblémový beh SPA aplikácií (resp. podporujú technológiu web assembly). Avšak ako bolo spomenuté v požiadavke P10 (Dostupnosť, viď v podkap. 1.1), hrozí, že firemné počítače môžu využívať staré verzie prehliadačov, na ktorých by SPA aplikácie nemuseli fungovať správne. Táto myšlienka vychádza z toho, že v prehliadači Google Chrome (ktorý je v roku 2023 podľa štatistík najpopulárnejším prehliadačom [8]) nebola technológia web assembly vôbec podporovaná do verzie 50 [9] a najvyššia podporovaná verzia prehliadača Google Chrome na počítačoch s operačným systémom Windows XP neprevyšuje verziu 50 [10] [11] [12].

3.3.2 Multi-page application

Multi-page application (MPA) [13] je druh webovej aplikácie, ktorá využíva klasický prístup – keď užívateľ klikne na nejaký odkaz, odošle sa serveru žiadosť, a ten vráti užívateľovi ako odpoveď celú stránku. Väčšina logiky aplikácie sa vykonáva na serveri a užívateľovi sa podáva iba HTML, CSS a JavaScript.

Autorovi sa nepodarilo nájsť žiadne JavaScriptové MPA frameworky, ale v prípade jazyka C# je známym ASP.NET MVC [14]. Ak by sme si zvolili tento framework, a teda využili MPA prístup, tak pri preklikávaní odkazmi navigácie by každá zmena obsahovej časti aplikácie (viď 2.1) síce znamenala načítanie novej stránky, ale to by nepredstavovalo problém. Takisto by sa dali pomocou frameworku ASP.NET MVC naimplementovať stránky s formulármi (detail bagra, prídavného zariadenia, aukčnej podnuky alebo sekcia Služby) a aj prepínanie medzi kartami v sekcii „Správa webu“ (spomenuté v 2.8).

Takže sa zdá, že aj implementácia pomocou MPA prístupu je možná a navyše má oproti SPA výhody:

- **Prvotný čas načítania stránky**

Narozdiel od SPA sa väčšina logiky vykonáva na serveri, kde sú aj uložené zdrojové kódy aplikácie, kvôli tomu si ich užívateľ nemusí sťahovať k sebe a prvotné načítanie stránky nie je pomalé.

- **MPA sú „SEO-friendly“**

Vyhľadávače, akým je napr. Google, nemajú problém pri prechádzaní obsahu takýchto aplikácií. Preto sú MPA aplikácie „SEO-friendly“.

- **Neexistuje problém so staršími verziami prehliadačov**

Kedže sa užívateľom podáva iba HTML, CSS, príp. JavaScript, a teda nie je nutné spúšťať kód typu web assembly, tak ani staršie verzie prehliadačov by nemali mať problém s behom aplikácie. Takže nedochádza k porušeniu požiadavky P10 (viď 1.1).

To boli výhody oproti SPA aplikáciám, ale MPA aplikácie majú takisto nevýhody:

- **Záťaž vyvíjaná na server**

Ako bolo spomenuté, užívateľ každým kliknutím na odkaz odosiela žiadosť na server, ktorú musí server spracovať a následne odoslať odpoveď – to vyvíja záťaž na server. V prípade veľkého množstva užívateľov by to mohlo predstavovať problém, ale my vytvárame aplikáciu pre malé firmy, kde sa nepredkladá obrovské množstvo pripojení v rovnakú chvíľu. Takže táto nevýhoda je pre nás nepodstatná.

- **Absencia komponent**

V predošlej časti textu bolo spomenuté, že jednou z výhod frameworkov ReactJS a Blazor WebAssembly je koncept komponentov. V prípade ASP.NET MVC tento koncept chýba, alebo presnejšie, nie je natívne podporovaný, ale dá sa napodobniť [15]. Takže napríklad implementácia časti s preklikovateľnými kartami (2.8) alebo častí, ktoré sú totožné, napr. formuláre v časti s detailami bagra, prídavného zariadenia alebo v sekcii Služby (viď pravú stranu obrázkov 2.4.5, 2.5.2, 2.6) by sa dala realizovať i bez duplicitného kódu, ale nebolo by to až tak prehľadné ako v spomínaných frameworkoch, ktoré podporujú koncept komponentov natívne.

Zdá sa, že MPA aplikácie by vyriešili nejaké z nevýhod SPA aplikácii, ale takisto nie sú dokonalé a nesú so sebou nejaké nevýhody, ktoré však pre nás nie sú príliš podstatné. Preto by sa zdalo, že MPA je prístup, ktorý by sme mali využiť. Existuje však ešte jedna alternatíva, a tou je Blazor Server, o ktorom si povieme v nasledujúcej časti textu.

3.3.3 Blazor Server

Blazor Server [16] je framework jazyka C# a je to framework na vývoj webových aplikácií, ktoré sú „tak niečo medzi“ SPA a MPA aplikáciami. Väčšina kódu aplikácie sa vykonáva na serveri ako v prípade MPA aplikácií. Rozdiel je v tom, že ak na strane klienta dôjde k zmene nejakej časti užívateľského rozhrania, tak namiesto toho, aby server odoslal klientovi celú stránku, odošle iba potrebné zmeny. To umožňuje vytváranie interaktívneho užívateľského rozhrania ako v prípade SPA aplikácií.

Ak by sme využili Blazor Server na vývoj našej aplikácie, tak by mala naša aplikácia rovnaké výhody ako má MPA aplikácia oproti SPA aplikácii (viď časť 3.3.2). No navyše oproti MPA aplikácii by mala výhody:

- **Existencia komponentov**

V SPA časti bolo v súvisi s frameworkmi ReactJS a Blazor WebAssembly spomenuté, že koncept komponentov by sa nám v našej aplikácii hodil (viď časť 3.3.1). Potom v časti MPA sme spomenuli, že ASP.NET MVC by nám síce umožnil využívať koncept komponentov, ale natívne tento koncept nepodporuje a implementácia by nebola taká jednoduchá, priamočiara ako v prípade frameworkov, ktoré tento koncept podporujú natívne (viď 3.3.2). Našťastie je Blazor Server jeden z frameworkov, ktorý tento koncept komponentov natívne podporuje.

Je síce pravda, že aj naďalej ostáva nevýhodou vyťažovanie servera ako tomu bolo v prípade MPA aplikácií, ale ako bolo napísané v prechádzajúcej časti textu (viď časť 3.3.2), táto nevýhoda pre nás nepredstavuje problém. Navyše výberom tohto frameworku budeme môcť písať aj frontendovú, aj backendovú časť v rovnakom jazyku, čo by mohlo zjednodušiť implementáciu. Preto si volíme ako frontendový jazyk C# s frameworkom Blazor Server.

3.4 Voľba databázy

Z požiadaviek na systém vyplýva, že budeme potrebovať databázu pre uloženie dát (napr. pre uloženie ponuky bagrov, viď požiadavku P2 v podkap. 1.1). Typicky sa databázy rozdeľujú na dva druhy – tradičné relačné databázy alebo alternatívne nerelačné, tzv. NoSql, databázy.

- **Relačné databázy [17]**

Ide o typ databáz, ktorý ukladá a usporiadáva dátové body s definovanými reláciami pre rýchly prístup. Relačné databázy usporiadávajú dáta do tabuliek, ktoré obsahujú informácie o jednotlivých entitách a predstavujú preddefinované kategórie prostredníctvom riadkov a stĺpcov. Štruktúrovanie dát týmto spôsobom umožňuje efektívny a flexibilný prístup. Relačné databázy rozumejú jazyku SQL, ktorý slúži na ukladanie, načítanie a manipuláciu s dátami.

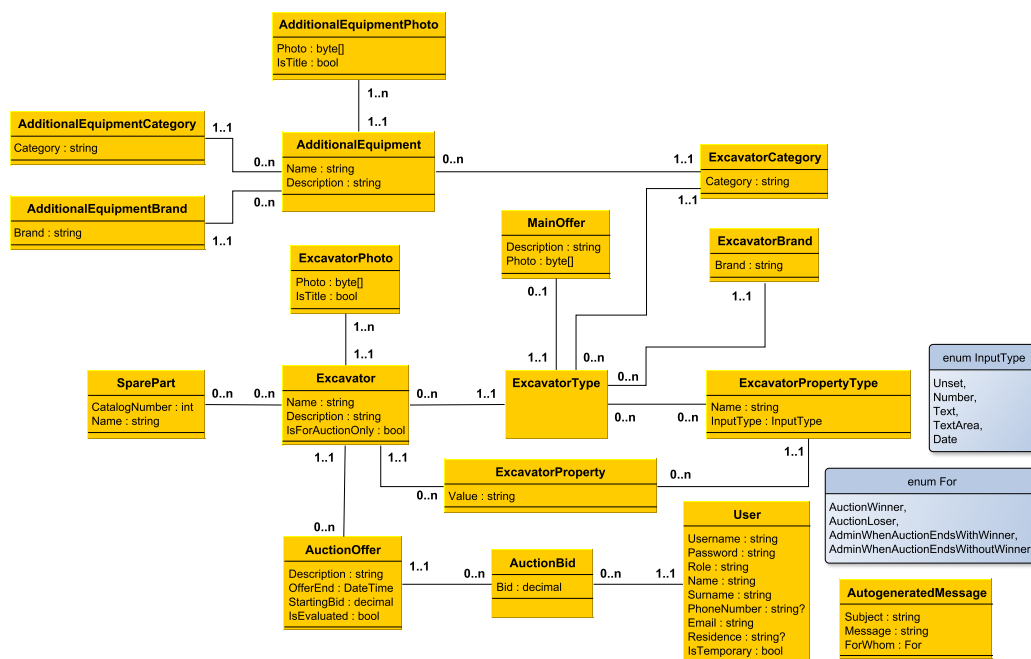
- **Nerelačné (NoSql) databázy [18]**

Tento typ databáz predstavuje databázy „iné než SQL“. Ide o databázy dokumentové, grafové, key-value atď. Tieto databázy dokážu spracovávať veľké množstvo rýchlo sa meniacich neštrukturovaných dát inými spôsobmi než relačné databázy s riadkami a tabuľkami.

NoSql databázy sa hodia pre veľké množstvo neštrukturovaných dát [19]. V našom prípade, keďže vytvárame softvér pre malú firmu, nepredpokladáme veľké množstvo dát (napr. počet bagrov pravdepodobne nepresiahne sto). Taktiež platí, že štruktúra dát je (v podstate) predom určená (viď požiadavky P2.1, P2.2, P2.3 atď. v podkap. 1.1). Navyše má autor trochu viacej skúseností s klasickými relačnými databázami než s NoSql. Kvôli týmto dôvodom si volíme relačnú databázu.

3.4.1 Návrh relačného modelu databázy

V predchádzajúcej časti textu (viď 3.4) sme si zvolili relačnú databázu, a preto bude teraz nasledovať vytváranie relačného modelu databázy [20] (ide o reprezentáciu dát uložených v relačnej databáze). Ako už bolo spomenuté v predchádzajúcej časti textu, štruktúra dát (a teda aj samotné entity) boli viac-menej zadané už v podkapitole s požiadavkami na systém (viď 1.1), v tejto kapitole si upresníme ich formu a vzťahy medzi nimi (resp. kardinality). Pre lepšiu predstavu vytváraného relačného modelu viď obr. 3.1.



Obr. 3.1: Relačný model databázy.

- **Entita Excavator**

V požiadavke P2.2 (Bager, viď podkap. 1.1) sme si stanovili, že každý bager má mať názov, značku, kategóriu, opis, fotky, vlastnosti, a tiež informáciu o tom, aké náhradné diely obsahuje. Taktiež tam bolo napísané, že môžu existovať bagre, ktoré sú určené iba pre aukciu. To, či je bager určený iba pre aukciu, vieme zachytiť pomocou boolovskej vlastnosti (viď entitu **Excavator** na obr. 3.1). Podobne vieme pomocou vlastností v entite zachytiť aj názov a opis.

- **Entita ExcavatorType**

Čo sa týka značky a kategórie, tak tie by by sme takisto mohli v entite reprezentujúcej bager zachytiť pomocou vlastností, ale to by znamenalo, že za každým pri vytváraní nového záznamu pre bager by sa musela špecifikovať hodnota týchto vlastností. Takisto platí, že viacero bagrov môže byť tej istej kategórie a značky. Ak by sa mali pri každom vytváraní záznamu bagra vypisovať hodnoty pre značku a kategóriu, mohlo by to viesť k zbytočným chybám (preklepom), a takisto by to pre administrátora nebolo príliš pohodlné. Preto vytvoríme novú entitu reprezentujúcu typ

bagra (viď ExcavatorType na obr. 3.1), ktorá nám bude určovať kategóriu a značku bagra. Tato entita však nebude obsahovať vlastnosti ani pre značku bagra, ani pre kategóriu bagra.

- **Entita ExcavatorBrand a ExcavatorCategory**

Značka bagra, rovnako ako aj kategória bagra, by mali byť novými osobitnými entitami (viď entity ExcavatorBrand a ExcavatorCategory na obr. 3.1), entita reprezentujúca značku bagra by obsahovala vlastnosť s údajom o značke bagra a entita reprezentujúca kategóriu bagra by obsahovala vlastnosť s údajom o kategórii bagra. To je z toho dôvodu, že môžu existovať viaceré typy bagra, ktoré by boli rovnakej značky, ale líšili by sa kategóriou (alebo naopak, boli by rovnakej kategórie, ale líšili by sa značkou). Tým, že by sme zo značky bagra a kategórie bagra spravili samostatné entity, zamedzili by sme opakujúcim sa hodnotám v databáze. Takisto kvôli tomuto bude stačiť ak si administrátor v aplikácii uloží značku (resp. kategóriu) bagra iba raz, čo môže zamedziť zbytočným chybám (preklepom).

- **Entita ExcavatorPhoto**

Tiež boli v súvislosti s bagrami spomenuté fotky bagrov – a tie budú našou ďalšou entitou (viď entitu ExcavatorPhoto na obr. 3.1). Každý bager môže mať viacero fotiek, pričom tieto fotky budeme chcieť zobrazovať napr. v časti aplikácie s detailom bagra, ktorá bola spomenutá v časti 2.4.5 (viď vľavo hore na obr. 2.10) alebo tiež na kartách strojov, ktoré boli spomenuté v časti 2.4.3 (viď obr. 2.8). Entita reprezentujúca fotku bagra by mala okrem samotnej fotky obsahovať aj boolovskú vlastnosť určujúcu, či je fotka titulná. Vďaka tejto vlastnosti budeme môcť presne určiť, aká fotka bagra sa má zobrazovať na jeho karte.

- **Entita SparePart**

Okrem fotiek boli v súvislosti s bagrami spomenuté aj náhradné diely. Náhradný diel bude predstavovať ďalšiu entitu (viď entitu SparePart na obr. 3.1) a jeho údaje boli definované v P7.1 – má obsahovať katalógové číslo (ide o celé číslo, neobsahuje iné znaky ako číslice) a názov náhradného dielu. Každý bager môže obsahovať viacero rôznych náhradných dielov a jeden náhradný diel sa môže nachádzať vo viacerých bagroch.

- **Entita ExcavatorPropertyType a ExcavatorProperty**

Ďalej boli v súvislosti s bagrami spomenuté aj vlastnosti bagra, ktoré sú určené jeho typom (ako bolo spomenuté v požiadavke P2.2). Tu je dôležité uviesť si, že typ bagra v skutočnosti určuje typ vlastnosti bagra. Napríklad typ bagra X môže mať vlastnosť hmotnosť a ďalší typ bagra Y môže mať takisto vlastnosť hmotnosť, ale okrem nej môže mať aj vlasnosť šírka lyžice. Všimnime si, že hovoríme o typoch vlastností bagra, ale nie o konkrétnych hodnotách vlastností, teda napr. koľko vážia bagre typu X, to nemôžeme vedieť, pretože presná hodnota je určená konkrétnym bagrom a nie jeho typom. Z toho vyplýva, že budeme potrebovať entitu na reprezentáciu typu vlastnosti bagra (viď entitu ExcavatorPropertyType na obr. 3.1),

ale takisto aj entitu reprezentujúcu (konkrétnu) vlastnosť, ktorá by v sebe niesla konkrétnu hodnotu určenú konkrétnym bagrom (viď entitu `ExcavatorProperty` na obr. 3.1). Entita reprezentujúca typ vlastnosti bagra by v sebe mala obsahovať okrem názvu (aby sme vedeli, čo je to za vlastnosť, napr. hmotnosť) aj typ hodnoty danej vlastnosti, aby sme vedeli, či ide o číslo, krátky text, dlhší opis alebo dátum (s určením typu hodnoty danej vlastnosti nám pomôže enum `InputType`, ktorý môžeme vidieť na obr. 3.1). Tento údaj nám dovoľí zlepšiť užívateľské rozhranie, konkrétne typy polí pre hodnoty vlastností bagra vo formulári pre vytvorenie/úpravu bagrov, ktorý bol spomenutý v časti 2.4.7 (viď obr. 2.11).

- **Entita `MainOffer`**

V požiadavke P2 sa vyžaduje, aby existovala aj hlavná ponuka – tá bude našou ďalšou entitou (viď entitu `MainOffer` na obr. 3.1). Konkrétne v požiadavke P2.1 sme si stanovili, že hlavná ponuka reprezentuje typ bagrov a má obsahovať opis, a tiež fotku daného typu. V predošlom texte bolo spomenuté, že môžu existovať bagre, ktoré by boli určené iba pre aukciu, a teda nenachádzali by sa v ponuke (nových) bagrov. Preto môže nastať situácia, kde by sme chceli, aby sa v databáze nachádzal typ bagra, ale nechceme, aby existovala v databáze hlavná ponuka pre tento typ bagra. Zároveň platí, že hlavná ponuka predstavuje bagre jedného typu.

- **Entita `AdditionalEquipment`**

V požiadavke P2 je tiež spomenutá aj ponuka prídavných zariadení, preto si vytvoríme aj entitu pre reprezentáciu prídavného zariadenia (viď entitu `AdditionalEquipment` na obr. 3.1), tá má podľa požiadavky P2.3 obsahovať názov, opis, fotky, značku a kategóriu prídavného zariadenia, ale takisto aj údaj o tom, pre akú kategóriu bagrov je prídavné zariadenie určené. Z rovnakého dôvodu prečo sme vytvorili novú entitu pre kategóriu bagra (entita `ExcavatorCategory`, ktorá bola spomenutá v prechádzajúcom texte), vytvoríme aj entity reprezentujúce značku prídavného zariadenia (viď entitu `AdditionalEquipmentBrand` na obr. 3.1) a kategóriu prídavného zariadenia (viď entitu `AdditionalEquipmentCategory` na obr. 3.1). Ďalej čo sa týka fotiek prídavného zariadenia, tak by sa naskytovala možnosť zmeniť entitu reprezentujúcu fotky bagrov (`ExcavatorPhoto`) na entitu reprezentujúcu fotky všeobecne (aj bagrov, aj prídavných zariadení). No tento prístup by nám umožnil priradiť fotku prídavného zariadenia bagru (a naopak), čo nie je správne. Preto namiesto toho vytvoríme, rovnako ako v prípade fotky bagra, novú entitu reprezentujúcu fotku prídavného zariadenia (viď entitu `AdditionalEquipmentPhoto` na obr. 3.1).

- **Entita `User`**

Ďalej z požiadavky P6 vieme, že budeme potrebovať entitu reprezentujúcu užívateľa (viď entitu `User` na obr. 3.1). Konkrétne z P6.3 vieme, že o užívateľovi potrebujeme vedieť jeho prihlasovacie meno, heslo, (krstné) meno, priezvisko, email, príp. aj telefónne číslo a bydlisko (mesto). No okrem týchto údajov sa nám tiež kvôli požiadavke P6.1 hodí údaj, či je užívateľ dočasný. Presnejšie je to kvôli tomu, aby aj neprihlásený užívateľ mohol ponúkať

sumy do dražby. Po ponúknutí sumy by sa do databázy uložila informácia o tom, kto ponúkol sumu (a tiež čo to bolo za aukčnú ponuku). Po prejedaní termínu konca aukčnej ponuky a vyhodnotení aukcie sa budú môcť tieto dočasné účty vymazať. Okrem toho by sa nám tiež hodil údaj o tom, či je užívateľ bežným zákazníkom alebo administrátorom, aby mu podľa toho vedel systém zobrazíť správny obsah (vychádza z P1 Roly užívateľa, viď podkap. 1.1). Všetky tieto údaje by mali byť zahrnuté vo vlastnostiach entity User.

- **Entita AuctionOffer**

Z požiadavky P4 vieme, že administrátor by mal byť schopný zaradiť bager do aukcie a vytvoriť tak aukčnú ponuku – to bude našou ďalšou entitou (viď entitu AuctionOffer na obr. 3.1). Takisto z požiadavky P4 vieme, že administrátor by mal mať možnosť zadať dátum konca aukčnej ponuky, počiatočnú sumu a popis ponuky (v ňom môže byť napísané napr. čo bolo v bagri opravené). Preto budú tieto údaje vlastnosťami entity AuctionOffer.

- **Entita AuctionBid**

Okrem entity AuctionOffer vyplýva z požiadavky P4 aj entita reprezentujúca ponuku (sumu) ponúknutú zákazníkom (viď entitu AuctionBid na obr. 3.1). Údaj o tom, koľko zákazník ponúkol, je vlastnosťou entity AuctionBid.

- **Entita AutogeneratedMessage**

Našou poslednou entitou bude entita pre automaticky generované správy, ktorej potreba vyplýva z požiadaviek P4.1 a P5.3. Z P4.1 vieme, že budeme potrebovať odosielať automaticky generované správy – konkrétne budeme potrebovať odosielať správy víťazom dražieb, porazeným, ale takisto aj administrátorom v prípade, že dražba skončila (a to aj v prípade, že skončila bez víťaza). Všetky tieto prípady zachytáva enum For, ktorý môžeme vidieť na obr. 3.1. Ďalej z P5.3 vieme, že chceme dovoliť administrátorom upravovať formát automaticky generovaných správ, preto potrebujeme entitu reprezentujúcu automaticky generovanú správu (viď entitu AutogeneratedMessage na obr. 3.1). V požiadavke P4.1 je napísané, že sa tieto automaticky generované správy majú posielajú ako emaily, preto by mala entita AutogeneratedMessage obsahovať vlastnosti pre predmet a samotný text správy. Tiež by mala obsahovať vlastnosť určujúcu pre koho, resp. pre akú príležitosť, je automaticky generovaná správa určená.

3.4.2 Voľba databázového servera

Čo sa týka databázového servera, tak si najprv poďme rozmyslieť, čo od neho budeme vyžadovať. Kvôli požiadavke P11 (Náklady, viď podkap. 1.1) chceme, aby bol server (v najlepšom prípade) bezplatný. Taktiež musí byť kompatibilný s platformou .NET. Z bežne používaných databázových serverov môžeme vybrať napr. Microsoft SQL Server (Express edition) [21], Oracle Database (Express edition) [22] alebo MySQL (Community server) [23]. V prípade prvých dvoch serverov je síce pravda, že poskytujú aj bezplatnú verziu, ale sú obmedzené na kapacitu

(pri serveri Microsoft SQL je to 10 GB, pri Oracle Database je to 12 GB), v prípade MySQL takéto obmedzenie nemáme. Čo sa týka obmedzenia kapacity, tak to je pre naše účely irelevantné, pretože ide o stále dostatočne veľkú kapacitu. Každý zo spomenutých databázových serverov spĺňa naše požiadavky, no v prípade prvých dvoch (komerčných) databázových serverov existuje väčšia šanca na to, že by prestali podporovať spomínané bezplatné verzie. Preto si kvôli väčšej stabilite volíme databázový server MySQL.

3.4.3 Object Relational Mapping

Object Relational Mapping (ORM) [24] je technika využívaná pri prepájaní objektovo orientovaného programovania (OOP) [25] a (vo väčšine prípadov) relačných databáz. Pri práci s relačnou databázou môžeme vytvárať, čítať, editovať a mazať dáta z databázy pomocou jazyka SQL (SQL skripty), ale takisto môžeme využiť ORM nástroje pre uľahčenie práce. Populárnymi ORM nástrojmi na platforme .NET sú napr. frameworky Dapper alebo Entity Framework Core (ďalej už len EF Core).

Ako bolo spomenuté, v našom programe by sme mohli využiť pre prácu s databázou („čisté“) SQL skripty. Výhodou tohto prístupu je väčšia kontrola nad SQL dotazmi, a rýchlosť vykonania dotazov. Nevýhodou je, že tento prístup nám neposkytuje automatické sanitizovanie dotazov [26], a takisto neposkytuje dopĺňanie kódu (nápovedu) ako napr. pri C# kóde vo Visual Studiu. Navyše je tento prístup dosť pracný, a to v tom zmysle, že by sme museli napísať veľa SQL kódu už len pre vytvorenie tabuliek. Alternatívou by bolo využitie nejakého ORM frameworku, napr. Dapper alebo EF Core, tento prístup by nám poskytol automatické dopĺňanie kódu vo Visual Studiu, pretože by sme pracovali so C# kódom. Obe frameworky si teraz porovnáme.

Dapper je jednoduchším frameworkom oproti frameworku EF Core s menším počtom funkcionalít. Umožňuje spúšťať SQL dotazy a ich výsledky mapovať na .NET objekty. Výhodou naďalej ostáva kontrola nad tvarom dotazu a rýchlosť vykonania dotazu, pretože stále pracujeme s dotazmi napísanými jazykom SQL. Ale naďalej ostáva nevýhodou, že by sme museli pracne pomocou jazyka SQL vytvárať všetky tabuľky.

EF Core je mohutnejší ORM framework s viacerými funkcionalitami. Jeho nevýhodou oproti predošlým možnostiam je, že vykonávanie dotazov je pomalšie, no stále ide o rozumné časy, ktoré pre nás nepredstavujú problém (porovnanie časovej výkonnosti spomínaných možností môžeme nájsť napr. v repozitári frameworku Dapper [27]). Oproti predošlým možnostiam nám však umožňuje využiť prístup code first [28] pri vytváraní tabuliek. Tento prístup nám umožní zamerať sa na doménu našej aplikácie, pričom bude stačiť vytvoriť pomocou C# kódu (doménové) modely (ide o C# triedy predstavujúce entity vytvorené v časti 3.4.1), a z nich si pomocou frameworku EF Core budeme schopný nechať vygenerovať SQL kód a vytvoriť tabuľky bez toho, aby sme museli manuálne písať SQL skripty. To nám môže potenciálne zjednodušiť a urýchliť implementáciu systému. Preto si zvolíme ako ORM framework EF Core. Viac informácií o frameworkoch Dapper a EF Core je možné nájsť napr. na stránke C# Corner [29].

3.5 Aukcia – odpočet a vyhodnocovanie

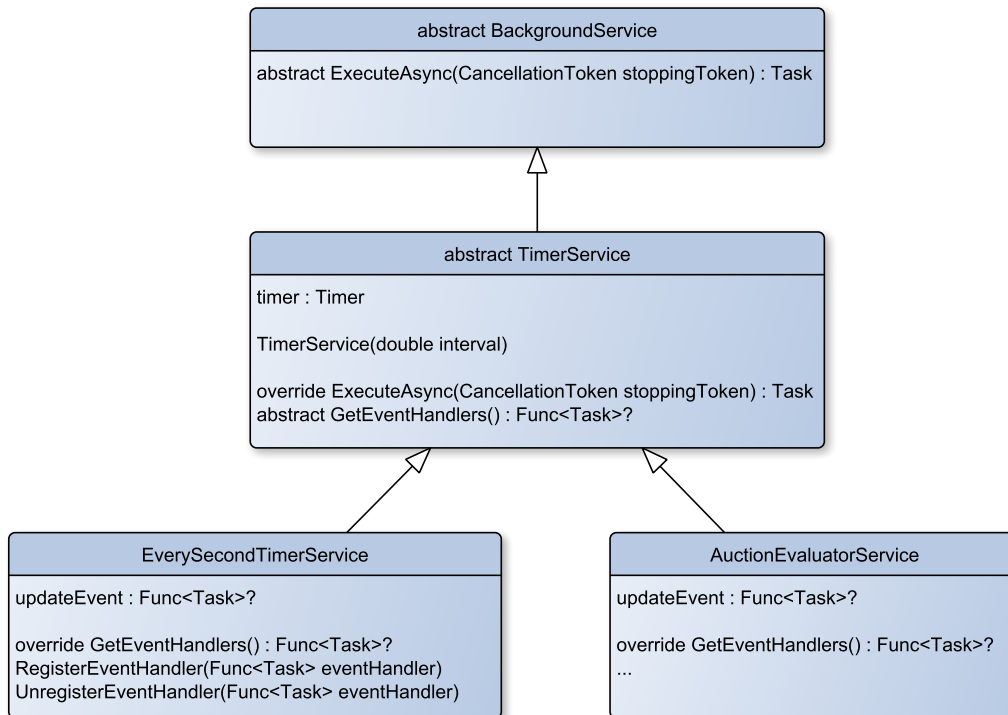
V požiadavke P4.2 (Odpočet a ďalšie údaje, viď v podkap. 1.1) sa vyžaduje odpočet do konca dražby. Túto požiadavku by sme splnili implementáciou častí aplikácie navrhnutých v častiach textu 2.5.1 (Aukčné ponuky, odpočet by sa nachádzal v spodnej časti vylisovaných kariet, viď obr. 2.14) a 2.5.2 (Detail aukčnej ponuky, odpočet by sa nachádzal v texte so základnými informáciami, viď vpravo hore na obr. 2.15). V tejto podkapitole si rozoberieme ako by sme mohli implementovať odpočet (a aj vyhodnocovanie aukčných dražieb) v spomínaných častiach aplikácie.

Keďže každá zo spomínaných častí aplikácie zobrazuje informácie o aukčnej ponuke, musí mať prístup k entite `AuctionOffer`, ktorú sme si zadefinovali v časti 3.4.1. Táto entita obsahuje okrem iného aj dátum, kedy končí aukčná ponuka. Ak od neho odčítame aktuálny dátum a čas (pomocou `DateTime.Now`), tak získame koľko času ostáva do konca aukčnej ponuky a tento údaj môžeme zobraziť užívateľovi. Lenže to nám nestačí, pre odpočet budeme potrebovať, aby sa nový údaj zobrazil každú sekundu. Na to by sme mohli využiť triedu `Timer` [30], ktorú nám platforma .NET ponúka. Po prejdení určitého intervalu (ktorý si vieme zadať pri vytvorení inštancie) vygeneruje udalosť. Takže ak si nastavíme interval jednu sekundu a zaregistrujeme si metódu, ktorá by prepočítala čas do konca aukčnej ponuky a prekreslila údaj na stránke, tak dosiahneme požadovaného odpočtu.

Implementácia by mohla byť realizovaná tak, že by pre každé miesto s odpočtom existovala jedna inštancia triedy `Timer`. To by však znamenalo mrhanie zdrojmi (viac informácií o uvoľňovaní zdrojov môžeme nájsť v dokumentácii [31]). Lepším riešením by bolo vytvoriť službu (triedu, ktorá by sa mohla volať napr. `EverySecondTimerService`), ktorá by nám dovolila registrovať ľubovoľné metódy (v našom prípade metódy pre prepočítanie a prekreslenie odpočtu) a každú sekundu by zavolała registrované metódy. Potom by nám stačilo vytvoriť jednu inštanciu spomínanej služby, ktorá by využívala jednu inštanciu triedy `Timer` a pomocou dependency injection [32], ktorý je vstavaný vo frameworku Blazor Server, si vyžiadať inštanciu triedy `EverySecondTimerService` z každého miesta s odpočtom v aplikácii podľa potreby.

Triedu `Timer` by sa nám tiež hodilo využiť aj pre vyhodnocovanie dražieb. Povedzme, že napr. každú minútu by systém prešiel všetky aukčné ponuky, ktoré by už neboli aktívne (t. j. prešiel ich termín konca) a pre každú z aukčných ponúk by sa zistilo, ktorý užívateľ ponúkol najvyššiu sumu a oboznámime ho, že vyhral. Taktiež by systém oboznámil ostatných účastníkov dražby s tým, že dražbu nevyhrali, a takisto by boli oboznámil aj administrátorov, že dražba skončila (a to aj v prípade, že skončila bez víťaza). Trieda, ktorá by bola zodpovedná za tento proces by sa mohla volať napr. `AuctionEvaluatorService`.

Zároveň si všimnime, že tu už máme dve triedy (`EverySecondTimerService` a `AuctionEvaluatorService`), ktoré by využívali triedu `Timer`. Obe chcú zadať interval (v prípade `EverySecondTimerService` ide o sekundu, v prípade `AuctionEvaluatorService` ide o minútu) a špecifikovať čo sa má diať (napr. pomocou `AuctionEvaluatorService` chceme každú minútu skontrolovať aukčné ponuky a v prípade potreby ich aj vyhodnotiť). Aby sme zamedzili duplicitnému kódu, tak by sme mohli vytvoriť rodičovskú triedu `TimerService` (viď obr. 3.2), ktorá by pokryla túto spoločnú logiku.



Obr. 3.2: Časové služby využívajúce triedu `Timer`.

Interval by sa mohol predať potomkami prostredníctvom parametra konštruktora triedy `TimerService` a pomocou abstraktnej metódy `GetEventHandlers` triedy `TimerService` by si potomkovia mohli zdefinovať čo sa má diať. Takže v prípade triedy `EverySecondTimerService` by to bolo prepočítanie a prekreslenie odpočtu (resp. zavolanie registrovaných metód) a v prípade triedy `AuctionEvaluatorService` by to bolo prejdenie aukčných ponúk a vyhodnotenie tých ktoré by boli neaktívne.

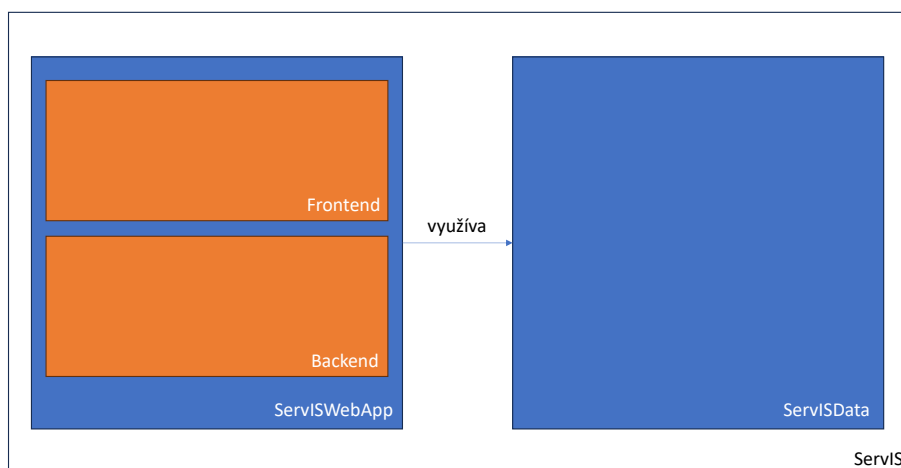
Ďalej je potrebné si uvedomiť, že kód oboch služieb (`EverySecondTimerService` a `AuctionEvaluatorService`) musí bežať na pozadí v osobitnom vlákne, aby neblokoval program a neznepríjemňoval tým užívateľovi prácu s aplikáciou. Pre implementáciu kódu bežiaceho na pozadí nám .NET poskytuje viacero možností, pre naše účely by sa hodili napr. rozhranie `HostedService` [33] alebo trieda `BackgroundService` [34]. Pre naše účely by sa dali využiť obe možnosti, no trieda `BackgroundService` je určená pre dlhotrvajúce úlohy bežiacie na pozadí [35], navyše autorovi vyhovuje viac možnosť s triedou `BackgroundService`, preto použijeme túto možnosť, a to tak, že trieda `TimerService` bude dediť od triedy `BackgroundService`.

Trieda `TimerService` zdedí od triedy `BackgroundService` abstraktnú metódu `ExecuteAsync` [36]. V triede `TimerService` by sme ju mohli naimplementovať tak, že po zavolaní by sa inicializovala inštancia triedy `Timer`, na ktorej by sa zavolala metóda `Start` [37]. Inicializácia by spočívala v tom, že na udalosť `Elapsed` [38] inštancie triedy `Timer` by sa zaregistrovali metódy (obsluhy udalostí) definované v potomkoch, tie vieme získať pomocou metódy `GetEventHandlers` (ktorú sme spomenuli skôr v texte). Ďalej nám už len stačí registrovať služby `EverySecondTimerService` a `AuctionEvaluatorService` pomocou metódy `AddHostedService` (viac informácií o implementácii služieb

bežiacich na pozadí môžeme nájsť v dokumentácii [39]). Potom po spustení aplikácie a registrácii služieb sa už framework postará o zavolanie našej implementácie metódy `ExecuteAsync` (a tým spustí na pozadí naše služby).

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 opísať správanie každého riadka kódu. Na to slúži kód samotný (prípadne komentáre, ktoré autor pridal na miesta, kde to uznal za vhodné). Kód aplikácie sa nachádza v elektronickej prílohe (pre prehľad obsahu elektronickej prílohy viď A.4) alebo je taktiež možné vidieť kód aplikácie v online repozitári (viď odkaz v prílohe A.1). Náš systém sa skladá z dvoch projektov – ServISWebApp a ServISData (viď obr. 4.1).



Obr. 4.1: Architektúra systému.

Zmyslom projektu ServISData je správa databázových entít a komunikácia s databázou. Projekt ServISWebApp predstavuje rozhranie pre užívateľov, a zároveň obsahuje logiku nášho systému. ServISWebApp využíva ServISData, aby sa dostal k dátam, ktoré potom môže užívateľom napr. zobrazovať. Navyše projekt ServISWebApp je rozdelený na dve časti – frontendová a backendová časť (rozdelenie bolo spomenuté v podkap. 3.2).

4.1 Adresárová štruktúra systému

V tejto podkapitole sa pozrieme na adresárovú štruktúru systému a povieme si na čo slúžia jednotlivé priečinky, resp. súbory v nich. Priečinky označené **modrou** farbou patria do frontendovej časti a priečinky označené **červenou** farbou patria do backendovej časti projektu ServISWebApp.

ServISWebApp

- Súbory v koreňovom priečinku projektu ServISWebapp - kód slúžiaci (zväčša) na konfiguráciu a spustenie programu
- **Auth** - triedy slúžiace na autentifikáciu užívateľa
- **BackgroundServices** - triedy procesov bežiacich na pozadí
- **Components** - komponenty využívané na stránkach z prieč. Pages
 - **Buttons** - komponenty reprezentujúce rôzne tlačidlá
 - **Forms** - komponenty reprezentujúce formuláre
 - **Managements** - obsahuje komponentu pre správu bagrov, prídavných zariadení atď.
- **CssProviders** - triedy zodpovedné za zmenu kaskádových štýlov vo formulároch
- **Pages** - stránky zložené z komponentov z priečinku Components
 - **Admin** - stránky určené výhradne pre administrátorov
- **Shared** - pomocné triedy využívané z rôznych miest programu
 - **Extensions** - obsahuje triedu s extension metódami

ServISData

- Súbory v koreňovom priečinku projektu ServISData - obsahuje kód pre prácu s databázou, rozhranie projektu ServISData, a takisto ďalší pomocný kód pre prácu s modelmi (triedy nachádzajúce sa v priečinku Models)
- **Attributes** - atribúty modelov (triedy, ktoré sa nachádzajú v priečinku Models)
- **DataOperations** - kód určený pre vykonávanie operácií s dátami (napr. filtrovanie, stránkovanie atď.)
- **Interfaces** - rozhrania (napr. rozhranie IItem, ktoré spĺňajú všetky modely (triedy nachádzajúce sa v priečinku Models))
- **Migrations** - priečinkov vygenerovaný frameworkom EF Core obsahujúci migrácie [40]
- **Models** - obsahuje modely (triedy reprezentujúce databázové entity)

4.2 Backendová časť

Backendová časť pokrýva logiku aplikácie mimo logiky užívateľského rozhrania, napr. vyhodnocovanie dražieb v aukcii. V tejto kapitole si prejdeme niektoré (významnejšie) z tried backendovej časti, ktoré sú využívané frontendovou časťou aplikácie.

4.2.1 Trieda CustomAuthenticationStateProvider

Táto trieda poskytuje metódy pre registráciu a prihlásenie užívateľa do systému. Takisto poskytuje metódy pre získanie informácií o prihlásenom užívateľovi.

4.2.2 Trieda PasswordHasher

Trieda obsahuje metódy na prácu s hashami. Umožňuje hashovať heslá, a tiež určovať, či hash patrí heslu.

4.2.3 Trieda ServISApi a rozhranie IServISApi

Trieda `ServISApi` predstavuje rozhranie (API [41]) projektu `ServISData`. Poskytuje metódy pre vytváranie/úpravu, mazanie a získavanie dát entít databázy opísaných v časti Návrh relačného modelu databázy (viď 3.4.1). Táto trieda je implementáciou rozhrania `IServISApi`. Pri registrácii `IServISApi` medzi služby, v súbore `Program.cs`, je ako implementácia tohto rozhrania dosadená trieda `ServISApi`. Preto keď z nejakého komponentu žiadame `IServISApi`, framework (dependency injection) nám poskytne inštanciu triedy `ServISApi`.

4.2.4 Trieda DataOperations

Trieda `DataOperations` sa využíva pre filtrovanie, zoradzovanie, stránkovanie alebo vyhľadávanie dát. V tejto triede existuje vnorená trieda `Configuration`, ktorá slúži pre nastavenie spomínaných operácií (napr. s jej pomocou vieme nastaviť, aby zoradenie dát bolo zostupne). Pomocou triedy `Configuration` je možné vytvoriť inštanciu triedy `DataOperations`, a tú je potom možné využiť v metódach, ktoré vracajú dáta (resp. kolekciu dát), napr. `DataOperations` sa využíva ako voliteľný parameter v niektorých metódach pre získavanie dát spomenutých v predchádzajúcej časti textu (viď 4.2.3).

4.2.5 Trieda EmailManager

Táto trieda poskytuje rôzne metódy pre prácu s emailami. Pomocou nej je možné napr. posilať emaily, získavať emaily z Gmailovej schránky, mazať emaily atď.

4.2.6 Trieda TimerService a jej potomkovia

Trieda `TimerService` je abstraktnou triedou, ktorá zakrýva prácu s triedou `Timer`. Umožňuje svojím potomkom definovať čo sa má vykonávať a v akom intervale.

Trieda `EverySecondTimerService` je potomkom triedy `TimerService`. Umožňuje svojím užívateľom (iným triedam) registrovať akciu, ktorá sa má vykonávať každú sekundu. V aplikácii sa využíva pre zobrazovanie odpočtu do konca aukčných ponúk (dražieb).

Trieda `AuctionEvaluatorService` je ďalším potomkom triedy `TimerService`. Každú minútu skontroluje všetky aukčné ponuky a tie, ktoré už skončili vyhodnotí (vyhodnocovanie už bolo opísané v podkapitole 3.5).

4.3 Frontendová časť

Frontendová časť má na starosti prezentovanie dát a interakciu s užívateľom. V tejto kapitole si prejdeme jednotlivé stránky webu a opíšeme si aké významnejšie triedy (komponenty) sa v nich využívajú.

4.3.1 Zobrazovanie obsahu podľa roly

Z požiadavky P1 (Roly užívateľa, viď v podkap. 1.1) vieme, že systém má rozlišovať medzi bežným zákazníkom a administrátorom, a podľa toho mu upraviť obsah na stránke. Z kapitoly Návrh užívateľského rozhrania (viď 2) vieme, že by mali existovať stránky, kde sa zobrazujú/schovávajú len určité časti. Napríklad v navigácii (viď vľavo na obr. 2.1) by sa mal zobrazovať odkaz „Správa webu“ iba administrátorom. Ale takisto existujú aj prípady, kedy by sme chceli obmedziť prístup na celú stránku. Napr. chceme, aby sa mohol na stránku pre vytvorenie (alebo úpravu) hlavnej ponuky (viď obr. 2.7) dostať iba administrátor.

Platforma .NET nám na to ponúka triedu `AuthorizeView` [42], presnejšie ide o komponent, pomocou ktorého vieme špecifikovať aký obsah (a komu, akej role) sa má zobrazovať na stránke. Takže napríklad v prípade navigácie vyzerá implementácia takto:

```
...
<div class="nav-item px-3">
  <NavLink ... href="kontakt">
    <span class="oi oi-plus" ..."></span> Kontakt
  </NavLink>
</div>
<AuthorizeView Roles="Administrator">
  <div class="nav-item px-3">
    <NavLink ... href="admin">
      <span class="oi oi-cog" ..."></span> Správa webu
    </NavLink>
  </div>
</AuthorizeView>
...
```

V prípade obmedzenia celej stránky využívame atribút `[Authorize]` [43]. Takže stačí ak do implementácie, napr. spomínanej stránky pre vytvorenie (alebo úpravu) hlavnej ponuky, dopíšeme atribút `[Authorize]` následovne:

```
@page "/admin/nova-ponuka"
@page "/admin/uprava-ponuky/{MainOfferId:int}"

@attribute [Authorize(Roles = "Administrator")]
...
```

4.3.2 Hlavné rozloženie stránky

Za hlavné rozloženie stránky, ktoré bolo navrhnuté v časti Hlavné rozloženie aplikácie (viď 2.1), je zodpovedný komponent `MainLayout`. V rámci tohto

komponentu sme využili komponent `SfDialogProvider` [44], ktorý nám umožní využívať potvrdzovacie modálne okná (tie si opíšeme v nasledujúcej časti textu, viď 4.3.3). Okrem toho sa v komponente `MainLayout` využíva komponent `NavMenu` (navigácia), a ten v sebe obsahuje komponenty `NavLink` (predstavujú odkazy v navigácii), `AuthorizeView` (viď 4.3.1) a `LoginPanel`.

Komponent `LoginPanel` slúži v prípade neprihláseného užívateľa na presmerovanie na stránky s formulármi pre prihlásenie alebo registráciu. V prípade prihláseného užívateľa slúži na presmerovanie na profil alebo na odhlásenie. Využíva preto komponent `AuthorizeView` (viď 4.3.1) a triedu `CustomAuthenticationStateProvider` (viď 4.2.1).

4.3.3 Modálne potvrdzovacie okno

V predchádzajúcej časti textu (viď 4.3.2) bolo v súvislosti s komponentom `SfDialogProvider` spomenuté potvrdzovacie modálne okno. Potvrdzovacie modálne okno bolo navrhnuté v podkapitole Modálne potvrdzovacie okno (viď 2.2) a naimplementované v triede `Modals` za pomoci služby (triedy) `SfDialogService` [45].

4.3.4 Stránky s podmienkami používania a zásadami ochrany osobných údajov

V podkapitole Sekcie Podmienky používania a Zásady ochrany osobných údajov (viď 2.3) boli navrhnuté stránky s podmienkami používania a zásadami ochrany osobných údajov. Implementáciu týchto stránok môžeme nájsť v súboroch `Terms.razor` a `PrivacyPolicy.razor`.

4.3.5 Stránka s hlavnou ponukou (domovská stránka)

Implementáciu časti aplikácie s vylistovanými kartami hlavnej ponuky (slúži ako domovská stránka), ktorá bola navrhnutá v časti Hlavná ponuka (viď 2.4.1), môžeme nájsť v súbore `Index.razor`. Stránka využíva komponenty `AuthorizeView` (viď 4.3.1) a `MainCard` (tento komponent predstavuje kartu hlavnej ponuky). Kód tejto stránky využíva pre získanie dát o hlavných ponukách (reprezentované triedou `MainOffer`) rozhranie (API) projektu `ServISData` (viď 4.2.3).

4.3.6 Stránka pre vytvorenie a úpravu hlavnej ponuky

Obe stránky, stránka pre vytvorenie novej, ale aj stránka pre úpravu existujúcej hlavnej ponuky (navrhnuté v časti 2.4.2), sú naimplementované v súbore `CreateMainOffer.razor`. Implementácia tejto stránky využíva atribút `[Authorize]` (viď 4.3.1) a komponent `MainOfferForm` (formulár pre vytváranie/úpravu hlavnej ponuky, ktorá je v kóde reprezentovaná triedou `MainOffer`). Komponent `MainOfferForm` využíva pre výber typu bagra (v kóde reprezentovaný triedou `ExcavatorType`) komponent `ItemsSelector`. V prípade, že ide o úpravu existujúcej hlavnej ponuky, jej dáta sa načítajú pomocou rozhrania projektu `ServISData` (viď 4.2.3) a predajú formuláru.

4.3.7 Stránka s ponukou (nových) bagrov

Návrh tejto časti aplikácie bol opísaný v časti textu Ponuka (nových) bagrov (viď 2.4.3). Implementácia stránky sa nachádza v súbore `Excavators.razor` a obsahuje komponenty `AuthorizeView` (viď 4.3.1) a `ExcavatorLister`. Pomocou rozhrania projektu `ServISData` (viď 4.2.3) sa získajú objekty typu `Excavator` reprezentujúce bagre a predajú sa už spomínanej komponente `ExcavatorLister` na vylistovanie – pre každý bager sa zobrazí karta bagra (komponent `ExcavatorCard`). V návrhu tejto časti aplikácie bolo tiež spomenuté, že sa majú vylistovať karty bagrov určitého typu, ktoré neboli určené iba pre aukciu. Dosiahli sme toho pomocou tried `DataOperations` a (jej vnorenej triedy) `Configuration`, ktoré implementácia stránky taktiež obsahuje (tieto triedy boli spomenuté už skôr v časti 4.2.4).

4.3.8 Stránka s ponukou prídavných zariadení

Návrh tejto stránky sa nachádza v časti textu Ponuka prídavných zariadení (viď 2.4.4). Implementácia stránky sa nachádza v súbore `AdditionalEquipments.razor` a obsahuje komponenty `AuthorizeView` (viď 4.3.1) a `AdditionalEquipmentLister`. Komponent `AdditionalEquipmentLister` slúži na vylistovanie kariet prídavných zariadení (komponenty `AdditionalEquipmentCard`), dáta prídavných zariadeniach (reprezentované triedami `AdditionalEquipment`) získava z rozhrania projektu `ServISData` (viď 4.2.3)

4.3.9 Stránka s detailom bagra, prídavného zariadenia

Táto časť aplikácie bola navrhnutá v časti Detail bagra, prídavného zariadenia (viď 2.4.5). Implementáciu stránky s detailom bagra môžeme nájsť v súbore `ExcavatorDetail.razor` a implementáciu stránky s detailom prídavného zariadenia môžeme nájsť v súbore `AdditionalEquipmentDetail.razor`. Obidve implementácie využívajú komponenty `AuthorizeView` (viď 4.3.1), `PhotoSlider` (pre zobrazenie fotiek) a `DemandForm` (formulár, ktorého návrh bol opísaný v časti 2.4.6). Komponent `DemandForm` využíva na odoslanie správy (emailu) triedu `EmailManager` (viď 4.2.5). Ďalej obe implementácie stránok čerpajú dáta na zobrazenie z rozhrania projektu `ServISData` (viď 4.2.3). V prípade stránky s detailom bagra sa získa trieda `Excavator` (obsahujúca dáta bagra) a v prípade stránky s detailom prídavného zariadenia sa získa trieda `AdditionalEquipment` (obsahujúca dáta prídavného zariadenia).

4.3.10 Stránka pre vytvorenie a úpravu bagra

Návrh tejto časti aplikácie môžeme nájsť v časti textu Vytvorenie nového a úprava existujúceho bagra (viď 2.4.7). Implementácia sa nachádza v súbore `CreateExcavator.razor` a obsahuje atribút `[Authorize]` (viď 4.3.1) a komponent `ExcavatorForm`. V prípade, že ide o úpravu existujúceho bagra, sa pomocou rozhrania projektu `ServISData` (viď 4.2.3) načítajú a následne predajú dáta komponente `ExcavatorForm`. `ExcavatorForm` je formulár pre vytváranie/úpravu bagra (bager je reprezentovaný triedou `Excavator`). V návrhu je napísané, že

do formulára by malo byť možné nahráť viacero fotiek, vybrať typ bagra, a tiež vybrať aké náhradné diely patria do bagra. Kvôli tomu formulár obsahuje komponenty `InputItemPhotos` (pre nahratie fotiek), `ItemsSelector` (pre výber typu bagra) a `ChecklistTable` (pre výber náhradných dielov).

4.3.11 Stránka pre vytvorenie a úpravu prídavného zariadenia

Implementácia časti aplikácie, ktorá bola navrhnutá v časti textu Vytvorenie nového a úprava existujúceho prídavného zariadenia (viď 2.4.9), obsahuje atribút `[Authorize]` (viď 4.3.1) a komponent `AdditionalEquipmentForm`. V prípade, že ide o úpravu existujúceho prídavného zariadenia, sa pomocou rozhrania projektu `ServISData` (viď 4.2.3) načítajú a následne predajú dáta komponente `AdditionalEquipmentForm`. `AdditionalEquipmentForm` je formulár pre vytváranie/úpravu prídavného zariadenia (prídavné zariadenie je reprezentované triedou `AdditionalEquipment`). V návrhu je napísané, že do formulára by malo byť možné nahráť viacero fotiek, vybrať značku a kategóriu prídavného zariadenia, ale aj vybrať typ bagra, pre ktorý je prídavné zariadenie určené. Kvôli tomu formulár obsahuje komponenty `InputItemPhotos` (pre nahratie fotiek) a `ItemsSelector` (túto komponentu obsahuje dokonca trikrát, pre výber značky a kategórie prídavného zariadenia, ale tiež aj pre výber typu bagra).

4.3.12 Stránka s aukčnými ponukami

Návrh stránky s aukčnými ponukami, ktorá bola navrhnutá v časti Aukčné ponuky (viď 2.5.1), sa nachádza v súbore `AuctionOffers.razor`. Implementácia stránky využíva komponent `AuthorizeView` (viď 4.3.1) a na vylistovanie kariet aukčných ponúk (komponentov typu `AuctionOfferCard`, ktoré v sebe obsahujú pre zobrazenie odpočtu komponent `CountdownDisplayer` využívajúci triedu `EverySecondTimerService` spomenutú v časti 4.2.6) využíva komponent `AuctionOffersLister`. `AuctionOffersLister` získava dáta aukčných ponúk (reprezentované triedami typu `AuctionOffer`) z rozhrania projektu `ServISData` (viď 4.2.3).

4.3.13 Stránka s detailom aukčnej ponuky

Stránka s detailom aukčnej ponuky, ktorej návrh je opísaný v časti Detail aukčnej ponuky (viď 2.5.2), je implementovaná v súbore `AuctionOfferDetail.razor`. Stránka využíva komponenty `AuthorizeView` (viď 4.3.1), `PhotoSlider` (pre zobrazenie fotiek), `CountdownDisplayer` (pre zobrazenie odpočtu, využíva na to triedu `EverySecondTimerService` spomenutú v časti 4.2.6) a `AuctionBidForm`. Stránka čerpá dáta aukčnej ponuky na zobrazenie z rozhrania projektu `ServISData` (viď 4.2.3). Na stránke sa tiež využíva trieda obsahujúca dáta o aukčnej ponuke – `AuctionOffer`. Okrem toho implementácia stránky obsahuje aj komponent `AuctionBidForm`, ktorého návrh bol opísaný v časti 2.5.3. Komponent `AuctionBidForm` je formulár, ktorý pracuje s triedou `AuctionBid` (predstavuje zákazníkovú ponúknutú sumu do dražby).

4.3.14 Stránka pre vytvorenie a úpravu aukčnej ponuky

Obidve stránky, stránka pre vytvorenie novej, ale aj stránka pre úpravu existujúcej aukčnej ponuky (navrhnuté v časti 2.5.4), sú naimplementované v súbore `CreateAuctionOffer.razor`. Implementácia tejto stránky využíva atribút `[Authorize]` (viď 4.3.1) a komponent `AuctionOfferForm` (formulár pre vytváranie/úpravu aukčnej ponuky, ktorá je v kóde reprezentovaná triedou `AuctionOffer`). V návrhu stránky je spomenuté, že formulár by mal umožniť výber bagra (v kóde reprezentovaný triedou `Excavator`) do dražby – na to `AuctionOffer` využíva komponent `ItemsSelector`. V prípade, že ide o úpravu existujúcej aukčnej ponuky, jej dáta sa načítajú pomocou rozhrania projektu `ServISData` (viď 4.2.3) a predajú formuláru.

4.3.15 Stránky Služby, Kontakt a O nás

Návrh stránky Služby bol opísaný v podkapitole Splnenie P8 (viď 2.6). Implementáciu stránky môžeme nájsť v súbore `Services.razor`, využíva komponent `DemandForm`.

Ďalej návrh stránok Kontakt a O nás bol opísaný v podkapitole Splnenie P9 (viď 2.7). Implementáciu stránky Kontakt môžeme nájsť v súbore `Contact.razor` a implementáciu stránky O nás môžeme nájsť v súbore `About.razor`.

4.3.16 Stránka pre správu webu

V podkapitole Splnenie P5, P7 a správa predmetov (viď 2.8) bola navrhnutá emailová schránka a časť aplikácie pre správu obsahu webu (administrátorská stránka). Implementácia administrátorskej stránky sa nachádza v súbore `AdministratorPage.razor` a využíva atribút `[Authorize]` (viď 4.3.1), komponent `TabControl` s komponentmi `TabPage` (tvoria panel s kartami Správy, Náhradné diely atď., viď vrchnú časť obr. 2.23).

Na karte Správy sa zobrazuje komponent `Message`, ktorý v sebe obsahuje komponenty `ThreadsView` (pohľad s vylistovanými konverzáciami, ktoré sú reprezentované komponentom `ThreadRow`), `ThreadView` (pohľad, ktorý sa zobrazí pri otvorení konverzácie, zobrazí správy v konverzácii reprezentované komponentmi typu `Message`) a `MessageSettings` (pohľad, ktorý sa zobrazí pri prekliknutí do časti s nastaveniami správ, využíva rozhranie projektu `ServISData`, viď 4.2.3 a komponenty typu `AutogeneratedMessageForm` pre editáciu tvaru automaticky generovaných správ, ktoré sú v kóde reprezentované triedou `AutogeneratedMessage`). Ďalej sa na karte pre získanie správ z Gmailu využíva trieda `EmailManager` (viď 4.2.5).

Ostatné karty (Náhradné diely, Bager atď.) zobrazujú komponent `ItemsManagement`, ktorý predstavuje tabuľku opísanú v časti 2.8.4. `ItemsManagement` tiež využíva rôzne komponenty na zobrazenie formulára podľa toho, na akej karte sa užívateľ nachádza. Teraz prejdeme zbytok kariet a opíšeme si, aké komponenty využívajú:

- **Karta Náhradné diely**

Táto karta využíva komponent `SparePartForm`, ktorý predstavuje formulár pre vytváranie/úpravu prídavných zariadení – tie sú reprezentované trie-

dami typu `SparePart`. V časti Karta Náhradné diely (viď 2.8.4) bolo spomenuté, že formulár by mal umožniť zadať do akých bagrov náhradný diel patrí, a preto `SparePartForm` obsahuje komponent `ChecklistTable`, ktorý to umožňuje. Pre získanie bagrov a v prípade upravovania existujúceho náhradného dielu pre získanie dát o upravovanom náhradnom diele sa využíva rozhranie projektu `ServISData` (viď 4.2.3).

- **Karta Bagre**

Ďalej táto karta využíva komponent `ExcavatorForm`, ktorý predstavuje formulár pre vytváranie/úpravu bagrov, ktoré sú reprezentované triedami typu `Excavator`. Komponent `ExcavatorForm` bol podrobnejšie opísaný v časti Stránka pre vytvorenie a úpravu bagra (viď 4.3.10).

- **Karta Typy bagrov**

Karta využíva komponent `ExcavatorTypeForm`, ktorý predstavuje formulár pre vytváranie/úpravu typov bagrov, a tie sú reprezentované triedami typu `ExcavatorType`. V časti Karty Bagre, Typy bagrov, Typy vlastností bagrov (viď 2.8.5) bolo spomenuté, že formulár by mal umožniť výber značky bagra a kategórie bagra, a preto `ExcavatorTypeForm` obsahuje dva komponenty typu `ItemsSelector`, ktoré to umožňujú. Okrem toho bolo napísané, že by mal formulár umožniť výber typov vlastností, ktoré bude mať vytváraný (resp. upravovaný) typ bagra. Kvôli tomu formulár tiež využíva komponent `ChecklistTable`. Pre získanie typov vlastností a v prípade upravovania existujúceho typu bagra pre získanie dát o upravovanom type bagra sa využíva rozhranie projektu `ServISData` (viď 4.2.3).

- **Karta Typy vlastností bagrov**

Tu sa využíva komponent `ExcavatorPropertyTypeForm` predstavujúci formulár pre vytváranie/úpravu typov vlastností bagrov, ktoré sú reprezentované triedami typu `ExcavatorPropertyType`.

- **Karta Kategórie a značky bagrov**

Na tejto karte sa nachádzajú dve tabuľky, jedna je pre správu kategórií bagrov, druhá pre správu značiek bagrov. Preto jedna tabuľka využíva komponent `ExcavatorCategoryForm` predstavujúci formulár pre vytváranie/úpravu kategórií bagrov, ktoré sú reprezentované triedami typu `ExcavatorCategory` a druhá tabuľka využíva komponent `ExcavatorBrandForm` predstavujúci formulár pre vytváranie/úpravu značiek bagrov, ktoré sú reprezentované triedami typu `ExcavatorBrand`.

- **Karta Kategórie a značky prídavných zariadení**

Podobne ako v predošlej karte aj tu sa nachádzajú dve tabuľky, jedna je pre správu kategórií prídavných zariadení, druhá pre správu značiek prídavných zariadení. Preto jedna tabuľka využíva komponent `AdditionalEquipmentCategoryForm` predstavujúci formulár pre vytváranie/úpravu kategórií prídavných zariadení, ktoré sú reprezentované triedami typu `AdditionalEquipmentCategory` a druhá tabuľka využíva komponent `AdditionalEquipmentBrandForm` predstavujúci formulár

pre vytváranie/úpravu značiek prídavných zariadení, ktoré sú reprezentované triedami typu `AdditionalEquipmentBrand`.

4.3.17 Stránka s prihlasovacím formulárom

Návrh tejto časti aplikácie bol opísaný v časti Prihlásenie (viď 2.9.2). Jej implementácia sa nachádza v súbore `Login.razor`. Pre získanie dát užívateľa (napr. aby sme zistili, či je prihlasovaný užívateľ vôbec evidovaný v systéme) sa využíva rozhranie projektu `ServISData` (viď 4.2.3). Pre skontrolovanie správnosti hesla sa využíva trieda `PasswordHasher` (viď 4.2.2), ďalej pre samotné prihlásenie sa využíva trieda `CustomAuthenticationStateProvider` (viď 4.2.1).

4.3.18 Stránka s registračným formulárom

Táto časť aplikácie bola navrhnutá v časti Registrácia (viď 2.9.3). Jej implementácia sa nachádza v súbore `Register.razor` a obsahuje formulár pre registráciu užívateľov reprezentovaný komponentom `UserForm`. `UserForm` pracuje s triedou `User`, ktorá predstavuje dáta užívateľa. Pomocou triedy `PasswordHasher` (viď 4.2.2) vytvorí hash hesla užívateľa a pomocou rozhrania projektu `ServISData` (viď 4.2.3) ho zaregistruje do systému (resp. uloží jeho dáta do databázy).

4.3.19 Stránka s profilom užívateľa

V časti Profil (viď 2.9.4) bola navrhnutá časť aplikácie s profilom užívateľa, jej implementácia sa nachádza v súbore `Profile.razor` a obsahuje komponent `UserForm`, predstavujúci formulár s informáciami o užívateľovi, ktorý už bol opísaný v predchádzajúcej časti textu (viď 4.3.18). Takisto využíva triedu `CustomAuthenticationStateProvider` (viď 4.2.1) pre získanie informácií o užívateľovi, ktoré predá komponentu `UserForm` na zobrazenie a prípadnú editáciu.

5. Uživatelská dokumentácia

Návod pre zákazníkov ako pracovať s aplikáciou je možné nájsť v podobe videí na adrese v prílohe (viď A.2), takisto je v prílohe možné nájsť aj videonávody na používanie aplikácie pre administrátorov (viď A.3). Čo sa týka procesov spustenia systému pre lokálne testovanie a nasadenia na internet, tak tie si opíšeme v tejto kapitole.

5.1 Databáza

Aplikácia ServIS využíva pre ukladanie dát databázu MySQL, ktorá nie je súčasťou aplikácie, a preto je nutné si ju osobitne stiahnuť a nainštalovať. Odkaz na stiahnutie je možné nájsť na stránkach databázy [23], rovnako ako aj návod na inštaláciu [46].

5.2 Predpoklady pre lokálne spustenie

Pre úspešné nainštalovanie, spustenie a fungovanie aplikácie budeme potrebovať:

- Počítač s operačným systémom Windows 10 alebo vyššie
- Prístup na internet
- Visual Studio 2022 [47] (budeme potrebovať .NET 6 alebo vyššie)
- Databáza MySQL (viac v podkapitole 5.1)

5.3 Návod na lokálne spustenie

V tejto podkapitole si povieme ako spustiť aplikáciu lokálne. Návod sa spoľieha, že sú splnené všetky predpoklady z podkapitoly 5.2, a takisto na to, že čitateľ už má k dispozícii repozitár aplikácie (jednou z možností ako získať repozitár aplikácie je stiahnuť si ho z online repozitára, pre odkaz naň viď prílohu A.1).

1. V priečinku (repozitári) aplikácie si dvojitým kliknutím na `ServIS.sln` otvoríme aplikáciu v programe Visual Studio 2022.
2. V `appsettings.json` sa nachádza konfigurácia aplikácie, kde je možné nastaviť napr. názov firmy, aký má byť minimálny rozdiel medzi ponukami v aukcii atď. Nastavíme ju podľa svojich údajov (`Logging` a `AllowedHosts` môžeme ignorovať).

3. Pravým tlačidlom myši klikneme na projekt ServISData, a potom klikneme na „Manage User Secrets“. Zobrazí sa nám súbor, ktorého obsah by mal vyzeráť takto:

```
{
  "ConnectionStrings": {
    "Default": "CONNECTION_STRING"
  }
}
```

Namiesto `CONNECTION_STRING` vložíme svoj connection string databázy (stačí ak využijeme štandardný tvar connection stringu [48]).

4. Pravým tlačidlom myši klikneme na projekt ServISWebApp, a potom klikneme na „Manage User Secrets“. Zobrazí sa nám súbor, ktorého obsah by mal vyzeráť takto:

```
{
  "SyncfusionLicenseKey": "LICENSE_KEY",
  "EmailAppPassword": "APP_PASSWORD"
}
```

Namiesto `LICENSE_KEY` vložíme svoj licenčný kľúč (postup získania kľúča je opísaný v dokumentácii spoločnosti Syncfusion [49]). Namiesto `APP_PASSWORD` vložíme svoje emailové heslo aplikácie (postup získania emailového hesla aplikácie je opísaný na stránkach podpory spoločnosti Google [50]).

5. Kliknutím na zelený play button v hornej časti Visual Studia sa program preloží a spustí. Po nejakej chvíli by sa nám malo zapnúť okno internetového prehliadača s našou aplikáciou.

5.4 Nasadenie do Azure

Čo sa týka reálneho nasadenia na internet, tak jednou z populárnych možností je nasadenie do cloudovej platformy Azure spoločnosti Microsoft [51]. Aplikácia napísaná pomocou frameworku Blazor Server sa spúšťa z ASP.NET Core aplikácie, preto je jej nasadenie podobné ako v prípade ASP.NET Core aplikácie [52]. Kvôli tomu je možné pre nasadenie nášeho systému využiť návod na nasadenie ASP.NET aplikácie s databázou opísaný v dokumentácii [53] s menšími obmenami – na stránke „Create Web App + Database“ si namiesto možnosti „Azure SQL Database“ vyberieme „MySQL – Flexible Server“, a takisto v časti „Configuration“ vyplníme potrebné kľúče s hodnotami, ktoré boli spomenuté v krokoch 2., 3. a 4. v predošlej podkapitole (vid 5.3). Navyše je možné preskočiť prácu s Azure Cache for Redis, pretože ju v programe nijako nevyužívame.

6. Záver

V závere zhodnotíme ako sa nám podarilo naplniť požiadavky definované v Úvode, konkrétne v podkapitole Požiadavky na systém (viď 1.1).

- **P1 Roly užívateľa**

Aplikácia rozlišuje zákazníka a administrátora, a podľa toho zobrazuje obsah. Administrátorovi nezobrazuje napr. na stránke s detailom bagra tlačidlo pre dopyt (a teda ani formulár pre dopyt) a zákazníkovi aplikácia nezobrazí napr. tlačidlá pre vymazanie hlavnej ponuky na domovskej stránke.

- **P2 Predstavenie ponuky zákazníkom**

Systém je schopný načítať z databázy hlavné ponuky, bagre, prídavné zariadenia a zobraziť ich užívateľom. Takisto umožňuje administrátorom tieto položky okrem pridávania aj upravovať alebo mazať.

- **P3 Posielanie dopytu**

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

- **P4 Aukcia**

V aplikácii existuje aukcia bagrov. Administrátorom je umožnené vytvárať aukčné ponuky, v ktorých sa draží (administrátorom vybraný) bager. Zákazníkom je v prípade záujmu umožnené ponúkať sumy do dražby. Po skončení odpočtu prebehne vyhodnocovanie aukčných ponúk, kde sa rozhodne kto je ich víťazom. Víťaz je oboznámený prostredníctvom emailu, že vyhral. Porazení sú informovaní o skutočnosti, že sa im aukciu nepodarilo vyhrať. Administrátor je upozornený, že aukčná ponuka skončila a kto je jej víťazom. Ak aukcia skončila bez víťaza (nikto sa jej nezúčastnil), tak administrátor je aj v tomto prípade o tom upozornený a dražba je reštartovaná (termín konca aukčnej ponuky je posunutý o týždeň).

- **P5 Správy**

V aplikácii sa nachádza emailová schránka pripomínajúca Gmail. Ak v aplikácii administrátor otvorí nejakú z konverzácií, ktorej prvý email (správa) pochádza z našej aplikácie, tak sa pri prvom emaile bude nachádzať vložené okno prepájajúce stránku aplikácie odkiaľ bol email odoslaný. Ďalej sa v aplikácii nachádzajú nastavenia správ, kde si môže administrátor meniť formát automaticky generovaných správ.

- **P6 Registrácia a prihlásenie užívateľov**

Systém umožňuje bežným užívateľom vytvoriť si účet v systéme, a takisto sa doň prihlásiť (možnosť odhlásenia je samozrejmosťou). A po prihlásení už nemusia do formulárov (pri posielaní dopytu alebo pri zapájaní sa do aukcie) zadávať svoje údaje. V aplikácii sa takisto nachádza aj profil užívateľa, kde si môže upravovať svoje údaje.

- **P7 Prístup k súčiastkam strojov**

V aplikácii si vie administrátor v sekcii Správa webu po kliknutí na kartu Bagre rozkliknúť detail nejakého konkrétneho bagra. Okrem iných údajov o spomínanom bagri sa mu zobrazí aj zoznam náhradných dielov, ktoré bager obsahuje. Podobne na karte Náhradné diely sa nachádza zoznam náhradných dielov. Okrem pridávania nových náhradných dielov tam môže administrátor upravovať existujúce náhradné diely. Navyše po rozkliknutí detailu nejakého z náhradných dielov sa mu zobrazí zoznam bagrov, ktoré obsahujú daný náhradný diel.

- **P8 Objednávanie výkopových prác**

V aplikácii sa nachádza sekcia (stránka) Služby, kde okrem informácií o službách firmy (výkopových prácach) sa nachádza aj formulár, ktorým môžu užívatelia požiadať o služby firmy.

- **P9 Sekcie O nás a Kontakt**

V aplikácii sa tiež nachádzajú sekcie (Stránky) O nás s miestom pre informácie o firme (napr. jej histórii) a Kontakt s miestom pre telefónne čísla a emaily na firmu (prípadne aj na pridružené firmy).

- **P10 Dostupnosť**

Náš systém je webovou aplikáciou, ktorá je dostupná pre užívateľov odkiaľkoľvek (samozrejme za predpokladu, že majú prístup k internetu). Čo sa týka problému s kompatibilitou spojenou so starými počítačmi so starým softvérom, ktoré by mohli firmy využívať, tak na to sme brali ohľad počas analýzy. Konkrétne pri výbere frameworku pre vývoj webovej aplikácie (viď podkapitolu 3.3).

- **P11 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.

6.1 GDPR

V podkapitole Sekcie Podmienky používania a Zásady ochrany osobných údajov (viď 2.3) bolo spomenuté, že by bolo vhodné, aby naša aplikácia spĺňala GDPR. Autor nie je právnikom, a preto nemôže prehlásiť, že aplikácia spĺňa GDPR, ale v snahe pripraviť ju pre splnenie tejto regulácie sa vytvorili stránky pre vloženie právnických textov (podmienky používania a zásady ochrany osobných údajov), a takisto sa na profil užívateľa pridali tlačidlá – jedno pre vymazanie svojho účtu zo systému a druhé pre exportovanie (stiahnutie) vlastných dát v textovej forme. Zároveň aplikácia užívateľa upozorňuje na podmienky a zásady používania systému v častiach, kde poskytuje svoje osobné údaje. Ďalej užívateľské dáta síce nie sú šifrované softvérovou, ale táto podmienka sa dá splniť pri výbere hostingu. Stačí si vybrať hosting, ktorý ponúka disky podporujúce šifrovanie dát.

6.2 Možné vylepšenia

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

- **Výkonnosť listovania ponuky**

Ponuku firmy (napr. bagre) prečítame z databázy a vylistujeme jednotlivé položky. Nevyužívame žiadnu virtualizáciu, ani stránkovanie. Keďže systém je určený pre malé firmy a nepredpokladá sa veľká ponuka, tak táto skutočnosť nepredstavuje problém. No v budúcnosti, ak by sa firme používajúcej náš systém darilo a rozhodla by sa rozšíriť ponuku, mohla by sa virtualizácia alebo stránkovanie hodiť.

- **Upozornenie na prehliadku bagra**

Vo svete to funguje tak, že keď si užívateľ zakúpi bager, tak po určitom čase by firma mala prísť na prehliadku a bager skontrolovať. Preto by sme do nášho systému mohli pridať funkcionalitu, ktorá by fungovala následovne. Ak užívateľ odošle dopyt na nejaký stroj a obchod prebehne úspešne, admin by túto skutočnosť zaznačil v systéme, čím by užívateľovi priradil dopytovaný bager. Potom by sa spustil časovač, ktorý by po uplynutí definovaného času upozornil administrátora na blížiacu sa prehliadku.

Seznam použité literatury

- [1] Stack Exchange: What is a Modal Dialog Window? <https://ux.stackexchange.com/questions/12045/what-is-a-modal-dialog-window>
- [2] Úrad pre ochranu osobných údajov: Všeobecné nariadenie o ochrane osobných údajov (GDPR). <https://www.uoou.cz/obecne-narizeni-o-ochrane-osobnich-udaju-gdpr/ds-3938/p1=3938>
- [3] Cloudflare: What is BaaS? <https://www.cloudflare.com/en-gb/learning/serverless/glossary/backend-as-a-service-baaS/>
- [4] MDN Web Docs Glossary: SPA (Single-page application). <https://developer.mozilla.org/en-US/docs/Glossary/SPA>
- [5] Mailchimp glossary: What is SEO? <https://mailchimp.com/marketing-glossary/seo/>
- [6] Novateus: Single Page Applications, most discussed pros and cons. <https://novateus.com/blog/single-page-applications-spas-most-discussed-pros-cons/>
- [7] Cloudways: How to Perform SEO for Single Page Applications. <https://www.cloudways.com/blog/single-page-website-spa-seo/>
- [8] Oberlo: Most popular web browsers in 2023. <https://www.oberlo.com/statistics/browser-market-share>
- [9] Can I Use: WebAssembly. <https://caniuse.com/wasm>
- [10] MSFN: Chrome 49 Update. <https://msfn.org/board/topic/175404-chrome-49-update/>
- [11] Chrome Blog: November updates to Chrome platform support (2015). <https://chrome.googleblog.com/2015/11/updates-to-chrome-platform-support.html>
- [12] Chrome Releases: September updates from (2015). <https://chromereleases.googleblog.com/2015/09/>
- [13] LinkedIn: Single-page application vs. Multi-page application. <https://www.linkedin.com/pulse/single-page-application-vs-multi-page-what-better-your-/>
- [14] Microsoft .NET: ASP.NET MVC Pattern. <https://dotnet.microsoft.com/en-us/apps/aspnet/mvc>
- [15] Stack Overflow: Creating reusable HTML view components using Razor in ASP.NET MVC. <https://stackoverflow.com/questions/23518499/creating-reusable-html-view-components-using-razor-in-asp-net-mvc>

- [16] Microsoft documentation: Blazor Server. <https://learn.microsoft.com/en-us/aspnet/core/blazor/hosting-models?view=aspnetcore-7.0#blazor-server>
- [17] Microsoft Azure: Co je relační databáze? <https://azure.microsoft.com/cs-cz/resources/cloud-computing-dictionary/what-is-a-relational-database/#whatis>
- [18] Microsoft Azure: Databáze NoSQL – co je NoSQL? <https://azure.microsoft.com/cs-cz/resources/cloud-computing-dictionary/what-is-nosql-database/>
- [19] Coursera: Relational vs. Non-relational Database. <https://www.coursera.org/articles/relational-vs-non-relational-database#2-when-to-use-a-relational-vs-a-non-relational-database>
- [20] GeeksforGeeks: Relational Model in DBMS. <https://www.geeksforgeeks.org/relational-model-in-dbms/>
- [21] Microsoft: SQL Server 2022 comparison. <https://www.microsoft.com/en-us/sql-server/sql-server-2022-comparison>
- [22] Oracle: Oracle Database XE. <https://www.oracle.com/database/technologies/appdev/xe.html>
- [23] MySQL: Community Server. <https://dev.mysql.com/downloads/mysql/>
- [24] freeCodeCamp: What is an ORM – The Meaning of Object Relational Mapping Database Tools. <https://www.freecodecamp.org/news/what-is-an-orm-the-meaning-of-object-relational-mapping-database-tools>
- [25] freeCodeCamp: The Four Pillars of Object-Oriented Programming. <https://www.freecodecamp.org/news/four-pillars-of-object-oriented-programming/>
- [26] LinkedIn: How do you handle user input validation and sanitization to avoid SQL injection vulnerabilities? <https://www.linkedin.com/advice/0/how-do-you-handle-user-input-validation-sanitization>
- [27] GitHub: Dapper repository. <https://github.com/DapperLib/Dapper/#performance>
- [28] Entity Framework Tutorial: What is Code-First? <https://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx>
- [29] C# Corner: Dapper vs. Entity Framework Core. <https://www.c-sharpcorner.com/article/dapper-vs-entity-framework-core/>
- [30] Microsoft documentation: Timer Class. <https://learn.microsoft.com/en-us/dotnet/api/system.timers.timer?view=net-6.0>
- [31] Microsoft documentation: Timer.Dispose Method. <https://learn.microsoft.com/en-us/dotnet/api/system.threading.timer.dispose?view=net-6.0>

- [32] Microsoft documentation: ASP.NET Core Blazor dependency injection. <https://learn.microsoft.com/en-us/aspnet/core/blazor/fundamentals/dependency-injection>
- [33] Microsoft documentation: IHostedService interface. <https://learn.microsoft.com/en-us/aspnet/core/fundamentals/host/hosted-services?view=aspnetcore-7.0&tabs=visual-studio#ihostedservice-interface>
- [34] Microsoft documentation: BackgroundService base class. <https://learn.microsoft.com/en-us/aspnet/core/fundamentals/host/hosted-services?view=aspnetcore-7.0&tabs=visual-studio#backgroundservice-base-class>
- [35] Compositional IT: Background Tasks in .NET. <https://www.compositional-it.com/news-blog/background-tasks-in-net>
- [36] Microsoft documentation: ExecuteAsync(CancellationToken) Method of BackgroundService Class. <https://learn.microsoft.com/en-us/dotnet/api/microsoft.extensions.hosting.backgroundservice.executeasync?view=dotnet-plat-ext-7.0>
- [37] Microsoft documentation: Timer.Start Method. <https://learn.microsoft.com/en-us/dotnet/api/system.timers.timer.start>
- [38] Microsoft documentation: Timer.Elapsed Event. <https://learn.microsoft.com/en-us/dotnet/api/system.timers.timer.elapsed>
- [39] Microsoft documentation: Implement the IHostedService interface. <https://learn.microsoft.com/en-us/dotnet/core/extensions/timer-service?pivots=dotnet-7-0>
- [40] Microsoft documentation: Migrations Overview. <https://learn.microsoft.com/en-us/ef/core/managing-schemas/migrations/?tabs=dotnet-core-cli>
- [41] Amazon: What Is An API (Application Programming Interface)? <https://aws.amazon.com/what-is/api/>
- [42] Microsoft documentation: AuthorizeView component. <https://learn.microsoft.com/en-us/aspnet/core/blazor/security/?view=aspnetcore-7.0#authorizeview-component>
- [43] Microsoft documentation: [Authorize] attribute. <https://learn.microsoft.com/en-us/aspnet/core/blazor/security/?view=aspnetcore-7.0#authorize-attribute>
- [44] Syncfusion documentation: SfDialogProvider Class. <https://help.syncfusion.com/cr/blazor/Syncfusion.Blazor.Popups.SfDialogProvider.html>

- [45] Syncfusion documentation: SfDialogService Class. <https://help.syncfusion.com/cr/blazor/Syncfusion.Blazor.Popups.SfDialogService.html>
- [46] MySQL: Installing and Upgrading MySQL. <https://dev.mysql.com/doc/refman/8.0/en/installing.html>
- [47] Microsoft: Visual Studio. <https://visualstudio.microsoft.com/vs/>
- [48] ConnectionStrings: MySQL connection strings. <https://www.connectionstrings.com/mysql/>
- [49] Syncfusion: Generate Syncfusion Blazor License Key. <https://blazor.syncfusion.com/documentation/getting-started/license-key/how-to-generate>
- [50] Google support: Sign in with app passwords. <https://support.google.com/accounts/answer/185833?hl=en>
- [51] Microsoft Azure: Homepage. <https://azure.microsoft.com/en-us/homepage-b/>
- [52] Microsoft documentation: Host and deploy Blazor Server. <https://learn.microsoft.com/en-us/aspnet/core/blazor/host-and-deploy/server?view=aspnetcore-7.0#deployment>
- [53] Microsoft documentation: Deploy an ASP.NET Core and Azure SQL Database app to Azure App Service. <https://learn.microsoft.com/en-gb/azure/app-service/tutorial-dotnetcore-sqldb-app>

Dostupnosť online zdrojov bola overená dňa 20.7.2023.

