

České vysoké učení technické v Praze  
Fakulta elektrotechnická



Aplikace Půjčovna aut

Uživatelská strana

BAKALÁŘSKÁ PRÁCE

*Vypracoval:* **Jiří Jindra**

*Vedoucí práce:* **doc. Ing. Ivan Jelínek, CSc.**

*Rok:* **2021**





# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Jindra** Jméno: **Jiří** Osobní číslo: **474688**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávací katedra/ústav: **Katedra počítačů**  
Studijní program: **Softwarové inženýrství a technologie**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Aplikace "Půjčovna aut" - uživatelská strana**

Název bakalářské práce anglicky:

**'Car rental' application - frontend-side**

Pokyny pro vypracování:

Cílem bakalářské práce je navrhnout klientskou stranu aplikace, která zajišťuje zapůjčení aut. Proveďte průzkum požadavků pro uživatelskou stranu aplikace (s uvážením požadavků serverové strany) a navrhnete prostředí pro práci s aplikací

- Navrhněte bezpečnou a stabilní komunikaci uživatelské strany s platebním prostředím na serverové straně
- Proveďte průzkum a navrhněte možnost zobrazení zapůjčitelných aut na mapě
- Definujte uživatelské scénáře a funkce, které budou možné na uživatelské straně aplikace
- Navrženou klientskou stranu implementujte, navažte na serverovou stranu a odzkoušejte komunikaci mezi oběma stranami aplikace
- Zvolte způsob testů a proveďte vyhodnocení testů
- Navrhněte možný další vývoj aplikace

Seznam doporučené literatury:

- Marijn Haverbeke, Eloquent JavaScript, [eloquentjavascript.net/Eloquent\\_JavaScript.pdf](http://eloquentjavascript.net/Eloquent_JavaScript.pdf)
- O'Reilly, DOM Enlightenment, <http://domenlightenment.com/>
- RIP Tutorials, Learning React, <https://riptutorial.com/Download/react.pdf>
- Bear Bibeault, JQuery in action, [docs.niwa.co.nz/library/public/1933988355update.pdf](https://docs.niwa.co.nz/library/public/1933988355update.pdf)

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**doc. Ing. Ivan Jelínek, CSc., kabinet výuky informatiky FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **11.02.2021**

Termín odevzdání bakalářské práce: **21.05.2021**

Platnost zadání bakalářské práce: **30.09.2022**

\_\_\_\_\_  
doc. Ing. Ivan Jelínek, CSc.  
podpis vedoucí(ho) práce

\_\_\_\_\_  
podpis vedoucí(ho) ústavu/katedry

\_\_\_\_\_  
prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta



### Čestné prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem k tomu pouze zdroje uvedené na konci práce, a to v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Nemám závažný důvod proti užívání tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským.

V Praze dne: .....

.....

Jiří Jindra

### Poděkování

Děkuji Ing. Ivanu Jelínkovi, CSc. za vedení mé bakalářské práce, zároveň tak za poskytnuté konzultace a podnětné návrhy, které práci obohatily. Děkuji též mé rodině za psychickou podporu.

.....

Jiří Jindra

*Název práce:* Aplikace “Půjčovna aut” - uživatelská strana  
*Autor:* Jiří Jindra  
*Obor:* Softwarové inženýrství a technologie  
*Druh práce:* Bakalářská práce  
*Vedoucí práce:* doc. Ing. Ivan Jelínek, CSc.  
Katedra počítačů, Fakulta elektrotechnická, České vysoké učení technické v Praze  
*Abstrakt:* Bakalářská práce představuje implementaci klientské části pro aplikaci půjčování automobilů. V první části se seznámíme s rešerší podobných služeb na trhu a shrneme si výhody naší implementace, v čem bude před konkurencí. Dále si přiblížíme požadavky na aplikaci, samotnou architekturu, uživatelské případy užití, a také procesní diagramy klientské části. V neposlední řadě práce shrnuje a přibližuje technologické aspekty aplikace, jako rámcový framework ReactJS, spuštění služby skrze NodeJS, či podpůrný nástroj pro vzhled prvků Bootstrap. Nakonec práce nastíní komunikaci se serverem, vykreslení vozů na mapě a testování aplikace s vyhodnocením.  
*Klíčová slova:* půjčovna, auto, mapa, aplikace, knihovna, uživatel, správce

*Title:* 'Car rental' application - frontend-side  
*Author:* Jiri Jindra  
*Abstract:* The bachelor thesis presents the implementation of the client part for the application of car rental. In the first part, we will get acquainted with the search for similar services on the market and summarize the advantages of our implementation, what will be ahead of the competition. We will also approach the requirements for the application, the architecture itself, user use cases, as well as process diagrams of the client part. Last but not least, the work summarizes and approaches the technological aspects of the application, such as the ReactJS framework, launching the service through NodeJS, or a support tool for the appearance of Bootstrap elements. Finally, the work outlines communication with the server, showing cars on a map and testing the application with evaluation.  
*Key words:* rental, car, map, application, library, user, administrator

# Obsah

Úvod.....	2
Kapitola 1 Existující řešení.....	3
1.1 Rozbor existujících řešení.....	3
1.2 Přínosy této aplikace.....	6
Kapitola 2 Návrh aplikace.....	7
2.1 Analýza požadavků.....	7
2.2 Funkční a nefunkční požadavky.....	8
2.3 Struktura aplikace.....	14
1.4 Případy užití.....	16
1.5 Procesní diagramy průchodů.....	18
Kapitola 3 Technologické prostředky.....	21
3.1 Rámcová knihovna.....	21
3.2 Prostředí pro spouštění aplikace.....	22
3.3 Doplnující knihovny.....	23
3.4 Mapová komponenta.....	25
3.5 Návaznost na serverovou část.....	27
Kapitola 4 Testování aplikace.....	30
4.1 Testy uživatelského rozhraní.....	30
4.2 Testy uživatelských vstupů.....	33
Kapitola 5 Možná rozšíření.....	34
5.1 Komunikace s podporou.....	34
5.2 Možnosti obnovení účtu.....	35
5.3 Schvalování účtu zaměstnancem.....	35
5.4 Monitoring pohybu automobilů.....	36
Závěr.....	37
Citovaná literatura.....	38
Seznam zkratk.....	40
Příloha A Postup pro spuštění aplikace.....	42
Příloha B Testy uživatelského rozhraní.....	44
Příloha C Testy vstupů aplikace.....	49



# Úvod

Cílem bakalářské práce je navrhnout aplikaci, která by zajišťovala provoz v půjčovně aut typu *carsharing*. Aplikace byla koncipována tak, že klientská strana (která je obsahem této práce) úzce spolupracuje se serverovou stranou (tato část byla předmětem jiné bakalářské práce), na který posílá nová data a zároveň z něj data získává pro vypsání na obrazovku uživatele a pro interakci s nimi.

Součástí je i návrh na rozšíření v případě navazujících činností na tuto práci. Výsledkem bakalářské práce bude mimo implementaci základních modulů řešení i návrh pro konečnou implementaci a testování.

V první kapitole *Existující řešení* si rozebereme již existující řešení, které se zabývají stejnou, či velmi podobnou oblastí, jako tato bakalářská práce, Poté vyzdvihneme, v čem bude navrhované řešení lepší než již fungující.

Definování funkčních a nefunkčních požadavků provedeme ve druhé kapitole *Návrh aplikace*. Tyto požadavky si také předvedeme pomocí grafického zpracování pomocí případů užití a diagramů průchodů. V této kapitole se také blíže seznámíme se strukturou aplikace v rovině adresářů.

Třetí kapitola, *Technologické prostředky*, se zaměřuje především na popsání fungování dané práce a jejích součástí. Sepsáno je rovněž zmapování a seznámení se s možnostmi zobrazení vozidel na dedikovaném mapovém podkladu a návrh komunikace frontend strany s platebním prostředím a zachování bezpečnosti plateb. Samozřejmě nechybí seznámení s podpůrnými knihovny použitých při vývoji a také popis hlavní rámcové knihovny, která představuje primární část aplikace.

Čtvrtá kapitola *Testování aplikace* shrnuje testy provedené nad finální verzí aplikace bakalářské práce. Vysvětluje také obsažené metodiky použité při testování. Konec této kapitoly je věnovaný shrnutí a zhodnocení proběhnuvších testů.

*Možná rozšíření* představují pátou kapitolu, kde se práce zabývá možnými rozšířeními aplikace do budoucna. To představuje například monitorování a zaznamenávání pohybu aut a pokročilá práce s daty o poloze.

Další součást rozšířeného návrhu se uvažuje ruční schvalování uživatelů skrze napojení na přidružené služby, jež podpoří bezpečnost služby a komunity jako takové. S tím souvisí podpora komunity, a to převážně pomocí online diskuse se správci, či implementace emailové knihovny na straně backendu pro možnost odesílání emailů určených pro uživatele této aplikace.

V přílohách se nachází detailnější návod, jak aplikaci spustit a jak s ní zacházet, a také podrobný soupis proběhlých testů.

# Kapitola 1

## Existující řešení

### 1.1 Rozbor existujících řešení

#### 1.1.1 Půjčovna TagGart [17]

Web půjčovny přímo na hlavní stránce shrnuje veškeré hlavní výhody dané platformy, a to v přehledné a srozumitelné formě, která je uživatelsky příjemná a univerzálně použitelná i pro jiné údaje této konkrétní služby.

Firma si dala na vzhledu a přehlednosti záležet. Web působí převážně svěže a moderním dojmem. Stránka firmy se ale nedrží konzistence co se týká barevného schématu, a zákazník si tudíž nemůže s firmou spojit určitou barvu. Tento nedostatek je z hlediska marketingu jedním z klíčových prohřešků.

Uživatel se nemůže registrovat a přehledně zkontrolovat proběhnuvší zápůjčky, což funkčnost této služby velmi degraduje a zhoršuje uživatelský zážitek. Na každé stránce je zřetelně uvedeno telefonní číslo do půjčovny, což usnadní uživatelům hledání kontaktu v případě problémů. Tento fakt ale působí zároveň i částečně redundantně, což je ale na subjektivním posouzení každého uživatele této služby.

Na stránce je výrazný prvek pro spočítání ceny pronájmu na daný počet dní bez nutnosti registrace. Po zadání data se uživatel dostane na přehled všech dostupných aut v daném rozmezí, které u každého vozu přehledně shrnuje jeho vlastnosti, přímé foto i přehledně uvedenou cenu s omezením kilometrů i bez omezení.

U každého vozu v přehledu je též výrazné tlačítko pro zapůjčení, což uživateli usnadňuje vybrání vozu a je jednodušší ho přimět k vytvoření výpůjčky daného vozidla.

Detail vozu je mírně zmatečný, což zapříčinila nekonzistence velikosti písma a přehuštnosti informací na stránce, a uživatel se může cítit zmateně. Chybí informace, kde se vůz právě nachází, či mapa všech vozů. Toto mírně zachraňuje možnost vybrání, kam vůz chce přistavět.

## 1.1.2 Nejlevnější-autopůjčovna.cz [18]

Vzhled služby odpovídá standardům a bude zákazníkovi příjemný při hledání správného vozu. Stránka je též plně responzivní, což je užitečné pro mobilní uživatele a přináší konzistentní zkušenost s používáním služby na rozličných zařízeních

Služba se drží jasné barevné palety, což ulehčuje identifikaci provozovatele a služby jako takové a definuje jednodušší marketingové oslovení potenciálních zákazníků, jelikož si danou barvu klientela této služby vždy spojí s touto společností a ulehčí rozlišování konkurence.

Na úvodní obrazovce zákazník ihned vidí nabídku vozů provozovatele a jasnou základní cenu pronájmu. Uživatelé se ale do služby nemohou registrovat, a tak zpětně zkontrolovat výpůjčky.

Nemožnost registrace snižuje uživatelský komfort zákazníků a také znemožňuje zpětně kontrolovat minulé rezervace. Toto omezuje používání služby například pro firemní účely.

Ve vrchní části jsou též shrnuty nejdůležitější důvody, proč zvolit právě tuto službu poskytující automobily k výpůjčce, což ujišťuje zákazníka v dobrém výběru služby.

Po rozkliknutí je přehledně umístěn box s cenami pro dlouhodobější pronájem vozidla. Uživatel též ihned vidí i další podrobnosti vozu včetně fotky konkrétního kusu automobilu vozového parku služby. Uživatel by ocenil filtrování vozů, aby nemusel číst kompletní nabídku vozů služby. Úplně chybí filtrování dle polohy uživatele, což by zvýšilo použitelnost.

Filtrování by stačilo na základní parametry, například dle ceny, výkonu, typu převodovky, typu pohonu a v poslední řadě dle značky daného vozu.

Zákazník by si vyfiltroval svoji oblíbenou konfiguraci a tato skutečnost by ho motivovala rezervaci uskutečnit.

### 1.1.3 Travel Car [19]

Vzhled a rozložení již neodpovídá moderním standardům a uživatel se může cítit nekomfortně. Není možná registrace a přihlášení, tudíž zaniká lehká kontrola minulých zápůjček

Na hlavní stránce společnost velice podrobně popisuje historii a důvody proč zvolit tuto službu. Nicméně se jedná o celkem nepřehledný a hlavně dlouhý text.

Určitě není nutné přímo na hlavní stránce, která představuje velmi důležitou destinaci služby popisovat historii společnosti. Doporučil bych tedy tento text přesunout spíše na samostatnou stránku, kterou si zájemci jistě rádi přečtou.

U výpisu chybí filtrování jednotlivých vozů, všechny vozy jsou na jedné stránce po skupinách vozů k jednotlivým využitím. Nicméně uživatel má znemožněno si zobrazit jen ty vozy, které ho skutečně zajímají

Na společné stránce také bohužel není nástin toho, kolik by dané vozidlo stálo za určitou jednotku času, daná informace je až v detailech o voze po rozkliknutí.

Webu chybí živá mapa vozů, nicméně tato služba s touto funkcí nepočítá, jelikož se jedná o dlouhodobější pronájem.

Doporučil bych ale provozovatelům této služby do budoucna zauvažovat o přidání této funkcionality, která zvyšuje přehlednost dané služby.

Nevýhodou je, že služba umožňuje auto pouze rezervovat bez možnosti zaplacení, a tak zákazník musí fyzicky navštívit provozovatele.

U vozů také není na první pohled graficky zřejmé, zda jsou volné k rezervaci, či nikoliv, což je pro klienta klíčové a toto představuje velký prohřešek vůči klientele této služby.

### 1.1.4 HoppyGo [20]

Služba navržena dle aktuálních trendů tak, aby byla co nejbliž zákazníkovi. Jasně dané prvky, které identifikují službu a zákazník si je s ní spojí. Web se drží jednoho stylu, a proto působí velmi konzistentně

Na hlavní stránce je dominantním prvek box pro zákazníka, od/do kdy si chce vůz půjčit, včetně udání polohy pro výběr, což usnadňuje uživateli výběr.

Pod tímto boxem pro výběr data výpůjčky provozovatel popisuje postup, jak vůz půjčit, což zvyšuje komfort. Při výpisu všech vozů můžeme využít pokročilého filtrování. Výpis vozů působí a přehledně, vždy je uvedena základní cena za 1 den pronájmu, foto a hodnocení daného auta.

Jednotlivá vozidla mají velmi exaktní fotografie přímo těch vozů, které jsou nabízeny. V daném výpisu vozů lehce chybí krátké seznámení s vozem, aby uživatel nemusel otevírat jednotlivé specifikace.

Úplně chybí informace, zda je v zadaném rozmezí dat pro eventuální výpůjčku vozidlo k dispozici.

V detailním popisu vozu jsou vypsány vlastnosti vozu, tak i poloha kde se vozidlo nachází. Web umožňuje jak rezervaci, tak i přímou platbu, což usnadňuje další kroky při výpůjčce.

## 1.2 Přínosy této aplikace

Design aplikace byl vytvořen podle aktuálních trendů. Rozmístění jednotlivých prvků bylo navrženo konzistentní, předvídatelné a uživatelsky přívětivé

Součástí aplikace bude též mapa vozů k zapůjčení s jejich polohou, která zpříjemní uživateli výběr správného vozu ve svém okolí.

Proces rezervace bude pro uživatele popsán a navržen tak, aby byl pro zákazníka co nejrychlejší a neobsahoval zbytečné kroky, které nesouvisí s výpůjčkou vozidla.

Aplikace je plně responzivní, takže ani uživatelé s mobilními telefony nebudou svým zařízením s menším displejem znevýhodněni a vše vyřídí stejně rychle a kvalitně, jako na osobním počítači.

Ceny, které zákazník očekává a které umožní zaujmout mezi relativně silnou konkurencí na trhu s podobnými řešeními.

# Kapitola 2

## Návrh aplikace

### 2.1 Analýza požadavků

Po důsledné analýze a rozboru zadání bakalářské práce pro uživatelskou stranu půjčovny s automobily byly stanoveny prioritní funkční a nefunkční požadavky pro zaměstnance a zákazníky, které by měla výsledná aplikace splňovat.

V drtivé většině se jedná o požadavky, které jsou podstatné pro fungování této části aplikace. Existuje spousta dalších požadavků a funkcionalit, které bych chtěl v případě rozšiřování tohoto projektu implementovat. Jejich soupis se nalézá v páté kapitole páté, *Možná rozšíření*

Jednotlivé funkční a nefunkční požadavky této aplikace jsou deklarovány kromě popisu také s důležitostí požadavku na fungování aplikace jako takové.

Z části tyto požadavky vycházejí z rozboru existujících řešení z podkapitoly 1.1 *Rozbor existujících řešení*, kdy se do požadavků zahrnuly dobře fungující celky, nebo se požadavky snaží vylepšit časté prohřešky těchto a podobných aplikací.

V případě rozšiřování funkčností této práce se pravděpodobně tento seznam funkčních a nefunkčních požadavků pro zaměstnance, zákazníky a aplikaci rozroste o další položky.

Funkční a nefunkční požadavky jsou sepsány v podkapitole 2.2 *Funkční a nefunkční požadavky* a v jednotlivých soupisech lze nalézt požadavky pro zaměstnance, zákazníky.

## 2.2 Funkční a nefunkční požadavky

### 2.2.1 Funkční požadavky pro správce

#### 1. Přihlášení

- Pokud má správce přiřazený účet, může se přihlásit přes daný formulář. Po přihlášení se zobrazí konkrétní funkce pro správce aplikace
- *Dopad na funkčnost:* Vysoká

#### 2. Přidání vozu

- Správce může přes speciální formulář přidávat nová auta do aplikace pro rezervace včetně jejich polohy. V rozšířené verzi se poloha bude načítat automaticky podle polohy konkrétního automobilu. Pro přidání vozu musí správce zadat všechny povinné parametry
- *Dopad na funkčnost:* Vysoká

#### 3. Úprava stávajícího vozu

- Po přidání vozu má zaměstnanec možnost atributy vozu v aplikaci upravit
- *Dopad na funkčnost:* Střední

#### 4. Výpis všech klientů

- Zobrazení seznamu všech klientů v tabulkové a grafické formě. Ve výpisu se zobrazují údaje daných klientů (kromě čísla platební karty), včetně informace, zda je onen klient zablokovaný, či nikoliv.
- *Dopad na funkčnost:* Střední

#### 5. Výpis všech rezervací

- Výpis všech rezervací klientů včetně veškerých údajů (datum a čas založení, zahájení a ukončení rezervace, uživatelské jméno konkrétního klienta, částku k placení, stav platby, a vůz související s rezervací)
- *Dopad na funkčnost:* Vysoký

#### 6. Výpis konkrétního klienta a jeho rezervací

- Zobrazení profilu konkrétního klienta včetně historie rezervací (datum a čas založení, zahájení a ukončení rezervace, konkrétního uživatele, částku k placení, stav platby, a vůz související s rezervací) a jeho údajů
- *Dopad na funkčnost:* Vysoký

## **7. Zablokování uživatele**

- V případě uživatele, který způsobuje službě škodu, či se chová nevhodně, má správce možnost danému klientovi zablokovat přístup do aplikace
- *Dopad na funkčnost:* Vysoký

## **8. Úprava stávajícího hesla**

- Správce disponuje možností si ve speciálním formuláři možnost změnit své stávající přístupové heslo služby
- *Dopad na funkčnost:* Střední

## **9. Výpis vozů služby**

- Zaměstnanec si v grafické formě může vypsat veškerá auta, která jsou aktuálně v aplikaci, a to jak půjčené, tak i nevypučené. Další možností je výpis jen těch vozů, které jsou dostupné uživatelům k založení nové výpůjčky. Zároveň s tím také aktivní vozy spolu s neaktivními automobily v aplikaci
- *Dopad na funkčnost:* Vysoký

## **10. Zobrazení všech aut aplikace na mapovém podkladu**

- Správce má možnost zobrazit veškerý vozový park na mapovém podkladu, což se liší od podobné klientské funkce hlavně tím, že klient má možnost zobrazit jen aktivní vozy připravené na výpůjčku.
- *Dopad na funkčnost:* Střední

## **11. Výpis vozů podle zvolených kritérií**

- Stejně jako klient, má správce možnost zobrazit si jednotlivá auta vozového parku služby podle zadaných kritérií. Výpis se zobrazí jako grafické znázornění karet nebo na mapovém podkladu.
- *Dopad na funkčnost:* Střední

## **12. Deaktivace vybraného vozu**

- V případě potřeby může správce daný vůz označit jako neaktivní a tím zamezí budoucím rezervacím na tento konkrétní vůz. Příčin, proč by toto správce požadoval je více, nejdůležitější je ale akutní závada na daném modelu, či nutný servisní zásah. Po deaktivaci tento automobil zmizí z výpisu dostupných automobilů, avšak ve výpisu všech automobilů zůstává i nadále.
- *Dopad na funkčnost:* Vysoký



## 2.2.2 Funkční požadavky pro klienta

### 1. Registrace

- Registrační formulář vždy slouží k vytvoření nového klienta. Při registraci musí uživatel zadat jméno, příjmení, datum narození, číslo řidičského průkazu, uživatelské jméno, emailovou adresu, heslo. Všechny údaje musí být vyplněny a zároveň emailová adresa, uživatelské jméno a číslo řidičského průkazu jsou unikátní pro každého uživatele
- *Dopad na funkčnost:* Vysoký

### 2. Přihlášení

- Pokud má uživatel vytvořený účet, může se přihlásit přes daný formulář. Po přihlášení se zobrazí konkrétní funkce pro klienta aplikace
- *Dopad na funkčnost:* Vysoká

### 3. Výpis dostupných vozů

- Klient si v grafické formě může vypsat aktivní (půjčitelná) auta, která jsou aktuálně v aplikaci
- *Dopad na funkčnost:* Vysoká

### 4. Výpis dostupných vozů na mapovém podkladu

- Možnost pro klienta zobrazit aktivní vozový park na mapovém podkladu, což usnadňuje rozhodnutí o výběru konkrétního vozu dle preferencí umístění daného vozidla
- *Dopad na funkčnost:* Střední

### 5. Výpis dostupných vozů podle zvolených kritérií

- Funkcionalita pro výpis jednotlivých vozů služby podle zadaných kritérií. Výpis se zobrazí jako grafické znázornění karet nebo na mapovém podkladu.
- *Dopad na funkčnost:* Střední

### 6. Vytvoření rezervace vozu

- Po vybrání vhodného auta v grafickém zobrazení, nebo na mapě, se po kliku na tlačítko Rezervovat vytvoří čekající rezervace spojená s účet klienta
- *Dopad na funkčnost:* Vysoký

## 7. Zrušení vytvořené rezervace

- Klient má zdarma 30 minut na zahájení rezervace. V tomto časovém okně je možné zrušení rezervace bez poplatku.
- **Dopad na funkčnost:** Střední

## 8. Zahájení rezervace

- Zahájit rezervaci může klient jen tehdy, pokud přiřadil platnou platební kartu, a to dvěma způsoby. Prvním z nich je překročení časového okna 30 minut zdarma. Po uplynutí této doby se rezervace automaticky zahájí a zároveň se začne počítat doba rezervace pro výpočet závěrečné ceny. Druhým způsobem je zahájení rezervace samotným klientem v bezplatném časovém okně 30 minut.
- **Dopad na funkčnost:** Vysoký

## 9. Ukončení rezervace

- Ukončit rezervaci lze jen tu, která byla klientem již zahájena. Po ukončení se spočítá částka dlužná za rezervaci a aktivuje se možnost rezervaci zaplatit
- **Dopad na funkčnost:** Vysoký

## 10. Zaplacení rezervace

- Zaplatit lze jen řádně ukončenou rezervaci klientem s přiřazenou platební kartou. Po stisku tlačítka pro platbu v rámci přehledu historie rezervací umístěném v účtu klienta dojde k zaplacení rezervace. Úspěšné zaplacení je závislé na dostatečném zůstatku na uložené platební kartě klienta
- **Dopad na funkčnost:** Vysoký

## 11. Přidání platební karty

- Po registraci má uživatel možnost prohlédnout si vozy připravené k půjčení včetně jejich specifikací včetně polohy vozů na mapovém podkladu. Bez přidané karty ale nemohou založit rezervaci. Tudíž přidat kartu pro správné využívání funkcí služby je nutností.
- **Dopad na funkčnost:** Vysoká

## 12. Odstranění platební karty

- Klientovi je nabídnuta funkce odstranění platební karty z aplikace. Pokud ale tuto funkci využije, nebude moci zakládat nové rezervace a zároveň před odstraněním karty je nutné mít všechny rezervace řádně zaplacené
- **Dopad na funkčnost:** Střední

### 13. Úprava stávajícího hesla

- Klient disponuje možností si ve speciálním formuláři možnost změnit své stávající přístupové heslo k účtu služby
- *Dopad na funkčnost:* Střední

### 14. Zobrazení historie rezervací

- V kartě profilu klienta se vypisuje historie rezervací, a to zvláště proplacené, aktivní a nezaplacené rezervace. U každé aktivní či neproplacené rezervace může klient měnit jejich aktuální stav, nebo rezervaci zaplatit.
- *Dopad na funkčnost:* Střední

### 15. Odhlášení

- Důležitá funkcionalita, která nepracuje jen s proměnnými uživatelské strany, ale i s funkcemi strany serverové. Odhlášení u klienta je důležitá funkcionalita, jeho výpadek může ohrozit vytváření rezervací od dalších klientů, nebo může končit až zneužitím klientova účtu. Jedná se o velmi důležitou základní funkcionalitu této aplikace.
- *Dopad na funkčnost:* Vysoká

### 16. Deaktivace uživatelského účtu

- V případě, že o službu klient ztratí zájem a nehodlá účet do budoucna používat, nabízí naše služba možnost deaktivace účtu, kdy se účet daného klienta nesmaže, ale přepne do režimu spánku.
- Deaktivace je možné pouze tehdy, pokud uživatel nemá aktivní rezervaci, a zároveň není potřeba u žádné provést platbu.
- Uspání účtu je přípravou na rozšíření bakalářské práce o podporu ze strany této služby, která bude posuzovat aktivace uspaných účtu. S tím také souvisí schvalování účtů správců skrze fotografie osobních dokladů budoucích klientů.
- *Dopad na funkčnost:* Nízký

## 2.2.3 Nefunkční požadavky

### 1. Webové rozhraní optimalizované pro Chromium

- Rozhraní a jeho funkcionalita bude optimalizované primárně pro jádro internetových prohlížečů Chromium, na kterém staví primárně Chrome od společnosti Google, a Edge vydávaný firmou Microsoft. Na ostatních jádrech s největší pravděpodobností bude aplikace fungovat bezchybně, avšak tato skutečnost již zaručena v rámci bakalářské práce není.
- *Dopad na funkčnost:* Vysoká

### 2. Použití ReactJS jako rámcový framework

- Knihovna pro tvorbu dynamických webových stránek ReactJS poskytuje funkce, které usnadňují tvorbu webových stránek a představuje základní stavební kámen bakalářské práce
- *Dopad na funkčnost:* Vysoký

### 3. Uživatelsky přívětivé GUI

- Pro uživatelský komfort a přehlednost je nutné, aby rozhraní služby bylo uzpůsobeno požadavkům klientů a hlavně poskytovalo předvídatelný, konzistentní a funkční stránku této aplikace
- *Dopad na funkčnost:* Vysoký

### 4. Bezpečnost dat

- Vstup dat od uživatelů do aplikace a jejich odeslání na server musí proběhnout bezpečně a v co nejkratší době. Také musí být zachována bezpečnost platebního prostředí pro klienty.
- *Dopad na funkčnost:* Vysoký

### 5. Integrita dat

- Data, které správce, nebo klient vloží do databáze je nutné uložit jako celek a nesmí docházet k situacím takových, že část dat bude ztracena
- *Dopad na funkčnost:* Vysoký

### 6. Responzivita

- Funkčnost zaměřena především na zákazníky s mobilními telefony a tablety, kterým responzivita usnadňuje práci s touto aplikací a upravuje uživatelské rozhraní pro menší velikost zobrazovacího zařízení
- *Dopad na funkčnost:* Střední

## 2.3 Struktura aplikace

Serverová část aplikace je součástí jiné bakalářské práce, nicméně obě práce úzce spolupracují při komunikaci a v praxi jsou na sobě funkčně závislé a vzájemně propojené. Vzájemné propojení je popsáno v podkapitole 3.5 *Návaznost na serverovou část*. Podle zadání bylo rozhodnuto, že aplikace se realizuje jako webová služba, jež bude uzpůsobena počítačovému i mobilnímu zobrazení.

V aplikační struktuře byly také k docílení lepší prezentace této aplikace klientům použity obrázky třetích stran. Jmenovitě na domovské obrazovce aplikace je použit prvek, který střídá fotografie páru u vozidla [27], muže při jízdě v autě [25] a radostného muže [26]. Dále byly z důvodu prezentace aplikace použity fotografie firmy Toyota [3], a to pod prvkem střídajícího fotografie výše uvedené na hlavní straně aplikace.

Jako rámcová knihovna byla zvolena technologie ReactJS [4], a to především kvůli předchozím zkušenostem s touto technologií, což jistě zrychlí a zkvalitní vývoj aplikace jako celku.

Prostředí pro překlad a spuštění aplikace, jaké je u knihovny ReactJS nejčastější a nejvhodnější, bylo vybráno Node.js [2]. Při vývoji aplikace bude využívána pouze základní podoba, kdy se ke spuštění užívá dedikovaný terminál Node.js a správce balíčků NPM [28].

Tento správce balíčků spravuje stávající závislosti aplikace, instaluje další potřebné závislosti a nachází konflikty v aplikaci, které by mohly v budoucnu negativně ovlivnit běh programu. Zároveň po výzvě stahuje aktualizace dodatečných i původních knihoven, přičemž je bohatě konfigurovatelný.

V aplikaci se samozřejmě nepoužívá pouze *Rámcová knihovna*, ale nachází se zde i další podpůrné knihovny, které byly podrobněji sepsány v Kapitola 3

Technologické prostředky. Ve zkratce jde převážně o *Bootstrap knihovnu*, jež usnadňuje tvorbu konzistentního uživatelského rozhraní. *Knihovna Axios* slouží pro asynchronní komunikaci, převážně pomocí http protokolů get, put, post, se serverovou stranou aplikace.

*Js-cookie* zpřijemňuje programátorským způsobem práci s ukládáním a načítáním cookies souborů na klientské straně aplikace.

V neposlední řadě také React-cookie-consent knihovna, jež přidává v posledních letech povinnou součást webových stránek a služeb, a to potřebný souhlas s ukládáním souborů cookies od klienta používajícího tuto službu.

Aplikace má přesně danou strukturální hierarchii adresářů a zdrojových souborů. Kořenovou složku tvoří toto uskupení adresářů:

- **Adresář public**
  - Složka se souborem *index.html*, který slouží jako hlavní rámec pro spuštění samotné aplikace a zároveň umožní načtení dalších přidružených knihoven v JavaScriptu.
  - V tomto adresáři se nachází také logo webové služby.
  - Dále obsahuje podadresář *assets*, kde se nachází veškeré obrázky použité ve webové službě. Je pravidlem a někdy i nutností, aby používané obrázky byly umístěny právě v tomto adresáři.
- **Adresář node\_modules**
  - Složka k uchování veškerých výchozích i podpůrných knihoven pro práci s technologií ReactJS.
  - Do této složky stahuje správce NPM nově přidané knihovny
  - V adresáři nesmí být žádný konflikt mezi knihovnami, jinak nedojde ke spuštění aplikace.
- **Adresář src**
  - Kořenový adresář pro JavaScript soubory, jež vytvářejí výslednou stránku na monitoru uživatele.
  - Jsou zde uloženy veškeré zdrojové soubory podstránek služby, včetně souboru *index.js*. Daný soubor slouží jako spojovací článek mezi souborem *App.js*, jež načítá jednotlivé komponenty, a hlavním souborem *index.html* ve složce *public*.
  - Adresář *src* je v případě této aplikace členěn na dílčí složky podle toho, zda se jedná o základní prvky stránky (patička, menu, Domů, Error, apod.), jestli dané stránky bude používat klient či správce, nebo zda se jedná o stylové soubory typu CSS.
  - Přímo v *src* se nachází soubor *App.js*, který načítá všechny zbylé soubory v tomto adresáři a jedná se tak mezi spojovací článek mezi komponentami a souborem *index.js*.

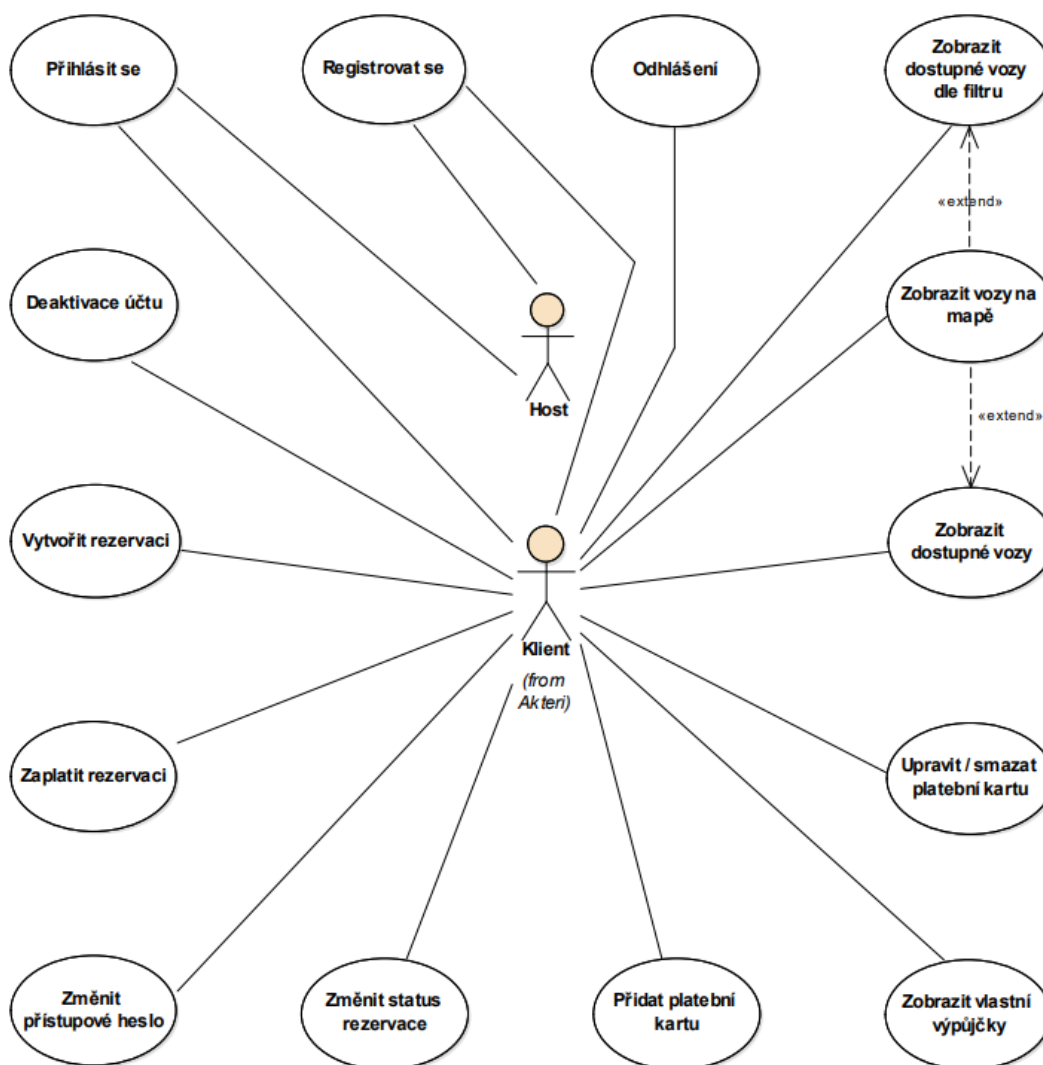
Důležitý soubor v kořenovém adresáři představuje také *package.json*, ve kterém je uložen soupis závislostí přidávaných do této aplikace a také seznam příkazů pro manipulaci s aplikací.

## 2.4 Případy užití

Základní úkony, jež je možné v aplikaci provést, byly kromě jejich sepsání do jednotlivých položek požadavků zakresleny také do diagramů případu užití. Určitá množina několika případů užití se opakuje u obou aktérů.

I přes to jsem se rozhodl je zakreslit do obou diagramů zvlášť, aby nedošlo ke zmatku v diagramech. Můžeme pozorovat, že správce má logicky více možností, jak ovlivňovat chod aplikace, avšak jen menší část úkonů je v průniku s úkony klienta.

### 2.4.1 Případy užití klienta



Obrázek 1 Schéma případů užití klienta

## 2.4.2 Případy užití správce

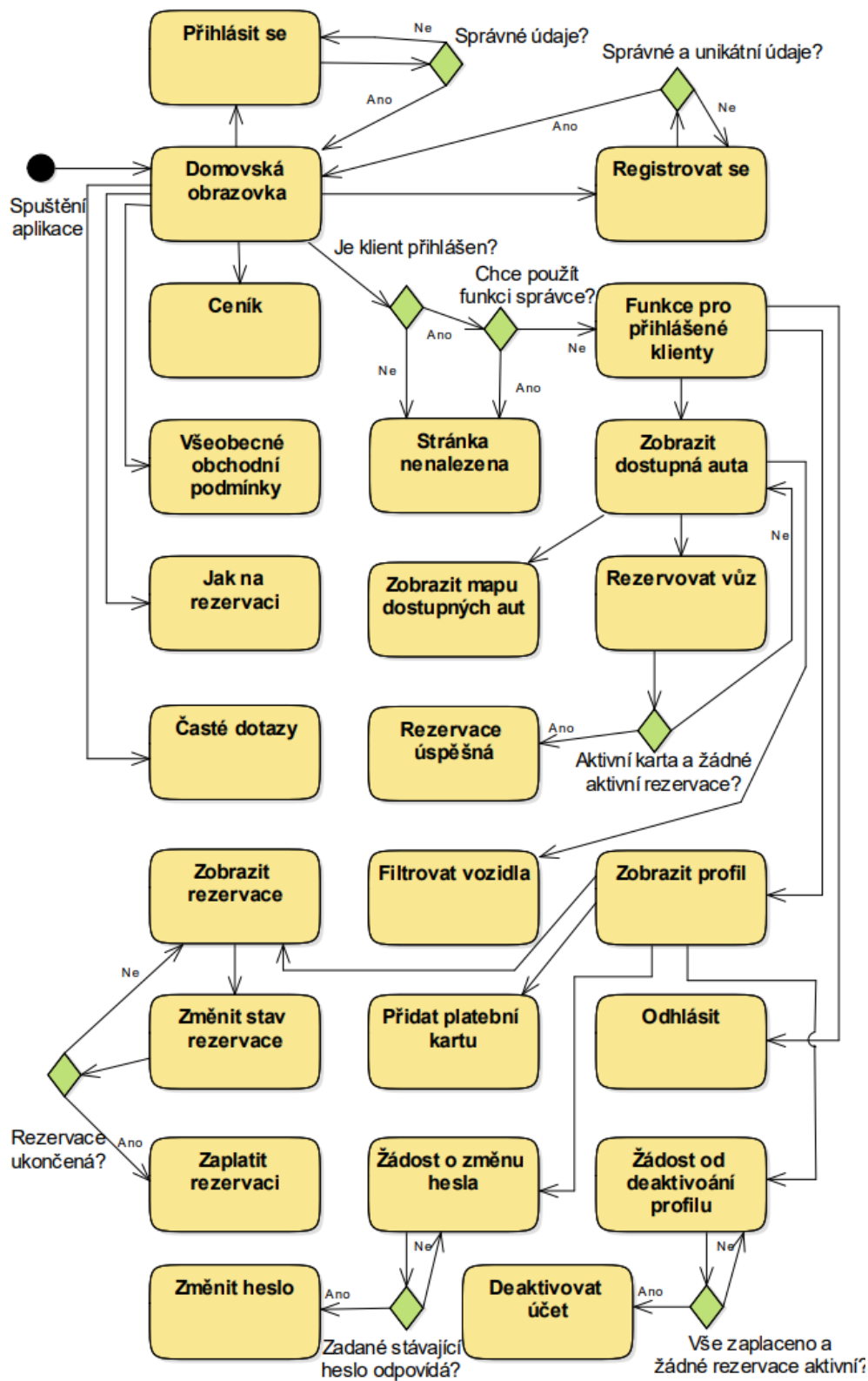


Obrázek 2 Schéma případů užití správce



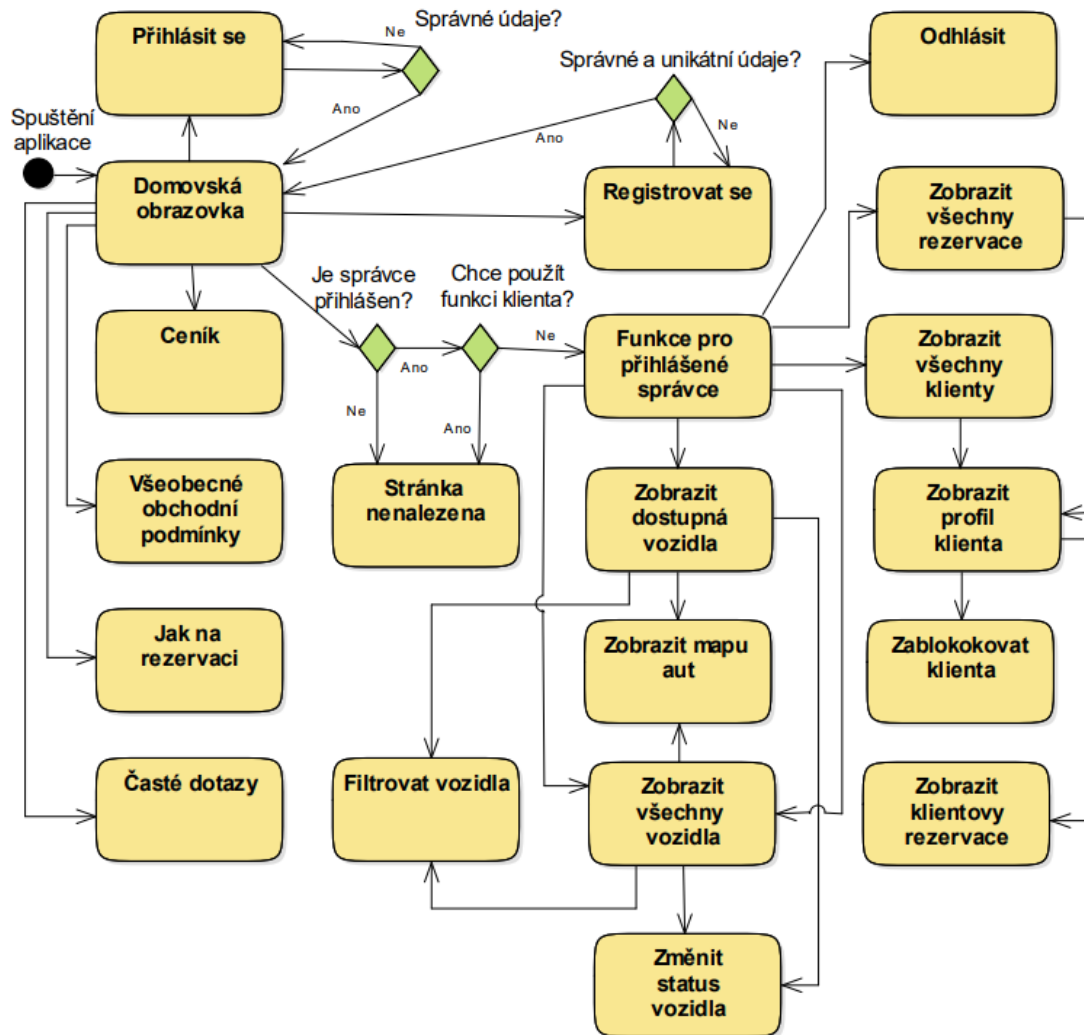
## 2.5 Procesní diagramy průchodů

### 2.5.1 Procesní průchody klienta

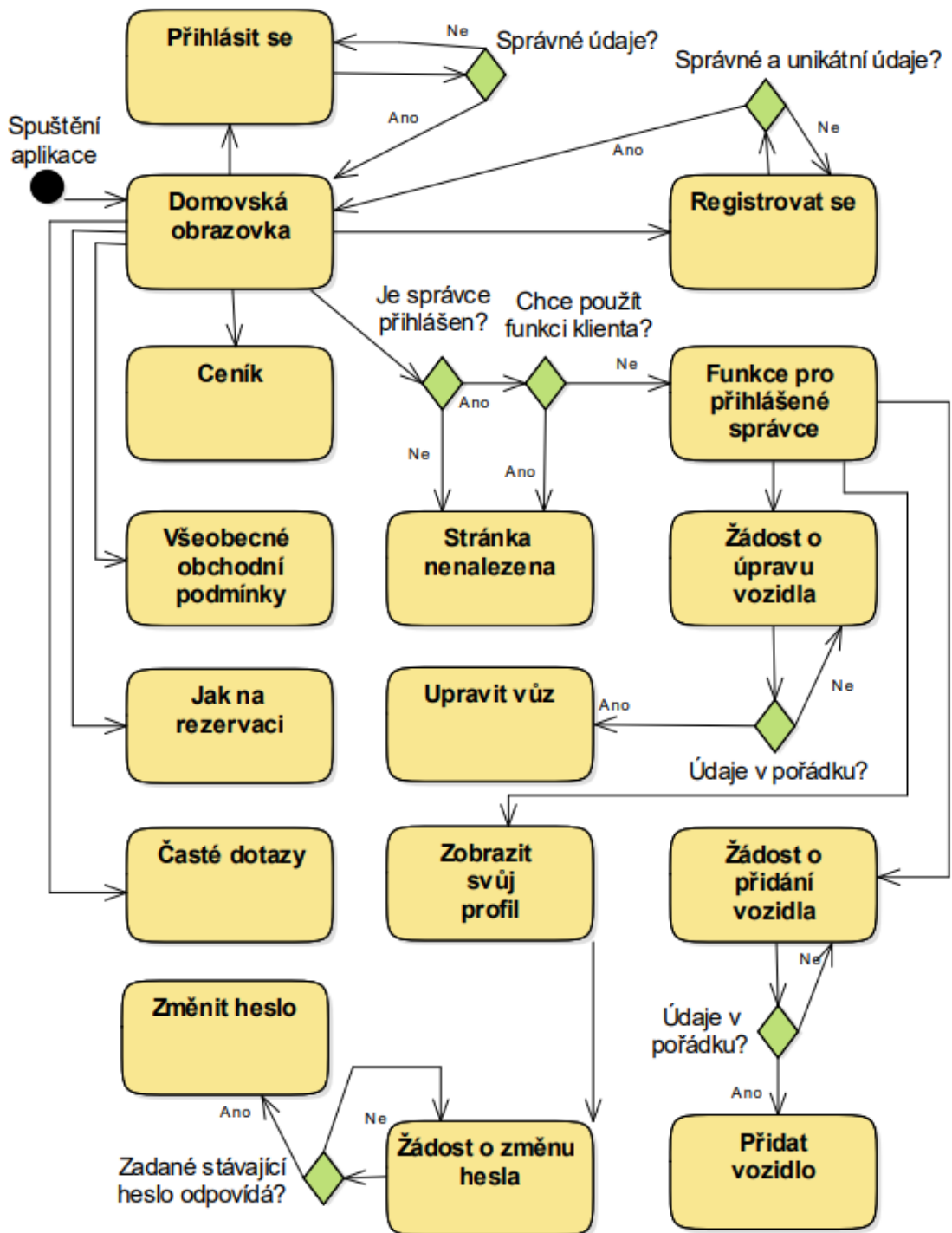


Obrázek 3 Diagram průchodů klienta

## 2.5.2 Procesní průchody správce



Obrázek 4 Diagram průchodů správce - 1. část



Obrázek 5 Diagram průchodů správce - 2. část

# Kapitola 3

## Technologické prostředky

### 3.1 Rámcová knihovna

Vzhledem k minulým zkušenostem s technologií pro tvorbu webových aplikací ReactJS [4] bylo rozhodnuto, že právě tato knihovna poslouží jako základní rámcový framework pro tvorbu tohoto bakalářské práce. Existují i další podobné technologie, zmiňme například Angular 2 [21], který je postavený na TypeScriptu vyvíjeném firmou Google.

Naproti tomu ReactJS představuje knihovnu vyvíjenou společností Facebook a staví převážně na JavaScriptu a přináší pokročilejší práci se strukturou DOM dané stránky a manipulaci s prvky DOM struktury. Manipulace s prvky DOM struktury nám umožňuje měnit zobrazení stránky bez nutnosti jejího znovunačtení, což přináší spoustu výhod pro uživatele, kteří danou službu používají. Další výhodou představuje univerzálnost ReactJS, jelikož umožňuje také tvorbu mobilních aplikací. V případě mobilních aplikací se ale jedná o odnož ReactJS, jmenovitě React Native. [16]

#### 3.1.1 ReactJS [4]

Open-source framework pro vytváření uživatelských rozhraní pro webové služby. Základním stavební kámen zde tvoří tzv. komponenty (components), což jsou různé znovupoužitelné HTML elementy se zapouzdřenou funkcionalitou, jejichž skládáním vzniká komplexní UI aplikace. Tyto komponenty pak mají své vlastnosti (props) a spravují svůj vnitřní stav (state). Tento deklarativní způsob práce s daty aplikace vede k více předvídatelnému chování i lehčímu ladění. [5]

Jednoduchá manipulace s **DOM** stromem stránky díky možnosti vytváření vlastních tříd a generování obsahu podle potřeby

Umožňuje vyvíjet aplikaci v **JSX** [4]

- Možnost psát opravdové HTML **přímo** do JavaScript kódu, ne jako pouhý řetězec

**JSX** nefunguje přímo v prohlížeči, nutné nejdřív přeložit pomocí Node.js

## 3.2 Prostředí pro spuštění aplikace

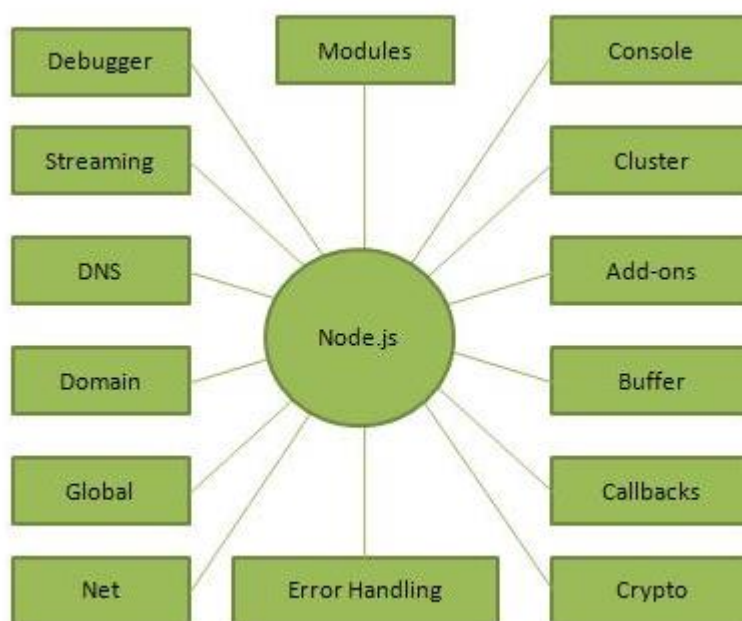
Stejně jako u rozhodnutí ohledně použití ReactJS jako hlavního prostředí pro vývoj aplikace, se i při výběru prostředí pro samotné zkompilování spuštění v průběhu vývoje zohledňovaly získané zkušenosti a proto bylo rozhodnuto používat technologii Node.js.

Ke spuštění postačuje terminál Node.js, kdy se následně v terminálu spustí správce balíčků NPM [28]. Tento správce balíčků spravuje závislosti aplikace, instaluje další potřebné závislosti a nachází konflikty v aplikaci, které by mohly v budoucnu negativně ovlivnit běh programu. Zároveň po výzvě stahuje aktualizace dodatečných i původních knihoven, přičemž je bohatě konfigurovatelný.

### 3.2.1 Node.js [2]

Node.js představuje open-source platformu, které je schopna spouštět kód napsaný v JavaScriptu pro snadnou tvorbu webových aplikací. Primárním účel jeho návrhu byla tvorba serverové části webových služeb, nicméně je možné ho použít i při vývoji klientských částí vyvíjené např. v ReactJS. [4]

Technologie je postavena na JavaScript enginu prohlížeče Chromium a obsahuje stejný základ, jako ve jmenovaném prohlížeči, tudíž zpracování kódů je velmi rychlé. Jádro celého Node.js tvoří tzv. smyčka událostí (event loop). Do ní vstupují všechny uživatelské požadavky jako události, které jsou poté přiděleny jednotlivým nezávislým vláknům. [6]



Obrázek 6 Schéma částí Node.js [11]

## 3.3 Doplnující knihovny

### 3.3.1 Knihovna uživatelského rozhraní

Ačkoliv ReactJS sám o sobě nabízí rozsáhlé možnosti tvorby uživatelského rozhraní, rozhodl jsem se při tvorbě aplikace využít knihovnu pro tvorbu uživatelských rozhraní (GUI) Bootstrap [22].

Rozhodování probíhalo mezi vícero knihovnami určených pro pokročilou tvorbu uživatelských rozhraní, jelikož dostupných knihoven je nepřehledné množství. Bylo těžké rozhodnout se jen pro jednu správnou variantu, a to přesně z důvodu, že každá knihovna nabízí své plusy a mínusy. Své rozhodnutí jsem tedy podpořil vlastními zkušenostmi a prostudováním možností knihoven v technických dokumentacích i na oficiálních repozitářích.

Do úvahy připadaly tyto tři knihovny:

#### 1. Material UI [12]

- Jedná se o populární knihovnu pro tvorbu UI webových stránek používající technologii React od společnosti Facebook. Instalace probíhá skrze NPM balíčkového manažera a jednotlivé komponenty se do samotného kódu importují jako další běžné součásti technologie React a jiných knihoven
- Design se nese v duchu Material designu, který pochází od společnosti Google a navozuje tak pocit z aplikací mobilního operačního systému Android

#### 2. Ant Design [13]

- Další možnost představuje knihovna, která je druhá nejpoužívanější pro tvorbu rozhraní při technologii ReactJS. Sada vysoce kvalitních komponent technologie React připravených ihned k použití.
- Komponenty jsou implementovány v jazyce TypeScript s předvídatelnými statickými typy. Balíček představuje celek s navrženými a přizpůsobitelnými komponentami a vývojovými nástroji. Mezinárodní podpora desítek jazyků.
- Jednotlivé komponenty nabízejí bohaté možnosti přizpůsobení konkrétním podmínkám aplikace a jsou navrženy tak, aby byly znovupoužitelné

### 3. React Bootstrap [22]

- Bootstrap představuje jednoduchou a volně stažitelnou sadu nástrojů pro tvorbu webu a webových aplikací.
- Obsahuje předem definované položky založené na CSS a HTML, které slouží pro definici prvků, jakými jsou například navigace, tlačítek, textových polí či písma, a další komponenty uživatelského rozhraní.
- Nabízí podporu responzivity a je kompatibilní s velmi širokou škálou prohlížečů, včetně Firefoxu i prohlížečů postavených na Chromiu.
- Bootstrap ztělesňuje modulární knihovnu a sestává v zásadě ze série LESS stylůpisů, které implementují různorodé komponenty toolkitu. Komponenty Bootstrapu jsou zahrnuty ve stylůpisu *bootstrap.less* [9]

Po přečtení a nastudování odpovídajících dokumentací jsem se rozhodl používat knihovnu Bootstrap. Bootstrap nenabízí tolik pokročilých možností jako Material UI či Ant Design. Nabízí ale vše potřebné, podle mého názoru, v jednodušší podobě a po vyzkoušení všech tří technologií jsem se rozhodl právě pro Bootstrap knihovnu.

#### 3.3.2 Js-cookie [23]

Oblíbená JavaScript knihovna pro práci s cookies. Nabízí rozhraní, které usnadňuje jejich ukládání a získávání na klientské straně aplikace. Benefitem je malá velikost knihovny a její snadná použitelnost v různorodých aplikacích.

Dále také skutečnost, že je poctivě otestovaná, tudíž by nemělo docházet k chybám při používání této knihovny.

Umožňuje ukládání cookies po předem danou dobu, nebo na dobu neurčitou. Údaj doby pro uložení si můžeme určit podle svých vlastních preferencí při vlastním ukládání.

Seznámení se základním použitím:

- Ukládání: `Cookies.set('name', 'value', { expires: 7 });`
- Načítání: `Cookies.get('name');`
- Mazání: `Cookies.remove('name');`

### 3.3.3 React-cookie-consent [24]

Z evropského nařízení o nabízení internetových služeb před pár lety přibyla všem webovým aplikacím nová povinnost. Všem uživatelům, jež navštíví konkrétní službu ukládající cookies soubory do počítače, je nutné sdělit informaci, že k ukládání dochází.

Pro tuto funkčnost by pravděpodobně postačovala knihovna Js-cookie z předchozího bodu, nicméně React-cookie-consent přináší výhodu připraveného unifikovaného vzhledu, jež implementuje odzkoušené chování žádané po této funkci. Knihovna mi tedy ulehčila implementaci této funkcionality a zrychlila tím vývoj důležitých prvků práce.

## 3.4 Mapová komponenta

Důležitou komponentou z hlediska klienta představuje mapová komponenta, jež zobrazuje dostupná auta na mapovém podkladu. Komponentu využije též správce webové služby, který uvidí přímo na mapě umístění všech vozů v databázi. Implementace mapového podkladu do webové služby pro mě představovala novou zkušenost s touto technologií, jelikož s mapami jsem dříve nikdy nepracoval.

Technologií, které umožňují zobrazení mapového podkladu existuje nepřeberné množství, nicméně dva nejvýraznější hráče na poli zobrazování map představují Google Maps API [14] a Leaflet [7].

Zde uvádím jen krátký výčet vlastností zmíněných dvou zástupců:

#### Google Maps API [14]

- Knihovna využívající službu Google Maps k vykreslení mapy na webovou stránku, lze ji ale částečně upravit pro vlastní účely
- Dovoluje vykreslit jakoukoliv React komponentu přímo na mapu.
- Je plně izomorfní a může se vykreslovat též na serveru.

#### LeafletJS [7]

- Flexibilní a open-source mapová knihovna pro vykreslení mapové komponenty, jež nejčastěji využívá podklady ze služby OSM
- Umožňuje jednoduchou práci s mapovými podklady a nabízí možnost vykreslení vytyčených bodů na mapě podle toho, jak body zadáme.
- Jedná se o velmi oblíbenou mapovou službu velkých společností, jmenovitě například Pinterest, OMS či Foursquare
- Služba umožňuje zpracovávat data se souborů formátu GeoJSON, KML, CSV, TopoJSON či WMS



Poté, co byly porovnány jednotlivé varianty a jejich dokumentace, byla vybraná jako mapová komponenta právě **LeafletJS**. Převážně díky své jednoduchosti, flexibilitě a hlavně otevřenému kódu, kvůli kterému si může komponentu každý přizpůsobit podle svých vlastních preferencí a potřeb.

Typické použití knihovny LeafletJS v bakalářské práci:

```
<MapContainer center={this.state.coords.mapCenter} zoom={11} scrollWheelZoom={true}>
  <TileLayer
    attribution='&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors'
    url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"
  />
  {this.state.cars.map(car =>
    <Marker position={[car.latitude, car.longitude]} key={car.id} id={"marker"+car.id}...>
  )}
  <Polygon id={"borders"} positions={this.state.coords.bordersForCars} />
</MapContainer>
```

Obrázek 7 Příklad použití LeafletJs

*Obrázek 7 Příklad použití LeafletJs* představuje ukázkou typického použití knihovny LeafletJS [7] v bakalářské práci, kdy nejprve vytvoříme povinný mapový kontejner (neboli React komponentu), do nějž umístíme odkaz na službu, odkud se čerpají data pro vykreslení mapového podkladu na webovou stránku.

Dále vytváříme komponenty „špendlíků“ znázorňujících jednotlivé vozy načtené z databáze rozmístěné na mapě.

Knihovna nám též nabízí možnost naplnit značku informacemi podle našeho uvážení, takže po kliknutí na danou značku se uživateli služby zobrazí podrobnosti o daném voze. Posledním prvkem přidaným na mapovou komponentu, který můžeme v příkladu použití LeafletJS vidět, je komponenta Polygon vytvářející zónu na mapovém podkladu, kde smějí klienti zaparkovat vypůjčené vozidlo.

Do budoucna při případném rozšiřování projektu bude implementována funkce kontroly polohy vozidla. Když se pokusí klient uzavřít rezervaci jinde, než uvnitř vyznačené zóny, nebude možné rezervaci uzavřít a tím také nemožnost ji zaplatit.

Nicméně pro úplnost věci uvádím, že udávání polohy automobilů je prozatím řešeno takovým způsobem, že při vkládání vozů správcem služby nastaví správce počáteční polohu vozu a ta se vlivem výpůjček nemění.

Změnit ji může opět je správce služby a to pomocí formuláře pro úpravu vozidla. Jde o přípravu na reálný provoz automobilů, do kterých umístíme lokalizační zařízení. Ty budou nastaveny na automatické odesílání aktuální polohy vozu do odpovídajícího záznamu automobilu v databázi. Tímto bude zaručena aktuálnost polohy daného vozidla.

## 3.5 Návaznost na serverovou část

Uživatelská strana je přímo vázaná na stranu serverovou, ze které čerpá data, posílá jí dotazy na data a také sama data posílá. Proto bylo nutné zvolit vyhovující metodu komunikace se serverovou stranou, která zároveň poskytuje asynchronní komunikaci se serverem.

Pro účel komunikace je již v JavaScriptu zabudovaná předpřipravená funkcionalita typu AJAX [1] skrze http request [8] pro asynchronní výměnu dat s webovým serverem (typicky je užíván formát XML). S daty se pracuje pomocí formátu JSON.

Jedná se však o již mírně zastaralou strukturu, u které se hůře zjišťují nastalé chyby při komunikaci. Její použití je také méně intuitivní, než u knihovny Axios [15], která také načítá data na pozadí. Proto zahrnutí knihovny Axios do aplikace bylo téměř okamžité rozhodnutí.

Načítání dat na pozadí umožňuje změnit strukturu DOM konkrétní stránky bez nutnosti jejího opětovného načtení. [10]

Zpočátku bylo zvykání na přístup ze strany knihovny Axios k posílání a získávání dat mírně složitější, ale v součtu je práce s touto knihovnou o notný kus rychlejší a intuitivnější, než práce se starším konstruktem XMLHttpRequest.

### 3.5.1 Axios [15]

- Vytváření http asynchronních požadavků na server za účelem načtení nebo uložení dat.
- Rychlé a bezpečné provedení dotazu
- Možnost pokročilého zachytávání chyb pomocí bloku catch
- Sdělení typu http požadavku objektu typu Axios pomocí volaných metod na tomto objektu: *get*, *put*, *post*, a další.
- V podrobnostech dotazu nastavujeme typ žádaných/poslaných dat, přístupové řetězce, hlavičky http dotazu a posílané data na server
- Dotazu na server vrací JavaScript objekt s informacemi o http statusu odpovědi, samotnou odpověď, data vrácena ze serveru (v případě dotazu na data), případně výsledek pokusu o uložení dat (v případě poslání dat na server)

*Obrázek 8 Použití knihovny Axios v aplikaci* umístěný níže představuje typické použití knihovny axios v bakalářské práci je znázorněno níže na *Obrázek 8 Použití knihovny Axios* v

Pro potřeby znovupoužití daného požadavku na server vložíme tentýž požadavek do asynchronní metody.

Nutností je stanovení URL adresy, na kterou požadavek směřuje. Následně dochází k zavolání odpovídající metody na objektu *axios*, v případě příkladu níže, metody *post*.

Parametry se mírně liší od použité metody (*get*, *post*, *put*, a *další*). V tomto konkrétním do volání metody předáváme URL adresu.

Další parametr představuje poslaná data. Zde se můžeme podívat na převod atributu *client* objektu *state* komponenty *Registration*, která představuje nově registrujícího se uživateli do této služby.

Poslední předávaný atribut obsahuje konfiguraci daného Axios http požadavku na server.

Sdělujeme jím požadavku informaci, že chceme zahrnout roli a token daného uživatele. Podle těchto informací server pozná, zda uživatel posílající požadavek má dostatečná práva k využívání žádané metody na serveru.

Také sdělujeme, že do hlavičky požadavku chceme přidat informaci, že daná přenášená data jsou ve formátu JSON.

Vrácená odpověď se vloží do proměnné *response*, kde samotná data najdeme ve vlastnosti *response.data*. Samotný objekt *response* obsahuje kromě získaných dat široký záběr doplňujících informací ohledně proběhnuvšího požadavku.

Z nich je nejdůležitější *response.status*, kterým zjistíme úspěšnost dotazu na serveru.

Pokud se http požadavek potýkal s vážnými problémy, blok *then* se neprovede vůbec, a rovnou se přeskočí do bloku *catch*.

```
sendRequest = async () => {
  const url = "http://localhost:8080/car-rental-app-server/rest/clients/persist";
  await axios.post(url, JSON.stringify(this.state.client),
    config: {
      withCredentials: true,
      headers: {"Content-Type": "application/json;charset=UTF-8"}
    }).then((response : AxiosResponse<any> ) =>{
    if(response.status < 300){
      if(this.state.errorMessage !== "") this.setState( state: {errorMessage : ""});
      alert("Registrace proběhla úspěšně.");
      window.location.replace("/")
    }
  }).catch(() => this.setState( state: {errorMessage : "Registrace nebyla úspěšná"}));
}
```

Obrázek 8 Použití knihovny Axios v aplikaci

### 3.5.2 Platební prostředí

Pro přenos informací v rámci platebního prostředí jsem se rozhodl použít výše zmiňovanou knihovnu Axios, jelikož se jedná o spolehlivou a bezpečnou technologii možnou použít k široké škále potřeb.

Umožní mi také větší kontrolu nad http dotazy a jejich směrování na odpovídající funkce serverové části kontrolující funkce platebního prostředí

Klientská strana je nastavena tím způsobem, že je nutné do aplikace vkládat validní platební karty a serverová strana provádí reálné transakce nad vloženými platebními kartami.

Z důvodu otestování funkcí platebního prostředí však aplikace obsahuje jeden přednastavený účet, který je nastaven na testovací platební prostředí, a je tedy potřeba vložit pouze testovací platební kartu.

V tomto případě se částky nestrhávají a lze tak jednoduše otestovat proces platby za rezervace. Tato skutečnost je podrobněji popsána i s údaji k tomuto účtu v příloze *Přístup do služby*.

### 3.5.3 Autorizace a autentizace

I v tomto případě byla využita knihovna Axios. Pomocí této knihovny se klientská strana v případě přihlašování dotazuje serverové části, zda aplikace obsahuje zadané údaje. Pokud je odpověď serveru kladná, na klientské straně se uloží informace o přihlášeném uživateli do souborů cookies. Přesněji jde o jeho roli, uživatelské jméno a heslo.

Přihlášeného uživatele nelze změnit jednoduchým přepsáním souborů cookies, ani mu nikterak měnit aktuální roli v systému. Tato funkcionalita funguje na principu mechanismu na straně serveru, pomocí něžž tyto změny detekuje (viz. text této souběžné bakalářské práce).

# Kapitola 4

## Testování aplikace

Testování aplikace bakalářského práce představovalo důležitou součástí, která v rámci implementace odhalila chybnou kontrolu vstupů od uživatelů při filtrování vozidel, nekorektně odesílaná data nové platební kartě na serverovou část, nebo naopak jejich nepřesné vypisování. Dále také testy pomohly při vývoji uživatelského prostředí a jeho uspořádání. Testování uvedeno v příloze této bakalářské práci se provedlo v rámci konečné verze aplikace a rozdělil jsem jej celkově na dvě části, které v závěru vytvořily potřebný obraz o aplikaci.

První část testů, otestování uživatelského rozhraní pomocí participantů byl sepsán v kapitole příloh, konkrétně v příloze *Testy uživatelského rozhraní*. Test vstupních polí aplikace se nachází v příloze *Testy vstupů aplikace*.

### 4.1 Testy uživatelského rozhraní

První část testování zahrnuje test uživatelského prostředí aplikace na reálných uživateli v praxi. Skutečnost, že prostředí aplikace musí být intuitivní, flexibilní, konzistentní a předvídatelné představuje jednu z hlavních priorit této bakalářské práce.

Na začátku testování uživatelského prostředí jsem určil dva participanty a jednotlivé scénáře s připravenými průchody službou, které by měly samostatně a organizovaně provést uživatele v aplikaci.

Průchody participantů stanovenými scénáři včetně problémů jeho i aplikace, výtek i pochval participanta jsem následně sepsal do shrnujících tabulek *Tabulka 1 Nálezy a doporučení - Participant 1* a *Tabulka 2 Nálezy a doporučení - Participant 2*.

Nakonec byly testy vyhodnoceny a výsledky dílčího testu participanta zapsány jako shrnutí tohoto konkrétního dílčího testu.

## 4.1.1 Nálezky a doporučení

Tabulka 1 Nálezky a doporučení - Participant 1

<b>Problém</b>	<b>Doporučení</b>
Ne úplně jasný formát data narození při registraci klienta	Přidání prvku s textovou vysvětlivkou, nebo úprava pole pro vložení data narození pro větší intuitivnost
Přidání platební karty součást výpisu rezervací	Přesun možnosti pro přidání platební karty do záložky <i>Upravit Účet</i>
Nižší intuitivnost zobrazování vlastností vozů na mapovém podkladu	Přidat informace o možnosti zobrazit podrobnosti o voze kliknutím na polohu vozu v mapovém podkladu
Chybí tlačítko pro přidání vozu ve výpisu všech automobilů	Přidání tlačítka pro vložení nového vozu do přehledu všech vozidel
Nutnost nově přidávanému autu pozici ručně z mapového podkladu	Rozšířit aplikaci o automatickou detekci pozice vozidel

Tabulka 2 Nálezky a doporučení - Participant 2

<b>Problém</b>	<b>Doporučení</b>
Méně jasný správný formát hesla při registraci klienta	Přidání textového prvku s jasnějším vysvětlením povoleného formátu
Nižší intuitivnost zobrazování vlastností vozů na mapovém podkladu	Přidat informace o možnosti zobrazit podrobnosti o voze kliknutím na polohu vozu v mapovém podkladu
Nejasnost funkčnosti přepínače statusu vozidla	Přidání textového prvku s vysvětlením funkcionality daného prvku, nebo nahradit prvek jednodušší komponentou
Chybí tlačítko pro přidání vozu ve výpisu všech automobilů	Přidání tlačítka pro vložení nového vozu do přehledu všech vozidel
Nově přidávaný vůz až poslední v pořadí výpisu	Přidat možnost seřazení vozů dle kritérií, např. datum vložení do aplikace, apod.

## 4.1.2 Vyhodnocení testů uživatelského rozhraní

Testy uživatelského rozhraní proběhly na dvou reálných participantech. Každý z nich vyzkoušel klientské i správcovi funkce aplikace.

Většinu času předepsané úkoly plnili samostatně a intuitivně. Asistence koordinátora testování probíhala jen v menším množství případů.

Největší potíže dělali participantům ne úplně jasné povinné formáty vstupních hodnot, převážně hesla a data narození. Také se objevily pochyby při ovládání některých grafických prvků obsažených na stránce. V ojedinělých případech též docházelo k situacím, kdy participant nenašli ovládací prvky stránky tam, kde by je logicky automaticky očekávali.

Docházelo též ke vznášení výtek ohledně logickému uspořádání prvků, kdy participantům v některých případech připadalo logičtější dané prvky přesunout na jiné podstránky a tím tak dosáhnout odpovídajícího a předvídatelnějšího umístění tohoto prvku.

Výtky a problémy byly sepsány do *Tabulka 1 Nálezy a doporučení - Participant 1* a *Tabulka 2 Nálezy a doporučení - Participant 2* a navrženy odpovídající řešení, které dané záležitosti řeší ke spokojenosti účastníku a přináší tak větší logiku aplikace jako celku.

Při testování pronesli i slova pochvaly, kdy například pozitivně hodnotili celkovou přívětivost služby, plnou responzivitu a přitažlivost vzhledu jako celku.

Dle dosažených výsledků se domnívám, že testy uživatelského rozhraní dopadly úspěšně.

## 4.2 Testy uživatelských vstupů

Druhá část představuje test vstupních hodnot a omezujících podmínek polí, kdy se zkouší stabilita aplikace a její integritní vlastnosti. Aplikace nesmí přijmout a případně odeslat vstupní hodnoty na server, které by zapříčinily pád klientské, či serverové strany aplikace, nebo by ohrozily integritu uložených dat.

Služba dále nesmí přijmout hodnoty, které pro danou položku nedávají smysl vzhledem ke kontextu zadávaných informací.

V neposlední řadě je také nutné otestovat skutečnost omezení konkrétní množiny vkládaných hodnot, jelikož např. číslo řidičského průkazu klienta se musí sestávat z 8 znaků, z toho první znak představuje samohlásku, druhý jakékoliv písmeno a zbylých 6 znaků musí reprezentovat číselné hodnoty zadané od klienta. To však představuje jen jednu položku z mnoha, která je omezena na podmnožinu korektních vstupů.

Jak už bylo zmíněno, z těchto skutečností vyplývá nutnost hlídat jak logické hodnoty vkládaných informací, tak i předem nastavenou množinu povolených vstupních hodnot, jež jsou zadané klientem, tak i hodnoty pocházející od správce webové služby. Soupis proběhnuvších testů uživatelských vstupů je sepsán v příloze *Testy vstupů aplikace*.

### 4.2.1 Vyhodnocení testů uživatelských vstupů

Testy uživatelských vstupů byly provedeny na každé obrazovce, která nabízí možný textový vstup od uživatele. Při testu konkrétní aplikace se nejprve vypsaly uživatelské vstupy a k nim možné stavy, které mohou vstupy nabývat.

Poté se sestavila tabulka scénářů, kdy každý sloupec představuje jednotlivý scénář a poslední řádek výsledek téhož scénáře. Možné kombinace byly ve všech testech zredukovány pomocí metody MCC (Multiple Condition Coverage).

Následně byly veškeré scénáře možných stavů vyzkoušeny vyvolat v aplikaci.

Dle výsledků všech testů uživatelských vstupů se ověřilo, že aplikace splnila požadavek na omezení těchto vstupů daných funkčními podmínkami, logickými konvencemi i potřebami serverové strany (jež je součástí jiné BP).



# Kapitola 5

## Možná rozšíření

V této kapitole se pokusím shrnout možná rozšíření aplikace, která by vedla ke zkvalitnění celé aplikace. Nejvíce důležitým rozšířením pro aplikaci představuje aktivní monitoring pohybu automobilů, který by celou službu přiblížil ještě více reálnému světu.

Výčet možných rozšíření však nemusí být konečný a v případných navazujících dalších pracích je velice pravděpodobné, že dojde k rozšíření tohoto seznamu.

Některá rozšíření také reflektují skutečnost, že uživatelská část je závislá na funkcích části serverové (souběžná bakalářská práce), a proto nebylo možné do této uživatelské části aplikace implementovat určité funkce.

Pokud to vyjádřím jinými slovy, uživatelská část by prezentovala grafické rozhraní funkcionality, kterou neumí server zpracovat.

Tudíž je nutné prvořadě reflektovat možnosti serverové části a dle této skutečnosti vyvíjet uživatelskou část aplikace. I přes to je uživatelská strana již nyní v některých svých částech na rozšíření připravena a s možným přidáním dalších funkcionalit počítá.

### 5.1 Komunikace s podporou

Komunikace s podporou je jedna z důležitých součástí většiny služeb. Proto i v případě této aplikace se uvažuje o zavedení obdobné funkcionality. Existovalo by více forem, a to živý chat přímo na stránkách této služby, případně pomocí emailové komunikace.

Při implementaci této funkcionality je důležitá též úprava serverové části, jelikož nyní naprosto chybí jakákoliv funkcionalita komunikace mezi klienty a správci přímo na stránkách aplikace.

Implementace mailového serveru na backendu umožní tak zablokovaným uživatelům opětovně zažádání o odblokování a vysvětlit celou nastalou situaci správcům.

Dalším příkladem může být živá komunikace mezi klientem a správcem přes chat na stránce služby při určitých událostech nebo při nutnosti získat rychlou radu od podpory.

## 5.2 Možnosti obnovení účtu

Dané rozšíření by řešilo případ, kdy se uživatel rozhodl deaktivovat účet a nyní by žádal o jeho obnovení. Případně řešení pro situaci, kdy byl daný uživatel zablokován správcem a nyní chce vyjasnit důvody blokace.

Toto řešení velmi závisí na podpoře serverové strany, jelikož nyní nepodporuje žádnou formu komunikace, a to ani pomocí živých zpráv, ani skrze emailovou komunikaci. Po implementaci této funkcionality na serverové straně (jiná souběžná bakalářská práce), se může uvažovat o implementaci i na straně uživatelské.

S touto možností souvisí také přidružená funkcionality, jež se týká obnovení hesla k účtu. Pokud si přeje uživatel změnit heslo po přihlášení, nyní se aplikace s tímto požadavkem počítá.

Když by ale zapomněl přihlašovací údaje, uživatelské jméno, nebo heslo, nastává problém s obnovením nebo získáním těchto údajů. Do budoucna se tedy počítá s přidáním funkce odeslání emailu s odkazem na obnovení hesla u stávajícího účtu uživatele.

## 5.3 Schvalování účtu zaměstnancem

Rozšiřující funkci, která zvýší bezpečnost webové služby, představuje schvalování nově založeného klientského účtu ve službě správcem aplikace.

Při registraci bude stále nutné klientem zadat své osobní údaje, které jsou potřebné k vytvoření účtu. K této činnosti ale přibude povinnost připojit k registračnímu formuláři fotografii občanského a řidičského průkazu, a to z obou stran.

Také bude nutné vyfotografovat sebe sama s přední stranou občanského průkazu z důvodu ověření, že účet zakládá vlastních těchto průkazech.

Fotografie, které klient k registračnímu formuláři připojí, se odešlou spolu s osobními údaji na serverovou stranu. Správce služby pak bude mít možnost po přihlášení zobrazit si obrazovku se seznamem žadatelů o založení klientského účtu.

Po rozkliknutí položky žadatele správce zkontroluje zadané osobní údaje. K dispozici má také klientem odeslané fotografie osobních dokladů, které prošetří podle vnitřních předpisů (které se teprve ustanoví).

Poté schválí, případně neschválí žádost klienta o založení účtu.

Informace o výsledku schvalovacího procesu obdrží klient pomocí emailové zprávy (s čímž se pojí část této kapitoly o rozšířeních, přesněji Komunikace s podporou)

## 5.4 Monitoring pohybu automobilů

Jako poslední možné rozšíření zde zmiňuji také přidání pokročilých polohových a mapových možností aplikace.

V nynějším stavu se vozy ani při průběhu výpůjčky, ani po jejím ukončení, nepohybují po mapě a vůz používaný klientem svou pozici nemění v závislosti na poloze, kde byla rezervace ukončená.

Jediná současná možnost změnit polohu vozidla je úprava parametrů daného vozu správcem služby.

K tomu slouží webové službě stránka určená pro úpravu parametrů automobilu.

Z tohoto faktu také vyplývá, že při rozšíření je nutné do aplikace zapojit GPS lokátory, které budou namapovány na jednotlivá vozidla, přičemž každé vozidlo představuje v aplikaci unikátní GPS lokátor.

Dle polohy těchto lokačních zařízení se bude upravovat poloha daných záznamů reprezentující vozy v aplikaci.

Zamýšlené rozšíření využije připravené konstrukty, jenž jsou v klientské části částečně implementovány. Část potřebných funkcí obsahuje již nyní i serverová část, a však je nutná hlubší integrace získávání a zobrazování polohy při získávání dat o vozidlech ze serverové části.

Implementací tohoto rozšíření nabyde práce reálnějšího rozměru. Umožní také aplikaci provozovat v pilotním provozu s reálnými auty zapojenými do webové služby. Zároveň umožní kontrolovat jejich aktuální umístění na mapovém podkladu v reálném čase. Naplno se tak využije potenciál proběhnuvších implementačních prací na mapovém podkladu.

Jak již bylo zmíněno, část sounáležitostí pro implementaci těchto lokalizačních služeb je již připravena na rozšiřující činnosti přidávající tuto funkcionalitu do aplikace.

# Závěr

V rámci této bakalářské práce byla provedena rešerše stávajících řešení a průzkum možných technologických postupů použitelných při vývoji. Na základě zadání bakalářské práce a těchto poznatků byl připraven návrh aplikace. Stanovily se funkční a nefunkční požadavky, případy užití pro klienta a správce, průchody jednotlivými procesy a požadavky na uživatelské rozhraní.

Poté se na základě těchto požadavků aktualizoval prototyp vytvořený v rámci semestrálního projektu. S ohledem na vytvořený prototyp a seznam požadavků jsem zvolil vhodné nástroje a technologie, na kterých byla práce postavena. Jedná se především o prostředí ReactJS, který pro aplikaci představuje rámcovou knihovnu vývoje a umožnil tuto aplikaci realizovat bez větších obtíží.

Na základě zadání bakalářské práce a předchozích analýz se vybraly doplňující knihovny pro implementaci do aplikace. Pro uživatelské rozhraní (UI) byla zvolena knihovna Bootstrap. Komponenty této knihovny se začlenily do ReactJS rámcového prostředí. Tím knihovna umožnila vytvořit konzistentní, intuitivní a předvídatelné rozhraní služby.

Z analýzy také vyplynulo vhodné prostředí pro spuštění a ladění této aplikace, které představuje Node.js s balíkovací knihovnou NPM.

V neposlední řadě se také stanovily knihovny pro pokročilejší práci se soubory cookies, které zrychlily vývoj a umožnily bezpečnější a předvídatelnější práci s těmito soubory.

V rámci prací se provedeny testy aplikace, které jsou součástí příloh. Jmenovitě se jedná o testy uživatelského rozhraní pomocí dvou participantů, kteří plnili sadu předem daných úkolů a hodnotily intuitivnost a konzistenci prostředí. Dále také test vstupních polí klientské strany, přičemž byly sepsány seznamy množin možných vstupů a očekávaných akcí aplikace na tyto hodnoty.

Zdrojové kódy aplikace, po úspěšné implementaci a otestování, byly umístěny na školní repozitář. Návod na zprovoznění aplikace se nachází v příloze *Postup pro spuštění aplikace*. I přesto, že již nyní je aplikace značně rozsáhlá, stále se nachází oblasti, které by bylo možné zdokonalit. Proto byl v páté kapitole sepsán seznam možných rozšíření, jež práci při navazujících implementacích obohatí a rozšíří.

Aplikace dle všech ukazatelů splňuje nejen veškeré požadavky zadání bakalářské práce, ale disponuje i potenciálem do budoucna oslovit zákazníky a potenciální investory.

# Citovaná literatura

- [1] Wikipedie, AJAX. *Wikipedia.cz* [online]. 2021 [cit. 2021-5-17]. Dostupné z: <https://cs.wikipedia.org/wiki/AJAX>
- [2] About Node.js. *NodeJS* [online]. 2021 [cit. 2021-5-17]. Dostupné z: <https://nodejs.org/en/about/>
- [3] Toyota. *Toyota* [online]. [cit. 2021-5-17]. Dostupné z: <https://www.toyota.cz/>
- [4] Getting Started. *ReactJS* [online]. [cit. 2021-5-17]. Dostupné z: <https://reactjs.org/docs/getting-started.html>
- [5] MÁCA, Jindřich. Úvod do React. *ITnetwork* [online]. 2019 [cit. 2021-5-17]. Dostupné z: <https://www.itnetwork.cz/javascript/react/zaklady/uvod-do-react/>
- [6] MÁCA, Jindřich. Úvod: Node.js. *ITnetwork* [online]. 2018 [cit. 2021-5-17]. Dostupné z: <https://www.itnetwork.cz/javascript/nodejs/uvod-do-nodejs>
- [7] Documentation. *Leaflet* [online]. [cit. 2021-5-17]. Dostupné z: <https://leafletjs.com/reference-1.7.1.html#map-example>
- [8] HAVERBEKE, Marijn. *Eloquent: JavaScript* [online]. 2018 [cit. 2021-5-17]. Dostupné z: [https://eloquentjavascript.net/Eloquent\\_JavaScript.pdf](https://eloquentjavascript.net/Eloquent_JavaScript.pdf)
- [9] Úvod: Bootstrap. *Wikipedia* [online]. 2021 [cit. 2021-5-17]. Dostupné z: <https://cs.wikipedia.org/wiki/Bootstrap>
- [10] LINDLEY, Cody. Enlightenment: DOM. *DOM Enlightenment* [online]. 2013 [cit. 2021-5-17]. Dostupné z: <http://domenlightenment.com/>
- [11] NodeJS Introduction. *Tutorialspoint* [online]. [cit. 2021-5-17]. Dostupné z: [https://www.tutorialspoint.com/nodejs/nodejs\\_introduction.htm](https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm)
- [12] Usage: Material-UI. *Material UI* [online]. [cit. 2021-5-17]. Dostupné z: <https://material-ui.com/getting-started/usage/>
- [13] Ant Design of React. *Ant Design* [online]. [cit. 2021-5-17]. Dostupné z: <https://ant.design/docs/react/introduce>

- [14] Google Map API. *NPMJS* [online]. 2020 [cit. 2021-5-17]. Dostupné z: <https://www.npmjs.com/package/google-map-react>
- [15] JACQUES, Nilson. Introducing: Axios. *Site Point* [online]. 2018 [cit. 2021-5-17]. Dostupné z: <https://www.sitepoint.com/axios-beginner-guide/>
- [16] D., Maria a Kostya G.. Angular vs. React. *CLEVEROAD* [online]. 2021 [cit. 2021-5-17]. Dostupné z: <https://www.cleveroad.com/blog/angular-vs-react>
- [17] *TagGart* [online]. [cit. 2021-5-17]. Dostupné z: <https://www.taggart.cz/>
- [18] *Nejlevnější-autopůjčovna* [online]. [cit. 2021-5-17]. Dostupné z: <https://www.nejlevnejsi-autopujcovna.cz>
- [19] *Travel Car* [online]. [cit. 2021-5-17]. Dostupné z: <http://travelcar.cz/>
- [20] *HoppyGo* [online]. [cit. 2021-5-17]. Dostupné z: <https://hoppygo.com/cs>
- [21] *Angular* [online]. [cit. 2021-5-17]. Dostupné z: <https://angular.io/>
- [22] *GetBootstrap* [online]. [cit. 2021-5-17]. Dostupné z: <https://getbootstrap.com/>
- [23] Js-Cookie. *GitHub* [online]. [cit. 2021-5-17]. Dostupné z: <https://github.com/js-cookie/js-cookie>
- [24] React-cookie-consent. *NPMJS* [online]. [cit. 2021-5-17]. Dostupné z: <https://www.npmjs.com/package/react-cookie-consent>
- [25] *Pexels* [online]. [cit. 2021-5-17]. Dostupné z: <https://www.pexels.com/photo/man-holding-the-steering-wheel-while-driving-620335/>
- [26] *Pexels* [online]. [cit. 2021-5-17]. Dostupné z: <https://www.pexels.com/photo/man-in-yellow-shirt-driving-a-white-luxury-car-3758114/>
- [27] *FreeDesignFile* [online]. [cit. 2021-5-17]. Dostupné z: <https://freedesignfile.com/463244-couple-buying-new-car-stock-photo-02/>
- [28] *NPMJS* [online]. [cit. 2021-5-17]. Dostupné z: <https://www.npmjs.com/>

# Seznam zkratek

ČVUT	České Vysoké Učení Technické
FEL	Fakulta elektrotechnická
DOM	Document Object Model
GUI	Grafické uživatelské rozhraní
BP	Bakalářská práce
UI	Uživatelské rozhraní
NPM	Node Package Manager
JSON	JavaScript Object Notation
OSM	Open Street Map
GPS	Global Positioning System
VOP	Všeobecné Obchodní Podmínky
API	Application Programming Interface
MCC	Multiple Condition Coverage

# **PŘÍLOHY**



# Příloha A

## Postup pro spuštění aplikace

### A.1 Spuštění lokální instance

První krok představuje samotné stažení souborů, jež daná aplikace obsahuje.

Celá aplikace bakalářské práce byla umístěn na školním repozitáři:

[gitlab.fel.cvut.cz/jindrji5/bakalarska-prace](https://gitlab.fel.cvut.cz/jindrji5/bakalarska-prace)

Poté stačí stažený zip soubor jednoduše rozbalit. Postup po získání a rozbalení zdrojových souborů obnáší nutnost stáhnout z následující instalační stránky prostředí Node.js :

<https://nodejs.org/en/>

Instalace obsahuje konzoli, přes kterou budeme aplikaci spouštět.

Další možností spuštění aplikace je skrze aplikace WebStorm (případně PhpStorm). Vývojové prostředí WebStorm získáme z následujícího odkazu:

<https://www.jetbrains.com/webstorm/download/#section=windows>

Po instalaci vývojového prostředí najdeme v dedikovaném průzkumníku složku s aplikací a otevřeme ji.

Obecně doporučuji spíše spouštění skrze Node.js, jelikož se jedná o nejrychlejší a nejpohodlnější způsob, jak aplikaci spustit.

V případě však, že daného uživatele zajímá struktura aplikace více do hloubky, je vhodné otevřít aplikaci ve vývojovém prostředí. Nabízí totiž grafické rozlišení jednotlivých prvků v kódu a uživatel se v daném programu vyzná nejlépe.

Kvůli problémům jsou všechny knihovny (složka `nodes_modules` ) zabalené v zip archivu, tudíž před prvním spuštěním aplikace jakoukoliv cestou je nutné tento archiv rozbalit do kořenového adresáře aplikace.

Následně existují, jak jsem již zmínil, dvě hlavní cesty ke zkompileování aplikace. První zahrnuje spuštění aplikace *Node.js command prompt*. Po spuštění se přesměrujeme do složky s aplikací (příkazy *cd* pro změnu složky a *dir* pro výpis) .

Poté zadáme příkaz *npm start*, který spustí preview naší stránky.

Další možnost je ve vývojovém prostředí nastavit pro spuštění aplikace možnost `npm start` z možností kompilací aplikace, a následně spustit samotnou aplikaci pomocí tlačítka `start`. V případě změn v kódu jsou tyto změny ihned viditelné v preview.

Aby tato klientská část mohla být vyzkoušena, musí být na počítači spuštěny obě části aplikace, klientská (tato bakalářská práce) tak i serverová část (součást jiné souběžné bakalářské práce). Proto si prostudujte i návod pro spuštění serverové části.

## A.2 Přístup do služby

Databáze této aplikace obsahuje sadu připravených účtů z důvodu možnosti vyzkoušení funkcionalit této práce.

### **Pro přístup do klientské části slouží tyto údaje:**

- *Uživatelské jméno:* user
- *Heslo:* abcd1234

### **Pro přístup do správcovské části slouží tyto údaje:**

- *Uživatelské jméno:* admin
- *Heslo:* abcd1234

Pro založení rezervace je nutné do aplikace vložit reálnou platební kartu.

Pro vyzkoušení komunikace s platebním prostředím na serverové straně bez nutnosti vkládat reálnou a validní platební kartu je v systému připraven účet, který využívá platební prostředí v režimu testování, tudíž nedochází k reálným transakcím.

### **Pro přístup k účtu s testovacím platebním prostředím slouží tyto údaje:**

- *Uživatelské jméno:* testcard
- *Heslo:* abcd1234

Pokud v danou chvíli nemá tento účet přiřazenou platební kartu, vložte, prosím následující:

### **Testovací platební karta:**

- *Číslo:* 4000001240000000
- *Datum platnosti:* libovolné v budoucnosti
- *CVV kód:* libovolný

# Příloha B

## Testy uživatelského rozhraní

Testování uživatelského rozhraní aplikace má za úkol prověřit na reálných participantech flexibilitu, konzistentnost a předvídatelnost daného UI.

Na začátku testování uživatelského prostředí jsem určil dva participanty a jednotlivé scénáře s připravenými průchody službou, které by měli samostatně a organizovaně provést uživatele v aplikaci.

Průchod participanta aplikací včetně jeho problémů, výtek i pochval jsem následně sepsal do shrnující tabulky.

Nakonec jej vyhodnotím já, jakožto autor této aplikace, tak i participant, který test prováděl a zapíše se výsledky dílčího testu participanta, v podobě shrnutí tohoto konkrétního dílčího testu.

### **Souhrnný scénář testování části pro klienta:**

1. Registrovat se jako klient
2. Přihlášení klienta
3. Úprava hesla
4. Přidání platební karty
5. Zobrazení mapy s vozidly
6. Rezervovat vůz
7. Zobrazit přehled svých rezervací
8. Aktivovat čekající rezervaci
9. Ukončit rezervaci a zaplatit ji
10. Odhlásit se

### **Souhrnný scénář testování části pro správce:**

1. Přihlášení správce (údaje poskytnuty)
2. Zobrazení všech vozidel
3. Deaktivovat libovolný vůz
4. Přidat nové vozidlo
5. Zobrazit všechny uživatele
6. Vypsát detail o uživateli
7. Zablokovat uživatele
8. Odhlásit se

## B.1 Participant 1

**Jméno:** Petr Pazdera

**Věk:** 35 let

**Pozice:** úřední pracovník

### B.1.1 Přepis testovací situace pro roli klienta:

Participant se při prvním spuštění aplikace krátce seznamuje s rozvržením služby a brzy začíná plnit první krok testu registrovat se jako klient. Po přechodu na registrační formulář chválí přehlednost a živé zobrazování případných chyb v zadaných hodnotách. Vznáší dotaz na kontrolu obsazenosti uživatelského jména.

Koordinátor na dotaz odpovídá, že uživatelské jméno je nutné zadat unikátní.

Napoprvé také zadává špatný formát data narození. Po upozornění aplikací formát upravuje. Správně zaškrtnává tlačítko pro souhlas s VOP a úspěšně se registruje.

Následuje úkol se změnou hesla. Participant je mírně zmatený kde přesně nalézt osobní profil. Po chvíli se obrací se na hlavní panel a přesouvá se na odpovídající obrazovku. Přirozeně se zaměří na záložky v horní části uživatelského účtu a přesouvá se na položku *Upravit účet*, ve které je obsažena možnost změnit heslo.

K formuláři změny hesla nemá výhrady. Zdá se mu srozumitelný a jednoduchý. Automaticky zadává stávající heslo a chvíli váhá při vkládání nového. Formulář však vyplňuje úspěšně a úkol zdárně dokončil.

Při následujícím úkolu přidat platební kartu je v prostředí mírně zmatený. Zdá se mu nelogické, že nabídka pro přidání platební karty je součástí záložky *Rezervace uživatele*. Po stisknutí tlačítka pro vložení nové platební karty začíná vyplňovat povinné údaje a samotné přidání do aplikace zvládá bez větších obtíží.

Participant se přesouvá k úkolu zobrazit mapový podklad z dostupnými auty. Intuitivně nalézá možnost v hlavním panelu. Po samotném zobrazení mapy se ptá koordinátora, jaké další činnosti musí provést pro otevření možnosti rezervovat vůz.

Koordinátor sděluje, že postačuje kliknout na libovolný vyznačený symbol, jenž reprezentuje dostupné vozidlo. Po vybrání a kliknutí na symbol v mapovém podkladu se otevírají podrobnosti o daném vozidle a vybírá možnost *Půjč si mě*.

Při přesunutí do přehledu vlastních rezervací ihned vidí tuto aktivní rezervaci a aktivuje ji pomocí odpovídajícího tlačítka, načež čeká několik minut. Po uplynutí dané doby pěti minut stejnou rezervaci ukončuje a bez problému zaplatí. Koordinátor ohlašuje konec první části testu a participant se samostatně odhlásí z klientského účtu.

### **B.1.2 Přepis testovací situace pro roli správce:**

Participantovi byly poskytnuty údaje k vytvořenému účtu správce, a stejně jako v předešlém testu se přihlašuje intuitivně a samostatně.

Úkol zobrazení všech vozidel v aplikaci plní intuitivně vybráním odpovídající položky v horním panelu a nenachází na tomto úkonu žádné potíže.

Po zobrazení grafického znázornění vozidel si okamžitě všimá přepínače statusu přítomného u všech karet představujících jednotlivé automobily a klikem na tento grafický prvek deaktivuje vybraný vůz. Chválí okamžité přepsání statusu vozidla v podrobnostech před sebou.

Tlačítko pro přidání nového auta nachází participant v horním panelu. Participant komentuje umístění daného tlačítka pro přidání automobilu pozitivně, vznáší však připomínku. Intuitivně by ho očekával také v dolní části obrazovky u výpisu všech vozidel.

Po vysvětlení jednotlivých vlastností vozidla připravených k vyplnění nemá participant s kompletností formuláře větší potíže. Komentuje pouze nutnost zadat pozici vozidla ručně, přičemž mu je vysvětlena plánované rozšíření aplikace o další možnosti zpracovávání poloh automobilů.

Následně přirozeně a samostatně kontroluje na konci přehledu všech vozidel, zda se nově přidáný vůz opravdu uložil do aplikace po přidání nového auta kontroluje, zda se vůz nachází mezi ostatními automobily.

Jako poslední úkol, jež tento participant plní, představuje zobrazení seznamu všech klientů uložených v aplikaci. Seznam zobrazuje bez obtíží a vybírá náhodného klienta pro zobrazení jeho profilu se seznamem rezervací a jeho osobními údaji. Po zobrazení těchto informací kliká na tlačítko *Blokovat* a tím plní i předposlední úkol ze seznamu. Jakmile obdržel hlášku aplikace o úspěšném zablokování, svoji pozornost směřuje k tlačítku pro odhlášení z aplikace a úspěšně test dokončuje.

## B.2 Participan 2

**Jméno:** Ondřej Matějovský

**Věk:** 24 let

**Pozice:** student

### B.2.1 Přepis testovací situace pro roli klienta:

Participant na první pohled vidí tlačítko pro registraci. Po přechodu na registrační formulář začíná s vyplňováním údajů. Chvilu váhá ohledně správného formátu hesla, jelikož zadal nepovolený znak.

Aplikace mu však správný formát sděluje a chybu opravuje. Dále mírně váhá u zadávání data narození, seznamuje se s požadovaným formátem. Po jeho přečtení však napoprvé vkládá správně vyplněné datum. Správně zaškrťává tlačítko pro souhlas s VOP a úspěšně se registruje.

Následuje úkol se změnou hesla. Participant hledá na hlavním menu položku pro zobrazení účtu a nachází ji bez potíží. Po zobrazení obrazovky s informacemi o účtu komentuje zobrazenou hlášku o nepřidané platební kartě, která nás zatím nezajímá.

Intuitivně se přesouvá na záložku *Upravit účet* v horní části, kde je již formulář pro změnu hesla obsažen. Po zadání stávajícího hesla přirozeně vkládá nové heslo a jeho kontrolu. Chápe, že nové heslo se musí lišit od starého a úkol úspěšně dokončuje. Participant se přesouvá na záložku *Rezervace*, kde kliká na tlačítko *Přidat platební kartu* v hlášce varující o této skutečnosti.

Dále komentuje okno pro samotné přidání, které se po stisku tlačítka zobrazilo. Dialogu rozumí a dle své platné platební karty vyplňuje většinu žádaných informací. U formátu data platnosti se táže, zda před měsíc, jež je menší než deset, musí psát nulu. Po kladné odpovědi formulář s podrobnostmi o kartě následně odesílá a karta byla přijata aplikací.

Participant komentuje dialog stránky, která hlásí že je možné rezervovat některé z dostupných vozidel. Proto se participant přesouvá k úkolu zobrazení mapy s auty.

Intuitivně se vrací k hlavnímu panelu a vybírá možnost zobrazení dostupných aut, přičemž v zobrazené stránce přepíná v horní oblasti na záložku *Mapa*. Mírně váhá při zobrazení poloh vozidel, jelikož si není úplně jistý, zda lze na označnky poloh vozů kliknout.

Poté co na označnky jednoho vozu klik, čte si podrobnosti o daném vozidle a vybírá možnost *Půjč si mě*. Poté se snaží zobrazit přehled svých rezervací, což mu nečiní žádné potíže.

V přehledu rezervací vidí zvlášť zobrazenou aktivní rezervaci a kliká na přidružené tlačítko *Aktivovat*. Následně je nucen počkat 5 minut a rezervaci na stejném místě tlačítkem *Ukončit* také ukončuje. Vše probíhá bez zásahu koordinátora testování.

Po ukončení rezervace vidí participant na obrazovce tlačítko zaplatit, což odpovídá aktivitě aktuálního úkolu. Na tlačítko tedy kliká a aplikace mu dává najevo, že rezervace byla úspěšně zaplacená.

Participant sděluje překvapení, že aplikace rezervaci zaplatil automaticky a bez opětovného zadání některého z údajů jeho platební karty. Po zaplacení rezervace by se měl participant odhlásit a tím ukončit první část testu, což na povel koordinátora provádí bez potíží a plně intuitivně.

## **B.2.2 Přepis testovací situace pro roli správce:**

Participant obdržel údaje k přednastavenému účtu správce, a intuitivně kliká na tlačítko *Přihlásit* na horním panelu. Po přesném zadání odpovídajících správcových údajů je participant vpuštěn do aplikace s právy odpovídající správci.

Při úkolu zobrazení všech aut automaticky a intuitivně nachází položku na horním panelu a zobrazuje seznam všech vozidel v aplikaci.

Mírně váhá při deaktivaci vozu. Nejdříve vybírá možnost *Upravit vůz*, kdy se zobrazuje kompletní formulář pro změnu veškerých údajů. I zde se ale nachází prvek pro změnu statusu vozidla.

Jedná se ale o klikací přepínač s grafickým znázorněním. Uživatel si není jistý, jestli stačí pouze kliknout, nebo musí za prvek táhnout.

Po vysvětlení koordinátora, že stačí kliknutí participant změnil status vozidla a změny úspěšně ukládá. Po návratu na výpis všech vozidel si všimá předem opomenuté části grafického výpisu všech vozidel, přesněji klikacího přepínače pro rychlou změnu statusu vozidla přímo z výpisu. To zkouší také a pochvaluje okamžitou změnu statusu vozidla přímo na stránce bez znovu-načtení.

Participant začíná další úkol, a to přidání nového auta, přičemž má mírné potíže s nalezením tlačítka pro přidání. Očekával by ho v dolní části stránky u seznamu všech vozidel. Následně se přesouvá na hlavní stránku aplikace. Nakonec nachází požadované tlačítko v horním panelu. Po vysvětlení jednotlivých vlastností vozidla připravených k vyplnění nemá participant s kompletací větší potíže.

Následně po přidání nového auta kontroluje, zda se vůz nachází mezi ostatními automobily. Neskryvá mírné překvapení, že se nový vůz nachází až na konci seznamu karet, a není jako první položka. Ke konci testu plní participant jeden z posledních úkolů, a to zobrazit seznam klienty v aplikaci. Intuitivně nachází možnost v horním panelu a seznam zobrazuje. Po výběru libovolného uživatele zobrazuje jeho profil proklikem skrze uživatelské jméno. Předposlední úkol obsahuje zablokování uživatele. Uživatel automaticky a přirozeně stiská tlačítko *Blokovat*. Po dokončení úkonu se odhlašuje z účtu správce a test ukončuje.

# Příloha C

## Testy vstupů aplikace

### C.1 Přihlašovací formulář

Vstupy uživatele a možné stavy

- Uživatelské jméno (1)
  - V pořádku (A), prázdná hodnota (B), neexistuje (C)
- Heslo (2)
  - V pořádku (A), prázdná hodnota (B), neodpovídá (C)

Výsledek operace

- Úspěch (1), neúspěch (0)

**Testovací scénáře:**

Tabulka 3 Testovací scénáře přihlášení

Vstup	S1	S2	S3	S4	S5
1	A	B	C	A	A
2	A	A	A	B	C
Výsledek	1	0	0	0	0

**Výsledek testovacích scénářů:** Testy úspěšné



## C.2 Formulář registrace

### Vstupy uživatele a možné stavy

- Křestní jméno (1)
  - V pořádku (A), mimo povolenou délku (B), prázdná hodnota (C), nepovolené znaky (D)
- Příjmení (2)
  - Pořádku (A), mimo povolenou délku (B), prázdná hodnota (C), nepovolené znaky (D)
- Uživatelské jméno (3)
  - Pořádku (A), mimo povolenou délku (B), prázdná hodnota (C), nepovolené znaky (D), neunikátní (E)
- Email (4)
  - V pořádku (A), špatný formát (B), prázdná hodnota (C), nepovolené znaky (D)
- Heslo (5)
  - V pořádku (A), mimo povolenou délku (B), prázdná hodnota (C), nepovolené znaky (D)
- Heslo znovu (6)
  - V pořádku (A), prázdné (B), neshoduje se (C)
- Telefon (7)
  - V pořádku (A), mimo povolenou délku (B), prázdná hodnota (C), nepovolené znaky (D), neunikátní (E)
- Číslo řidičského průkazů (8)
  - V pořádku (A), mimo povolenou délku (B), prázdná hodnota (C), nepovolené znaky (D), neunikátní (E)
- Datum narození (9)
  - V pořádku (A), prázdná hodnota (B), méně než 18 let (C), špatný formát (D)
- Souhlas s VOP (10)
  - Ano (A), ne (B)

## Výsledek operace

- Úspěch (1), neúspěch (0)

## Testovací scénáře:

Tabulka 4 Testovací scénáře registrace

Vstup	S1	S2	S3	S4	S5	S6	S7
1	A	B	C	D	A	A	A
2	A	A	A	A	B	C	D
3	A	A	A	A	A	A	A
4	A	A	A	A	A	A	A
5	A	A	A	A	A	A	A
6	A	A	A	A	A	A	A
7	A	A	A	A	A	A	A
8	A	A	A	A	A	A	A
9	A	A	A	A	A	A	A
10	A	A	A	A	A	A	A
Výsledek	1	0	0	0	0	0	0

Tabulka 5 Testovací scénáře registrace

Vstup	S8	S9	S10	S11	S12	S13	S14
1	A	A	A	A	A	A	A
2	A	A	A	A	A	A	A
3	A	A	A	A	A	A	A
4	A	A	A	A	A	A	A
5	B	C	D	A	A	A	A
6	A	A	A	B	C	A	A
7	A	A	A	A	A	B	C
8	A	A	A	A	A	A	A
9	A	A	A	A	A	A	A
10	A	A	A	A	A	A	A
Výsledek	0	0	0	0	0	0	0

Tabulka 6 Testovací scénáře registrace

Vstup	S14	S15	S16	S17	S18	S19
1	A	A	A	A	A	A
2	A	A	A	A	A	A
3	A	A	A	A	A	A
4	A	A	A	A	A	A
5	A	A	A	A	A	A
6	A	A	A	A	A	A
7	D	E	A	A	A	A
8	A	A	B	C	D	E
9	A	A	A	A	A	A
10	A	A	A	A	A	A
<b>Výsledek</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

Tabulka 7 Testovací scénáře registrace

Vstup	S20	S21	S22	S23
1	A	A	A	A
2	A	A	A	A
3	A	A	A	A
4	A	A	A	A
5	A	A	A	A
6	A	A	A	A
7	A	A	A	A
8	A	A	A	A
9	B	C	D	A
10	A	A	A	B
<b>Výsledek</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

**Výsledek testovacích scénářů:** Testy úspěšné

## C.3 Formulář pro přidání platební karty

Vstupy uživatele a možné stavy

- Číslo platební karty (1)
  - V pořádku (A), mimo povolenou délku (B), prázdná hodnota (C), špatný formát (D), nepovolené znaky (E)
- Datum platnosti (2)
  - V pořádku (A), mimo povolenou délku (B), prázdná hodnota (C), špatný formát (D), nepovolené znaky (E)
- CVV kód (3)
  - V pořádku (A), mimo povolenou délku (B), prázdná hodnota (C), špatný formát (D), nepovolené znaky (E)

Výsledek operace

- Úspěch (1), neúspěch (0)

**Testovací scénáře:**

Tabulka 8 Testovací scénáře přidání platební karty

Vstup	S1	S2	S3	S4	S5	S6	S7
1	A	B	C	D	E	A	A
2	A	A	A	A	A	B	C
3	A	A	A	A	A	A	A
Výsledek	1	0	0	0	0	0	0

Tabulka 9 Testovací scénáře přidání platební karty

Vstup	S8	S9	S10	S11	S12	S13
1	A	A	A	A	A	A
2	D	E	A	A	A	A
3	A	A	B	C	D	E
Výsledek	0	0	0	0	0	0

**Výsledek testovacích scénářů:** Testy úspěšné

## C.4 Filtrace dostupných aut

Vstupy uživatele a možné stavy

- Model (1)
  - Aktivní a neprázdné (A), neaktivní a nezadané (B), neaktivní a neprázdné (C), aktivní a nezadané (D)
- Přezdívka (2)
  - Aktivní a neprázdné (A), neaktivní a nezadané (B), neaktivní a neprázdné (C), aktivní a nezadané (D)
- Poznávací značka (3)
  - Aktivní a neprázdné (A), neaktivní a nezadané (B), neaktivní a neprázdné (C), aktivní a nezadané (D)
- Typ pohonu (4)
  - Aktivní a neprázdné (A), neaktivní a nezadané (B), neaktivní a neprázdné (C), aktivní a nezadané (D)
- Typ převodového ústrojí (5)
  - Aktivní a neprázdné (A), neaktivní a nezadané (B), neaktivní a neprázdné (C), aktivní a nezadané (D)
- Minimální cena (6)
  - Aktivní a neprázdné (A), neaktivní a nezadané (B), neaktivní a neprázdné (C), aktivní a nezadané (D), > než maximální cena (E)
- Maximální cena (7)
  - Aktivní a neprázdné (A), neaktivní a nezadané (B), neaktivní a neprázdné (C), aktivní a nezadané (D), < než minimální cena (E)

Výsledek operace

- Úspěch (1), neúspěch (0)

**Testovací scénáře:**

Tabulka 10 Testovací scénáře filtrace dostupných aut

<b>Vstup</b>	<b>S1</b>	<b>S2</b>	<b>S3</b>	<b>S4</b>	<b>S5</b>	<b>S6</b>	<b>S7</b>
<b>1</b>	A	B	C	D	A	A	A
<b>2</b>	A	A	A	A	B	C	D
<b>3</b>	A	A	A	A	A	A	A
<b>4</b>	A	A	A	A	A	A	A
<b>5</b>	A	A	A	A	A	A	A
<b>6</b>	A	A	A	A	A	A	A
<b>7</b>	A	A	A	A	A	A	A
<b>Výsledek</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>

Tabulka 11 Testovací scénáře filtrace dostupných aut

<b>Vstup</b>	<b>S8</b>	<b>S9</b>	<b>S10</b>	<b>S11</b>	<b>S12</b>	<b>S13</b>	<b>S14</b>
<b>1</b>	A	A	A	A	A	A	A
<b>2</b>	A	A	A	A	A	A	A
<b>3</b>	B	C	D	A	A	A	A
<b>4</b>	A	A	A	B	C	D	A
<b>5</b>	A	A	A	A	A	A	B
<b>6</b>	A	A	A	A	A	A	A
<b>7</b>	A	A	A	A	A	A	A
<b>Výsledek</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>

Tabulka 12 Testovací scénáře filtrace dostupných aut

<b>Vstup</b>	<b>S15</b>	<b>S16</b>	<b>S17</b>	<b>S18</b>	<b>S19</b>	<b>S20</b>	<b>S21</b>
<b>1</b>	A	A	A	A	A	A	A
<b>2</b>	A	A	A	A	A	A	A
<b>3</b>	A	A	A	A	A	A	A
<b>4</b>	A	A	A	A	A	A	A
<b>5</b>	C	D	A	A	A	A	A
<b>6</b>	A	A	B	C	D	E	A
<b>7</b>	A	A	A	A	A	A	B
<b>Výsledek</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>

Tabulka 13 Testovací scénáře filtrace dostupných aut

Vstup	S22	S23	S24
1	A	A	A
2	A	A	A
3	A	A	A
4	A	A	A
5	A	A	A
6	A	A	A
7	C	D	E
Výsledek	0	0	0

Výsledek testovacích scénářů: Testy úspěšné

## C.5 Hledání vozu dle přezdívky

Vstupy uživatele a možné stavy

- Přezdívka vozu
  - V pořádku (A), nezadaná (B), neodpovídá (C)

Výsledek operace

- Úspěch (1), neúspěch (0)

**Testovací scénáře:**

Tabulka 14 Testovací scénáře hledání vozu dle přezdívky

Vstup	S1	S2	S3
1	A	B	C
Výsledek	1	0	0

Výsledek testovacích scénářů: Testy úspěšné

## C.6 Filtrace všech vozidel

Vstupy uživatele a možné stavy

- Model (1)
  - Aktivní a neprázdné (A), neaktivní a nezadané (B), neaktivní a neprázdné (C), aktivní a nezadané (D)
- Přezdívka (2)
  - Aktivní a neprázdné (A), neaktivní a nezadané (B), neaktivní a neprázdné (C), aktivní a nezadané (D)
- Poznávací značka (3)
  - Aktivní a neprázdné (A), neaktivní a nezadané (B), neaktivní a neprázdné (C), aktivní a nezadané (D)
- Typ pohonu (4)
  - Aktivní a neprázdné (A), neaktivní a nezadané (B), neaktivní a neprázdné (C), aktivní a nezadané (D)
- Typ převodového ústrojí (5)
  - Aktivní a neprázdné (A), neaktivní a nezadané (B), neaktivní a neprázdné (C), aktivní a nezadané (D)
- Minimální cena (6)
  - Aktivní a neprázdné (A), neaktivní a nezadané (B), neaktivní a neprázdné (C), aktivní a nezadané (D), > než maximální cena (E)
- Maximální cena (7)
  - Aktivní a neprázdné (A), neaktivní a nezadané (B), neaktivní a neprázdné (C), aktivní a nezadané (D), < než minimální cena (E)

Výsledek operace

- Úspěch (1), neúspěch (0)



**Testovací scénáře:**

Tabulka 15 Testovací scénáře filtrace všech vozů

Vstup	S1	S2	S3	S4	S5	S6	S7
1	A	B	C	D	A	A	A
2	A	A	A	A	B	C	D
3	A	A	A	A	A	A	A
4	A	A	A	A	A	A	A
5	A	A	A	A	A	A	A
6	A	A	A	A	A	A	A
7	A	A	A	A	A	A	A
<b>Výsledek</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>

Tabulka 16 Testovací scénáře filtrace všech vozů

Vstup	S8	S9	S10	S11	S12	S13	S14
1	A	A	A	A	A	A	A
2	A	A	A	A	A	A	A
3	B	C	D	A	A	A	A
4	A	A	A	B	C	D	A
5	A	A	A	A	A	A	B
6	A	A	A	A	A	A	A
7	A	A	A	A	A	A	A
<b>Výsledek</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>

Tabulka 17 Testovací scénáře filtrace všech vozů

Vstup	S15	S16	S17	S18	S19	S20	S21
1	A	A	A	A	A	A	A
2	A	A	A	A	A	A	A
3	A	A	A	A	A	A	A
4	A	A	A	A	A	A	A
5	C	D	A	A	A	A	A
6	A	A	B	C	D	E	A
7	A	A	A	A	A	A	B
<b>Výsledek</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>

Tabulka 18 Testovací scénáře filtrace všech vozů

<b>Vstup</b>	<b>S22</b>	<b>S23</b>	<b>S24</b>
<b>1</b>	A	A	A
<b>2</b>	A	A	A
<b>3</b>	A	A	A
<b>4</b>	A	A	A
<b>5</b>	A	A	A
<b>6</b>	A	A	A
<b>7</b>	C	D	E
<b>Výsledek</b>	<b>0</b>	<b>0</b>	<b>0</b>

**Výsledek testovacích scénářů:** Testy úspěšné

## C.7 Formulář pro změna hesla

Vstupy uživatele a možné stavy

- Stávající heslo (1)
  - V pořádku (A), prázdná hodnota (B), neodpovídá stávajícímu (C)
- Nové heslo (2)
  - V pořádku (A), mimo povolenou délku (B), prázdná hodnota (C), nepovolené znaky (D)
- Nové heslo znovu (3)
  - V pořádku (A), prázdná hodnota (B), neshoduje se (C)

Výsledek operace

- Úspěch (1), neúspěch (0)

**Testovací scénáře:**

Tabulka 19 Testovací scénáře pro změnu hesla

Vstup	S1	S2	S3	S4
1	A	B	C	A
2	A	A	A	B
3	A	A	A	A
Výsledek	1	0	0	0

Tabulka 20 Testovací scénáře pro změnu hesla

Vstup	S5	S6	S7	S8
1	A	A	A	A
2	C	D	A	A
3	A	A	B	C
Výsledek	0	0	0	0

**Výsledek testovacích scénářů:** Testy úspěšné

## C.8 Formulář pro přidání nového vozu

Vstupy uživatele a možné stavy

- Model (1)
  - V pořádku (A), prázdná hodnota (B)
- Přezdívka (2)
  - V pořádku (A), prázdná hodnota (B) , neunikátní (C)
- SPZ vozidla (3)
  - V pořádku (A), prázdná hodnota (B)
- Typ převodovky (4)
  - V pořádku (A), prázdná hodnota (B) , nevalidní hodnota (C)
- Maximální výkon (5)
  - V pořádku (A), prázdná hodnota (B), nečíselná hodnota (C)
- Typ pohonu (6)
  - V pořádku (A), prázdná hodnota (B) , nevalidní hodnota (C)
- Cena za minutu (7)
  - V pořádku (A), prázdná hodnota (B), nečíselná hodnota (C)
- Status (8)
  - V pořádku (A), prázdná hodnota (B) , nevalidní hodnota (C)
- Poloha (9)
  - V pořádku (A), prázdná hodnota (B) , nevalidní hodnota (C)

Výsledek operace

- Úspěch (1), neúspěch (0)

**Testovací scénáře:**

Tabulka 21 Testovací scénáře pro přidání vozu

<b>Vstup</b>	<b>S1</b>	<b>S2</b>	<b>S3</b>	<b>S4</b>	<b>S5</b>	<b>S6</b>
<b>1</b>	A	B	A	A	A	A
<b>2</b>	A	A	B	C	A	A
<b>3</b>	A	A	A	A	B	A
<b>4</b>	A	A	A	A	A	B
<b>5</b>	A	A	A	A	A	A
<b>6</b>	A	A	A	A	A	A
<b>7</b>	A	A	A	A	A	A
<b>8</b>	A	A	A	A	A	A
<b>9</b>	A	A	A	A	A	A
<b>Výsledek</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

Tabulka 22 Testovací scénáře pro přidání vozu

<b>Vstup</b>	<b>S7</b>	<b>S8</b>	<b>S9</b>	<b>S10</b>	<b>S11</b>	<b>S12</b>
<b>1</b>	A	A	A	A	A	A
<b>2</b>	A	A	A	A	A	A
<b>3</b>	A	A	A	A	A	A
<b>4</b>	C	A	A	A	A	A
<b>5</b>	A	B	C	A	A	A
<b>6</b>	A	A	A	B	C	A
<b>7</b>	A	A	A	A	A	B
<b>8</b>	A	A	A	A	A	A
<b>9</b>	A	A	A	A	A	A
<b>Výsledek</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

Tabulka 23 Testovací scénáře pro přidání vozu

<b>Vstup</b>	<b>S13</b>	<b>S14</b>	<b>S15</b>	<b>S16</b>	<b>S17</b>
<b>1</b>	A	A	A	A	A
<b>2</b>	A	A	A	A	A
<b>3</b>	A	A	A	A	A
<b>4</b>	A	A	A	A	A
<b>5</b>	A	A	A	A	A
<b>6</b>	A	A	A	A	A
<b>7</b>	C	A	A	A	A
<b>8</b>	A	B	C	A	A
<b>9</b>	A	A	A	B	C
<b>Výsledek</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>

**Výsledek testovacích scénářů:** Testy úspěšné

## C.9 Formulář pro úpravu vozidla

Vstupy uživatele a možné stavy

- Model (1)
  - V pořádku (A), prázdná hodnota (B)
- Přezdívka (2)
  - V pořádku (A), prázdná hodnota (B) , neunikátní (C)
- SPZ vozidla (3)
  - V pořádku (A), prázdná hodnota (B)
- Typ převodovky (4)
  - V pořádku (A), prázdná hodnota (B) , nevalidní hodnota (C)
- Maximální výkon (5)
  - V pořádku (A), prázdná hodnota (B), nečíselná hodnota (C)
- Typ pohonu (6)
  - V pořádku (A), prázdná hodnota (B) , nevalidní hodnota (C)
- Cena za minutu (7)
  - V pořádku (A), prázdná hodnota (B), nečíselná hodnota (C)
- Status (8)
  - V pořádku (A), prázdná hodnota (B) , nevalidní hodnota (C)
- Poloha (9)
  - V pořádku (A), prázdná hodnota (B) , nevalidní hodnota (C)

Výsledek operace

- Úspěch (1), neúspěch (0)

**Testovací scénáře:**

Tabulka 24 Testovací scénáře pro úpravu vozu

<b>Vstup</b>	<b>S1</b>	<b>S2</b>	<b>S3</b>	<b>S4</b>	<b>S5</b>	<b>S6</b>
<b>1</b>	A	B	A	A	A	A
<b>2</b>	A	A	B	C	A	A
<b>3</b>	A	A	A	A	B	A
<b>4</b>	A	A	A	A	A	B
<b>5</b>	A	A	A	A	A	A
<b>6</b>	A	A	A	A	A	A
<b>7</b>	A	A	A	A	A	A
<b>8</b>	A	A	A	A	A	A
<b>9</b>	A	A	A	A	A	A
<b>Výsledek</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

Tabulka 25 Testovací scénáře pro úpravu vozu

<b>Vstup</b>	<b>S7</b>	<b>S8</b>	<b>S9</b>	<b>S10</b>	<b>S11</b>	<b>S12</b>
<b>1</b>	A	A	A	A	A	A
<b>2</b>	A	A	A	A	A	A
<b>3</b>	A	A	A	A	A	A
<b>4</b>	C	A	A	A	A	A
<b>5</b>	A	B	C	A	A	A
<b>6</b>	A	A	A	B	C	A
<b>7</b>	A	A	A	A	A	B
<b>8</b>	A	A	A	A	A	A
<b>9</b>	A	A	A	A	A	A
<b>Výsledek</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>



Tabulka 26 Testovací scénáře pro úpravu vozu

<b>Vstup</b>	<b>S13</b>	<b>S14</b>	<b>S15</b>	<b>S16</b>	<b>S17</b>
<b>1</b>	A	A	A	A	A
<b>2</b>	A	A	A	A	A
<b>3</b>	A	A	A	A	A
<b>4</b>	A	A	A	A	A
<b>5</b>	A	A	A	A	A
<b>6</b>	A	A	A	A	A
<b>7</b>	C	A	A	A	A
<b>8</b>	A	B	C	A	A
<b>9</b>	A	A	A	B	C
<b>Výsledek</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>

**Výsledek testovacích scénářů:** Testy úspěšné