

České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra počítačů



Aplikace „Půjčovna aut“ – serverová  
strana

BAKALÁŘSKÁ PRÁCE

*Vypracoval:* **Artur Sarhsjan**

*Vedoucí práce:* **Ivan Jelínek**

*Rok:* **2021**





# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Sarhsjan** Jméno: **Artur** Osobní číslo: **475597**  
 Fakulta/ústav: **Fakulta elektrotechnická**  
 Zadávající katedra/ústav: **Katedra počítačů**  
 Studijní program: **Softwarové inženýrství a technologie**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Aplikace "Půjčovna aut" - serverová strana**

Název bakalářské práce anglicky:

**'Car rental' application - server-side**

Pokyny pro vypracování:

Cílem bakalářské práce je navrhnout serverovou stranu aplikace, spolupracuje s klientskou stranou a spolu zajišťuje možnost zapůjčení aut.

- proveďte přezkum požadavků pro backend stranu aplikace
- zmapujte vhodné nástroje pro integraci do backend strany aplikace
- definujte možné scénáře a funkce na backend straně
- navrhnete implementaci platebního prostředí na backend stranu a zachování bezpečnosti plateb.
- serverovou stranu aplikace implementujte a odzkoušejte v návaznosti na část klientskou
- navrhnete vhodný způsob testování a výsledky funkce aplikace otestujte a zhodnotte výsledky testů
- navrhnete možnosti dalšího rozvoje aplikace

Seznam doporučené literatury:

- Vahid Yaghini. Create Web Applications Efficiently with the Spring Boot MVC Framework [online]. OpenClassRooms. Poslední změna 7-3-2020 [cit. 23-11-2020]. Dostupné z: [openclassrooms.com/en/courses/5684146-create-web-applications-efficiently-with-the-spring-boot-mvc-framework/6156961-organize-your-application-code-in-three-tier-architecture](https://openclassrooms.com/en/courses/5684146-create-web-applications-efficiently-with-the-spring-boot-mvc-framework/6156961-organize-your-application-code-in-three-tier-architecture)
- Baeldung. The DAO Pattern in Java [online]. Baeldung. Poslední změna 21-3-2020 [cit. 23-11-2020]. Dostupné z: [baeldung.com/java-dao-pattern](https://baeldung.com/java-dao-pattern)
- Thorben Janssen. Standardized schema generation and data loading with JPA 2.1 [online]. © 2020, Thorben Janssen. Dostupné z: [thorben-janssen.com/standardized-schema-generation-data-loading-jpa-2-1/#Simple\\_configuration\\_to\\_create\\_the\\_databases](https://thorben-janssen.com/standardized-schema-generation-data-loading-jpa-2-1/#Simple_configuration_to_create_the_databases)

Jméno a pracoviště vedoucí(ho) bakalářské práce:

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

Jméno a pracoviště druhého(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: .....

.....

Artur Sarhsjan

Poděkování

Děkuji doc. Ing. Ivanu Jelínkovi, CSc., za vedení mé bakalářské práce a za podnětné návrhy, které ji obohatily.

.....

Artur Sarhsjan

*Název práce:* Aplikace „Půjčovna aut“ - serverová strana

*Autor:* Artur Sarhsjan

*Program:* 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:* Tato práce se zabývá vytvořením serverové části aplikace na půjčování aut. V první části obsahuje analýzu již existujících řešení, včetně silných a slabých stránek, a uvádí zlepšení, která server bude poskytovat oproti zmíněným řešením. Ve druhé kapitole jsou popsány požadavky, kladené na vyvíjený server. Sem patří uživatelské role, funkční i nefunkční požadavky, životní cykly objektů a modelování hlavních procesů, které budou probíhat při využívání serveru. Třetí kapitola popisuje technologie, které byly použity k naplnění fungování serveru. Jedná se o architekturu serveru a úložiště dat a jejich vzájemnou komunikaci, použitý framework, implementovanou autentizaci a autorizaci a platební bránu a testování funkčnosti serveru jako takového. Čtvrtá kapitola je věnovaná popisu testování vstupů a průchodů serverem a shrnutí výsledků testování. Poslední část práce obsahuje popis rozšíření, které by mohly být implementovány v budoucnu s cílem zlepšení fungování serveru.

*Klíčová slova:* bakalářská práce, webová aplikace, frontend, klient, klientská strana, backend, server, serverová strana, automatizace, platební brána, j2Pay, Stripe, projekt, Java, Spring, SQL, databáze, třívrstvá architektura, ORM, DAO, REST, autentizace, autorizace, HTTP request, enum, case-sensitive

*Title:* ,Car rental‘ application – server-side

*Author:* Artur Sarhsjan

*Abstract:* This work describes the development and implementation of the server side of the application designed for renting cars. The first chapter is dedicated to the analysis of the already existing solutions. The analysis includes discovered pros and cons and introduces improvements that our server presents compared to the mentioned solutions. The second chapter of the thesis contains requirements for the developed server. Those include user roles, functional and non-functional requirements, objects’ lifecycles, modeling and visualization of the most important processes that will take place during the server’s usage. The third chapter describes technologies that were used to develop and implement this server per the defined requirements. Those technologies include architecture of both the server and data storage and communication between them, a framework that will be used for the server’s development, authentication and authorization mechanisms, payment gateway and server functioning testing tools. The fourth chapter goes into detail about which functionalities and expected inputs and outputs are tested. The last chapter focuses on possible improvements that can be implemented in the future.

*Key words:* Bachelor Thesis, Web Application, Frontend, Client, Client Side, Backend, Server, Server Side, Automation, Payment Gateway, J2Pay, Stripe, Project, Java, Spring, SQL, Database, Three-Tier Achitecture, ORM, DAO, REST, Authentication, Authorization, HTTP Request, Enum, Case-Sensitive

# Obsah

Úvod	10
1. Analýza existujících řešení	11
1.1. Popis existujících řešení a jejich vyhodnocení	11
1.2. V čem budeme lepší a efektivnější	13
1.3. Shrnutí	13
2. Analýza požadavků	14
2.1. Uživatelské role	14
2.2. Funkční požadavky	14
2.3. Nefunkční požadavky	19
2.4. Diagram tříd	20
2.5. Diagramy případů užití (Use Case Diagram)	21
2.6. Model hlavních procesů	23
2.7. Stavové diagramy	24
3. Použité technologie	25
3.1. Úložiště dat	25
3.2. Framework	25
3.3. Architektura serveru	26
3.4. Autentizace a autorizace	27
3.5. Platební brána	28
3.6. Testování	30
4. Testování	31
4.1. Testování vstupů	31
4.2. Průchody procesem - zákazník	36
4.3. Průchody procesem – zaměstnanec	42
4.4. Výsledky testů	45
5. Možná rozšíření	46
5.1. Validace účtu pomocí emailu	46
5.2. Validace účtu zaměstnancem – ověření řidičského průkazu	46
5.3. Možnost pro zaměstnance vypsát rezervace podle zákazníka	47
5.4. Dvoufázové přihlášení	47
5.5. Lightweight Cache	47
5.6. Historie pohybu auta	47
6. Závěr a zhodnocení	48



<b>Literatura</b>	50
<b>Přílohy</b>	52
<b>Spuštění aplikace</b>	52
<b>JUnit testy</b>	54
<b>Postman testy</b>	56
<b>Testování vstupů</b>	60
<b>Testy průchodů procesem zákazníkem</b>	78
<b>Testy průchodů procesem zaměstnancem</b>	80

# Úvod

Cílem této práce je navržení backendu, který by poskytl naplnění požadavků, kladených na aplikaci, určenou pro umožnění rychlého a efektivního půjčování aut, a který by mohl vyřešit předpokládané problémy, se kterými se serverové strany platformy k půjčování aut potýkají. Součástí této práce byl průzkum již existujících řešení a jejich nedostatků. Na základě tohoto průzkumu a také naší vize o platformě na půjčení aut jsme definovali požadavky, které by serverová část měla splnit. S ohledem na tyto požadavky byly definovány nástroje, které budou k jejich naplnění sloužit, a struktura serverové strany počínaje přístupem k úložišti dat a končící umožněním komunikace se stranou klientskou.

Výsledek této práce by mohl sloužit nejen jako návod k celkovému plánování a vývoji serveru pro platformu na půjčování vozidel, ale také k implementaci jednotlivých funkcionalit včetně platební brány, přístupu k úložišti dat a logiky serverové strany aplikace.

# 1. Analýza existujících řešení

## 1.1. Popis existujících řešení a jejich vyhodnocení

- **Autorent Praha (autorentpraha.cz)**

- Auta jsou kategorizovaná podle typů vozidla. U každého vozidla je uvedena cena za půjčení, vždy specifikovaná pro úsek času, na který si chce klient vozidlo zapůjčit. Ve většině případů je maximální doba výpůjčky 1 měsíc.
- Rezervace vozidla probíhá prostřednictvím jednoduchého formuláře, v rámci kterého klient posílá k zpracování svoje jméno a název firmy, typ vozidla, kontaktní údaje, den rezervace a dobu, po kterou chce zákazník vozidlo používat.
- Stránka disponuje platformou pro nákup vozidla, a také pro prodej vozidla.
- Server také disponuje funkcí, která umožňuje auta kategorizovat, čímž naplňuje jeden z předpokládaných cílů, konkrétně se má na mysli zjednodušení vyhledávání požadovaného auta.

### **Nedostatky:**

- Není jasné, zda půjčovna disponuje platformou umožňující úhradu prostřednictvím online platby, bankovním převodem či jinak.
- Také zde chybí funkcionalita, která by umožnila zákazníkovi zjistit, zda je jím zvolené auto vůbec k dispozici ve zvolený časový úsek.
- Zákazník uvádí požadovaný typ vozidla do jednoduchého formulářového pole. Na serverovou stranu bude zaslán řetězec, ze kterého nemusí být vůbec jasné, jaké auto zákazník chce rezervovat. Na základně této skutečnosti dělám závěr, že serverová strana nedisponuje funkcí, umožňující zákazníkovi samostatně rezervovat jim požadované auto.
- Ze dvou předchozích bodů vyplývá, že proces rezervace auta je nedostatečně automatizovaný a že v mnoha případech je potřeba do databáze zasahovat manuálně – jestli vůbec.

- **GetRentaCar (getrentacar.com)**

- Aplikace obsahuje funkci, umožňující vyhledávat vozidla, která se nacházejí v okolí zákazníkem zadané adresy.

- Ke každému vozidlu jsou uvedeny rok vydání, kapacita motoru, pohon, spotřeba paliva a cena za půjčení za den.
- Rezervovat auto může pouze zákazník, přihlášený ke svému účtu. K přihlášení jsou na server poslány email a heslo.
- Server také obsahuje funkčnost, umožňující kategorizaci vozidel podle kritérií, čímž zákazníkovi zjednodušuje nalezení požadovaného auta.

**Nedostatky:**

- Proces registrace je velmi pomalý, po odeslání osobních údajů na server je třeba čekat až 20 min, než server zašle aktivační kód na email.
- Chybí funkcionalita, která by umožnila zákazníkovi zjistit, zda je jím zvolené auto k dispozici ve zvolený časový úsek.
- Není jasné, zda stránka disponuje platební bránou.
- Není uvedeno, kdy a jakým způsobem bude probíhat platba.

● **Autopůjčovna v Praze (autopujcovnavpraze.cz)**

- Prostřednictvím kalendářního formuláře zákazník pošle na server požadavek na výpis všech aut, který jsou v jím zvolený časový úsek k dispozici.
- Auta jsou detailně kategorizovaná podle typů vozidla. U každého vozidla je uvedena cena za půjčení, vždy specifikovaná pro úsek času, na který si chce klient vozidlo zapůjčit.
- Rezervace vozidla probíhá pomocí formuláře, prostřednictvím kterého klient posílá k zpracování svoje jméno a kontaktní údaje, dobu zapůjčení vozidla, informaci zda bude za půjčení platit hotově či platební/debetní kartou, a také adresu, na kterou dané auto bude dostaveno.

**Nedostatky:**

- Zákazník nemusí mít přehled o všech svých výpůjčkách - nemá účet, ke kterému by byly výpůjčky přiřazené.
- Stránka nedisponuje platební bránou.

● **Prima Půjčovna (primapujcovna.cz)**

- Prostřednictvím formuláře, do kterého se zadává doba zapůjčení auta a sídlo pobočky (Praha/Plzeň) zákazník pošle na server požadavek na výpis všech aut, který jsou v jím zvolený časový úsek a jím zvoleném okolí k dispozici.
- Rezervace vozidla probíhá pomocí formuláře, prostřednictvím kterého klient posílá k zpracování svoje jméno a kontaktní údaje, dobu zapůjčení vozidla, adresu, na které zákazník vozidlo vyzvedne, a adresu na kterou zákazník vozidlo vrátí.
- Rezervace probíhá rychle a jednoduše bez potřeby registrace

### **Nedostatky:**

- Zákazník nemusí mít přehled o všech svých výpůjčkách - nemá účet, ke kterému by byly výpůjčky přiřazené.
- Stránka nedisponuje platební bránou.

## **1.2. V čem budeme lepší a efektivnější**

- Jednoduchá a rychlá registrace.
- Automatizovaný proces rezervace auta.
- Zákazník bude vědět, zda je jím zvolené auto v daném časovém úseku k dispozici.
- Platební prostředí, které umožní zákazníkovi z domova provést platbu za půjčení vozidla.
- Zákazník bude mít přehled o všech svých rezervacích.

## **1.3. Shrnutí**

V současné době vidíme rozvoj a nárůst popularity autopůjčoven. Ovšem proces půjčení auta v malých a středních podnicích vyžaduje četné množství zásahů do aplikace jak ze strany zaměstnanců, tak i ze strany zákazníků. Některým aplikacím chybí nástroje, umožňující zákazníkům založit a uzavřít (zaplatit za) výpůjčku rychle a odkudkoliv, a také nástroje, které by umožnily zákazníkům mít přehled o svých výpůjčkách. Naším cílem je navrhnout server, který poskytuje větší automatizaci procesů, souvisejících s půjčením auta, včetně založení a uzavření rezervace a její zaplacení, a také umožní svým zákazníkům mít přehled o svých výpůjčkách.

## 2. Analýza požadavků

### 2.1. Uživatelské role

- **Uživatel**

Uživatel se může registrovat a přihlásit se. Po registraci se může uživatel přihlásit jako zákazník nebo jako zaměstnanec.

- **Zákazník**

Zákazník si může zobrazit auta, která jsou momentálně teď k dispozici pro rezervaci, filtrovat je podle kritérií (výčet kritérií jsou k nalezení v následující podkapitole), připojit k účtu platební kartu, upravit osobní údaje, měnit heslo ke svému účtu, upravit údaje o platební kartě, zakládat a rušit rezervaci auta, zobrazit všechny rezervace, které založil a deaktivovat účet.

- **Zaměstnanec**

Zaměstnanec si může zobrazit všechna auta, která jsou nebo byla používána, přidávat nová auta do systému a stahovat je z nabídky pro zákazníky. Zaměstnanec si může zobrazit záznamy o všech rezervacích, založených zákazníky v rámci této aplikace, deaktivovat účet zákazníka a registrovat dalšího zaměstnance. Zaměstnanec také může upravovat osobní údaje a měnit heslo ke svému účtu.

### 2.2. Funkční požadavky

- **Registrace**

Dostupné pro jakéhokoliv uživatele: systém umožní založit účet. Uživatel uvede uživatelské jméno, heslo, křestní jméno a příjmení, email, telefonní číslo a číslo řidičského průkazu.

Žádný z uvedených údajů nesmí zůstat nevyplněný. Uživatelské jméno musí být unikátní (neexistuje účet se stejným uživatelským jménem), obsahovat pouze písmena latinské abecedy a číslice. Heslo musí obsahovat minimálně 8 znaků, povolena jsou pouze písmena latinské abecedy a číslice. Křestní jméno a příjmení obsahují pouze písmena latinské abecedy. Email musí být

unikátní a být ve správném formátu ([nazev]@[domena]). Telefonní číslo musí také být unikátní a obsahovat pouze 9 číslic.

- **Přihlášení**

Dostupné pro jakéhokoliv uživatele: systém umožní přihlásit se do vlastního účtu. Pokud uživatel s daným uživatelským jménem a heslem v databázi existuje, server pošle na frontend objekt typu **AuthenticationToken** obsahující objekt uživatele, jemuž dané uživatelské jméno a heslo odpovídá, a také práva jenž danému uživateli odpovídají.

Pokud objekt nebude nalezen, server vrátí chybovou odpověď.

- **Výpis všech aut**

Dostupné pro zaměstnance: systém umožní zobrazit informace o všech autech, které naše firma má nebo měla k dispozici.

- **Výpis aut, které v daný okamžik je možné půjčit**

Dostupné pro zákazníky: systém umožní zobrazit informace o autech, která si zákazník může v daný okamžik půjčit, tzn. nejsou na nich vázány neuzavřené rezervace.

- **Výpis aut podle zvolených kritérií**

Dostupné pro zákazníky: systém umožní zobrazit informace o autech, která odpovídají specifickým kritériím, jež zákazník uvedl. Mezi tyto kritéria patří: název modelu auta, typ pohonu (benzin/diesel/elektrický), typ převodovky (manual/automatic), cena za minutu, výkon.

- **Přidání auta**

Dostupné pro zaměstnance: systém umožní přidání nových aut do systému. Údaje, které k autu musí patřit: model auta, název, typ pohonu (benzin/diesel/elektrický), typ převodovky (manual/automatic), cena za minutu (min. 15 Kč), výkon, státní poznávací značka (dále jen SPZ), zeměpisná šířka a zeměpisná délka (poloha auta; můžou být nulové).

Žádný z uvedených údajů nesmí být prázdný. To samé platí i pro výkon. SPZ musí být unikátní, tzn. v databázi není žádné auto, které má stejnou SPZ a je stále v provozu. Hodnota zeměpisné šířky nesmí být větší než 90 či menší než -90. Hodnota zeměpisné délky nesmí být větší než 180 či menší než -180.

- **Deaktivace auta**

Dostupné pro zaměstnance: systém umožní změnit status auta na neaktivní. Takové auto již není v provozu, a nemůže se zobrazit zákazníkovi při hledání auta na půjčení.

- **Přiřazení platební karty**

Dostupné pro zákazníky: systém umožní připojit k účtu platební kartu.

Zákazník vyplní číslo karty, měsíc a rok platnosti a CVV. Číslo karty musí obsahovat 16 číslic, měsíc a rok platnosti v budoucnosti a CVV musí obsahovat 3 číslice. Pokud zákazník již má přiřazenou platební kartu a chce připojit k účtu jinou, musí nejdříve odebrat současnou.

Pro ověření validity platební karty systém strhne 15 Kč z bankovního účtu. Pokud transakce nebyla úspěšná, systém vrátí chybovou odpověď. Pokud transakce byla úspěšná, systém zkontroluje, zda v databázi již je uložena platební karta s tímto číslem. Pokud ano, systém připojí k uživateli již existující kartu, pokud ne, systém uloží novou platební kartu do databáze.

- **Úprava údajů o platební kartě**

Dostupné pro zákazníky: systém umožní změnit údaje platební karty, připojené k danému účtu.

Pokud zákazník změní číslo karty, systém zkontroluje, zda toto číslo nenáleží jiné platební kartě v databázi. Pokud ano, systém vrátí chybovou odpověď. Pokud ne, systém vloží nový záznam do databáze.

Číslo karty musí obsahovat 16 číslic, měsíc a rok platnosti v budoucnosti a CVV musí obsahovat 3 číslice.

- **Odebrání platební karty**

Dostupné pro zákazníky: systém umožní odebrat platební kartu ze svého účtu. Pokud zákazník má nějakou rezervaci neuzavřenou či nezaplacenou, systém vrátí chybovou odpověď a kartu neodebere.

- **Založení rezervace**

Dostupné pro zákazníky: systém umožní rezervovat jakékoliv auto, které je momentálně k dispozici. Zákazník musí mít přiřazenou platební kartu, jejíž datum platnosti nevypršel.



- **Aktivace rezervace**

Dostupné pro zákazníky: systém umožní změnit status rezervace na neaktivní při převzetí rezervovaného auta.

Pokud od založení rezervace uběhlo 30 minut a rezervace nebyla ukončená či zrušená, systém provede automatickou aktivace rezervace.

- **Zrušení rezervace**

Dostupné pro zákazníky: systém umožní zrušit rezervaci do 30 minut od jejího založení. Pokud rezervace je již aktivovaná, rezervaci zrušit nelze.

- **Ukončení rezervace**

Dostupné pro zákazníky: systém umožní řádné ukončení aktivované rezervace, tzn. změnit status rezervace z aktivní na ukončenou.

Při okamžitém ukončení rezervace, systém vypočítá částku, kterou zákazník musí zaplatit za danou rezervaci, a tuto částku mu strhne z účtu, případně ji uživatel zaplatí sám. Minimální částka je 15 Kč. Pokud by platba nebyla úspěšná, zákazník vždy může rezervaci zaplatit sám.

- **Zaplacení rezervace**

Dostupné pro zákazníky: systém umožní řádně ukončenou rezervaci zaplatit.

Při okamžitém ukončení rezervace, systém vypočítá částku, kterou zákazník musí zaplatit za danou rezervaci, a tuto částku mu strhne z účtu. Pokud by platba nebyla úspěšná, zákazník může rezervaci zaplatit sám.

Částka se vypočítá na základě každé započaté minuty. Např. jestliže rezervace trvala 19 minut 18 sekund, zákazník je povinen zaplatit částku rovnou uvedené ceně za minutu vynásobené 20krát.

- **Zobrazení historie vlastních rezervací**

Dostupné pro zákazníky: systém umožní zákazníkovi vypsát všechny rezervace, které kdy založil.

- **Zobrazení historie všech rezervací**

Dostupné pro zaměstnance: systém umožní zobrazit všechny rezervace, které byly založeny v této aplikaci.

- **Výpis svých osobních údajů**

Dostupné pro zákazníky i zaměstnance: systém umožní zobrazit své údaje, uvedené v registračním formuláři. Jediným údajem, jenž se nezobrazí, je heslo.

- **Výpis osobních údajů každého zákazníka i zaměstnance**

Dostupné pro zaměstnance: systém umožní zobrazit osobní údaje každého zákazníka a zaměstnance, jenž byly založeny v této aplikaci. Jediným údajem, jenž se nezobrazí, je heslo.

- **Úprava svých osobních údajů**

Dostupné pro zákazníky i zaměstnance: systém umožní upravit své současné osobní údaje kromě uživatelského jména a hesla.

U každého z údajů platí, že nesmí zůstat nevyplněný. Křestní jméno a příjmení obsahují pouze písmena latinské abecedy. Email musí být unikátní a být ve správném formátu ([nazev]@[domena]). Telefonní číslo musí také být unikátní a obsahovat pouze 9 číslic.

- **Změna hesla**

Dostupné pro jakéhokoliv uživatele: systém umožní změnit heslo ke svému účtu. Uživatel uvede svoje současné heslo, heslo nové a zopakované nové heslo. Pro nové heslo platí stejné požadavky, jako v případě hesla uvedeného při registraci. Nové heslo a zopakované nové heslo se musí shodovat.

- **Deaktivace účtu zákazníka**

Dostupné pro zaměstnance a zákazníky: systém umožní deaktivovat účet zákazníka. Účet lze deaktivovat pouze v případě, kdy zákazník ještě nemá žádné rezervace založené, či v případě, kdy zákazník má všechny rezervace ukončené a zaplacené (zrušené rezervace nemusí být zaplacené). Zákazník může deaktivovat pouze svůj účet, zaměstnanec může deaktivovat účet jakéhokoliv zákazníka. Po deaktivaci nelze účet znovu aktivovat.

## 2.3. Nefunkční požadavky

- **Integrace**

Jelikož v rámci projektu implementujeme serverovou část aplikace, je potřeba zajistit, aby server mohl komunikovat se stranou klientskou, umožnit mu přístup ke svým funkcionalitám, posílat data klientovi či tato data od něj přijímat.

- **Přístupnost**

Jelikož ne všechny funkcionality smí být přístupné všem uživatelům, je potřeba zajistit, aby přístup k jednotlivým funkcionalitám vycházel z role a práv, kterými uživatel disponuje, tzn. přístup k jednotlivým funkcionalitám záleží na tom, jakou roli (nepřihlášený uživatel/zákazník/zaměstnanec) a jaká práva jsou uživateli přiřazena.

- **Integrita dat**

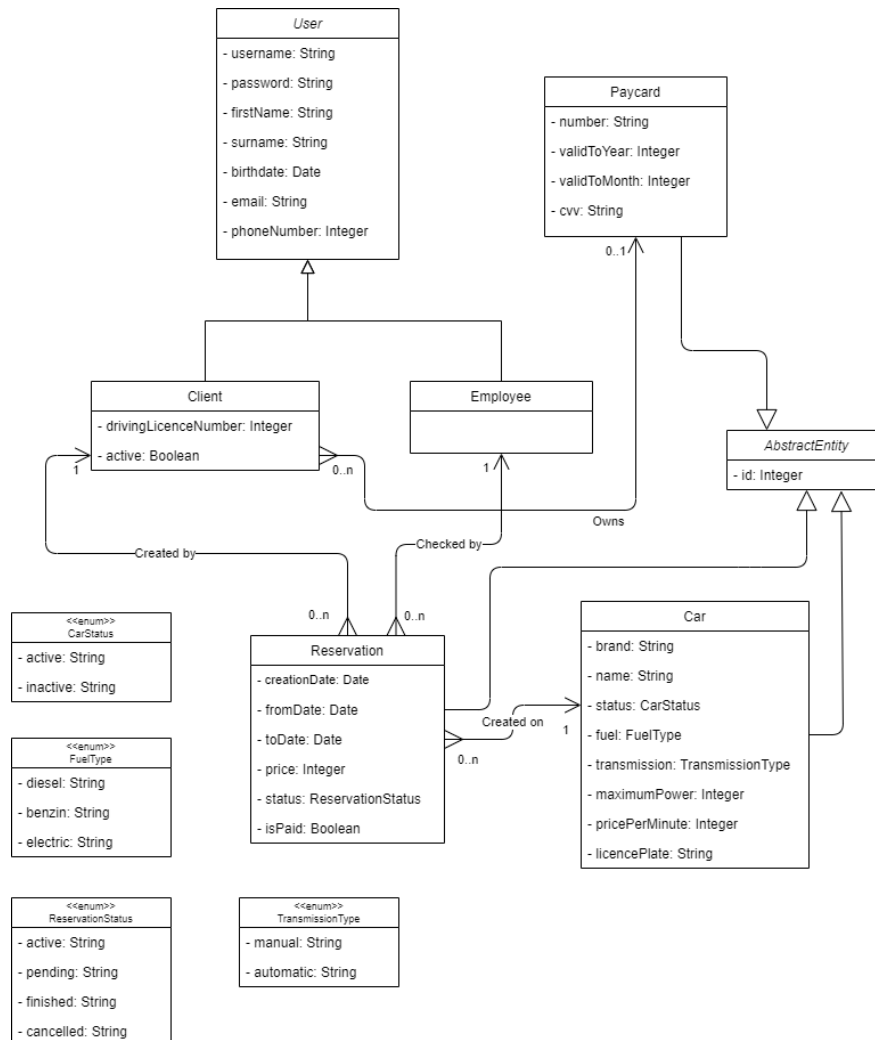
Protože se mezi serverem a klientem budou posílat různá data, je důležité, aby při posílání či příjmu dat serverem nedošlo k neočekávaným a nepožadovaným poškozením, ztrátám, či změnám datových objektů.

- **Bezpečnost**

Vzhledem k tomu, že aplikace bude pracovat s citlivými daty, server musí zamezit nebezpečným útokům SQL Injection a Cross-Site Request Forgery (CSRF).

## 2.4. Diagram tříd

Na základě funkčních požadavků bylo rozhodnuto navrhnout konceptuální model, který znázorňuje, v jaké podobě budou zpracovávány data na serveru. Viz. Obrázek 2.1.

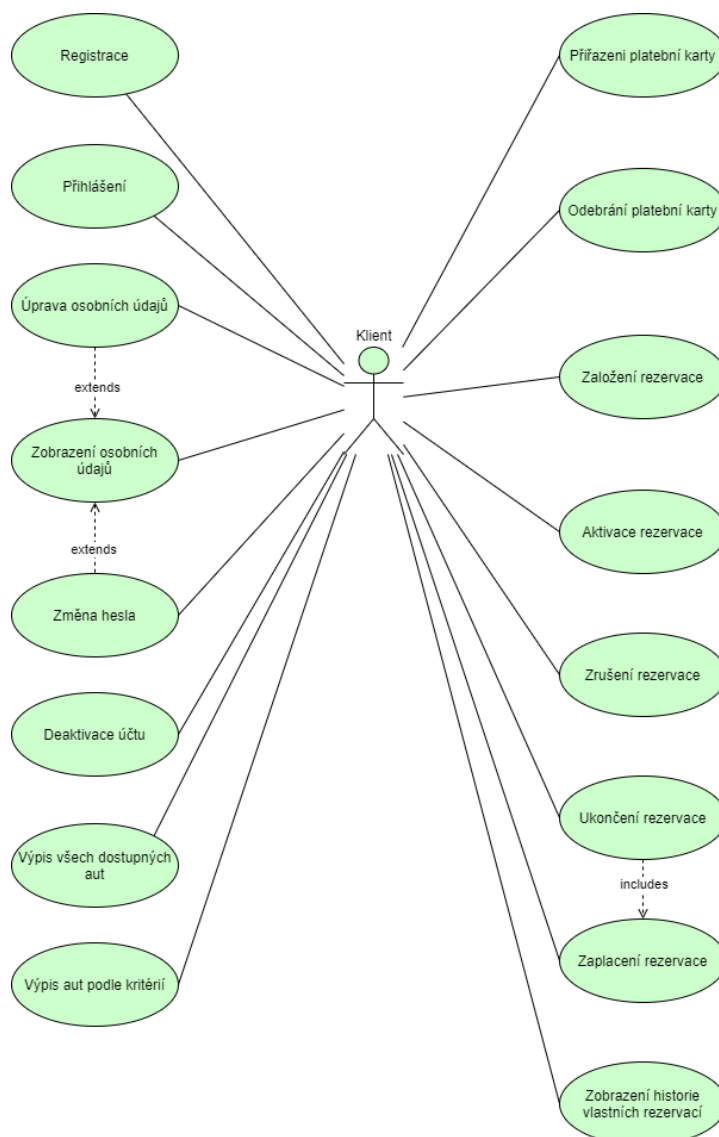


Obrázek 2.1

## 2.5. Diagramy případů užití (Use Case Diagram)

Znázorňují akce, jenž uživatel může vykonat v rámci dané aplikace. V daném případě diagramy vizualizují výše popsané funkční požadavky. Viz. Obrázek 2.2: Případy užití - Zákazník, Obrázek 2.3: Případy užití - Zaměstnanec, Čas.

### Zákazník



Obrázek 2.2: Případy užití - Zákazník

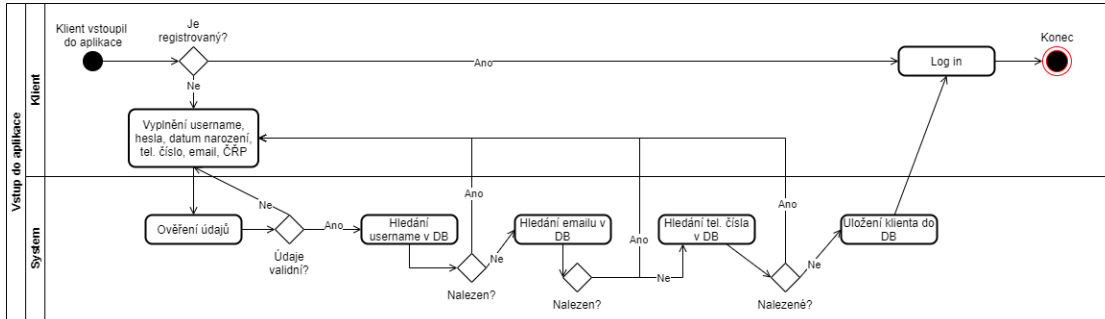
## Zaměstnanec, Čas



Obrázek 2.3: Případy užití - Zaměstnanec, Čas

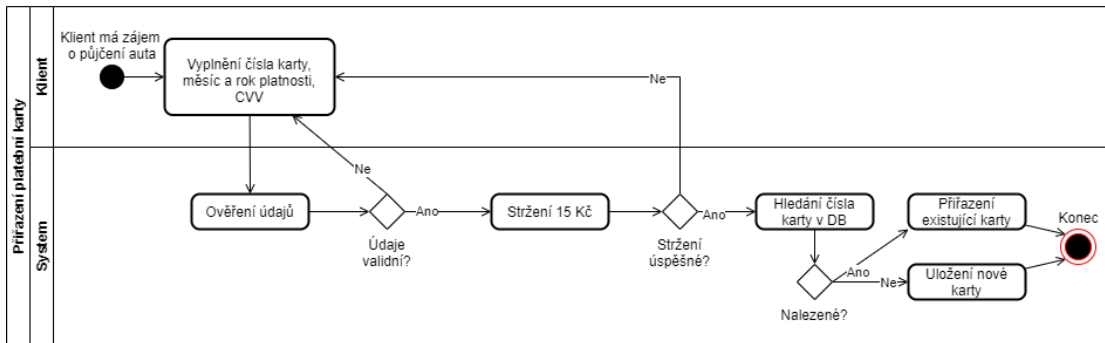
## 2.6. Model hlavních procesů

### Vstup do systému



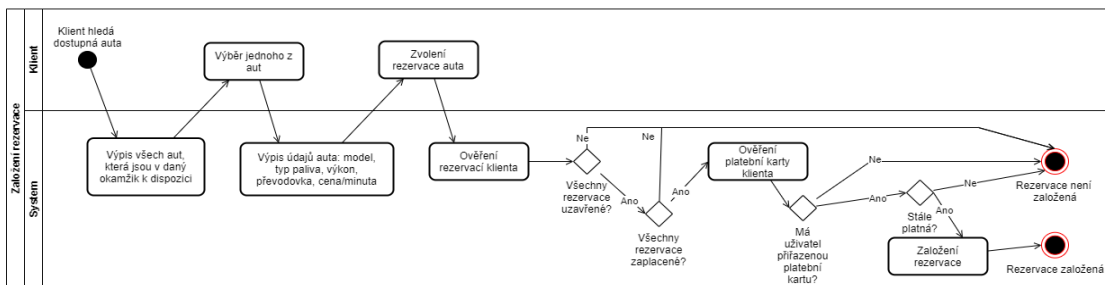
Obrázek 2.4: Model procesu vstupu do systému

### Připojení (přiřazení) platební karty k uživateli



Obrázek 2.5: Model procesu připojení platební karty k uživateli

### Založení rezervace

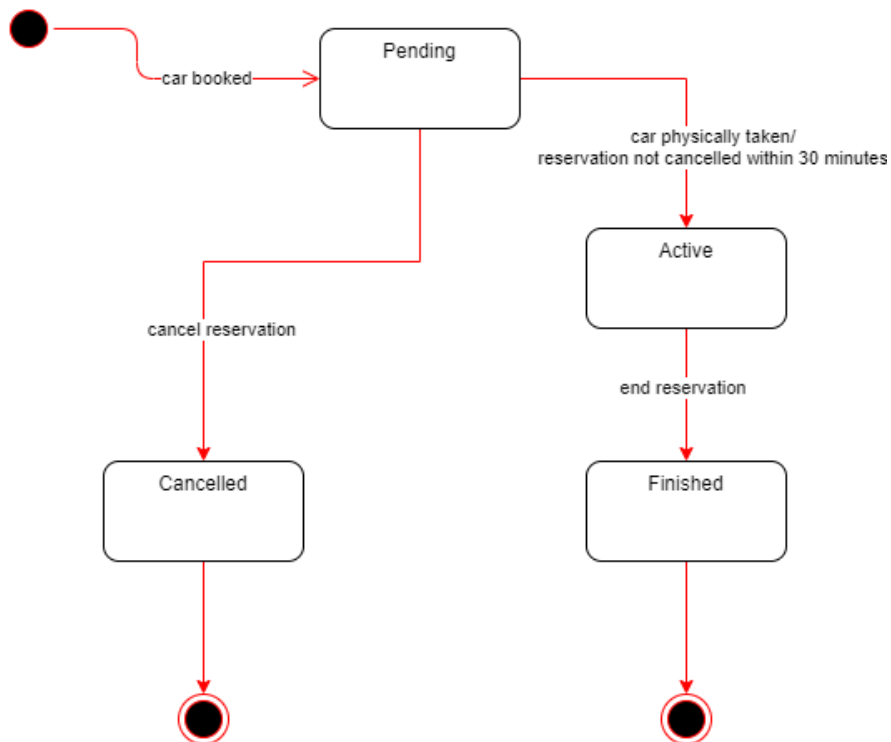


Obrázek 2.6: Model procesu založení rezervace

## 2.7. Stavové diagramy

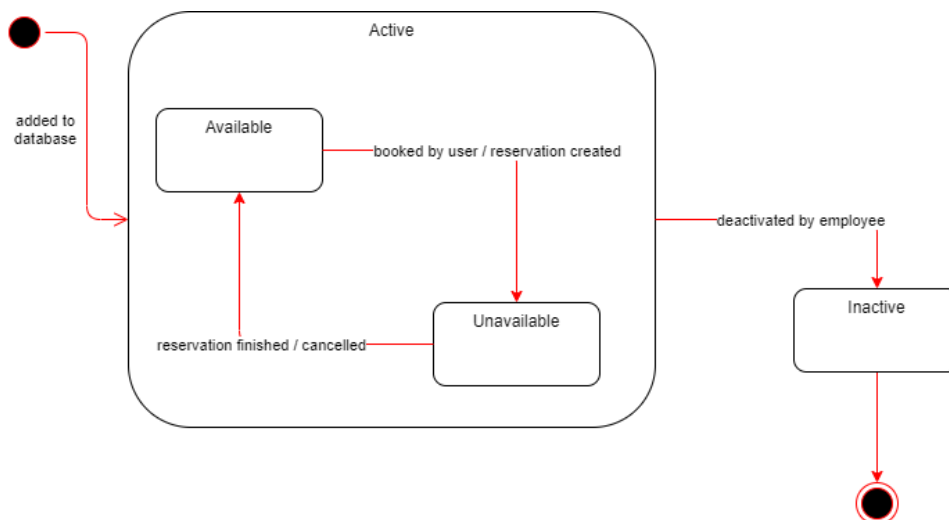
Stavové diagramy slouží ke grafickému znázornění životních cyklů objektů. Viz. Obrázek 2.7: Rezervace - stavy a přechody mezi nimi, Obrázek 2.8: Auto - stavy a přechody mezi nimi.

### Rezervace



Obrázek 2.7: Rezervace - stavy a přechody mezi nimi

### Auto



Obrázek 2.8: Auto - stavy a přechody mezi nimi



## 3. Použité technologie

### 3.1. Úložiště dat

Pro uchovávání dat byla zvolena **relační databáze**. Jedná se o model, který existuje na trhu již od 80. let a je používaný organizacemi všech typů pro širokou škálu informačních potřeb. Relaçní databáze může být zvažována pro jakoukoliv potřebu informací, ve kterých se záznamy vztahují k sobě navzájem a musí být řízeny konzistentním způsobem.[1]

**Hlavní výhodou** relační databáze je skutečnost, že její transakce definují čtyři základní vlastnosti – atomicita, konzistence, izolovanost a odolnost (ACID). **Atomicita (atomicity)** označuje skutečnost, že transakce se buď provede celá, nebo se neprovede vůbec. Pokud nějaká část transakce selže, databáze se vrátí do stavu, ve kterém byla před začátkem transakce. **Konzistence (consistency)** znamená že transakce nenarušují integritu dat a podmínky, které byly definovány na úrovni dané databáze. Databáze je vždy v konzistentním stavu. **Izolovanost (isolation)** označuje vlastnost, při níž vnitřní operace transakce jsou izolovány od vnějších zásahů. Souběžně probíhající transakce se navzájem neovlivňují a jsou vykonávány postupně.[2] Efekty transakce nejsou viditelné, dokud transakce není provedená. **Odolnost (durability)** znamená že změny, které nastanou po provedení transakce, jsou trvale uloženy a nemůže dojít k jejich ztrátám.

Pro uplatnění relačního databázového modelu byl zvolen systém **MySQL**. Jedná se o nejvíce používaný databázový systém, v němž komunikace probíhá pomocí jazyka SQL a který je dostupný téměř na všech typech webhostingu.[2] Jeho hlavní výhodou je skutečnost, že je k dispozici jako open source, tedy je dostupný zdarma. Navíc se jedná o multiplatformní databázi, která je podporovaná na všech počítačových platformách. Hlavním nedostatkem systému MySQL je skutečnost, že nedisponuje dostatečným výkonem v zatěžovaných webových aplikacích.[3]

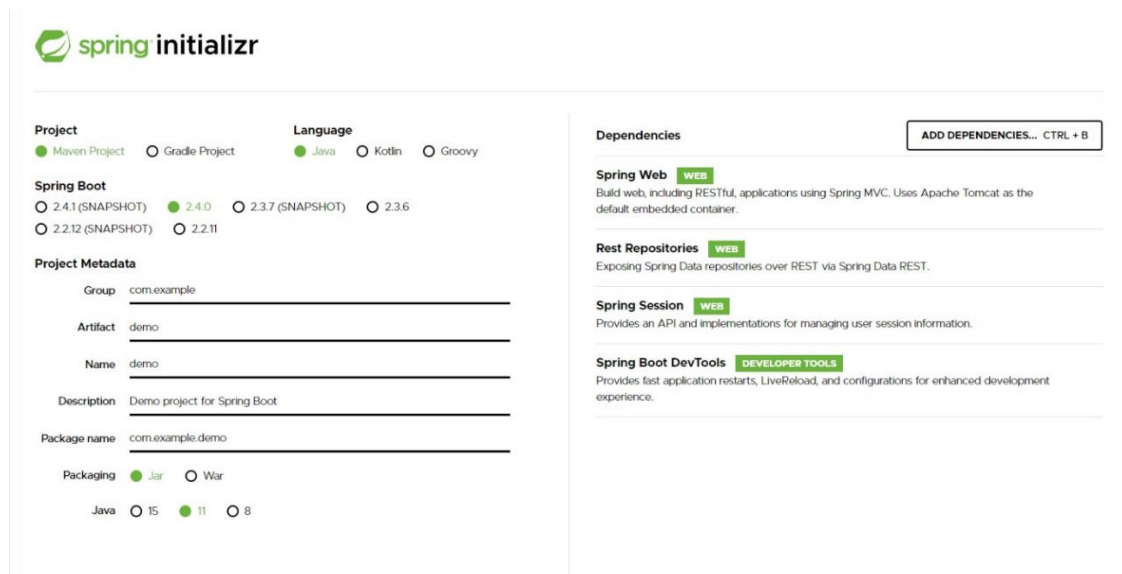
Relaçní databáze pro tuto aplikaci byla vytvořena na základě diagramu tříd, viz. Obrázek 2.4.

### 3.2. Framework

Pro vývoj serveru byl použit open-source framework **Spring**. Jedná se o nejpopulárnější framework pro vývoj aplikací v jazyce Java. Spring poskytuje API, umožňující implementaci technologií JDBC i Java Persistence API pro přístup k datům v databázi. Spring umožňuje vyvíjet aplikace, využívající tzv. Plain Old Java Objects (POJOs), tudíž není potřeba implementace Enterprise JavaBeans Container, jehož vývoj trvá déle, je složitější a komplikovanější a zachytit chyby je více problematické, jelikož chyba může být nejen v mém kódu, ale i na straně samotného containeru.[5] Mezi technologie, které Spring poskytuje k využití, patří Object Relational Mapping, Jakarta Enterprise Edition a Java Development Kit. Framework také implementuje tzv.

Dependency Injection, což je jedna z technik **Inversion of Control (IoC)**, umožňující vkládání závislosti mezi jednotlivými komponentami, aniž by objekt musel zodpovídat za životní cyklus komponentu; za životní cyklus komponentu zodpovídá tzv. **Spring IoC Container**. Testování funkcionalit implementovaných v aplikaci Spring pomocí JUnit zjednodušuje možnost použití Dependency Injection pro vložení testovacích dat.[4]

Tento framework byl použit prostřednictvím projektu **Spring Boot**, jenž poskytuje automatickou konfiguraci Spring a knihoven třetích stran.[6] Spring Boot projekt se všemi potřebnými závislostmi je možné vygenerovat na stránce start.spring.io, viz. Obrázek 3.1.

The image shows the Spring Initializr web interface. It features a header with the Spring logo and the text 'spring inicializr'. Below the header, there are several configuration sections: 'Project' with radio buttons for 'Maven Project' (selected) and 'Gradle Project'; 'Language' with radio buttons for 'Java' (selected), 'Kotlin', and 'Groovy'; 'Spring Boot' with radio buttons for versions 2.4.1 (SNAPSHOT), 2.4.0 (selected), 2.3.7 (SNAPSHOT), 2.3.6, 2.2.12 (SNAPSHOT), and 2.2.11; 'Project Metadata' with input fields for 'Group' (com.example), 'Artifact' (demo), 'Name' (demo), 'Description' (Demo project for Spring Boot), and 'Package name' (com.example.demo); 'Packaging' with radio buttons for 'Jar' (selected) and 'War'; and 'Language' with radio buttons for 'Java' (selected), '15', '11' (selected), and '8'. On the right side, there is a 'Dependencies' section with a button 'ADD DEPENDENCIES... CTRL + B'. Below this, there are four dependency cards: 'Spring Web' (WEB), 'Rest Repositories' (WEB), 'Spring Session' (WEB), and 'Spring Boot DevTools' (DEVELOPER TOOLS), each with a brief description.

Obrázek 3.1

### 3.3. Architektura serveru

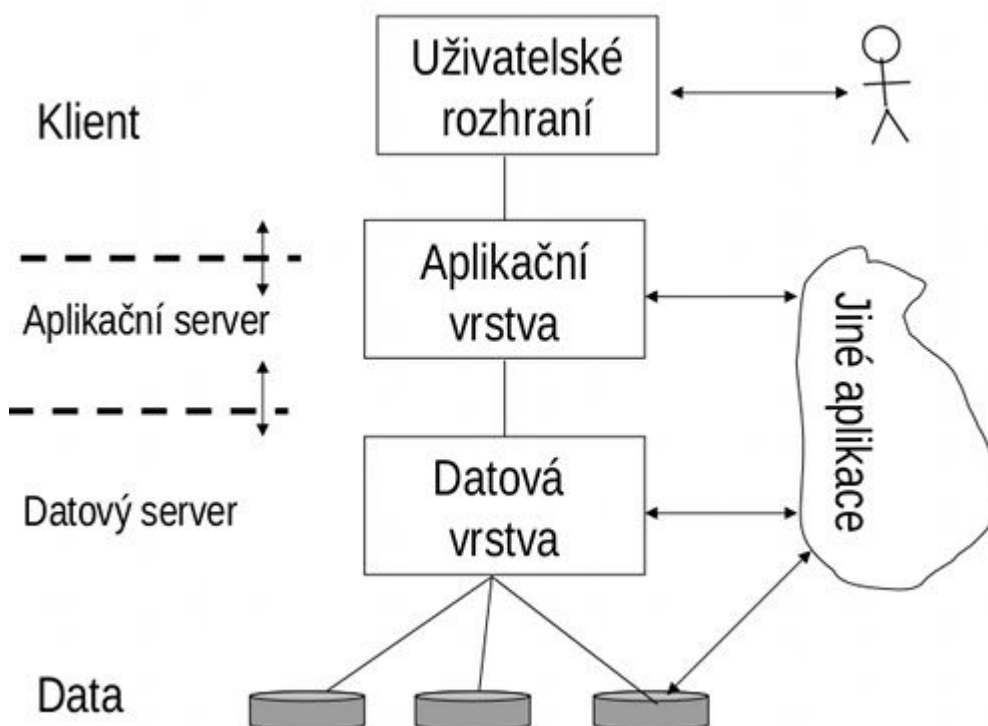
Pro vývoj serveru jsem zvolil **třívrstvou architekturu**. Využívá ji většina podnikových aplikací. Je vhodná pro strukturizaci kódu v rámci celé aplikace. Ve třívrstvé architektuře je závislost mezi vrstvami na takové úrovni, která umožňuje provádět záměny implementace jedné z vrstev, aniž by to ovlivnilo vrstvy na vyšší úrovni. Díky oddělení logiky od databázových funkcí třívrstvá architektura poskytuje vylepšenou škálovatelnost a lepší vyrovnání zátěže. Slabou stránkou architektury je skutečnost, že během předávání mezi vrstvami datové entity prochází transformací, což vyžaduje určitý výpočetní výkon a zpomalí běh aplikace.[7][10]

Kód je rozložen do tří na sobě nezávislých vrstev. První vrstva, **datová (Data Access Object Layer)**, poskytuje rozhraní pro přístup k datům v databázi, aniž by odhaloval vlastnosti databáze samotné. Tato vrstva odděluje relační databázi od zbytku aplikace/business logiky. Skládá se ze 3 částí: **Interface** definuje názvy operací, které budou prováděny na objektech (vždy by měla zahrnovat standardní operace get, getAll, persist, update, remove); **Concrete Class** předepisuje chování operací, které budou prováděny na konkrétním objektu; **Model Object** "mapuje" tabulku v DB, s níž budou prováděny operace.[8][9]

**Aplikační vrstva (Business Logic Layer, také Business vrstva)** obsahuje základní funkcionality aplikace. Jedná se o “mozek” aplikace, protože obsahuje veškerou logiku a operace, které slouží k realizaci výše definovaných funkčních požadavků kladených na server.

Poslední vrstvou je **prezentační vrstva (Presentation Layer)**, jenž zajišťuje jednak vstup požadavků odeslaných stranou klientskou, jednak prezentaci odpovědi na přijatý požadavek klientské straně.[7] Komunikaci s frontendem a přístup k funkcionalitám poskytuje tzv. **REST** rozhraní. Každá funkce má unikátní **Uniform Resource Identifier** (dále **URI**), na který frontend HTTP požadavek posílá. Po zpracování požadavku je na frontend odeslaná odpověď v podobě HTTP response.

Vrstvy jsou propojeny takovým způsobem, že při zpracování požadavku prezentační vrstva volá vrstvu aplikační, ta zas volá vrstvu datovou, která přistupuje k úložišti dat. Viz. Obrázek 3.2.



Obrázek 3.2: Grafické znázornění 3-vrstvé architektury serverové strany a přístupu k databázi[13]

### 3.4. Autentizace a autorizace

Vzhledem k tomu že server je implementován pro aplikaci desktopovou, přihlašování a přidělení práv funguje na principu **Basic Auth**.

Server zpracuje přihlašovací údaje, které mu poslal klient. Údaje se skládají z uživatelského jména a hesla, které jsou přijaty v podobě textového řetězce. Pokud existuje objekt, jemuž odpovídá dané uživatelské jméno, server porovná zašifrované heslo, náležící danému objektu, s heslem obdržným z klientské strany. Pokud hesla si navzájem odpovídají, server vytvoří

objekt typu **AuthenticationToken**, do kterého vloží objekt s přidělenými právy, napřed však skryje heslo. Aby přihlášený zákazník měl přístup k funkcím, k nimž by měl mít možnost přistupovat na základě práv, je vytvořen **bezpečnostní kontext**, jenž obsahuje objekt **AuthenticationToken** a je uchováván v úložišti daného vlákna.[11]

Basic Auth je výhodný tím, že je podporován všemi prohlížeči a je snadný na implementaci. Nevýhodou Basic Auth je skutečnost, že nezabezpečuje odposlech hesla při posílání údajů mezi klientem a serverem.

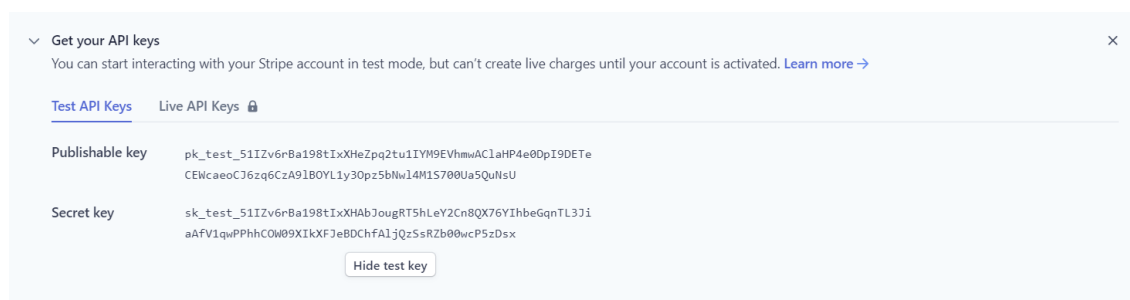
### 3.5. Platební brána

Pro platební bránu byla použita knihovna **J2Pay**. Jedná se o knihovnu v Javě, který poskytuje API k zpracování plateb. Umožňuje flexibilitu k implementaci univerzálního kódu pro všechny platební brány.

API poskytuje tzv. **GatewayFactory**, která slouží k vytvoření objektu platební brány. GatewayFactory umožňuje přijímat platby přes platební brány Stripe, NMI, Authorize, PayFlowPro, EasyPay, BillPro, BrainTree, Checkout a PayEezy. Vývojář se nemusí zabývat implementací metod pro přijímání plateb, objekt platební brány již obsahuje připravené funkce – **purchase, rebill a refund**.

Pro tuto aplikaci bylo rozhodnuto přijímat platby na účet platební brány **Stripe**. Platba přes tuto platební bránu vyžaduje údaje o uživateli, údaje o platební kartě, skrytý API klíč (jehož vložení je předepsáno vývojářem a jenž není viditelný pro běžného uživatele), měnu a částku. Stripe také poskytuje tzv. testovací API key, vhodný pro použití při testování funkčnosti platební brány, viz. Obrázek 3.3.

Údaje o uživateli zahrnují křestní jméno a příjmení, stát, město, ulici, poštovní směrovací číslo (dále jen PSČ), telefonní číslo, email a IP adresa místa, ze kterého je proces spuštěn. Údaje o platební kartě zahrnují 16-místné číslo platební karty, měsíc a rok platnosti a CVV. Měna je určená vývojářem, v daném případě je to **česká koruna (Kč)**.



Obrázek 3.3: Platební brána Stripe poskytuje i tzv. testovací API klíč pro otestování funkčnosti platební brány

Účet Stripe poskytuje přehled o platbách, zpracovávaným naším účtem. Jedná se jak o platby v testovacím režimu, tak i o platby v režimu reálného života, viz. Obrázek 3.4.

The screenshot shows the Stripe Payments interface. On the left is a navigation menu with options like Home, Payments, Balances, Customers, etc. The main area displays a table of payments. The table has columns for AMOUNT, DESCRIPTION, CUSTOMER, and DATE. All payments listed are for 15.00 CZK and are marked as 'Succeeded'.

AMOUNT	DESCRIPTION	CUSTOMER	DATE
Kč15.00 CZK	ch_11f4008a19811x0Hf6taw8	cus_11PQkxP86j00Qn	Apr 13, 5:56 PM
Kč15.00 CZK	ch_11f4008a19811x0Hf6taw8	cus_11PQkxP86j00Qn	Apr 13, 5:54 PM
Kč15.00 CZK	ch_11f4008a19811x0Hf6taw8	cus_11PQkxP86j00Qn	Apr 13, 5:42 PM
Kč15.00 CZK	ch_11f4008a19811x0Hf6taw8	cus_11PQkxP86j00Qn	Apr 13, 5:41 PM
Kč15.00 CZK	ch_11f4008a19811x0Hf6taw8	cus_11PQkxP86j00Qn	Apr 13, 5:40 PM
Kč15.00 CZK	ch_11f4008a19811x0Hf6taw8	cus_11PQkxP86j00Qn	Apr 13, 1:37 PM
Kč15.00 CZK	ch_11f4008a19811x0Hf6taw8	cus_11PQkxP86j00Qn	Apr 13, 1:32 PM
Kč15.00 CZK	ch_11f4008a19811x0Hf6taw8	cus_11PQkxP86j00Qn	Apr 13, 1:28 PM
Kč15.00 CZK	ch_11f4008a19811x0Hf6taw8	cus_11PQkxP86j00Qn	Apr 13, 1:25 PM
Kč15.00 CZK	ch_11f4008a19811x0Hf6taw8	cus_11PQkxP86j00Qn	Apr 13, 10:42 AM
Kč15.00 CZK	ch_11f4008a19811x0Hf6taw8	cus_11PQkxP86j00Qn	Apr 13, 10:41 AM
Kč15.00 CZK	ch_11f4008a19811x0Hf6taw8	cus_11PQkxP86j00Qn	Apr 7, 3:59 PM

Obrázek 3.4: Účet Stripe poskytuje přehled o všech platbách, zpracovaných účtem jak v režimu testovacím, tak i v režimu reálného života

U každé platby je uveden datum, částka, odesílatel, způsob platby a zpráva o tom, jestli platba proběhla úspěšně. Položka Payment details obsahuje informace o příjemci, přijaté částce a zpráva o tom, jestli platba proběhla úspěšně. Položka Payment method obsahuje identifikační číslo (dále jen ID) transakce a údaje o platbě – skryté číslo platební karty, měsíc a rok platnosti této karty, typ karty, ověření CVV a adresa, ze které platba byla spuštěna. Viz. Obrázek 3.5.

The screenshot shows the Stripe Payment details page for a specific payment. The payment is for 15.00 CZK and is marked as 'Succeeded'. The page is divided into sections: Payment details, Timeline, and Payment method.

Payment details
Statement descriptor: ARTUR SARHSJAN
Amount: Kč15.00
Fee: Kč0.94
Net: Kč8.06
Status: Succeeded
Description: No description

Payment method	
ID: card_11f4008a19811x0Hf6taw8	Address: CZ
Number: **** 4444	Origin: United States
Fingerprint: RfG4ZjQRkbyyBf	CVC check: Passed
Expires: 07 / 2022	
Type: Mastercard credit card	
Issuer: CIAGROUP	

Obrázek 3.5.

## 3.6. Testování

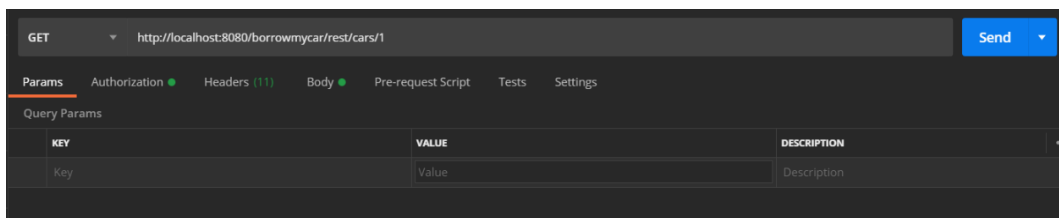
Testování aplikace bude probíhat dvěma způsoby:

- **Java Unit Tests**

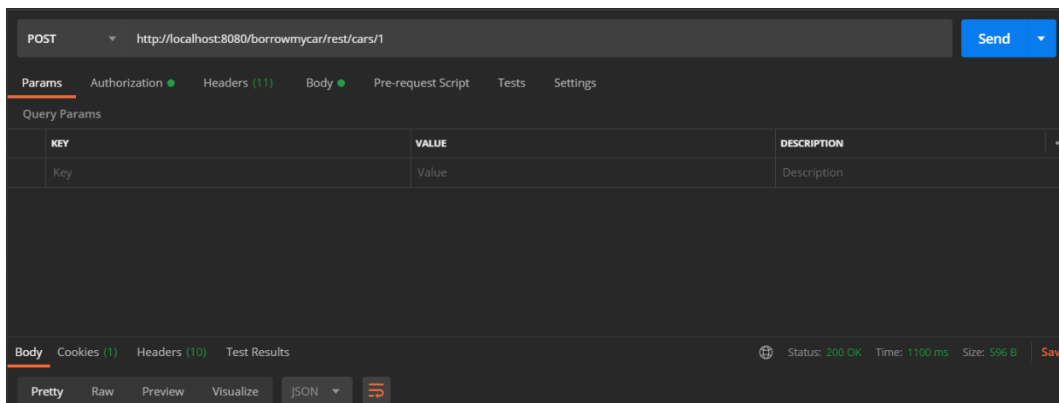
- Slouží k testování a ověřování správného fungování jednotlivých funkcionalit a komponent systému
- Jednotlivé testy jsou nezávislé na ostatních
- Vytvářejí se pomocné objekty, jenž slouží k simulaci daného procesu
- Po ukončení testu je systém zachován ve stavu, v jakém byl před začátkem testu
- V případě frameworku Spring se k testování používají Dependency Injection a také Mock objekty

- **Postman**

- Umožňuje testovat přístup k datům přes jejich unikátní URI
- Umožňuje testovat autorizační omezení přístupu k datům



Obrázek 3.6: Ukázka využití Postmanu. Krok 1: vložíme URL funkcionality, cestu k němuž definujeme v RESTu, a definujeme typ metody pro přístup k datům (GET/POST/PUT/DELETE)



Obrázek 3.7: Ukázka využití Postmanu. Krok 2. V případě metod GET, POST, PUT a DELETE nám Postman musí vrátit kladnou odpověď (200 OK - 299 OK). Pokud osoba s danými přihlašovacími údaji nemá přístup k dané metodě, Postman vrátí odpověď 401 Unauthorized.

## 4. Testování

Realistické scénáře, které mohou nastat. Jsou pokryty funkcionality, které do databáze ukládají nové údaje nebo v databázi upravují údaje stávající.

Názvy vstupních polí, začínající velkým písmenem, jsou hodnoty primitivního datového typu. Číslování některých scénářů není posloupné z důvodů, že z důvodů přehlednosti vytvářené testy obsahovaly v názvu číslo, odpovídající číslu scénáře v tabulce. Testy byly vytvářeny souběžně s vytvářením scénářů, a změna posloupnosti těchto scénářů při změně testovaných vstupních hodnot by vyžadovala i změnu názvu vytvořených testů, což s narůstajícím počtem testů se stalo velmi náročným.

Vstupní hodnoty polí a odpovídající testy naleznete v příloze, viz. **Testování vstupů**.

### 4.1. Testování vstupů

- **Registrace – založení zákazníka**

**Vstupní pole:**

- uživatelské jméno (UJ)
- heslo (PSW)
- křestní jméno (KJ)
- příjmení (PŘ)
- datum narození (DN)
- email (E)
- telefonní číslo (TEL)
- číslo řidičského průkazu (ČŘP)

**Použitá technika:**

- MCC testing

- **Registrace – založení zaměstnance**

**Vstupní pole:**

- uživatelské jméno (UJ)
- heslo (PSW)
- křestní jméno (KJ)
- příjmení (PŘ)
- datum narození (DN)
- email (E)
- telefonní číslo (TEL)

**Použitá technika:**

- MCC testing

- **Přihlášení**

**Vstupní pole:**

- uživatelské jméno
- heslo
- uživatelský status

**Použitá technika:**

- MCC testing

- **Změna hesla**

**Vstupní pole:**

- Uživatel
- staré heslo
- nové heslo

**Použitá technika:**

- MCC testing

- **Úprava údajů zákazníka**

**Vstupní pole:**

- Zákazník (Z)
- křestní jméno (KJ)
- příjmení (PŘ)
- datum narození (DN)
- email (E)
- telefonní číslo (TEL)
- číslo řidičského průkazu (ČŘP)

**Použitá technika:**

- MCC testing



- **Úprava údajů zaměstnance**

**Vstupní pole:**

- Zákazník (Z)
- křestní jméno (KJ)
- příjmení (PŘ)
- datum narození (DN)
- email (E)
- telefonní číslo (TEL)
- číslo řidičského průkazu (ČŘP)

**Použitá technika:**

- MCC testing

- **Založení auta, úprava údajů auta**

*Poznámka: Id je hodnota, kterou nebude moct ukládat ani upravit nikdo, proto v případě Id se kontrola neprovádí.*

**Vstupní pole:**

- Zaměstnanec (Z)
- model (brand; B)
- jméno (name; N)
- status (S)
- typ pohonné hmoty (fuel; F)
- typ převodovky (transmission; T)
- maximální výkon (max. power; MP)
- cena/min (price per minute; PPM)
- SPZ (licence plate; LP)
- zeměpisná šířka (ZŠ)
- zeměpisná délka (ZD)

**Použitá technika:**

- MCC testing

- **Založení platební karty**

*Poznámka: Id je hodnota, kterou nebude moct ukládat ani upravit nikdo, proto v případě Id se kontrola neprovádí.*

**Vstupní pole:**

- Zákazník
- číslo karty
- validita do: měsíc
- validita do: rok
- CVV/CVC kód

**Použitá technika:**

- MCC testing

- **Úprava údajů o platební kartě**

*Poznámka: Id je hodnota, kterou nebude moct ukládat ani upravit nikdo, proto v případě Id se kontrola neprovádí.*

**Vstupní pole:**

- číslo karty
- validita do: měsíc
- validita do: rok
- CVV/CVC kód

**Použitá technika:**

- MCC testing

- **Založení rezervace**

*Poznámka: Id je hodnota, kterou nebude moct ukládat ani upravit nikdo, proto v případě Id se kontrola neprovádí.*

**Vstupní pole:**

- Rezervace zákazníka
- Platební karta zákazníka
- Zaměstnanec
- Auto

**Použitá technika:**

- MCC testing

- **Zaplacení rezervace**

*Poznámka: Id je hodnota, kterou nebude moct ukládat ani upravit nikdo, proto v případě Id se kontrola neprovádí.*

*Poznámka 2: vycházíme z toho, že uživatel bez platební karty nemůže rezervaci založit, a také z toho, že uživatel, jenž má neuzavřenou či dosud nezaplacenou rezervaci, nemůže kartu odebrat*

**Vstupní pole:**

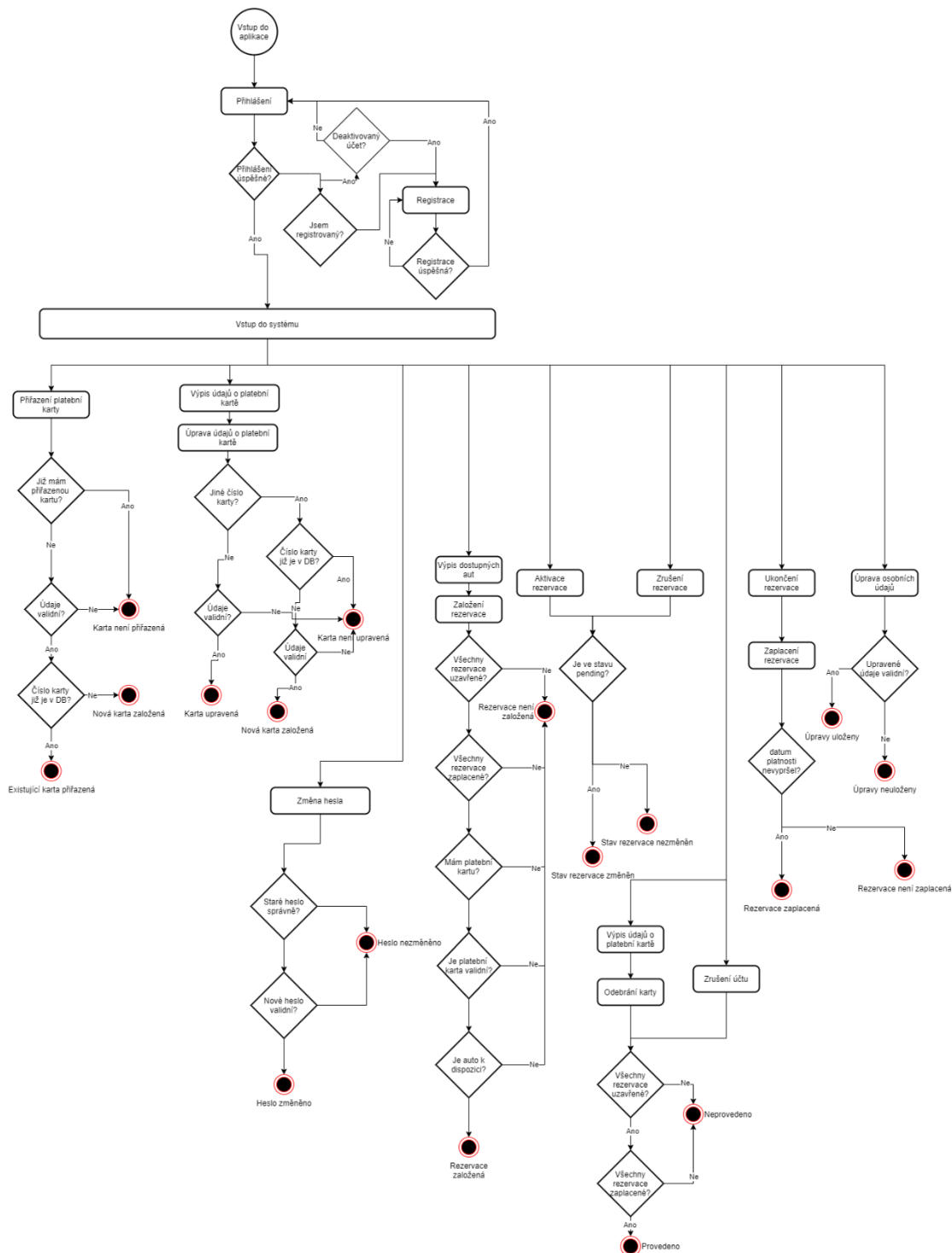
- Rezervace
- Platební karta zákazníka

**Použitá technika:**

- MCC testing

## 4.2. Průchody procesem - zákazník

Jsou zahrnuty pouze průchody, jejichž ukončením je rozhodnuto o uložení záznamů či úpravě záznamů existujících.



Obrázek 4.1: Průchody procesem zákazníkem

- **Větvící body**

Tabulka 1: Průchody procesem - větvící body

<b>Větvící body</b>	
Vstup do aplikace	
přihlášení úspěšné	PÚ
jsem registrovaný	JS
registrace úspěšná	RÚ
deaktivovaný účet	DÚ
již mám přiřazenou kartu	PK
údaje platební karty validní	ÚPKV
číslo karty již je v DB	ČKDB
jiné číslo karty	JČK
všechny rezervace uzavřené	VRU
všechny rezervace zaplacené	VRZ
mám platební kartu	MPK
platební karta validní	PKV
datum platnosti platební karty	DPPK
osobní údaje validní	OÚV
auto k dispozici	AkD
rezervace je ve stavu pending	RjP
staré heslo správně	SHS
nové heslo validní	NHV
Karta není přiřazená	
Nová karta založená	
Heslo změněno	
Heslo nezměněno	
Existující karta přiřazená	
Karta není upravena	
Karta upravena	
Rezervace není založená	
Rezervace založená	
Stav rezervace změněn	
Stav rezervace nezměněn	
Rezervace zaplacená	
Rezervace není zaplacená	

Větvící body
Osobní údaje uloženy
Osobní údaje neuloženy
Provedeno
Neprovedeno

- **Možné akce**

Tabulka 2: Průchody procesem - možné přechody mezi větvícími body

Možné akce		
	Z bodu	Do bodu
1	Vstup do aplikace	PÚ
2	PÚ	JR
3	JR	DÚ
4	JR	RÚ
5	RÚ	PÚ
6	RÚ	RÚ
7	PÚ	PK
8	PK	Karta není přiřazená
9	PK	ÚPKV
10	ÚPKV	Karta není přiřazená
11	ÚPKV	ČKDB
12	ČKDB	Nová karta založená
13	ČKDB	Existující karta přiřazená
14	PÚ	JČK
15	JČK	ČKDB
16	JČK	ÚPKV
17	ČKDB	Karta není upravena
18	ČKDB	ÚPKV
19	ÚPKV	Nová karta založená
20	ÚPKV	Karta není upravena
21	ÚPKV	Karta upravena
22	PÚ	VRU
23	VRU	Rezervace není založená

<b>Možné akce</b>		
	<b>Z bodu</b>	<b>Do bodu</b>
24	VRU	VRZ
25	VRZ	Rezervace není založená
26	VRZ	MPK
27	MPK	Rezervace není založená
28	MPK	PKV
29	PKV	Rezervace není založená
30	PKV	AkD
31	AkD	Rezervace není založená
32	AkD	Rezervace založená
33(AR)	PÚ	RjP
34(AR)	RjP	Stav rezervace změněn
35(AR)	RjP	Stav rezervace nezměněn
36(ZR)	PÚ	RjP
37(ZR)	RjP	Stav rezervace změněn
38(ZR)	RjP	Stav rezervace nezměněn
39	PÚ	DPPK
40	DPPK	Rezervace zaplacená
41	DPPK	Rezervace není zaplacená
42	PÚ	OÚV
43	OÚV	Osobní údaje uloženy
44	OÚV	Osobní údaje neuloženy
45(OK)	PÚ	VRU
46(OK)	VRU	Neprovedeno
47(OK)	VRU	VRZ
48(OK)	VRZ	Neprovedeno
49(OK)	VRZ	Provedeno
50(ZÚ)	PÚ	VRU
51(ZÚ)	VRU	Neprovedeno
52(ZÚ)	VRU	VRZ
53(ZÚ)	VRZ	Neprovedeno
54(ZÚ)	VRZ	Provedeno
55	DÚ	PÚ
56	DÚ	JR

Možné akce		
	Z bodu	Do bodu
57	PÚ	SHS
58	SHS	Heslo nezměněno
59	SHS	NHV
60	NHV	Heslo změněno
61	NHV	Heslo nezměněno

- **Vstupy a výstupy jednotlivých větvících bodů**

Tabulka 3: Průchody procesem - vstupy a výstupy jednotlivých větvících bodů

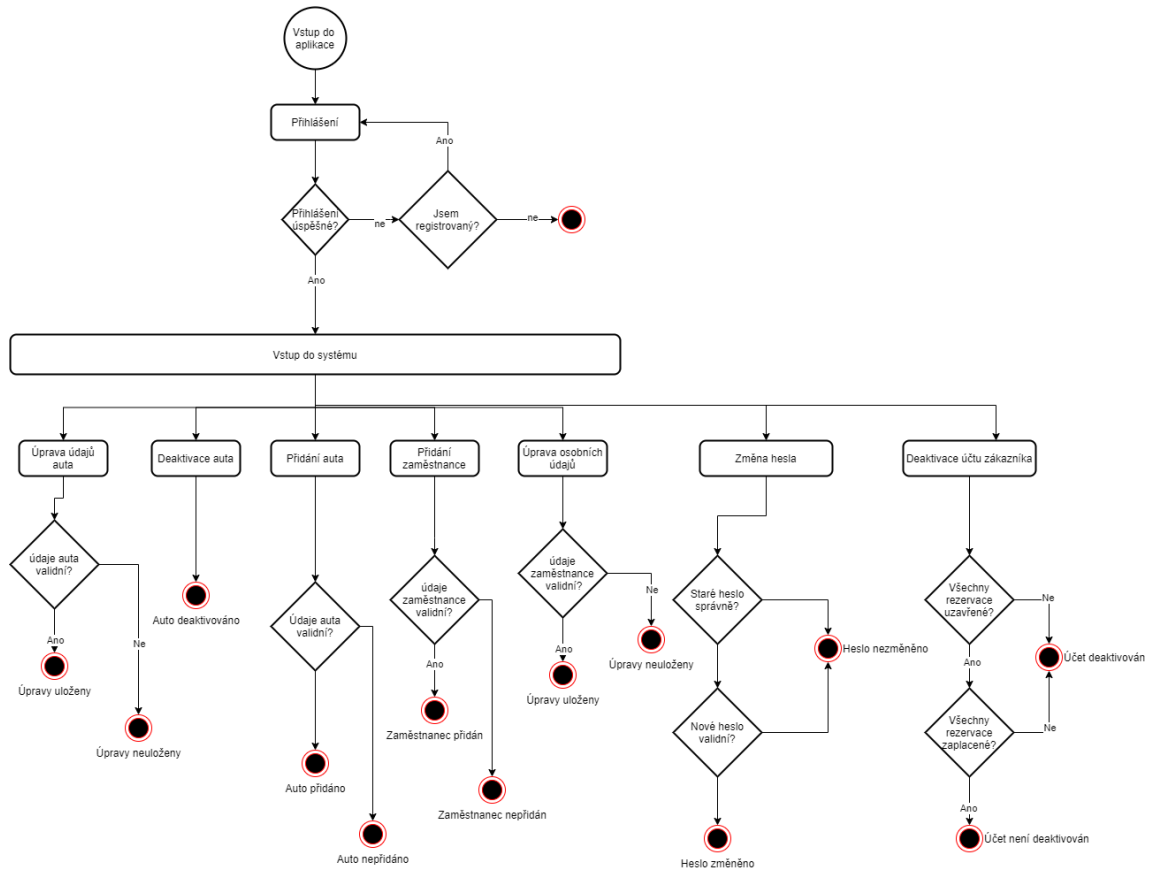
Vstupy a výstupy jednotlivých větvících bodů		
Větvící body	Vstupy	Výstupy
Vstup do aplikace		1
PÚ	1, 5, 55	2, 7, 14, 22, 33(AR), 36(ZR), 39, 42, 45(OK), 50 (ZÚ), 57
JR	2, 56	3, 4
RÚ	4, 6	5, 6
PK	7	8, 9
ÚPKV	9, 16, 18	10, 11, 19, 20, 21
ČKDB	11, 15	12, 13, 17, 18
JČK	14	15, 16
VRU	22, 45(OK), 50 (ZÚ)	23, 24, 46(OK), 47(OK), 51(ZÚ), 52(ZÚ)
VRZ	24, 47(OK), 52(ZÚ)	25, 26, 48(OK), 49(OK), 53(ZÚ), 54(ZÚ)
MPK	26	27, 28
PKV	28	29, 30
AkD	30	31, 32
RjP	33(AR), 36(ZR)	34(AR), 35(AR), 37(ZR), 38(ZR)
DPPK	39	40, 41
OÚV	42	43, 44
DÚ	3	55, 56
SHS	57	58, 59
NHV	59	60, 61
Karta není přiřazená	8, 10	
Nová karta založená	12, 19	
Existující karta přiřazená	13	
Karta není upravena	17, 20	
Karta upravena	21	



<b>Vstupy a výstupy jednotlivých větvících bodů</b>		
<b>Větvící body</b>	<b>Vstupy</b>	<b>Výstupy</b>
Rezervace není založená	23, 25, 27, 29, 31	
Rezervace založená	32	
Stav rezervace změněn	34(AR), 37(ZR)	
Stav rezervace nezměněn	35(AR), 38(ZR)	
Rezervace zaplacená	40	
Rezervace není zaplacená	41	
Osobní údaje uloženy	43	
Osobní údaje neuloženy	44	
Provedeno	49(OK), 54(ZÚ)	
Neprovedeno	46(OK), 48(OK), 51(ZÚ), 53(ZÚ)	
Heslo změněno	60	
Heslo nezměněno	58, 61	

### 4.3. Průchody procesem – zaměstnanec

Jsou zahrnuty pouze průchody, jejichž ukončením je rozhodnuto o uložení záznamů či úpravě záznamů existujících.



Obrázek 4.2: Průchody procesem zaměstnancem

- **Větvící body**

Tabulka 4: Průchody procesem - větvící body

Větvící body	
Vstup do aplikace	
přihlášení úspěšné	PÚ
jsem registrovaný	JR
údaje auta validní	ÚAV
údaje zaměstnance validní	ÚZV
staré heslo správně	SHS

nové heslo validní	NHV
všechny rezervace uzavřené	VRU
všechny rezervace zaplacené	VRZ
úpravy uloženy	
zaměstnanec nepřihlášen	
úpravy neuloženy	
auto přidáno	
auto nepřidáno	
auto deaktivováno	
zaměstnanec přidán	
zaměstnanec nepřidán	
účet deaktivován	
účet není deaktivován	

- **Možné akce**

Tabulka 5: Průchody procesem - možné přechody mezi větvími body

Možné akce		
	Z bodu	Do bodu
1	Vstup do aplikace	PÚ
2	PÚ	JR
3	JR	PÚ
4	JR	Zaměstnanec nepřihlášen
5	PÚ	ÚAV
6	ÚAV	Auto přidáno
7	ÚAV	Úpravy uloženy
8	ÚAV	Auto nepřidáno
9	ÚAV	Úpravy neuloženy
10	PÚ	Auto deaktivováno
11	PÚ	ÚZV
12	ÚZV	Zaměstnanec přidán
13	ÚZV	Zaměstnanec nepřidán
14	ÚZV	Úpravy uloženy
15	ÚZV	Úpravy neuloženy
16	PÚ	SHS
17	SHS	Heslo nezměněno
18	SHS	NHV
19	NHV	Heslo nezměněno

Možné akce		
	Z bodu	Do bodu
20	NHV	Heslo změněno
21	PÚ	VRU
22	VRU	Účet není deaktivován
23	VRU	VRZ
24	VRZ	Účet není deaktivován
25	VRZ	Účet deaktivován

- **Vstupy a výstupy jednotlivých větvících bodů**

Tabulka 6: Průchody procesem - vstupy a výstupy jednotlivých větvících bodů

Vstupy a výstupy jednotlivých větvících bodů		
Větvící body	Vstupy	Výstupy
Vstup do aplikace		
PÚ	1, 3	2, 5, 10, 11, 16, 21
JR	2	3, 4
ÚAV	5	6, 7, 8, 9
ÚZV	11	12, 13, 14, 15
SHS	16	17, 18
NHV	18	19, 20
VRU	21	22, 23
VRZ	23	24, 25
úpravy uloženy	7, 14	
zaměstnanec nepřihlášen	4	
úpravy neuloženy	9, 15	
auto přidáno	6	
auto nepřidáno	8	
auto deaktivováno	10	
zaměstnanec přidán	12	
zaměstnanec nepřidán	13	
heslo změněno	19	
heslo nezměněno	17, 20	
účet deaktivován	22, 24	
účet není deaktivován	25	

## 4.4. Výsledky testů

Při vytváření testů jsem se snažil o pokrytí testování vstupů a testování průchodu pomocí jak JUnit testů, tak i Postman testů. V případě založení rezervace, scénář testování vstupů č.5 byl otestován pouze prostřednictvím JUnit, jelikož testování v Postmanu by vyžadovalo vytvoření takového počtu HTTP dotazů (requestů), které by pro mě bylo časově již příliš náročné. To samé platí i v případě scénářů úpravy údajů zákazníka č. 9 a 11, scénářů úpravy údajů zaměstnance č. 9 a 11, scénářů úpravy údajů o platební kartě č. 2 a 3. Naopak scénáře, týkající se změny hesla a stavu, zda je zákazník či zaměstnanec přihlášen, jsou otestovány pouze prostřednictvím Postman. Testované scénáře a jejich číslování naleznete v příloze.

- **JUnit testy**

Všechny testy prošly úspěšně při každém spuštění, s výjimkou testu scénáře na úpravu jména auta č. 3, který v některých případech hlásí že konečný výsledek se liší od výsledku požadovaného. V takovém případě je potřeba provést vyčištění cache projektu. Pokud ovšem ani vyčištění cache projektu nezabere, je potřeba daný test spustit individuálně.

- **Postman testy**

Všechny testy prošly úspěšně při každém spuštění, s výjimkou testů na ukládání či úpravu údajů zákazníka či zaměstnance, které v některých případech hlásí že konečný výsledek se liší od výsledku požadovaného. Je to z toho důvodu, že náhodně vygenerované telefonní číslo je 8-místné místo 9-místného. V takovém případě je potřeba provést daný HTTP request znovu, dokud náhodně vygenerované telefonní číslo není 9-místné.

## 5. Možná rozšíření

V dané kapitole se pojednává o implementaci funkcionalit a rozšíření, které by serverovou část aplikace mohly vylepšit. Některá možná rozšíření se můžou navzájem vylučovat, proto není vhodné implementovat všechny uvedené možnosti, o každé z nich se však uvažuje jako o takové, která by mohla být implementovaná.

### 5.1. Validace účtu pomocí emailu

Při založení účtu, pokud jsou všechny údaje validní, uživateli přijde link s validací na uvedený email. Po kliknutí na link je účet aktivován, do té doby se však uživatel nebude moci přihlásit pomocí uživatelského jména a hesla, uvedených v registraci. Validace pomocí emailu je jednak dalším způsobem ověření, zda je email validní, jednak zabrání uživateli uvést email, ke kterému nemá přístup, jestliže nezná přihlašovací údaje.

Pro implementaci daného rozšíření bude potřeba založit tabulku v databázi, kam se budou ukládat tzv. **verifikační tokeny**. Daná tabulka bude v relačním vztahu 1:1 vůči tabulce, do které se ukládají údaje o zákazníkovi. Tento token bude založen v okamžiku, kdy uživatel klikne na verifikační link, uvedený v emailu. [12]

### 5.2. Validace účtu zaměstnancem – ověření řidičského průkazu

Při založení účtu uživatel uvede další údaj – fotografie řidičského průkazu. Pokud jsou všechny údaje validní, je záznam uložen do databáze. Fotografie je uložena do speciální složky na serveru, určené pro fotografie řidičského průkazu, a do databáze se uloží název dané fotky. Pak uživatel čeká, než mu na email přijde rozhodnutí od zaměstnance.

Zaměstnanec u daného účtu zkontroluje, zda číslo řidičského průkazu uvedené v samostatném poli se shoduje s číslem řidičského průkazu na fotografii, a zda je řidičský průkaz platný. Pokud obě podmínky jsou splněné, zaměstnanec změní status účtu na validní a uživateli přijde email s potvrzením úspěšné registrace. Pokud je jedna z těchto podmínek nesplněná, je uživateli odeslán email se zamítnutím registrace, účet je smazán z databáze a spolu s ním i fotografie ve složce.

### **5.3. Možnost pro zaměstnance vypsat rezervace podle zákazníka**

Místo toho, aby zaměstnanec vypisoval všechny založené rezervace a poté je filtrovat podle zákazníka, bude most vybrat konkrétní osobu, jejíž rezervace chce vypsat. Snižuje tím množství dat, které z databáze vypíše a které budou z časem narůstat a jejich načtení bude pomalejší a obtížnější.

### **5.4. Dvoufázové přihlášení**

V první fázi uživatel uvede přihlašovací údaje, uživatelské jméno a heslo. Pokud v databázi existuje záznam s odpovídajícím uživatelským jménem a heslem, je uživateli na jeho telefonní číslo odeslána zpráva z vygenerovaným kódem, který uživatele uvede do formuláře ve druhé fázi. Pokud kód, který uživatel uvedl do formuláře, odpovídá tomu, který mu byl odeslán na telefonní číslo, uživatel je úspěšně přihlášen.

### **5.5. Lightweight Cache**

Při úspěšném přihlášení je objekt uložen do tzv. Lightweight Cache neboli úschovny o kapacitě n-položek na serveru. Při opětovném přihlášení se stejnými přihlašovacími údaji pak server nemusí znovu zasahovat do databáze - pošle klientovi objekt, který již měl uložený v této úschovně.

### **5.6. Historie pohybu auta**

Do databáze se budou záznamy umístění auta. Záznamy budou obsahovat zeměpisnou šířku a délku v okamžik aktivace rezervace, a zeměpisnou šířku a délku v okamžik ukončení. Záznam bude vytvořen v okamžik aktivace rezervace.

## 6. Závěr a zhodnocení

V rámci této práce byly provedeny rešerše již existujících řešení pro vytvoření serveru aplikace na půjčování vozidel, byla navržena a následně implementována, otestována a následně nasazena serverová část aplikace na půjčování aut – AllTimeCar.

Nejdříve byly provedeny rešerše již existujících řešení. V rámci rešerše byl kladen důraz především na backend aplikace. Vzhledem k tomu, že nahlédnout do backendů bylo nemožné, na základě funkcí, nabízených frontendem, bylo vydedukováno předpokládané fungování serverové části.

Na základě silných a slabých stránek, nalezených při rešerši, jsem se rozhodl o tom, že server naší aplikace musí být více automatizovaný s cílem minimalizace zásahů do serveru zaměstnancem. Některé ze zkoumaných aplikací vyžadovaly zásahy zaměstnanců pro rozhodnutí o potvrzení rezervace a přijetí platby, což mi přišlo jako zbytečné a zpomalující proces rezervace auta jako takový. U některých řešení chyběla možnost zaplacení rezervace a nebylo jasné, jakým způsobem by tato platba probíhala, proto bylo rozhodnuto o implementaci funkcionality, která by umožnila zákazníkovi zaplatit za rezervaci přes aplikaci.

Na základě výsledků těchto analýz byly vypracovány funkční a nefunkční požadavky doplněné o případy užití, databázový model, modely hlavních procesů a životní cykly jednotlivých objektů. Technická struktura, včetně technologií potřebných pro její realizaci, byla navržena na základě předchozích zkušeností a znalostí, nabytých při studiu na fakultě.

Pro implementaci platební brány bylo rozhodnuto implementovat knihovnu j2Pay. Je to multiplatformní knihovna pro zpracování plateb a poskytuje již předepsané funkcionality pro přijetí jednorázových plateb, pravidelných plateb a navrácení peněz u platebních brán Stripe, NMI, Authorize, PayFlowPro, EasyPay, BillPro, BrainTree, Checkout a PayEezy. J2Pay zjednodušuje vývojáři práci, jelikož nemusí studovat implementace jednotlivých platebních brán. Pro tuto aplikaci bylo rozhodnuto o tom, aby zpracování plateb pro bankovní účet probíhalo přes platební bránu Stripe. Tato platební brána také poskytuje data pro testování zpracování plateb. Tyto data jsou k dispozici na webové adrese <https://stripe.com/docs/testing>.

Aplikace byla úspěšně implementována a společně s Postman testy ji lze najít v repozitáři <https://gitlab.fel.cvut.cz/sarhsart/bachelor-thesis.git>. Její funkčnost a komunikaci s frontendem lze také vyzkoušet přes samotný frontend, který lze najít v repozitáři <https://gitlab.fel.cvut.cz/jindrji5/bakalarska-prace.git>. Databáze je implementovaná na webové adrese <https://remotemysql.com/>, z bezpečnostních důvodů přihlašovací údaje uvést není možné. V průběhu implementace byla aplikace podrobená testování pomocí JUnit a Postman testů. Na základě výsledků byly opraveny technické chyby.

Při testování backendu pomocí Postman testů bylo zjištěno, že názvy enum hodnot v databázi jsou case-sensitive, tzn. backend nedokáže uložit enum hodnotu do databáze, jestliže název enum hodnoty na backendu a v databázi se liší i ve velikosti písmen. Např. „MANUAL“ a „manual“



jsou vnímány jako odlišné názvy. Z toho důvodu byly názvy enum hodnot na backendu upraveny tak, aby odpovídaly názvům enum hodnot v databázi včetně velikosti písmen. Také bylo zjištěno, že ukládání datumu do databáze jako hodnoty typu Date je nevhodné, pokud chceme zaznamenat i čas. Místo toho je potřeba definovat typ hodnoty jako Timestamp. Současně odpovídající atribut na backendu stále zůstává hodnotou typu `java.util.Date`.

S ohledem na nedostatek času, některé funkcionality implementovány nebyly. Tyto funkcionality však byly zaznamenány jako možná rozšíření, které by v budoucnu mohly aplikaci zlepšit.

## Literatura

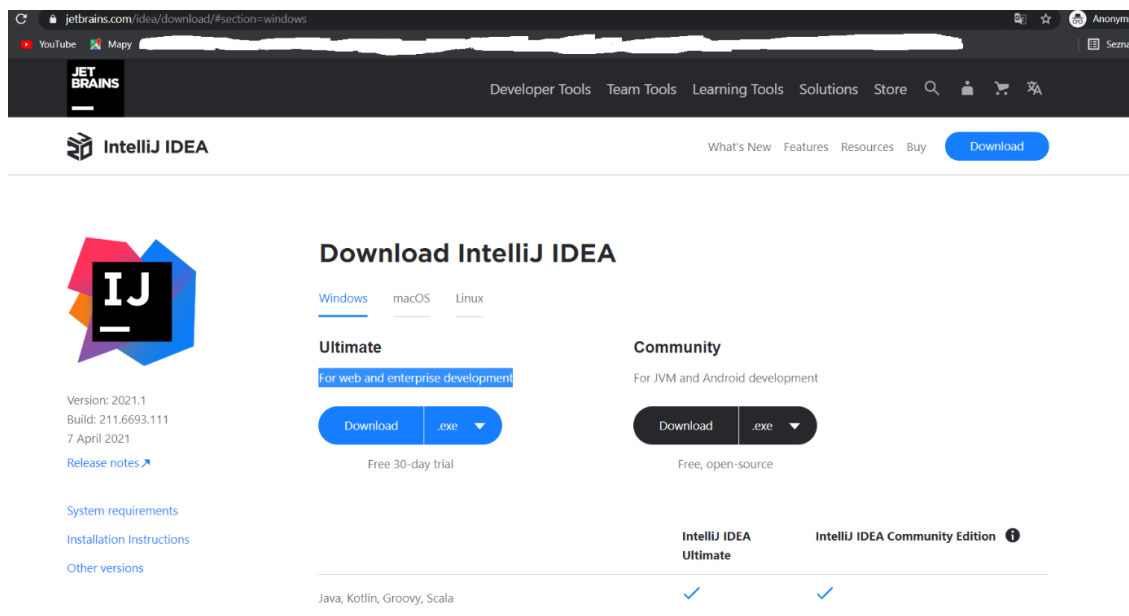
- [1] Co je relační databáze. Oracle.com [online]. [cit. 2021-04-16]. Dostupné z: <https://www.oracle.com/cz/database/what-is-a-relational-database/>
- [2] MySQL krok za krokem: Úvod do MySQL a příprava prostředí. *Itnetwork.cz: Sociální síť pro IT profesionály* [online]. 2012 [cit. 2021-04-16]. Dostupné z: <https://www.itnetwork.cz/mysql/mysql-tutorial-uvod-a-priprava-prostredi/?all-comments>
- [3] ŠTRÁFELDA, Jan. Co je MySQL. *Strafelda.cz* [online]. [cit. 2021-04-16]. Dostupné z: <https://www.strafelda.cz/mysql>
- [4] Spring Framework - Overview. *TutorialsPoint: Simple, Easy Learning* [online]. [cit. 2021-04-16]. Dostupné z: [https://www.tutorialspoint.com/spring/spring\\_overview.htm](https://www.tutorialspoint.com/spring/spring_overview.htm)
- [5] SHEIL, Humphrey. To EJB, or not to EJB?: Addressing the issues and decisions that go into adopting an EJB-based solution. *InfoWorld from IDG* [online]. [cit. 2021-04-16]. Dostupné z: <https://www.infoworld.com/article/2075890/to-ejb--or-not-to-ejb-.html>
- [6] YAGHINI, Vahid. Create Web Applications Efficiently With the Spring Boot MVC Framework: Organize your application code in three-tier architecture. *Open Class Rooms* [online]. 2020-03-07 [cit. 2021-04-16]. Dostupné z: <http://openclassrooms.com/en/courses/5684146-create-web-applications-efficiently-with-the-spring-boot-mvc-framework/6156961-organize-your-application-code-in-three-tier-architecture>
- [7] Třívrstvá architektura (Three-tier architecture): Co je Třívrstvá architektura (Three-tier architecture). *ManagementMania* [online]. 2015-12-05 [cit. 2021-04-16]. Dostupné z: <https://managementmania.com/cs/trivrstva-architektura-three-tier-architecture>
- [8] Data Access Object Pattern. *TutorialsPoint: Simple, Easy Learning* [online]. [cit. 2021-04-16]. Dostupné z: [https://www.tutorialspoint.com/design\\_pattern/data\\_access\\_object\\_pattern.htm](https://www.tutorialspoint.com/design_pattern/data_access_object_pattern.htm)
- [9] The DAO Pattern in Java. *Baeldung* [online]. 2020-03-21 [cit. 2021-04-16]. Dostupné z: <https://www.baeldung.com/java-dao-pattern>
- [10] DRESLER, Robert. *Vícevrstvé architektury aplikací* [online]. 2011-04-26 [cit. 2021-04-17]. Dostupné z: <http://www.robertdresler.cz/2011/04/vicevrstve-architektury-aplikaci.html>

- [11] Spring Security: Part II. Architecture and Implementation. *Spring* [online]. [cit. 2021-04-17]. Dostupné z: <https://docs.spring.io/spring-security/site/docs/3.2.0.CI-SNAPSHOT/reference/html/technical-overview.html>
  
- [12] AMARAPALA, Angela. Account activation by sending email confirmation with Java Spring MVC. *Medium: Where good ideas find you* [online]. 2018-08-20 [cit. 2021-04-22]. Dostupné z: <https://medium.com/@angela.amarapala/sending-email-confirmation-for-account-activation-with-spring-java-cc3f5bb1398e>
  
- [13] N-INS 3. Data management, Architektury systémů. *Státnice na FI MUNI* [online]. 2020-04-12 [cit. 2021-5-21]. Dostupné z: <http://statnice.dqd.cz/mgr-szz:in-ins:3-ins>

# Přílohy

## Spuštění aplikace

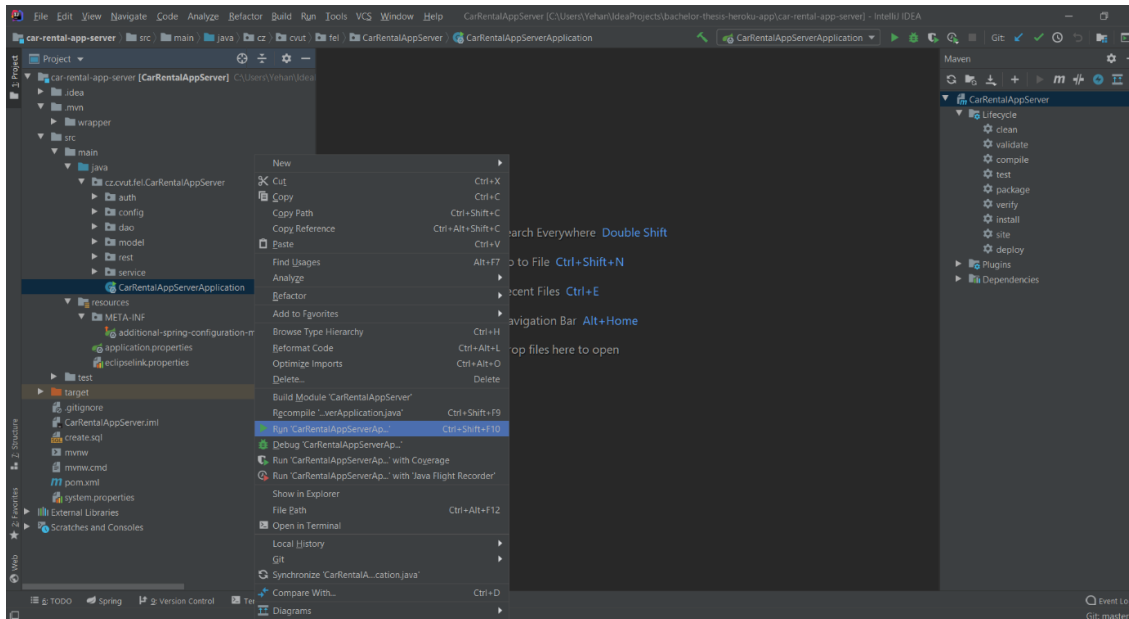
- **Krok 1** – instalace IntelliJ IDEA, **For web and enterprise development**, viz. Obrázek přílohy 1
  - Návod naleznete na webové stránce <https://treehouse.github.io/installation-guides/windows/intellij-idea-win.html>



Obrázek přílohy 1: Stažení instalačního balíčku IntelliJ IDEA

- **Krok 2** – instalace Java Development Kit, verze 11.0.4 a 1.8.0\_202
- **Krok 3** – žádost o přístup k repozitáři <https://gitlab.fel.cvut.cz/sarhsart/bachelor-thesis.git>. Žádost odešlete na [sarhsart@fel.cvut.cz](mailto:sarhsart@fel.cvut.cz) a počkejte než je Vám přístup udělen
- **Krok 4** – stažení aplikace
  - Zvolte složku, do které chcete projekt stáhnout, v této složce otevřete příkazovou řádku
  - Do příkazové řádky zadejte příkaz **git clone** <https://gitlab.fel.cvut.cz/sarhsart/bachelor-thesis.git>
- **Krok 5** – otevření aplikace
  - Otevřete IntelliJIDEA a zvolte možnost **Open**. Otevře se vám Průzkumník souborů. Následně vyberte stažený projekt

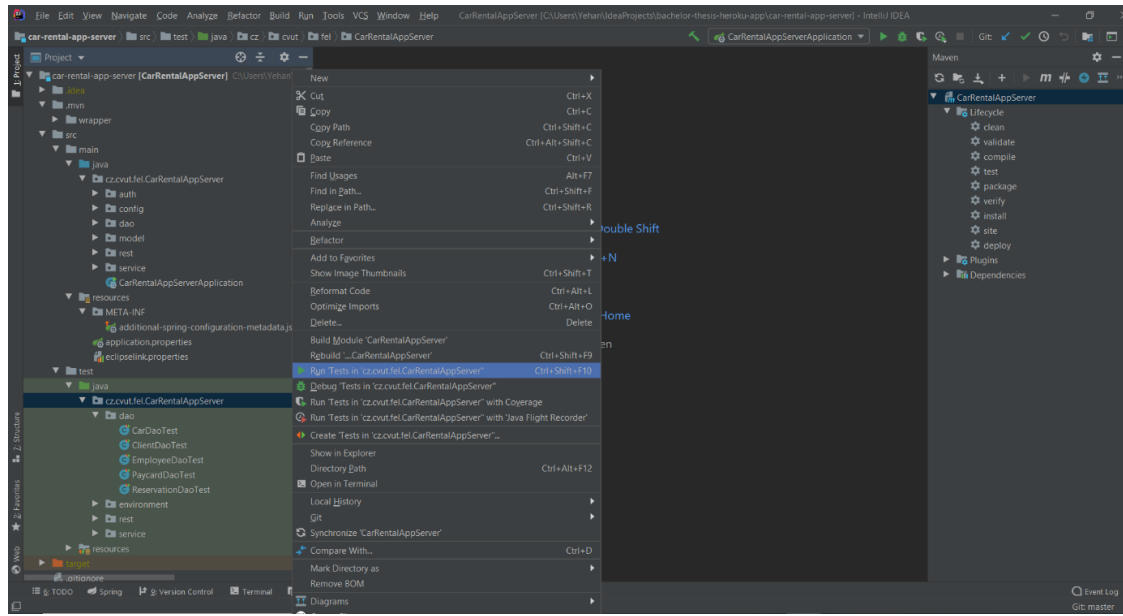
- **Krok 6** – spuštění aplikace
  - Klikněte pravým tlačítkem myši na hlavní třídu a zvolte možnost **Run**, viz. Obrázek přílohy 2



Obrázek přílohy 2: Spuštění aplikace

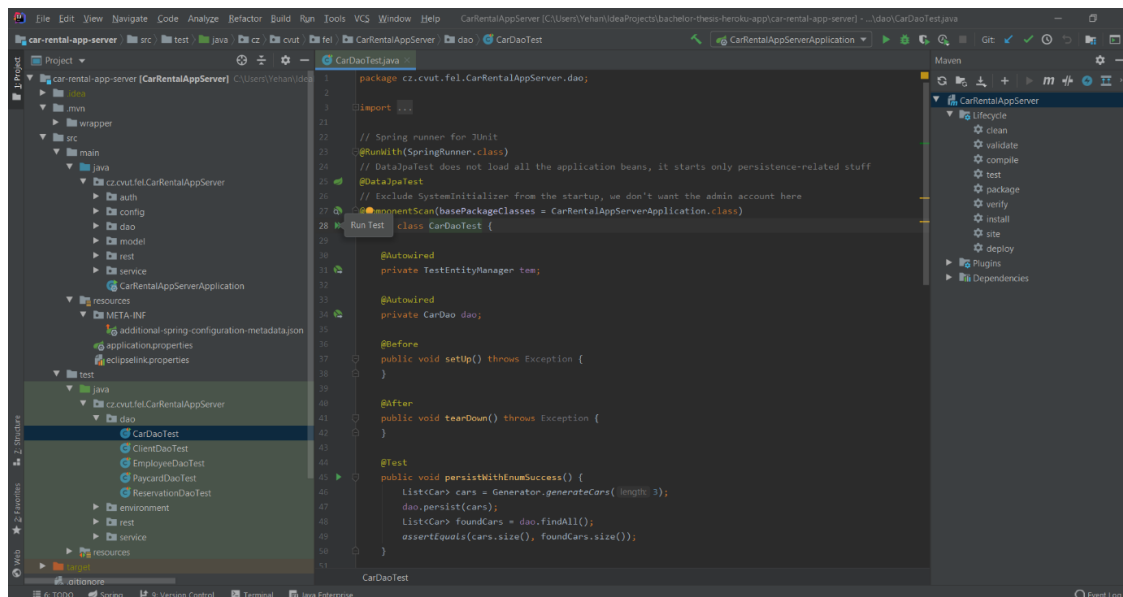
## JUnit testy

- **Spuštění všech testů / všech testů ve složce:** pravým tlačítkem klikněte na složku a zvolte možnost **Run**, viz. Obrázek přílohy 3



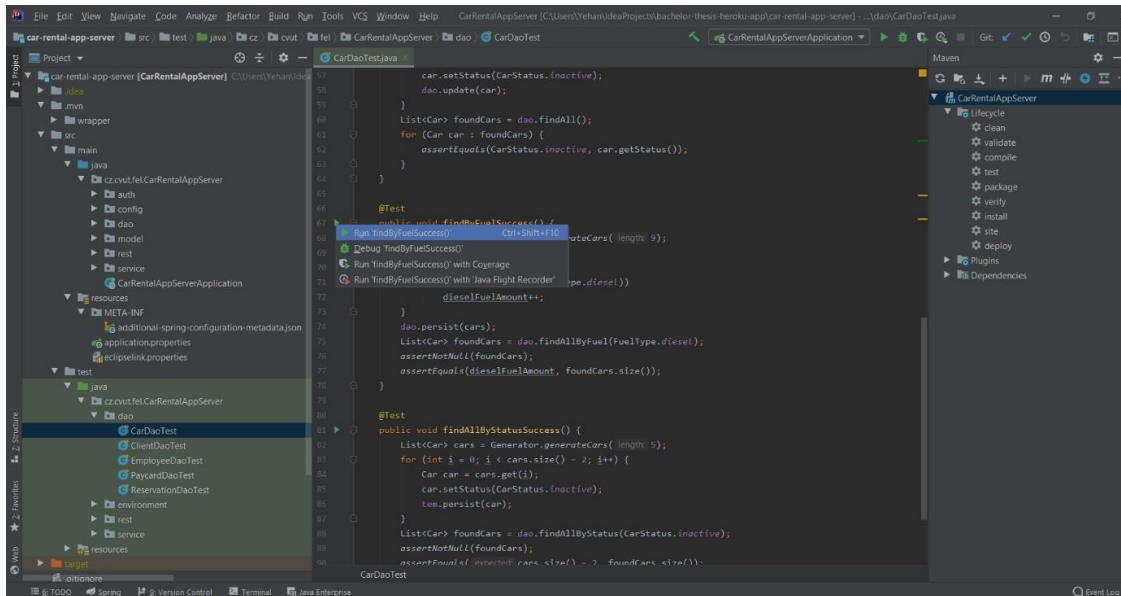
Obrázek přílohy 3: Spuštění všech testů

- **Spuštění všech testů v souboru:** rozklikněte soubor a zvolte možnost **Run**, viz. Obrázek přílohy 4



Obrázek přílohy 4: Spuštění všech testů v souboru

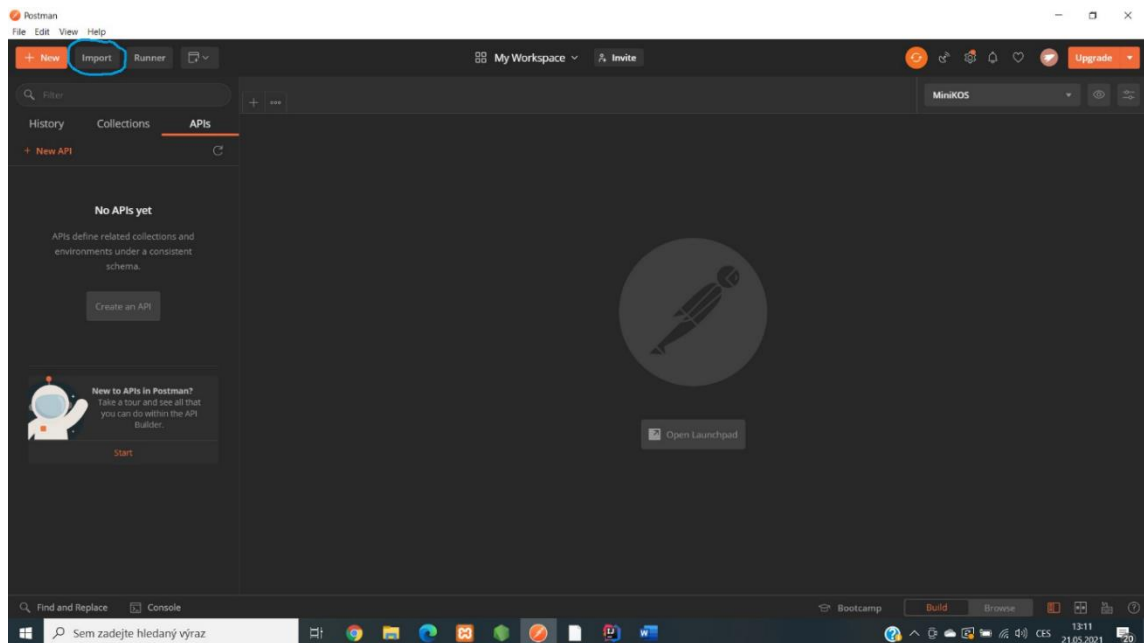
- **Spuštění jednoho testu v souboru:** u daného testu zvolte možnost **Run**, viz. Obrázek přílohy 5



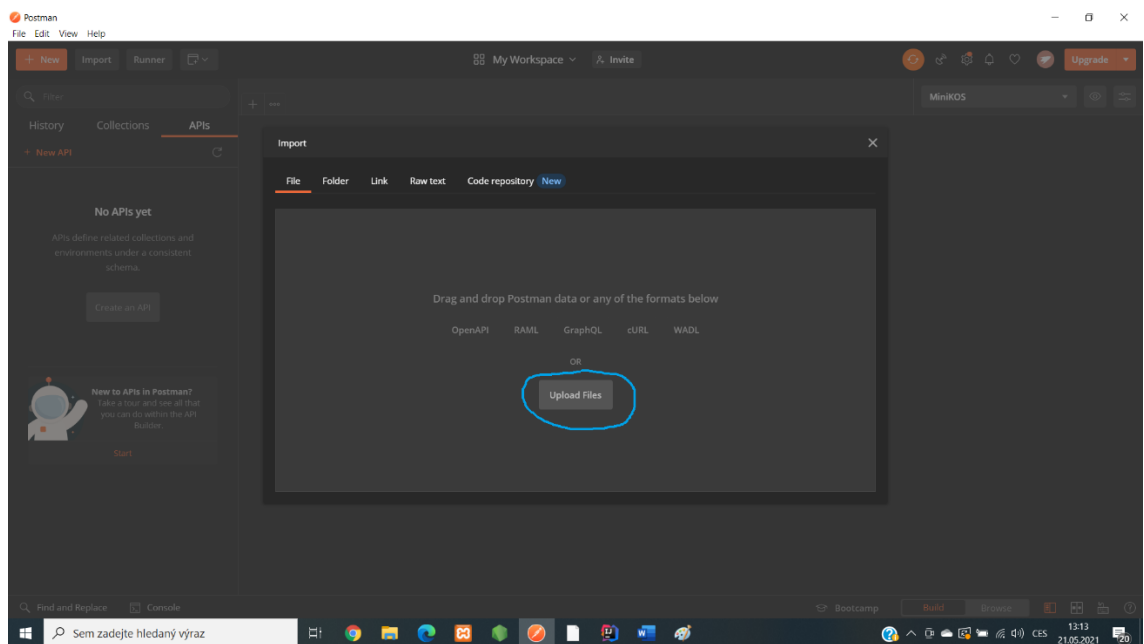
Obrázek přílohy 5: Spuštění jednoho testu v souboru

## Postman testy

- **Krok 1** – instalace Postman na <https://www.postman.com/downloads/>
- **Krok 2** – rozbalit ZIP soubor ze staženého repozitáře. Soubor se jmenuje **CarRentalAppServer-postman\_collection-19-05-2021.zip**
- **Krok 3** – rozbalený soubor vložit do Postman, viz. Obrázek přílohy 6 a Obrázek přílohy 7



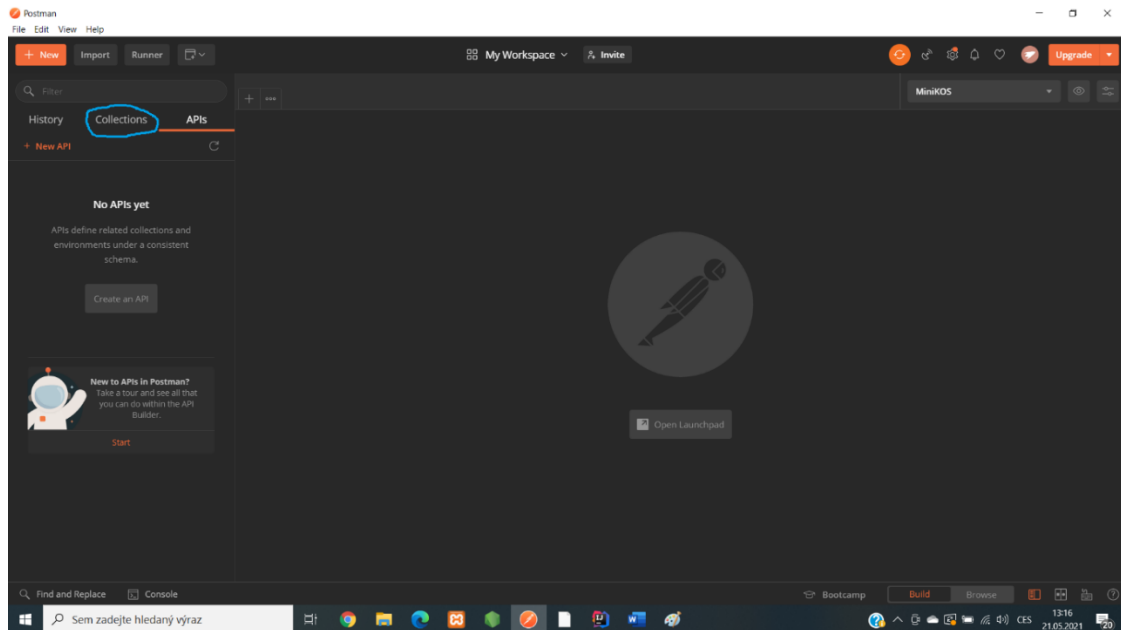
Obrázek přílohy 6: Postman testy lze vložit po zvolení možnosti Import



Obrázek přílohy 7: Nahrát Postman testy



- Testy najdete ve složce Collections, viz. Obrátek přílohy 8



Obrátek přílohy 8: Možnost Collections v Postmanu

- Testovány jsou pouze příkazy, které modifikují data v databázi
- Příkazy GET jsou použité pro získání ID objektu, jehož modifikaci chceme otestovat. U příkazů s výpisem seznamu položek se vybírá poslední ID v seznamu (ID s nejvyšší hodnotou)
- Příkazy jsou roztrženy do složek a podsložek podle následujících kritérií:
  - Typ REST controlleru
  - Testovaná funkcionality, pokud je potřeba víc než 1 příkaz
  - Příkazy v dané složce jsou nezávislé na výsledku příkazů ze složky, která nepatří do složky současné a je v hierarchickém stromu na stejné nebo vyšší úrovni oproti současné složce
- Všechny příkazy jsou prováděny v pořadí, v jakém se nachází ve složce
- Podmínky, které mají být splněné po provedení příkazů:
  - Všechny příkazy složky Car Controller a její podsložek:
    - 12 aut (1 se stavem „neaktivní“) a 9 zaměstnanců jsou přidány do databáze
  - Všechny příkazy složky Client Controller a její podsložek:
    - Podsložka Deactivate Account Success
      - 1 zákazník (stav „neaktivní“), 1 auto, 1 zaměstnanec, 1 rezervace (status „ukončená“, „zaplacená“) jsou přidány do databáze
    - Podsložka Deactivate Account With Unclosed Reservation Failure
      - 1 zákazník, 1 auto, 1 zaměstnanec, 1 rezervace (status „aktivní“, „nezaplacená“) jsou přidány do databáze

- Podložka Deactivate Account With Unpaid Reservation Failure
  - 1 zákazník, 1 auto, 1 zaměstnanec, 1 rezervace (status „ukončená“, „nezaplacená“) jsou přidány do databáze
- Podložka Deactivate Account When Unauthorized Failure
  - 1 zákazník přidán do databáze
- 12 zákazníků (1 se statusem „neaktivní“) jsou přidány do databáze
- 4 rezervace (2 se statusem „finished“ z nichž 1 je zaplacená, 1 se statusem „active“) jsou přidány do databáze
- 5 zaměstnanců jsou přidány do databáze
- 4 auta jsou přidány do databáze
- Všechny příkazy složky Employee Controller a její podložek:
  - Podložka Deactivate Account Success
    - 1 zákazník (stav „neaktivní“), 1 auto, 1 zaměstnanec, 1 rezervace (status „ukončená“, „zaplacená“) jsou přidány do databáze
  - Podložka Deactivate Account With Unclosed Reservation Failure
    - 1 zákazník, 1 auto, 1 zaměstnanec, 1 rezervace (status „aktivní“, „nezaplacená“) jsou přidány do databáze
  - Podložka Deactivate Account With Unpaid Reservation Failure
    - 1 zákazník, 1 auto, 1 zaměstnanec, 1 rezervace (status „ukončená“, „nezaplacená“) jsou přidány do databáze
  - Podložka Deactivate Account When Unauthorized Failure
    - 1 zákazník, 1 zaměstnanec jsou přidány do databáze
  - 10 zaměstnanců jsou přidány do databáze
  - 4 zákazníků jsou přidány do databáze
  - 3 auta jsou přidány do databáze
  - 3 rezervace jsou přidány do databáze
- Všechny příkazy složky Paycard Controller a její podložek:
  - 3 auta, 10 zákazníků, 3 zaměstnanců a 3 rezervací jsou přidány do databáze
- Všechny příkazy složky Reservation Controller a její podložek:
  - Podložka Post Reservation 1 Success:
    - 1 auto, 1 zaměstnanec, 1 zákazník, 1 rezervace přidány do databáze
  - Podložka Post Reservation 2 Failure:
    - 1 auto, 1 zaměstnanec, 0 zákazníků, 0 rezervací přidány do databáze
  - Podložka Post Reservation 3 Failure:
    - 2 auta, 2 zaměstnanců, 1 zákazník, 1 rezervací přidány do databáze
  - Podložka Post Reservation 4 Failure:
    - 2 auta, 1 zaměstnanec, 1 zákazník, 1 rezervací přidány do databáze
  - Podložka Post Reservation 6 Failure:
    - 1 auto, 1 zaměstnanec, 1 zákazník, 0 rezervací přidány do databáze

- Podložka Post Reservation 7 Failure:
  - 1 auto, 1 zaměstnanec, 1 zákazník, 0 rezervací přidány do databáze
- Podložka Post Reservation 8 Failure:
  - 1 auto, 2 zaměstnanců, 2 zákazník, 1 rezervací přidány do databáze
- Podložka Post Reservation 9 Failure:
  - 1 auto se stavem „neaktivní“, 1 zaměstnanců, 1 zákazník, 0 rezervací přidány do databáze
- Podložka Post Reservation 10 Failure:
  - 2 auta, 1 zaměstnanců, 1 zákazník, 2 rezervací přidány do databáze
- Podložka Pay Reservation 1 Success:
  - 2 auta, 2 zaměstnanců, 2 zákazník, 1 ukončená a zaplacená rezervace přidány do databáze
- Podložka Pay Reservation 2 Failure:
  - 1 auto, 1 zaměstnanců, 1 zákazník, 1 aktivní a nezaplacená rezervace přidány do databáze
- Podložka Has No Unfinished Business
  - 1 auto, 1 zaměstnanců, 1 zákazník, 1 ukončená a zaplacená rezervace přidány do databáze
- Podložka Has Unclosed Reservation
  - 1 auto, 1 zaměstnanců, 1 zákazník, 1 aktivní a nezaplacená rezervace přidány do databáze
- Podložka Has Unpaid Reservation
  - 1 auto, 1 zaměstnanců, 1 zákazník, 1 ukončená a nezaplacená rezervace přidány do databáze
- Všechny příkazy složky Login Controller a její podložek:
  - 2 zákazníci (1 se stavem „neaktivní“) a 1 zaměstnanec jsou přidány do databáze
- Počet přidanych platebních karet do databáze
  - Po provedení všech testů, když je databáze prázdná – 8 karet
  - Po opětovném provedení všech testů – 5 karet

## Testování vstupů

- **Registrace – uložení zákazníka**

### Vstupní hodnoty

Tabulka 7: Testované stavy vstupních hodnot a jejich reprezentace v testech jako číselná hodnota. Část 1

	<b>UJ</b>	<b>PSW</b>	<b>KJ</b>	<b>PŘ</b>
0	v pořádku	v pořádku	v pořádku	v pořádku
1	Již použité	Méně než 8 znaků	S nepovoleným znakem <sup>1</sup>	S nepovoleným znakem <sup>2</sup>
2	S nepovoleným znakem <sup>3</sup>	S nepovoleným znakem <sup>4</sup>	prázdné	prázdné
3	prázdné			

Tabulka 8: Testované stavy vstupních hodnot a jejich reprezentace v testech jako číselná hodnota. Část 1

	<b>DN</b>	<b>E</b>	<b>TEL</b>	<b>ČRP</b>
0	Takový, že výsledný věk je 18+let	V pořádku	V pořádku	V pořádku
1	Takový, že výsledný věk je pod 18 let	Již používané	Již používané	Již používané
2		nevalidní	Délka není 9 číslic	Délka není 8 znaků
3				První 2 znaky nevalidní <sup>5</sup>
4				Posledních 6 znaků nevalidních <sup>6</sup>

<sup>1</sup> Jakýkoliv znak jiný než písmeno

<sup>2</sup> Jakýkoliv znak jiný než písmeno

<sup>3</sup> Jakýkoliv znak jiný než písmeno či číslice

<sup>4</sup> Jakýkoliv znak jiný než písmeno či číslice

<sup>5</sup> Obsahují jakýkoliv znak jiný než písmeno

<sup>6</sup> Obsahují jakýkoliv znak jiný než číslici

**Testovací scénáře***Tabulka 9: Registrace zákazníka - testovací scénáře*

Vstup									Výstup
	UJ	PSW	KJ	PŘ	DN	E	TEL	ČŘP	
1.	0	0	0	0	0	0	0	0	OK
2.	1	0	0	0	0	0	0	0	ERR
3.	2	0	0	0	0	0	0	0	ERR
4.	0	1	0	0	0	0	0	0	ERR
5.	0	2	0	0	0	0	0	0	ERR
6.	0	0	1	0	0	0	0	0	ERR
7.	0	0	2	0	0	0	0	0	ERR
8.	0	0	0	1	0	0	0	0	ERR
9.	0	0	0	2	0	0	0	0	ERR
10.	0	0	0	0	1	0	0	0	ERR
11.	0	0	0	0	0	1	0	0	ERR
12.	0	0	0	0	0	2	0	0	ERR
13.	0	0	0	0	0	0	1	0	ERR
14.	0	0	0	0	0	0	2	0	ERR
15.	0	0	0	0	0	0	0	1	ERR
16.	0	0	0	0	0	0	0	2	ERR
17.	3	0	0	0	0	0	0	0	ERR
18.	0	0	0	0	0	0	0	3	ERR
19.	0	0	0	0	0	0	0	4	ERR

- **Registrace – založení zaměstnance**

**Vstupní hodnoty**

Tabulka 10: Testované stavy vstupních hodnot a jejich reprezentace v testech jako číselná hodnota. Část 1

	<b>UJ</b>	<b>PSW</b>	<b>KJ</b>	<b>PŘ</b>
0	v pořádku	v pořádku	v pořádku	v pořádku
1	Již použité	Méně než 8 znaků	S nepovoleným znakem <sup>7</sup>	S nepovoleným znakem <sup>8</sup>
2	S nepovoleným znakem <sup>9</sup>	S nepovoleným znakem <sup>10</sup>	prázdné	prázdné
3	prázdné			

Tabulka 11: Testované stavy vstupních hodnot a jejich reprezentace v testech jako číselná hodnota. Část 2

	<b>DN</b>	<b>E</b>	<b>TEL</b>
0	Takový, že výsledný věk je 18+let	V pořádku	V pořádku
1	Takový, že výsledný věk je pod 18 let	Již používané	Již používané
2		nevalidní	Délka není 9 číslic

**Testovací scénáře**

Tabulka 12: Registrace zaměstnance - testovací scénáře

	<b>Vstup</b>							<b>Výstup</b>
	<b>UJ</b>	<b>PSW</b>	<b>KJ</b>	<b>PŘ</b>	<b>DN</b>	<b>E</b>	<b>TEL</b>	
1.	0	0	0	0	0	0	0	OK
2.	1	0	0	0	0	0	0	ERR

<sup>7</sup> Jakýkoliv znak jiný než písmeno

<sup>8</sup> Jakýkoliv znak jiný než písmeno

<sup>9</sup> Jakýkoliv znak jiný než písmeno či číslice

<sup>10</sup> Jakýkoliv znak jiný než písmeno či číslice

Vstup								Výstup
	UJ	PSW	KJ	PŘ	DN	E	TEL	
3.	2	0	0	0	0	0	0	ERR
4.	0	1	0	0	0	0	0	ERR
5.	0	2	0	0	0	0	0	ERR
6.	0	0	1	0	0	0	0	ERR
7.	0	0	2	0	0	0	0	ERR
8.	0	0	0	1	0	0	0	ERR
9.	0	0	0	2	0	0	0	ERR
10.	0	0	0	0	1	0	0	ERR
11.	0	0	0	0	0	1	0	ERR
12.	0	0	0	0	0	2	0	ERR
13.	0	0	0	0	0	0	1	ERR
14.	0	0	0	0	0	0	2	ERR
15.	3	0	0	0	0	0	0	ERR

- **Přihlášení**

### Vstupní hodnoty

Tabulka 13: Testované stavy vstupních hodnot a jejich reprezentace v testech jako číselná hodnota

	<b>uživatelské jméno</b>	<b>heslo</b>	<b>uživatelský status</b>
0	Je v databázi	Náleží uživateli s daným uživatelským jménem	Aktivní
1	Není v databázi	Nenáleží uživateli s daným uživatelským jménem	Neaktivní

### Testovací scénáře

Tabulka 14: Přihlášení - testovací scénáře

	<b>Vstup</b>			<b>Výstup</b>
	<b>uživ. jméno</b>	<b>heslo</b>	<b>uživ. status</b>	
1.	0	0	0	Přihlášení úspěšné
2.	0	1	0	Chybová hláška
3.	1	0	0	Chybová hláška
4.	0	0	1	Chybová hláška



- **Změna hesla**

**Vstupní hodnoty***Tabulka 15: Testované stavy vstupních hodnot a jejich reprezentace v testech jako číselná hodnota*

	<b>Uživatel</b>	<b>staré heslo</b>	<b>nové heslo</b>
0	Přihlášen	Správné	V pořádku
1	Nepřihlášen	Špatné	Méně než 8 znaků
2			S nepovoleným znakem <sup>11</sup>

**Testovací scénáře***Tabulka 16: Změna hesla - testovací scénáře*

	<b>Vstup</b>			<b>Výstup</b>
	<b>Uživatel</b>	<b>staré heslo</b>	<b>nové heslo</b>	
1.	0	0	0	Přihlášení úspěšné
2.	0	0	1	Chybová hláška
3.	0	0	2	Chybová hláška
4.	1	0	0	Chybová hláška
5.	0	1	0	Chybová hláška

---

<sup>11</sup> Jakýkoliv znak jiný než písmeno či číslice

- **Úprava údajů zákazníka**

### Vstupní hodnoty

Tabulka 17: Testované stavy vstupních hodnot a jejich reprezentace v testech jako číselná hodnota. Část 1

	<b>Z</b>	<b>KJ</b>	<b>PŘ</b>
0	přihlášen	v pořádku	v pořádku
1	nepřihlášen	S nepovoleným znakem <sup>12</sup>	S nepovoleným znakem <sup>13</sup>
2		prázdné	prázdné

Tabulka 18: Testované stavy vstupních hodnot a jejich reprezentace v testech jako číselná hodnota. Část 2

	<b>DN</b>	<b>E</b>	<b>TEL</b>	<b>ČŘP</b>
0	Takový, že výsledný věk je 18+let	V pořádku	V pořádku	V pořádku
1	Takový, že výsledný věk je pod 18 let	Již používané jiným zákazníkem	Již používané jiným zákazníkem	Již používané
2		nevalidní	Délka není 9 číslic	Délka není 8 znaků
3				První 2 znaky nevalidní <sup>14</sup>
4				Posledních 6 znaků nevalidních <sup>15</sup>

<sup>12</sup> Jakýkoliv znak jiný než písmeno

<sup>13</sup> Jakýkoliv znak jiný než písmeno

<sup>14</sup> Obsahují jakýkoliv znak jiný než písmeno

<sup>15</sup> Obsahují jakýkoliv znak jiný než číslici

**Testovací scénáře***Tabulka 19: Úprava údajů zákazníka - testovací scénáře*

	Vstup							Výstup
	Z	KJ	PŘ	DN	E	TEL	ČŘP	
1.	0	0	0	0	0	0	0	OK
4.	0	1	0	0	0	0	0	ERR
5.	0	2	0	0	0	0	0	ERR
6.	0	0	1	0	0	0	0	ERR
7.	0	0	2	0	0	0	0	ERR
8.	0	0	0	1	0	0	0	ERR
9.	0	0	0	0	1	0	0	ERR
10.	0	0	0	0	2	0	0	ERR
11.	0	0	0	0	0	1	0	ERR
12.	0	0	0	0	0	2	0	ERR
13.	0	0	0	0	0	0	1	ERR
14.	0	0	0	0	0	0	2	ERR
15.	0	0	0	0	0	0	3	ERR
16.	0	0	0	0	0	0	4	ERR
17.	1	0	0	0	0	0	0	ERR

- **Úprava údajů zaměstnance**

**Vstupní hodnoty:**

Tabulka 20: Testované stavy vstupních hodnot a jejich reprezentace v testech jako číselná hodnota. Část 1

	<b>Z</b>	<b>KJ</b>	<b>PŘ</b>
0	přihlášen	v pořádku	v pořádku
1	nepřihlášen	S nepovoleným znakem <sup>16</sup>	S nepovoleným znakem <sup>17</sup>
2		prázdné	prázdné
3			

Tabulka 21: Testované stavy vstupních hodnot a jejich reprezentace v testech jako číselná hodnota. Část 2

	<b>DN</b>	<b>E</b>	<b>TEL</b>
0	Takový, že výsledný věk je 18+let	V pořádku	V pořádku
1	Takový, že výsledný věk je pod 18 let	Již používané jiným zaměstnancem	Již používané jiným zaměstnancem
2		nevalidní	Délka není 9 číslic

**Testovací scénáře**

Tabulka 22: Úprava údajů zaměstnance - testovací scénáře

	<b>Vstup</b>						<b>Výstup</b>
	<b>Z</b>	<b>KJ</b>	<b>PŘ</b>	<b>DN</b>	<b>E</b>	<b>TEL</b>	
1.	0	0	0	0	0	0	OK
4.	0	1	0	0	0	0	ERR
5.	0	2	0	0	0	0	ERR
6.	0	0	1	0	0	0	ERR

<sup>16</sup> Jakýkoliv znak jiný než písmeno

<sup>17</sup> Jakýkoliv znak jiný než písmeno

	<b>Vstup</b>						<b>Výstup</b>
	<b>Z</b>	<b>KJ</b>	<b>PŘ</b>	<b>DN</b>	<b>E</b>	<b>TEL</b>	
7.	0	0	2	0	0	0	ERR
8.	0	0	0	1	0	0	ERR
9.	0	0	0	0	1	0	ERR
10.	0	0	0	0	2	0	ERR
11.	0	0	0	0	0	1	ERR
12.	0	0	0	0	0	2	ERR
13.	1	0	0	0	0	0	ERR

- **Založení auta, úprava údajů auta**

### Vstupní hodnoty

Tabulka 23: Testované stavy vstupních hodnot a jejich reprezentace v testech jako číselná hodnota. Část 1

	<b>Z</b>	<b>B</b>	<b>N</b>	<b>S</b>	<b>F</b>
0	přihlášen	není prázdná	unikátní	enum existuje	enum existuje
1	nepřihlášen	prázdná	již používané	enum neexistuje	enum neexistuje
2			prázdné		

Tabulka 24: Testované stavy vstupních hodnot a jejich reprezentace v testech jako číselná hodnota. Část 2

	<b>T</b>	<b>MP</b>	<b>PPM</b>	<b>LP</b>	<b>ZŠ</b>	<b>ZD</b>
0	enum existuje	větší než 0	větší než 0	aktuálně nepoužívaná	ZŠ ∈ <-90; 90>	ZD ∈ <-180; 180>
1	enum neexistuje	menší nebo rovno 0	menší nebo rovno 0	aktuálně používaná	ZŠ je větší než 90 nebo menší než -90	ZD je větší než 180 nebo menší než -180
2				prázdná	null	null

### Testovací scénáře

Tabulka 25: Založení auta, úprava údajů auta - testovací scénáře

	Vstup											Výstup
	<b>Z</b>	<b>B</b>	<b>N</b>	<b>S</b>	<b>F</b>	<b>T</b>	<b>MP</b>	<b>PPM</b>	<b>LP</b>	<b>ZŠ</b>	<b>ZD</b>	
1.	0	0	0	0	0	0	0	0	0	0	0	OK
2.	0	1	0	0	0	0	0	0	0	0	0	ERR
3.	0	0	1	0	0	0	0	0	0	0	0	ERR
4.	0	0	0	1	0	0	0	0	0	0	0	ERR
5.	0	0	0	0	1	0	0	0	0	0	0	ERR
6.	0	0	0	0	0	0	1	0	0	0	0	ERR
8.	0	0	0	0	0	0	0	1	0	0	0	ERR

	Vstup											Výstup
	Z	B	N	S	F	T	MP	PPM	LP	ZŠ	ZD	
10.	0	0	0	0	0	0	0	0	1	0	0	ERR
11.	0	0	0	0	0	0	0	0	2	0	0	ERR
12.	0	0	0	0	0	1	0	0	0	0	0	ERR
13.	0	0	2	0	0	0	0	0	0	0	0	ERR
14.	1	0	0	0	0	0	0	0	0	0	0	ERR
15.	0	0	0	0	0	0	0	0	0	1	0	ERR
16.	0	0	0	0	0	0	0	0	0	2	0	OK
17.	0	0	0	0	0	0	0	0	0	0	1	ERR
18.	0	0	0	0	0	0	0	0	0	0	2	OK

- **Založení platební karty**

### Vstupní hodnoty

Tabulka 26: Testované stavy vstupních hodnot a jejich reprezentace v testech jako číselná hodnota

	Zákazník	číslo karty	validní do: měsíc validní do: rok	CVV/CVC kód
0	přihlášen	v pořádku	měsíc $\in <0; 12>$ , rok $\geq$ současný rok, budoucnost	počet cifer je 3
1	nepřihlášen	počet cifer je jiný než 16	měsíc $\in <0; 12>$ , rok $\geq$ současný rok, minulost	počet cifer je jiný než 3
2		již používané	měsíc $\in <0; 12>$ , rok $<$ současný rok	obsahuje jiné znaky než číslice
3		obsahuje jiné znaky než číslice	měsíc je menší než 0 nebo větší než 12	
4			měsíc obsahuje jiné znaky než číslice	
5			rok obsahuje jiné znaky než číslice	

### Testovací scénáře

Tabulka 27: Založení platební karty - testovací scénáře

	Vstup				Výstup
	Zákazník	číslo karty	měsíc a rok	CVV	
1.	0	0	0	0	Karta založená
2.	0	1	0	0	Chybová hláška
3.	0	2	0	0	Chybová hláška
4.	0	3	0	0	Chybová hláška
5.	0	0	1	0	Chybová hláška



	<b>Vstup</b>				<b>Výstup</b>
	<b>Zákazník</b>	<b>číslo karty</b>	<b>měsíc a rok</b>	<b>CVV</b>	
6.	0	0	2	0	Chybová hláška
7.	0	0	3	0	Chybová hláška
8.	0	0	4	0	Chybová hláška
9.	0	0	5	0	Chybová hláška
10.	0	0	0	1	Chybová hláška
11.	0	0	0	2	Chybová hláška
12.	1	0	0	0	Chybová hláška

- **Úprava údajů o platební kartě**

### Vstupní hodnoty

Tabulka 28: Testované stavy vstupních hodnot a jejich reprezentace v testech jako číselná hodnota

	Zákazník	číslo karty	validní do: měsíc validní do: rok	CVV/CVC kód
0	přihlášen	stejně	měsíc $\in \langle 0; 12 \rangle$ , rok $\geq$ současný rok, budoucnost	počet cifer je 3
1	nepřihlášen	jiné, nepoužívané	měsíc $\in \langle 0; 12 \rangle$ , rok $\geq$ současný rok, minulost	počet cifer je jiný než 3
2		jiné, již používané	měsíc $\in \langle 0; 12 \rangle$ , rok $<$ současný rok	obsahuje jiné znaky než číslice
3		počet cifer je jiný než 16	měsíc je menší než 0 nebo větší než 12	
4		obsahuje jiné znaky než číslice	měsíc obsahuje jiné znaky než čísllice	
5			rok obsahuje jiné znaky než čísllice	

### Testovací scénáře

Tabulka 29: Úprava údajů o platební kartě - testovací scénáře

	Vstup				Výstup
	Zákazník	číslo karty	měsíc a rok	CVV	
1.	0	0	0	0	Údaje karty změněny
2.	0	1	0	0	Nová karta založená
3.	0	2	0	0	Chybová hláška
4.	0	3	0	0	Chybová hláška
5.	0	4	0	0	Chybová hláška
6.	0	0	1	0	Chybová hláška
7.	0	0	2	0	Chybová hláška

	<b>Vstup</b>				<b>Výstup</b>
	<b>Zákazník</b>	<b>číslo karty</b>	<b>měsíc a rok</b>	<b>CVV</b>	
8.	0	0	3	0	Chybová hláška
9.	0	0	4	0	Chybová hláška
10.	0	0	5	0	Chybová hláška
11.	0	0	0	1	Chybová hláška
12.	0	0	0	2	Chybová hláška
13.	1	0	0	0	Chybová hláška

- **Založení rezervace**

### Vstupní hodnoty

Tabulka 30: Testované stavy vstupních hodnot a jejich reprezentace v testech jako číselná hodnota

	Zákazník	Rezervace	Platební karta	Auto	Zaměstnanec
0	přihlášen	žádná není / všechny uzavřené a zaplacené	validní	je k dispozici	existuje
1	nepřihlášen	min. 1 rezervace neuzavřená	datum platnosti vypršel	není k dispozici	neexistuje
2		min. 1 rezervace nezaplacená	není	není v provozu	
3		min. 1 rezervace zrušená			

### Testovací scénáře

Tabulka 31: Založení rezervace - testovací scénáře

	Vstup					Výstup
	Zákazník	Rezervace	Platební karta	Zaměstnanec	Auto	
1.	0	0	0	0	0	Rezervace založena
2.	1	0	0	0	0	Chybová hláška
3.	0	1	0	0	0	Chybová hláška
4.	0	2	0	0	0	Chybová hláška
5.	0	0	1	0	0	Chybová hláška
6.	0	0	2	0	0	Chybová hláška

	Vstup					Výstup
	Zákazník	Rezervace	Platební karta	Zaměstnanec	Auto	
7.	0	0	0	1	0	Chybová hláška
8.	0	0	0	0	1	Chybová hláška
9.	0	0	0	0	2	Chybová hláška
10.	0	4	0	0	0	Rezervace založena

- **Zaplacení rezervace**

### Vstupní hodnoty

Tabulka 32: Testované stavy vstupních hodnot a jejich reprezentace v testech jako číselná hodnota

	Rezervace	Platební karta
0	je ukončena	validní
1	není ukončená	datum platnosti vypršel

### Testovací scénáře

Tabulka 33: Zaplacení rezervace - testovací scénáře

	Vstup		Výstup
	Rezervace	Platební karta	
1.	0	0	Rezervace zaplacená
2.	1	0	Chybová hláška
3.	0	1	Chybová hláška

## Testy průchodů procesem zákazníkem

- **Kombinace pro testy úrovně pokrytí 1**

Tabulka 34: Průchody procesem - kombinace pro testy úrovně pokrytí 1

<b>Kombinace pro testy úrovně pokrytí 1</b>	
<b>Větvící bod</b>	<b>Výstup</b>
Vstup do aplikace	
PÚ	2, 7, 14, 22, 33(AR), 36(ZR), 39, 42, 45(OK), 50 (ZÚ)
JR	3, 4
RÚ	5, 6
PK	8, 9
ÚPKV	10, 11, 19, 20, 21
ČKDB	12, 13, 17, 18
JČK	15, 16
VRU	23, 24, 46(OK), 47(OK), 51(ZÚ), 52(ZÚ)
VRZ	25, 26, 48(OK), 49(OK), 53(ZÚ), 54(ZÚ)
MPK	27, 28
PKV	29, 30
AkD	31, 32
RjP	34(AR), 35(AR), 37(ZR), 38(ZR)
DPPK	40, 41
OÚV	43, 44
DÚ	55, 56
Karta není přiřazená	
Nová karta založená	
Existující karta přiřazená	
Karta není upravena	
Karta upravena	
Rezervace není založená	
Rezervace založená	
Stav rezervace změněn	
Stav rezervace nezměněn	
Rezervace zaplacená	
Rezervace není zaplacená	
Osobní údaje uloženy	
Osobní údaje neuloženy	

- **Vytvoření testů z kombinací pro úroveň pokrytí 1 (pokrytí cest)**

Tabulka 35: Vytvoření testů z kombinací pro úroveň pokrytí 1 (pokrytí cest)

	<b>Průchod</b>
1	1-2-4-6-7-8
2	1-2-4-6-5-14-15-12
3	1-7-9-11-18-10
4	1-22-24-26-28-30-31
5	1-45(OK)-47(OK)-48(OK)
6	1-33(AR)-34(AR)
7	1-33(AR)-35(AR)
8	1-36(ZR)-37(ZR)
9	1-36(ZR)-38(ZR)
10	1-39-40
11	1-39-41
12	1-42-43
13	1-42-44
14	1-50(ZÚ)-52(ZÚ)-53(ZÚ)
15	1-45(OK)-47(OK)-49(OK)
16	1-45(OK)-46(OK)
17	1-7-9-11-13
18	1-7-9-19
19	1-14-16-20
20	1-14-15-17
21	1-7-9-21
22	1-22-23
23	1-50(ZÚ)-51(ZÚ)
24	1-22-24-25
25	1-22-24-26-27
26	1-22-24-54(ZÚ)
27	1-22-24-26-28-29
28	1-22-24-26-28-30-32
29	1-2-3-55
30	1-2-3-56
31	1-57-58
32	1-57-59-60
33	1-57-59-61

## Testy průchodů procesem zaměstnancem

- **Kombinace pro testy úrovně pokrytí 1**

Tabulka 1: Průchody procesem - kombinace pro testy úrovně pokrytí 1

Kombinace pro testy úrovně pokrytí 1	
Větvící bod	Výstupy
Vstup do aplikace	
PÚ	2, 5, 10, 11, 16, 21
JR	3, 4
ÚAV	6, 7, 8, 9
ÚZV	12, 13, 14, 15
SHS	17, 18
NHV	19, 20
VRU	22, 23
VRZ	24, 25
úpravy uloženy	
zaměstnanec nepřihlášen	
úpravy neuloženy	
auto přidáno	
auto nepřidáno	
auto deaktivováno	
zaměstnanec přidán	
zaměstnanec nepřidán	
účet deaktivován	
účet není deaktivován	

- **Vytvoření testů z kombinací pro úroveň pokrytí 1 (pokrytí cest)**

Tabulka 2: Vytvoření testů z kombinací pro úroveň pokrytí 1 (pokrytí cest)

	Průchod
1	1-2-4
2	1-2-3-5-6
3	1-5-7
4	1-5-8
5	1-5-9



	<b>Průchod</b>
6	1-10
7	1-11-12
8	1-11-13
9	1-11-14
10	1-11-15
11	1-16-17
12	1-16-18-19
13	1-16-18-20
14	1-21-22
15	1-21-23-24
16	1-21-23-25