

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Mobilná aplikácia - Autoservis

Mobilné technológie a aplikácie

Autori: Matej Lánik, Jakub Sorád

Cvičenia: Štvrtok 8:00

Prednášajúci: doc. Ing. Peter Trúchly, PhD.

Akademický rok: 2021/22

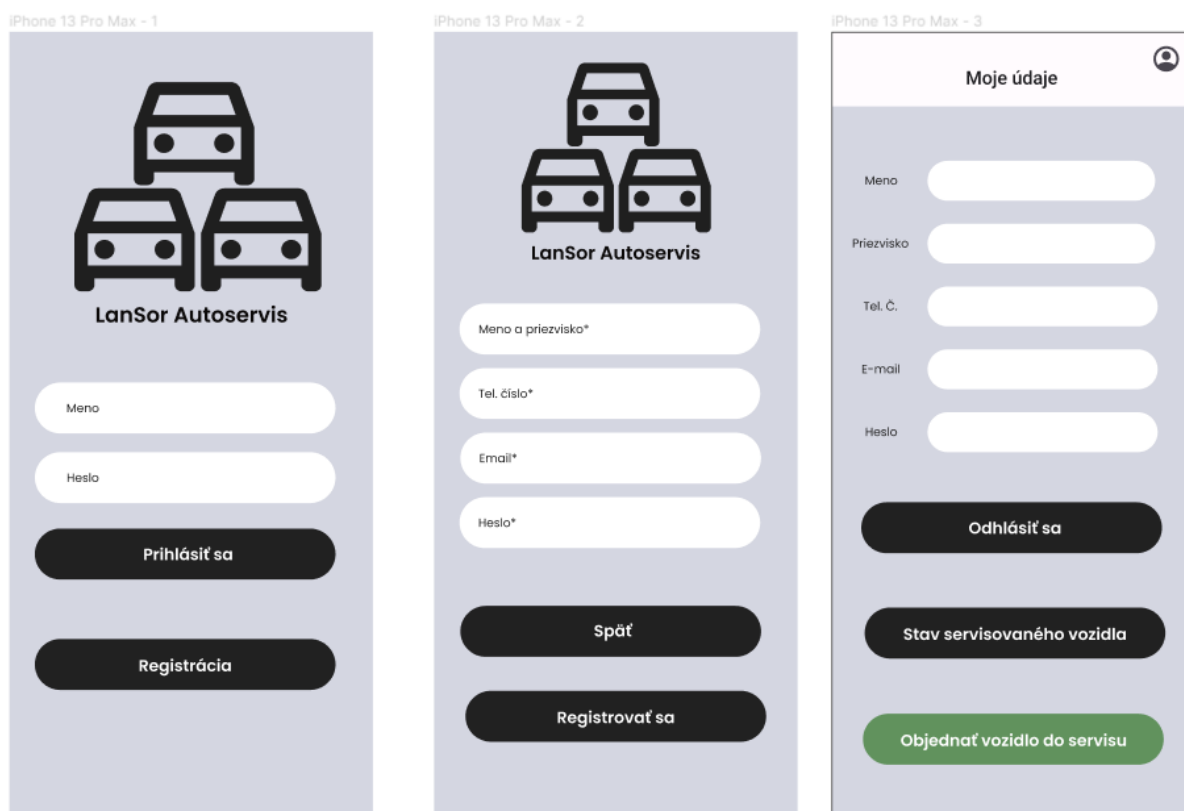
Obsah

Návrh wireframov	2
Návrh databázy	5
Návrh API endpointov	6
Akceptačné testy	8
Frontend aplikácie	8
Kladné testy	8
Záporné testy	11
Backend	13
Kladné testy	13
Záporné testy	15

Návrh wireframov

Použili sme nástroj [Figma](#), kde sme vytvorili jednotlivé wireframy a tiež aj user flows, takže náš návrh je aj interaktívny. Po konzultácií sme zapracovali na pripomienkach a možných zlepšeniach, medzi ne patria aj tieto dve:

- Pri scenári, keď sa prihlasuje technik sme použili inú úvodnú obrazovku, ale dopredu nevieme, kto sa bude prihlasovať, či zákazník alebo technik. Preto bude platiť pre oboch typoch používateľoch jednotná prihlasovacia obrazovka.
- Keď technik opraví auto, potvrdí to v aplikácii a následne aj zákazník potvrdí opravu auta, tak sme vymazali dané auto z databázy. V našej aplikácii by bolo veľmi vhodné si práve tieto opravené autá ukladať do histórie opravených áut. Túto históriu si potom technik môže pozrieť po prihlásení do aplikácie.



iPhone 13 Pro Max - 4

Objednať vozidlo do servisu

Značka*

Model*

Rok výroby*

☐ Želám si aby servis môjho vozidla vykonával konkrétny technik.

Späť

iPhone 13 Pro Max - 7

Zvoľte požadované úkony

☒ Výmena motorového oleja
☒ Výmena filtrov
☒ Výmena pneumatík
☒ Servis motora

Doplňujúce informácie...

Späť

Nahrať fotografie vozidla

iPhone 13 Pro Max - 8

Nahrať fotografiu vozidla

Výbrať obrázok

Späť

Objednať vozidlo do servisu

iPhone 13 Pro Max - 11

Stav vozidla

Značka

Model

Pridelený technik

Servis vozidla bol dokončený
Potvrdiť vyzdvihnutie vozidla

iPhone 13 Pro Max - 6

Stav vozidla

Značka

Model


Pridelený technik

Stav vozidla

Začať videohovor s technikom

Pokračovať

login



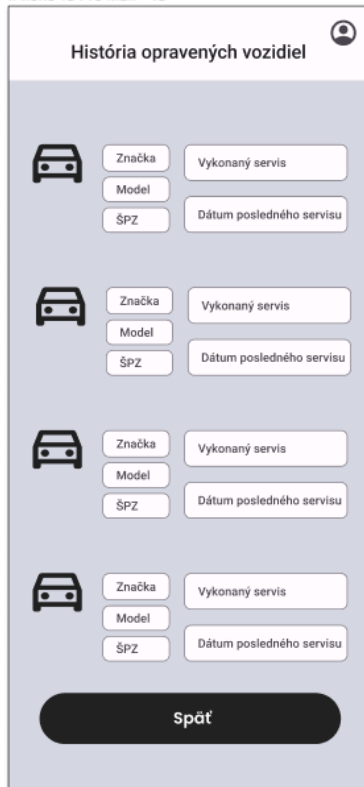
LanSor Autoservis

Meno

Heslo

Prihlásiť sa

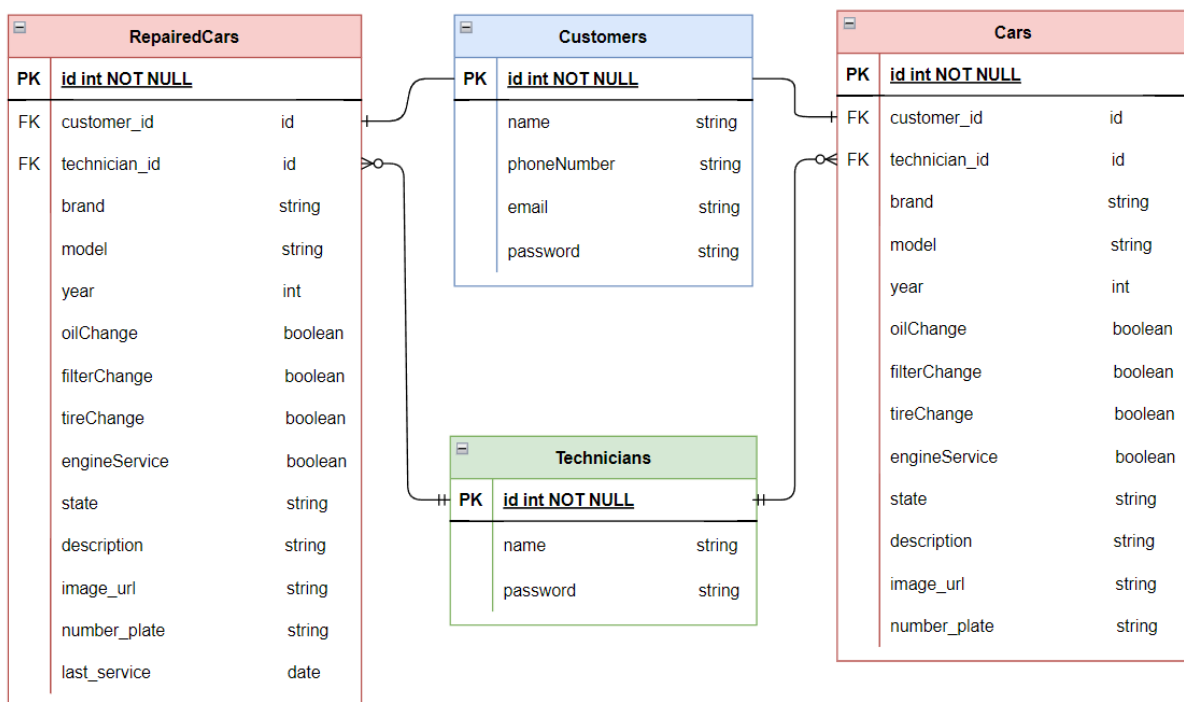
Registrácia



Návrh databázy

Súčasťou projektu mobilnej aplikácie je návrh fyzického modelu databázy, ktorý použijeme. Na vytvorenie návrhu sme použili nástroj draw.io, kde sme jednotlivé schémy vedeli navrhnuť a tiež aj vzťahy medzi nimi.

Po zapracovaní na pripomienke ohľadom histórie opravených áut sme upravili náš databázový model. Použijeme ďalšiu schému, ktorá slúži na uloženie už opravených áut. Bude tam aj pridaný údaj kedy prebehol posledný servis alebo oprava auta, čo môže pomôcť technikovi pri riešení problémov s autom. Taktiež prebehli aj drobné úpravy v jednotlivých schémach, kde u zákazníka ani technika nemali uložené heslo pre prístup od aplikácie. Ďalej sme pridalí evidenčné číslo pre každé auto.



Návrh API endpointov

Použili sme nástroj [Swagger](#), pre správu API dokumentácií. Poskytuje množstvo funkcionalít, ktoré nám umožnia v prehľadnej forme poskytovať údaje a detaily o API volaniach a endpointoch, ktoré budú súčasťou nášho projektu.

Zoznam používateľov ^		
GET	/api/v1/customers	Vráti zoznam všetkých používateľov. ▾
Vkladanie používateľa do DB ^		
POST	/api/v1/customers	Vytvorí nový záznam používateľa v databáze ▾
Zobrazenie informácií o používateľovi ^		
GET	/api/v1/customers/{id}	Vráti informácie o používateľovi ▾
Zoznam vozidiel ^		
GET	/api/v1/cars	Vráti zoznam všetkých vozidiel ▾
Vloženie auta do DB ^		
POST	/api/v1/cars	Vytvorí nový záznam používateľa v databáze ▾
Zoznam opravených vozidiel ^		
GET	/api/v1/repairedCars	Vráti zoznam všetkých vozidiel ▾
Vloženie auta do histórie DB ^		
POST	/api/v1/repairedCars	Vytvorí nový záznam používateľa v databáze ▾
Zobrazenie informácií o vozidle ^		
GET	/api/v1/cars/{id}	Vráti údaje o vozidle ▾
Odstránenie vozidla z DB ^		
DELETE	/api/v1/cars/{id}	Odstráni vozidlo z DB ▾
Úprava stavu vozidla ^		
PUT	/api/v1/cars/{id}	Upraví údaje o Vozidle ▾
Zoznam technikov ^		
GET	/api/v1/technicians	Vráti zoznam všetkých technikov ▾
Zobrazenie informácií o technikovi ^		
GET	/api/v1/technicians/{id}	Vráti údaje o technikovi ▾

Schemas	
Cars >	↩
Customers >	↩
Technicians >	↩
RepairedCars >	↩
RepairedCars201 >	↩
RepairedCars422 >	↩
Cars201 >	↩
Cars422 >	↩
Customers201 >	↩
CustomersID200 >	↩
CustomersID404 >	↩
CarsID404 >	↩
Technicians404 >	↩
CarsDELETE200 >	↩

Príklad API volania a jeho detailov. Podobne sú vypracované všetky API volania.

Zoznam používateľov

GET /api/v1/customers Vráti zoznam všetkých používateľov.

Bude volaný vtedy, keď budeme potrebovať získať zoznam používateľov kvôli prihláseniu.

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	<p>Pole všetkých používateľov</p> <p>Media type</p> <p>application/json</p> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre> { "items": [{ "id": 1, "name": "misko", "phoneNumber": "9045876541", "email": "misko@miskovemail.com", "password": "silneheslo1337" }] } </pre>	No links

Akceptačné testy

Vytvorili sme dokopy 10 akceptačných testov pre frontendovú časť aplikácie a tiež pre backend aplikácie. Rozdelili sme testy na dva typy, kedy správanie používateľa je kladné a kedy je záporné. Podľa toho dostávame rôzne výstupy pri interakcii s našou aplikáciou.

Frontend aplikácie

Kladné testy

Test 1: Prihlásenie sa	
Vstupné podmienky:	Používateľ, ktorý má vytvorený účet a nachádza sa v databáze.
Výstupné podmienky:	Používateľ sa úspešne prihlási do svojho účtu.
Postup:	<ol style="list-style-type: none">1. Používateľ zadá svoje prihlasovacie údaje.2. Používateľ stlačí tlačidlo PRIHLÁSIŤ SA.3. Používateľovi sa zobrazí obrazovka s jeho údajmi a možnosťou objednania sa do servisu.
Výsledok: PASS / FAIL	

Test 2: Vyplnenie požadovaných servisných úkonov.	
Vstupné podmienky:	Používateľ, ktorý zadal základné informácie o jeho aute a stlačil tlačidlo POKRAČOVAŤ.
Výstupné podmienky:	Zákazník vyplnil formulár s požiadavkami o servise a prechádza na posledný krok objednávky do servisu.
Postup:	<ol style="list-style-type: none"> 1. Používateľ zaškrtnie checkboxy všetkých požadovaných úkonov. 2. Ak používateľ nemá žiadne konkrétnejšie požiadavky k servisu, preskočiť na krok 4. 3. Ak používateľ má ďalšie servisné požiadavky, napíše ich do poľa DOPLŇUJÚCE INFORMÁCIE. 4. Po vyplnení všetkých požadovaných servisných úkonov klikne používateľ na tlačidlo NAHRAŤ FOTOGRAFIE VOZIDLA. 5. Používateľovi sa zobrazí obrazovka, v ktorej musí nahrať fotografiu/fotografie vozidla.
Výsledok: PASS / FAIL	

Test 3: Vloženie fotografií servisovaného vozidla	
Vstupné podmienky:	Používateľ, ktorý vyplnil všetky potrebné informácie o servisnom úkone.
Výstupné podmienky:	Používateľove auto bude zaradené do databázy. Servisovanie auta následne bude pridelené požadovanému technikovi(ak si takého používateľ zvolil) a zákazník si bude môcť pozrieť stav opravy vozidla.
Postup:	<ol style="list-style-type: none"> 1. Používateľ klikne na tlačidlo VYBRAŤ OBRÁZOK. 2. Používateľ vloží fotografiu/fotografie jeho vozidla. 3. Používateľovi sa zobrazia vložené fotografie na obrazovke. 4. Používateľ klikne na OBJEDNAŤ VOZIDLO DO SERVISU. 5. Používateľovi je zobrazená začiatková obrazovka s jeho údajmi, kde už môže skontrolovať stav jeho servisovaného vozidla.
Výsledok: PASS / FAIL	

Záporné testy

Test 4: Vytvorenie nového účtu	
Vstupné podmienky:	Používateľ, ktorý už je zaregistrovaný sa nachádza v login obrazovke aplikácie.
Výstupné podmienky:	Používateľ sa bude musieť vrátiť na prihlasovaciu obrazovku, keďže účet už vytvorený mal.
Postup:	<ol style="list-style-type: none">1. Používateľ stlačí tlačidlo REGISTRÁCIA.2. Zobrazí sa registračný formulár.3. Používateľ vyplní všetky potrebné polia, no zadá tel.č. alebo e-mail, ktorý už existuje v databáze.4. Používateľ klikne na tlačidlo REGISTROVAŤ.5. Používateľ bude upozornený, že tento účet už existuje.6. Používateľ stlačí tlačidlo Späť.
Výsledok: PASS / FAIL	

Test 5: Vyplnenie základných informácií o vozidle	
Vstupné podmienky:	Používateľ, ktorý je registrovaný a chce objednať svoje auto do servisu.
Výstupné podmienky:	Zákazník úspešne informácie o jeho vozidle, a prejde na druhý krok objednávky do servisu. V prípade nevyplnenia jedného z povinných polí je používateľ upozornený.
Postup:	<ol style="list-style-type: none"> 1. Používateľ stlačí tlačidlo OBJEDNAŤ VOZIDLO DO SERVISU. 2. Používateľovi sa zobrazí formulár, v ktorom musí vyplniť informácie o jeho aute. 3. Používateľ vyplní formulár, s tým že zabudne zadať niektoré z povinných polí. 4. Ak používateľ nemá požiadavku vybrať si konkrétneho technika, preskočiť na krok 7. 5. Ak si používateľ žiada konkrétneho technika, musí zaškrtnúť checkbox. 6. Zákazník si zo zobrazeného dropdown menu zvolí preferovaného technika, ktorému bude servis pridelený. 7. Zákazník klikne na tlačidlo POKRAČOVAŤ. 8. Zákazník bude upozornený, že zabudol vyplniť všetky povinné údaje. 9. Zákazník po upozornení vyplní potrebné údaje, preskočiť na krok 7.
Výsledok: PASS / FAIL	

Backend

Kladné testy

Test 6: Správna registrácia	
Vstupné podmienky:	Pri registrácii užívateľ užívateľ polia.
Výstupné podmienky:	Užívateľské údaje sú zapísané v databáze.
Postup:	<ol style="list-style-type: none">1. Je odoslaná POST požiadavka na endpoint <code>/api/v1/customers</code>, ktorá obsahuje request body s vyplnenými údajmi2. V backende sa spracujú údaje a skontroluje sa, či povinné polia majú správny tvar a boli vyplnené3. Ak boli správne vyplnené, tak vraciame response so status kódom 200, inak príde odpoveď vo formáte JSON s informáciou, aká nastala chyba
Výsledok: PASS / FAIL	

Test 7: Dokončený servis auta	
Vstupné podmienky:	Používateľ má auto v servise, je opravené a pripravené na vyzdvihnutie.
Výstupné podmienky:	Opravené auto sa vymaže z databázy áut, ktoré sú servisované.
Postup:	<ol style="list-style-type: none"> 1. Je odoslaná DELETE požiadavka na endpoint <code>/api/v1/cars/:id</code> 2. Obsahuje request body s údajmi o konkrétnom aute, ktoré databáza spracuje 3. Následne príde response s kódom 200 a prázdny body, prípade úspešného odstránenia. Inak príde informácia v response body o chybe, že dané aute nie je evidované
Výsledok: PASS / FAIL	

Test 8: Získanie údajov o aute zákazníka	
Vstupné podmienky:	Používateľ má aute momentálne v servise.
Výstupné podmienky:	Používateľ vidí stav jeho auta, aké úkony sa vykonali na jeho aute.

Postup:	<ol style="list-style-type: none"> 1. Je odoslaná GET požiadavka na endpoint <code>/api/v1/cars/:id.</code> 2. V backende sa request spracuje a hľadá sa auto s daným ID. 3. Ak sa auto s konkrétnym a jeho údaje sa vrátia v reponse body spolu s kódom 200, v prípade ak také auto nie je v databáze, vracia sa chybový kód a prázne response body
Výsledok: PASS / FAIL	

Záporné testy

Test 9: Nepridaná fotografia auta	
Vstupné podmienky:	Používateľ pri objednávke nepridá fotografiu auta
Výstupné podmienky:	Správa o tom, že je potrebné nahrat' fotografiu
Postup:	<ol style="list-style-type: none"> 1. Je odoslaná POST požiadavka na endpoint <code>/api/v1/cars.</code> a request body s obrázkom 2. Na backende sa zistí, že nie je request body je prázdne 3. Posiela sa response body s chybovým kódom a správou, že nebol pridaný obrázok
Výsledok: FAIL	

Test 10: Nekompletné vyplnenie údajov auta	
Vstupné podmienky:	Používateľ zadáva objednávku, ale zabudol vyplniť správne rok výroby auta, čo je povinný údaj.
Výstupné podmienky:	Používateľovi sa zobrazí chybová správa o tom, že je potrebné tento údaj vyplniť.
Postup:	<ol style="list-style-type: none"> 1. Je odoslaná POST požiadavka na endpoint <code>/api/v1/cars</code>. 2. Na backende sa zistí, že nie sú vyplnené všetky povinné polia pri pridávaní auta a jeho údajov do databázy. 3. Posiela sa response body s chybovým kódom a správou, údaje o aute neboli úspešne registrované.
Výsledok: FAIL	