



Zpráva o návrhu

Tvorba uživatelských rozhraní (ITU 23/24Z)

Gabriel Biel
Adam Gabrys
Jakub Mikyšek

xbielg00
xgabry01
xmikys03

1. Navržené téma

Duosomethingo - Aplikace pro sebevzdělání (xmikys03)

Můj nápad byl vytvořit aplikaci rozšiřující a spojující vzdělávací aplikace jako Duolingo (pro cizí jazyky) a Flashcards (pro zapamatování pojmů). Celá aplikace by měla pojmut témata primárně v oblastech IT světa, například - Řecká abeceda, rozpoznání elektrotechnických součástek, procvičování návrhových vzorů. Způsob procvičování by byl od formy vybírání z 4 možností, přes spojování pojmů až po doplňování názvů uživatelem.

CarTrack - Elektronická kniha jízd (xgabry01)

Nápad vznikl se zaměřením na profesionální řidiče, kdy jsem je nejednou slyšel stěžovat si na zápis do papírové knihy jízd. S tímto přišel nápad na vytvoření elektronické knihy jízd pro firmy/profesionální řidiče. V aplikaci by bylo možné si vybrat z nabídky aut vlastněné firmou konkrétní vozidlo použité při jízdě a pouze zadat konečný stav tachometru. V aplikaci by se zobrazil počet ujetých kilometrů za jízdu, jelikož by si aplikace uchovávala všechny záznamy a odečetla by nejnovější záznam tachometru od záznamu minulé jízdy. V úvahu padla i možnost rozšířit aplikaci o informaci o konkrétních autech či tlačítko 'Potřebuji pomoc', kdy by se uživateli zobrazilo číslo na asistenční službu/ZS či jak postupovat v případě nehody.

Restauranto - Aplikace pro číšníky (xbielg00)

Mým nápadem je vytvořit aplikaci "Restauranto" pro číšníky, která zjednoduší jejich práci a zlepší zážitek hostů v restauracích. Klíčové funkce zahrnují:

Rozložení stolů: Možnost rychlé úpravy uspořádání stolů v restauraci. **Jednoduché objednávání:** Hosté mohou snadno objednat jídlo a nápoje přes aplikaci. **Přidávání poznámek:** Možnost specifikace individuálních požadavků hostů. **Dělení účtů:** Snadné dělení účtů mezi různými osobami. **Přesun stolů:** Možnost rychlého přesunu stolů podle potřeby. **Tisk účtů:** Rychlý tisk účtů pro zrychlení procesu odchodu hostů.

Tímto by se vytvořila aplikace, která by zefektivnila práci číšníků a zlepšila celkový zážitek hostů v restauracích.

2. Vybrané téma

Po zhodnocení jsme zvolili variantu firemní aplikace pro efektivní správu jízd aut na způsob Kniha jízd. Varianta se nám zdála být nejpraktičtější a zároveň nejzajímavější pro implementaci a návrh. Při výběru jsme brali v úvahu osobní zkušenosti s tímto tématem.

3. Analýza uživatelských potřeb a klíčových problémů

Otázky byly typu **ANO/NE**, **bodové ohodnocení 1**(rozhodně ne) - **5**(rozhodně ano), **výběr z odpovědí** nebo **otevřená odpověď**.

Otázky použité v dotazníku:

1. Používali jste někdy knihu jízd nebo něco tomu podobného?

Z dotazníku vidíme, že více než 2/3 respondentů má s knihou jízd nějaké zkušenosti.

2. *Používáte pro zápis služebních cest něco jiného než papírovou knihu jízd? Pokud něco jiného, co?*

Zde se objevila pouze 1 odpověď: **Elektronickou variaci v podobě Google sheetu**. Za povšimnutí stojí, že nikdo z tázaných nemá zkušenost s žádnou formou aplikace, klidně i nějaké interní.

3. *Jste spokojeni se zapisováním do papírové knihy jízd (pokud ji používáte)?*

Setkáváme se zde s celou řadou různorodých odpovědí - 2 lidé jsou spokojeni, 4 jsou neutrální, 3 se přiklání spíše, že spokojeni nejsou a 1 člověk není spokojen. Pro někoho staré dobré zapisování na arch papírů může být v pořádku a někdo to už ani nemůže vystát.

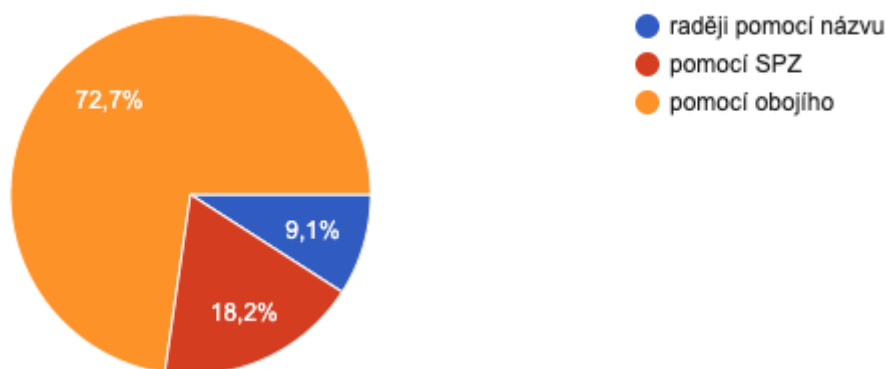
4. *Líbí se vám myšlenka jednoduchého zápisu koncového stavu kilometrů do aplikace?*

90% odpovědí bylo pro myšlenku chytrého zapisování v aplikaci, má to smysl!

5. *Vnímáte jako výhodu mít v aplikaci všechna firemní auta pohromadě a jednoduše si z nabídky pouze vybrat auto, které jste použili?*

U této odpovědi, stejně jako u té předchozí se setkáváme s velmi kladnou vlnou odpovědí **přes 80% by tuto funkci určitě uvítalo**. Tuto funkcionalitu bude tedy určitě dobré naimplementovat.

6. *Prováděli byste raději výběr auta z nabídky pomocí firmou zvoleného názvu nebo pomocí SPZ?*

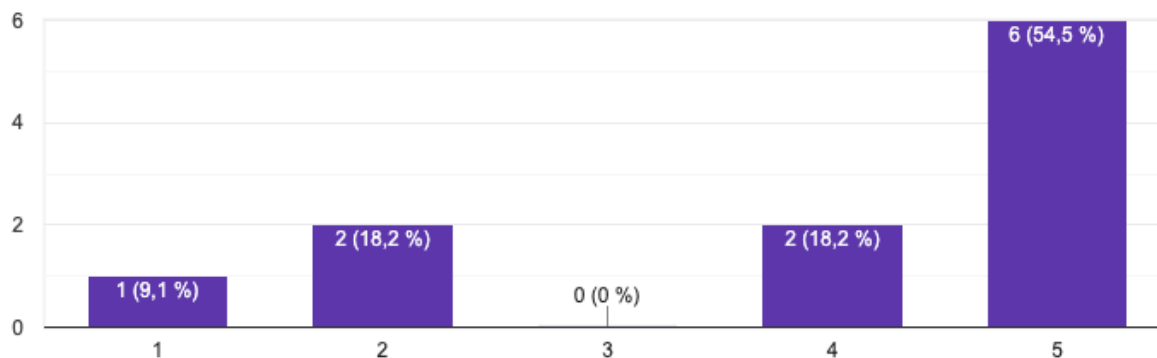


Nejlepší volbou z pohledu uživatelů je vyhledávání pomocí SPZ i názvu auta.

7. *Vnímáte jako výhodu si pouze vybrat vašeho uživatele z nabídky aplikace místo repetitivního zapisování vašeho jména do knihy jízd?*

Opětovně se **90%** dotazovaných vyjádřilo **velmi pozitivně**, jen 1 zůstal neutrální. Výběr uživatele v aplikaci je důležitá funkce pro pohodlné používání.

8. *Ocenili byste mít v aplikaci číslo na asistenční službu, kterou máte placenou z vašeho povinného ručení a tím předejít placení navíc/hledání jakou službu firma používá?*



Zde už popularita klesala, i přesto by o to někteří měli zájem.

9. *Ocenili byste kdyby aplikace umožňovala také uvádět záznamy o tankování jednotlivých aut?*

Tady již se vyjádřilo pozitivně trochu více lidí (přes 80% hlasovalo pro hodnocení 4 nebo lepší). V otevřené reakci na konci dotazníku jsme získali dobrý podnět, že by bylo žádoucí vidět i momentální stav nádrže - ne jenom to, kdy bylo naposledy natankováno.

10. *Ocenili byste možnost přidání poznámek k jednotlivým autům? (např. zanechání vzkazu pro ostatní řidiče, kde jste uložili technický průkaz či přidání záznamu o poruše/nehodě)*

Pozitivně byla hodnocena i tato funkčnost, pro hodnocení 5 hlasovalo 65% respondentů (pozitivní přes 80%).

11. *Ocenili byste tlačítko "Potřebuji pomoci", kdy by se vám po stisknutí zobrazilo info, jak postupovat v případě nehody/poruchy i např. s číslem na asistenční službu či ZS.*

U této otázky se naopak vyjádřilo pozitivně **jen 50% respondentů**, zbytek u tuto funkci nemá zájem.

12. *Líbilo by se vám, kdyby si aplikace uchovávala stav tachometru z předchozích jízd a při zadání konečného stavu kilometrů po vaší jízdě vám zobrazila počet nově najetých kilometrů?*

Naopak tato otázka sklidila **90% pozitivních reakcí** (80% pro hodnocení 5). Případní uživatelé by o tuto funkci, která by eliminovala potřebu něco dopočítávat "z hlavy" nebo na kalkulačce, měli opravdu zájem.

13. *Ocenili byste mít možnost zobrazit si detail auta? Jednalo by se např. o typ paliva (DIESEL/BENZÍN).*

Detail vozidla se zdá být také důležitý (5 - 72%, 4 - 18%), s čísel jde vidět, že by funkce mohla být žádaná.

14. *Uvítali byste ještě nějakou další funkcionalitu aplikace kromě výše popsanych?*

Respondenti zvažovali následující rozšiřující funkcionality:

- Zapsání účelu jízdy
- Možnost zpětného editování
- Zobrazení historie používání vozidla
- Zobrazení stavu nádrže (málo, hodně...)

Respondenti:

- xmikys03: Michal B. (bývalý řidič rozvozu KFC), David Š. (COO v ARBO, jízda firemním autem s kameramani pro skenování stromů), Samuel Č. (Wolt řidič), Patrik K. (bezpečností specialista v Red Hat), Michal H. (PHP developer)
- xgabry01: Jan K. (profesionální řidič nákladních vozidel), Veronika K. (řidička pro nemocniční zásobování)
- xbielg00: Ludmila M. (zdravotní sestra v terénu), Patrik Š. (podnikatel)

4. Analýza existující aplikace

Papírová podoba (xgabry01)

Nejklasičtější podoba Knihy jízd, která se nachází v každém autě firmy, pokud tuto podobu používá. Tyto knihy jsou obvykle v tištěné formě a jsou používány pro účely evidování a správy firemních nebo osobních vozidel. Je třeba do ni ručně zapisovat všechny údaje a velkou nevýhodou je nezálohovatelnost těchto údajů v případě ztráty jednotlivých knih. Každý uživatel má také jiný rukopis, takže je také někdy obtížné dané záznamy přečíst. Výhodou je však pořízovací cena a snadné zacházení. Díky aplikaci, kterou chceme implementovat se snažíme všechny tyto problémy eliminovat a to jak naimplementovat jednoduché zadávání koncového stavu kilometrů, tak jistou čitelnost údajů i zálohovanost dat.

Excel Tabulka (xmikys03)

Implementací tohoto problémů není opravdu mnoho, jedno z řešení může být sdílená excelová tabulka. Mezi přednosti bych uvedl snadnou přenositelnost a přístupnost napříč všemi chytrými zařízeními bez nutnosti instalace. Oproti papírové podobě ještě můžeme vypíchnout již nějakou formu digitalizace. Mimo to se ale jedná o variantu, která nemusí být vždy intuitivní a jednoduše může dojít k poškození či **ztrátě/smazání dat, při špatné manipulaci uživatelem**. Řešení naší aplikace: Uživatel přidává data a nemůže editovat nebo mazat data, která již byla napsána jinými uživateli.

Elektronická kniha jízd (xbielg00)

[Satelitní Sledování CZ - Elektronická kniha jízd](#)

Elektronická kniha jízd je produkt, který umožňuje monitorovat a evidovat jízdy vozidel pomocí GPS lokátorů. Produkt má některé funkce, které se zdají být užitečné jako je automatické zpracování dat, rozlišení typu a účelu jízdy, zobrazení jízd na mapě, pravidelné odesílání dat do účtárny, mobilní aplikace a připomínky a upozornění na servis a pojištění.

Nicméně, produkt má také mnoho nedostatků. Mezi jeho nedostatky patří nutnost instalace GPS lokátorů do vozidel, nutnost spoléhat na kvalitu a dostupnost signálu GPS.

Produkt má různé ceny podle typu a funkčnosti GPS lokátorů, které se pohybují od 150 Kč do 200 Kč za měsíc za každý lokátor, plus jednorázová cena za koupi lokátoru, která se pohybují od 2 390 Kč do 5 590 Kč.

5. Uživatelské potřeby a klíčové problémy

Na základě provedené analýzy uživatelských potřeb a existujících řešení jsme identifikovali několik klíčových potřeb a problémů, které naše aplikace bude řešit:

1. **Snadný a efektivní zápis jízd:** Uživatelé vyjádřili potřebu snadného a rychlého zaznamenání jízd, což zahrnuje jednoduché zadání koncového stavu tachometru.
2. **Sledování ujetých kilometrů:** Uživatelé chtějí mít přehled o tom, kolik kilometrů ujeli při každé jízdě, a to bez složitého ručního výpočtu.
3. **Centrální evidenci vozidel:** Profesionální řidiči ocenili možnost mít veškerá firemní vozidla pohromadě v jedné aplikaci a snadno vybírat konkrétní vozidlo pro každou jízdu.
4. **Asistenční služba a informace v případě potřeby:** Umožnění rychlého přístupu k asistenčním službám nebo informacím o postupu v případě nehody nebo poruchy vozidla bylo vnímáno jako užitečná funkce.
5. **Historie jízd a rychlé vyhledávání:** Uživatelé chtějí mít možnost sledovat historii svých jízd a rychle vyhledávat konkrétní záznamy.
6. **Detaily o vozidle:** Možnost zobrazit informace o jednotlivých vozidlech, včetně typu paliva, je pro uživatele důležitá.
7. **Poznámky a komunikace mezi uživateli:** Někteří uživatelé chtějí možnost přidávání poznámek nebo komunikace ohledně vozidel a jízd.

Tyto klíčové potřeby a problémy budou zohledněny při návrhu a vývoji naší aplikace, abychom poskytli uživatelům efektivní a uživatelsky přívětivé řešení pro správu knihy jízd a souvisejících úkolů.

6. Návrh aplikace - Rozdělení práce

Rozhodli jsme se pro 2. způsob - rozdělení částí aplikace do jednoho společného celku.

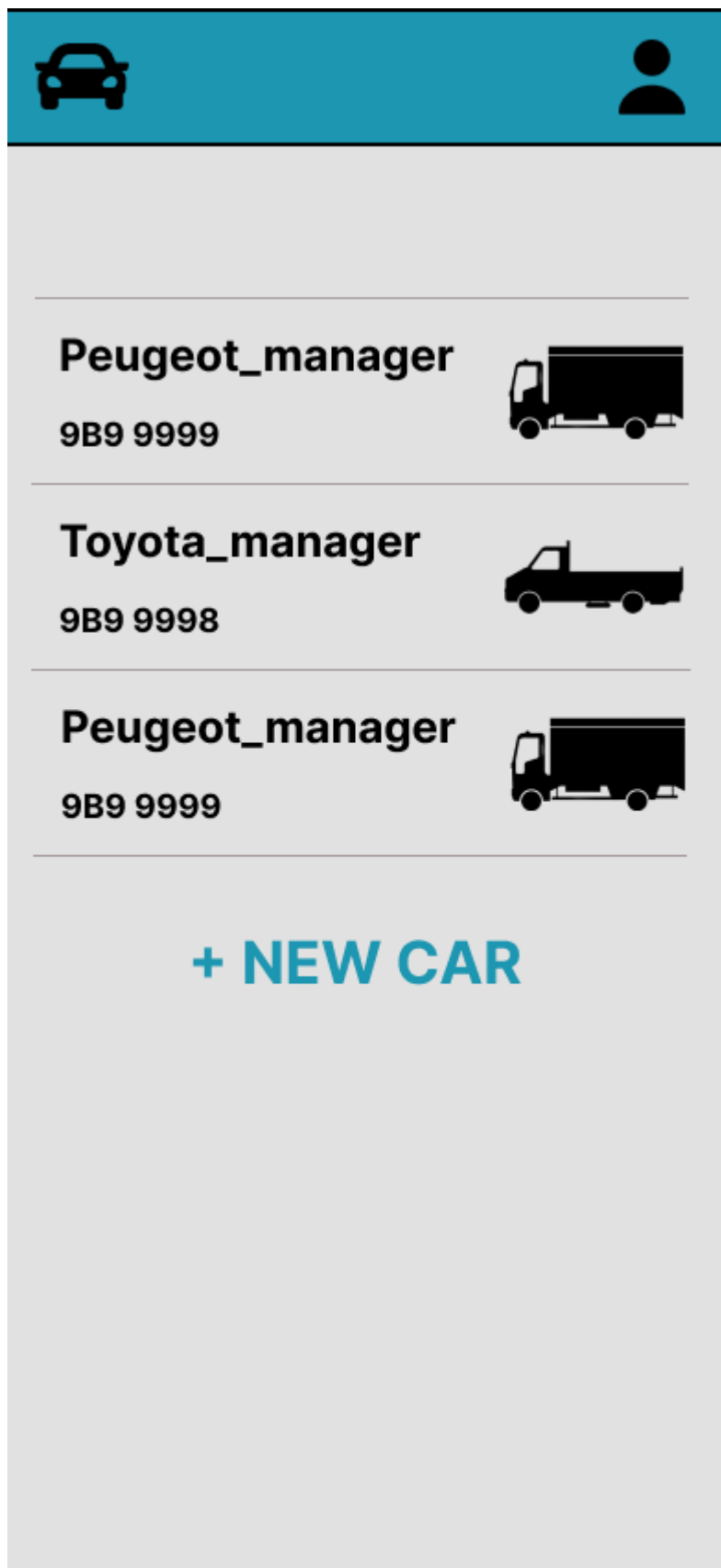
Rozdělení práce:

- xmikys03: menu pro výběr aut, editace a přidání aut
- xgabry01: základní komponenty, uživatel - registrace a výběr
- xbielg00: zapisování kilometrů a detaily s tím spojené

The image displays 20 mobile app wireframes for 'CAR TRACK', organized into four rows of five. The wireframes illustrate the following screens and features:

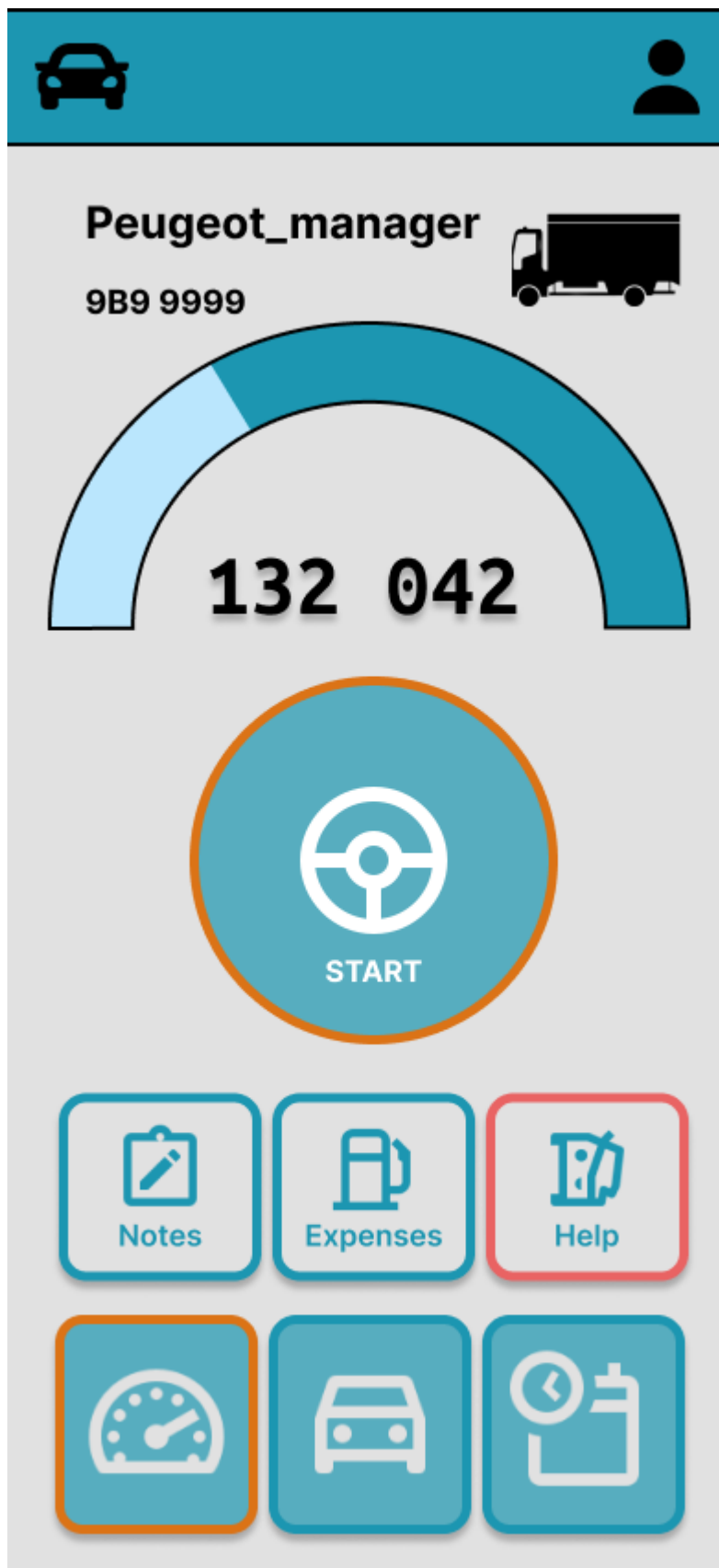
- Row 1:**
 - Screen 1: Login screen with 'CAR TRACK' logo, 'Enter your company code' field, and 'LOGIN'/'CREATE COMPANY' buttons.
 - Screen 2: User selection screen with 'Choose user' dropdown, 'LOGIN'/'CREATE USER' buttons.
 - Screen 3: User details screen showing 'User_name', 'Telephone number', and a 'DELETE USER' button.
 - Screen 4: Confirmation screen asking 'Are you sure?' with 'YES'/'NO' buttons.
- Row 2:**
 - Screen 5: Login screen (duplicate of Row 1, Screen 1).
 - Screen 6: User selection screen (duplicate of Row 1, Screen 2).
 - Screen 7: User selection screen with 'Choose user' dropdown and 'user_1', 'user_2' buttons.
 - Screen 8: User selection screen with 'user_x' dropdown and 'LOGIN'/'CREATE USER' buttons.
 - Screen 9: Vehicle management screen showing ' Peugeot_manager user xxxx' with vehicle icons and a '+ NEW CAR' button.
 - Screen 10: Vehicle details screen with fields for 'VIN-plate', 'Car license plate', 'Old name', 'Car type', and 'Part info', plus an 'ADD CAR' button.
- Row 3:**
 - Screen 11: Login screen with 'CAR TRACK' logo and 'GENERATE CODE' button.
 - Screen 12: Confirmation screen stating 'Your CODE has been generated!' with a 'CONTINUE' button.
 - Screen 13: User registration form with 'Name *', 'Telephone number', and 'ADD USER' button.
 - Screen 14: User selection screen with 'user_x', 'user_1', 'user_2' buttons and 'LOGIN'/'CREATE USER' buttons.
 - Screen 15: 'Fresh Drive' screen showing a speedometer at '132 042', a 'FRESH DRIVE' button, and a 'START 10-100 1717 00.00s' timer.
 - Screen 16: 'You drove 121 km' screen with 'Type of file', 'Start destination', and 'End destination' fields, plus a 'CONFIRM' button.
 - Screen 17: 'Fresh Drive' screen (duplicate of Row 3, Screen 15).
 - Screen 18: 'You drove 121 km' screen (duplicate of Row 3, Screen 16).
- Row 4:**
 - Screen 19: User registration form (duplicate of Row 3, Screen 13).
 - Screen 20: User registration form (duplicate of Row 3, Screen 13).
 - Screen 21: Vehicle management screen showing ' Peugeot_manager user xxxx' with a speedometer at '132 042' and an 'ADD' button.
 - Screen 22: Insurance details screen for ' Peugeot_manager' showing 'License plate', 'Vehicle', 'Insurance', 'Responsible person', and 'Contact info'.
 - Screen 23: Vehicle management screen (duplicate of Row 4, Screen 21).
 - Screen 24: Insurance details screen (duplicate of Row 4, Screen 22).

Přehled celkové namodelované makety



Centrální evidence vozidel

Uživatelé si přáli jednoduše vybrat jedno z firemních vozidel, k tomu jsme vytvořili i možnost přidat nová firemní auta.



Karta pro začátek jízdy a rozcestník pro práci s knihou jízd po vybrání auta

Náhledová stránka s přehledem stavu tachometru, detailu vozidla i možnost přidání poznámek/záznamů tankování.

The screenshot displays the Peugeot_manager application interface. At the top, there is a blue header bar with a car icon on the left and a user profile icon on the right. Below the header, the text "Peugeot_manager" is prominently displayed, followed by the license plate "9B9 9999" and a truck icon. A large, semi-circular speedometer graphic is centered, showing a reading of "132 042". Below the speedometer, the text "Enter kilometers" is followed by a white input field containing six dots. An orange button labeled "FINISH DRIVE" is positioned below the input field. Further down, the text "Start: DD-MM-YYYY HH:MM" is displayed. The bottom section features a grid of six square buttons with icons: "Notes" (notepad icon), "Expenses" (gas pump icon), "Help" (question mark icon), a speedometer icon, a car icon, and a clock/calendar icon. The "Help" button is highlighted with a red border, and the speedometer icon button is highlighted with an orange border.

Stav po započetí jízdy

Potom co začneme jízdu se nám zobrazí následující obrazovka, kde pro dokončení jízdy zadáme údaj z tachometru a stiskneme tlačítko "FINISH DRIVE".



The image shows a mobile application interface for a ride summary. At the top, there is a blue header bar with a car icon on the left and a user profile icon on the right. The main content area has a light gray background. It displays the text "You drove" in white, followed by "121 km" in large blue font. Below this, there are three white input fields with black borders, each preceded by a label: "Type of ride", "Start destination", and "Finish destination". At the bottom, there are two buttons: a white button with an orange border and the text "CONFIRM" in blue, and a red square button with a white exclamation mark inside a triangle.

Souhrn po dokončení jízdy

Zobrazení počtu ujetých kilometrů společně se záměrem a destinací jízdy + zobrazení tlačítka s informacemi o asistenční službě

8. Návrh aplikace – Testování

Uživatelům, kterým bylo umožněno testovat náš návrh aplikace byla zpřístupněna funkční maketa ve webové aplikaci Figma.

Nedostatky odhalené při používání se vyskytly v následujících scénářích:

- **Nemožnost se vrátit o krok zpět**
 - Častá nespokojenost směřovala k nemožnosti vrátit se zpátky, uživatelé se zaseknuli a nevěděli, jak v programu pokračovat, v tomto ohledu budeme muset doimplementovat možnost nějaké zpětné šipky pro vrácení se o krok dozadu, je možné implementovat i z hlediska gesta (potáhnutí doprava, jak jsou standardně novodobé OS telefonů tvořeny)
- **Hláška o provedení akce**
 - Připomínka od uživatele David Š. směřovala na nedostatek/až absenci chybových a úspěšných hlášek, podle testera bylo dobře zvolené červené podbarvení tlačítka v chybovém stavu, ale navíc by si přál získat i text s konkrétní chybou, to stejné pro úspěch např. vytvoření/smazání uživatele nám bude doprovázet hláška o úspěchu
- **Design Input Fields**
 - Uživatel Michal A. (FE vývojář) namítl, že mu nepřijde zřejmé jaké pole jsou možné editovat v kartě detailu uživatele, bylo by tedy dobré v implementované verzi tuto designovou chybkou ošetřit, ať je opravdu zřejmé.

Testeři: Michal B., David Š., Jan K., Patrik Š., Michal A. (bližší informace o uživatelích poskytnuty v sekci pro dotazníky)

9. Technické řešení + technologie

Pro náš projekt jsme jednoznačně zvolili technologii **Flutter** pro vývoj mobilní aplikace. Flutter je známý svou výkonností, rychlostí vývoje a schopností poskytnout křížovou kompatibilitu mezi *iOS* a *Android* platformami.

Architektura naší aplikace bude postavena na návrhovém vzoru **MVC** (Model-View-Controller). **Model** bude obsahovat *datové struktury* pro ukládání informací o jízdách, vozidlech a uživatelích, a bude propojen s **Firestore databází**. Bude také zahrnovat funkce pro zpracování dat, výpočet kilometrů a uchovávání historie jízd. **View** bude zodpovědné za vizuální reprezentaci aplikace a uživatelské rozhraní, zatímco **Controller** bude obsahovat aplikační logiku, která řídí interakci mezi modelem a view.

Architektura FE:

- **Model (Data):** V FE budeme používat datový model pro uchovávání informací o uživatelích, firmách, autech, jízdách, poznámkách a výdajích. Firma má uživatele, kteří přidávají jízdy jednotlivým autům. Uživatelé mohou také přidávat poznámky a záznamy o výdajích za auto. Auta, která existují v rámci firmy, mají několik důležitých atributů, jako je počet ujetých kilometrů a další relevantní údaje.
- **View (Uživatelské rozhraní):**
 - **Login (Přihlášení do firmy):** První obrazovka, kde uživatel zadá přihlašovací údaje pro přístup do firmy.

- **User List (Seznam uživatelů):** Seznam všech uživatelů v rámci firmy s možností výběru jednoho uživatele pro zobrazení detailů.
 - **User Edit (Úprava uživatele):** Umožní uživateli upravit svůj profil (např. jméno, kontakt).
 - **Car List (Seznam vozidel):** **Hlavní obrazovka** s výpisem všech vozidel v rámci firmy.
 - **Car Detail (Detail vozidla):** Zobrazí informace o konkrétním vozidle, včetně jeho technických údajů a historie jízd.
 - **Car Edit (Úprava vozidla):** Umožní úpravu informací o vozidle (např. SPZ, typ paliva).
 - **Add Ride (Přidání jízdy):** Obrazovka **akce**, kde uživatel, pro vybrané vozidlo, zadává údaje o jízdě.
 - **Add Expense (Přidání výdaje):** Přidání nového výdaje k vozidlu.
 - **Add Note (Přidání poznámky):** Přidání poznámky k vozidlu.
 - **List Rides (Seznam jízd):** Zobrazí seznam všech jízd auta.
- **Controller (Řídící logika):** Řídící část aplikace, která bude zprostředkovávat interakci mezi Modelem a View. Bude obsahovat funkce pro zpracování uživatelských akcí, validaci dat a komunikaci s backendem.

Architektura BE (Firebase):

- **Firebase Realtime Database (Databáze):** Použijeme Firebase Realtime Database jako databázi pro ukládání uživatelských dat, jízd, výdajů a poznámek. Ta nám poskytuje snadnou synchronizaci dat mezi FE a BE.
- **Firebase Authentication (Autentizace):** Pro případnou autentizaci uživatelů by jsme využívali Firebase Authentication.

Použití Firebase Realtime Database a API:

V rámci naší aplikace jsme se rozhodli využít *Firebase Realtime Database* pro ukládání a správu dat. Firebase Realtime Database poskytuje cloudové úložiště s reálným časem, které je vhodné pro naše potřeby. Díky této volbě **nemusíme implementovat tradiční backend server s explicitními API endpointy**, protože Firebase nám umožňuje přistupovat k datům **přímo z frontendu**.

Pro komunikaci s Firebase Realtime Database v naší aplikaci budeme používat balíček `firebase_database` pro Flutter.

Při získávání dat z databáze v naší aplikaci budeme vytvářet reference na konkrétní cesty (kolekce a dokumenty) v databázi a následně provádět operace nad těmito daty. Například pro získání dat konkrétního uživatele budeme používat následující kód:

```
import 'package:firebase_database/firebase_database.dart';

final DatabaseReference usersRef =
  FirebaseDatabase.instance.reference().child('users');

void fetchUserData(String userID) {
  usersRef.child(userID).once().then((DataSnapshot snapshot) {
    if (snapshot.value != null) {
      Map<dynamic, dynamic> data = snapshot.value;
      // Zde můžeme provádět operace s daty
      print('Jméno uživatele: ${data['name']}');
    }
  });
}
```

V tomto příkladu vytváříme referenci na cestu users v databázi a následně získáváme data konkrétního uživatele na základě jeho ID. Získaná data můžeme dále zpracovat v naší aplikaci.

Tímto způsobem budeme komunikovat s Firebase Realtime Database, a to včetně operací pro **získání, aktualizaci, přidání a mazání** dat podle potřeby v naší aplikaci.