

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Smart vertikální parkování pro elektromobilitu

Diplomová práce

Autor: Jakub Střihavka

Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Karel Mls, Ph.D

Hradec Králové

listopad 2023

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 11.11.2023

.....
Jakub Střihavka

Poděkování:

Děkuji vedoucímu diplomové práce Ing. Karlu Mlsovi, Ph.D za metodické vedení práce, cenné rady a ochotu.

Anotace

Tato diplomová práce se zaměřuje na smart vertikálního parkoviště pro elektromobilitu. V úvodu práce je čtenář seznámen s problematikou nabíjení elektrických vozidel a jejich parkování na veřejných parkovištích. Teoretická část práce představuje elektromobily, jejich nabíjení a potenciální optimalizaci nabíjení. Dále je pak seznámen s konceptem smart vertikálního parkoviště. Na tomto základě byla vytvořena mobilní a webová aplikace pro podporu takového parkoviště, které lze díky aplikaci ovládat a zadávat mu nabíjecí preference. Tyto preference mohou být použity centrálním systémem parkoviště pro optimalizaci nabíjení připojených vozidel. V aplikaci je taky možné si dopředu rezervovat parkovací místo, mít přehled o zaparkovaném vozidle a platit za služby.

Klíčová slova: SoC, Vertikální parkoviště, Elektromobil, Optimalizace, Flutter

Annotation

Title: Smart vertical parking for electromobility

This thesis focuses on smart vertical parking for electromobility. The opening section of the thesis provides an overview of the problem of charging electric vehicles and parking them in public parking lots. The theoretical part of the thesis explores the concept of electric vehicles, their charging and potential charging optimization. Then, the concept of smart vertical parking is introduced. On this basis, a mobile and web application has been developed to support such parking lots, which can be controlled and charging preferences can be given thanks to the application. These preferences can be used by the central parking system to optimize the charging of connected vehicles. The app can also be used to book a parking space in advance, keep track of the parked vehicle and pay for services.

Keywords: SoC, Vertical parking lot, Electric vehicle, Optimization, Flutter

Obsah

1	Úvod	1
2	Cíl a metodika práce	2
3	Elektromobil.....	4
3.1	Výhody elektromobilů	4
3.2	Nevýhody elektromobilů.....	4
3.3	Nabíjení elektromobilů	5
3.3.1	Domácí nabíjení	6
3.3.2	Veřejné nabíječky.....	7
3.3.3	Battery swapping.....	8
3.4	Konektory pro nabíjení elektromobilů	9
3.5	Řízení dobíjení	11
3.5.1	Proprietární.....	11
3.5.2	Lokální energetická řešení.....	12
3.5.3	Řízení centrálním systémem.....	13
4	Parkovací domy.....	15
4.1	Smart parkoviště.....	17
4.2	Automatické vertikální parkovací systémy.....	18
4.2.1	Rotační systém.....	20
4.2.2	Paletový systém.....	21
4.2.3	Bezpaletový systém	22
5	Optimalizace nabíjení více připojených vozidel	24
5.1	Rovnoměrné nabíjení.....	24
5.2	Optimalizované nabíjení v závislosti na čase příjezdu.....	25
5.3	Optimalizované nabíjení v závislosti na čase odjezdu.....	25
5.4	Nabíjení založené na čase	26
5.5	Load balancing (rozložení zátěže)	26
5.6	Nabíjecí skupiny s frontami	27
5.7	Inteligentní systém řízení nabíjení	28
5.7.1	V1G (jednosměrné chytré nabíjení).....	28

5.7.2	V2G (obousměrné chytré nabíjení – vehicle-to-grid)	29
5.7.3	V2B/V2H (vehicle-to-building, vehicle-to-home)	29
5.8	Využití obnovitelných zdrojů a externích úložišť	29
5.9	Uživatelské rozhraní a aplikace	30
5.10	Prediktivní analýza a umělá inteligence	31
5.11	Ekonomický faktor	31
6	Návrh chytrého parkoviště	32
6.1	Protokol komunikace – OCPP	32
6.2	Centrální řídicí systém	32
6.3	Integrace s uživatelskou aplikací	33
6.4	Technické možnosti chytrého parkoviště	33
7	Mobilní aplikace pro smart parkoviště	35
7.1	Návrh funkce aplikace ve spojení s centrálním systémem a parkovištěm 35	
7.1.1	Rezervace	35
7.1.2	Zaparkování vozidla	36
7.1.3	Nastavení nabíjení	36
7.1.4	Odjezd	37
7.2	Funkční požadavky aplikace	37
7.2.1	Administrace účtu	37
7.2.2	Správa vozidel	37
7.2.3	Správa parkovacího místa	37
7.2.4	Ovládání nabíjení	38
7.2.5	Monitorování stavu baterie	38
7.2.6	Oznámení	38
7.2.7	Platby	38
7.2.8	Získávání informací o parkovišti	38
7.2.9	Navigace k parkovišti	38
7.3	Non-funkční požadavky aplikace	39
8	Vývoj mobilní aplikace	40
8.1	Použité technologie	40

8.1.1	Flutter	40
8.1.2	FlutterFlow	42
8.1.3	Firebase.....	44
8.2	Návrh obrazovek.....	46
8.3	Implementace aplikace	51
8.3.1	Vstupní obrazovka – přihlášení a registrace.....	54
8.3.2	Hlavní obrazovka	56
8.3.3	Platby	59
8.3.4	Onboarding.....	60
8.3.5	Rezervace	61
8.3.6	Přidání vozidla	63
8.3.7	Možnosti nabíjení.....	65
8.3.8	Navigace k parkovišti	67
8.3.9	Historie plateb.....	69
8.3.10	Profil	70
8.3.11	Děkovací obrazovka s instrukcemi	71
8.4	Problém získání informací.....	72
8.5	Komunikace s databází – Cloud Firestore.....	73
9	Testování aplikace	75
10	Shrnutí výsledků	77
11	Závěry a doporučení.....	78
12	Seznam použité literatury.....	80
13	Přílohy	86
13.1	Přiložený zip soubor.....	86
13.2	Oskenované zadání diplomové práce	86

Seznam obrázků

Obrázek 1 - Wallbox Volkswagen ID. Charger (zdroj: [11])	7
Obrázek 2 - Vozidlo Tesla Model 3 nabíjeno na Tesla Supercharger (zdroj: [17]).....	8
Obrázek 3 - Nio battery swapping stanice (zdroj: [19]).....	9
Obrázek 4 - Typy nabíjecích konektorů (zdroj: [22]).....	10
Obrázek 5 - Nabíjecí křivka konstantního napětí a proudu (zdroj: [25]).....	11
Obrázek 6 - Schéma cloudového řízení výkonu společnosti Unicorn pro ChargeUp (zdroj: [29])	14
Obrázek 7 - Parkovací dům v Hradci Králové (zdroj: [32])	16
Obrázek 8 - BIKETOWER v Hradci Králové (zdroj: [36]).....	19
Obrázek 9 - Vertikální parkoviště v New Yorku. Rok 1920 (zdroj: [37])	20
Obrázek 10 - Rotační vertikální parkoviště pro automobily (zdroj: [38]).....	21
Obrázek 11 - Paletové parkování v objektu domu Na Příkopě 14, Praha 1 (zdroj: [39])	22
Obrázek 12 - Pohled zespodu na roštové řešení při předání vertikálního a horizontálního roštu (zdroj: [41])	23
Obrázek 13 - Nabíjecí skupiny s frontami (zdroj: vlastní tvorba, založeno na [42])	27
Obrázek 14 - Tesla Megapack (zdroj: [46])	30
Obrázek 15 - Zjednodušený use case diagram (zdroj: vlastní zpracování)	39
Obrázek 16 - Příklad Flutter kódu (zdroj: vlastní zpracování)	42
Obrázek 17 - Vývojové prostředí FlutterFlow (zdroj: vlastní zpracování).....	43
Obrázek 18 - Příklad akcí ve FlutterFlow (zdroj: vlastní zpracování)	44
Obrázek 19 - Databáze v prostředí FlutterFlow (zdroj: vlastní zpracování)	46
Obrázek 20 - Model základních funkcí hlavní obrazovky (zdroj: vlastní zpracování)	47
Obrázek 21 - Model možností nabíjení (zdroj: vlastní zpracování)	47
Obrázek 22 - Drátěný model obrazovek a možných přechodů mezi nimi (zdroj: vlastní zpracování)	48

Obrázek 23 - Vývojový diagram průchodu aplikací od přihlášení až po hlavní obrazovku (zdroj: vlastní zpracování).....	49
Obrázek 24 - Vývojový diagram provedení platby (zdroj: vlastní zpracování)	50
Obrázek 25 - Hlavní obrazovka šablony Fleet Cars (zdroj: vlastní zpracování).....	51
Obrázek 26 - Diagram databázové struktury (zdroj: vlastní zpracování).....	53
Obrázek 27 - Obrazovky přihlášení a registrace (zdroj: vlastní zpracování)	54
Obrázek 28 - Akce pro přihlašování ve FlutterFlow (zdroj: vlastní zpracování)	55
Obrázek 29 - Hlavní obrazovka aplikace (zdroj: vlastní zpracování)	57
Obrázek 30 - Stripe klíče v nastavení FlutterFlow (zdroj: vlastní zpracování)	59
Obrázek 31 - Onboarding obrazovka (zdroj: vlastní zpracování).....	61
Obrázek 32 - Obrazovky rezervací (zdroj: vlastní zpracování)	63
Obrázek 33 - Obrazovka přidání vozidla (zdroj: vlastní zpracování)	65
Obrázek 34 – Možnost výběru limitu nabíjení (zdroj: vlastní zpracování)	66
Obrázek 35 – Možnost výběru času odjezdu (zdroj: vlastní zpracování).....	66
Obrázek 36 - Možnosti nabíjení (vlevo) a otevřená nabídka režimů (vpravo) (zdroj: vlastní zpracování)	67
Obrázek 37 - API klíče pro Google Maps (zdroj: vlastní zpracování)	68
Obrázek 38 - Mapa pro navigaci k parkovišti (zdroj: vlastní zpracování)	69
Obrázek 39 – Seznam plateb (zdroj: vlastní zpracování)	70
Obrázek 40 - Obrazovka profilu uživatele (zdroj: vlastní zpracování).....	71
Obrázek 41 - Obrazovka instrukcí k vyparkování (zdroj: vlastní zpracování)	72

1 Úvod

Elektromobilita je jedním z klíčových trendů v oblasti dopravy, který má potenciál přinést řadu výhod jak pro životní prostředí, tak pro společnost. Elektrická vozidla (EV) produkují méně emisí skleníkových plynů a lokálních znečišťujících látek než vozidla se spalovacími motory a snižují závislost na fosilních palivech. Elektrická vozidla také umožňují integraci obnovitelných zdrojů energie do elektrické sítě pomocí technologie V2G (Vehicle to Grid), která umožňuje vozidlům dodávat energii zpět do sítě v době vysoké poptávky nebo nízké nabídky z obnovitelných zdrojů.

Nicméně, elektromobilita také přináší některé problémy a výzvy, které je třeba řešit pro její úspěšné rozšíření. Jedním z hlavních problémů je nedostatek infrastruktury pro parkování a nabíjení elektrických vozidel, zejména ve velkých městech, kde je omezený prostor a vysoká hustota dopravy. Tradiční parkoviště a nabíjecí stanice často nejsou schopny uspokojit rostoucí poptávku po elektromobilitě, což vede k dlouhým frontám, nedostatku parkovacích míst a nízké úrovni nabíjení. Tyto problémy mohou odradit potenciální uživatele od přechodu na elektromobilitu a snížit její výhody pro životní prostředí a společnost.

Proto je potřeba najít nové a inovativní způsoby, jak zlepšit infrastrukturu pro parkování a nabíjení elektrických vozidel, které by umožnily efektivní využití prostoru, snížení emisí a podporu obnovitelných zdrojů energie. Jedním z takových způsobů je koncept smart vertikálního parkoviště pro elektromobilitu, který využívá automatizované systémy pro parkování a nabíjení elektrických vozidel ve vertikálních parkovacích věžích, které šetří cenné místo a chytrými systémy zmenšují náročnost na energetickou síť.

2 Cíl a metodika práce

Cílem této diplomové práce je prozkoumat a analyzovat problematiku nabíjení elektromobilů na vertikálních parkovištích. Zvláštní důraz bude kladen na inteligentní funkce těchto parkovišť, různé metody dobíjení elektromobilů a výzvy spojené se simultánním dobíjením více vozidel při plné rychlosti. Práce si klade za cíl nejen zdůraznit výhody tohoto revolučního konceptu, ale také identifikovat technické a praktické problémy, které je třeba vyřešit, aby bylo možné tuto technologii efektivně implementovat.

Smart parkoviště mohou nabízet funkce, jako je například navigace na volné parkovací místo. To u vertikálního parkoviště možné není, jelikož takové parkoviště má často pouze jeden vjezd podobný výťahu. Do tohoto vjezdu řidič zaparkuje vozidlo a parkovací věž poté vozidlo uloží na místo (například otočením celé věže). Pro integrování smart technologií byl zvolen jiný přístup. A to aplikace, přes kterou je možné toto parkoviště ovládat a případně pomoci optimalizovat využití elektrické sítě při nabíjení zaparkovaných vozidel.

V následujících kapitolách budou rozebrány klíčové aspekty smart vertikálních parkovišť, včetně technických detailů nabíjecí infrastruktury. Dále budou zkoumány různé metody nabíjení elektromobilů a technické výzvy, které vznikají při současném dobíjení více vozidel. Na základě rešerše a analýzy relevantních zdrojů budou navržena doporučení a potenciální řešení těchto výzev.

V praktické části bude na těchto základech vytvořena mobilní a webová aplikace pro tato parkoviště. Prvním krokem bude návrh požadavků pro aplikaci. Následně bude popsána funkcionality jednotlivých obrazovek a jejich význam. Výsledkem této práce bude aplikace umožňující komunikaci s centrálním systémem, ve které si uživatel bude moci rezervovat parkovací místo, mít přehled o stavu svého vozidla, ovládat možnosti nastavení a platit za použité služby. Tato aplikace bude sloužit i pro následné přistavení (vyparkování) vozidla z věže.

Při práci bude využita umělá inteligence ChatGPT 4.0¹ pro formátování textu a hledání nových poznatků a zdrojů. Nalezené poznatky a výsledky od umělé inteligence budou podrobeny kritickému zkoumání a ověření, jelikož tato technologie si často fakta vymýšlí.

¹ Dostupné z <https://chat.openai.com/> a <https://www.bing.com/>

3 Elektromobil

Jedná se o vozidlo, které je poháněno elektrickým motorem. Tento motor je nejčastěji napájen z baterie, takže nepotřebuje benzín ani naftu. Konstrukce elektromobilů je jednodušší než u klasických aut se spalovacími motory, protože motor je nahrazen jedním, nebo více elektromotory. Existují i vozidla hybridní. Ta mají elektromotor, který pomáhá běžnému spalovacímu motoru (například při rozjezdu nebo náhlém zrychlení) a tím snižuje jeho spotřebu. Mohou se nabíjet jako běžné elektromobily, nebo se jejich baterie nabíjí za jízdy. [1]

Ekologie elektromobilů je sporná a neustále debatována. Elektromobily nevydávají při svém provozu žádné emise. Tím jsou ideální do města, protože neznečišťují okolní prostředí.

3.1 Výhody elektromobilů

Kromě ekologických výhod mají elektromobily také další výhody. Hlavní výhodou jsou nižší provozní náklady. Elektřina je obecně levnější než benzín či nafta. K nižším provozním nákladům také přispívá menší počet pohyblivých částí ve vozidle. Elektromotory totiž neobsahují olej, svíčky, filtry a další části, které by bylo potřeba pravidelně servisovat a měnit. Elektromobily navíc zvyšují komfort a bezpečnost jízdy, protože mají obecně lepší akceleraci, stabilitu a ovladatelnost než vozy se spalovacími motory.

3.2 Nevýhody elektromobilů

Momentálně je největší nevýhodou elektromobilů jejich dojezd a degradace baterie. Většina elektromobilů má stále dojezd na jedno nabití okolo 300 km. [2] Novější elektromobily už zvládnou ujet i více než 600 km na nabití, ale kvůli jejich vysoké ceně však nejsou tak moc rozšířené. [3] To mnoha lidem nemusí stačit a při delším a častém cestování musí často zastavovat a delší dobu nabíjet na veřejných nabíječkách, kde stráví často desítky minut v závislosti na rychlosti nabíječky. [4]

Degradace baterie je proces, při kterém dochází ke snižování kapacity a výkonu baterie v průběhu času a používání. V průměru baterie ztratí 13.5 % své kapacity za 6.5 roku. Navíc životnost těchto baterií je pouze kolem 8 let. [5] K tomu odpůrci

elektromobilů namítají tím, že baterie obsažené v těchto vozidlech jsou velice náročné na výrobu, a i následnou likvidaci.

Tyto problémy se už firmy snaží aktivně řešit vymýšlením nových typů baterií, i efektivnější recyklací. V současné době není žádná veřejně dostupná a používaná baterie, která by měla dlouhou životnost, nabíjela se za pár minut a k tomu byla ekologická. Další velkou nevýhodou v dnešní době je jejich pořizovací cena. Ta je často vyšší než u podobného auta se spalovacím motorem.

3.3 Nabíjení elektromobilů

Elektrická vozidla se nabíjejí na parkovištích u nabíjecích stanic, nebo běžných domácích, či třífázových zásuvkách. Rychlost nabití závisí výkonu nabíječky.

Většina majitelů elektromobilů nabíjí své vozidlo v práci nebo doma. Po příjezdu do práce či domů dají své vozidlo nabíjet a po odchodu mají vozidlo nabité a mohou s plnou baterií odjet. Ne každý má ale možnost nabíjet doma nebo v práci. Například lidé, kteří bydlí v bytě nemají často vůbec žádnou možnost doma auto nabít, jelikož nemají svoji garáž a na ulici vozidlo jen tak nabíjet nelze. Musí tedy využívat nabíjecích parkovišť (často placených), kde musí své vozidlo zaparkovat a dát nabít. Jelikož nabíjení může trvat dlouhou dobu v závislosti na typu nabíječky, přiváděném proudu a také počtu ostatních právě nabíjených vozidel, mohou buď u vozidla čekat, dokud se nenabije, nebo odejít na jídlo, či na nákup. Při návratu zpět bude jejich vozidlo dobité. Existují i lidé, kteří používají tato nabíjecí parkoviště jako běžné parkoviště, kde nechávají své vozidlo odstavené a nabíjené celý den. Právě proto, že je na těchto parkovištích omezen počet nabíjecích stanic není vhodné, aby zde lidé nechávali své vozidlo delší dobu, než je nutné pro jeho nabití, jelikož zabírají místo pro ostatní. Tento problém bude více řešen v dalších kapitolách této práce.

Lidé žijící v bytech musí tedy počítat s tím, že při cestě do své destinace budou muset zastavit a delší dobu nabíjet. Mnozí však nemají tolik času a preferují vozidla se spalovacími motory, které lze dotankovat během pár minut. Právě toto omezení způsobuje menší zájem o elektromobily a omezuje technologický růst.

3.3.1 Domácí nabíjení

V domácích podmínkách lze jednoduše připojit elektromobil do běžné 230 V zásuvky se střídavým proudem. Ta poskytuje maximální nabíjecí proud 16 A, což je přibližně 3,6 kW, a jedná se tedy o nejpomalejší metodu nabíjení. Například elektromobil, který má baterii o velikosti 30 kWh by se touto rychlostí nabíjel teoreticky 8,3 hodin ($30 \text{ kWh} / 3,6 \text{ kW} = 8,333 \text{ hod}$). Velice rozšířený a oblíbený elektromobil Tesla Model 3 Long Range má aktuálně velikost baterie 82 kWh. Plně nabití z běžné domácí zásuvky by tedy trvalo skoro 25 hodin a je ještě třeba započítat ztráty. [6]

Kromě nabíjení z běžné 230 V zásuvky lze také nabíjet značně rychlejší 400 V třífázovou zásuvkou. Ta nabízí nejčastěji výkon až 22 kW (nejčastěji však 11 kW). Tento výkon nabije 82 kWh baterii v elektromobilu Tesla Model 3 Long Range teoreticky za necelé 4 hodiny, ale reálně je tento čas větší, jelikož baterie má různé rychlosti nabíjení v závislosti na jejím aktuálním nabití, teplotě, nabíjecí křivce baterie a dalších faktorech. [7]

Jak je zřejmé, domácí nabíjení trvá relativně dlouho. Je vhodné tedy vozidlo nabíjet přes noc, nebo kdykoli se vozidlo nepoužívá. Pro nabíjení doma z výše zmíněných zásuvek je navíc třeba mít speciální kabel, který slouží jako adaptér.

Alternativou pro domácí nabíjení jsou takzvané wallboxy. Výhodou wallboxů je integrace nabíjecího kabelu, možnost řízení výkonu a i cena. Nabíjení přes wallbox je také levnější než z klasické zásuvky, jelikož nabízí větší efektivitu. Jeho výkon je většinou 11 kW, 22 kW, až 40 kW. [8] Nevýhodou wallboxů je jejich pořizovací cena, která se pohybuje v řádech jednotek až vyšších desítek tisíc, podle výrobce. [9]

Například wallbox od výrobce Tesla má v této době pořizovací cenu 13 900 Kč. [10] K této ceně je však ještě nutno připočítat cenu za odborné zapojení elektrikářem.

K nabíjení elektromobilu doma lze také pořídit speciální tarif elektřiny, určený přímo k nabíjení elektromobilu.



Obrázek 1 - Wallbox Volkswagen ID. Charger (zdroj: [11])

3.3.2 Veřejné nabíječky

Veřejné nabíječky jsou k dispozici všem elektromobilům. Nachází se často v obchodních centrech na parkovištích. Lze ale najít i samostatná parkoviště, kde každé parkovací místo je osazeno vlastní nabíječkou. Poblíž je často alespoň restaurace či obchod, kde může člověk trávit čas mezitím co se mu jeho vůz nabíjí.

Nabíječky se dělí na AC (střídavý proud) a DC (stejnoseměrný proud). AC stanice jsou pomalejší, méně energeticky náročné a nabízí příkon až 43 kWh. Oproti tomu DC stanice nabízí již mnohem vyšší rychlost. Ta se může pohybovat i ve stovkách kWh. Automobilky Porsche a BMW udělali prototyp nabíječky, která nabízí příkon až 450 kWh a zvládli s ní nabít elektromobil BMW i3 z 0 % na 80 % za 15 minut. [12] Zajímavostí je, že Česku je momentálně k dispozici nejvýkonnější nabíjecí stanice s výkonem 400 kW. [13]

Přesný počet veřejných stanic nelze říct. Každým dnem se jejich počet navyšuje. I tak ale v množství nabíjecích stanic zaostáváme za svými sousedy. U nás se aktuální počet stanic k datu 11.08.2023 pohybuje okolo 1300. [14]

Velké množství nabíjecích stanic je placených. Existuje ale velké množství nabíjecích stanic, které nabízí nabíjení úplně zdarma. Například na

www.mapotic.com/nabijim-zadarmo lze zobrazit mapa, kde se takové nabíjecí stanice dají najít.

Nějací výrobci aut navíc nabízí ke svým vozidlům nabíjení zdarma na nabíječkách dané firmy. Toto nabízí například již zmíněný výrobce Tesla, který při koupi vozidla nabízí nabíjení na nabíjecích stanicích zvaných Tesla Supercharger. Jedná se o nabíječky, které umožňují velice rychlé nabíjení rychlostí až 250 kWh. Touto rychlostí lze teoreticky nabít již zmíněný elektromobil Tesla s baterií o velikosti 83 kWh za 18 minut. U elektromobilů se také objevuje výraz „počet kilometrů/mil za X minut“. V tomto případě se jedná o 120 km za 5 minut. [15]

Reálná rychlost nabíjení je často mnohem menší a závisí na aktuálních podmínkách jako například: Kolik je připojeno dalších vozidel, jak je stará baterie, jakou má baterie teplotu. Reálně se tak Tesla Model 3 nabije nikoliv za 18 minut, ale přibližně hodinu. Čím více je baterie nabitá, tím pomaleji Tesla Superchargery nabíjí. Jsou tedy schopné nabíjet rychlostí 250 kWh, ale nikoliv po celou dobu nabíjení. [16, 17]



Obrázek 2 - Vozidlo Tesla Model 3 nabíjeno na Tesla Supercharger (zdroj: [17])

3.3.3 Battery swapping

Pomalou rychlost nabíjení je možné obejít výměnou vybité baterie za nabitou. Tato technologie zkracuje nabíjení z desítek minut na pár minut. Řidič elektrického vozidla nemusí čekat desítky minut až hodin, dokud se jeho vozidlo nenabije na

požadovanou hodnotu, ale pouze se během pár minut jeho současná (vybitá) baterie vymění za plně nabitou baterii. Vybitá baterie se dá mezi tím nabít, aby byla později dostupná pro jiné vozidlo. Tento koncept se dá snadno představit jako výměna baterií v ovladači k televizi, nebo v dětské hračce.

Tento koncept již funguje v Číně, kde automatické systémy vymění baterii vozidla za nabitou během 3 minut a nabízí ji výrobce elektromobilů Nio pro svá vozidla. [18]







Obrázek 3 - Nio battery swapping stanice (zdroj: [19])

V Evropě se o tuto technologii pokouší prosadit automobilka Renault. [20]

Největším problémem, který zatím zamezuje masovému rozšíření této technologie, je nekompatibilita baterií a konstrukční zpracování elektromobilů od různých výrobců. Každý výrobce má baterii ve voze umístěnou trochu na jiném místě a nemusí být snadno přístupná pro automatickou výměnu. [21]

3.4 Konektory pro nabíjení elektromobilů

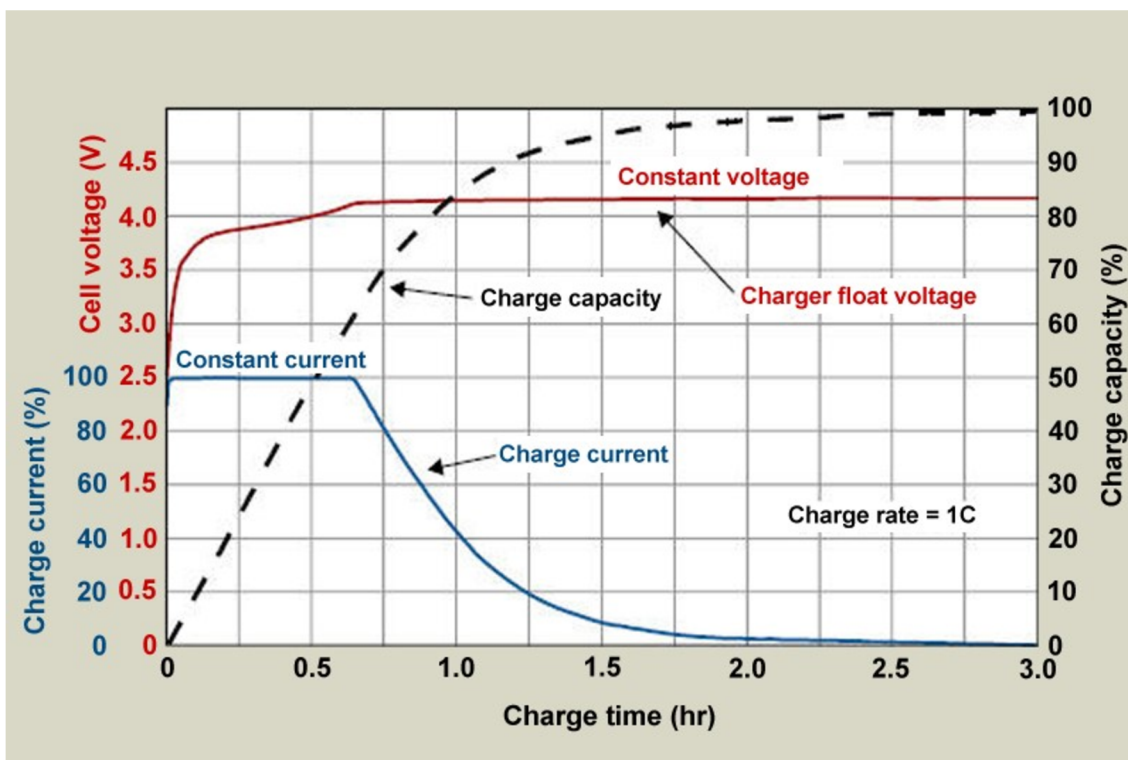
Konektory umožňují přenos elektrické energie z nabíjecí stanice do baterie elektromobilu. Rozlišuje se několik druhů nabíjecích konektorů. Ty se liší podle tvaru, velikosti, počtu pinů, typu proudu a maximálního výkonu. V Evropě je nejrozšířenější CCS2 a Mennekes. Jednotlivé rozdíly nejlépe popisuje následující obrázek.

AC nabíjení	Výkon až 43 kW	Mennekes Evropský standard pro AC nabíjení Obvykle musíte mít vlastní kabel	
DC nabíjení	Výkon (aktuálně) až 350 kW	CCS2 (Combo) Evropský standard pro DC nabíjení Nabíjecí kabel součástí stojanu	
	Výkon až 50 kW	CHAdeMO Japonský standard pro DC nabíjení, v Evropě na ústupu ve prospěch CCS2 Nabíjecí kabel součástí stojanu	
	Výkon až 250 kW	Tesla Supercharger Standard pro vozy Tesla, v Evropě na ústupu ve prospěch CCS2 Nabíjecí kabel součástí stojanu	

Obrázek 4 - Typy nabíjecích konektorů (zdroj: [22])

V případě smart parkoviště může být připojení konektoru zautomatizováno. Automatické smart nabíjecí stanice umožňují bezkontaktní dobíjení elektromobilů, kdy člověk nemusí ručně připojovat konektor. V případě chytrého parkoviště, které je dále popsáno v této práci, může ale větší množství konektorů způsobit, že se automatickému systému nepodaří nabíjecí kabel připojit. V budoucnu je možné, že bude existovat pouze jeden standardizovaný typ nabíjecího konektoru pro všechny elektromobily. To by usnadnilo rozvoj automatických nabíjecích stanic a také zvýšilo pohodlí pro řidiče elektromobilů.

Jak již bylo zmíněno, rychlost nabíjení závisí na výkonu nabíječky, teplotě, ale i na technologii a nabíjecí křivce baterie. Například čas nabíjení pro dvě vozidla se stejnou kapacitou baterie se může značně lišit, jelikož jejich baterie má jinou teplotu a nabíjecí křivku. Li-ion baterie je ideální udržovat v rozsahu 20–80% nabití. Nabíjení mimo tento rozsah může baterii více opotřebovat a zkracovat její životnost. Proto mají tyto baterie nabíjecí křivku. Obecně lze říct, že čím nabitější je baterie, tím je její nabíjení pomalejší. Poté kolem 80 % dojde k dramatickému zpomalení nabíjení. Různá vozidla a baterie nemají tuto křivku úplně stejnou. S tímto je třeba počítat při nabíjení a případné optimalizaci nabíjení více vozidel. [23, 24]



Obrázek 5 - Nabíjecí křivka konstantního napětí a proudu (zdroj: [25])

3.5 Řízení dobíjení

Řídit dobíjení je vhodné pro optimalizaci nákladů a zajištění odběrů tak, aby velký odběr nezapříčinil aktivaci jistících a ochranných prvků, což by mělo za následek úbytek výkonu a tím zpomalení rychlosti nabíjení jednotlivých vozidel.

Způsoby řízení odběru dobíjecích stanic máme několik a můžeme je rozdělit do tří úrovní:

3.5.1 Proprietární

Jako proprietární řešení může sloužit HDO spínač (pro spínání, když zrovna je k dispozici nízký/noční tarif elektřiny), časovaná zásuvka nebo třeba nastavování limitů v chytrém wallboxu.

Wallboxy mají často možnosti nastavit maximální limity odběru v ampérech [A] nebo kilowatthodinách [kWh]. Wallbox je typicky připojen pomocí ethernetového kabelu nebo WiFi. Někteří výrobci wallboxů dokonce nabízí možnost vložit do wallboxu SIM kartu. Po připojení na internet lze přes mobilní nebo webovou aplikaci

sledovat aktuální stav a nastavovat limity a jednoduchá pravidla (rychlost nabíjení, kdy začít nabíjet, kdy přestat a další). [26]

3.5.2 Lokální energetická řešení

Jako lokální energetická řešení se řadí například lokální chytrá síť, regulátory výkonu, nebo třeba lokální stacionární uložení energie.

Pro větší množství dobíjecích stanic na jednom místě a také velké požadované výkony by dimenzování přípojky na maximální možný odběr bylo velice neefektivní. Pouze 4 DC stanice s 50 kW konektory a 22 kW AC konektorem, nebo třeba 6 AC stanic (s výkonem až 43 kW) v jednom areálu vyžadují přípojku větší než 250 kW. Při překročení tohoto odběru již nestačí měření typu C, ale je potřeba měření typu A nebo B. [27] To s sebou nese další náklady a také potřebu zavést další firemní procesy.

Dnes existují systémy, které dokážou monitorovat maximální výkon nebo čtvrt hodinové odběry připojených odběrných míst. Systém se skládá z centrální jednotky, která je předřazena jističi. Navíc se skládá z kontrolérů, které sbírají data z jednotlivých odběrných míst. Kontroléry na základě pokynů centrální jednotky ovládají výkon odběrných míst. Zjednodušeně řečeno, systém sleduje, kolik elektřiny každé odběrné místo spotřebovává, a v případě potřeby může omezit jeho výkon. To pomáhá zajistit, že nedojde k přetížení elektrické sítě.

Čtvrt hodinové maximum je jedno z opatření, jak dosáhnout konstantní odběr díky zamezení odběrovým špičkám jednotlivých zákazníků, či odběrných míst. Zákazník se zaváže, že nesmí spotřebovat více energie, než je maximální sjednané množství, které odebere během patnáctiminutového intervalu. Toto sjednané množství energie se jmenuje čtvrt hodinové maximum. Při překročení tohoto limitu zákazník platí smluvní pokutu. Pro efektivní dodržení této maximální hodnoty bez překročení se využívají regulátory čtvrt hodinového maxima. To jsou zařízení, která snímají aktuální odběr a pokud začne hrozit překročení nasmlouvaného odběru, tak automaticky odpojí některé spotřebiče. [28, 29]

Výrobci však své systémy neustále zlepšují a vznikají verze, které jsou připojeny ke cloudovému prostředí v reálném čase. Umožňují pracovat nejen s odběrem, ale i

s lokální výrobou (např. solární panely). Rozšiřují také možnosti konfigurace, řízení priorit a limitů pro jednotlivá odběrná místa. [30]

Pro minimalizaci nákladů a snadné nastavení limitů pro všechny propojené stanice na jednom místě se také v dnešní době používá řešení master-slave. Jedná se o propojení více stanic k jedné, která slouží jako hlavní a ovládá všechny ostatní.

Pro více nabíječek na jednom místě se často používají stacionární uložení elektřiny. Ty většinou slouží jako velká baterie, která v době nepotřeby akumuluje energii a poskytuje ji, když je potřeba. Například když se nabíjí větší počet aut zároveň.

Lokální energetická řešení mohou většinou fungovat bez závislosti na okolním prostředí a jeho znalosti. Často bývá jediným vstupem celkový limit pro odběr. Ten může být ale proměnný v čase, a proto znalost okolí zvyšuje efektivitu a může přinést další benefity.

3.5.3 Řízení centrálním systémem

Oproti lokálním řešením umožňuje centrální systém řídit výkon pouze pomocí řídicích jednotek dobíjecích stanic.

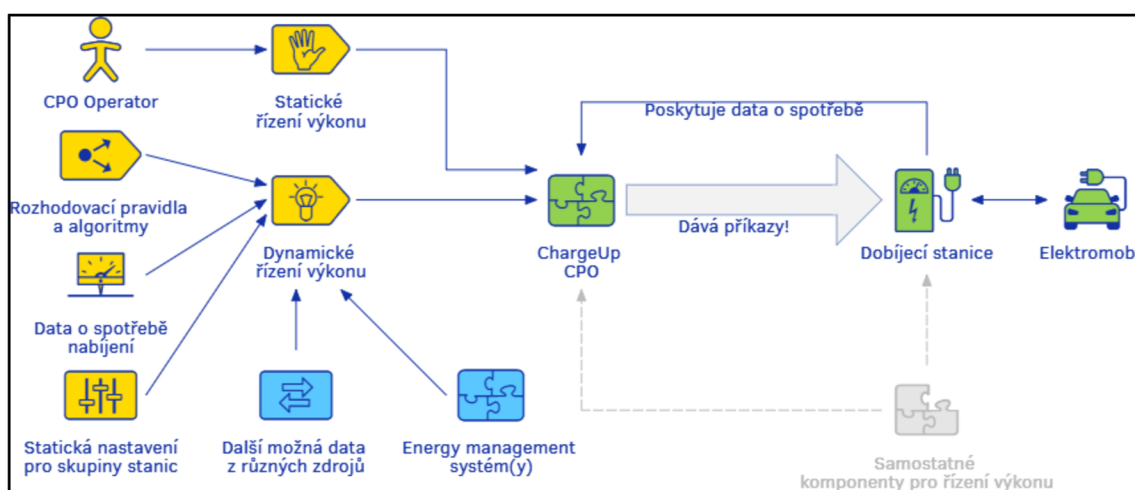
Umožňuje přímé řízení dobíjecí infrastruktury díky otevřenému standardu OCPP na základě znalosti potřeb sítě a také zákazníka. OCPP se používá pro komunikaci mezi dobíjecí stanicí a řídicím systémem. [31]

Tento protokol obsahuje služby a operace umožňující řízení výkonu stanice a také jednotlivých dobíjení z řídicího systému. Díky tomu umožňuje implementovat jednoduché i komplexní řídicí systémy.

Z pohledu složitosti můžeme rozdělit řízení na kategorie:

- **Statické řízení výkonu** – Umožňuje vzdáleně nastavit limity výkonu pro jednotlivé stanice. Oproti proprietárnímu řešení, kde lze nastavit jeden limit ve firmwaru stanice je výhodou, že operátor může sestavit profily založené proměnné v čase a přiřadit je hromadně skupinám stanic. To se dá využít pro aplikování jednoduchých pravidel ke snížení výkonu dobíjení při opakujících se výkonnostních špičkách nebo při jiných energeticky náročných procesech.

- Dynamické řízení výkonu na úrovni jednotlivých dobíjecích transakcí –**
 Statické řízení se zaměřuje převážně na stanice jako takové a je vhodné pro pravidla, která se jednou nastaví a následně občas změní. To naopak dynamické řízení umožňuje reagovat na aktuální situaci. Řídicí systém tedy může dle aktuálního odběru řídit proměnná maxima. K dělení výkonu mezi jednotlivé transakce lze využívat jednoduchá pravidla (příkladem může být poměrné dělení na základě maximálního možného výkonu jednotlivých dobíjecích stanic) nebo komplexní algoritmy pracující s různorodými daty z okolních systémů a podle nich se rozhodují.



Obrázek 6 - Schéma cloudového řízení výkonu společnosti Unicorn pro ChargeUp (zdroj: [29])

4 Parkovací domy

V dnešním rychle se rozvíjejícím automobilovém světě se stává parkování vozidel stále závažnějším problémem hlavně ve městech. Rostoucí počet vozidel a omezené množství dostupného parkovacího místa klade na města a jejich obyvatele stále větší tlak. Není tedy překvapením, že hledání volného parkovacího místa se stává často frustrujícím zážitkem pro řidiče a vede k dopravním zácpám a znečištění ovzduší. V tomto kontextu hrají parkovací domy klíčovou roli při hledání efektivního a udržitelného řešení pro parkovací potřeby moderního města.

Cílem parkovacích domů je nabídnout optimální využití dostupného prostoru a zároveň zlepšit pohodlí a bezpečnost pro řidiče. Tradiční parkovací domy představují první krok směrem k efektivnějšímu využití pozemních ploch. Tyto struktury jsou navrženy tak, aby nabídly kapacitu pro uložení velkého počtu vozidel ve více úrovních. Parkovací domy mohou podzemní i nadzemní. Tím se nejen snižuje potřeba vodorovného prostoru pro parkování, ale také se vozidla chrání před nepříznivými povětrnostními podmínkami a neoprávněným přístupem.

Nicméně s postupující technologií a potřebami měst se vyvinuly chytré parkovací systémy. Tyto systémy využívají senzory, kamery a automatizaci, aby umožnily řidičům snadněji najít volné parkovací místo a maximalizovat využití dostupných zón pro parkování. To významně přispívá k redukci zácpy, snižování emisí a zlepšení životního prostředí ve městech.

Parkovací domy nabízí většinou možnost uložit své vozidlo za poplatek. Jedná se o stavby, které jsou budovány pro stovky až tisíce parkovacích stání. Jejich výhodou je možnost kamerového hlídání a bezpečného zaparkování vozidla. Tyto domy jsou budovány v centru měst, kde je nedostatek parkovacích míst a díky využití mnoha pater nabízí skvělé možnosti, jak možnosti parkování rozšířit.



Obrázek 7 - Parkovací dům v Hradci Králové (zdroj: [32])

Velice podobné parkování je i ve větších nákupních centrech (například Westfield Chodov v Praze).

Hlavní problémy parkovacích domů:

- **Problém najít volné místo** – například o víkendech, svátcích, nebo různých akcích jsou obchody a jejich parkoviště často plné. Nedostatek míst nebo jejich špatné nacházení způsobuje zácpy na parkovištích, frustraci lidí i více uhlíkových plynů.
- **Nesprávné parkování** – Pokud auto není zaparkováno správně, může zabírat dvě parkovací místa místo jednoho. Následkem toho je menší množství míst.

Parkovací domy mohou nabízet i „smart“ funkce, po kterých je dnes stále větší poptávka. Ty dokážou pomoci například s problémem najít volné místo.

4.1 Smart parkoviště

Pojem „Smart“, neboli „chytrý“ si lze vyložit mnoha způsoby. Existují pojmy „smart cities“, „smart grids“, „smart home“, či třeba „smart osvětlení“. Pojem smart u každých znamená něco trochu jiného, ale jedno mají společné. A tím je to, že nové technologie u všech dokážou spoustu věcí zlepšit, zautomatizovat, nebo ulehčit.

Smart technologie znamená využití moderních technologií, které nějakým způsobem pomáhají člověku. V případě parkovacích domů se může jednat o zjištění volných parkovacích míst, možnost rezervace volného místa dopředu přes aplikaci (nebo webový prohlížeč), navigace na volné místo i uvnitř parkovacího domu, rozpoznávání vozidel, ochrana proti krádeži, placení bezhotovostně či dokonce využití umělé inteligence pro optimalizaci parkovacích míst. Smart parkovací domy využívají senzory, kamery, GPS a internet věcí (IoT) pro sledování aktuálního stavu parkoviště a komunikaci mezi sebou.

Ve smart parkování je pozornost zaměřena na parkovací místa, která o sobě vědí. Právě díky tomu je možné zjistit, zda na daném místě je nějaké auto zaparkováno či nikoliv.

S rostoucím počtem elektromobilů na silnicích se stávají parkovací místa s nabíječkami pro elektromobily stále žádanější. Nový stavební zákon č. 283/2021 Sb., který vstoupí v platnost od 1. ledna 2024, stanovuje, že nové i rekonstruované stavby, kromě těch určených k bydlení, musí být vybaveny alespoň jednou dobíjecí stanicí a kabelovody pro pozdější instalaci dobíjecí stanice pro elektrická vozidla pro každé páté parkovací stání, pokud má stavba více než 10 parkovacích stání. Pokud je stavba určena pro bydlení a má více než 10 parkovacích stání, musí mít instalaci kabelovodů pro každé parkovací místo pro pozdější instalaci dobíjecí stanice. [33]

Nicméně přidání nabíječky na parkovací místo z něho ihned neudělá chytré parkovací místo. Problém, který se začíná objevovat u podobných parkovišť vybavených nabíječkou je velká náročnost na energetickou síť. [34] Zde by smart technologie mohly individuálně a lépe řídit nabíjecí výkon každé nabíječky. Příkladem tohoto může být nabíjení 2 stejných elektromobilů zároveň. V tomto případě, pokud by teoreticky každý zvládl nabíjet takovou rychlostí, jakou mu nabízí

sít', tak každému nakonec připadne jen 50 % výkonu. Při čtyřech elektromobilech je to 25 % pro každého. Při zavedení smart technologie, která by se starala o prioritizaci určitých vozidel (parkovacích míst), tak by mohl mít více procent ten automobil, který to potřebuje více – neboli potřebuje dříve odjet.

Přidání nabíječky na parkovací místo nemusí být vždy jednoduché kvůli tomu, se ke každé nabíječce musí přivést napájení. Na vybudování nových parkovišť, která už budou s nabíječkou počítat už v dnešní době ve městech nezbyvá moc místa. Stejně jako u budov, když už není prostor stavět do šířky, je potřeba stavět do výšky. Řešením místa může být vertikální parkování.

S přirůstajícím počtem hybridních a čistě elektrických vozidel ale roste i poptávka po nabíjecích stanicích na parkovištích. Proto je důležité, aby nově vzniklý parkovací dům nabízel nabíjecí stanice.

Nevýhodou parkovacího domu je jeho velikost. Nelze postavit úplně kdekoliv, a tak již hustě zastavěná města nemají tolik prostoru na vybudování nových parkovacích domů. To se města snaží řešit budováním parkovišť pod zemí. Je třeba ale brát v potaz zavedenou infrastrukturu jako elektrické a síťové kabely, plynovody, vodovody a kanalizace. Což je komplikované a není vždy možné.

Mezi nejnovější a nejzajímavější inovace v oblasti parkovacích systémů patří automatické vertikální parkovací věže. Tyto konstrukce redefinují způsob, jakým vozidla jsou uložena a přístupná. Tyto věže jsou velice prostorově úsporné. Mohou zabírat pouze 30 m² neboli 2 standardní parkovací místa. [35]

4.2 Automatické vertikální parkovací systémy

Parkovací věže již většina lidí zná. Převážně v podobě parkovacích věží pro bicykly, které se vyskytují v centrech měst a u nádraží. Jedná se o věž s výtahem, do kterého člověk přiveze kolo a parkovací věž si kolo sama vyveze a uloží na volné místo. Vlastník kola dostane při zaparkování pin či stvrzenku s čárovým kódem, díky kterému si lze následně kolo z parkoviště vyzvednout. Po zadání pinu automatický výtah vyzvedne dané kolo a přiveze ho vlastníkovi.

Příkladem takové věže je BIKETOWER, který lze naléznout i v Hradci Králové.



Obrázek 8 - BIKETOWER v Hradci Králové (zdroj: [36])

Parkovací věže však nemusí sloužit pouze pro kola. Existují i automatizované věže pro vozidla. Právě takové věže se začínají pomalu objevovat ve světě, jelikož v centru měst je veliký problém s parkováním a není prostor na vybudování běžných parkovišť. Momentálně již funkční řešení vertikálních parkovacích věží se nachází většinou v Číně a jiných asijských zemích, kde je parkovacích nedostatek kvůli velké populaci. Tyto parkovací věže mohou být řešením, protože na prostor zabírající pár parkovacích míst se vejde mnohem více vozidel v závislosti na výšce věže.

Výhodou takových věží je především úspora místa a také menší znečištění ovzduší, jelikož auta nemusí několikrát objíždět (klasické) parkoviště a hledat místo. To je obzvláště užitečné v hustě obydlených oblastech.

Bezesporu další výhodou je také snadné „nalezení“ svého vozidla. Člověk nemusí své vozidlo hledat po rozsáhlých parkovištích a pamatovat si v jaké sekci, řadě, nebo patře zaparkoval své vozidlo. Automatická věž také eliminuje poničení vozidla cizí osobou. Ať už jde o odření při otevírání dveří, nebo odřený nárazník při parkování.

Za další výhodu se dá považovat i větší ochrana před zloději, protože věž není pro nikoho přístupná, takže mezi zaparkovanými auty se procházet nedá.

Princip je jednoduchý. Řidič zaparkuje své vozidlo ve spodní části věže a hydraulické nebo mechanické zvedáky zvednou vozidlo a uloží ho na volné místo, nebo se věž otočí. Existují totiž různé typy vertikálních parkovišť.

Vertikální parkoviště nejsou úplně nová věc. Tato technologie existovala i ve 20. letech minulého století. Jejich popularita začala bohužel upadat kvůli poruchovosti a dlouhé době čekání, než se auto dopraví k majiteli. To je hlavní nevýhoda vertikálních garáží. Stejně tak pokud přijede větší počet vozidel naráz k jednomu vjezdu, musí stát v koloně, než se všechny předešlá auta zaparkují. [37]



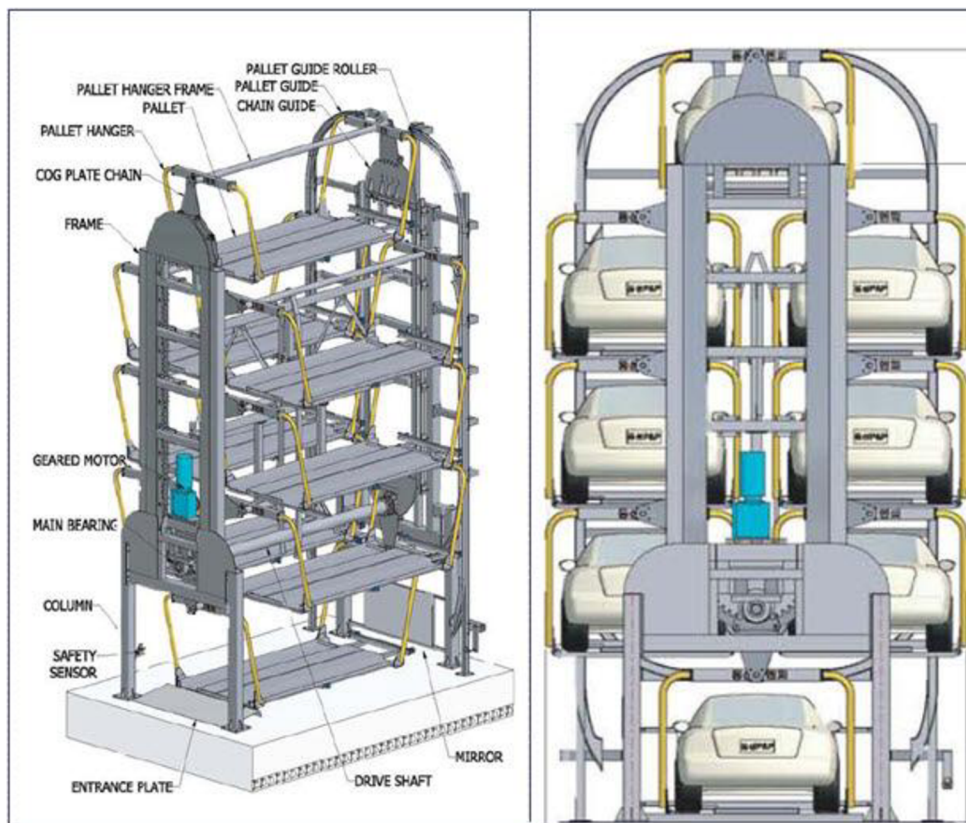
Obrázek 9 - Vertikální parkoviště v New Yorku. Rok 1920 (zdroj: [37])

V dnešní době máme již lepší technologie, a tak může být výtah rychlejší a čekání menší.

4.2.1 Rotační systém

Věže s rotačním systémem fungují na principu páternoster výtahů. V ideálním případě zabírají prostor o něco málo větší než 2 parkovací místa. Po vjezdu do

tohoto parkoviště řidič odbaví své vozidlo a celý systém se následně otočí na další volnou pozici. Systém může být krytý i otevřený. Velkou výhodou je jednoduchost systému.



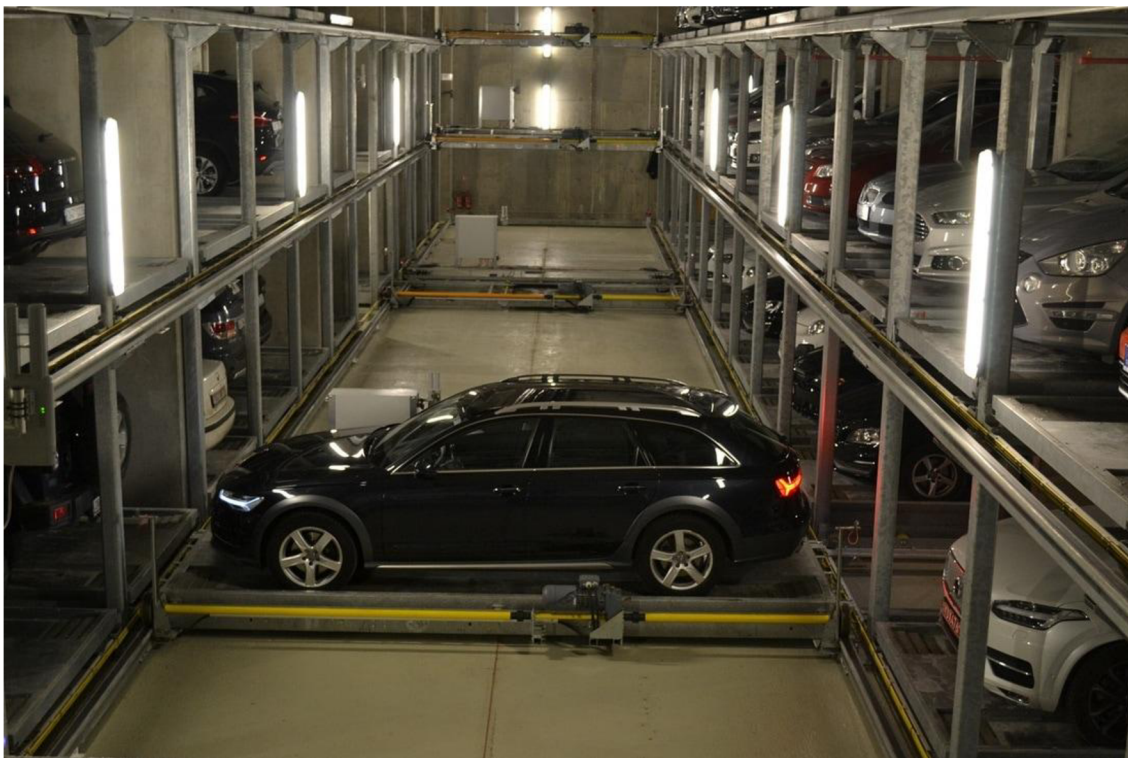
Obrázek 10 - Rotační vertikální parkoviště pro automobily (zdroj: [38])

4.2.2 Paletový systém

Tento systém je navržen tak, že vozidlo zaparkované na paletu je i s paletou odvezeno a uloženo na volné místo. Mezitím je do prostoru vjezdu přivezena paleta prázdná. Paleta je schopna se hýbat ve vertikálním i horizontálním směru.

Tento systém je vhodný pro kruhové i obdélníkové stavby, věžové i rozlehlé, podzemní či nadzemní.

Oproti bezpaletovému systému je výhoda v tom, že paleta může sloužit i pro uložení jiných předmětů, než jsou vozidla, v případě nezaplnění garáže. Také voda či jiné možné tekutiny z aut stéká na paletu, a ne na auto zaparkované pod paletou. Paleta má v sobě žlábký, které tekutinu odvedou.



**Obrázek 11 - Paletové parkování v objektu domu Na Příkopě 14, Praha 1
(zdroj: [39])**

4.2.3 Bezpaletový systém

V tomto případě systém nehýbe plošinou, na které je vozidlo, ale manipuluje přímo s vozidlem. Je několik možností, jako jsou například pásové dopravníky, rošty, či válcové přepravníky. Například rošty fungují tak, že je hlavní vertikální rošt, který vozí auto nahoru a dolů a v požadovaném patře vyjede rošt, který jezdí vertikálně a vozidlo si převezme – vertikální a horizontální rošty jsou posunuté a mohou sebou „projet“. Tyto systémy však nechraní sebe ani ostatní vozidla před nečistotou, jak je například vidět na obrázku č. 12.

Hlavními nevýhodami automatických všech parkovacích systémů je náročnost na elektrickou energii a také nutnost obeznámení uživatelů s pravidly.

„Před vjezdem do automatických parkovacích garáží je nutné prostudovat a obeznámit se s pravidly a postupy při parkování. Většina z několika málo nehod, ke kterým došlo v poslední době v těchto garážích, byla zaviněna nedodržením předepsaných bezpečnostních postupů. Důsledkem nedodržení předepsaných postupů při parkování a neukázněností řidičů je např. vytrhnutí antény, zablokování

systemu apod. V některých systémech má být anténa před opuštěním vozidla v terminálu předepsaným způsobem upravena (pokud zasahuje do manipulačního prostoru, hrozí její poškození).“ (Ing. Bohumír Číhal) [40]



Obrázek 12 - Pohled zespodu na roštové řešení při předání vertikálního a horizontálního roštu (zdroj: [41])

Při modernizaci majitelé věží vyvíjejí snahu zapůsobit na co největší množství řidičů. Dobrý nápad je rozšířit parkovací místo o možnost nabíjení elektrických vozidel. Tím by se zvedla popularita těchto věží převážně u majitelů těchto vozidel, jelikož by mohli parkovat kdekoli v centru města na libovolně dlouhou dobu a rovnou by se jejich vůz nabíjel. Řidič vozidla by při zaparkování akorát připojil auto k nabíječce, která by byla upevněna k plošině, na které auto stojí. Při zaparkování vozidla by vozidlo zůstalo připojené a mohlo by se nabíjet.

Problémem ale je, že při nabíjení většího množství vozidel naráz se kapacita sítě naplní a síť není schopna dodávat plnou nabíjecí rychlost do všech vozidel současně.

5 Optimalizace nabíjení více připojených vozidel

Před optimalizací je nutné porozumět tomu, proč je důležitá. Optimalizaci je nutné použít, aby se urychlil čas nabíjení a zmenšila se zátěž na elektrickou síť. Právě zkrácením nabíjecího času se zajistí spokojenost uživatelů (řidičů elektromobilů). Bez optimalizace je všechna energie rozdělena rovnoměrně. Pokud tedy řidič přijíždí na parkoviště s tím, že potřebuje co nejrychleji nabít své vozidlo a pokračovat v cestě, tak nelze přesně dopředu určit kolik mu nabíjení zabere času, jelikož rychlost nabíjení závisí na mnoha faktorech. Jedním z nich je kolik vozidel se bude současně s ním nabíjet. Více připojených vozidel zvyšuje nároky na energetickou síť a při velkém vytížení bude rychlost menší.

Optimalizace nabíjení více připojených elektromobilů vyžaduje kombinaci hardwarových a softwarových řešení. Zde je několik strategií, které mohou být použity k optimalizaci procesu nabíjení.

5.1 Rovnoměrné nabíjení

V nijak neoptimalizovaném prostředí je nabíjení více připojených vozidel prováděno rovnoměrným dělením maximálního nabíjecího výkonu. Když už parkoviště dosáhne svého maxima, které může poskytnout, začnou se všechny nabíječky zpomalovat. Maximální výkon je tedy závislý na počtu připojených zařízení (nabíječky a k nim připojené elektromobily). Zjednodušeně v našem případě (neberou se v potaz ztráty, různé typy vozidel, které nemusí podporovat takovou rychlost nabíjení, ...) lze říct, že platí vzorec:

$$\text{dostupný výkon pro nabíječku} = \frac{\text{maximální možný výkon sítě}}{\text{počet aktivních nabíječek}}$$

Příklad: Parkoviště má 20 nabíječek. Každá z nich má maximální výkon 100 kW. To dohromady dává potřebný výkon 2000 kW. Nicméně toto celé toto parkoviště může poskytnout maximálně výkon 1000 kW (nemá dobré pokrytí elektrické sítě, má staré vedení, nebo jakékoliv jiné důvody). To znamená že není schopna poskytovat maximální nabíjecí výkon pro všech 20 nabíječek. Pokud by se stalo, že by všech 20 nabíječek bylo obsazených a každý elektromobil k nim připojený by byl teoreticky schopen nabíjet maximální rychlostí, kterou tyto nabíječky nabízí (100 kW), tak při dosažení do výše zmíněného vzorce:

$$\text{dostupný výkon pro nabíječku} = \frac{1000}{20}$$

vychází dostupný výkon pro jednu nabíječku 50 kW. Elektromobil by byl tedy nabíjen pouze poloviční rychlostí, jelikož do nabíječky se dostává výkon 50 kW místo maximálních 100 kW.

Možností optimalizace je několik.

5.2 Optimalizované nabíjení v závislosti na čase příjezdu

První možností optimalizace může být že auta, která přijela dříve budou mít větší rychlost nabíjení než ta, která přijela později. Cílem by bylo nabít vozidlo co nejrychleji a po jeho nabití si nejvyšší výkon převezme vozidlo, které přijelo jako další v pořadí. Při výše zmíněném příkladu by se maximální možnou rychlostí zvládlo nabíjet 10 vozidel. Pokud by přijelo jedenácté vozidlo, bylo by nutné nějakému již nabíjenému vozidlu zmenšit jeho nabíjecí výkon. Vozidlo, které přijelo jako první by podle tohoto schématu stále mohlo nabíjet plnou rychlostí. U této metody je ale potřeba správně určit poměr nabíjecích rychlostí, což může být problém.

5.3 Optimalizované nabíjení v závislosti na čase odjezdu

Druhá možnost je založena na zadaném času odjezdu. Řidič vozidla po zaparkování zadá v aplikaci plánovaný čas odjezdu. Čím dříve bude chtít odjet, tím více energie bude autu dodáno, aby vozidlo mohlo odjet s dostatečným či požadovaným (zadaným od řidiče) nabitím. Vše je založené na odhadu, jak dlouho se při současném vytížení sítě budou zaparkovaná vozidla dobíjet. Pokud přijede řidič s

vozidlem, se kterým plánuje odjet až druhý den, je zbytečné, aby se vozidlo nabíjelo maximálním možným výkonem. Pokud plánuje odjet až za týden, vozidlo by se mohlo začít nabíjet až 24 hodin před odjezdem stálou rychlostí, která by byla vypočítána podle kapacity baterie vozidla. Do té doby by mohl elektromobil poskytovat svoji přebytečnou energii do sítě pomocí V2G a zvýšit tím kapacitu sítě a pomáhat vyrovnávat výkyvy výroby a spotřeby elektřiny (tzv. Load balancing). Pokud by ale přijel řidič, který potřebuje nabít své vozidlo rychle, aby mohl co nejdříve odjet, bude jeho vozidlo preferováno a nabíjecí síť mu nabídne co nejvyšší výkon.

5.4 Nabíjení založené na čase

Nastavení nabíjení vozidel pouze v určitý čas. Plánování nabíjení mimo špičkové hodiny, kdy je poptávka po elektrické energii nižší, může být efektivnější a někdy i ekonomicky výhodnější. Uživatelé mohou nastavit, aby se jejich vozidlo nabíjelo v noci nebo v jiných nešpičkových hodinách.

5.5 Load balancing (rozložení zátěže)

Jedná se o proces rozdělení elektrického proudu mezi více nabíječek tak, aby byla zachována stabilita elektrické sítě. Nabíjecí stanice nebo systém inteligentně rozděluje dostupný výkon mezi jednotlivá vozidla podle jejich potřeb a stavu baterie. Tím se zabrání přetížení sítě, zvýšení nákladů na elektřinu nebo snížení rychlosti nabíjení. To znamená, že když je celková poptávka po energii vysoká, systém může snížit rychlost nabíjení, a naopak.

Load balancing může být realizován na různých úrovních:

- Na úrovni jedné nabíjecí stanice, která má více nabíjecích přípojek.
- Na úrovni skupiny nabíjecích stanic (např. na parkovišti nebo v garáži)
- Na úrovni lokality (např. v rámci města nebo regionu)
- Na úrovni celé elektrické sítě

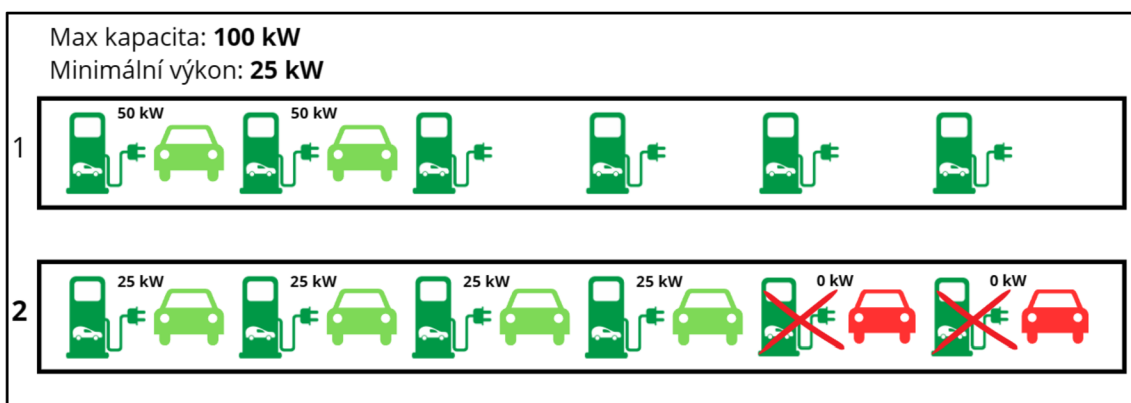
Rozložení může být jednosměrné nebo obousměrné. Jednosměrné (unidirekční) load balancing znamená, že nabíjecí stanice nebo systém pouze upravují rychlost

nebo čas nabíjení podle signálů z elektrické sítě. Obousměrné (bidirekční) umožňuje elektromobilům dodávat energii do sítě, pokud je potřeba, a tím pomáhat vyrovnávat výkyvy výroby a spotřeby elektřiny.

5.6 Nabíjecí skupiny s frontami

Podle určených pravidel založených na výše zmíněných optimalizacích je možné vytvořit nabíjecí skupiny a tím určit horní limit nabíjených vozidel. Nabíjecí skupiny (v případě menšího parkoviště je pravděpodobné že bude pouze jedna skupina) budou mít definovanou maximální kapacitu a minimální nabíjecí výkon, pod který by se síť neměla dostat. Vozidla ve skupině budou nabíjena rovnoměrně, ale skupina je omezena pouze na počet vozidel, při kterém síť neklesne pod minimální hranici. Všechna další vozidla budou zařazena do fronty, dokud nějaké vozidlo ve skupině nebude nabito.

Pro jednoduchost je lepší toto demonstrovat na příkladu: Definuje se skupina s maximální kapacitou 100 kW a minimální nabíjecí výkon 25 kW. Pokud přijede pouze jedno vozidlo, bude se nabíjet výkonem 100 kW (pokud jsou toho nabíječka i vozidlo schopné). Při dvou vozidlech bude nabíjecí výkon 50 kW, při třech vozidlech 33.33 kW, při čtyřech vozidlech 25 kW. To je minimální hranice a tedy jakékoliv další připojené vozidlo se nezačne nabíjet, dokud je skupina plná. Poté co se nějaké vozidlo nabilo, je ze skupiny uvolněno a na jeho místo je do skupiny přidáno vozidlo, které přijelo dříve.



Obrázek 13 - Nabíjecí skupiny s frontami (zdroj: vlastní tvorba, založeno na [42])

5.7 Inteligentní systém řízení nabíjení

V1G, V2G a V2B/V2H/V2X jsou všechno prvky chytrého nabíjení (smart charging). Chytré nabíjení je to proto, že umožňuje sledování a optimalizaci procesu nabíjení prostřednictvím cloudových technologií. Datové připojení je nezbytné k inteligentnímu přizpůsobení množství energie spotřebované vozidlem na základě stavu sítě při nabíjení. Zatímco V1G je spíše inteligentní nabíjení v jednom směru, V2X (vehicle-to-everything, neboli vozidlo do čehokoliv) vyžaduje několik věcí ke svému fungování. Patří mezi ně obousměrná nabíječka, komunikační protokol pro interakci mezi nabíječkou a vozidlem, vozidlo se všemi možnostmi pro V2G a dobrý řídicí systém. Tyto požadavky jsou způsobeny tím, že zahrnuje využití energie v elektrických vozidlech pro různé použití. Vzhledem k tomu, že se předpokládá nárůst elektrických vozidel, je důležité abychom je, a jejich energii, využívali efektivně, zejména v zájmu rozvodných sítí. Chytré nabíjení a jeho prvky, jako je V1G, zapadají do řešení, které je potřebné pro efektivní využívání elektrických vozidel, jak je vidět v následujících bodech.

5.7.1 V1G (jednosměrné chytré nabíjení)

Jedná se o nejjednodušší chytré nabíjení, tedy v jednom směru. „Chytré“ znamená, že umožňuje elektrickému vozidlu dynamicky upravovat sazby a čas nabíjení, protože spojuje vozidlo se stanicí pomocí datového připojení. Výhodou tohoto řešení mohou být menší náklady na nabíjení vozidla, pokud je cena elektřiny závislá na době použití (například v noci nebo ve špičce). V1G také umožňuje vozidlu komunikovat s nabíjecí stanicí a sdělit jí co potřebuje, pomocí metody strojového učení (**machine learning**).[43] Mezi další výhody patří bezpečnější nabíjení, monitorování spotřeby elektrické energie a optimalizovaná doba nabíjení. Kromě toho může také rozhodovat o nabíjení, když je energie zrovna levnější a čistší, s ohledem na dostupné informace z trhu a systému s elektřinou. Díky V1G má řidič elektrického vozidla snadný přístup k podrobnostem o spotřebě elektřiny, protože senzory ji měří a poskytují aktuální hodnoty, které mu mohou pomoci se lépe rozhodnout.

5.7.2 V2G (obousměrné chytré nabíjení – vehicle-to-grid)

V2G označuje systém, kdy jsou elektromobily schopný nejen přijímat elektrickou energii ze sítě (pro nabíjení), ale také ji vracet zpět do sítě ze své baterie v době potřeby. To může být užitečné pro vyrovnávání zátěže v elektrické síti, zejména v době špičkové poptávky, a může také pomoci v integraci obnovitelných zdrojů energie. V2G totiž umožňuje ukládat přebytečnou energii i z obnovitelných zdrojů v bateriích elektromobilů. Ta může být později uvolněna zpět do sítě, když je produkce nízká. Tato technologie má i finanční výhody. Uživatelé mohou prodávat (poskytovat) přebytečnou energii ze svých aut zpět do sítě za vyšší ceny během špičkových hodin. [44]

Nevýhodou neustálého nabíjení a vybíjení může být degradace baterie. Baterie totiž mají omezený počet cyklů a postupně se snižuje jejich kapacita. Nicméně bylo publikováno, že inteligentní řízení, optimalizace cyklů a správná termoregulace baterií mohou mít potencionálně pozitivní vliv na baterie a tím tuto nevýhodu vyřešit. Autor ale uvedl, že je stále třeba další výzkum v této oblasti, aby se toto tvrzení dalo považovat za potvrzené.[45]

5.7.3 V2B/V2H (vehicle-to-building, vehicle-to-home)

Komunikace mezi vozidlem a budovou, případně domem. energii uloženou v elektromobilech lze také využít pro napájení budov. Mimo dodávání energie zpět do sítě lze také využít energii v bateriích pro napájení domácích spotřebičů či jiných elektrických zařízení. To může dočasně pokrýt výpadky proudu a sloužit jako náhradní zdroj energie. Je také možné převést energii uloženou v baterii vozidla do baterie budovy (pokud je jí budova vybavena – např. v případě že má solární panely, ke kterým se často baterie pořizuje).

5.8 Využití obnovitelných zdrojů a externích úložišť

Pokud je možné, integrace obnovitelných zdrojů energie, jako je solární nebo větrná energie, do nabíjecí infrastruktury může přispět k udržitelnějšímu a efektivnějšímu řešení nabíjení více vozidel zároveň a také ke snížení ceny za nabíjení. Pro parkovací dům, který plánuje mít možnost nabíjení pro elektromobily a chytré prvky je ideální mít na střeše solární panely, pokud je to možné. Energie z nich zvýší efektivitu

parkovacího domu. Při spojení obnovitelných zdrojů s velkou stacionární baterií lze uchovávat přebytečnou energii, dokud nebude potřeba.

Podobný koncept již nabízí mnoho výrobců. Zde je zobrazen Megapack od firmy Tesla.



Obrázek 14 - Tesla Megapack (zdroj: [46])

Tyto velkokapacitní úložiště umožňují dodávat energii k vyrovnávání zátěže, nebo i při kompletním výpadku sítě. Jsou škálovatelné. Lze tedy připojit zvyšovat kapacitu donekonečna. Omezením je místo a samozřejmě peníze. Každý Megapack umí uchovat přes 3 MWh energie a stojí v řádech desítek milionů korun. [47]

Další alternativou těchto baterií je využívat baterie zaparkovaných elektromobilů.

5.9 Uživatelské rozhraní a aplikace

Poskytnutí uživatelského rozhraní nebo aplikace, která umožňuje uživatelům snadno nastavit preference pro nabíjení, sledovat stav nabíjení a přijímat oznámení o dokončení nabíjení. Většinu těchto funkcí již ale mají uživatelé od výrobců svých elektromobilů. V aplikaci lze vidět jakou rychlostí se auto aktuálně nabíjí, předpokládaný čas dobití, aktuální stav baterie a další funkce.

Aplikaci ale může mít i přímo chytré parkoviště, případně nabíječka. V této aplikaci lze nastavit rychlost, kterou se auto bude nabíjet, časovač, kdy se má auto začít nebo přestat nabíjet, omezení proudu a podobně.

5.10 Prediktivní analýza a umělá inteligence

Využití dat a algoritmů k předpovědi chování spotřeby energie a požadavků na nabíjení, což umožňuje se aktivně přizpůsobit a optimalizovat proces nabíjení.

Implementací těchto strategií a technologií je možné vytvořit inteligentní nabíjecí infrastrukturu, která optimalizuje proces nabíjení více připojených elektromobilů, zvyšuje efektivitu, snižuje náklady a přispívá k udržitelnosti.

Umělá inteligence by mohla časem vyhodnocovat a zlepšovat algoritmus nabíjení. Měla by data, kdy auta nejčastěji přijíždí, odjíždí, nebo třeba kdy je k dispozici levný tarif a obnovitelné zdroje.

5.11 Ekonomický faktor

Optimalizovat nabíjení lze i pomocí účtování poplatků za určité možnosti nabíjení. Například je možné zavést poplatky za větší rychlost či poplatek za nabíjení přes určitou hranici procenta baterie. Obecně platí doporučení, že aby byla zajištěna co nejdelší životnost baterie, je ideální, aby se nabití baterie pohybovalo v rozsahu 20-80 %. Je tedy možné zavést větší poplatky, pokud by řidič vozidla chtěl nabít více energie, než je tento rozsah, nebo větší rychlost. Právě poplatky by mohly řidiče odradit od využívání těchto možností, což by zmenšilo vytížení sítě a umožnilo poskytnout více energie tam, kde je to potřeba.

6 Návrh chytrého parkoviště

Pro maximalizaci potenciálu vertikálního parkoviště pro elektromobilitu je klíčové, aby každé parkovací místo bylo vybaveno nabíjecím konektorem. Při zaparkování by uživatel zapojil konektor do svého elektromobilu, což by umožnilo přenos energie a komunikaci mezi vozidlem a systémem parkoviště. Po úspěšném spojení se vozidlo uloží do věže.

6.1 Protokol komunikace – OCPP

Po připojení dobíjecího konektoru by chytré parkoviště mělo získávat informace o baterii vozidla, aktuálním nabití a další relevantní údaje pomocí univerzálního komunikačního protokolu, jako je OCPP (Open Charge Point Protocol). Tento protokol by umožnil systému vzájemně poskytovat služby a efektivně spolupracovat různým typům elektromobilů, dobíjecích stanic a centrálních systémů (interoperabilita). Protokol je založen na architektuře klient-server. Nabíjecí stanice je server a všechny ostatní systémy jsou její klienti. Umožňuje oboustrannou komunikaci nabíjecí stanice a ostatních systémů. Stanice tedy může přijímat i odesílat informace. Mezi tyto informace patří data nabíjecích relací, chybové zprávy a informace o tarifech. Součástí protokolu je také sada standardizovaných zpráv umožňující komunikaci nabíjecí stanice s ostatními systémy standardizovaným způsobem. [48]

Přes OCPP protokol stanice poskytuje informace centrálnímu systému.

6.2 Centrální řídicí systém

O veškeré výpočty, optimalizace a pravidla se stará centrální řídicí systém. Ten dostává přes OCPP protokol informace z dobíjecích stanic o připojených vozidlech a může s nimi přes tento protokol komunikovat a posílat příkazy jako jsou zapnutí/vypnutí nabíjení, nastavení nabíjecího profilu či jiné konfigurace nabíjecích stanic. Dále tento systém umožňuje spravovat transakce a fakturaci za služby nabíjení, autorizovat platby a sbírat a analyzovat data o využití energie z nabíjení elektromobilů. Tato data lze využít pro optimalizaci provozu a plánování budoucího nabíjení.

Optimalizace nabíjení může systém dosáhnout pomocí bodů zmíněných v předešlé kapitole. Právě v aplikaci by uživatel mohl zvolit, kterou možnost zrovna potřebuje a na tom bude záviset rychlost nabíjení a výsledná cena.

6.3 Integrace s uživatelskou aplikací

Uživatel by měl mít možnost naskenování QR kódu umístěného na obrazovce parkoviště nebo načtení NFC tagu. Po naskenování by se v uživatelské aplikaci spojilo zaparkované vozidlo s jeho uživatelským účtem a zobrazily se informace o zaparkovaném vozidle, jako je aktuální stav nabití, odhadovaný čas do plného nabití, možnost nastavení preferencí nabití, platby a přivolání vozidla zpět (vyparkování).

Právě tyto informace budou předány nabíjecí stanicí centrálnímu řídicímu systému pomocí OCPP protokolu. Řídicí systém poté pomocí API nebo real time databáze zobrazí informace v aplikaci odkud je bude moci uživatel kontrolovat a měnit preference nabíjení.

6.4 Technické možnosti chytrého parkoviště

Aby parkoviště bylo chytré a užitečné pro uživatele, může mít následující možnosti:

- **Rezervace míst** – Aplikace by mohla nabídnout možnost rezervace parkovacího místa, což by usnadnilo plánování a zefektivnilo využití parkoviště. Rezervace by zamezila případům, kdy řidič dojedle k parkovišti a zjistí že již nejsou žádná volná místa.
- **Stanovení odjezdu** – Aby centrální řídicí systém mohl správně optimalizovat nabíjení, měl by znát datum a čas plánovaného odjezdu. Na jeho základě podle historických dat nebo předurčených pravidel vypočítá kdy je ideální začít nabíjet vozidlo, aby bylo v požadovaný čas nabitě na požadovanou hodnotu.
- **Dynamické cenové modely** – Zavedení dynamického cenového modelování by mohlo umožnit variabilní cenovou strukturu v závislosti na poptávce, čase dne, a dalších faktorech, jako jsou například dostupné obnovitelné zdroje, požadované rychlosti, přebytek energie, ochota umožnit uložení přebytečné energie v baterii vozidla a dalších.

- **Energetický management** – V rámci udržitelnosti by systém mohl integrovat energetický management, který by optimalizoval spotřebu energie a využíval obnovitelné zdroje energie, např. solární panely či větrné turbíny.
- **Bezkontaktní platby** – Pro zjednodušení procesu by bylo vhodné implementovat možnost bezkontaktních plateb přímo přes uživatelskou aplikaci.
- **Notifikace a upozornění** – Uživatelé by měli mít možnost dostávat notifikace o stavu nabití, dostupnosti parkovacích míst, a případných změnách v rezervacích či cenách.

Další možností, co by chytré parkoviště mohlo umožňovat při používání roštového systému je možnost battery swappingu. Na roštových platformách jsou vozidla přístupná zespodu, jelikož rošt je pouze pod koly vozidla. Během parkování by byla ve vozidle vyměněna baterie za nabitou a stará (vybitá) by se mohla nabít kdykoliv je volná kapacita sítě. Toto řešení je však v dnešní době nestandardizovaných vozidel a baterií nereálné a je užitečné spíše na rychlé nabití a pokračování v jízdě než pro parkovací věž, kde vozidlo bude stát delší dobu, za kterou by se normálně stihlo nabít.

7 Mobilní aplikace pro smart parkoviště

Fungování systému centrálního řídicího systému není postaveno jen na jeho serverové části (backend). Snadné využívání tohoto systému spočívá v intuitivním uživatelském rozhraní (frontend).

Tato kapitola obsahuje praktickou část práce, která demonstruje použití prezentovaných poznatků z předchozích kapitol a zabývá se návrhem a vývojem frontendové části mobilní aplikace pro smart vertikální parkoviště.

7.1 Návrh funkce aplikace ve spojení s centrálním systémem a parkovištěm

K popisovanému konceptu je třeba navrhnout, jak mobilní aplikace může fungovat ve spojení s centrálním řídicím systémem a parkovací věží. Aplikace umožňuje uživatelům interagovat s parkovacím a nabíjecím systémem pomocí jednoduchého a intuitivního rozhraní. Aplikace se skládá ze čtyř hlavních částí: rezervace místa dopředu, zaparkování vozidla (příjezd), nastavení nabíjecích možností a vyparkování (odjezd).

Je ideální, aby základní stav parkovací věže byl takový, že bude vždy připraveno volné místo, aby řidič vozidla mohl vjet a zaparkovat.

Instalovaná aplikace je převážně určena pro řidiče, kteří budou parkoviště využívat opakovaně. Například zaměstnanec okolních budov, který zde bude parkovat během své práce, nebo rezident, který bude parkoviště využívat pro nabití a parkování přes noc. Pro turisty nebo návštěvníky, kteří parkoviště využijí pouze jednou, tato aplikace může být použita také, ale nemá význam ji instalovat. Pro takové případy je lepší použít webovou verzi aplikace, která je dostupná i bez instalace. Pro motivaci k instalaci aplikace by mohla webová verze nabízet méně funkcí, zobrazovat reklamy nebo naopak by instalovatelná aplikace mohla nabízet bonus za stažení.

7.1.1 Rezervace

Aby se zabránilo tomu, že řidič přijede k parkovišti a zjistí, že už není žádné volné místo, je v aplikaci možnost si dopředu rezervovat parkovací místo. Při rezervaci

místa přestane být místo nabízeno ostatním uživatelům a jsou účtovány poplatky za každou minutu rezervace.

7.1.2 Zaparkování vozidla

Při zaparkování řidič vjede vozem dovnitř věže, zabezpečí své vozidlo a připojí ho k nabíječce. Poté naskenuje QR kód na obrazovce. Přes ten se uživateli přidá do aplikace jeho zaparkované vozidlo a uživatel pouze v aplikaci zadá požadované možnosti nabíjení, jako jsou nabíjecí limit nebo plánovaný čas odjezdu. Poté se parkovací věž vozidlo uloží, aby již k vozidlu nebyl přístup. V případě rotačního systému se otočí, v případě paletového a jiného systému vozidlo uloží na místo. Zaparkované vozidlo s daty od uživatele se zapíše do databáze.

Data zadaná uživatelem využije centrální řídicí systém k plánování vhodné doby a rychlosti nabíjení. Pokud například uživatel přijede v pátek a bude chtít odjet až v pondělí, řídicí systém zatím může nabíjet ostatní vozidla, která to potřebují více a toto vozidlo zatím nenabíjet, nebo nabíjet pomalou konstantní rychlostí. Tím se potenciálně zmenší vytížení sítě, pokud by byla vytížena, a zrychlí nabíjení ostatních vozidel.

7.1.3 Nastavení nabíjení

Během parkování bude mít uživatel v aplikaci možnost měnit a přizpůsobovat nabíjení jeho potřebám. Vybrané možnosti se předají centrálnímu řídicímu systému, který ovládá nabíjecí stanice a je schopen podle zadaných možností optimalizovat nabíjení připojených vozidel. Mezi tyto možnosti patří například změna požadovaného nabíjecího limitu (vybraného při přidání vozidla), plánovaného času odjezdu, povolení V2G pro ukládání energie v baterii vozidla a zpětnému využití v případě potřeby, nebo nastavení nabíjecích režimů jako jsou nabíjení pouze z obnovitelných zdrojů, nabíjení pouze v přebytku energie, nebo třeba nabíjení za levný proud.

Při zvolení těchto možností v aplikaci je zpracuje řídicí systém, který pravidla použije k optimalizaci nabíjecích stanic. Centrální řídicí systém monitoruje stav baterie, spotřebu energie a dojezd každého vozidla a upravuje rychlost a směr

nabíjení podle potřeby. Všechny požadavky na změny jsou přes OCPP protokol zaslány nabíjecím stanicím.

7.1.4 Odjezd

Když si uživatel přeje své vozidlo vyparkovat, zvolí v aplikaci možnost odjezdu. Tato možnost mu zobrazí okno pro platbu a po úspěšné platbě pomocí karty, Apple Pay nebo Google Pay parkovací věž přistaví požadované vozidlo do vjezdového/výjezdového prostoru. Uživatel poté odpojí nabíjecí kabel a vyjede vozidlem ven. Ideálně za použití senzorů se zjistí, zda uživatel odjel a místo se v databázi označí jako dostupné, takže může být znovu rezervováno i použito.

7.2 Funkční požadavky aplikace

Hlavním účelem mobilní aplikace je umožnit ovládání chytrého parkoviště a jeho nabíječek. To umožní lepší optimalizaci nabíjení, sledování aktuálního stavu zaparkovaného vozidla, prodloužit či zkrátit dobu parkování, zaplatit za nabíjení a přistavit zaparkované vozidlo. Byly proto specifikovány následující funkční požadavky:

7.2.1 Administrace účtu

- Registrace nového účtu
- Přihlášení k účtu
- Resetování hesla
- Odhlášení

7.2.2 Správa vozidel

- Skenování QR kódu pro rychlé přidání zaparkovaného vozidla k účtu
- Umožnit uživatelům přidat, upravit a odstranit informace o vozidlu

7.2.3 Správa parkovacího místa

- Umožnit uživatelům rezervovat parkovací místo v určitém čase a datumu.
- Umožnit uživatelům přistavit své vozidlo (ukončit parkování)

7.2.4 Ovládání nabíjení

- Umožnit uživatelům nastavit čas odjezdu, takže systém nabíjení může automaticky plánovat optimální dobu nabíjení
- Umožnit uživatelům zvolit nabíjecí režim podle jejich preference
- Umožnit uživatelům zapnout nebo vypnout V2G funkci, která umožňuje, aby energie z jejich baterie byla použita pro nabíjení ostatních vozidel v případě energetické špičky či nedostatku kapacity sítě

7.2.5 Monitorování stavu baterie

- Umožnit uživatelům sledovat aktuální stav baterie svého vozidla v reálném čase

7.2.6 Oznámení

- Posílat uživatelům oznámení o stavu nabíjení, dokončení nabíjení, změnách v rezervaci parkoviště atd.

7.2.7 Platby

- Umožnit uživatelům provádět platby za parkování a nabíjení.
- Umožnit zobrazení aktuální ceny
- Umožnit zobrazení historie plateb

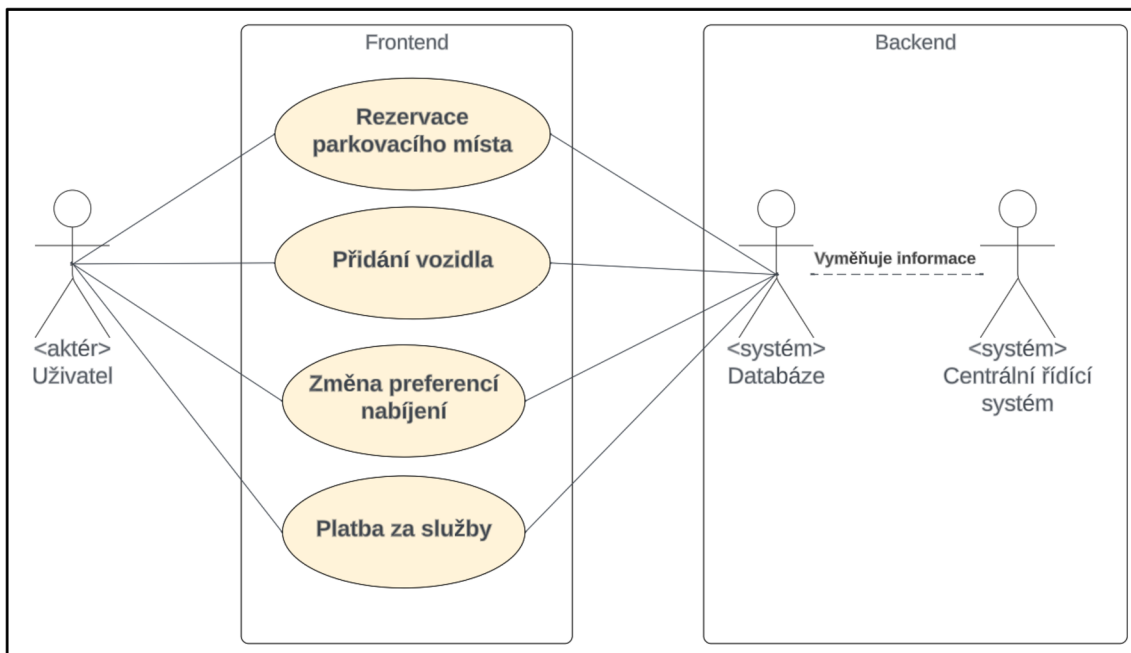
7.2.8 Získávání informací o parkovišti

- Poskytnout uživatelům aktuální informace o dostupnosti parkovacích míst, stavu nabíjecích stanic atd.

7.2.9 Navigace k parkovišti

- Poskytnout uživatelům navigační instrukce k parkovišti.

Podle těchto požadavků byl vytvořen zjednodušený use case diagram, který znázorňuje nejdůležitější akce uživatele. Ty přes databázi komunikují s centrálním řídicím systémem.



Obrázek 15 - Zjednodušený use case diagram (zdroj: vlastní zpracování)

7.3 Non-funkční požadavky aplikace

Non-funkční požadavky se týkají kvality nebo omezení systému, nikoli jeho funkcí. Mohou mít vliv na výběr technologií, návrh architektury nebo testování aplikace. Pro aplikaci sloužící k obsluze parkovacího místa, kam uživatel zaparkuje své vozidlo, může jeho nabíjení vzdáleně spravovat a následně si vozidlo zavolat, byly specifikovány následující požadavky:

- **Výkon** – Pro možnost zaparkování a následně vyparkování svého vozidla musí být aplikace rychlá, aby akcemi uživatel neztratil mnoho času a neblokoval ostatní řidiče.
- **Bezpečnost** – Pro citlivé údaje, platby a polohu je důležité, aby data byla bezpečně uložena a nedošlo k jejich odcizení či neoprávněnému přístupu.
- **Podporovaná zařízení** – Aplikace bude určena pro mobilní zařízení a web.
- **Multiplatformnost** – Je důležité, aby aplikace fungovala na nejrozšířenějších mobilních platformách. V tuto chvíli se tedy jedná o iOS a Android.

Právě kvůli multiplatformnosti byl zvolen framework Flutter. Ten navíc splňuje všechny další požadavky a je poměrně jednoduchý.

8 Vývoj mobilní aplikace

Při vývoji aplikace byl záměr se zaměřit především na parkoviště pro elektromobily. V aplikaci je tedy přizpůsobena právě elektromobilům. Vývoj aplikace i pro běžné automobily by zvedlo komplexitu aplikace bez přidaného užítku, jelikož všechny základní funkce chytrého parkoviště, které by měly běžné automobily mají dostupné i elektromobily.

Cílem je vytvoření aplikace, která umožňuje uživateli spravovat nabíjení svého zaparkovaného elektromobilu, stanovit cíle a tím pomoci centrálnímu řídicímu systému optimalizovat nabíjení připojených vozidel.

Aplikaci lze nainstalovat na operační systémy Android a iOS uživatele, kteří budou využívat parkování opakovaně (například rezidenti). Pro jednorázové uživatele lze také aplikaci spustit ve webovém rozhraní přes prohlížeč. Díky tomu není potřeba instalovat aplikaci, kterou by uživatel využil jednou. Výsledkem této kapitoly bude funkční aplikace, ve které uživatel bude moci rezervovat dopředu parkovací místo, přidat vozidlo po zaparkování, měnit preference nabíjení svého vozidla a zaplatit za služby. Aplikace bude dostupná pro stažení z GitHub² a i jako webová aplikace³.

8.1 Použité technologie

Aplikace je implementována ve frameworku Flutter za použití nástroje FlutterFlow. Přihlašování a databáze jsou implementovány přes Firebase.

8.1.1 Flutter

Jedná se o otevřený framework od společnosti Google. Umožňuje vytvářet multiplatformní aplikace pomocí jazyka Dart a kompiluje do strojového kódu, kterému hostitelská zařízení rozumějí. To zajišťuje rychlý a efektivní výkon. Dart je velice podobný jazyku Java nebo C. Flutter se používá k vývoji aplikací pro různé platformy, jako jsou webový prohlížeč, Android, iOS, Linux, Windows i macOS. [49] Flutter má single codebase, což znamená že stačí napsat pouze jeden kód a ten lze

² Dostupné na https://github.com/strihavkaJakub/optimized_parking_app

³ Dostupné na <https://parking-app-gtnu18.flutterflow.app/>

použit pro všechny podporované platformy. Flutter místo spoléhání na platformní specifické nástroje pro vykreslování používá grafickou open-source knihovnu Skia od Googlu. Ta poskytuje uživatelům konzistentní vizuální efekty bez ohledu na jejich platformu, což pomáhá multiplatformnosti. [50] Vývoj ve Flutteru probíhá za pomoci widgetů, které se do sebe zanořují. Vše ve Flutteru je widget a jedná se o stavební prvky celé aplikace. Koncept může být přirovnán HTML struktuře, kde se do sebe zanořují `<div>`, `<table>`, `<p>` a jiné elementy. Flutter obsahuje mnoho předdefinovaných widgetů, ze kterých lze vytvořit jednoduché i komplexní aplikace. Kromě toho lze vytvořit vlastní widgety a používat je opakovaně v různých částech aplikace. [51]

Na obrázku č. 16 lze vidět vygenerovanou předlohu aplikace pro zobrazení seznamu. Na této stránce je Scaffold widget, což je hlavní widget umožňující AppBar, tělo a spodní navigační lištu. AppBar je vrchní lišta s názvem a možností akcí. Nejčastěji je jednou z těchto akcí šipka zpět na předešlou stránku. V tělu lze použít libovolný widget. V tomto případě byl použit widget ListView, který slouží pro zobrazení seznamu.

```

class SampleItemListView extends StatelessWidget {
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Sample Items'),
        actions: [
          IconButton(
            icon: const Icon(Icons.settings),
            onPressed: () {
              Navigator.restorablePushNamed(context, SettingsView.routeName);
            },
          ), // IconButton
        ],
      ), // AppBar
      body: ListView.builder(
        restorationId: 'sampleItemListView',
        itemCount: items.length,
        itemBuilder: (BuildContext context, int index) {
          final item = items[index];

          return ListTile(
            title: Text('SampleItem ${item.id}'),
            leading: const CircleAvatar(
              foregroundImage: AssetImage('assets/images/flutter_logo.png'),
            ), // CircleAvatar
            onTap: () {
              Navigator.restorablePushNamed(
                context,
                SampleItemDetailsView.routeName,
              );
            }
          ); // ListTile
        }
      );
    }
  }
}

```

Obrázek 16 - Příklad Flutter kódu (zdroj: vlastní zpracování)

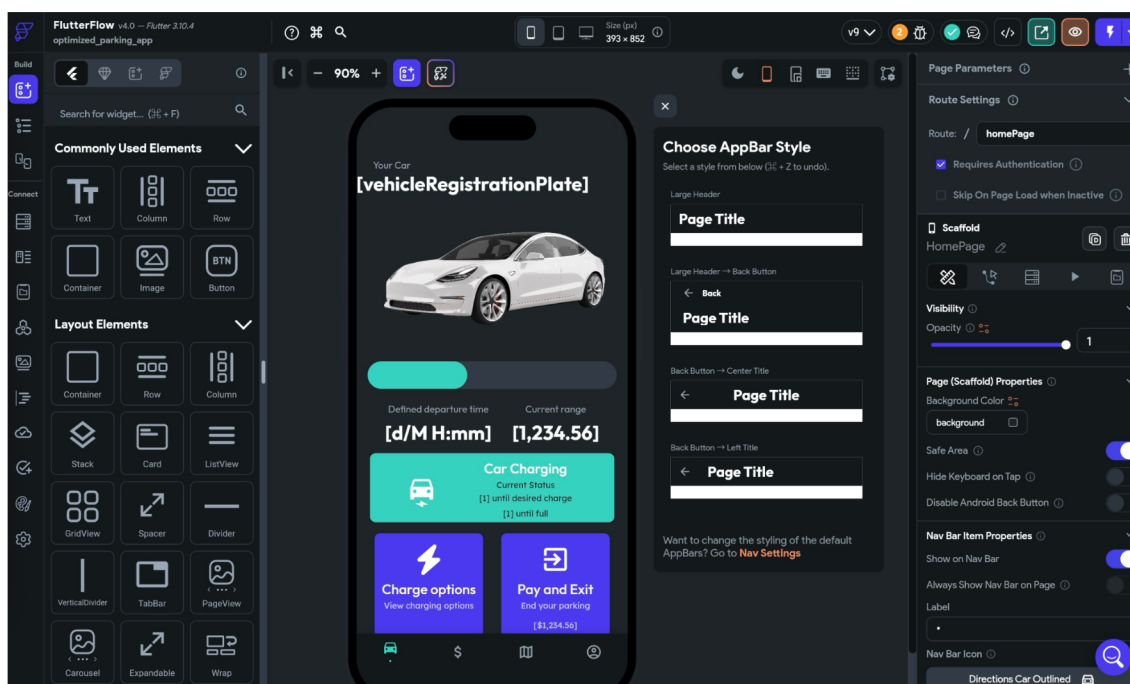
8.1.2 FlutterFlow

FlutterFlow je online nástroj pro vytváření nativních aplikací pro různé platformy právě díky Flutter frameworku. Umožňuje uživatelům využít jednoduché rozhraní typu “Drag and Drop” (neboli přetáhni a pusť) k velice rychlému a snadnému sestavení aplikace bez nutnosti psát kód. Uživatelům nabízí přidání a úpravu prvků uživatelského rozhraní, jako jsou tlačítka, textová pole, obrázky, seznamy a další. Uživatelé mohou nastavovat vlastnosti, jako jsou barvy, velikosti, zarovnání a animace. Další výhodou je snadná integrace dat a API. FlutterFlow podporuje Firebase a umí ho nastavit automaticky a propojit již při vytváření aplikace. Podporuje také využití API třetí strany či vlastní. Také je možnost snadno a rychle přeložit celou aplikaci do jiného jazyka pomocí integrovaného Google Translate.

FlutterFlow generuje velice čistý kód, který lze exportovat, upravit, stáhnout, nahrát na GitHub či rovnou nasadit do obchodu s aplikacemi jako je Apple AppStore nebo Google Obchod Play. [52–54] Velkou výhodou je, že veškerý kód a aplikace které se pomocí tohoto frameworku vytvoří jsou majitelem toho, kdo kód napsal. FlutterFlow si nenárokují žádná práva na kód v něm vytvořený. [55]

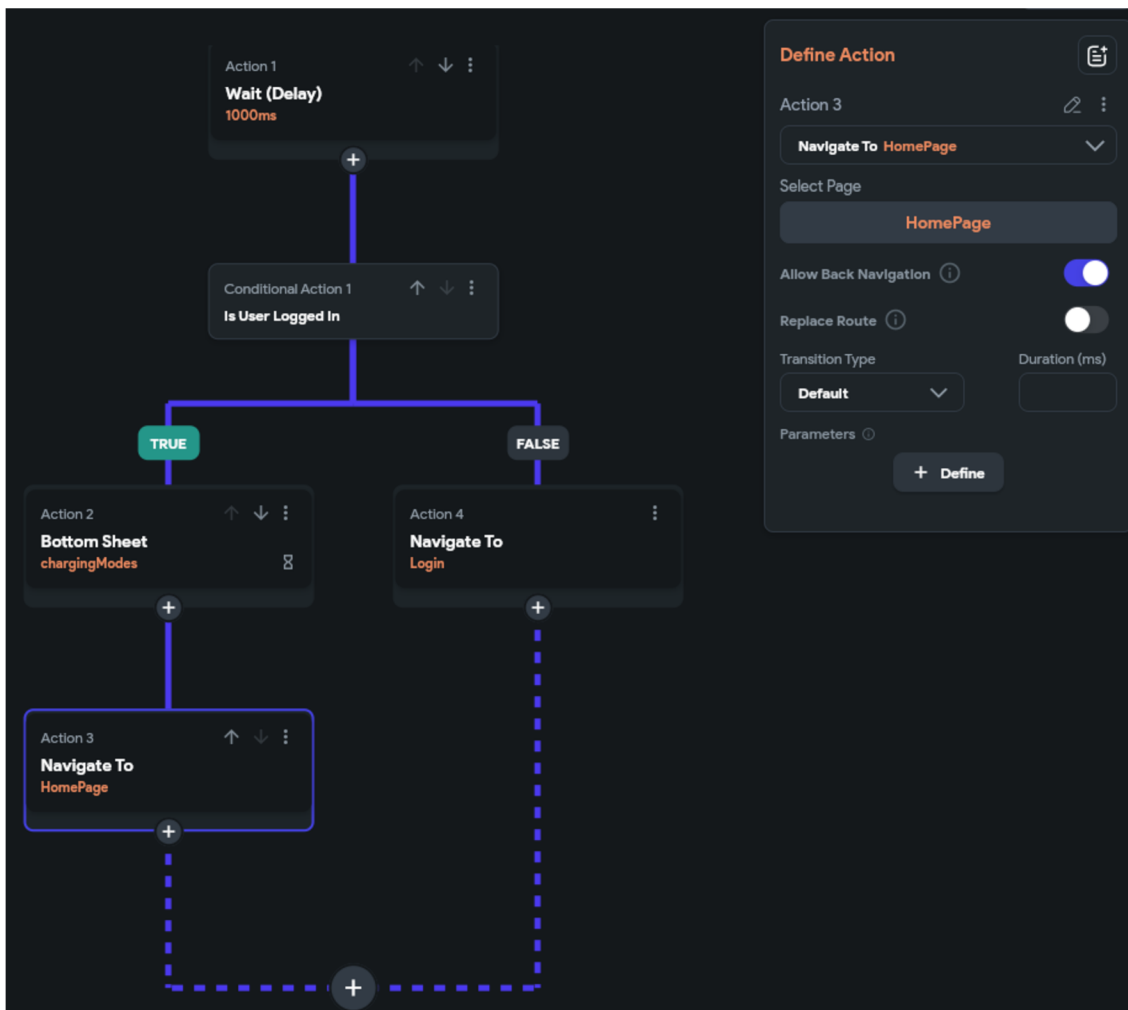
FlutterFlow umožňuje jednoduché přidání funkcí, změn v databázi i spuštění a testování vytvořené aplikace.

Práce ve FlutterFlow je znázorněna na obrázku č. 17. Veškeré widgety je možné ihned přesunout do prostřední obrazovky, poskládat je podle potřeby a přidat jim požadovanou funkcionalitu, vzhled a vlastnosti.



Obrázek 17 - Vývojové prostředí FlutterFlow (zdroj: vlastní zpracování)

Logika a funkčnost aplikace se ve FlutterFlow řeší pomocí akcí. Ty se dají přidat na akce od uživatele, jako třeba kliknutí, dvojitě kliknutí, dlouhé kliknutí, změnu formuláře, nebo na jiné akce jako dokončení animace, změna dat a mnoho dalších. Akce umožňují provádět mnoho operací, jako je například otevření nové stránky, nastavení proměnné, volání API, spuštění Firebase funkcí, nebo třeba změna dat v databázi. Akce jsou jednoduché k nastavení a lze je kombinovat do sekvencí, přidat podmínky i smyčky. [56]



Obrázek 18 - Příklad akcí ve FlutterFlow (zdroj: vlastní zpracování)

Vytvořená aplikace lze kdykoliv spustit v testovacím režimu, ve kterém se po pár minutách zkompiluje a spustí v simulátoru, který funguje velice podobně jako fyzické zařízení.

8.1.3 Firebase

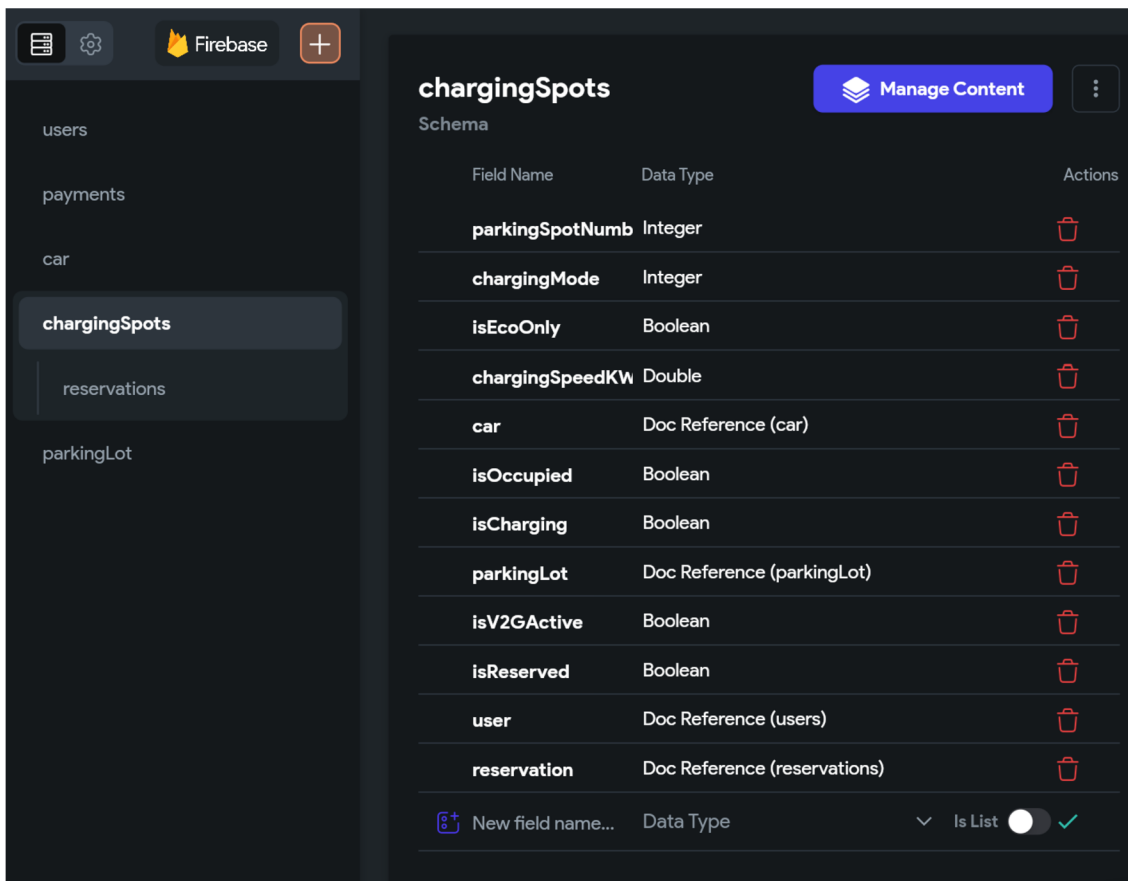
Firebase je platforma pro vývoj mobilních a webových aplikací, kterou poskytuje firma Google. Nabízí řadu služeb, které usnadňují tvorbu, testování, nasazování a škálování aplikací. Některé z hlavních funkcí jsou:

- Realtime Databáze – jedná se o cloudovou databázi, která umožňuje ukládat a synchronizovat data mezi různými zařízeními v reálném čase. Data jsou uložena ve formátu JSON. [57]

- Autentizace – Tato služba zajišťuje ověřování identity uživatelů aplikace. Firebase podporuje různé metody autentizace, jako je email, heslo, telefonní číslo, sociální sítě (Facebook, Twitter, GitHub) nebo i Google účet. [58]
- Cloud Firestore – Další cloudová NoSQL databáze, ale nabízí flexibilnější a škálovatelnější řešení než Realtime Databáze. Umožňuje ukládat a dotazovat se na strukturovaná a hierarchická data pomocí SQL-like dotazovacího jazyka. Všechna data jsou ve formě dokumentů a kolekcí. [59]
- Cloud Functions – Funkce, které běží v cloudu a reagují na různé události v aplikaci. Umožňují provádět logiku na serveru (backend) bez nutnosti spravovat infrastrukturu. Lze je psát v jazycích JavaScript, TypeScript nebo Dart. Kód je uložen v Google cloudu a běží ve spravovaném prostředí, takže není třeba spravovat a škálovat vlastní servery. [60]
- Cloud Messaging – Tato služba umožňuje posílat zprávy a notifikace mezi zařízeními nebo serverem. [61]

Dále Firebase nabízí nástroje na sledování a řešení chyb v aplikaci (Crashlytics), měření a optimalizování výkonu aplikace (Performance Monitoring), nebo také nástroj na analyzování chování uživatelů aplikace (Google Analytics) [62]

Přístup do Firebase databáze je možný přímo z FlutterFlow. Lze tak přímo při vývoji aplikace měnit schéma databáze i konkrétní data.



Obrázek 19 - Databáze v prostředí FlutterFlow (zdroj: vlastní zpracování)

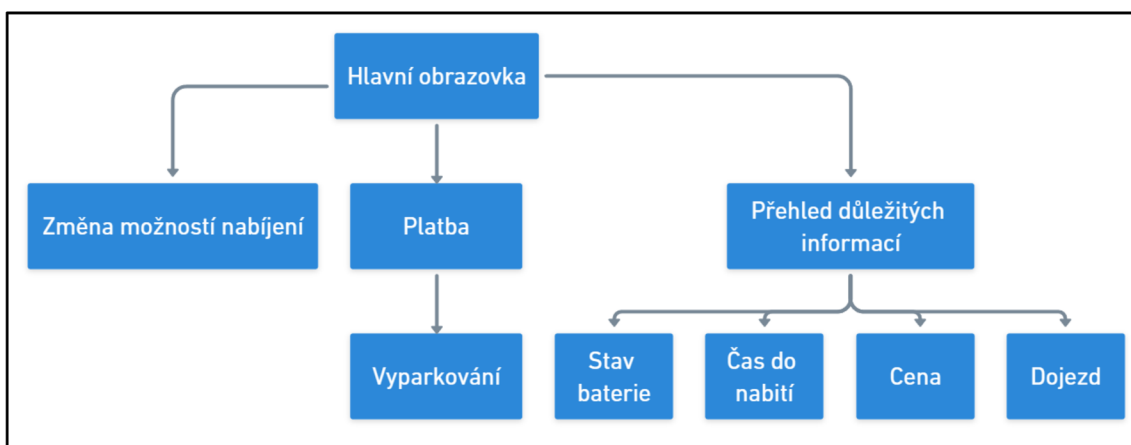
8.2 Návrh obrazovek

Po rozmyšlení, co vše má aplikace umět je nutné tyto informace správně implementovat a zobrazit. Význam dobře navrženého uživatelského rozhraní spočívá v jeho schopnosti poskytovat intuitivní, efektivní a příjemný zážitek z používání. K tomuto účelu lze použít drátěný model, ve kterém je vidět návrh jednotlivých obrazovek, jejich vzhled a přibližná funkce. Možností je na internetu nespočet. Příkladem je například Figma, Whimsical, Wireframe a mnoho dalších.

Zvolen byl nástroj Whimsical pro jeho snadnou registraci. V tomto nástroji byly navrženy modely základních funkcí obrazovek a jejich propojení s ostatními obrazovkami.

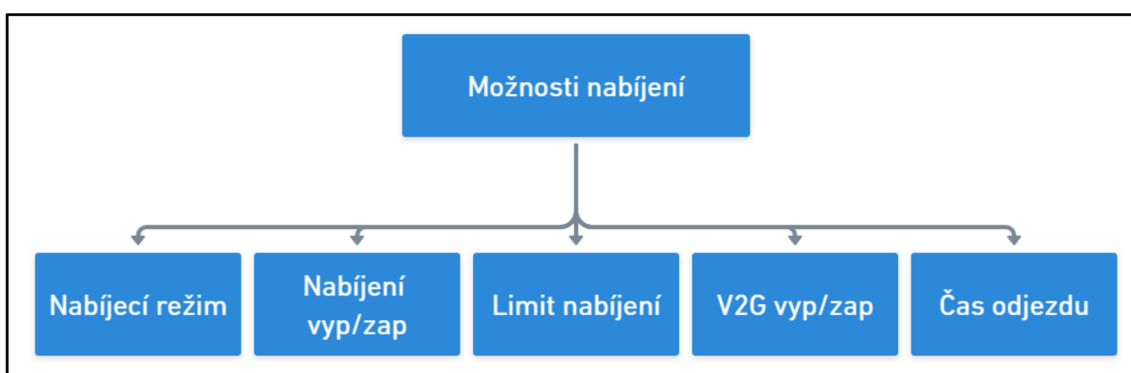
Hlavní obrazovka byla navržena tak, aby zobrazovala důležité informace o vozidle s možností rychlého přístupu do nejdůležitějších částí aplikací. To umožní snadné a

intuitivní ovládání. Pro snazší plánování a přehled byl vytvořen model důležitých funkcí hlavní obrazovky.



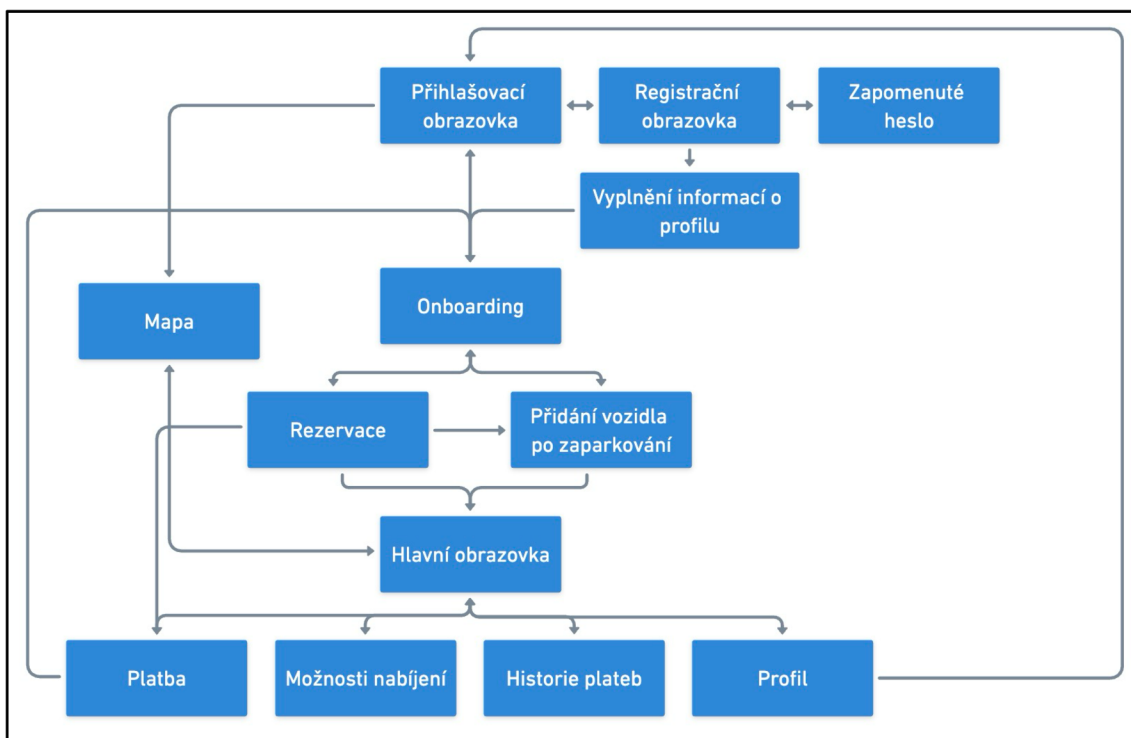
Obrázek 20 - Model základních funkcí hlavní obrazovky (zdroj: vlastní zpracování)

Z hlavní obrazovky by měly být přístupné možnosti nabíjení. Pro rychlou změnu možností nabíjení byla navržena obrazovka, na které budou ve formě tlačítek dostupné jednotlivé možnosti. Mezi těmito možnostmi by měla být možnost pro změnu nabíjecího režimu, vypnutí/zapnutí nabíjení, nastavení maximálního limitu pro nabíjení, volba času odjezdu a povolení/zakázání V2G pro umožnění uchovávání přebytečné energie ve vozidle baterie a poté ji využít pro nabíjení jiných vozidel v případě potřeby. Pro možnost zaplatit za služby musí být implementována i možnost platby



Obrázek 21 - Model možností nabíjení (zdroj: vlastní zpracování)

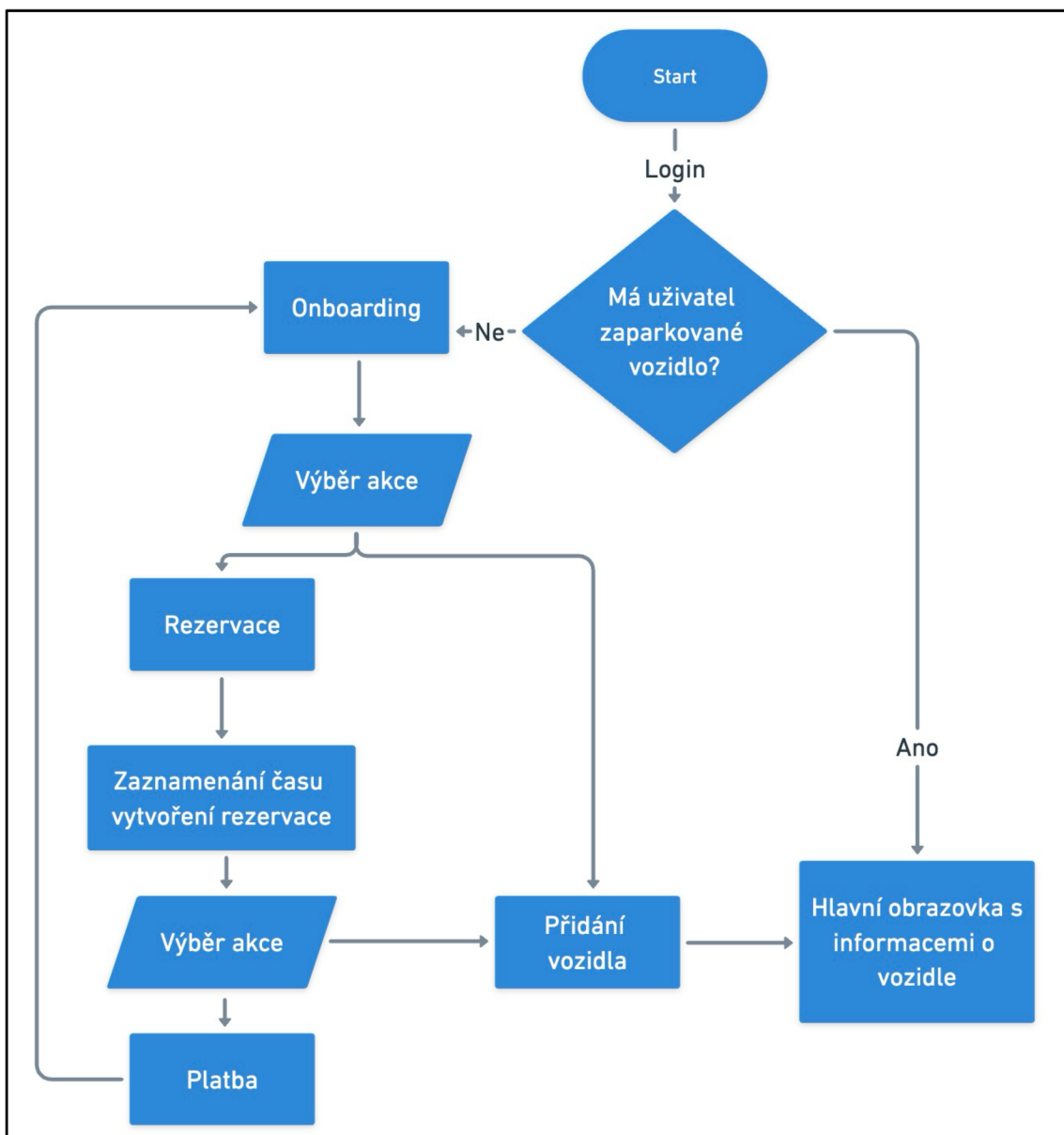
Pro znázornění potřebných obrazovek a jejich propojení s ostatními obrazovkami byl vytvořen jednoduchý drátový model. Na tomto modelu lze vidět názvy jednotlivých obrazovek a přechody mezi nimi.



Obrázek 22 - Drátěný model obrazovek a možných přechodů mezi nimi (zdroj: vlastní zpracování)

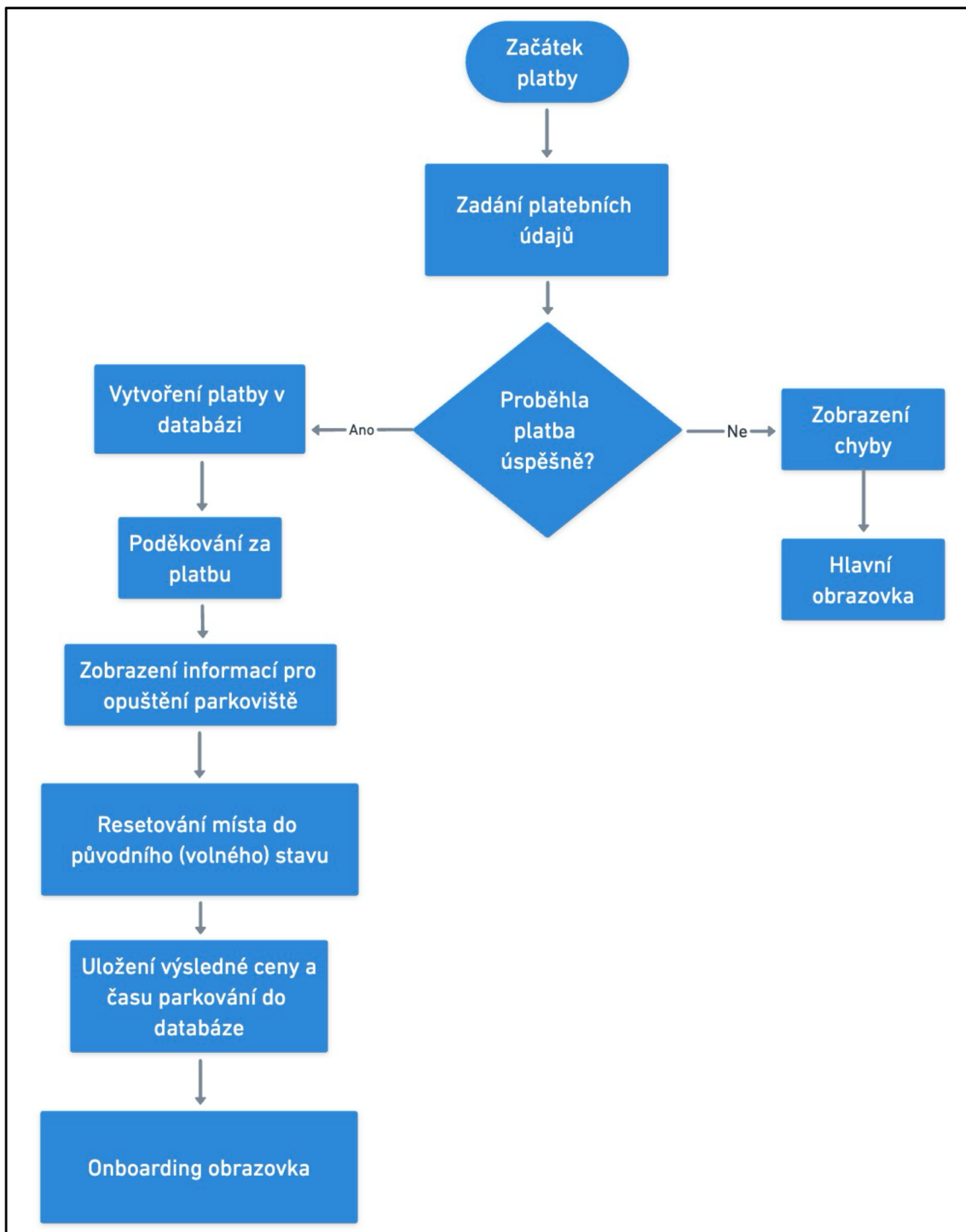
Uživatel by měl mít možnost z přihlašovací obrazovky přejít na obrazovku vytvoření účtu, resetování hesla, nebo pokračovat na Onboarding stránku. Ta slouží jako rozcestník pro rezervaci parkovacího místa nebo přidání vozidla po zaparkování. Při vytvoření rezervace je do databáze k ní zapsán čas vytvoření. Od tohoto času se bude počítat čas za rezervaci. Během rezervace má uživatel možnost zrušit rezervaci a zaplatit, nebo přidat vozidlo po zaparkování, pokud na parkovací místo dorazil.

Na obrazovce přidání vozidla uživatel vyplní potřebné informace a je přesměrován na hlavní obrazovku. Průchod uživatele aplikací po přihlášení až po hlavní obrazovku je popsán vývojovým zjednodušeným vývojovým diagramem.



Obrázek 23 - Vývojový diagram průchodu aplikací od přihlášení až po hlavní obrazovku (zdroj: vlastní zpracování)

Navržen byl také diagram pro případ platby a následného opuštění parkoviště. Tento diagram se ve výsledné aplikaci stal složitější s větším počtem interakcí a možností, ale jeho hlavní body jsou stále dodržovány.



Obrázek 24 - Vývojový diagram provedení platby (zdroj: vlastní zpracování)

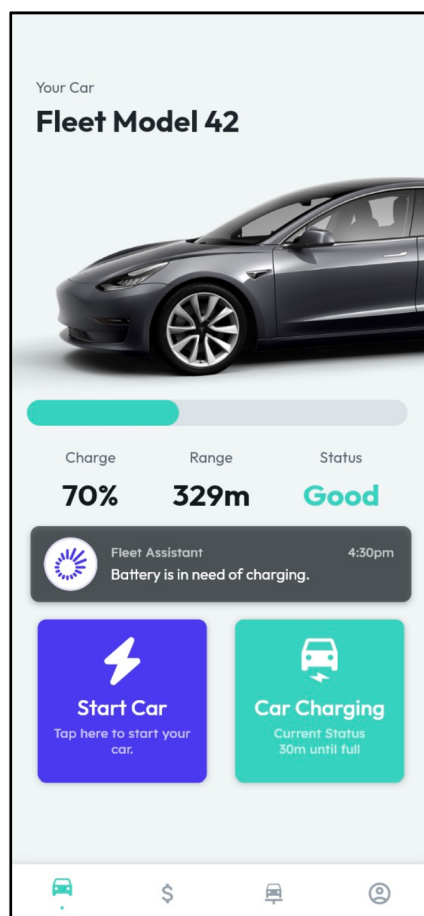
Modely byly původně realizovány do aplikace přímo psaním Flutter kódu ve vývojovém prostředí. Nicméně při pokusu udělat aspoň základní funkční kostru aplikace byl na internetu nalezen framework FlutterFlow, který již má předpřipravené šablony a nastýlované elementy přímo od FlutterFlow vývojářů i

ostatních uživatelů tohoto frameworku. To velice ulehčilo práci jak s návrhem a stylováním, tak s programováním aplikace.

8.3 Implementace aplikace

Vývoj proběhl ve webové aplikaci FlutterFlow a je založen na volně dostupné šabloně Fleet Cars. Ta převážně pomohla při vytvoření vzhledu aplikace a je určena pro správu vozidel vozového parku a sledování jejich servisních prohlídek. V této šabloně jsou již vyřešené obrazovky pro přihlašování, registraci, přidávání vozidel a platby. Je tedy ideální k použití pro potřeby této práce. Většina obrazovek a funkcionality bylo přesto potřeba upravit a přidat.

Na obrázku je znázorněn vzhled hlavní stránky šablony Fleet Cars. Tato hlavní stránka je pouze šablona, a tak veškeré texty jsou napsané přímo a nejsou proměnné.

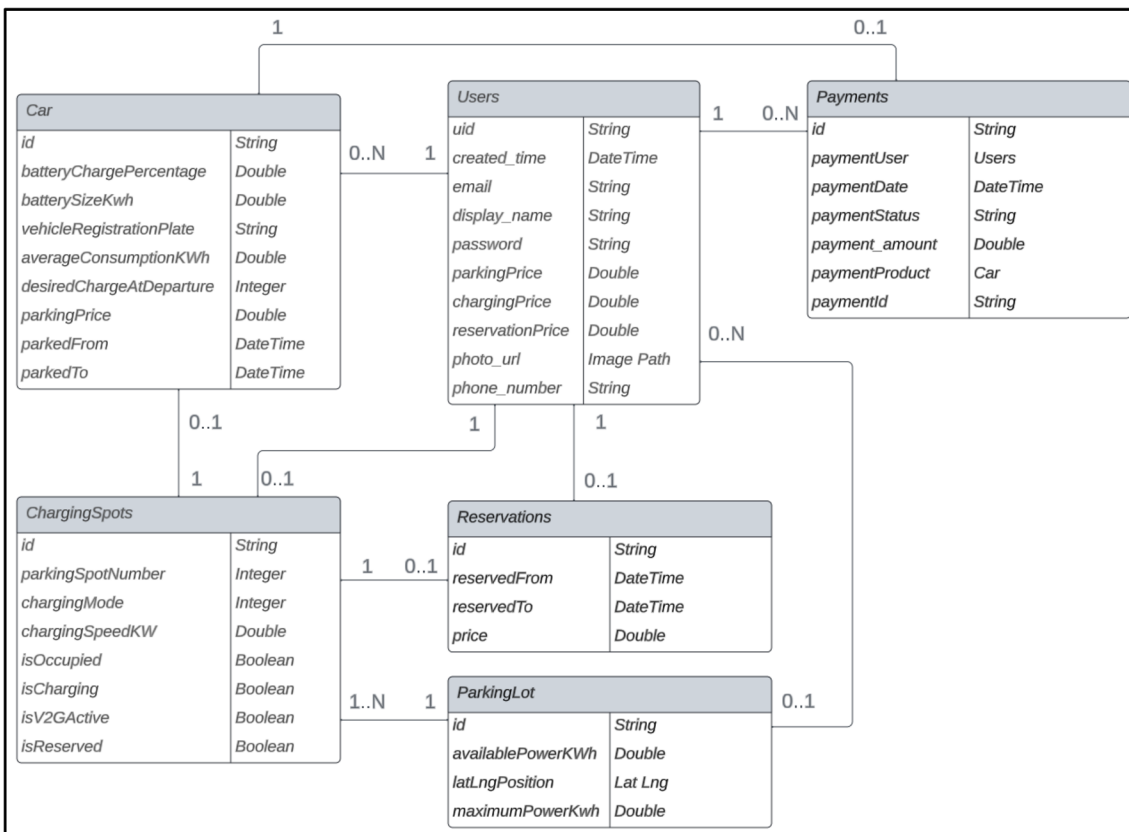


Obrázek 25 - Hlavní obrazovka šablony Fleet Cars (zdroj: vlastní zpracování)

První věcí bylo vytvoření účtu na FlutterFlow a projektu. Už při vytváření projektu FlutterFlow umožňuje propojit aplikaci s Firebase pro přihlašování i databázi. Buď již k existujícímu, nebo je možné pár kliknutími vytvořit a nastavit nový. Jednoduché návody potom provází nastavením a propojením a vše je během chvíle hotovo a připraveno k použití s funkční databází i přihlašováním.

Druhou důležitou věcí je vytvoření schématu databáze. K tomu byly definovány tyto kolekce (přehled kolekcí lze také vidět na obrázku č. 26):

- Users
- Car – kolekce vozidel
- ChargingSpots – kolekce parkovacích míst (neboli nabíjecích míst)
- Reservations
- Payments
- ParkingLot – tato kolekce slouží pro případ správy více parkovišť či nabíjecích věží. Obsahuje maximální možný nabíjecí výkon parkoviště, cenu rezervace za minutu a souřadnice parkoviště pro navigaci k němu.



Obrázek 26 - Diagram databázové struktury (zdroj: vlastní zpracování)

Jako databáze se využívá Cloud Firestore od Firebase, ve které jsou data hned synchronizovány a všechny změny se hned v aplikaci projeví.

Veškeré obrázky použité v aplikaci jsou staženy z aplikace Shutterstock během zkušebního období a mají standardní licenci, což znamená neomezené použití pro aplikace a web, a také tištěné materiály pod 500 000 výtisků. [63]

Kód Flutter aplikace je velice rozsáhlý a zabírá mnoho řádků. Není tedy ideální, aby zde byl vložen kód pro celé obrazovky. Z tohoto důvodu je alespoň vložen kód nejdůležitějších funkcí a widgetů. Celý kód aplikace je k dispozici na GitHub⁴. API klíče pro mapy, Firebase databázi a přihlašování jsou umístěny přímo v kódu aplikace. Z důvodu zabezpečení těchto klíčů před zneužitím budou tyto klíče po obhajobě práce zrušeny, nebo striktněji zabezpečeny. To pravděpodobně způsobí nefunkčnost nasazené webové aplikace.

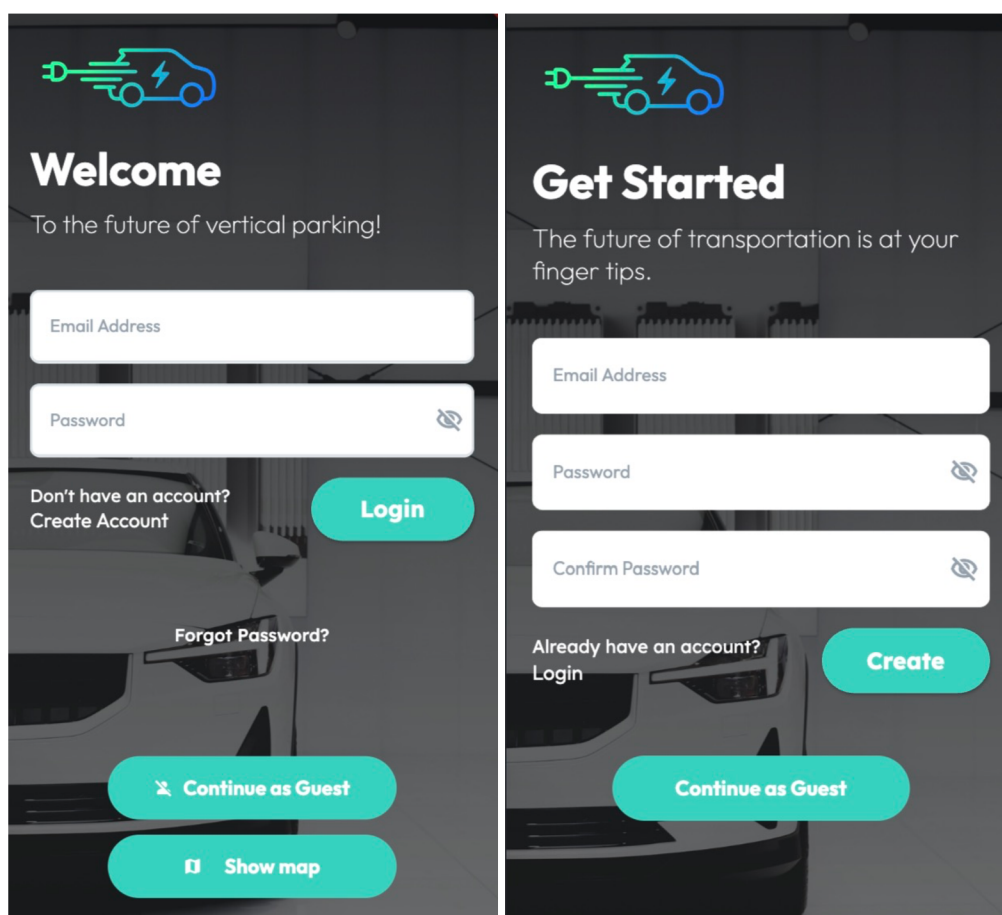
⁴ Volně přístupný na https://github.com/strihavkaJakub/optimized_parking_app

8.3.1 Vstupní obrazovka – přihlášení a registrace

Pro používání aplikace je potřeba si vytvořit účet. Účet je možné zaregistrovat pomocí emailu a hesla přes Firebase, ale jednoduše lze přidat i možnost přihlášení přes Apple, Google nebo Facebook. Tyto možnosti však v této práci implementovány nebyly, jelikož by přinesly zbytečnou a nepotřebnou složitost. Kromě vytvoření vlastního účtu lze také v aplikaci použít možnost “Continue as Guest”. Tato možnost vytvoří anonymní účet a aplikaci lze začít hned používat. Funkcionalita aplikace zůstane stejná.

Na obrazovce přihlášení je také možnost zobrazit mapu pro snadné nalezení parkoviště a navigaci k němu.

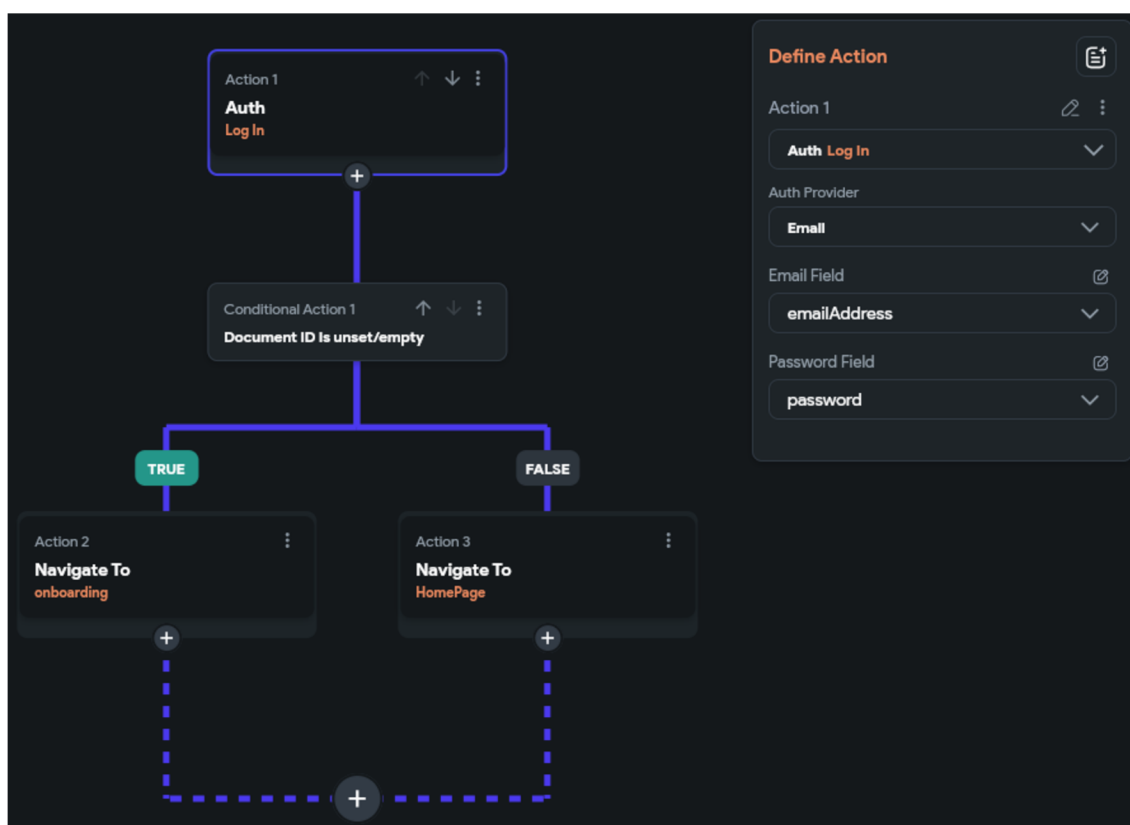
Tyto obrazovky mají jako vstupní parametry email a heslo uživatele.



Obrázek 27 - Obrazovky přihlášení a registrace (zdroj: vlastní zpracování)

Po zadání údajů a odeslání formuláře je zavolán strom akcí. První akce zkusí autorizovat uživatele pomocí zadaného emailu a hesla. V případě neúspěchu je

vyhozena chybová hláška a kód dále nepokračuje. Pokud tato akce proběhne úspěšně, je zjištěno, zda uživatel má již přidání vozidlo. V případě že nemá, je uživatel přesměrován na onboarding stránku, kde si může vozidlo po zaparkování přidat, nebo rezervovat parkovací místo. V případě, že již vozidlo má, je přesměrován na hlavní obrazovku s informacemi o vozidle.



Obrázek 28 - Akce pro přihlašování ve FlutterFlow (zdroj: vlastní zpracování)

Kód widgetu login tlačítka vypadá včetně akcí následovně. Ukázka je pro zjednodušení zkrácena o stylování tlačítka, jelikož kompletní kód widgetu zabírá 40 řádků. Celý kód vstupní obrazovky je dlouhý 492 řádků a nebyl proto uveden.

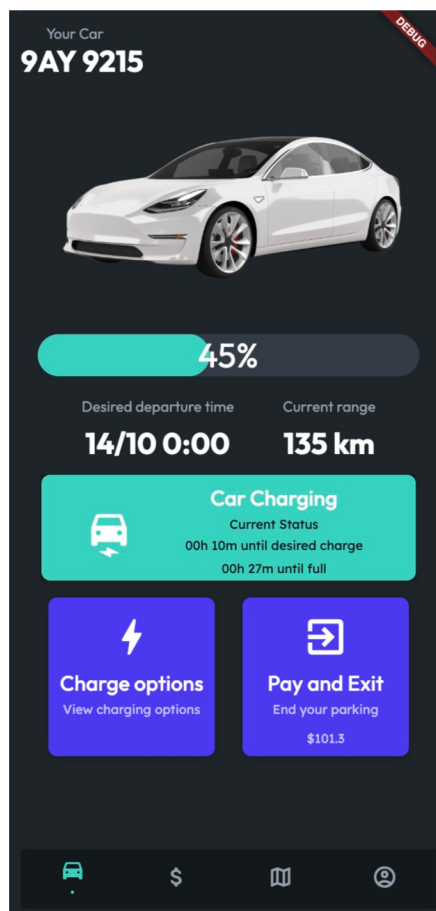

```

FFButtonWidget(
  onPressed: () async {
    GoRouter.of(context).prepareAuthEvent();
    final user = await authManager.signInWithEmail(
      context,
      _model.emailAddressController.text,
      _model.passwordController.text,
    );
    if (user == null) {
      return;
    }
    if (currentUserDocument?.car?.id == null ||
        currentUserDocument?.car?.id == "") {
      context.goNamedAuth('onboarding', context.mounted);
    } else {
      context.pushNamedAuth('HomePage', context.mounted);
    }
  },
  text: 'Login',
  (zkráceno o stylování tlačítka)
)

```

8.3.2 Hlavní obrazovka

Nejdůležitější částí aplikace je hlavní obrazovka. Ta umožňuje přehled o hlavních informacích zaparkovaného vozidla. Mezi ty patří aktuální nabití, naplánovaný čas odjezdu, aktuální dojezd, čas do nabití, rychlost nabíjení a cena za nabíjení a parkování. K tomu jsou zde odkazy pro nastavení nabíjení a zaplacení.



Obrázek 29 - Hlavní obrazovka aplikace (zdroj: vlastní zpracování)

V aplikaci je zbývající čas nabíjení kalkulován funkcí, která jako své vstupní argumenty:

- `currentCharge` (double) - aktuální procento nabití baterie
- `chargingSpeed` (double) - aktuální rychlost nabíjení v kW
- `desiredCharge` (int) - požadované procento nabití baterie
- `batterySizeKwh` (double) - maximální kapacita baterie v kWh

Tato funkce vypočte, kolik je potřeba dobít procent a potom převede procenta na kapacitu v kWh. Čas v minutách je roven šedesátinásobku podílu kapacity, kterou chce je potřeba nabít, a nabíjecí rychlosti. Logika výpočtu zbývajících času vypadá následovně:

```
if (desiredCharge > currentCharge) {  
    double percentToCharge = desiredCharge - currentCharge;  
    double capacityToCharge = batterySizeKwh * percentToCharge / 100;  
    double time = capacityToCharge / chargingSpeed * 60;  
    return getTimeStringFromMinutes(time);  
}  
return null;
```

Dostupný dojezd je vypočítáván funkcí calculateRange. Ta má vstupní argumenty:

- batterySizeKwh (double) - maximální kapacita baterie v kWh
- currentChargePercentage (double) - aktuální procento nabití baterie
- averageConsumptionKwh (double) - průměrná spotřeba kWh na 1 km

Vzorec pro výpočet je následující:

$$\text{dojezd} = \frac{\text{velikost baterie} * \text{aktuální procento nabití}}{\text{průměrná spotřeba}}$$

Celková cena za nabíjení není v aplikaci nijak počítána. Ta bude do databáze vložena centrálním řídicím systémem, podle reálné spotřeby energie a její aktuální ceny. Výsledná cen pro zaplacení je součet ceny za parkování, nabíjení a rezervaci.

Při kliknutí na tlačítko Charge Options je uživatel přesměrován na nabíjecí možnosti, ve kterém může ovládat nabíjení svého vozidla.

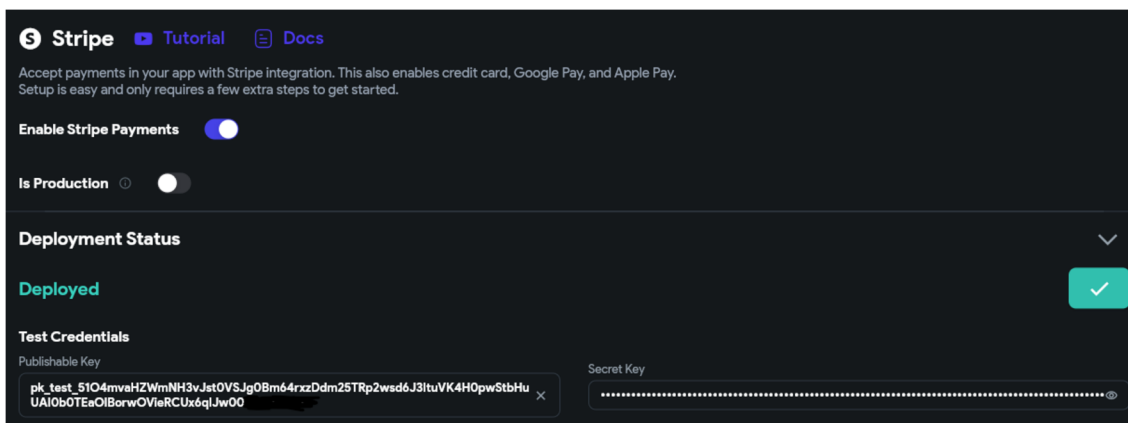
Při kliknutí na tlačítko Pay and Exit se zobrazí platební formulář. Po úspěšném zaplacení je uživatel přesměrován na děkovnou obrazovku a poté na obrazovku s instrukcemi, jak opustit parkoviště.

8.3.3 Platby

Důležitou funkcí aplikaci je možnost platby za rezervaci, parkování a nabíjení vozidla. Tato funkce umožňuje uživateli snadno a rychle zaplatit za služby, které využil. FlutterFlow nabízí jednoduchou implementaci plateb přes kreditní kartu, Apple Pay, Google Pay, PayPal, webové bankovníctví a další. Tyto služby jsou realizovány přes poskytovatele Stripe, Braintree nebo Razorpay. Ti umožňují bezpečné a spolehlivé zpracování plateb pro uživatele aplikace. Díky tomu je platba za parkování bezpečná a jednoduchá.

Pro implementaci je nutné se registrovat u jednoho z těchto poskytovatelů a poskytnuté API klíče, které jsou nalezeny v uživatelském profilu vložit do FlutterFlow nastavení. K dispozici je testovací i produkční prostředí.

Při vývoji aplikace byl použit poskytovatel Stripe. Pro integraci je zapotřebí registrace na stránce Stripe⁵. Po registraci jsou vygenerovány klíče pro testovací i produkční prostředí. Tyto klíče je třeba zadat do nastavení Stripe ve FlutterFlow.



Obrázek 30 - Stripe klíče v nastavení FlutterFlow (zdroj: vlastní zpracování)

Tím dojde ke spojení Stripe účtu s aplikací a je nyní možné začít používat Stripe widget pro platby.

Na webové stránce Stripe lze kdykoliv zobrazit dashboard, ve kterém lze sledovat výdělky a statistiky plateb.

⁵ Dostupné na <https://stripe.com/en-gb-cz>

Limitací Stripe je minimální částka pro zaplacení 50 centů. Kvůli tomuto omezení musely být ceny v aplikaci upraveny, aby byly minimálně 50 centů.

Pro testování byly použity předdefinované karty pro vývojáře dostupné na stránkách Stripe. K dispozici jsou 3 testovací karty:

- Karta pro úspěšné zaplacení (karta s číslem 4242 4242 4242 4242)
- Karta pro neúspěšné zaplacení (karta s číslem 4000 0025 0000 3155)
- Karta požadující další potvrzení (karta s číslem 4000 0000 0000 9995)

Po zadání některé z těchto karet v testovacím režimu je možné si ověřit funkcionální aplikaci.

Při platbě uživatel zadá číslo své karty, splatnost a CVC kód do formuláře, který se zobrazí. Po úspěšné platbě je pole **isOccupied** nastaveno na false a reference na vozidlo je smazáno z dokumentu parkovacího místa. Parkovací místo se tak stane dostupné pro další parkování.

Podobně tomu je i v případě platby za rezervované místo. Po platbě je z dokumentu parkovacího místa odebrána rezervace a pole **isReserved** je nastaveno na false.

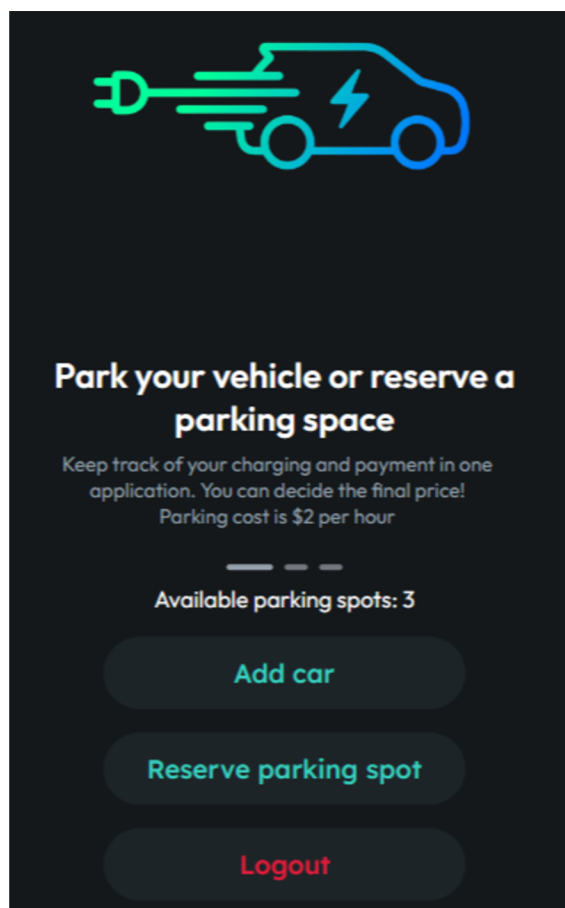
Widget pro platbu, do kterého se zadávají údaje o kartě, je přímo od Stripe. Uživatel tedy vkládá citlivé údaje přímo do platebního widgetu a údaje v aplikaci nejsou nikde ukládány ani odposlouchávány. To zvyšuje bezpečnost.

Jak již bylo zmíněno, Stripe umožňuje přidání Apple Pay i Google Pay, nicméně tyto možnosti nebyly v této práci implementovány, jelikož pro obě tyto možnosti je třeba dalších účtů (Apple a Google) a nastavení, což by pro tuto práci zvýšilo složitost bez přidané hodnoty. Pro produkční a reálné nasazení je však lepší tyto možnosti implementovat, jelikož výrazně zlepší pohodlí uživatelů.

8.3.4 Onboarding

V případě že uživatel nemá v databázi aktuálně žádné vozidlo, je přesměrován na onboarding stránku. Jedná se o rozcestník, který uživatele přesměruje na požadovanou obrazovku. Na výběr je zde buď zaparkovat a přidat nové vozidlo (možnost Add car), nebo si rezervovat parkovací místo (možnost Reserve a parking

spot). Po úspěšném přidání nebo rezervování nebude místo k dispozici ostatním uživatelům, dokud nebude zase volné. Jsou zde zobrazeny informace o ceně parkovního a počet dostupných parkovacích míst.



Obrázek 31 - Onboarding obrazovka (zdroj: vlastní zpracování)

8.3.5 Rezervace

Obrazovka pro rezervaci je jednoduchá. Zobrazuje počet volných míst a poplatek za rezervaci. Po rezervaci místa se do databáze uloží čas, kdy byla rezervace vytvořena. Podle tohoto času se počítá a zobrazuje cena rezervace oproti aktuálnímu času. Uživatel zde má možnost buď pokračovat po zaparkování vozu a přidat ho, nebo zrušit svoji rezervaci a zaplatit aktuální cenu za rezervaci.

Při přidání vozidla se poplatek za rezervaci připočte k výsledné ceně za parkování a nabíjení.

Cena rezervace byla stanovena na 0.01 dolarů za minutu. Minimální platba pro poskytovatele Stripe je však 0.50 dolarů. Byla tedy stanovena minimální částka za rezervaci na 50 centů, což znamená rezervaci na padesát minut. Po padesáti minutách je k částce připočítán výše zmíněný poplatek 0.01 dolarů za minutu.

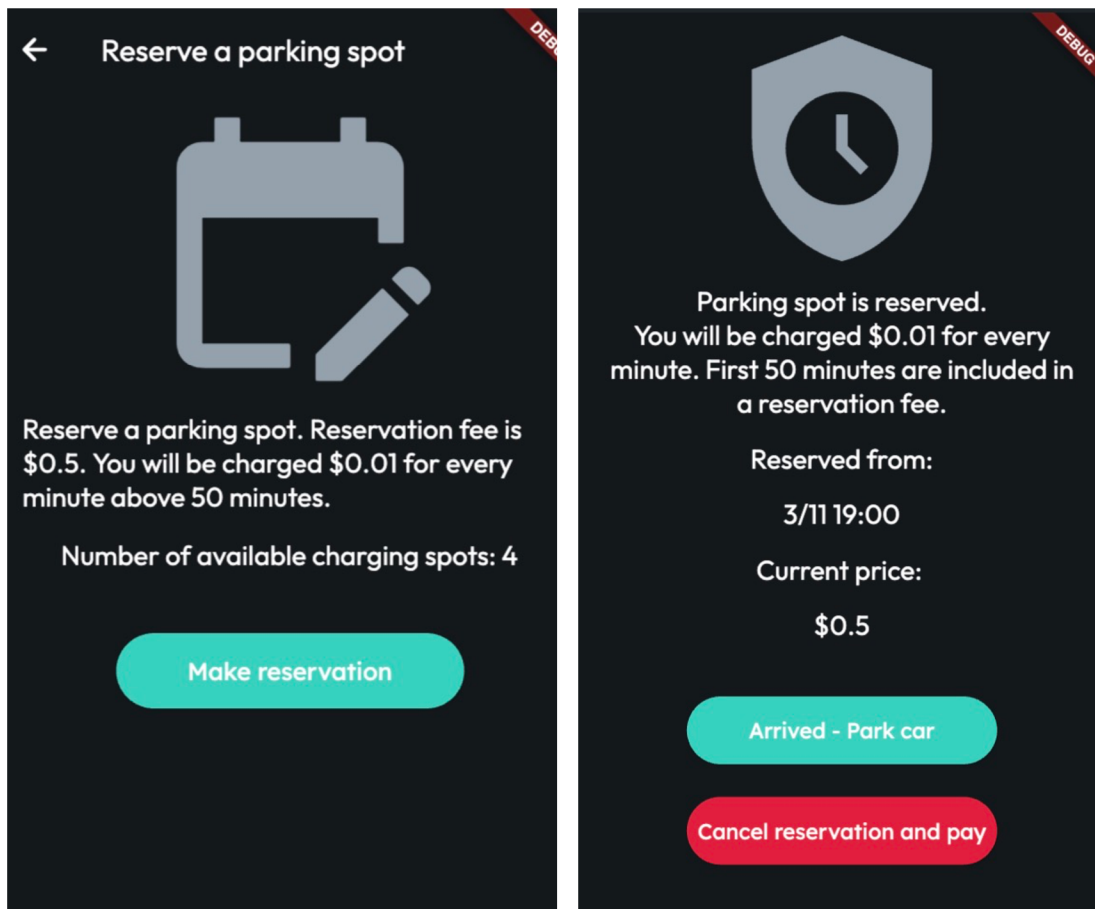
Pro výpočet ceny za rezervaci je použita následující funkce:

```
double price = (reservationTo.millisecondsSinceEpoch -  
    reservationFrom.millisecondsSinceEpoch) / 1000 / 60 * reservationPricePerMinute;  
double roundedPrice = double.parse((price).toStringAsFixed(2));  
if (roundedPrice < 0.5) return 0.50;  
return roundedPrice;
```

Nejdříve se odečte čas rezervace (reservationFrom) od aktuálního času (reservationTo), které jsou převedeny na počet milisekund od začátku unixové epochy (unixový čas) - neboli od 1.1.1970 00:00. Výsledný počet milisekund se vydělí tisícem, pro převod na vteřiny a poté vydělí šedesáti pro převod na minuty. Tento výsledek se vynásobí cenou za minutu - reservationPricePerMinute, neboli \$0.01. Nakonec se výsledek zaokrouhlí dolů na 2 desetinná čísla. Poté se porovná výsledná cena a pokud je menší než 0.5, je vrácena cena 0.5. V opačném případě je vrácena zaokrouhlená vypočítaná cena.

Tento vzorec může být optimalizován, ale byl ponechán takto pro lepší přehlednost.

Pro možnost měnit částku za minutu bylo navrženo, aby se cena získávala z databáze, nikoliv jako fixní hodnota. Tím se při změně ceníku nemusí upravovat kód aplikace, ale pouze hodnota v databázi. To lze několika způsoby - přímou změnou v databázi, API endpointem, nebo vytvořením admin účtu, který bude moci tyto hodnoty spravovat. Taková možnost ale v této práci implementována nebyla, jelikož hodnotu lze rychle a snadno změnit přímo v databázi přes FlutterFlow. Minutová cena za rezervaci se nachází v kolekci ParkingLot pod názvem reservationPricePerMinute.



Obrázek 32 - Obrazovky rezervací (zdroj: vlastní zpracování)

8.3.6 Přidání vozidla

Při zaparkování řidič ideálně připojí vozidlo k nabíjecí stanici, naskenuje QR kód na obrazovce či naskenuje NFC tag a vozidlo se mu propojí s aplikací. Tento přístup ale momentálně není úplně možný, jelikož mobilní stanice neumí z vozidla získat všechny potřebné informace. Více na toto téma viz kapitola **8.4 Problém získání informací**. Ve FlutterFlow lze jednoduše na pár kliknutí čtečku QR kódu přidat, nicméně v testovacím prostředí FlutterFlow není možné tuto funkcionalitu použít, jelikož QR čtečka je pouze pro iOS a Android, nikoliv pro webové testovací prostředí. Proto byla vytvořena obrazovka, kde uživatel sám vozidlo přidá a informace o něm vyplní manuálně.

Důležitými informacemi, které je třeba vyplnit jsou velikost baterie a průměrná spotřeba kWh. K tomu je potřeba vyplnit SPZ vozidla. Ta slouží pouze pro snazší identifikaci vozidla na hlavní stránce, kde je tato hodnota zobrazena.

Jako volitelná hodnota na této stránce se dá zadat plánovaný čas odjezdu a také je zde posuvník na jakou maximální hodnotu se baterie nabije (výchozí stav je 80 %). Podle těchto hodnot může centrální řídicí systém určovat priority nabíjení a lépe plánovat optimalizaci nabíjení.

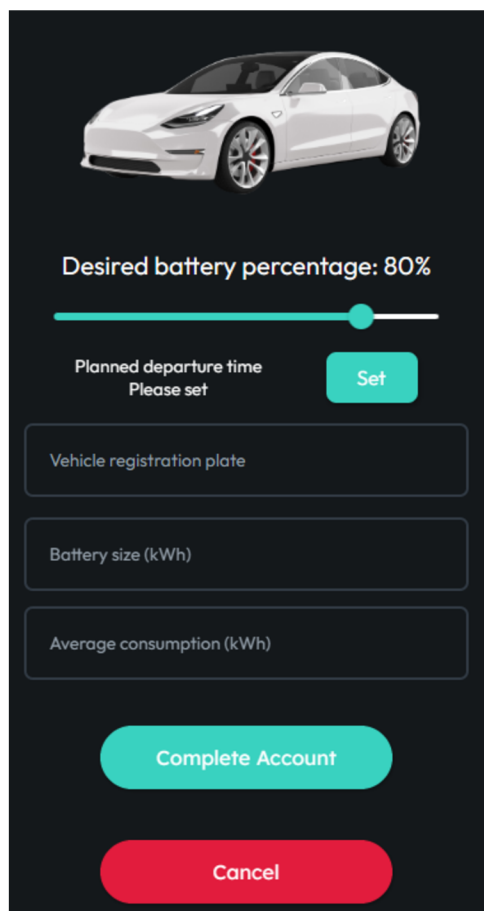
Aktuální procento nabití je možné získávat jako SoC (State of Charge) z nabíjecí stanice přes protokol OCPP. V reálném prostředí by centrální řídicí systém získával SoC a předával ho aplikaci přes API nebo databázi, ale pro potřeby testování a kvůli absenci centrálního systému pro tuto práci je tato hodnota generována náhodně.

Textové vstupy jsou ošetřeny, aby uživatel nemohl zadat špatné nebo nesmyslné hodnoty. Každé textové pole musí být vyplněno, Vehicle registration plate musí mít minimálně 5 znaků. Battery size a Average consumption jsou filtrovány podle následujícího regexu: `^\d+(\.\d+)?$`

Přesněji tento regex znamená libovolně dlouhé číslo, které může být následováno tečkou a libovolně dlouhým číslem, ale také nemusí.

Po odeslání formuláře jsou textová pole zkontrolována. Pokud není splněn minimální počet znaků, nebo text nesplňuje zmíněný regex, formulář není odeslán a zobrazí se chyba pod chybným textovým polem.

V případě, že jsou textová pole v pořádku, formulář je odeslán a v databázi je vytvořen nový dokument vozidla se zadanými hodnotami. V databázi se poté přidá reference na vozidlo do dokumentu parkovacího místa a zároveň je místo označeno jako obsazené pomocí pole isOccupied. Do dokumentu přihlášeného uživatele jsou uloženy reference na parkovací místo i vozidlo. Nakonec je uživatel přesměrován na hlavní stránku aplikace.



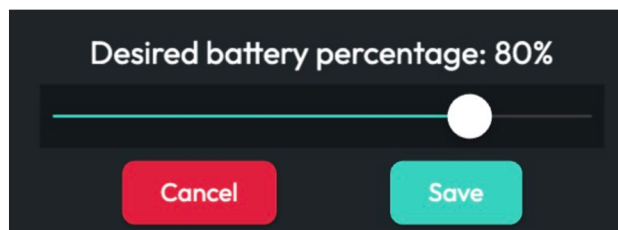
Obrázek 33 - Obrazovka přidání vozidla (zdroj: vlastní zpracování)

8.3.7 Možnosti nabíjení

Pro změnu nabíjecích preferencí a parametrů byla vytvořena obrazovka možností nabíjení. Na té lze nastavit různé parametry, které pomohou centrálnímu systému optimalizovat nabíjení. Nastavené hodnoty jsou hned po uložení zapsány do dokumentu vozidla nebo nabíjecího místa v databázi.

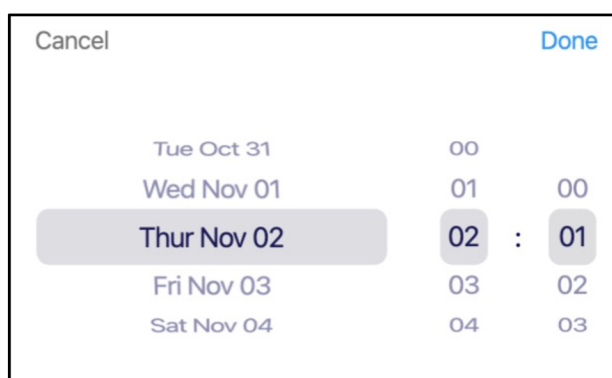
Nachází se tu:

- Změnu požadovaného procenta nabití – Uživatel tuto hodnotu nastaví při přidávání vozidla do aplikace a zde ji může opravit, pokud by se jeho preference změnily. Třeba v případě, že se změnily jeho plány a potřebuje větší dojezd. Zadaná hodnota se zapíše do databáze do pole `desiredChargeAtDeparture`.



Obrázek 34 – Možnost výběru limitu nabíjení (zdroj: vlastní zpracování)

- Změna času odjezdu – Tuto hodnotu také uživatel zadává při přidávání vozidla do aplikace a zde ji v případě potřeby také může upravit. Zadaná hodnota se zapíše do databáze do pole `desiredDepartureDateTime`.



Obrázek 35 – Možnost výběru času odjezdu (zdroj: vlastní zpracování)

- Přepínač V2G – Umožnění uchování přebytečné energie a jejímu zpětnému využití z baterie vozidla. V databázi tato možnost mění pole `isV2GActive` na `true` nebo `false`.
- Přepínač nabíjení – Možnost zaslat požadavek pro zastavení nabíjení svého vozidla. Tato možnost může být prospěšná, pokud si uživatel přeje už dále neplatit za nabíjení a tím ušetřit. V databázi tato možnost mění pole `isCharging` na `true` nebo `false`.
- Přepínač režimů nabíjení – Jedná se o režimy definované řídicí systémem. Po kliknutí se zobrazí nabídka všech módů jejich popisem. Vybraný režim je poté zobrazen v nabídce. Po vybrání režimu se v databázi změní `chargingMode` (int). V aplikaci jsou implementovány:
 - Běžný režim, při kterém se využívá jakákoliv dostupná energie (**Regular mode**)

- Nabíjení pouze z obnovitelných zdrojů (**Eco mode**)
- Nabíjení pouze pokud je dostupná přebytečná energie (**Surplus mode**)
- Nabíjení, pokud je aktuální cena elektřiny nízká a uživatel tím ušetří peníze za nabíjení (**Saving mode**). Například pro využití levnějšího nočního tarifu.

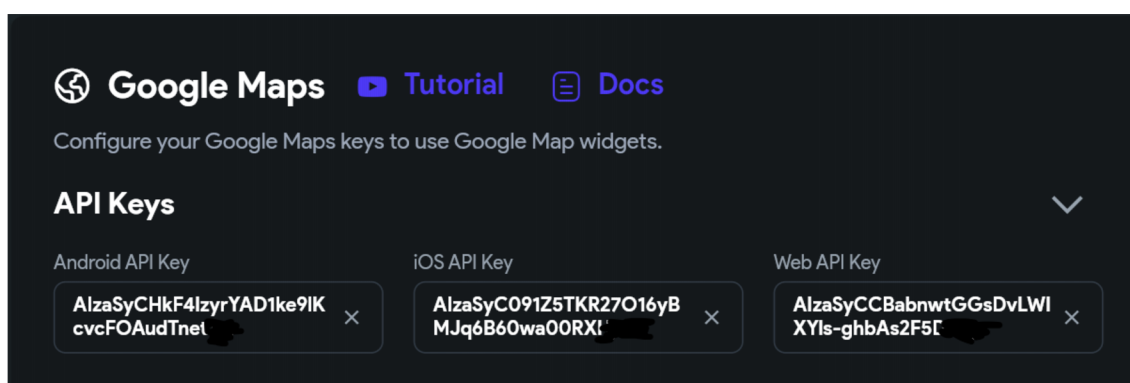


Obrázek 36 - Možnosti nabíjení (vlevo) a otevřená nabídka režimů (vpravo) (zdroj: vlastní zpracování)

8.3.8 Navigace k parkovišti

Pro možnost nalezení parkoviště a navigaci k němu byla vytvořena obrazovka zobrazující mapu od společnosti Google. V této mapě se zobrazuje parkoviště ve formě červené značky. Ta je umístěna na mapu ze souřadnic, které se nachází v databázové kolekci ParkingLot.

Pro funkční mapu je zapotřebí Google Maps API klíč. Ten lze vygenerovat v Google Cloud konzoli⁶. Pro potřeby projektu byly vygenerovány 3 klíče. Každý pro jednu platformu, tedy web, iOS a Android. V nastavení klíčů je možné je zabezpečit před neoprávněným použitím a omezit jejich funkcionalitu. Ta byla nastavena pro každý klíč pouze pro Maps API pro danou platformu. V konzoli lze poté API klíče zobrazit. Ty je potřeba vložit do Google Maps v nastavení FlutterFlow projektu. Tím je umožněno aplikaci zobrazit mapu pomocí Google Maps widgetu.

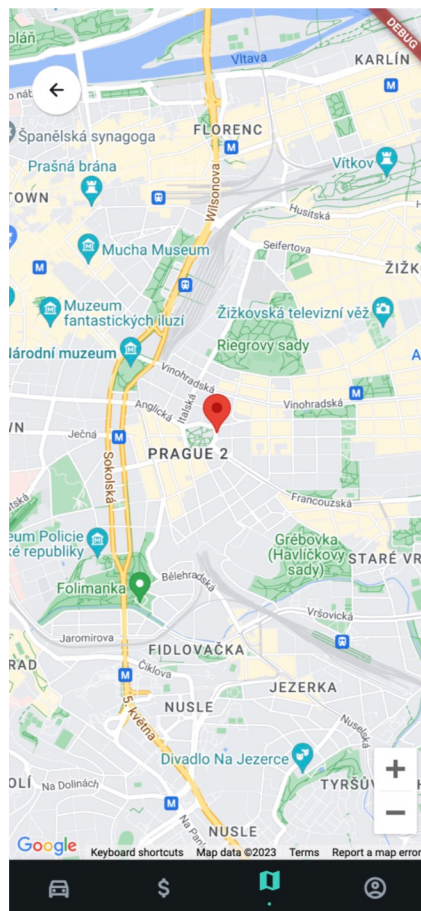


Obrázek 37 - API klíče pro Google Maps (zdroj: vlastní zpracování)

Mapa je interaktivní, ale přímou možnost navigace nenabízí. Navigace je přenechána externí aplikaci Google Maps, kterou lze otevřít při kliknutí na místo, kam si uživatel přeje navigovat. Zvýrazněné a předvybrané je parkoviště.

Mapu je možné zobrazit buď ze spodní navigační lišty jako přihlášený uživatel, nebo z přihlašovací obrazovky, pokud uživatel není přihlášen.

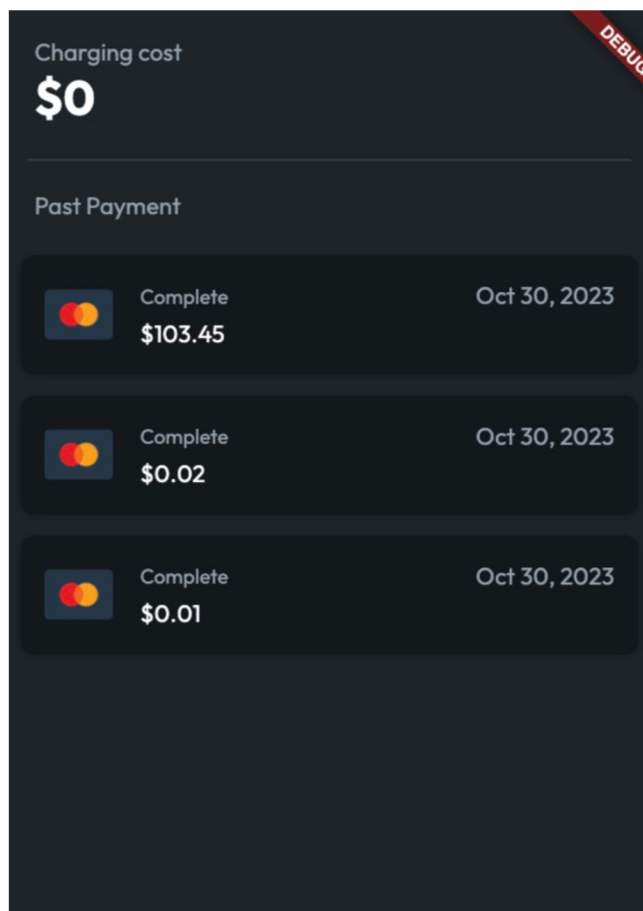
⁶ Dostupné z <https://console.cloud.google.com/apis/credentials>



Obrázek 38 - Mapa pro navigaci k parkovišti (zdroj: vlastní zpracování)

8.3.9 Historie plateb

Pro přehled útraty má uživatel možnost si pro svůj účet zobrazit historii všech plateb. Jedná se o jednoduchý seznam s cenami a daty uskutečnění plateb. Na seznam se přihlášený uživatel dostane snadno ze spodní navigační lišty. Jedná se o jednoduchý list všech plateb pro jeho uživatelské id.

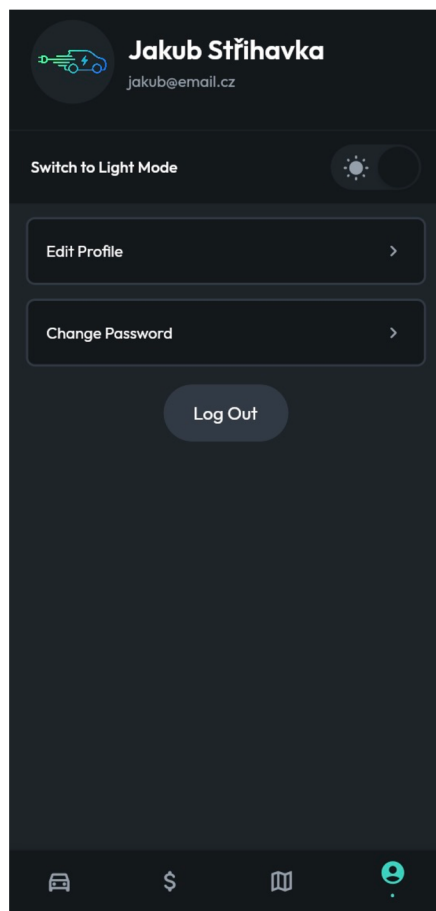


Obrázek 39 – Seznam plateb (zdroj: vlastní zpracování)

8.3.10 Profil

Poslední možností na spodní navigační liště je profil uživatele. Zde si uživatel může vybrat mezi světlým a tmavým režimem, změnit si zobrazované jméno, heslo k účtu, nebo se odhlásit.

Změna hesla probíhá přes Firebase. Po vyplnění emailu je zaslán uživateli email s odkazem, přes který si heslo změní.



Obrázek 40 - Obrazovka profilu uživatele (zdroj: vlastní zpracování)

8.3.11 Děkovací obrazovka s instrukcemi

Po úspěšné platbě je uživatel přesměrován na děkovací obrazovku s instrukcemi, jak opustit parkoviště. Na této obrazovce je započat pětiminutový časovač. Po skončení časovače, nebo po kliknutí na tlačítko “Mark as Complete” je parkovací místo resetováno do původního nastavení, čímž je označeno jako dostupné. Vozidlu je v databázi nastaven čas “parkedTo” a uživatel je přesměrován na onboarding obrazovku.



Obrázek 41 - Obrazovka instrukcí k vyparkování (zdroj: vlastní zpracování)

8.4 Problém získání informací

Hodnota kapacity baterie (v kWh) a průměrná spotřeba (v kWh/km) jsou v aplikaci aktuálně zadávány uživatelem, což není ideální. Uživatel může snadno udělat chybu a zadat špatnou informaci buď omylem, nebo neznalostí. Tím by centrální systém optimalizoval špatné údaje, což by mohlo vést ke špatné prioritizaci a neefektivitě. K optimalizaci bez těchto hodnot tedy systém může použít pouze aktuální stav nabití (SoC) což není ideální, jelikož kapacity baterií zaparkovaných vozidel se mohou velice lišit. Například Škoda Citigo^e iV má kapacitu baterie 36.8 kWh. [64] Oproti tomu GMC Hummer EV Pickup EV3x má kapacitu baterie 246 kWh. [65] Optimalizace těchto dvou vozidel pouze podle SoC bude velice nepřesná, jelikož nabití jednoho procenta u těchto dvou vozidel znamená mnohonásobně více potřebné energie u větší baterie.

Nejdeálnější by bylo získat tyto i další informace přímo z vozidla pomocí nabíjecí stanice, ke které je vozidlo připojeno. Nicméně po usilovném hledání nebyla nalezena žádná technologie ani protokol, který tato data momentálně umožní získat z vozidla a poskytnout je centrálnímu systému.

Druhou možností je (alespoň částečná) eliminace lidského chybového faktoru pomocí interaktivního vybrání vozidla a načtení potřebných dat z API. Uživatel by vybral značku, model vozidla a případně i další potřebné informace, aplikace by poslala API požadavek a ten by vrátil všechna potřebná data. Jedním z těchto API je například ev-database (web: <https://ev-database.org/data-services-api>), bohužel toto i další podobné API jsou placené, a navíc je potřeba pro ně zažádat individuální přístup po dohodě přes email. K tomu ev-database je k dispozici pouze pro registrované společnosti, ne pro soukromé použití, takže pro tuto práci nebyly API databáze použity. O přístup do API pro studentské účely bylo během psaní této práce zažádáno u chargeprice.net, ale žádná odpověď nebyla do konce psaní této práce obdržena.

8.5 Komunikace s databází – Cloud Firestore

Flutter i FlutterFlow nabízí široké možnosti pro komunikaci s databází. Ať se už jedná o REST API, Firebase Realtime Database, nebo Cloud Firestore. Jak již bylo zmíněno, v aplikaci byla použita Cloud Firestore databáze.

Cloud Firestore je databáze, kde jsou data organizována do kolekcí a dokumentů. Kolekci si lze představit jako třídu a dokument jako instanci této třídy. Dokument má pole, která se mapují na hodnoty. [66]

Pro komunikaci s touto databází využívá Flutter balíček s názvem `cloud_firestore`, který je dostupný na `pub.dev`. Tento balíček poskytuje třídu `FirebaseFirestore`, která reprezentuje instanci databáze. Pomocí této třídy lze získat odkaz na kolekci nebo dokument pomocí metody `collection()` nebo `doc()`. Cloud Firestore databáze podporuje dva způsoby čtení dat: jednorázové čtení nebo průběžné naslouchání změnám. Jednorázové čtení znamená, že je požádáno o data jednou a vrátí se odpověď ve formě objektu `QuerySnapshot` nebo `DocumentSnapshot`. Průběžné naslouchání znamená, že se widget přihlásí k odběru streamu dat, který bude posílat

aktualizace těchto dat ve formě objektu QuerySnapshot nebo DocumentSnapshot pokaždé, když dojde k nějaké změně v databázi.

Pro jednorázové čtení dat se používá metoda `get()` na odkazu na kolekci nebo dokument. Pro průběžné naslouchání změnám dat se využívá metoda `snapshots()`.

Pro zápis dat do databáze je možné použít metodu `set()`, `update()` nebo `delete()` na odkazu na dokument. [67]

V aplikaci se jednorázové čtení dat používá například při přihlašování uživatele do aplikace. Průběžné naslouchání se používá téměř na všech widgetech v aplikaci, aby byla zaručena aktuálnost dat. To se využívá například na domovské obrazovce pro widget zobrazující stav baterie. V kódu je použit stream `CarRecord.getDocument(currentUserDocument!.car!)` pro načítání dokumentu zaparkovaného vozidla a průběžného získání jeho stavu baterie. Jakákoliv změna v dokumentu vozidlo se projeví ihned. Ukázka je pro zjednodušení zkrácena a trochu upravena, jelikož Flutter kód často zabírá mnoho řádků.

```
Widget build(BuildContext context) {
  return Padding(
    padding: EdgeInsetsDirectional.fromSTEB(16.0, 0.0, 16.0, 0.0),
    child: AuthUserStreamWidget(
      builder: (context) => StreamBuilder<CarRecord>({
        stream: CarRecord.getDocument(currentUserDocument!.car!),
        builder: (context, snapshot) {
          final progressBarCarRecord = snapshot.data!;
          return LinearPercentIndicator(
            percent: progressBarCarRecord.batteryChargePercentage / 100,
            center: Text(
              valueOrDefault<String>({
                formatNumber(
                  progressBarCarRecord.batteryChargePercentage
                ),
                '80%',
              },
            ),
          ),
        ),
      ),
    ),
  ),
),
```

9 Testování aplikace

Testování aplikace probíhalo v průběhu samotného vývoje aplikace autorem. Následně při funkční a zdánlivě hotové verzi bylo uskutečněno testování v úzkém kruhu rodiny a přátel ve webové aplikaci, kde k ní každý měl přístup.⁷ Při prvním zapnutí a sledování přes rameno byly ihned zjištěny nedostatky, kdy se například uživatelé snažili zadat nesprávné informace, nebo některé části aplikace nefungovaly správně. Například uživatelé zadávali procento nabití baterie větší než 100, kapacitu baterie abnormálně velkou (v řádech statisíců), nebo extrémně malou (například 1), dále některá políčka vůbec nevyplnili, nebo dokonce vyplnili text do políčka očekávající pouze číslo. Po nesprávném vyplnění těchto dat aplikace přestala fungovat, jelikož tato data neočekávala. Byly proto přidány kontroly vstupních dat, aby uživatelé mohli zadat pouze čísla, případně čísla s tečkou a minimální počet znaků. Před změnou, která vyplynula z testování, bylo aktuální procento nabití zadáno uživatelem ručně při přidávání vozidla kvůli absenci centrálního systému, který by za běžného provozu zapisoval do databáze aktuální procento nabití vozidla. Nyní pro lepší ochranu před zadáním špatných hodnot se při přidávání vozidla do databáze automaticky vygeneruje náhodné číslo pro procento nabití baterie, které simuluje vyplnění této hodnoty centrálním systémem.

Dalším objeveným problémem bylo načítání formuláře pro zadání kreditní karty. Tento formulář má pár sekundovou prodlevu, než se otevře, takže nedočkaví uživatelé klikali na tlačítko zaplatit vícekrát, což po načtení zobrazilo několik těchto formulářů, které se poté překrývaly a nedaly správně používat ani zavřít.

Dalším problémem bylo, že někteří uživatelé otevřeli adresu na svém počítači, s čímž nebylo počítáno. Aplikace nebyla na počítačové obrazovky optimalizována, takže se veškeré obrázky nepoměrně roztáhly a znetvořily (výška byla fixní, ale šířka se zvětšovala podle velikosti prohlížeče). Také menu nabíjecích možností je pro mobilní zařízení zobrazené jako mříž maximálně dvou tlačítek vedle sebe, což na

⁷ Aplikace je dostupná na webové adrese <https://parking-app-gtnu18.flutterflow.app/>

počítači způsobilo, že jedno tlačítko zabíralo půlku celého monitoru. Mříž tedy byla pro počítače upravena pro 5 položek vedle sebe.

Ze zpětné vazby z tohoto testování vyplynulo několik doporučení pro následující verze. Převážně vstup dat do aplikace o vozidlu by neměl být zadáván manuálně uživatelem, ale měl by být získán přímo z vozidla přes nabíjecí stanici, nebo alespoň přes API, kde uživatel vybere své vozidlo, což poskytne potřebná data. Viz předešlá kapitola **8.4 Problém získání informací**.

10 Shrnutí výsledků

V průběhu práce bylo získáno velké množství informací o vertikálních parkovištích a možnostech optimalizace nabíjení. Na těchto základech byla navržena a vyvinuta mobilní aplikace pro smart vertikální parkoviště pro elektromobily. Aplikace je navržena, aby komunikovala s centrálním řídicím systémem, který bude mít na starost komunikaci s nabíjecími stanicemi a optimalizaci nabíjení podle různých kritérií, jako je limit nabíjení, do kterého chce uživatel vozidlo nabíjet, plánovaný čas odjezdu, přepínání nabíjecích režimů a zapnutí V2G (Vehicle to Grid) pro možnost ukládání energie v baterii vozidla, která může být poté použita pro posílení sítě v případě potřeby. Tím se může dosáhnout vyšší efektivity využití energie a snížení zátěže na síť. V aplikaci je možné rezervovat parkovací místo, přidat vozidlo po zaparkování do věže, nastavovat možnosti nabíjení, vidět aktuální stav vozidla a platit za služby. Sjednocením služeb parkoviště na jedno místo může zvýšit pohodlí a spokojenost uživatelů.

Navrhnuté řešení má potenciál přinést řadu výhod jak pro uživatele, tak pro provozovatele a distributory energie. Uživatelé mohou ušetřit čas a peníze za parkování a nabíjení a zvýšit životnost baterie. Provozovatelé a distributoři energie mohou zlepšit stabilitu a kvalitu sítě, snížit potřebu investic do infrastruktury a lépe využít obnovitelných zdrojů energie.

11 Závěry a doporučení

V této práci byla popsána problematika smart vertikálního parkoviště pro elektromobilitu, které umožňuje efektivní využití prostoru a optimalizaci nabíjení. Byla navržena a vyvinuta mobilní aplikace pro smart vertikální parkoviště pro elektromobily, která slouží jako uživatelské rozhraní pro komunikaci s centrálním řídicím systémem parkoviště. Aplikace umožňuje uživatelům aktivně ovlivnit podíl na optimalizaci sítě, rezervovat parkovací místo, mít přehled o stavu zaparkovaného vozidla a platit za služby. Aplikace je navržena, aby zasílala požadavky centrálnímu řídicímu systému, který přímo komunikuje s nabíjecími stanicemi a dává jim pokyny podle zadaných pravidel.

Výsledky práce ukazují, že smart vertikální parkoviště pro elektromobilitu má potenciál přinést pro uživatele, zaparkovaná vozidla i poskytovatele řadu výhod.

V této práci byly odhaleny některé problémy a výzvy, které je třeba řešit pro úspěšné nasazení smart vertikálního parkoviště. Hlavním problémem podobné implementace je v současné době problém získání potřebných dat, potřebných pro optimalizaci, z vozidla pomocí nabíjecí stanice a předání těchto dat centrálnímu řídicímu systému. Pro efektivní optimalizaci a správné plánování je potřeba znát alespoň maximální kapacitu baterie. Bez tohoto a dalších potřebných dat systém nemůže efektivně optimalizovat a prioritizovat vozidla, která nabíjení potřebují nejvíce a může dojít k situacím, kdy vozidlo nebude nabito na požadovanou hodnotu při odjezdu, což může nemile překvapit řidiče vozidla, který počítal s více nabitou baterií. K optimalizaci bez těchto hodnot tedy systém může použít pouze aktuální stav nabití (SoC). Ten není ideální pro optimalizaci kvůli různým kapacitám baterií. Bylo proto implementováno ruční zadávání těchto hodnot uživatelem. To by mohlo být zlepšeno výběrem vozidla z databázové nabídky všech elektrických vozidel přes API. I přes to tento přístup není ideální.

Je proto doporučeno, aby se v budoucím výzkumu a vývoji zaměřilo na zlepšení komunikace mezi nabíjecími stanicemi, vozidly a centrálním systémem.

Další možnou cestou optimalizace je využití umělé inteligence a strojového učení pro predikci chování a potřeby uživatelů na základě historických dat o jejich

předchozích nabíjení a nastavení. Z historických dat by mohly být učiněny odhady o velikosti baterie a dalších potřebných datech. Aplikace by se tím postupně stávala přesnější a lepší. V práci také nebyla do podrobnosti rozebrána tematika rozdílu AC a DC nabíjení. Oba typy nabíjení s sebou nesou celou řadu omezení.

Aplikace je přístupná jako webová aplikace, a i jako aplikace pro iOS a Android. Z důvodu nutných poplatků za publikaci aplikace a potřebě jejího schválení však nebyla aplikace publikována na Apple App Store ani na Obchod Play. Její spustitelný kód lze najít na GitHub. Z podobných důvodů nebyly také implementovány platební metody Apple Pay a Google Pay, které by pro uživatele byly pohodlnější.

12 Seznam použité literatury

- [1] PATEL, Nil, Akash Kumar BHOI, Sanjeevikumar PADMANABAN a Jens Bo HOLM-NIELSEN, ed. *Electric Vehicles: Modern Technologies and Trends* [online]. Singapore: Springer Singapore, 2021 [vid. 2023-09-10]. Green Energy and Technology. ISBN 9789811592508. Dostupné z: doi:10.1007/978-981-15-9251-5
- [2] WWW.FG.CZ, 2023, FG Forrest, a s. Dojezd a nabíjení elektromobilu: Dobíjet můžete i doma přes noc | Skupina ČEZ. *Skupina ČEZ - Produktová sekce* [online]. [vid. 2023-08-12]. Dostupné z: <http://www.cez.cz/cs/clanky/elektromobilita/dojezd-a-nabijeni-elektromobilu-dobijet-muzete-i-doma-pres-noc-174022>
- [3] Jak daleko dojedou elektromobily na jedno nabití? Žebříček více než 150 modelů podle deklarovaného dojezdu. Tvrdá data. *AutoŽivě.cz* [online]. 10. březen 2023 [vid. 2023-09-10]. Dostupné z: <https://www.autozive.cz/jak-daleko-dojedou-elektromobily/>
- [4] *How long does it take to charge an electric car? | EVBox* [online]. [vid. 2023-09-10]. Dostupné z: <https://blog.evbox.com/ev-charging-times>
- [5] *Elektromobily: Jak degradují baterie?* [online]. [vid. 2023-08-12]. Dostupné z: <https://www.uamk.cz/informace/autostyl/stripsy/204-elektromobily-jak-degraduji-baterie>
- [6] *Tesla Model 3 charging time, range, and cost | EVBox* [online]. [vid. 2023-08-11]. Dostupné z: <https://evbox.com/en/electric-cars/tesla/tesla-model-3>
- [7] Reasons Your Tesla Is Charging Slow. *PimpMyEV* [online]. 3. říjen 2022 [vid. 2023-08-29]. Dostupné z: <https://pimpmyev.com/blogs/speed-style-carbon-fiber/reasons-your-tesla-is-charging-slow>
- [8] Přehled wallboxů a nabíjecích stanic. *Memodo* [online]. [vid. 2023-08-29]. Dostupné z: <https://www.memodo.cz/prehled-wallboxu-a-nabijecich-stanic>
- [9] *Wallbox vs. Household Plug Sockets* [online]. [vid. 2021-02-04]. Dostupné z: <https://www.easyway.site/index.php?l=en&u=wallbox-vs-schuko-steckdose>
- [10] *Instalace pro domácí nabíjení* [online]. 9. červen 2016 [vid. 2021-02-04]. Dostupné z: https://www.tesla.com/cs_CZ/support/home-charging-installation
- [11] HORČÍK, Jan. *Volkswagen uvádí domácí nabíječku pro elektromobily ID* [online]. 15. červen 2020 [vid. 2023-11-01]. Dostupné z: <https://www.hybrid.cz/volkswagen-uvadi-domaci-nabijecku-pro-elektromobily-id/>

- [12] S.R.O, Ušetřeno cz. 7 problémů, které budou muset elektromobily vyřešit [online]. [vid. 2021-02-08]. Dostupné z: <https://www.elektrina.cz/problemy-elektromobilu>
- [13] S.R.O, 24net. E.ON spustil v Česku nejvýkonnější nabíječku pro elektromobily. Má 400 kW. *fDrive.cz* [online]. 27. září 2023 [vid. 2023-11-05]. Dostupné z: <https://fdrive.cz/clanky/eon-spustil-v-cesku-nejvykonnejsi-nabijecku-pro-elektromobily-ma-400-kw-11298>
- [14] *Počet nabíjecích stanic pro elektromobily loni v Česku stoupl o třetinu - Novinky* [online]. [vid. 2023-08-11]. Dostupné z: <https://www.novinky.cz/clanek/auto-pocet-nabijecich-panic-pro-elektromobily-loni-v-cesku-stoupl-o-tretinu-40426920>
- [15] *Introducing V3 Supercharging* [online]. 6. března 2019 [vid. 2021-02-08]. Dostupné z: https://www.tesla.com/cs_CZ/blog/introducing-v3-supercharging
- [16] SHAHAN, Cynthia. Tesla Model 3 V3 Supercharging Times: 2% To 100% State of Charge (Video). *CleanTechnica* [online]. 18. listopad 2019 [vid. 2021-02-08]. Dostupné z: <https://cleantechnica.com/2019/11/18/tesla-model-3-v3-supercharging-times-2-to-100-state-of-charge-video/>
- [17] HOFFMAN, Connor. Tesla's 250-kW Supercharger Only Saved Us 2 Minutes vs. a 150-kW Charger. *Car and Driver* [online]. 15. duben 2020 [vid. 2021-02-08]. Dostupné z: <https://www.caranddriver.com/news/a32132062/tesla-250-kw-vs-150-kw-supercharger-tested/>
- [18] *NIO Power / NIO* [online]. [vid. 2023-08-11]. Dostupné z: <https://www.nio.com/nio-power>
- [19] EV battery swapping by Nio and Geely – A silly gimmick or an elegant solution? | WapCar. *WapCar News* [online]. 22. květen 2021 [vid. 2023-08-30]. Dostupné z: <https://www.wapcar.my/news/ev-battery-swapping-by-nio-and-geely---a-silly-gimmick-or-an-elegant-solution-27717>
- [20] EV charging without the wait: Battery swap stations open in Europe. *euronews* [online]. 10. duben 2023 [vid. 2023-08-11]. Dostupné z: <https://www.euronews.com/next/2023/04/09/tired-of-waiting-ev-charging-stations-first-nio-battery-swap-stations-open-in-europe>
- [21] CHUDY, Aleksander. BATTERY SWAPPING STATIONS FOR ELECTRIC VEHICLES. *Informatyka, Automatyka, Pomiar w Gospodarce i Ochronie Środowiska* [online]. 2021, **11**(2), 36–39. ISSN 2391-6761, 2083-0157. Dostupné z: doi:10.35784/iapgos.2654
- [22] *Začínáme s elektromobilitou II – Jak na veřejné nabíjení? - Garáž.cz* [online]. [vid. 2021-02-08]. Dostupné z: <https://www.garaz.cz/clanek/zaciname-s-elektromobilitou-ii-jak-na-verejne-nabijeni-21005189>

- [23] FINNERTY, Joe. *GRIDSERVE | What is an electric car charging curve?* [online]. 17. únor 2023 [vid. 2023-11-05]. Dostupné z: <https://www.gridserve.com/2023/02/17/what-is-an-electric-car-charging-curve/>
- [24] *Understanding Charge-Discharge Curves of Li-ion Cells • EVreporter* [online]. 20. leden 2022 [vid. 2023-11-05]. Dostupné z: <https://evreporter.com/understanding-charge-discharge-curves-of-li-ion-cells/>
- [25] Figure 2. Constant voltage/Constant current charging waveform [40]... *ResearchGate* [online]. [vid. 2023-11-05]. Dostupné z: https://www.researchgate.net/figure/Constant-voltage-Constant-current-charging-waveform-40-Courtesy-of-CADEX_fig2_367342612
- [26] Zaptec Portal. *Zaptec* [online]. [vid. 2023-08-27]. Dostupné z: <https://zaptec.com/en/zaptec-portal/>
- [27] WWW.FG.CZ, 2023, FG Forrest, a s. Změna typu měření. *ČEZ Distribuce* [online]. [vid. 2023-08-27]. Dostupné z: <http://www.cezdistribuce.cz/cs/pro-zakazniky/potrebuji-vyresit/elektromery-a-odecty/zmena-typu-mereni>
- [28] Řízení čtvrt hodinového maxima. *TZB-info* [online]. [vid. 2023-08-27]. Dostupné z: <https://elektro.tzb-info.cz/inteligentni-budovy/13513-rizeni-ctvrthodinoveho-maxima>
- [29] Nabíjení elektromobilů, zatížení sítě a řízení výkonu - část II., řízení výkonu. *TZB-info* [online]. [vid. 2023-08-27]. Dostupné z: <https://energetika.tzb-info.cz/elektroenergetika/21052-nabijeni-elektromobilu-zatizeni-site-a-rizeni-vykonu-cast-ii-rizeni-vykonu>
- [30] Desigo system. *Siemens Česká republika* [online]. [vid. 2023-08-27]. Dostupné z: <https://www.siemens.com/cz/cs/products/technologie-budov/automation/desigo.html>
- [31] *OCPP 2.0.1, Protocols, Home - Open Charge Alliance* [online]. [vid. 2023-09-02]. Dostupné z: <https://www.openchargealliance.org/protocols/ocpp-201/>
- [32] *File:Hradec Králové, Resslerova, parkovací dům.jpg - Wikipedia* [online]. [vid. 2022-11-26]. Dostupné z: https://commons.wikimedia.org/wiki/File:Hradec_Kr%C3%A1lov%C3%A9_Resslerova_parkovac%C3%AD_d%C5%AFm.jpg
- [33] INFO@AION.CZ, AION CS-. 266/2021 Sb. Vyhláška, kterou se mění vyhláška č. 268/2009 Sb., o technických požadavcích na stavby, ve znění poz... *Zákony pro lidi* [online]. [vid. 2023-11-01]. Dostupné z: <https://www.zakonyprolidi.cz/cs/2021-266>

- [34] Země s největším počtem nabíječek začíná mít problém, který čeká i Česko. Řešení bude velmi drahé. *Autosalon.tv* [online]. [vid. 2023-10-25]. Dostupné z: <https://www.autosalon.tv/novinky/nova-auta/zeme-s-nejvetsim-poctem-nabijecek-zacina-resit-problem-ktery-ceka-i-nas-sit-elektromobily-nezvlada>
- [35] *Smart Parking CZ* [online]. [vid. 2022-11-26]. Dostupné z: <http://www.smart-parking.cz/>
- [36] Reference and map. *BIKETOWER - EN* [online]. [vid. 2021-02-04]. Dostupné z: <https://www.biketower.cz/en/reference-and-map/>
- [37] *Vertikální parkování: Vymoženost, která rozhodně není novinkou - SKODAKLASIK.cz* [online]. [vid. 2023-09-07]. Dostupné z: <https://www.skodaklasik.cz/vertikalni-parkovani-vymozenost-ktera-rozhodne-neni-novinkou/>
- [38] SIMON, Kazi. *Vertical Car Parking System* [online]. 2019. Dostupné z: https://www.researchgate.net/publication/336287016_Vertical_Car_Parking_System
- [39] *FOTO - Taranis Invest* [online]. [vid. 2023-09-08]. Dostupné z: <https://www.taranisinvest.com/foto.html>
- [40] *Automatické parkovací systémy (APS) | Stavební právo, kontrolní listy, normy, příklady a doporučení pro oblast stavebnictví.* [online]. [vid. 2023-09-09]. Dostupné z: https://www.stavebniklub.cz/33/automaticke-parkovaci-systemy-aps-uniqueidmRRWSbk196FNf8-jVUh4EvlcoRunmztNko3gSVnCbO9PpA5B5rrwHw/?uri_view_type=44&uid=1iY1Ex8xfv9gV8xprwwchWA&e=1G8XxY4m_b1nk7TpQnJrY5A
- [41] *Tower parking -smart parking and automatic parking -maoyuan* [online]. 2016 [vid. 2023-09-08]. Dostupné z: <https://www.youtube.com/watch?v=ZJaDq6BYFys>
- [42] Dynamic Load Management. *eDRV* [online]. [vid. 2023-10-23]. Dostupné z: <https://docs.edrv.io/docs/dynamic-load-management-introduction>
- [43] VANITHA, V., R. RESMI a Karri Naga Sai Vineela REDDY. Machine learning-based charge scheduling of electric vehicles with minimum waiting time. *Computational Intelligence* [online]. 2021, **37**(3), 1047–1055. ISSN 1467-8640. Dostupné z: doi:10.1111/coin.12333
- [44] *Everything You Need To Know About V1G, V2H, V2B, V2G, And V2X* [online]. [vid. 2023-06-12]. Dostupné z: <https://www.hivepower.tech//blog/everything-you-need-to-know-about-v1g-v2h-v2b-v2g-and-v2x>
- [45] MOJUMDER, Md Rayid Hasan, Fahmida AHMED ANTARA, Md HASANUZZAMAN, Basem ALAMRI a Mohammad ALSHAREF. Electric Vehicle-

to-Grid (V2G) Technologies: Impact on the Power Grid and Battery. *Sustainability* [online]. 2022, **14**(21), 13856. ISSN 2071-1050. Dostupné z: doi:10.3390/su142113856

- [46] *MET Group selects Tesla Megapack for latest battery energy storage project* [online]. [vid. 2023-09-06]. Dostupné z: <https://group.met.com/en/media/press-releases/met-group-selects-tesla-megapack-for-latest-battery-energy-storage-project>
- [47] Order Megapack. *Tesla* [online]. [vid. 2023-09-06]. Dostupné z: <https://www.tesla.com/megapack/design>
- [48] *The OCPP Handbook (2023) - AMPECO* [online]. 8. března 2023 [vid. 2023-11-05]. Dostupné z: <https://www.ampeco.com/guides/complete-ocpp-guide/>
- [49] *Multi-Platform* [online]. [vid. 2023-09-17]. Dostupné z: [//flutter.dev/multi-platform/](https://flutter.dev/multi-platform/)
- [50] *Flutter - Build apps for any screen* [online]. [vid. 2023-09-17]. Dostupné z: [//flutter.dev/](https://flutter.dev/)
- [51] What is Flutter? - Flutter App Explained - AWS. *Amazon Web Services, Inc.* [online]. [vid. 2023-09-18]. Dostupné z: <https://aws.amazon.com/what-is/flutter/>
- [52] *Introduction* [online]. [vid. 2023-09-18]. Dostupné z: <https://docs.flutterflow.io/>
- [53] *Firebase* [online]. [vid. 2023-09-18]. Dostupné z: <https://docs.flutterflow.io/data-and-backend/firebase>
- [54] *API Calls* [online]. [vid. 2023-09-18]. Dostupné z: <https://docs.flutterflow.io/data-and-backend/api-calls>
- [55] *FlutterFlow - Terms of Service* [online]. [vid. 2023-10-14]. Dostupné z: <https://flutterflow.io/tos>
- [56] *Actions* [online]. [vid. 2023-11-03]. Dostupné z: <https://docs.flutterflow.io/actions/actions>
- [57] Firebase Realtime Database. *Firebase* [online]. [vid. 2023-09-20]. Dostupné z: <https://firebase.google.com/docs/database>
- [58] Firebase Authentication. *Firebase* [online]. [vid. 2023-09-20]. Dostupné z: <https://firebase.google.com/docs/auth>
- [59] Firestore. *Firebase* [online]. [vid. 2023-09-20]. Dostupné z: <https://firebase.google.com/docs/firestore>

- [60] Cloud Functions for Firebase. *Firebase* [online]. [vid. 2023-09-20]. Dostupné z: <https://firebase.google.com/docs/functions>
- [61] Firebase Cloud Messaging. *Firebase* [online]. [vid. 2023-09-20]. Dostupné z: <https://firebase.google.com/docs/cloud-messaging>
- [62] What is Google Firebase? *Mobile Computing* [online]. [vid. 2023-09-20]. Dostupné z: <https://www.techtarget.com/searchmobilecomputing/definition/Google-Firebase>
- [63] *Shutterstock Standard Image License vs. Enhanced Image License - 24/7 Shutterstock Customer Support & Help* [online]. [vid. 2023-10-22]. Dostupné z: https://support.shutterstock.com/s/article/Standard-License-vs-Enhanced-License?language=en_US
- [64] *TD-CITIG0e-iV-cz.pdf* [online]. [vid. 2023-10-29]. Dostupné z: <https://cdn.skoda-storyboard.com/2019/09/TD-CITIG0e-iV-cz.pdf>
- [65] *GMC Hummer EV Pickup EV3x - tech specs and prices | myEVreview* [online]. [vid. 2023-10-29]. Dostupné z: <https://www.myevreview.com/tech-specs/gmc/hummer-ev-pickup/ev3x>
- [66] Cloud Firestore Data model. *Firebase* [online]. [vid. 2023-11-02]. Dostupné z: <https://firebase.google.com/docs/firestore/data-model>
- [67] *Cloud Firestore | FlutterFire* [online]. [vid. 2023-11-02]. Dostupné z: <https://firebase.flutter.dev/docs/firestore/usage>

13 Přílohy

13.1 Přiložený zip soubor

V přiloženém souboru se nachází složka se zdrojovým kódem vytvořené aplikace. K tomu je přiložen textový soubor s odkazy. Jeden odkaz je na online GitHub repositář, kde se také nachází zdrojový kód aplikace. Druhý odkaz je nasazená webová aplikace.

13.2 Oskenované zadání diplomové práce



Zadání diplomové práce

Autor: Bc. Jakub Střihavka

Studium: I1900287

Studijní program: N1802 Aplikovaná informatika

Studijní obor: Aplikovaná informatika

Název diplomové práce: **Smart vertikální parkování pro elektromobilitu**

Název diplomové práce AJ: Smart vertical parking for electromobility

Cíl, metody, literatura, předpoklady:

Cíl: Analýza možností IT podpory smart parkování a dobíjení pro elektrické dopravní prostředky.

Osnova:

Analýza stávajícího stavu

Parkování a dobíjení ve smart city

Návrh řešení - mobilní aplikace a technologické možnosti

Zhodnocení výsledků, závěr

SØRENSEN, Åse Lekang, et al. Smart ev charging systems for zero emission neighbourhoods. 2018.

LONGO, Michela; FOIADELLI, Federica; YAĪCI, Wahiba. Simulation and optimisation study of the integration of distributed generation and electric vehicles in smart residential district. *International Journal of Energy and Environmental Engineering*, 2019, 10.3: 271-285.

WAN, Yanni, et al. Game theoretic-based distributed charging strategy for PEVs in a smart charging station. *IEEE Transactions on Smart Grid*, 2020, 12.1: 538-547.

Zadávací pracoviště: Katedra informačních technologií,
Fakulta informatiky a managementu

Vedoucí práce: Ing. Karel Mls, Ph.D.

Datum zadání závěrečné práce: 15.10.2021