

Sem vložte zadání Vaší práce.

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

Mobilní aplikace pro řidiče přepravních služeb

Lukáš Kořán

Vedoucí práce: Ing. Jiří Hunka

11. května 2015

Poděkování

Tímto bych chtěl poděkovat vedoucímu práce Ing. Jiřímu Hunkovi za poskytnuté náměty a cenné rady. Zároveň bych chtěl poděkovat své rodině a přátelům za podporu během studia a vytváření bakalářské práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. Dále prohlašuji, že jsem s Českým vysokým učením technickým v Praze uzavřel dohodu, na základě níž se ČVUT vzdalo práva na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona. Tato skutečnost nemá vliv na ust. § 47b zákona č. 111/1998 Sb., o vysokých školách, ve znění pozdějších předpisů.

V Praze dne 11. května 2015

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2015 Lukáš Kořán. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Kořán, Lukáš. *Mobilní aplikace pro řidiče přepravních služeb*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstrakt

Tato bakalářská práce se zabývá analýzou, návrhem a implementací prototypu mobilní aplikace pro řidiče přepravních služeb. První část práce popisuje analytické procesy, na jejichž základě byl proveden návrh aplikace, který je popsán v následující části. Poslední část shrnuje vývoj prototypu aplikace, testování, zhodnocení přínosů a plány budoucího vývoje.

Klíčová slova řidič, potvrzování zásilek, správa jízd, dodávky zboží, mobilní aplikace, Java, Spring, Android

Abstract

This bachelor thesis deals with analysis, design and implementation of a mobile application for delivery drivers in form of a prototype. First part of the thesis describes analytical processes that lead to the application design, which is described in the next part. The last part of the thesis summarizes development of the application prototype, testing, evaluation of its benefits and plans for further development.

Keywords driver, delivery confirmation, ride management, mobile application, Java, Spring, Android

Obsah

Úvod	1
1 Analýza	3
1.1 Analýza existujících řešení	3
1.2 Analýza současných procesů	8
1.3 Analýza požadavků	8
2 Návrh	17
2.1 Doménový model	17
2.2 Výběr technologií	21
2.3 Datový model	25
2.4 Komunikační rozhraní	25
2.5 Uživatelské rozhraní	27
3 Realizace	31
3.1 Vývojové nástroje	31
3.2 Použité knihovny	31
3.3 Nasazení	35
3.4 Budoucí vývoj	35
4 Testování	37
4.1 Testování programátorem	37
4.2 Kognitivní průchod	37
4.3 Heuristické vyhodnocení	41
Závěr	45
Použité zdroje	47
A Seznam použitých zkratk	51

B	Diagramy aktivit	53
C	Databázový model	57
D	Návrh uživatelského rozhraní	61
E	Ukázka prototypu aplikace	63
F	Obsah přiloženého CD	67

Seznam obrázků

1.1	Ukázka rozhraní aplikace SmartPOD	5
1.2	Ukázka editoru aplikace Canvas	7
1.3	Případy užití	11
2.1	Doménový model	18
2.2	Seznam destinací	28
2.3	Detail destinace	29
2.4	Detail dodacího listu	30
3.1	RecyclerView	32
B.1	Zahájení jízdy	53
B.2	Průběh jízdy	54
B.3	Změny v průběhu jízdy	55
B.4	Ukončení jízdy	56
B.5	Předání zboží	56
C.1	Databázový model	59
D.1	Směna přepravních obalů	61
D.2	Tankování	61
D.3	Podmínky přepravy zboží	62
D.4	Neplánované zastavení	62
D.5	Application Flow	62
E.1	Seznam destinací	63
E.2	Detail destinace	63
E.3	Uzavření destinace	64
E.4	Detail dodacího listu	64
E.5	Tankování	64
E.6	Směna přepravních obalů	64

E.7	Informace o jízdě	65
E.8	Přihlášení	65
E.9	Podpis zákazníka	65
E.10	Dokončená jízda	65

Seznam tabulek

1.1	Pokrytí funkčních požadavků	15
2.1	Zastoupení mobilních operačních systémů na trhu	21
2.2	Zastoupení verzí OS Android	22

Úvod

S doručováním zásilek se jako zákazníci setkáváme dnes a denně. Mnoho služeb nabízí sledování zásilek pro zákazníky, změny času dodání a další operace. Ve své práci jsem se zaměřil na práci se zásilkami z druhé strany. Na to, aby měli kontrolu nad zásilkami i dispečeri, kteří řidičům rozdělují práci. Také aby řidiči měli co nejméně nadbytečné práce s administrací. Administrativní procesy se díky digitálnímu zpracování více rozmělní mezi automaticky sebrané informace a informace zadané ručně řidiči.

Aplikací s tímto účelem příliš mnoho neexistuje a pokud ano, jsou často zastaralé nebo nepohodlné pro užívání. O kvalitně zpracované řešení, které zdigitalizuje procesy spojené s přepravou zboží, je na trhu zájem, z toho důvodu mi toto téma bylo nabídnuto pro zpracování v bakalářské práci. V rámci práce budou zanalyzována současná řešení a na jejich základě budou spolu se zadavatelem práce specifikovány požadavky na nové řešení.

Cílem této bakalářské práce je navrhnout a implementovat prototyp aplikace pro řidiče přepravních služeb provozovaný na OS Android. Pomocí této aplikace bych chtěl dosáhnout snížení administrativní zátěže v oblasti přepravy zboží a umožnit větší kontrolu nad aktuálně převáženým zbožím pro dispečery.

Práce sestává z několika částí. Analytická část popisuje analýzu existujících řešení, na jejímž základě jsou následně sestaveny požadavky na mobilní aplikaci. Následující část popisuje jednotlivé části návrhu aplikace, od zmapování tříd problémové domény v podobě doménového modelu, po návrh uživatelského a komunikačního rozhraní. Poslední část se zabývá samotným vývojem a testováním prototypu. Na závěr je shrnut přínos a plány pro budoucí vývoj aplikace.

Analýza

1.1 Analýza existujících řešení

Při tvorbě nové aplikace je důležité prozkoumat existující řešení. Důvodů je hned několik. Prvním je ověření, zda neexistuje nějaké, na kterém lze založit vlastní vývoj. Dalším důvodem je možnost poučit se z chyb jiných aplikací a zároveň zjistit, které části fungují správně a není důvod na jejich fungování něco měnit. V této sekci se tudíž zaměřím na analýzu již existujících řešení s obdobným zaměřením.

1.1.1 Microlise SmartPOD

Jedná se o velmi komplexní řešení vyvíjené společností Microlise. Hlavní náplní aplikace je management jízd pro řidiče přepravních služeb. Ten umožňuje srozumitelný soupis jednotlivých míst doručení spolu s adresami a kontakty, potvrzování zásilek v reálném čase, focení poškozených zásilek, skenování čárových kódů a podpisy zákazníků v rámci aplikace. Mezi doplňkové služby patří kontrola a potvrzení vozidla, telefonní a e-mailový seznam s možností odesílat SMS, e-maily a uskutečňování telefonátů na povolená čísla. Umožněna je i navigace na cílovou destinaci a management úkolů při dodání. [1]

Aplikace se dodává i s Customer Service Module, který data exportuje do systémů třetích stran, umí také generovat poznámky k dodáním v PDF a statistiky kvality dodání. Řešení umožňuje i sledování vozidel v reálném čase, pokud jsou zařízení aktuálně připojena k GPS a internetu. Infrastruktura tohoto systému je tedy velmi robustní, sestává ze samotné mobilní aplikace, centrálního serveru pro sledování dat z mobilní aplikace, ale i modulu pro export dat do zákazníky již využívaných systémů. Systém je využíván například známou přepravní společností DHL [2].

Z dostupných zdrojů bych toto řešení označil jako jedno z nejrobustnějších na trhu. Provozováno je na široké škále zařízení. Hlavní mobilní aplikace má verzi pro OS Android a pro Windows Mobile. Nabízí rozšíření pomocí

telematické jednotky pro zjištění polohy vozidla a kvality jízdy, případně zařízení DriveTab (upravený tablet s OS Android) pro snazší ovládání na velkém displeji.

Dle produktové brožury [3] došlo díky použití aplikace k 80% redukci papírování. Z takového čísla lze usoudit, že podobná aplikace přináší kýžené výsledky v redukci administrativní zátěže a má smysl její vývoj a nasazení do provozu.

Nabízeny jsou i vcelku netradiční funkce, které mě zaujaly. Pokud se řidič dramaticky odchýlí od naplánované trasy či časového harmonogramu, je vyzván k odůvodnění činnosti. Další zajímavostí je ochrana proti zneužití aplikace v podobě podpisu řidiče před zahájením jízdy. Je to jakási forma dvoufázového ověření, kdy je k přihlášení požadováno nejen jméno a heslo, ale i podpis. Podpis řidiče je možné vyžádat i při plnění různých úkolů, u kterých by případné nesplnění mohlo vést k problémům se zákonem.

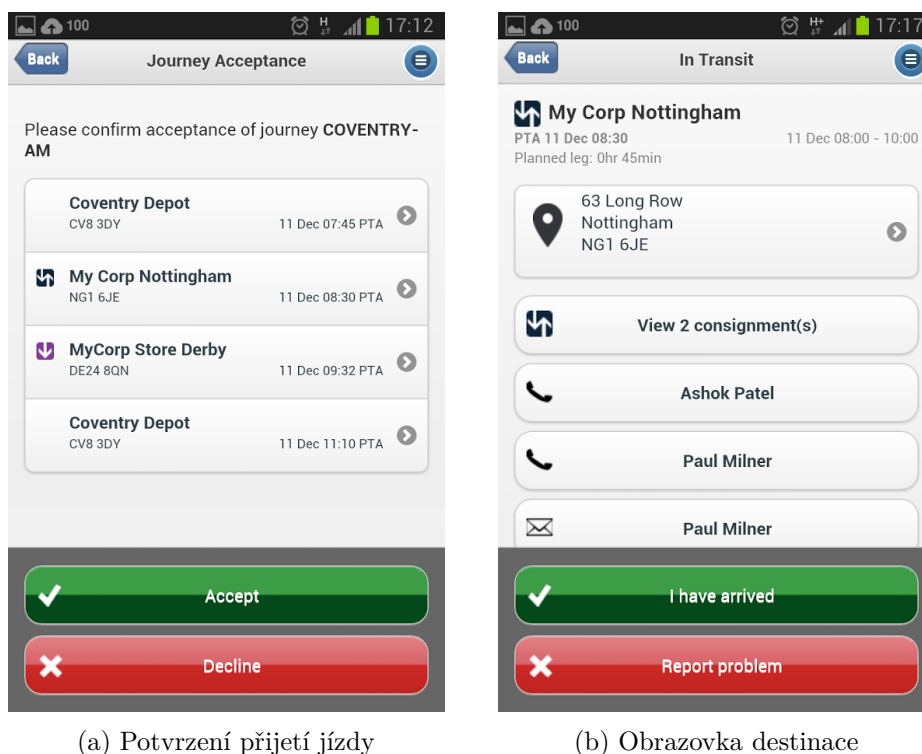
Jediné, čím mě aplikace příliš nepřesvědčila, je její vzhled. Nedrží se design guidelines pro OS Android. Pravděpodobně je to způsobeno konverzí z původní verze pro Windows Mobile. Designové principy pro tento systém byly navrženy Android UX týmem, a mají za cíl sjednotit principy interakce uživatele s aplikací. Uživatelé, kteří jsou na ekosystém daného operačního systému zvyklí, se lépe orientují v prostředí nových aplikací, již ví, co mají očekávat a jak s aplikací interagovat. [4] Bylo by třeba aplikaci spolu s jejími uživateli podrobit vhodným testům, aby bylo zjištěno, zda vlastně jsou uživatelé zvyklí na dané principy, případně zda principy interakce v rámci aplikace nejsou pro dané aktivity vhodnější, jednodušší. Ukázka rozhraní je k nalezení na obrázku 1.1.

1.1.2 DPD mobilní terminál

Klíčovou částí řešení využívaného v přepravní společnosti DPD je specializovaný mobilní terminál. Z hlediska hardwaru se jedná o zařízení Honeywell Dolphin 99EX. [6] V tomto terminálu se slučují přednosti i problémy celého řešení. V následujících odstavcích shrnu výhody a nevýhody v současnosti používaného specializovaného zařízení.

Z pozitiv bych jmenoval pohodlnou čtečku 1D a 2D kódů, tento model nabízí i přídatnou rukojeť, držení je potom obdobné jako u samostatných skenerů čárových kódů. Další výhodou zařízení na míru je odolnost, nemělo by mít problém s pády na beton z běžné výšky. Výdrž na baterii přesahující 8 hodin běžné zátěže je také vhodná, řidič nemusí mít o baterii zařízení v průběhu pracovní směny starost. Lze použít i přídatnou baterii, která zvyšuje výdrž a kapacitu baterie cca o třetinu.

Na druhé straně ale spatřuji nevýhody takového proprietárního řešení. Obsahuje zastaralý hardware i software. Dané je to dlouhým testováním před nasazením do provozu jak ze strany výrobce, tak ze strany korporace. Disponuje pouze 256MB RAM, jednojádrovým procesorem s frekvencí 1 GHz a OS Windows Embedded Handheld 6.5. Fotoaparát má rozlišení pouhých 3,1 Mpx.



(a) Potvrzení přijetí jízdy

(b) Obrazovka destinace

Obrázek 1.1: Ukázka rozhraní aplikace SmartPOD [5]

Takové parametry mohou vést k nedostatečné rychlosti odezvy přístroje a také k nekvalitním fotografiím určeným pro kontrolu procesu přepravy zboží.

Dalším problémem, který se týká zejména tuzemských podmínek, je 3,75G modul určený pro komunikaci po síti. V České republice se pokrytí UMTS/HSPA+ signálem vyskytuje zejména ve velkých městech, daleko větší pokrytí poskytují naši operátoři v sítích LTE, jinak nazývané 4G. [7] V důsledku toho jsou s tímto zařízením ve skladech, ze kterých je často potřeba zboží vozit, řidiči odkázáni na používání sítí druhé generace, konkrétně GPRS, nebo častěji EDGE. Datové přenosy v těchto sítích jsou obvykle limitovány na pouhých 135 kbit/s, což může vést ke zhoršené odezvě zařízení [8]. Problémy to může způsobovat zejména při přenosu objemnějších dat, například fotografií. Displej zařízení je rezistivní, což znamená méně přesné snímání dotyků prsty. Ovládání prsty je však pohodlnější, než stylusem. Zařízení tento nedostatek mohou řešit využitím hardwarové alfanumerické klávesnice. Tento displej byl nejspíše využit kvůli podpoře podpisů, nyní již ale lze nalézt i mnohá řešení využívající kapacitní displej s podobnou, či dokonce vyšší přesností. Nová řešení obvykle využívají stylus připojený přes Bluetooth. Problematickým faktorem zařízení je i jeho velká hmotnost, váží přes 500 g. Vestavěný A-GPS modul je využitelný pro ověření správné polohy, pro navigaci je však displej s nízkým rozlišením (480x640 pixelů) a úhlopříčkou 3,7 palců nevhodný.

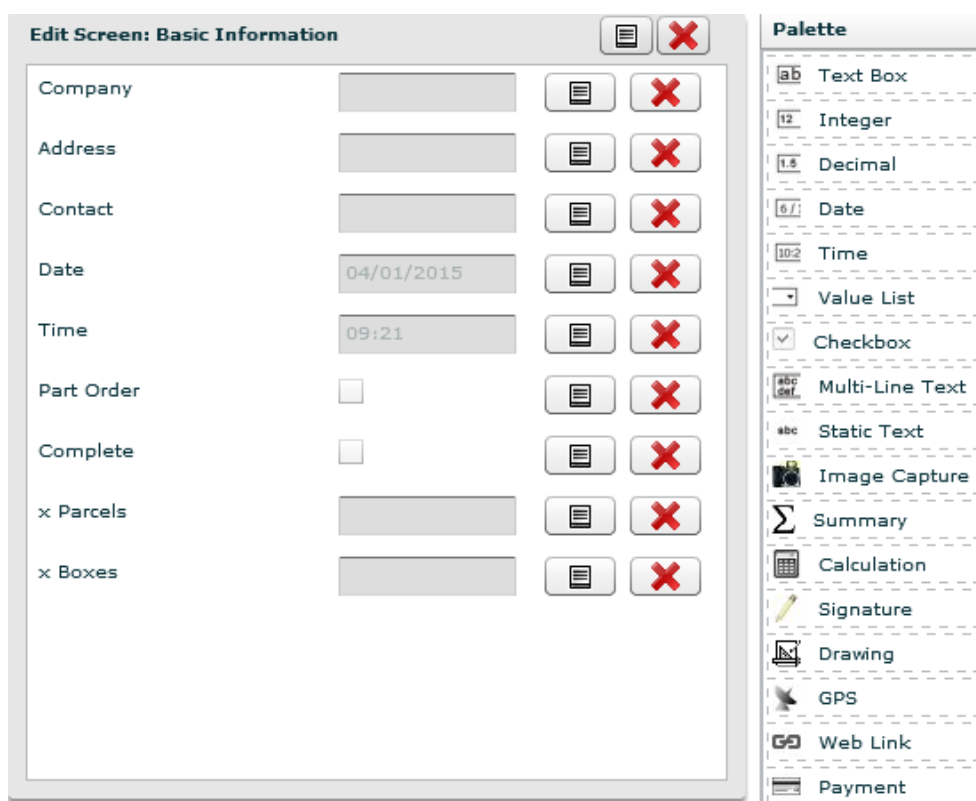
1.1.3 Canvas

Hlavním cílem tohoto nástroje je eliminace papírových formulářů. Snaží se o to skrze šablony aplikací, které lze snadno upravit pro potřeby dané společnosti. Jsou poskytovány šablony pro různé účely, jedna z nich je určena právě k potvrzování předání zboží. Osobně jsem k takovým řešením skeptický. Pomocí předpřipravených šablon je možné vytvořit jednoduchý prototyp, ale nikoliv robustní řešení na míru potřebám společnosti. Mé obavy se v průběhu testování aplikace potvrdily.

Webový portál dává na výběr z několika předpřipravených šablon, případně je možné aplikaci tvořit od nuly. Pro potřeby testování byla zvolena předpřipravená aplikace pro potvrzování dodávek zboží. Po vytvoření je možné aplikaci rovnou používat, případně upravit ve webovém editoru. Tento editor je velmi jednoduchý na užívání, ale na druhou stranu nenabízí příliš mnoho funkcí. K jeho funkčnosti je třeba mít nainstalovaný Flash Player. V editoru je možné přidávat jednotlivé obrazovky aplikace a do nich vkládat prvky z palety. Paleta obsahuje běžná vstupní pole pro zadávání textu, čísel a data a času. Dalšími prvky jsou checkbox, běžný text a automatický výpočet z vybraných polí pomocí zadaného vzorce. Méně tradičními možnostmi jsou pořízení fotografie, podpisu, případně GPS souřadnic. Rozhraní editoru je zobrazeno na obrázku 1.2. Atributy mají i různé vlastnosti, například výchozí hodnotu, minimum a maximum, zda mají být automaticky odeslané e-mailem, případně zda jejich hodnoty mají být viditelné ve vygenerovaných PDF. Velmi užitečnou možností je nahrání vlastního formuláře, na jehož základě se vytvoří aplikace obsahující pole identická s těmi na formuláři. Na vyplňování formuláře může spolupracovat i více uživatelů.

Hlavním problémem editoru a v důsledku i vygenerované aplikace je nemožnost měnit layout aplikace. Všechny prvky rozhraní jsou automaticky řazeny pod sebe. Rozhraní tudíž nevypadá dobře a nevyužívá rovnoměrně prostor displeje. Je to pravděpodobně daň za multiplatformní aplikaci. Bez složitějšího nastavování funguje vytvořená aplikace na široké škále zařízení, jsou podporovány platformy Android, iOS, BlackBerry, Windows Mobile i běžná Windows. Dokonce nabízí i některé specializované funkce, jako využití Samsung S Pen a podobně. Nasazení je realizováno formou zastřešující aplikace, do které lze po přihlášení stáhnout vytvořené aplikace příslušící danému uživatelskému účtu.

Systém nabízí export do PDF a s tím související nastavení formátu exportu, pro kompatibilitu s původními papírovými dokumenty. Také nabízí prohlížení dat ve webovém rozhraní, ale i export dat pomocí API. Je nabízeno REST API, které pracuje s daty ve formátu XML. [9]



Obrázek 1.2: Prostředí editoru aplikace Canvas [9]

1.1.4 Řešení bez mobilní aplikace

Pokud firma vůbec nevyužívá mobilní aplikace pro řidiče, tak je práce s potřebnými dokumenty velmi zdoluhavá a dispečer plánující jízdy pro řidiče nemá žádný přehled v průběhu jízdy. Před jízdou naplňuje dispečer jednotlivé zastávky složené z nakládek a vykládek zboží, spolu s jejich pořadím, ve specializované desktopové aplikaci. Následuje převzetí přepravních dokumentů řidičem. Řidič před jízdou musí rovněž vyplnit záznam o vozidle a jeho aktuálním stavu do stazky (dokument se záznamem o provozu vozidla).

Hlavní fází je samotná jízda, v průběhu které se do stazky zapisují informace o tankování a stavu tachografu na jednotlivých zastávkách se zbožím. V případě problémů s dodávkami zboží jsou tyto informace zapisovány do dodacích listů. Tamtéž jsou zapisovány směny přepravních obalů. Pokud se v průběhu jízdy vyskytne změna, je třeba se s řidičem telefonicky spojit. Pokud je řidič právě v oblasti bez pokrytí signálu, je situace problematická. Řidič může jet na nějaké místo zbytečně, případně je třeba hovor opakovat, což zdržuje čas pracovníků na dispečinku. Po uskutečnění celé jízdy řidič odevzdá ručně vyplněné dokumenty (tzn. stazku a dodací listy). Následně musí být dokumenty manuálně přepsány do elektronické podoby. Tato činnost může

zabrat již v případě několika desítek zaměstnaných řidičů celou pracovní dobu jednoho zaměstnance. Problematickou částí je také duplikace informací, některé věci musí být zapsány do dodacích listů i do stazky. Část informací je známá již před cestou, ale přesto musí být vepsány ručně do papírových dokumentů. Tyto problémy může řešit mobilní aplikace či serverová aplikace, která dokáže vyplňovat automaticky „zjevné“ informace, ať už jde o aktuální čas, rozdíly stavu tachografu, či něco obdobného. Znázornění těchto procesů pomocí diagramů je k nalezení v sekci 1.2. Tato práce se zaměřuje zejména na nasazení v prostředí, kde v současné době mobilní aplikace využívány nejsou.

1.2 Analýza současných procesů

Diskusí se zadavatelem a pracovníky byly zmapovány aktuální procesy probíhající na pracovišti nevyužívajícím mobilní aplikace pro řidiče. Zachyceny byly pomocí diagramů aktivit, které jsou k nalezení v příloze B.

1.3 Analýza požadavků

Na základě diskusí se zadavatelem a analýze stávajících procesů byly stanoveny následující požadavky na aplikaci.

1.3.1 Funkční požadavky

Aplikace

- **Autentizace uživatelů** Před jízdou se řidič přihlásí do aplikace pomocí svých přihlašovacích údajů. Po autentizaci proběhne stažení nových zakázek a informací o jízdě (SPZ auta, SPZ návěsu a dalších).
- **Procházení zakázek** Zakázky budou v aplikaci seřazeny podle termínu dodání určeného dispečerem, seskupeny dle místa doručení a rozděleny na vykládky a nakládky. V rámci podrobností zakázky bude obsažena adresa klienta, souřadnice, kontaktní informace a jednotlivé zásilky.
- **Potvrzení dodávek zboží** Dispečer má zájem o aktuální přehled dodaných zásilek, s pomocí mobilní aplikace bude řidič po uskutečnění dodávky zboží ihned zaznamenávat informaci o předání/přijetí zboží. V rámci potvrzení dodávky bude možné zaznamenat i výměnu přepravních obalů. Zároveň bude možné potvrdit přijetí zboží pomocí podpisu zákazníka přímo v aplikaci. V rámci potvrzení dodávky bude vyplněn stav dodání, tedy zda byla zásilka dodána v pořádku, nelze ji dodat, případně byla poškozená a podobně.

- **Sledování polohy vozidla** Aplikace bude při potvrzení dodávky a v průběhu jízdy periodicky hlásit aktuální polohu vozidla pro možnost hodnocení kvality jízdy z pozice dispečera.
- **Specializované pokyny pro přepravu zboží** Pro různé zásilky, případně posloupnosti zásilek, je třeba dodržet speciální postup při přepravě. V jednotlivých místech nakládky/vykládky bude tento postup, ať už generovaný dispečerovým systémem, či zapsaný dispečerem manuálně, zobrazen. Jedná se například o pokyny k dodržení GMP norem, které mimo jiné požadují vyčištění vozidla po převozu zboží škodlivého pro následně převážené zboží.
- **Pořizování médií k zásilkám** K zásilce bude možné pořídit fotografii a podpis, kvůli důkazu o stavu zásilky a předání.
- **Záznam provozu vozidla** Řidič před jízdou, na každé zastávce se zbožím a po ukončení jízdy zaznamená stav tachografu pro možnost vyplnění stazky. Také bude zaznamenávat čerpání pohonných hmot v průběhu jízdy.
- **Záznam pracovního výkonu** Bude umožněn záznam informací o jízdě, spolu s časem prováděných akcí. Jedná se zejména o vyplnění začátku jízdy spolu s detaily o vozidle, konce jízdy, případně neplánovaných zastávek v jejím průběhu.

API

- **Autentizace uživatelů** Požadavky na server budou autentizovány.
- **Download** Mobilní aplikace stáhne přepravní listy, případně jejich aktualizace, připravené dispečerem pro konkrétního řidiče.
- **Upload** Nahrávání potvrzení dodávek zboží, podrobností o dodání, potvrzujících fotografií, stavu vozidla a dalších informací.

1.3.2 Nefunkční požadavky

- **Architektura klient-server** Aplikace bude sestávat ze serverové a klientské části.
 - Na serveru se bude nacházet úložiště dat, ve kterém budou uložena data importovaná z aplikací dispečerů, kteří určují práci řidičům. Na serveru také bude provozováno rozhraní pro komunikaci mezi mobilní aplikací a serverem (API).
 - Klientská část, mobilní aplikace pro OS Android, bude se serverem komunikovat pomocí API. Zároveň bude klient disponovat místním

úložištěm, které uchová důležitá data potřebná i v případě ztráty signálu.

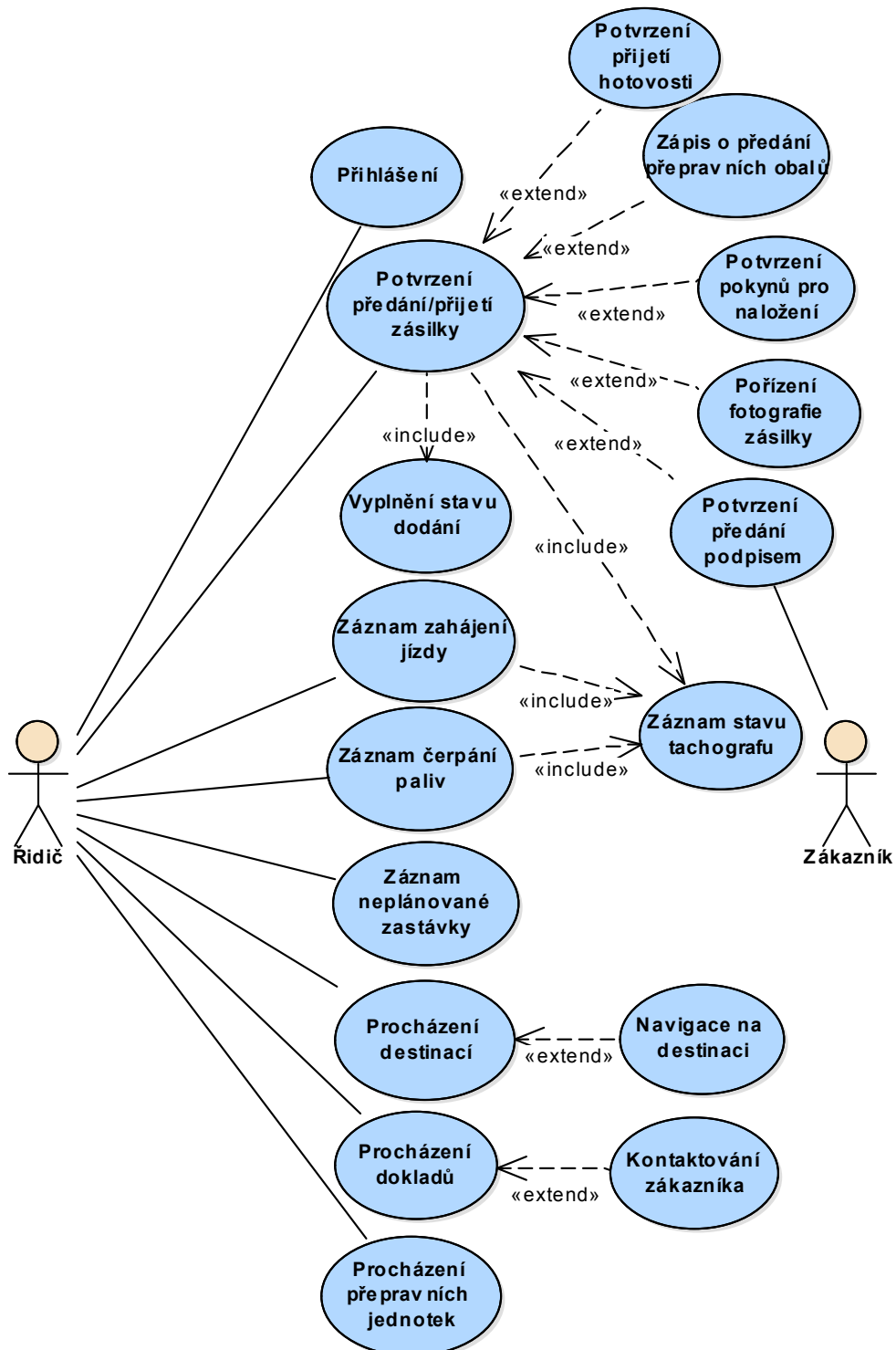
- **Platforma OS Android** Mobilní aplikaci bude možné provozovat na platformě Android. Aplikace bude optimalizovaná pro užití na mobilním telefonu.
- **Provoz online i offline** Přístup k internetu je vyžadován pouze pro počáteční přihlášení, stažení nové jízdy a synchronizaci dat po ukončení jízdy. Celá jízda může probíhat v offline režimu, potřebná data budou uložena v telefonu.
- **Uživatelské rozhraní** Bude zajištěna konzistence rozhraní, dodržování designových principů a standardního vzhledu cíleného operačního systému.

Hlavním cílem mobilní aplikace je nahradit současnou papírovou stazku, do které řidiči zaznamenávají informace o provozu vozidla v rámci jízdy. Tento dokument totiž slouží pouze pro interní informace firmy, na rozdíl od dodacích listů, které je často třeba předat v papírové podobě firmám třetí strany. Dodací listy v papírové formě je tudíž mnohem náročnější ze dne na den odstranit z procesu dodání. Jejich úplná digitalizace záleží na dohodě s mnoha třetími stranami. Z toho důvodu bude předmětem projektu digitalizace zejména těch částí, které jsou relevantní pro firmu, která zboží dodává, a pro řidiče, který je prostředníkem mezi dodavatelem a zákazníkem.

1.3.3 Případy užití

Z výše popsaných funkčních požadavků vyplývají případy užití znázorněné na diagramu 1.3. V následujících odstavcích jsou popsány scénáře případů užití.

- **Přihlášení** Umožňuje řidiči přihlásit se do mobilní aplikace a stáhnout zadanou jízdu
 1. Řidič spustí mobilní aplikaci
 2. Pokud není řidič přihlášen, zobrazí aplikace přihlašovací obrazovku
 3. Řidič vyplní své přihlašovací údaje
 4. Po přihlášení se stáhne aktuální zadaná jízda, v případě, že žádná není dostupná, o tom aplikace informuje řidiče
- **Procházení destinací** Umožňuje procházet destinaci v rámci jízdy
 1. Řidič spustí mobilní aplikaci
 2. Aplikace zobrazí dostupné destinace
 3. Řidič vybere destinaci, ke které chce zobrazit rozšířené informace



Obrázek 1.3: Diagram případů užití

4. Aplikace zobrazí informace o dané destinaci
- **Procházení dokladů** Umožňuje procházet doklady přidružené k jednotlivým destinacím
 1. Řidič provede případ užití *Procházení destinací*
 2. Řidič vybere ze seznamu dokladů doklad, o kterém chce zobrazit rozšířené informace
 3. Aplikace zobrazí detaily o vybraném dokladu
 - **Procházení přepravních jednotek** Umožňuje procházet přepravní jednotky přidružené k dokladům
 1. Řidič provede případ užití *Procházení dokladů*
 2. Aplikace na obrazovce zobrazí přepravní jednotky přidružené k danému dokladu
 - **Navigace na destinaci** Umožňuje navigovat řidiče na vybranou destinaci
 1. Řidič vybere kýžnou destinaci
 2. Aplikace zobrazí rozšířené informace o destinaci
 3. Řidič zvolí možnost navigovat na vybranou destinaci
 4. Aplikace otevře aplikaci umožňující navigaci
 5. Řidič je navigován na danou destinaci
 - **Kontaktování zákazníka** Umožňuje řidiči kontaktovat zákazníka
 1. Řidič vybere destinaci
 2. Aplikace zobrazí rozšířené informace o destinaci a přidružené dodací listy
 3. Řidič vybere dodací list, ohledně kterého má zájem kontaktovat zákazníka
 4. Aplikace zobrazí rozšířené informace o dodacím listu
 5. Řidič vybere možnost kontaktovat zákazníka
 - **Potvrzení předání/přijetí zásilky** Umožňuje řidiči potvrdit předání zásilky zákazníkovi
 1. Řidič vybere destinaci, na které se nachází
 2. Aplikace zobrazí vybranou destinaci
 3. Řidič potvrdí příjezd na danou destinaci
 4. Řidič vybere dodací list, pro který chce potvrdit předání

5. Aplikace zobrazí dodací list spolu se zbožím, které obsahuje
 6. Řidič provede případ užití *Vyplnění stavu dodání*
 7. Řidič potvrdí předání zásilky
- **Vyplnění stavu dodání** Umožňuje řidiči vyplnit stav dodání zásilky
 1. Řidič provede případ užití *Potvrzení přijetí zásilky*
 2. Řidič vybere stav dodání z předdefinovaného seznamu stavů
 - **Potvrzení přijetí hotovosti**
 1. Řidič provede případ užití *Potvrzení předání zásilky*
 2. Řidič vyplní, zda přijal hotovost
 - **Zápis o předání přepravních obalů** Umožňuje řidiči zápis o předání přepravních obalů
 1. Řidič provede případ užití *Procházení dokladů*
 2. Řidič vybere možnost zápis předání přepravních obalů
 3. Aplikace zobrazí přepravní obaly již přidružené k danému dokladu
 4. Řidič vybere přidání předaného obalu
 5. Řidič vyplní počet předaných obalů
 6. Řidič vybere typ předaných obalů z předdefinovaného seznamu
 7. Řidič vybere, zda obaly předal, či obdržel
 8. Řidič potvrdí směnu obalů
 - **Sledování polohy vozidla** Aplikace bude zaznamenávat polohu vozidla
 - **Potvrzení pokynů pro naložení** Umožňuje řidiči potvrdit rozšířené pokyny pro naložení
 1. Řidič provede případ užití *Procházení destinací*
 2. Řidič vybere možnost zobrazení pokynů pro naložení
 3. Aplikace zobrazí pokyny pro naložení
 4. Řidič splní pokyny pro naložení a v aplikaci vybere potvrzení jejich splnění
 - **Pořízení fotografie zásilky** Umožňuje řidiči pořídit fotografii zásilky
 1. Řidič provede případ užití *Procházení dokladů*
 2. Řidič vybere možnost vyfocení fotografie
 3. Aplikace spustí výchozí aplikaci pro pořizování fotografií

4. Řidič vyfotí fotografii
 5. Řidič potvrdí užití fotografie k danému dokladu
- **Potvrzení předání podpisem** Umožňuje řidiči připojit zákazníkův podpis k dokladu
 1. Řidič provede případ užití *Procházení dokladů*
 2. Řidič vybere možnost pořízení podpisu
 3. Přístroj bude předán zákazníkovi
 4. Zákazník se podepíše na řidičově zařízení
 5. Řidič potvrdí vytvořený podpis
 - **Záznam čerpání paliv**
 1. Řidič spustí mobilní aplikaci
 2. Řidič vybere možnost tankování
 3. Aplikace zobrazí obrazovku tankování
 4. Řidič vyplní údaje o tankování
 5. Řidič vybere tankované palivo z předdefinovaného seznamu
 6. Řidič potvrdí ukončení tankování
 - **Záznam stavu tachografu** V rámci akcí zahájení jízdy, potvrzení předání zásilky a záznamu čerpání paliv bude vyplněn aktuální stav tachografu
 - **Záznam neplánované zastávky** Budou evidovány neplánované zastávky v průběhu jízdy
 1. Řidič spustí mobilní aplikaci
 2. Řidič vybere možnost přidání neplánované zastávky
 3. Aplikace zobrazí obrazovku pro přidání neplánované zastávky
 4. Řidič vybere možnost začít neplánovanou zastávku
 5. Při ukončení zastávky řidič vybere možnost ukončení
 - **Záznam zahájení jízdy** Řidič eviduje, kdy začíná jízdu, tedy pracovní směnu
 1. Řidič provede případ užití *Přihlášení*
 2. Řidič vybere možnost zahájit jízdu
 3. Aplikace zobrazí obrazovku zahájení jízdy
 4. Řidič provede případ užití *Záznam stavu tachografu*

5. Řidič zkontroluje předvyplněné údaje, případně je opraví
6. Řidič potvrdí vyplněné údaje
7. Aplikace zobrazí seznam destinací

Funkční požadavky Případy užití	Autentizace uživatelů	Procházení zakázek	Potvrzení dodávek zboží	Sledování polohy vozidla	Pokyny pro přepravu zboží	Pořizování médií k zásilkám	Záznam provozu vozidla	Záznam pracovního výkonu
Přihlášení	↑							
Procházení destinací		↑						
Procházení dokladů		↑						
Procházení přepravních jednotek		↑						
Navigace na destinaci		↑						
Potvrzení předání/přijetí zásilky			↑					
Vyplnění stavu dodání			↑					
Potvrzení přijetí hotovosti			↑					
Zápis o předání přepravních obalů			↑					
Sledování polohy vozidla				↑				
Potvrzení pokynů pro naložení					↑			
Pořízení fotografie zásilky			↑			↑		
Potvrzení předání podpisem			↑			↑		
Záznam čerpání paliv							↑	
Záznam stavu tachografu							↑	
Záznam neplánované zastávky								↑
Záznam zahájení jízdy								↑

Tabulka 1.1: Pokrytí funkčních požadavků

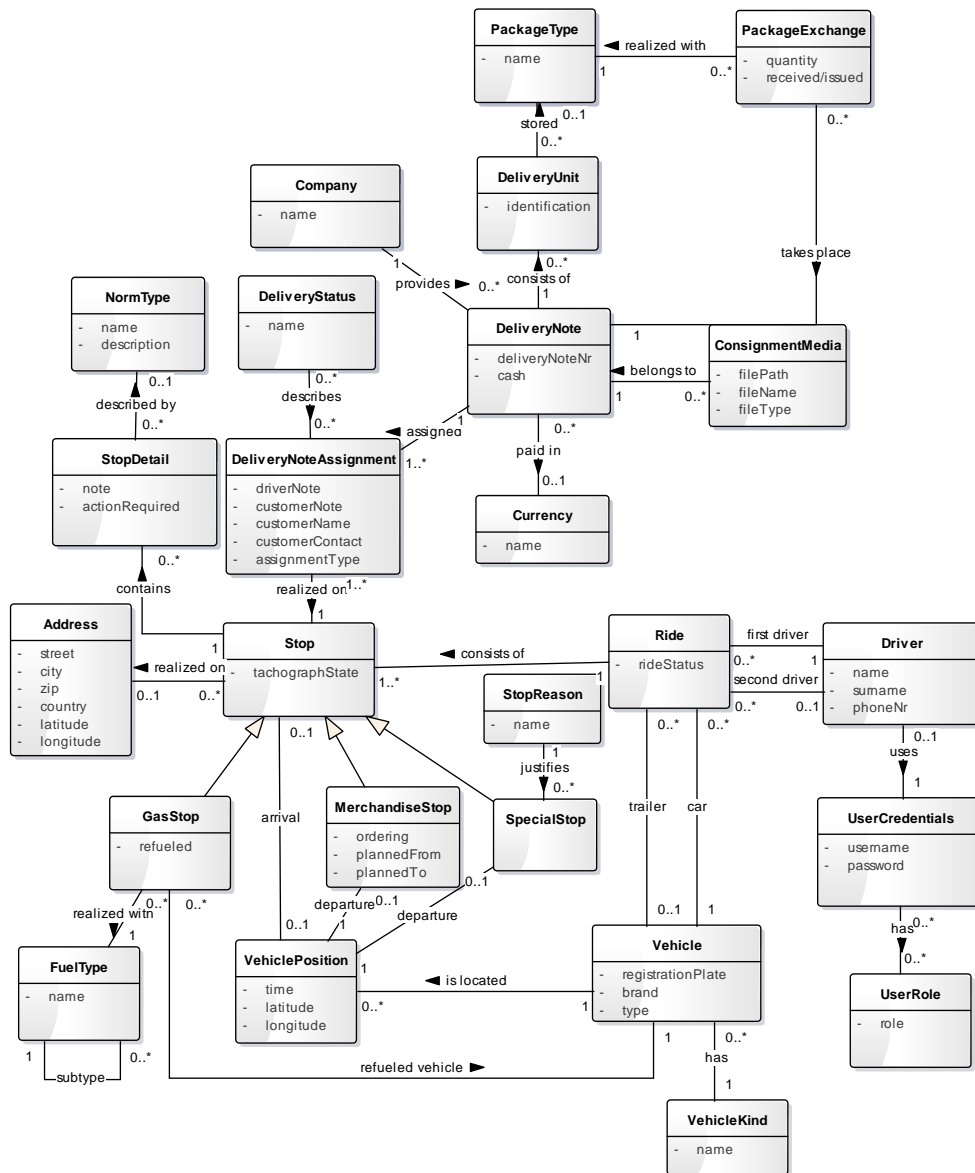
Návrh

2.1 Doménový model

Doménový model znázorňující klíčové entity a vztahy mezi nimi je zobrazen na diagramu 2.1.

- **UserCredentials** přihlašovací údaje sloužící k přihlášení k webovému API a do mobilní aplikace
 - **username** přihlašovací jméno
 - **password** heslo
- **UserRole** role uživatele, uplatněná v API při ověření práv na požadavek
 - **role** název uživatelské role
- **Driver** údaje o řidiči, kontaktní údaje
 - **name** jméno řidiče
 - **surname** příjmení řidiče
 - **phoneNr** telefonní číslo
- **Vehicle** vozidlo a jeho parametry
 - **registrationPlate** SPZ vozidla
 - **brand** značka vozidla
 - **type** model vozidla
- **VehicleKind** druh vozidla
 - **name** název druhu vozidla, například tahač
- **Ride** sloužící pro zapouzdření celé jízdy, zpravidla jedné řidičovy směny

2. NÁVRH



Obrázek 2.1: Doménový model

- **rideStatus** stav jízdy
Může nabývat následujících hodnot:
 - 0 – zadaná jízda
 - 1 – probíhající jízda
 - 2 – dokončená jízda
- **Stop** základ pro jednu řidičovu zastávku v průběhu jízdy
 - **tachographState** stav tachografu ve vozidle
- **MerchandiseStop** destinace naplánovaná dispečerem spolu s jejím pořadím
 - **ordering** řazení vytvořené dispečerem
 - **plannedFrom** kdy je řidič očekáván na destinaci
 - **plannedTo** dokdy je možné uskutečnit dodání na dané destinaci
- **SpecialStop** neplánované zastavení v průběhu jízdy
- **StopReason** důvod neplánovaného zastavení
 - **name** název důvodu zastavení
- **GasStop** zastávka z důvodu tankování
 - **refueled** natankované palivo v litrech
- **FuelType** typ paliva, může obsahovat podtypy, například nafta může mít podtypy běžná a bio
 - **name** název typu paliva
- **VehiclePosition** pozice vozidla, může být přiřazena k zastávce, případně může být zachycena i samostatně, pak je přiřazena pouze k vozidlu
 - **time** čas zachycení
 - **latitude** zeměpisná šířka
 - **longitude** zeměpisná délka
- **Address** adresa spolu s GPS souřadnicemi
 - **street** ulice
 - **city** město
 - **zip** poštovní směrovací číslo
 - **country** země

2. NÁVRH

- **latitude** zeměpisná šířka
- **longitude** zeměpisná délka
- **StopDetail** detail zastávky, může obsahovat poznámku od dispečera, případně vynucení akce od řidiče
 - **note** poznámka dispečera
 - **actionRequired** zda je vyžadováno provedení akce řidičem
- **NormType** typ normy, kterou je třeba splnit
 - **name** název normy
 - **description** popis normy
- **DeliveryNote** dodací list
 - **deliveryNoteNr** číslo dodacího listu přiřazené v externím systému, shodné s číslem na papírovém formuláři
 - **cash** hotovost, kterou je třeba převzít při vydání zboží
- **DeliveryNoteAssignment** přiřazení dodacích listů k zastávkám, dodací list je zpravidla přiřazen ke 2 zastávkám, nakládce a vykládce, každá má své stavy dodání a kontaktní osobu
 - **driverNote** poznámka řidiče
 - **customerNote** poznámka zákazníka
 - **customerName** jméno zákazníka
 - **customerContact** telefonní číslo zákazníka
 - **assignmentType** typ přiřazení, zda se jedná o nakládku či vykládku
- **DeliveryStatus** stav doručení zboží v dodacím listu na dané destinaci
 - **name** název stavu dodání
- **DeliveryUnit** jednotka zboží
 - **identification** viditelné značení jednotky zboží, například číslo palety, balíku a podobně
- **PackageType** typ přepravního obalu
 - **name** název obalu
- **PackageExchange** směna obalů
 - **quantity** množství předaných obalů

- **received/issued** zda byly obaly přijaty či vydány
- **ConsignmentMedia** multimédia
 - **filePath** umístění souboru
 - **fileName** název souboru
 - **fileType** typ souboru
- **Company** firma, pro kterou je daná zakázka dovážena
 - **name** název firmy
- **Currency** měna, ve které je prováděna hotovostní platba
 - **name** název měny

2.2 Výběr technologií

2.2.1 Klient

Pro implementaci mobilní aplikace pro řidiče byla zvolena platforma OS Android. Jedná se o nejrozšířenější platformu na trhu, konkrétní zastoupení je 76,6 % dle dat společnosti IDC, viz tabulka 2.1. Většina současných řešení existuje ve formě aplikace pro specializovaný mobilní terminál založený na OS Microsoft Windows Embedded Handheld 6.5 a starší. Jedním z cílů práce je vyzkoušet jinou platformu, která má v současné době majoritní postavení na trhu mobilních zařízení a je stále v aktivním vývoji. Zařízení s OS Android jsou navíc dostupná v široké škále variant a cen a levnější modely jsou tedy dostupné i menším dopravním společností.

Aplikace bude kompatibilní s OS Android od verze 4.1. Jak vyplývá z tabulky 2.2, aplikace bude teoreticky dostupná na zhruba 87,5 % zařízení, která se připojují ke Google Play Store. Nejstarší kompatibilní verze byla vybrána jako kompromis mezi zastoupením zařízení a funkcemi, které jsou dostupné v novějších verzích systému. Cílením zejména na společnosti, které v současné době nevyužívají mobilní aplikace, by bylo možné dovolit i menší zpětnou kompatibilitu, nicméně novější verze nenabízejí tak zásadní nové funkce, bez

Operační systém	Podíl na trhu
Android	76.6 %
iOS	19.7 %
Windows Phone	2.8 %
BlackBerry OS	0.4 %
Ostatní	0.5 %

Tabulka 2.1: Zastoupení mobilních operačních systémů na trhu [10]

Verze	Kódové označení	API	Zastoupení
2.2	Froyo	8	0.4 %
2.3.3 – 2.3.7	Gingerbread	10	6.4 %
4.0.3 – 4.0.4	Ice Cream Sandwich	15	5.7 %
4.1.x	Jelly Bean	16	16.5 %
4.2.x		17	18.6 %
4.3		18	5.6 %
4.4	KitKat	19	41.4 %
5.0	Lollipop	21	5.0 %
5.1		22	0.4 %

Tabulka 2.2: Zastoupení verzí OS Android [13]

kterých by se nedalo obejít. Aplikace bude napsána v nejpoužívanějším a oficiálním jazyce pro tuto platformu, v Javě. Hlavním zdrojem pro návrh a implementaci aplikací pro OS Android bude publikace „Professional Android 4 Application Development“ [11] a webová stránka „Android Developers“ [12].

2.2.2 Hardware

Software vyvíjený pro OS Android je možné nasadit na širokou škálu hardwaru. Dle prostředí, ve kterém je plánováno nasazení, lze zvolit vhodné zařízení. Existuje tedy možnost používat aplikaci na voděodolném, případně extra odolném telefonu.

Jedním z požadavků na aplikaci je možnost pořídit elektronicky podpis přímo v aplikaci. Zařízení Samsung Galaxy Note, které obsahují vestavěný Wacom digitizér [14], spojují výhody kapacitního displeje, tzn. možnost rychle a spolehlivě ovládat zařízení dotyky prstů, s výhodami displeje rezistivního, tzn. přesné psaní stylusem. Z osobní zkušenosti bych navíc dodal, že psaní pomocí Samsung stylusu je přesnější, než to, které se používá u rezistivního displeje. Společnost Samsung poskytuje knihovnu S Pen, která podporuje vývoj aplikací využívajících právě zmíněný stylus pro Note zařízení. Existují i pasivní stylusy, které lze použít s každým zařízením, ale jedná se zjednodušeně jen o zmenšeninu prstu. Přesnost tedy není na vysoké úrovni, navíc není možné položit si ruku na displej bez toho, abyste porušili podpis, tak jako to umožňují S Pen řešení. V rámci prototypu bude využita právě platforma S Pen pro možnost podpisů zákazníků. Pro pořizování podpisů uvnitř prototypu bude vyžadováno zařízení Galaxy Note, na jiných zařízeních bude aplikace provozuschopná, ale funkce podpisu nebude dostupná.

2.2.3 Server

REST API

Z centrálního serveru je třeba poskytovat data pro klienty využívající mobilní zařízení. Standardním postupem pro komunikaci mobilního zařízení se vzdáleným úložištěm dat je vytvoření webové služby. Takovou službu nazýváme API, specifikuje, jak by měly služby využívající ji komunikovat. Je vhodné, aby API využívalo standardizovaný formát, který je snadné používat. Standardními architektonickými styly pro tvorbu webového API jsou zejména REST (Representational State Transfer) a SOAP (Simple Object Access Protocol) [15]. Dle [16] je vhodné použít REST v případě omezené šířky pásma, což pro připojení mobilním připojením jistě platí. Dalším z aspektů jsou bezstavové operace REST protokolu, při využití v terénu může dojít k výpadku signálu, tudíž nám tato vlastnost vyhovuje. Pro využití v rámci tohoto projektu byl na základě vhodných vlastností zvolen REST protokol. Webová služba využívající REST architekturu se nazývá RESTful API.

Spring Framework

Aplikace poskytující RESTful rozhraní bude vytvořena pomocí Spring IO platformy nabízející modulární enterprise framework [17]. Výhodou tohoto frameworku je shodný programovací jazyk jako u klientské části. To umožňuje dodržovat obdobné konvence pro psaní kódu a jednoduše sdílet model dat mezi serverovou a klientskou aplikací.

Hlavním modulem usnadňujícím vytvoření a nasazení aplikace vhodným pro použití v rámci práce je Spring Boot. Poskytuje prostředí v příkazové řádce, nastavení součástí frameworku pomocí anotací či XML konfiguračních souborů, případně kombinací obojího. Další důležitou součástí je integrace nástrojů potřebných k běhu serveru, v důsledku také usnadňuje a sdružuje potřebná nastavení webové aplikace. Konkrétně obsahuje podporu pro HTTP engine Tomcat, Undertow a Jetty. V rámci projektu bude jako webový server využit právě vestavěný HTTP engine Tomcat. [18]

JSON

RESTful rozhraní umožňuje užití široké škály formátu dat pro přenos mezi klientem a serverem, v implementaci byl použit formát JSON. Oproti obdobnému datovému formátu XML je datově úspornější a snadněji parsovatelný [19]. Kvůli integraci s frameworkem Spring bude využita knihovna Jackson [20] pro serializaci a deserializaci dat ve formátu JSON.

Hibernate ORM

Hlavním účelem knihovny Hibernate je objektově relační mapování. To umožňuje mapování relačních databází na objekty a naopak. V objektově orientova-

ném jazyce, jakým je Java, je to obzvláště žádoucí. Podporuje také objektově orientované principy, jako například dědičnost, polymorfismus, či kolekce.

Tato knihovna, respektive celý princip ORM, zvyšuje produktivitu práce v objektově orientovaných jazycích. Může zvýšit i efektivitu kódu, nabízí totiž princip tzv. lazy inicializace, která dotazuje databázi teprve ve chvíli, kdy je daná část dat třeba. Umožňuje také generování databáze, případně její úpravy, tato vlastnost se hodí zejména pro verzování databáze při práci v týmu. Užitečnou funkcí je také podpora mnoha databázových strojů, aplikace je tedy snáze přenositelná mezi různými databázovými stroji, než kdyby byl použit pouze přímý SQL kód na databázovém stroji. [21]

OAuth2

OAuth 2 protokol slouží k autorizaci požadavků klienta na server. Klientská aplikace pošle prvotní požadavek na ověřovací token. Tento požadavek obsahuje ID klientské aplikace a její tajný klíč registrovaný na serveru, spolu s ním jsou zasílány přihlašovací údaje uživatele. Po ověření na serveru dojde k odeslání tokenu, kterým se uživatel ověřuje do té doby, než vyprší tokenu platnost. Po vypršení platnosti je možné jej obnovit pomocí obnovovacího tokenu. [22]

Pro integraci s frameworkem Spring bude použita Spring Security OAuth2 knihovna. [23]

HTTPS

Komunikace se serverem bude probíhat protokolem HTTPS. Tento protokol nabízí dodatečnou vrstvu k HTTP v podobě SSL nebo TLS. Operuje na portu 443, který je podporován i operátory mobilních sítí. Slouží k ověření, zda komunikace probíhá se správným serverem, tedy chrání proti útoku „Man in the middle“, kdy je odposlouchávána, případně pozměněna komunikace třetí stranou na cestě mezi klientem a serverem. Další funkcí je šifrování dat na cestě mezi klientem a serverem. Pro plnou důvěryhodnost je třeba k užívání zakoupit SSL certifikát, s pomocí kterého je důvěryhodnost certifikátu ověřena certifikační autoritou. [24]

BCrypt

Funkce pro odvození klíče založená na symetrické blokové šifře Blowfish. Právě v šifře Blowfish je obsažena nejnáročnější operace v rámci celé funkce, výměna klíčů. To je ale zároveň výhoda, protože zpomaluje slovníkový útok. Funkci BCrypt ve webovém API použijeme pro bezpečné uložení uživatelských hesel ve formě jejich hashů spolu se solí. [25]

Gradle

Jedná se o open source systém automatizující build, testování a deployment. Také umožňuje správu závislostí knihoven, lze definovat závislosti na lokálních jar knihovnách, složkách se zdrojovými soubory, případně na vzdálených Maven repozitářích. V podstatě se jedná o modernější alternativu k obdobnému systému Maven, díky sdíleným repozitářům není problém ani migrace mezi systémy. [26] Rozdílem viditelným na první pohled je forma build skriptů. Starší systém Maven používá XML, kdežto Gradle využívá Groovy. Groovy je dynamický objektově orientovaný programovací jazyk využívající platformu Java [27]. Systém Gradle bude použit jak v serverové, tak v klientské aplikaci.

PostgreSQL

Požadovaný je databázový systém dostupný zdarma, z těch nejrozšířenějších se nabízí PostgreSQL a MySQL. Preferovaným systémem ze strany zadavatele byl PostgreSQL. Tento systém byl po analýze dostupných systémů vybrán pro použití k implementaci. Jedná se o open-source objektově relační databázový systém využívaný pro uložení dat na serveru. Systém je v aktivním vývoji již 15 let, podporuje ho i komunita, není tedy pravděpodobné, že by v blízké době přišel o podporu. [28]

2.3 Datový model

Datový model vychází z modelu doménového. V našem případě není příliš důležité, zda použijeme transformaci na databázový model nebo na model tříd. V rámci API budou třídy mapovány na relační databázi pomocí frameworku Hibernate ORM. Pro větší kontrolu nad aspekty, které se vyskytují v relačních databázích a nikoli v objektovém přístupu, byl vytvořen databázový diagram. Doplním, co z diagramu není zjevné. Automaticky generované id jsou vytvářeny pomocí sekvencí. Při využití PostgreSQL se přímo nabízí použít typ serial, který obstará vytvoření sekvence a výchozí hodnotu na následující hodnotu v dané sekvenci [29]. Ten nebyl použit pro větší nezávislost na databázovém stroji. Díky tomuto přístupu a využití ORM Hibernate je vytvářené API platformně nezávislé a až na drobné výjimky by neměl být problém databázi provozovat na jakémkoliv databázovém stroji spolupracujícím s Hibernate frameworkem.

Výsledný model je zobrazen na diagramu C.1. V této podobě je využit na serveru a s drobnými úpravami i v klientské aplikaci.

2.4 Komunikační rozhraní

Tato sekce popisuje komunikační rozhraní poskytované serverovou aplikací. Cesty k jednotlivým zdrojům jsou členěny do logických celků dle účelu jejich

2. NÁVRH

užití. Celé rozhraní bude zabezpečeno, přístupné pouze po přihlášení. Data jsou na server odesílána ve formátu JSON v těle požadavku, další informace o technické stránce jsou popsány v sekci 2.2. V rozhraní jsou použity metody GET pro stažení dat, POST pro nahrání nových dat a PUT pro změnu existujících dat.

2.4.1 Jízda

<code>/ride/download</code>	GET	Stažení jízdy pro přihlášeného uživatele
<code>/ride/update</code>	PUT	Změna detailů o jízdě (například použitého vozidla)

2.4.2 Zastávka

<code>/stop/reason</code>	GET	Stažení důvodů zastavení
<code>/stop/gas/type</code>	GET	Stažení typů paliva

Následující cesty začínají řetězcem `/ride/{id jízdy}`

<code>/stop/gas</code>	POST	Vytvoření zastávky pro tankování
<code>/stop/gas/{id}</code>	PUT	Změna zastávky pro tankování
<code>/stop</code>	POST	Vytvoření neplánované zastávky
<code>/stop/{id}</code>	PUT	Změna neplánované zastávky
<code>/stop/{id}/position</code>	POST	Vytvoření pozice vozidla

2.4.3 Dodací listy

Všechny cesty v této části začínají řetězcem `/delivery/{id}`.

<code>/media</code>	POST	Upload médií
<code>/exchange</code>	POST	Vytvoření nové směny obalů
<code>/exchange</code>	PUT	Změna směny obalů
<code>/status</code>	POST	Nový status dodání
<code>/delivery/package/type</code>	GET	Stažení typů přepravních obalů
<code>/delivery/status/type</code>	GET	Stažení povolených typů dodání

2.4.4 Ostatní

<code>/oauth/token</code>	POST	Přihlášení uživatele k rozhraní, vyžádání tokenu
<code>/oauth/token</code>	POST	Obnovení tokenu
<code>/user</code>	POST	Vytvoření nového uživatele, vyžaduje roli administrátora

2.5 Uživatelské rozhraní

Koncept uživatelského rozhraní byl vytvořen pomocí nástroje Balsamiq Mockups [30]. Tento nástroj umožňuje vytvoření wireframů, v překladu drátěných modelů. Wireframy slouží k rychlému a snadnému vytvoření prototypu uživatelského rozhraní. Prototyp obsahuje obdobné prvky jako reálné rozhraní cíleného operačního systému. Nástroj je založen na použití drag and drop (přetahování prvků rozhraní do modelu), tudíž podporuje rychlost prototypování a snadné opravy problémů identifikovaných při diskusi nad modelem se zákazníkem. Pro snadnější orientaci v prostupnosti obrazovek mezi sebou byl vytvořen diagram zobrazený v příloze D.5.

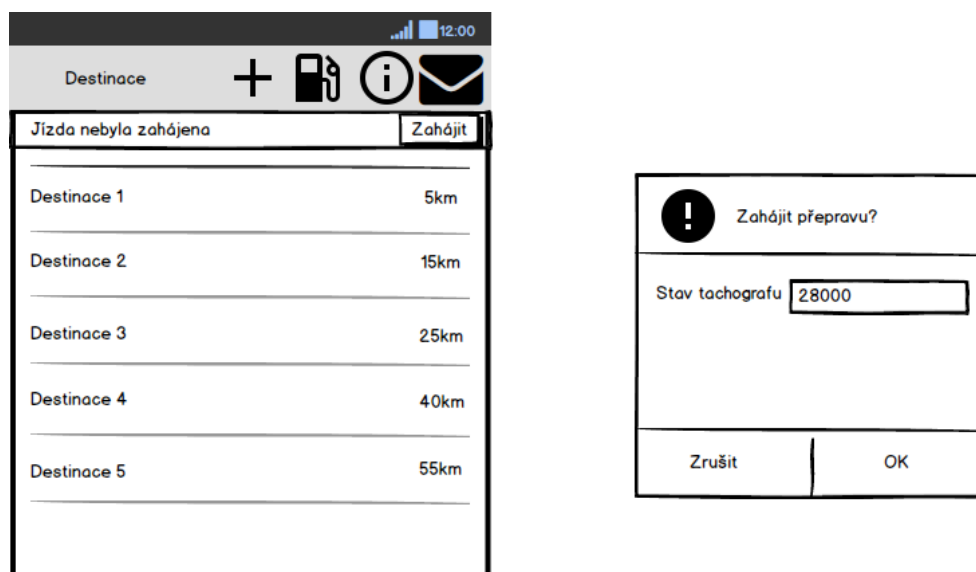
2.5.1 Hlavní menu

Prvotní koncept hlavního menu aplikace vybízel k aplikování toho, co je užité v obdobných systémech. Jednalo by se o menu s několika tlačítky, které by obsahovalo standardní operace, se kterými se řidič setkává. Přihlášení, tankování, neplánované zastavení, destinace, začátek, případně konec jízdy. Nicméně jak vyplynulo z analýzy procesů probíhajících v rámci pracovní směny, tento koncept není příliš efektivní.

V rámci prototypu rozhraní byl zvolen odlišný koncept pojetí hlavní obrazovky. Pro řidiče jsou nejdůležitější dispečerem naplánované destinace. Ty v modelu rozhraní zabírají největší část prostoru. Řidič má zájem projít si destinace i předtím, než vyrazí na danou jízdu, například pro zvážení vhodného času k výjezdu, vytvořený model tuto činnost podporuje. Pokud se řidič rozhodne začít jízdu, může tak učinit prostřednictvím tlačítka v horní části obrazovky, které se zobrazuje pouze za situace, že jízda nebyl zahájena, nerozptyluje tedy uživatele v průběhu užívání aplikace. Ve chvíli, kdy řidič zahájí jízdu, nabízí se mu již nejen možnost procházet destinace a jejich detaily, ale i zastávky vytvářet. Ovládání činností je v rámci celé aplikace koncentrováno do horního panelu, v prostředí OS Android nazývaného action bar.

Procházení destinacemi je realizováno pomocí klepnutí na položku seznamu zastávek. Do tohoto seznamu se řadí nejen plánované destinace, ale i usku-tečněné zastávky na tankování, případně neplánovaná zastavení. Seznam je řazen podle manuálního seřazení vytvořeného dispečerem a podle času od-jezdu z destinace. Zobrazeny jsou také orientační vzdálenosti k jednotlivým destinacím. Návrh hlavní obrazovky se seznamem destinací je k nalezení na obrázku 2.2. Po stisknutí zahájení jízdy se zobrazí dialog znázorněný vpravo na témže obrázku. Horní lišta obsahuje kromě popsaného neplánovaného zastavení a tankování tlačítko pro přechod na informace o jízdě, které byly později vyhodnoceny jako podstatné pouze při zahájení jízdy, kdy se potvrzují. V průběhu se již neobjevují. Poslední položkou v horním menu je ikona zpráv, která by v budoucnu mohla obsahovat zprávy od dispečera a případné aktualizace jízdy. Tato možnost nebyla obsahem této fáze prototypu.

2. NÁVRH



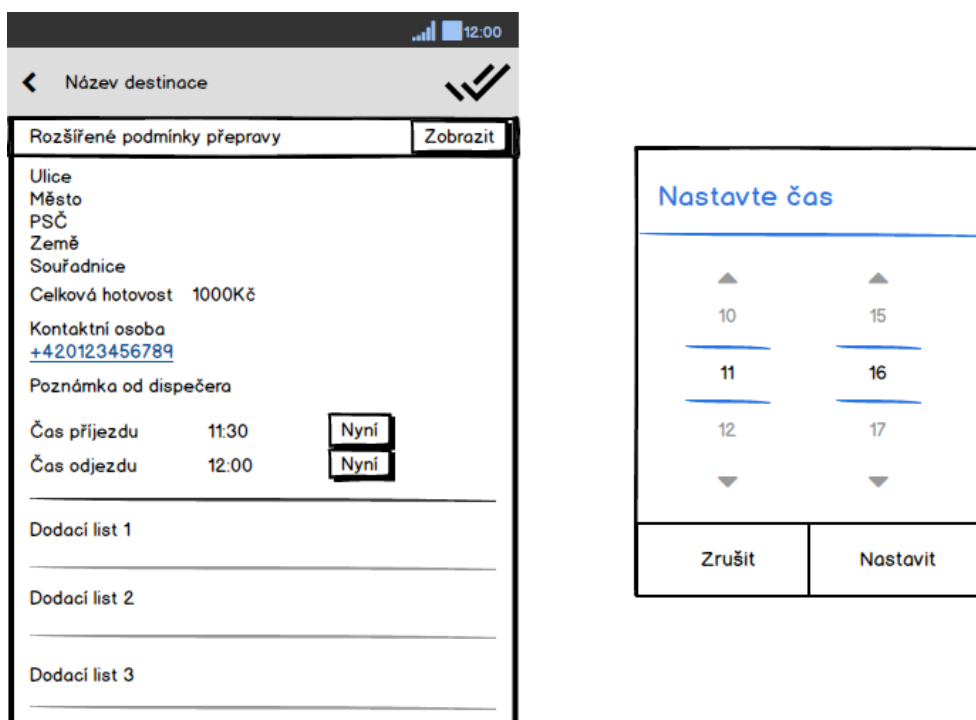
Obrázek 2.2: Seznam destinací

2.5.2 Přihlášení

V případě, že uživatel není k aplikaci přihlášen, zobrazí se mu po spuštění aplikace přihlašovací obrazovka. Pro přihlášení bude vyžadováno uživatelské jméno a heslo, které řidič obdrží od zaměstnavatele. V současné době není uvažována možnost obnovení zapomenutého hesla přímo v aplikaci. Ve vnitropodnikových aplikacích taková možnost nebývá zvykem, navíc se nenabízí bezpečná možnost řešení zapomenutí hesla. Zařízení nemusí být vázáno na řidiče, tudíž nemáme příliš možností, jak zapomenuté heslo automatizovaně a zároveň bezpečně řešit. Pokud se v reálném nasazení ukáže tato možnost jako nezbytná, bude provedena další analýza možných řešení.

2.5.3 Detail destinace

Na obrazovce detailu destinace je zobrazena adresa destinace, celková hotovost za všechny dodací listy a kontaktní osoba, pokud je totožná pro všechny dodací listy. Dalším detailem, vyplněným až řidičovou aktivitou, je čas příjezdu a odjezdu. Pomocí dialogu lze automaticky předvyplněný čas poupravit, pokud se nehodí vyplnění ihned po příjezdu, respektive při odjezdu. Upravitelnost vyplněného času je omezená, čas nelze posunout dopředu a zpět lze posunout o hodinu. Tento čas byl zvolen pro možnost vyplnění požadovaných informací do aplikace až po uskutečnění vykládky, která má dle analýzy maximální běžnou dobu jednu hodinu. Zbytek obrazovky zaplňuje seznam dodacích listů náležících dané lokaci. Na horním panelu je k dispozici možnost uzavřít hromadně všechny neuzavřené dodací listy, detaily k uzavření se nastaví na



Obrázek 2.3: Detail destinace

následně zobrazeném dialogu.

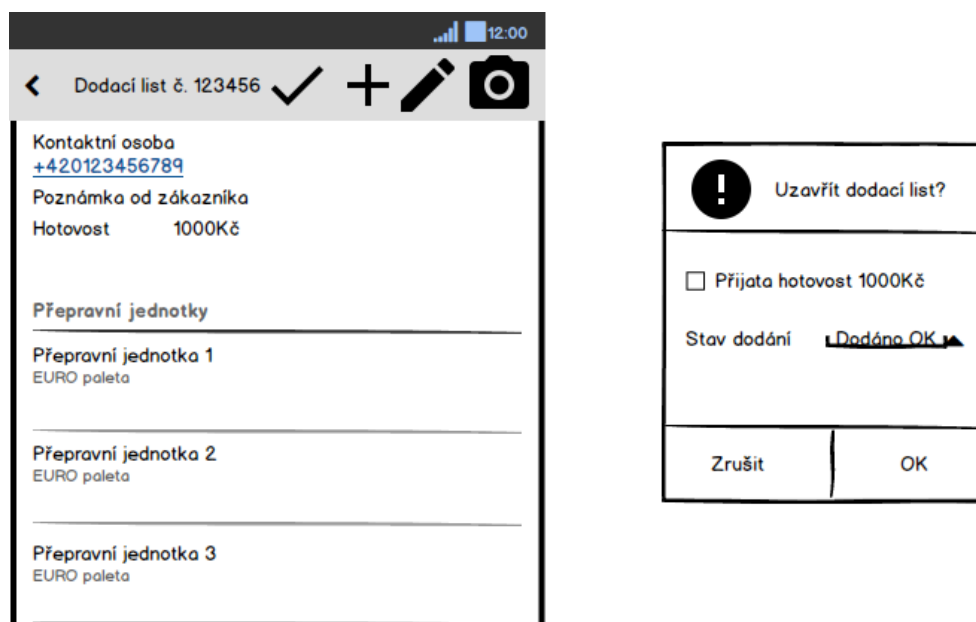
2.5.4 Detail dodacího listu

Obrazovka dodacího listu nabízí obdobné informace, jako detail destinace, s rozdílem, že jsou data vztažena ke konkrétnímu dodacímu listu. Ústředním obsahem je v tomto detailu seznam obsahující přepravní jednotky spolu s jejich obaly. V horním menu lze vybrat vytvoření fotografie k danému dodacímu listu, přidání podpisu od zákazníka k ověření převzetí zboží, případně editovat směnu obalů. Směna obalů je přidružena k jednotlivým dodacím listům pro sledování celkového stavu obalů vůči daným zákazníkům.

2.5.5 Směna přepravních obalů

V rámci obrazovky směny obalů se nachází seznam směn obalů tvořený kartami obsahující pole pro počet předaných obalů, výběr typu obalu z předdefinovaného seznamu a výběr, zda byl obal předán nebo obdržen. Návrh je zobrazen v příloze D.1.

2. NÁVRH



Obrázek 2.4: Detail dodacího listu

2.5.6 Tankování

Jedním z požadavků na aplikaci je i záznam provozu vozidla. V rámci něj je třeba zaznamenat detaily tankování. Pole obrazovky zůstávají obdobná, jako na současných papírových dokumentech. Konkrétně se jedná o aktuální stav tachografu vozidla, počet natankovaných litrů, čas tankování a typ paliva. Pokud má typ paliva definovaný i svůj podtyp, je následně možné vybrat i ten. Například je vyplněno, zda je tankována běžná, či bio nafta. Návrh je zobrazen v příloze D.2.

2.5.7 Neplánované zastavení

Obrazovka pro neplánované zastavení obsahuje pouze výběr důvodu zastavení, možnost potvrdit příjezd a následně odjezd. Neplánované zastavení je obvykle využito jednou v průběhu jízdy, pro evidenci pauzy. Návrh je zobrazen v příloze D.4.

2.5.8 Podmínky přepravy zboží

Zejména pro vozidla přepravující zboží bez obalů, tedy přímo na ložné ploše, je důležité splnit rozšířené podmínky přepravy. Jedná se typicky o normy GMP pro dopravu [31], případně o doplňující detaily navržené dispečerem. Na obrazovce se nachází seznam pokynů k vykonání spolu s možností jejich potvrzení a pořízení fotografie pro důkaz o provedení. Návrh je zobrazen v příloze D.3.

Realizace

3.1 Vývojové nástroje

3.1.1 Vývojové prostředí

Aplikace pro OS Android byla vyvíjena ve vývojovém prostředí Android Studio. Toto prostředí je přizpůsobeno právě pro vývoj aplikací pro OS Android. Umožňuje generovat šablony kódu pro často využívané prvky, upravovat grafické rozhraní aplikace pomocí WYSIWYG editoru, využívat verzovací systém přímo z prostředí IDE a další funkce. Toto prostředí je oficiálním nástrojem pro vývoj Android aplikací, odvozeným z vývojového prostředí IntelliJ IDEA. [32]

Pro vývoj serverové aplikace bylo použito integrované vývojové prostředí IntelliJ IDEA [33]. Dokumentaci k oběma aplikacím lze generovat přímo z prostředí IntelliJ IDEA, případně Android Studio pomocí *Tools* → *Generate JavaDoc*.

3.1.2 Verzovací systém

Zdrojový kód je verzován z důvodu možnosti vývoje na více zařízeních, dohledání změn, které způsobily různé problémy a možnost vytváření vývojových větví. V rámci projektu byl použit verzovací systém Git [34], konkrétně privátní repozitář poskytovaný ve studentském programu GitHub Education hostovaný na portálu GitHub [35].

3.2 Použité knihovny

3.2.1 Android Support Library

Knihovna vytvářející kompatibilní rozhraní funkcí a tříd na starších verzích systému Android, než pro které vyvinuty. V aplikaci je využita zejména pro podporu material designu nativně podporovaného až v Androidu 5. [36]

3.2.2 Google Play Services

Knihovna obsahující rozhraní pro komunikaci s Google Play službou běžící na pozadí. Poskytuje sjednocené rozhraní pro komunikaci aplikací se systémovými službami. V důsledku pomáhá jak jednoduchosti vytváření programů záviselých na poskytovaných funkcích, ale i úspoře baterie. V aplikaci je využita pro rozpoznání aktuální polohy. [37]

3.2.3 RecyclerView

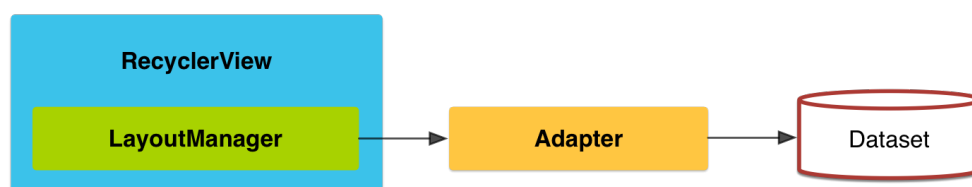
Pro plynulé posouvání seznamů v systému Androidu je doporučeno využívat návrhový vzor ViewHolder pro recyklaci polí využitých v zobrazení seznamu. Recyklace položek v seznamu ListView musela být v minulosti obstarávána programátorem, novější implementace seznamu RecyclerView ale již automaticky položky seznamu recykluje. Zajišťuje, že je do viditelných položek seznamu a několika položek nad a pod viditelným rozsahem zobrazen obsah z dané kolekce objektů. RecyclerView usnadňuje i nastavení dalších vlastností seznamu, například animací. Pro zobrazení vlastních objektů je třeba rozšířit třídu RecyclerView.Adapter a ViewHolder, který mapuje prvky objektu na definici zobrazení. Způsob fungování RecyclerView je zobrazen na diagramu 3.1. Pro funkčnost RecyclerView i na starších systémech než Android 5 je využita implementace z knihovny Support Library. [38]

3.2.4 CardView

Podpora CardView umožňuje jednoduchým způsobem vytvářet vzhledově konzistentní karty, ve kterých je možné zobrazovat obsah. V implementovaném prototypu je právě CardView využit pro zobrazení směny obalů. Ukázka využití je k nahlédnutí v příloze na obrázku E.6. [39]

3.2.5 Retrofit

Knihovna pro spolupráci s REST API. Umožňuje snadno definovat rozhraní pro práci s REST službou pomocí Java interfaců. Data lze stahovat jak pomocí synchronních metod, tak pomocí callbacků, které jsou volány po asynchronním stažení dat. Data mohou být stažena jako prostý text, případně rovnou dese-



Obrázek 3.1: RecyclerView [39]

realizována do poskytnutých tříd. Pro parsování JSONu je v mobilní aplikaci z důvodu integrace s Retrofit knihovnou a snadného použití využita knihovna Gson [40].

```
public interface StopApiService {
    @POST(value = "/ride/{rideId}/stop")
    int createStop(@Body Stop stop, @Path("rideId") int rideId);
}
```

Ukázka 3.1: Příklad definice REST rozhraní pomocí knihovny Retrofit

3.2.6 AndroidAnnotations

Knihovna poskytující anotace sloužící k zpřehlednění napsaného kódu. Umožňuje jednoduše mapovat proměnné na prvky rozhraní, vytvářet metody běžící na pozadí a vkládat závislosti. Knihovna při kompilaci programu generuje kód, který zajistí funkčnost aplikace pod OS Android. Nevýhodou tohoto přístupu je nutnost v konfiguraci aktivit používat knihovnou generované aktivity, které jsou dostupné až po následující kompilaci. Knihovna generuje kód již při kompilaci pravděpodobně z důvodu zpětné kompatibility. V době před Androidem 4 totiž bylo užívání anotací v Androidu pomalé vinou chyby v systému [41]. Z hlediska přehlednosti kódu knihovna do jisté míry nahrazuje podporu Javy 8 (konkrétně lambda výrazů) na platformě Android. V následujících ukázkách kódu nabízím porovnání mezi standardním Android kódem v Javě 7, ekvivalent v Javě 8 (pokud by ji platforma podporovala) a AndroidAnnotations.

```
runOnUiThread(new Runnable() {
    @Override
    public void run() {
        mAdapter.notifyDataSetChanged();
    }
});
```

Ukázka 3.2: Java 7

```
runOnUiThread(() -> { mAdapter.notifyDataSetChanged(); });
```

Ukázka 3.3: Java 8

```
@UiThread
void notifyDataSetChanged() {
    mAdapter.notifyDataSetChanged();
}
```

Ukázka 3.4: AndroidAnnotations

3. REALIZACE

Porovnání mapování proměnných na prvky rozhraní

```
RecyclerView mDestinationView = (RecyclerView) findViewById(R.id.  
    destination_list);
```

Ukázka 3.5: Java 7

```
@ViewById(R.id.destination_list)  
RecyclerView mDestinationView;
```

Ukázka 3.6: AndroidAnnotations

3.2.7 Crouton

Pokud chce vývojář zobrazovat upozornění uvnitř aplikace, nemá příliš mnoho možností vestavěných přímo v systému. Notifikační lišta se pro taková upozornění nehodí, slouží zejména pro oznámení při vypnuté aplikaci. Zbývají tedy tzv. toast oznámení, která ovšem nevypadají dobře a nemají možnost vytvořit na nich tlačítko. Z toho důvodu byla využita knihovna Crouton, která dovoluje vytvořit vlastní oznámení. Ty se zobrazují přímo v aplikaci, s možností definovat vlastní možnosti zobrazení oznámení, délku i přidat tlačítka a jiné prvky rozhraní. Ukázka je k nahlédnutí na obrázku E.8. [42]

3.2.8 ActiveAndroid

Původním záměrem bylo nevyužívat databázi v mobilní aplikaci. V průběhu vývoje se ale tento koncept ukázal jako obtížně splnitelný. Data byla původně ukládána do tzv. Shared Preferences, které umožňují uchovat běžné datové typy i po vypnutí aplikace. Tento přístup vyžadoval serializaci či deserializaci objektů při každém uložení, respektive načtení dat, protože rozhraní neumožňuje přímo ukládat objekty implementující rozhraní Serializable ani Parcelable. Data byla následně uložena jako řetězec ve formátu JSON. Práce s takto uloženými daty následně nebyla jednoduchá ani efektivní. Pro práci s kolekcemi objektů, které byly deserializovány, není v Javě 7 vhodné rozhraní. Až Java 8 nabízí tzv. Stream API, které přehledným a efektivním způsobem pracuje s kolekcemi objektů [43]. Pro práci s daty byla tedy nakonec zvolena konvenční cesta ve formě dotazů nad databází.

Pro pohodlnější práci s databází byla zvolena knihovna ActiveAndroid. Tato knihovna nabízí objektově relační mapování pro SQLite databázi využívanou v operačním systému Android. Nad objekty spravovanými knihovnou je možnost vytvářet jednoduché dotazy pomocí vestavěných funkcí. Pro složitější dotazy lze využít SQL dotazů přímo nad SQLite databází spravovanou knihovnou. Kromě ORM poskytuje i automatické vytváření databáze z definovaných objektů, cachování dat načtených z databáze, ale i možnost jednoduše definovat jakým způsobem (de)serializovat typy přímo nepodporované databází SQLite. [44]

3.2.9 Pen SDK

Knihovna umožňující využití Samsung S Pen k zadávání ručně psaného vstupu stylusem. Je využita dynamická verze knihovny, ke které je třeba navíc stáhnout aplikaci z Google Play, která zajistí spojení mezi aplikačním rozhraním Samsung zařízení a aplikací. Tato verze je využita z toho důvodu, že aplikace nebude využívána pouze na zařízeních Samsung, velikost výsledného balíčku aplikace se v důsledku zmenší o část, která by často ani nebyla využita. Ukázka využití je zobrazena na obrázku E.9. [45]

3.3 Nasazení

V současné době je prototyp mobilní aplikace distribuován pomocí instalačního souboru ve formátu apk. Aplikace prozatím není nasazena do oficiálního distribučního kanálu pro OS Android, Google Play. Pro běžného uživatele této služby aplikace nemá žádný účel, proto není zájem o veřejnou distribuci. V budoucnu však aplikace bude nasazena do prostředí Google Play formou uzavřeného testování, které tato služba umožňuje [46]. Tímto způsobem bude zajištěna jednoduchá distribuce aktualizací aplikace a zároveň omezený přístup k instalaci aplikace.

Serverová aplikace je v rámci prototypu distribuována ve formě jar souboru, který obsahuje jak samotnou aplikaci, tak webový server, tudíž je distribuce a nasazení velmi jednoduché. K funkčnosti serverové aplikace je na daném systému potřeba nainstalovat Java Runtime Environment a vytvořit PostgreSQL databázi. Tabulky v databázi jsou automaticky vytvořeny při spuštění aplikace pomocí Hibernate frameworku.

3.4 Budoucí vývoj

Vývoji aplikace se hodlám věnovat i nadále. Z navržených částí aplikace nebyl z důvodu celkového rozsahu implementován funkční požadavek *Specializované pokyny pro přepravu zboží*, ten plánuji doimplementovat. Další podněty k vylepšení jistě vyplynou z testování uživateli. V současné době se připravuje napojení externího systému na serverovou aplikaci poskytující data pro mobilní aplikaci, následně budou aplikaci testovat jednotky řidičů. Budoucí vývoj na mobilní aplikaci pravděpodobně bude zahrnovat i přizpůsobení grafického rozhraní pro tablet, vhodnost větší obrazovky se může ukázat jako důležitý faktor pro pohodlnou práci řidičů s určitým typem zboží či specifickými povinnostmi v průběhu jízdy. Další úpravy rozhraní jsou také žádoucí, zejména pro větší zdůraznění vzniklých problémů na nějaké destinaci (nepředání zásilky, nepřevzetí hotovosti). Možností je i větší spolupráce aplikace s aktuální polohou, pro upozornění řidiče o odchylení od stanovené trasy, případně pro

3. REALIZACE

zdůraznění, že by měly být informace o dodání vyplněny pouze v blízkosti destinace.

V pokročilé fázi vývoje bylo také zjištěno, že původní koncept webového rozhraní, který počítal s nasazením jednoho webového API pro každou firmu, která by měla zájem o využívání mobilní aplikace, není příliš ideální. Jeden řidič totiž může plnit zakázky pro různé firmy, pokud by každá z nich měla vlastní server s API, musela by být možnost v rámci aplikace nastavit, které API bude využíváno, případně pracovat s více API zároveň. Lepším přístupem je jedno serverové API, které bude umožňovat plnění dat jednotlivými firmami. Tento přístup bude vyžadovat lehkou změnu datového modelu, hlavní změna se ale promítne na serverové aplikaci. Pro možnost interakce s aplikacemi třetích stran je nutné implementovat rozhraní, pomocí kterého bude možné nahrávat jízdy pro řidiče na server. V průběhu vývoje byla tato funkce vnímána jako druhotná, hlavní důraz byl zaměřen na vývoj mobilní aplikace. Prototyp bude využit interně a data budou prozatím plněna přímo ze zadavatelova systému do databáze webové služby. Platforma Spring nicméně umožňuje snadnou rozšiřitelnost a současné API je vhodným základním kamenem pro další vývoj.

V rámci vylepšení API je v plánu také možnost změny jízd v jejich průběhu. Spolu s ní bude třeba implementovat push notifikace, které klientskou aplikaci informují o změně na serveru. K implementaci bude pravděpodobně využít Google Cloud Messaging [47].

Dalším aspektem, který je třeba v budoucnu vylepšit, je zabezpečení. Současný prototyp sice využívá OAuth2 ověření uživatele, ale pro jednoduchost nasazení zatím není využit protokol HTTPS, ale pouze nezabezpečený HTTP. Po nasazení na produkční server namísto prozatímního testovacího serveru bude aplikace přizpůsobena pro HTTPS a pořízen certifikát pro ověření certifikační autoritou.

Testování

4.1 Testování programátorem

V průběhu vývoje aplikace byly využity testy programátorem. Byly zaměřeny zejména na potenciálně problematická data a situace, které mohou nastat. Následně byla aplikace ručně procházena a pomocí logování či krokování aplikace bylo zjišťováno, zda nedochází k problémům. Problémy byly odhaleny zejména v serializaci a deserializaci dat a v práci s vlákny. V rámci průchodů kódu docházelo i k revizi kódu, která, často podpořena upozorněním od vývojového prostředí, odhalila zejména chyby ve stylu kódu, případně drobné překlepy v názvech proměnných, které nebránily kompilaci. Spolu s popsány testy, kontrolou kódu ze strany vývojového prostředí a kompilátoru byla nakonec pro otestování systému jako celku využita varianta na systémové testy, která byla provedena na scénářích popsanych v sekci 4.2 s cílem odhalit případné chyby.

4.2 Kognitivní průchod

Kognitivní průchod se zabývá vyhodnocením srozumitelnosti uživatelského rozhraní pomocí průchodu souboru úkolů [48]. V rámci průchodu byly zodpovídány následující otázky, body 1–3 byly procházeny na konceptu rozhraní, bod 4 na implementovaném prototypu.

1. Má uživatel důvod provést právě takovou akci?
2. Všimne si uživatel, že je dostupná správná akce?
3. Přiřadí uživatel správnou akci k úkolu, který chce vykonat?
4. Je uživateli srozumitelná odezva po vykonání správné akce?

4. TESTOVÁNÍ

Test byl proveden pro jednotlivé cíle uživatele, výsledky jsou popsány v následujících odstavcích. Vstupem testu je očekávaný průchod k dosažení cíle uživatele, následuje vyhodnocení testu.

- **Projít destinace, které budou vykonávány, spolu s doklady a přepravními jednotkami, které jsou v nich zařazeny**

Předpokládaný průchod od zobrazení destinací po zobrazení přepravních jednotek

1. Spuštění mobilní aplikace
2. Aplikace zobrazí seznam destinací
3. Výběr destinace
4. Aplikace zobrazí seznam dokladů
5. Výběr dokladu
6. Aplikace zobrazí přepravní jednotky

Uživatel začátečník si nemusí být jistý, zda jednotlivé položky vykonávají nějakou činnost. Tento problém je řešen následujícím způsobem. Aplikace uživatele upozorní při dotyku na oblast, která po klepnutí vykoná nějakou činnost. Konkrétní forma upozornění se liší dle verze systému. Po dotyku položky seznamu s destinací či dokladem je v Androidu 5 zobrazena animace měnící pozadí prvku, ve starších verzích se pouze změní pozadí položky.

- **Být navigován na příští destinaci**

1. Výběr příští destinace ze seznamu naplánovaných destinací otevře detail destinace
2. Klepnutí na adresu destinace spustí navigaci

V průběhu testu byl nalezen problém v otázce 2, ve chvíli klepnutí na adresu destinace. Uživatel nemusí být obeznámený s principy operačního systému Android, který v rámci svých guidelines umožňuje stisknutí prstem na kterémkoliv místě daného řádku pro vyvolání přidružené akce. Tyto akce jsou ovšem v prostředí systému natolik standardní (využívané jsou v systémovém kalendáři, SMS zprávách a dalších aplikacích), že je vhodné je používat. Uživatelé, kteří systém znají, toto chování již očekávají, případní uživatelé, kteří nejsou na systém Android zvyklí, mohou být zaškoleni.

- **Kontaktování zákazníka, kterému je zásilka doručována**

1. Výběr destinace ze seznamu naplánovaných destinací otevře detail destinace

2. Klepnutí na kontaktní osobu začne aplikace vytáčet číslo zákazníka

Obdobné jako v předchozím bodě.

- **Potvrdit předání zásilky**

1. Výběr destinace ze seznamu naplánovaných destinací otevře detail destinace
2. Výběr dodacího listu, pro který je vyžadováno potvrzení předání
3. Klepnutí na ikonu dokončení v action baru vyvolá dialog pro potvrzení předání

V testu nebyly zjištěny problémy.

- **Pořídít fotografii zásilky**

1. Výběr destinace ze seznamu naplánovaných destinací otevře detail destinace
2. Výběr dodacího listu, pro který je třeba vytvořit fotografii
3. Klepnutí na ikonu fotoaparátu v action baru spustí fotoaparát
4. Fotografie je vyfocena a potvrzeno její použití

V testu nebyly zjištěny problémy.

- **Potvrdit předání podpisem** Průchod je obdobný s předchozím, pouze je v action baru vybrána ikona podpisu, na další obrazovce je uživatel podepsán a podpis je potvrzen.

- **Vyplnit stav dodání zásilky, ať už provedené, či ukončené chybou**

1. Výchozím bodem je průchod pro potvrzení předání zásilky
2. Vyplnění stavu dodání a potvrzení tlačítkem OK

V testu nebyly zjištěny problémy.

- **Potvrdit přijetí či nepřijetí hotovosti pro danou zakázku** Průchod je shodný s předchozím, s výjimkou vyplnění přijetí či nepřijetí hotovosti.

- **Zapsání předání přepravních obalů**

1. Výběr destinace ze seznamu naplánovaných destinací otevře detail destinace
2. Výběr dodacího listu, pro který se uplatňuje směna obalů

4. TESTOVÁNÍ

3. Klepnutí na ikonu přidání obalu v action baru otevře obrazovku směny obalů
4. Klepnutí na ikonu + umožňuje přidat novou směnu obalů
5. V rámci jedné směny obalů je vyplněno pole pro počet obalů, vybrán typ obalu a zda byly obaly předány či obdrženy

V testu nebyly zjištěny problémy.

- **Potvrdit pokyny pro naložení**

1. Výběr destinace ze seznamu naplánovaných destinací otevře detail destinace
2. Klepnutí na zobrazení detailů přepravy otevře obrazovku s podmínkami přepravy
3. Jsou zobrazeny podmínky přepravy a lze je potvrdit pomocí tlačítka OK v horní liště

Byl zjištěn problém v bodě 1, podmínky přepravy je obvykle třeba splnit s předstihem, tedy již po předcházejícím vyložení zboží. V tuto chvíli uživatel ale nemá informaci o tom, že následující destinace obsahuje podmínky pro přepravu. Řešením je zobrazení podmínek přepravy ihned po dokončení předchozí destinace, zároveň budou ve zkratce zobrazeny i na seznamu destinací.

- **Zaznamenat čerpání paliv**

1. Na hlavní obrazovce uživatel vybere ikonu tankování v action baru
2. Vyplnění detailů tankování a potvrzení ikonou OK v action baru

V testu nebyly zjištěny problémy.

- **Zaznamenat neplánovanou zastávku**

1. Na hlavní obrazovce uživatel vybere v action baru ikonu + k přidání nové zastávky
2. Stisknutím tlačítka začátek v action baru zaznamená začátek zastávky, z předdefinovaného seznamu vybere důvod zastávky
3. Při ukončení zastávky v action baru vybere tlačítko odjezd

V testu nebyly zjištěny problémy.

4.3 Heuristické vyhodnocení

Na artefaktu bylo provedeno heuristické vyhodnocení. Pro body 1–3 byly jako artefakt použity koncepty rozhraní, zbytek byl z důvodu povahy otázek otestován na prototypu. Test byl proveden na základě Nielsenových heuristik [49], konkrétní podoba otázek pochází ze zdroje [50], chybějící body jsou volně přeloženy s přihlédnutím k mobilní platformě Android.

1. Jsou názvy, popisky a jinak používané termíny v souladu s terminologií cílové skupiny? *Match between system and the real world*

Ano, názvy vychází z terminologie používané v současných dokumentech.

2. Poskytuje každá akce zpětnou vazbu srozumitelnou pro uživatele? *Visibility of system status*

Ano, akce vyvolá animaci zvýrazňující zvolenou činnost. Následuje animace přechodu na jinou aktivitu, případně doprovázená systémovým progress barem.

3. Ví uživatel vždycky, kde se nachází? Může se uživatel dostat z každé situace? Obejde se takové opuštění bez nutnosti opakovat dlouhou sekvenci akcí? Je použití tlačítka ← správné? *User control and freedom*

Ano, navigace se vždy nachází v horní části obrazovky, v podobě horní lišty, action baru. Uživatel se může dostat z každé situace. V některých případech je nutné použít dlouhou sekvenci tlačítka ←. V následujících bodech jsou popsány situace, kde problém může nastat.

- **Navigace** Po klepnutí na destinaci se otevře aplikace Google Maps s přednastavenou destinací pro navigaci. Aplikace je pouze spuštěna s parametry pozice aktuální destinace, tudíž nemáme příliš kontrolu nad tím, jak se bude chovat. Pro uživatele neznalé standardních přístupů k rozhraní může dojít k tomu, že se uživatel nebude umět dostat z nastalé situace. Nicméně existují hned dvě možnosti, jak se ze situace dostat, několikanásobné zmáčknutí tlačítka zpět, případně využití tlačítka pro multitasking k navrácení do aplikace pro řidiče. Tento problém by bylo vhodné řešit ve finální aplikaci buď pomocí dialogů s nápovědou při prvních spuštění, případně zaškolením řidičů.
- **Hluboké zanoření uvnitř aplikace** Na diagramu D.5 je zobrazena posloupnost akcí potřebná k tomu, aby se uživatel dostal na určitou obrazovku. V případě velmi zanořených akcí může být náročnější dostat se zpět na úvodní obrazovku. Řešením by bylo dát uživateli k dispozici tlačítko pro návrat na úvodní obrazovku,

nicméně akce byly navrženy tak, že pokud se k nim uživatel potřebuje dostat, musí je obvykle postupně cestou zpět na úvodní obrazovku potvrdit, případně doplnit o různé detaily. Částečným řešením je zavedení možnosti změny stavu dodacího listu již na detailu destinace. Tato funkcionality byla implementována a je použita, pokud nejsou využity obaly u přepravovaného zboží. Jedná se tedy zejména o zboží přepravované přímo ve vozidle. V takovém případě se tomuto problému aplikace vyhne.

4. **Jsou všechny objekty, akce a jiné prvky vidět, kdykoliv je jich potřeba, aniž by si je uživatel musel pamatovat?** *Recognition rather than recall*

Ano, všechny funkce jsou vidět, když je jich třeba, za předpokladu, že uživatel plní podmínky užívání aplikace, tedy potvrzuje vykonané akce v tu dobu, kdy se reálně stanou. Jedinou výjimkou je změna času příjezdu a odjezdu. Cílem je, aby tyto časy zůstaly vyplněné automaticky, tzn. aby příjezd i odjezd byl potvrzen v tu chvíli, kdy k němu dochází. Tyto funkce jsou mapovány na dlouhé stisknutí času příjezdu, respektive odjezdu.

5. **Jsou všechny akce, uspořádání a význam konzistentní s očekáváním cílové skupiny (zvyky odjinud...)?** *Consistency and standards*

Ano, rozhraní je konzistentní s prostředím operačního systému Android. Jsou využity standardní ovládací prvky operačního systému. Vzhled je přizpůsoben funkcím. Grafika je použita zejména ve formě ikon, které slouží pro rychlou orientaci v obsahu na displeji.

6. **Je plocha zobrazení využita přiměřeně (vzhledem k celkovému vzhledu)?** *Aesthetic and minimalist design*

Ano. Na obrazovce detailu destinace by bylo teoreticky vhodnější při posunu seznamu dodacích listů skrýt detaily destinace a zobrazit pouze dodací listy. Na druhou stranu to může vést ke zmatení uživatele, který si není jistý, jak se vrátit k detailům destinace. Bylo by třeba provést další testy, které zjistí, kolik dodacích listů je obvyklých pro jednu destinaci, zda by takový posun vůbec měl smysl. V prototypu vytvořeném v rámci této práce posun podporuje pouze samotný seznam s dodacími listy, na základě reálných dat a uživatelů, na kterých bude aplikace v budoucnu testována, se rozhodne, který přístup je vhodnější.

7. **Snaží se aplikace zabránit chybám uživatele?** *Error prevention*

Ano. Aplikace nutí uživatele zadat povinné vstupní údaje. V případě, že je nezadá, nemůže potvrdit danou akci.

8. **Poskytuje aplikace „urychlovače“ pro expertní uživatele?** *Flexibility and efficiency of Use*

Ne. Aplikace pracuje se všemi uživateli stejným způsobem.

9. **Pomáhá aplikace uživateli zjistit, jakou chybu udělal?** *Help users recognize, diagnose, and recover from errors*

Ano. Aplikace zobrazuje, jaký problém nastal a proč uživatel nemůže danou akci vykonat. Příkladem je nemožnost připojení k síti při přihlášení, nevyplněné povinné údaje a podobně.

10. **Je uživateli poskytnuta nápověda pro nejasné části aplikace?** *Help and documentation*

Ano, nicméně jedná se zejména o nápovědy k chybám, viz předchozí bod.

Závěr

Cílem této práce bylo navrhnout a implementovat prototyp mobilní aplikace pro řidiče přepravních služeb, který by umožňoval procházení zadaných zakázek, potvrzování dodávek zboží, záznam provozu vozidla a funkce s těmito spojené.

Výsledkem práce je funkční prototyp aplikace, který umožňuje digitalizovat pracovní úkony řidiče přepravních služeb. Aplikace slouží jako průvodce řidiče jeho jízdou, s možností zobrazení informací o nadcházejících i proběhlých událostech na jednom místě, namísto listování mnoha dokumenty. Využitelná je řidiči, kteří již mají zkušenosti s platformou Android, pro její ovládání jsou využívány standardní postupy známé z jiných aplikací na tomto systému. Pro řidiče bez zkušeností s danou platformou bude třeba zajistit školení.

Funkce aplikace zabývající se digitalizací současných stazek, do kterých je zaznamenáván pracovní výkon řidiče, jsou dle mého názoru využitelné v praxi v aktuální podobě, případně s drobnými úpravami na základě testů v reálném provozu. Současný prototyp umožňuje vyplnění stazek v přehlednějším prostředí, než nabízely původní komplikované papírové dokumenty. Navíc odpadá nutnost ručně je přepisovat do elektronické podoby. Každý přepis a kontrola zabraly několik minut času dalšího zaměstnance, pro větší počet řidičů takovou úsporu času firma jistě ocení.

Úplné nahrazení papírových dokladů je však dlouhodobým procesem závislým na typu firmy, která bude řešení s mobilní aplikací využívat. Zejména pokud často interaguje s jinými firmami, které stále papírové dodací listy vyžadují, nemůže je nijak jednoduše z procesu vyřadit. Mé řešení by mělo zejména posloužit k digitalizaci vnitrofiremních procesů, jako jsou právě zmíněné stazky, stavy dodání či záznamy o směně obalů. Tento cíl dle mého názoru splňuje a je vhodným stavebním kamenem pro další vývoj, který bude nadále upravovat řešení na míru potřebám jednotlivých firem.

Použité zdroje

- [1] Microlise Limited: Proof of Delivery, Total visibility of delivery operations [online]. 2015, [cit. 21.3.2015]. Dostupné z: <http://microlise.com/products/proof-of-delivery/>
- [2] DHL International GmbH: SmartPOD (Proof of Delivery) app [online]. [cit. 21.3.2015]. Dostupné z: http://www.dhl.com/en/logistics/supply_chain_solutions/what_we_do/smart_pod.html
- [3] Microlise Limited: Proof of Delivery, Improved accuracy & reduced administration [online]. Brožura, 2015, [cit. 21.3.2015]. Dostupné z: <http://marketing.microlise.com/acton/attachment/13299/f-0008/1/-/-/-/-/Microlise-Proof-of-Delivery.pdf>
- [4] Grant, K.: Android's design guidelines: what are they, and why should you care? [online]. 2014, [cit. 21.3.2015]. Dostupné z: <http://www.androidauthority.com/android-design-guidelines-514918/>
- [5] Microlise Limited: Microlise SmartPOD [software]. 2014, [cit. 21.3.2015]. Dostupné z: <https://play.google.com/store/apps/details?id=com.microlise.smartpod&hl=cs>
- [6] DPD: Nové mobilní terminály pro společnost DPD [online]. 2013, [cit. 22.3.2015]. Dostupné z: http://www.dpd.com/cz/home/o_dpd/pro_media/tiskove_zpravy/archiv/archiv_2013/tiskove_zpravy_z_roku_2013/nove_mobilni_terminaly_pro_spolecnost_dpd
- [7] Český telekomunikační úřad: Veřejné širokopásmové mobilní sítě, Mapa pokrytí. 2013, [cit. 22.3.2015]. Dostupné z: <http://lte.ctu.cz/pokryti/>
- [8] GSMArena: EDGE (Enhanced Data for Global Evolution) [online]. [cit. 22.3.2015]. Dostupné z: <http://www.gsmarena.com/glossary.php3?term=edge>

- [9] Canvas Solutions, Inc.: Canvas [software]. [cit. 22.3.2015]. Dostupné z: <http://www.gocanvas.com/>
- [10] IDC Corporate USA: Smartphone OS Market Share, Q4 2014 [online]. 2014, [cit. 26.4.2015]. Dostupné z: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
- [11] Meier, R.: *Professional Android 4 application development*. Indianapolis, IN: Wiley/Wrox, 2012, ISBN 978-1-118-10227-5.
- [12] Google, Inc.: Android Developers [online]. [cit. 24.4.2015]. Dostupné z: <https://developer.android.com/index.html>
- [13] Google, Inc.: Platform Versions [online]. 2015, [cit. 24.4.2015]. Dostupné z: <https://developer.android.com/about/dashboards/index.html>
- [14] Wacom Technology Corporation: Wacom's Feel IT Technologies Sharpen the Edge of Samsung's Innovation in GALAXY Note [online]. [cit. 27.3.2015]. Dostupné z: <http://www.prnewswire.com/news-releases/wacoms-feel-it-technologies-sharpen-the-edge-of-samsungs-innovation-in-galaxy-note-132714288.html>
- [15] Mlejnek, J.: Softwarové inženýrství I, přednáška Integrace aplikací. 2015, [cit. 4.3.2015].
- [16] Rozlog, M.: REST and SOAP: When Should I Use Each (or Both)? [online]. 2010, [cit. 4.3.2015]. Dostupné z: <http://www.infoq.com/articles/rest-soap-when-to-use-each>
- [17] Pivotal Software, Inc.: Spring IO platform [software]. [cit. 6.3.2015]. Dostupné z: <http://platform.spring.io/platform/>
- [18] Pivotal Software, Inc.: Spring Boot [software]. [cit. 6.3.2015]. Dostupné z: <http://projects.spring.io/spring-boot/>
- [19] JSON: The Fat-Free Alternative to XML [online]. [cit. 6.3.2015]. Dostupné z: <http://www.json.org/xml.html>
- [20] FasterXML, LLC.: Jackson [software]. [cit. 1.5.2015]. Dostupné z: <https://github.com/FasterXML/jackson>
- [21] Hibernate Community: Hibernate ORM [software]. [cit. 25.4.2015]. Dostupné z: <http://hibernate.org/orm/>
- [22] OAuth 2.0 [online]. [cit. 26.4.2015]. Dostupné z: <http://oauth.net/2/>
- [23] Pivotal Software, Inc.: Spring Security OAuth [software]. [cit. 26.4.2015]. Dostupné z: <http://projects.spring.io/spring-security-oauth/>

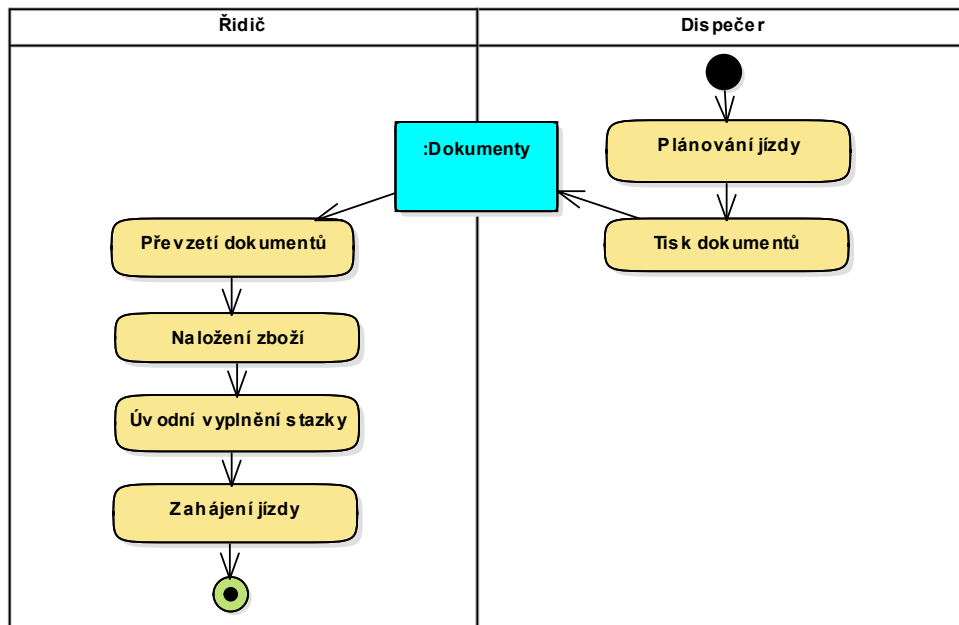
-
- [24] Rouse, M.: HTTPS (HTTP over SSL or HTTP Secure) definition. [cit. 2.5.2015]. Dostupné z: <http://searchsoftwarequality.techtarget.com/definition/HTTPS>
- [25] Provos, N.; Mazieres, D.: A Future-Adaptable Password Scheme. 1999. Dostupné z: https://www.usenix.org/legacy/events/usenix99/full_papers/provos/provos.pdf
- [26] Gradle, Inc.: Gradle [software]. [cit. 6.3.2015]. Dostupné z: <https://gradle.org/overview>
- [27] Groovy [software]. [cit. 6.3.2015]. Dostupné z: <http://groovy-lang.org/>
- [28] PostgreSQL Global Development Group: PostgreSQL [software]. [cit. 26.4.2015]. Dostupné z: <http://www.postgresql.org/>
- [29] PostgreSQL Global Development Group: Numeric types, Serial Types [online]. [cit. 26.3.2015]. Dostupné z: <http://www.postgresql.org/docs/9.1/static/datatype-numeric.html>
- [30] Balsamiq Studios, LLC: Balsamiq [software]. [cit. 24.4.2015]. Dostupné z: <https://balsamiq.com/>
- [31] GMP+ International B.V.: GMP+ B4 Transport [online]. Brožura, 2013, [cit. 24.4.2015]. Dostupné z: <https://gmplus.org/bestand/941/inlegvel-gmp-b4-20100528.pdf.ashx>
- [32] Google, Inc.: Android Studio [software]. [cit. 29.4.2015]. Dostupné z: <https://developer.android.com/sdk/index.html>
- [33] JetBrains s.r.o.: IntelliJ IDEA [software]. [cit. 29.4.2015]. Dostupné z: <https://www.jetbrains.com/idea/>
- [34] Git Community: Git [software]. [cit. 29.4.2015]. Dostupné z: <http://git-scm.com/>
- [35] GitHub, Inc.: Student Developer Pack [online]. [cit. 29.4.2015]. Dostupné z: <https://education.github.com/pack>
- [36] Google, Inc.: Maintaining Compatibility [online]. [cit. 30.4.2015]. Dostupné z: <https://developer.android.com/training/material/compatibility.html>
- [37] Google, Inc.: Google Play Services [online]. [cit. 30.4.2015]. Dostupné z: <https://developer.android.com/google/play-services/index.html>
- [38] Google, Inc.: Support Library Features [online]. [cit. 30.4.2015]. Dostupné z: <https://developer.android.com/tools/support-library/features.html>

- [39] Google, Inc.: Creating Lists and Cards [online]. [cit. 30.4.2015]. Dostupné z: <https://developer.android.com/training/material/lists-cards.html>
- [40] Google, Inc.: Gson [software]. [cit. 1.5.2015]. Dostupné z: <https://github.com/google/gson>
- [41] Issue 7811: Class.isAnnotationPresent()/getAnnotation()/getAnnotations() is slow [online]. [cit. 2.5.2015]. Dostupné z: <https://code.google.com/p/android/issues/detail?id=7811>
- [42] Weiss, B.: Crouton [software]. [cit. 30.4.2015]. Dostupné z: <https://github.com/keyboardsurfer/Crouton>
- [43] Hartveld, D.: Introduction to the Stream API [online]. [cit. 2.5.2015]. Dostupné z: <http://blog.hartveld.com/2013/03/jdk-8-33-stream-api.html>
- [44] Pardo, M.: ActiveAndroid [software]. [cit. 30.4.2015]. Dostupné z: <http://www.activeandroid.com/>
- [45] Samsung Electronics. Co. Ltd.: Pen SDK [software]. [cit. 25.4.2015]. Dostupné z: <http://developer.samsung.com/galaxy>
- [46] Google, Inc.: Use alpha/beta testing & staged rollouts [online]. [cit. 2.5.2015]. Dostupné z: <https://support.google.com/googleplay/android-developer/answer/3131213?hl=en>
- [47] Google, Inc.: Google Cloud Messaging for Android [online]. [cit. 2.5.2015]. Dostupné z: <https://developer.android.com/google/gcm/index.html>
- [48] Zhang, Z.: Cognitive Walkthrough [online]. [cit. 29.4.2015]. Dostupné z: <http://www.pages.drexel.edu/~zwz22/CognWalk.htm>
- [49] Nielsen, J.: 10 Usability Heuristics for User Interface Design [online]. 1995, [cit. 22.4.2015]. Dostupné z: <http://www.nngroup.com/articles/ten-usability-heuristics/>
- [50] Schmidt, J.: Tvorba uživatelského rozhraní, cvičení Analýza příkladů webových rozhraní – heuristické vyhodnocení. 2015, [cit. 22.4.2015].

Seznam použitých zkratek

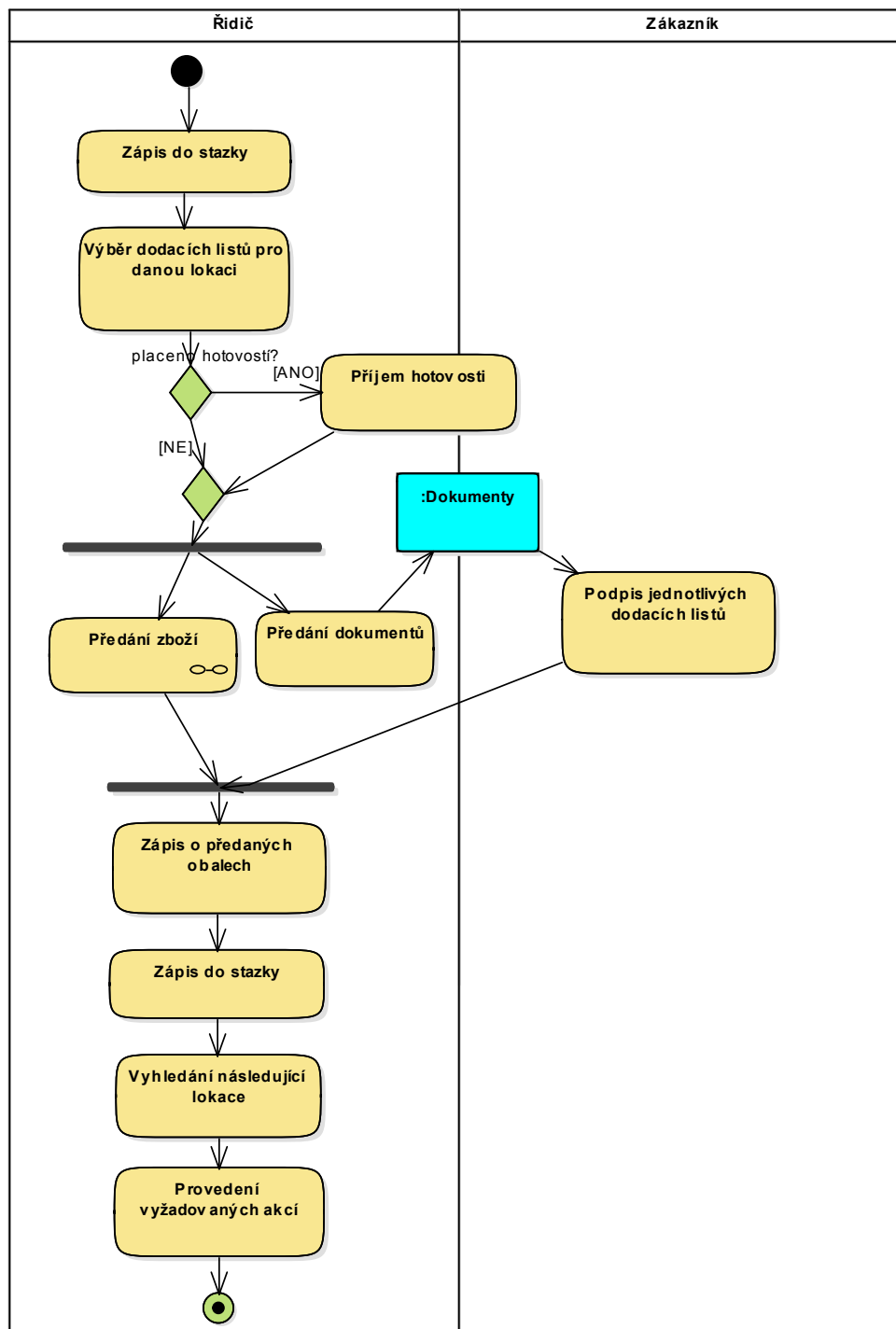
- API** Application Programming Interface
- EDGE** Enhanced Data rates for GSM Evolution
- GPRS** General Packet Radio Service
- GPS** Global Positioning System
- HTTP** Hypertext Transfer Protocol
- HTTPS** Hypertext Transfer Protocol Secure
- IDE** Integrated Development Environment
- JSON** JavaScript Object Notation
- LTE** Long-Term Evolution
- ORM** Object-relational mapping
- OS** Operating System
- REST** Representational State Transfer
- SDK** Software Development Kit
- SOAP** Simple Object Access protocol
- SQL** Structured Query Language
- SSL** Secure Sockets Layer
- TLS** Transport Layer Security
- WYSIWYG** What You See Is What You Get
- XML** Extensible markup language

Diagramy aktivit

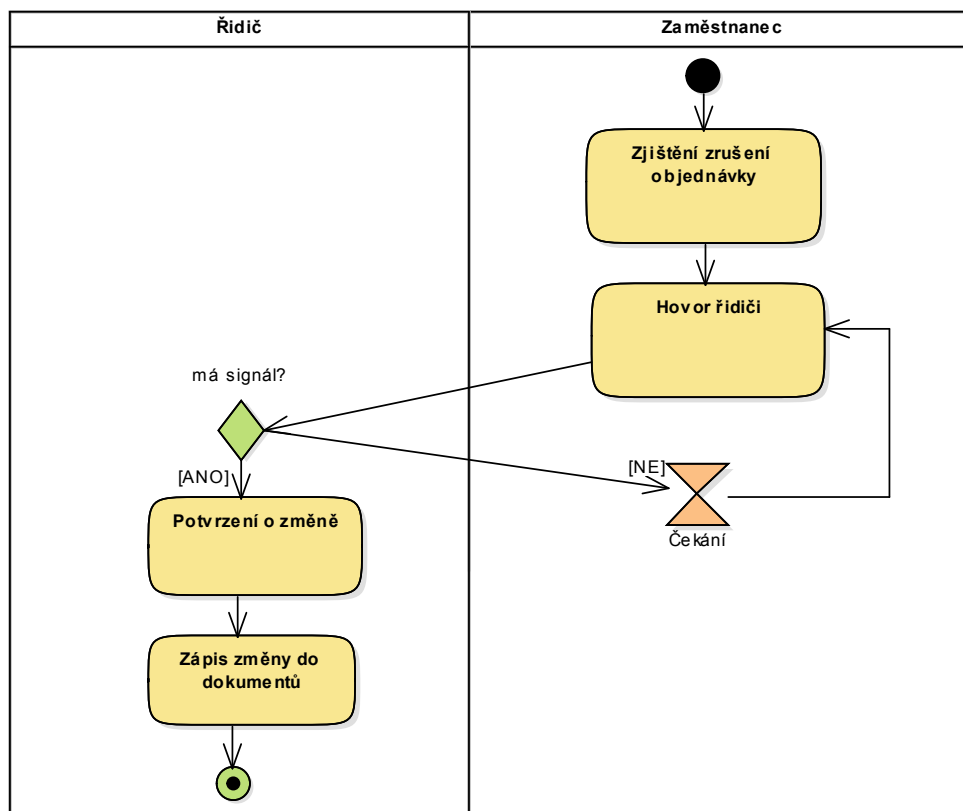


Obrázek B.1: Zahájení jízdy

B. DIAGRAMY AKTIVIT

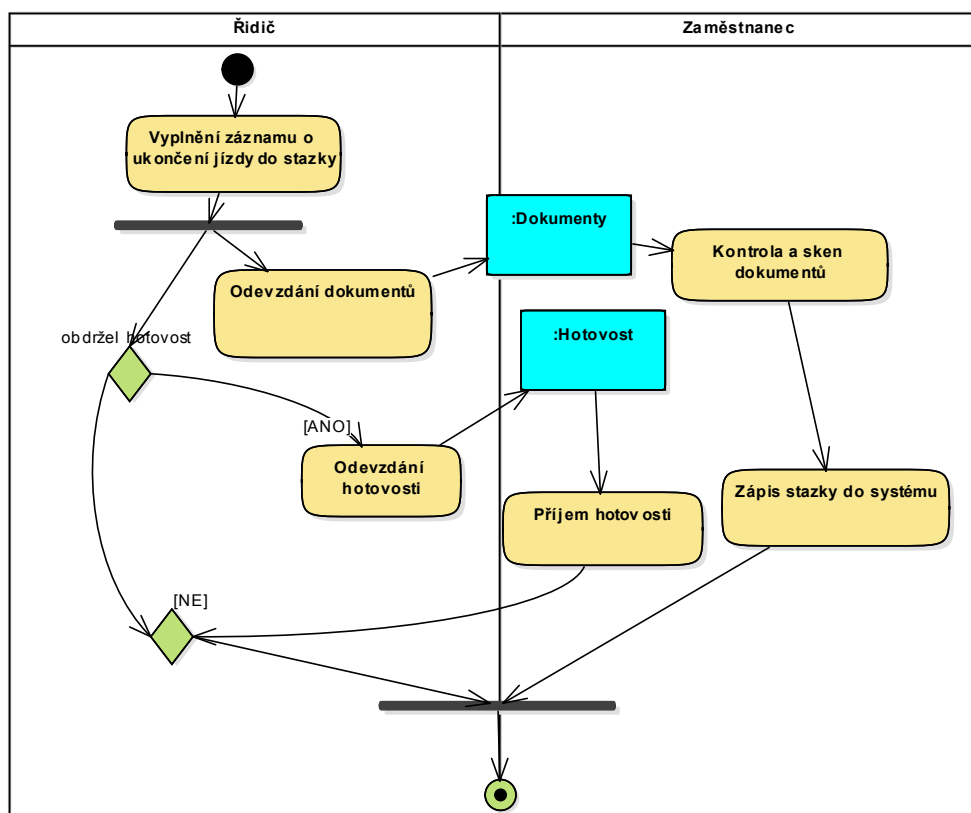


Obrázek B.2: Průběh jízdy

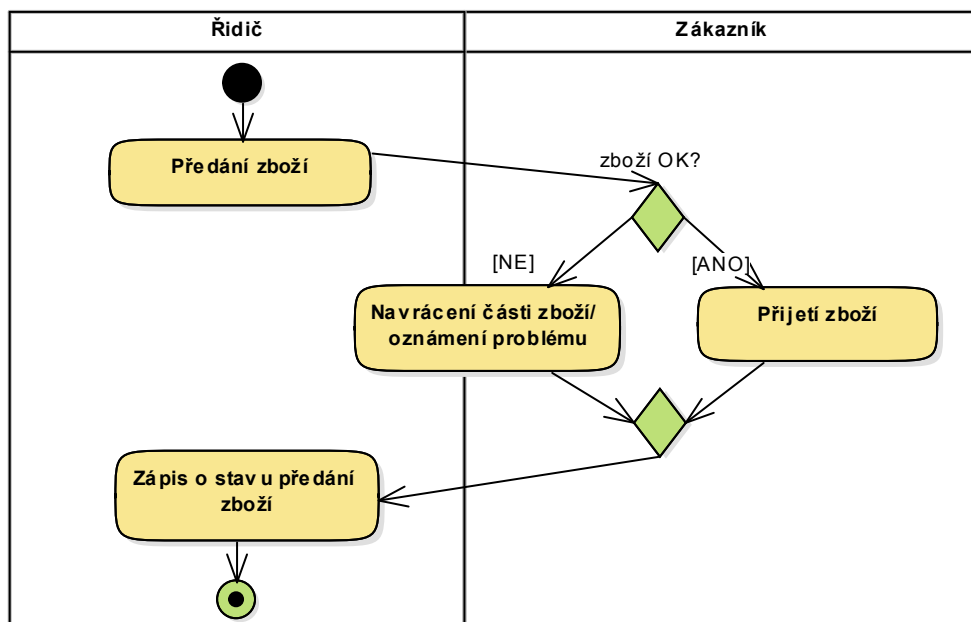


Obrázek B.3: Změny v průběhu jízdy

B. DIAGRAMY AKTIVIT



Obrázek B.4: Ukončení jízdy



Obrázek B.5: Předání zboží

Databázový model

Vzhledem k rozměrům se databázový model nachází na následující dvoustraně.

Návrh uživatelského rozhraní

Směna obalů

Počet

Typ obalu EURO paleta

Typ směny Přijem Výdej

Počet

Typ obalu EURO paleta

Typ směny Přijem Výdej

Obrázek D.1: Směna přepravních obalů

Tankování

Stav tachografu

Typ paliva Diesel

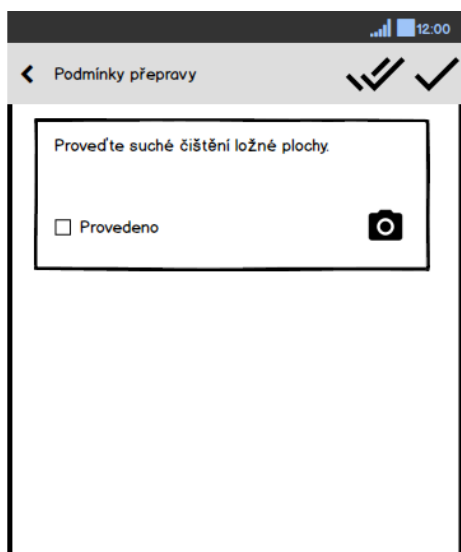
Podtyp paliva Bio

Natankováno

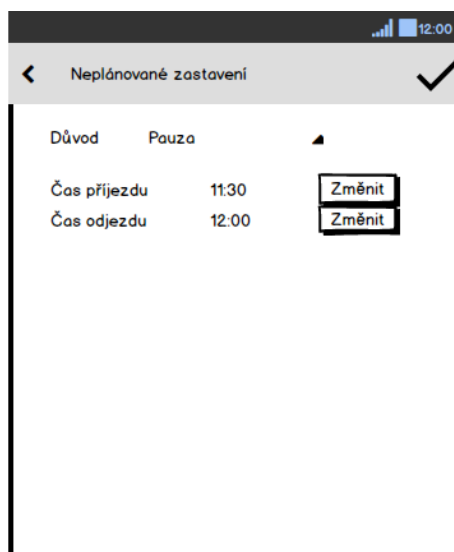
Čas 12:00

Obrázek D.2: Tankování

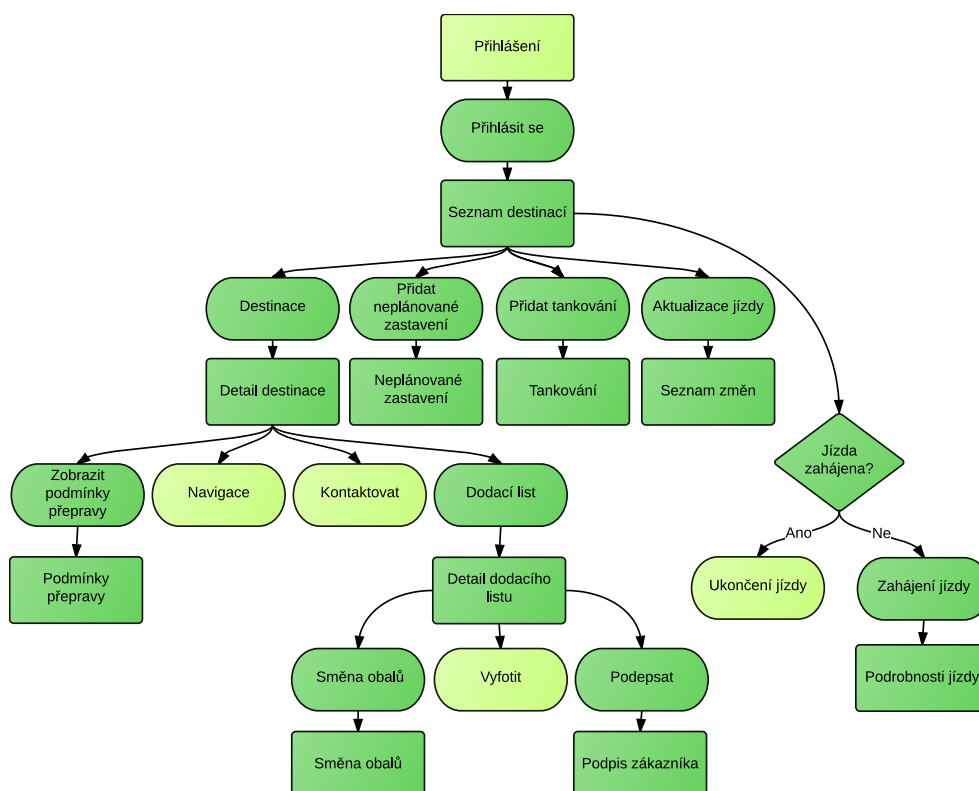
D. NÁVRH UŽIVATELSKÉHO ROZHRAŇÍ



Obrázek D.3: Podmínky přepravy zboží

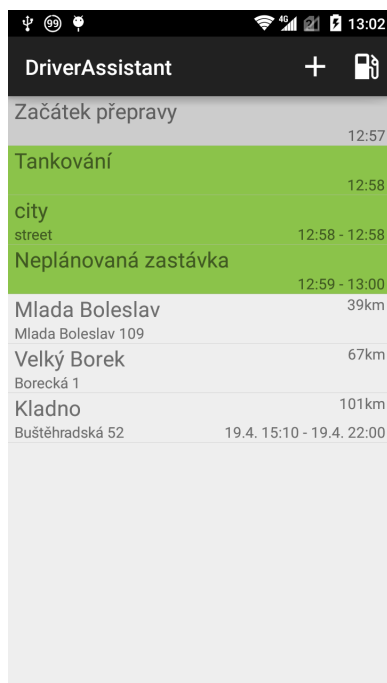


Obrázek D.4: Neplánované zastavení

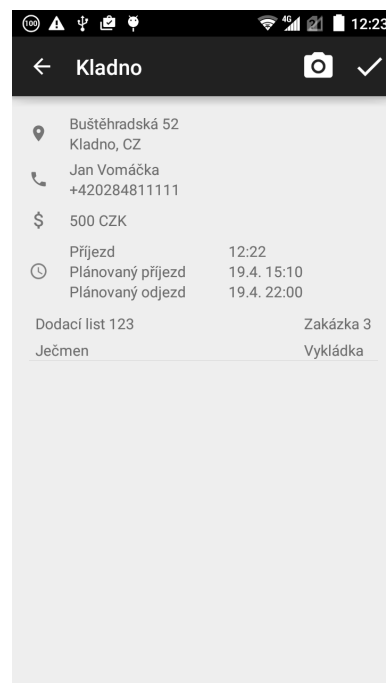


Obrázek D.5: Application Flow

Ukázka prototypu aplikace

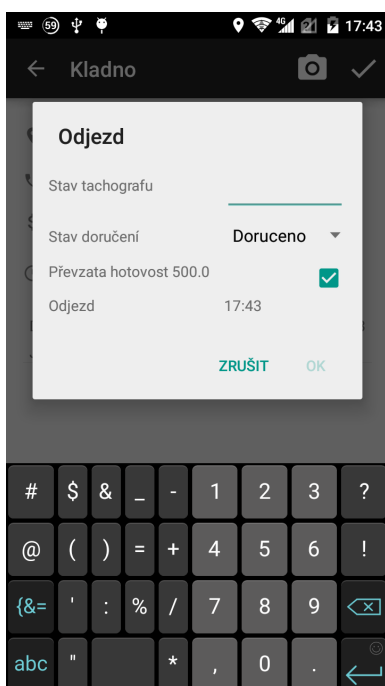


Obrázek E.1: Seznam destinací

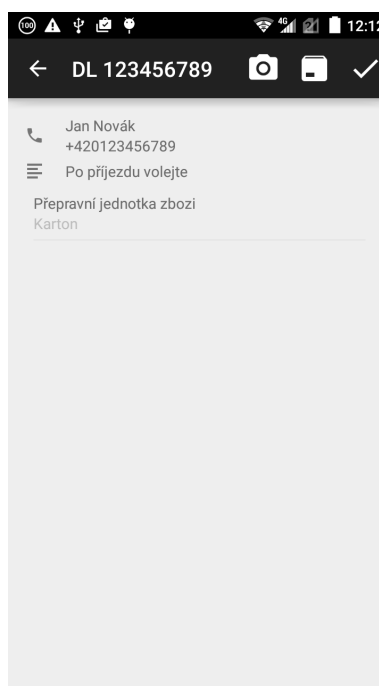


Obrázek E.2: Detail destinace

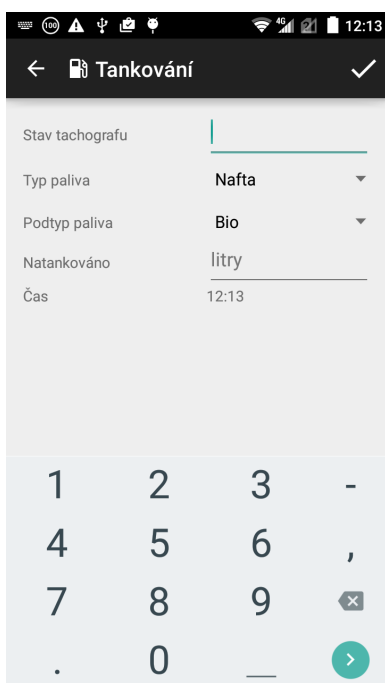
E. UKÁZKA PROTOTYPU APLIKACE



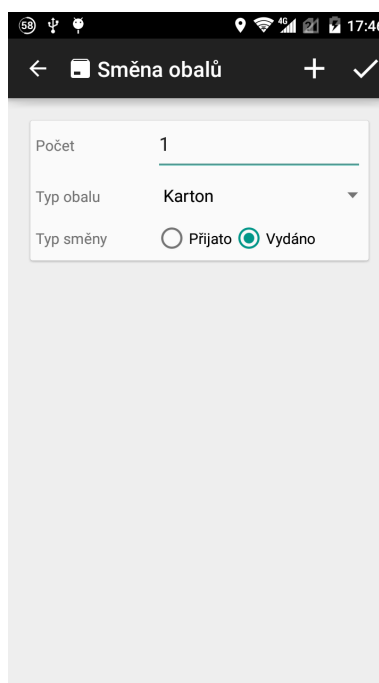
Obrázek E.3: Uzavření destinace



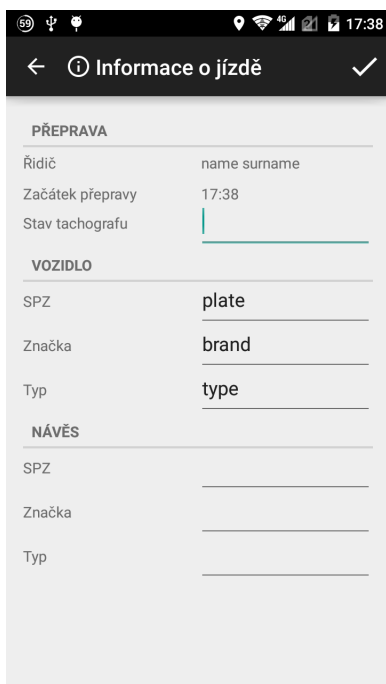
Obrázek E.4: Detail dodacího listu



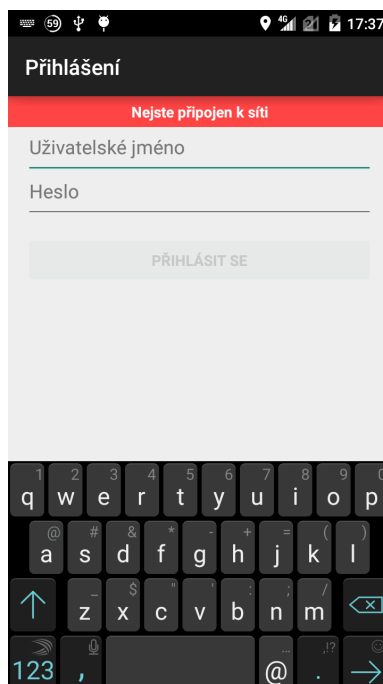
Obrázek E.5: Tankování



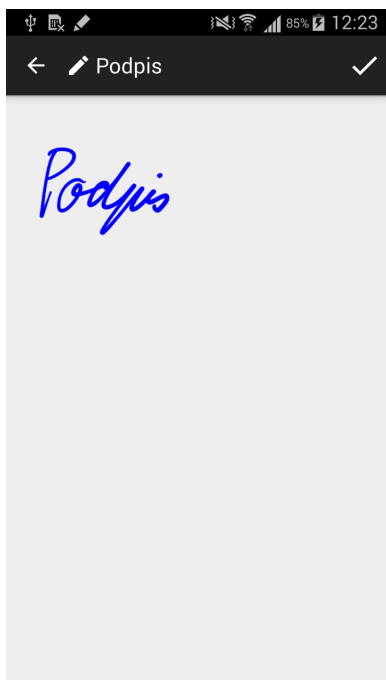
Obrázek E.6: Směna obalů



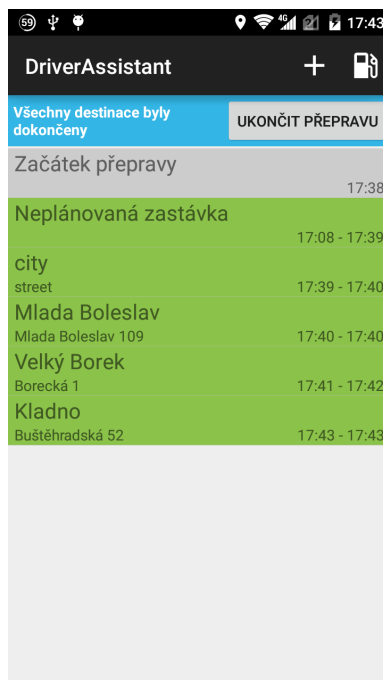
Obrázek E.7: Informace o jízdě



Obrázek E.8: Přihlášení



Obrázek E.9: Podpis zákazníka



Obrázek E.10: Dokončená jízda

Obsah přiloženého CD

readme.txt	stručný popis obsahu CD
exe	adresář se spustitelnou formou implementace
├ driver-assistant.apk	instalační balíček mobilní aplikace
├ driver-assistant-api.jar	spustitelný soubor serverové aplikace
└ inst.txt	instalační příručka
src		
├ impl	zdrojové kódy implementace
├ └ DriverAssistant	zdrojové kódy mobilní aplikace
├ └ DriverAssistantAPI	zdrojové kódy serverové aplikace
└ thesis	zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
text	text práce
└ thesis.pdf	text práce ve formátu PDF