



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	Evidence optické sít - iOS aplikace
Student:	Jakub ur ík
Vedoucí:	Ing. Tomáš Herout
Studijní program:	Informatika
Studijní obor:	Informa ní technologie
Katedra:	Katedra po íta ových systém
Platnost zadání:	Do konce letního semestru 2016/17

Pokyny pro vypracování

Vytvo te mobilní aplikaci pro za ízení Apple iPad, která bude vycházet z existující webové aplikace sloužící k evidenci optické sít a využije existujícího REST API této aplikace.

1. Prostudujte existující webovou aplikaci. Mobilní aplikace by m la poskytovat veškerou její základní funk nost.
2. Navrh te uživatelské rozhraní mobilní aplikace.
3. Navrh te další funkcionality využívající benefit mobilního za ízení - například polohové služby.
4. Zam te se na práci s mapou - hledání pravd podobného místa závady na základ selhání více spoj , návrh doporu ené cesty pro vytvo ení spojení na základ zadaných koncových bod .
5. V p ípad pot eby nových funkcí vhodn rozhodn te o zpracování dat na stran serveru i za ízení.
6. Zajist te bezpe nost dat aplikace a podporu protokolu IPv6.
7. Otestujte aplikaci.

Seznam odborné literatury

Dodá vedoucí práce.

L.S.

prof. Ing. Róbert Lórencz, CSc.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
d kan

V Praze dne 23. prosince 2015

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA POČÍTAČOVÝCH SYSTÉMŮ



Bakalářská práce

Evidence optické sítě - iOS aplikace

Jakub Ďurčík

Vedoucí práce: Ing. Tomáš Herout

17. května 2016

Poděkování

Na tomto místě bych především rád poděkoval vedoucímu práce Ing. Tomáši Heroutovi za jeho čas a cenné rady při vypracování této bakalářské práce. Dále bych rád poděkoval své rodině a přátelům za podporu během mého studia.

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. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 17. května 2016

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2016 Jakub Ďurčák. 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

Ďurčák, Jakub. *Evidence optické sítě - iOS aplikace*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

Tato práce se zabývá vývojem mobilní aplikace sloužící k evidenci optické sítě. Tato aplikace vychází z existující webové aplikace a doplňuje ji o další funkcionality. V první části je popsán význam jednotlivých evidovaných prvků a jejich vzájemný vztah. Následuje popis procesu analýzy, návrhu a implementace aplikace pro tablety iPad s operačním systémem iOS.

Klíčová slova mobilní aplikace, evidence optické sítě, iPad, iOS, Swift

Abstract

This thesis deals with development of a mobile application used for a registration of fiber network. This application is based on an existing web application and adds more features. In the first part, there is described meaning of each registered elements and relation of them. The next part describes process of analysis, design and implementation of application for iPad tablets with the operating system iOS.

Keywords mobile application, optical network register, iPad, iOS, Swift

Obsah

Úvod	1
1 Uvedení do problematiky	3
1.1 Evidované prvky	3
2 Cíle a požadavky	5
2.1 Funkční požadavky	5
2.2 Nefunkční požadavky	6
3 Analýza	7
3.1 Existující řešení	7
3.2 Platforma	7
4 Návrh	11
4.1 Architektura (MVC)	11
4.2 Doménový model	12
4.3 Návrh uživatelského rozhraní	13
5 Realizace	21
5.1 Použité externí knihovny	21
5.2 Uživatelské rozhraní	22
5.3 Datové třídy	23
5.4 Získávání dat ze serveru	24
5.5 Bezpečnost aplikace	24
5.6 Podpora IPv6	25
5.7 Přidané funkcionality	25
5.8 Zobrazení nejbližších POPů	27
6 Testování	29
6.1 Uživatelské testování	29

6.2 Crash reporting	29
Závěr	31
Literatura	33
A Seznam použitých zkratk	35
B Ukázky obrazovek aplikace	37
C Obsah příloženého flash disku	53

Seznam obrázků

3.1	Vrstvy systému iOS. Převezato z [6]	8
4.1	Model MVC. Převezato z [12]	11
4.2	Doménový model	13
4.3	Ukázka prvku Navigation Bar	14
4.4	Ukázka prvku Tab Bar	14
4.5	Zobrazení mapy POPů a kabelů	16
4.6	Pohled na koncový bod - organizér.	18
4.7	Výsledek návrhu doporučené trasy	19
B.1	Přihlašovací obrazovka	38
B.2	Zobrazení mapy POPů a kabelů	39
B.3	Zobrazení mapy POPů a kabelů s vyhledáváním	40
B.4	Detail POPu	41
B.5	Pohled na organizér	42
B.6	Pohled na ODF	43
B.7	Pohled na vývody	44
B.8	Trasa vlákna	45
B.9	Zobrazení seznamu kabelů	46
B.10	Detail kabelu	47
B.11	Nástroje	48
B.12	Návrh doporučené trasy	49
B.13	Výsledek návrhu doporučené trasy	50
B.14	Vyhledávání závady	51
B.15	Výsledek vyhledávání závady	52

Úvod

Společně s rozšiřujícím se množstvím multimediálního obsahu na internetu, ale i s přenosem hlasu a obrazu pomocí internetu, rostly požadavky na rychlost sítě jako takové. Tento požadavek splňují optické sítě, které nejenom že umožňují rychlejší tok dat, ale jsou schopny překonávat delší vzdálenosti než sítě metalické či bezdrátové. Díky tomu se staly optické sítě důležitou součástí všech páteřních sítí.

S masivním rozvojem optických sítí je zapotřebí kvalitní evidence. Obzvláště v prostředí větších společností jako jsou poskytovatelé připojení k internetu. Přístup techniků v terénu k těmto informacím je velmi důležitý.

Výsledkem této práce bude aplikace, která bude funkčně vycházet z existující webové aplikace sloužící k evidenci metropolitní sítě firmy TETA, s.r.o. v Ústí nad Labem a blízkém okolí.

Struktura práce

Tato práce dále pokračuje v následující struktuře:

V první části se práce věnuje popisu problematiky evidence optické sítě. Smysl této části je především v představení jednotlivých evidovaných prvků, tak aby s těmito pojmy bylo možné v dalších částech pracovat.

V druhé části následuje popis cílů práce a je detailněji přiblíženo její zadání, z čehož vyplývají funkční a nefunkční požadavky na aplikaci.

Třetí část se poté věnuje analýze současného řešení problému a cílové platformy, kde popisuje iPad z pohledu hardwaru, operační systém iOS a prostředky používané k vývoji aplikací pro zařízení se systémem iOS.

Ve čtvrté kapitole je nejprve přiblížena architektura aplikace, doménový model a velká část této kapitoly je věnována návrhu uživatelského rozhraní aplikace.

Pátá kapitola se zabývá samotnou realizací aplikace. Nejprve představením použitých knihoven, následně tvorbou uživatelského rozhraní, práci s daty, bezpečností aplikace a v závěru řešením několika problémů.

ÚVOD

Poslední kapitola se věnuje testování výsledné aplikace.

V závěru je shrnut obsah práce a vyhodnoceno splnění cílů včetně nastínění možného rozšíření aplikace do budoucnosti.

Uvedení do problematiky

Optické sítě na rozdíl od sítí metalických, se kterými přichází koncový uživatel do styku častěji, používají k přenosu dat světelné signály. Optický kabel je svazek jednotlivých optických vláken, optické vlákno se skládá ze skla tenkého jako lidský vlas, které je chráněno primární a sekundární ochranou, díky níž není křehké a dá se s ním manipulovat aniž by se poškodilo. [1]

Optická síť se skládá z jednotlivých uzlů a spojů mezi nimi. Spojení mezi jednotlivými koncovými uzly probíhá pomocí sváření jednotlivých vláken ve více optických kabelech na místech, kde se jednotlivé kabely potkávají. Tím pádem se běžně stává, že jedním kabelem vedou jednotlivá vlákna do různých míst.

1.1 Evidované prvky

V této části následuje popis jednotlivých nejdůležitějších prvků, které se evidují a vztah mezi nimi. Evidovány jsou pasivní prvky sítě, tedy prvky, které přenášené signály žádným způsobem nemodifikují, nevytváří je a ani je nepřijímají.

1.1.1 Optický kabel

Optický kabel je svazek jednotlivých optických vláken. Optická vlákna se dělí na single-mode (používá se zkratka SM, česky jednovidová) a multi-mode (zkratka MM, česky mnohavidová). Kabel může obsahovat vlákna jednoho typu, ale i kombinaci vláken obou typů.

1.1.2 POP

Označení POP je zkratkou anglického Point of Presence. Jedná se o pojem z oblasti telekomunikačních technologií, který obecně označuje místo, kde se potkávají dvě části sítě. Z pohledu optické sítě se jedná o místo, do kterého

přichází optické kabely. Optické kabely tedy vedou z jednoho POPu do druhého. POP může být skříň uvnitř budovy, venku stojící sloupek, ale i pod zemí umístěný vodotěsný rozvaděč. V POPu se nacházejí koncové body.

1.1.3 Koncový bod

Označení koncový bod se používá pro místo, ve kterém jsou ukončeny jednotlivá vlákna optického kabelu. Koncový bod může obsahovat organizér, ODF nebo organizér i ODF.

1.1.4 Organizér

Organizér slouží k uspořádání jednotlivých vláken a jejich svárů. Organizér obsahuje kazety, ve kterých jsou uloženy sváry jednotlivých vláken a zároveň je zde umístěna rezervní délka vlákna. Umístění sváru do kazety usnadňuje orientaci ve svárech a zároveň zvyšuje ochranu sváru.

1.1.5 ODF

ODF je zkratkou anglického Optical Distribution Frame, v češtině se používá označení optický rozvaděč a slouží k ukončení optických vláken. Optická vlákna jsou ukončena konektory typu samice, které jsou umístěny v panelu vedle sebe. Do těchto konektorů jsou pomocí takzvaných patch kabelů (jedná se o propojovací kabel s konektory typu samec na obou stranách) připojeny aktivní prvky sítě.

Cíle a požadavky

Cílem této bakalářské práce je vytvořit mobilní aplikaci pro zařízení iPad, sloužící k evidenci optické sítě. Aplikace bude vycházet z existující webové aplikace a bude využívat existující serverové části této aplikace. Jedná se o aplikaci používanou v prostředí firmy TETA s.r.o., která provozuje metropolitní síť a poskytuje připojení k internetu v krajském městě Ústí nad Labem. Výsledná mobilní aplikace bude určena převážně pro operativní přístup techniků k informacím v terénu.

Tato kapitola se dále zabývá funkčními a nefunkčními požadavky na aplikaci. Tyto požadavky vychází ze zadání a obecných požadavků na mobilní aplikace.

2.1 Funkční požadavky

Funkční požadavky popisují požadované vlastnosti systému a služby, které poskytuje [2]. Funkční požadavky na tuto aplikaci vycházejí především z existující webové aplikace a doplňují ji o další funkcionality.

Funkční požadavky na tuto aplikaci jsou následující:

- Aplikace bude obsahovat tyto hlavní funkce z webové aplikace:
 - a) mapa POPů a kabelů
 - b) seznam POPů
 - c) detail POPu obsahující koncové body a kabely ukončené v tomto POPu
 - d) detail koncového bodu bude obsahovat informace o organizéru, ODF a vývodech
 - e) seznam kabelů
 - f) detailu kabelu, obsahující seznam vláken, která obsahuje.
- Při práci s mapou bude aplikace využívat polohových služeb.

- Hledání pravděpodobného místa závady na základě selhání více spojů.
- Návrh doporučené cesty pro vytvoření spojení na základě zadaných koncových bodů.

2.2 Nefunkční požadavky

Nefunkční požadavky popisují vlastnosti systému jako celku, omezení kladená na systém či proces vývoje. Může se jednat například o požadavky na výkon, odezvu, bezpečnost či udržitelnost [2].

Aplikace pro iPad

Výsledná aplikace bude určena pro iPad. Zařízení s větší úhlopříčkou než mají mobilní telefony bylo zvoleno, protože aplikace na jednotlivých obrazovkách bude prezentovat velmi velké množství dat, které by nebylo možné na menším displeji přehledně zobrazit.

Při výběru platformy, a tedy i výrobce tabletu, byl zohledněn fakt, že technici již využívají mobilní telefon iPhone, který je také produktem firmy Apple, a tak jsou zvyklí na uživatelské rozhraní systému iOS, které se v případě iPadu od menšího iPhoneu zásadně neliší.

Aplikace bude využívat existující REST API

Aplikace bude využívat existující REST API využívané v současné době webovou aplikací. V případě potřeby nových funkcí je možné tyto funkce přidat do existujícího API.

Důraz na bezpečnost

Důraz na bezpečnost by měl být v dnešní době samozřejmostí. Jinak by tomu nemělo být ani v případě této aplikace. Jedná se především o zabezpečení komunikace se serverem, poskytujícím API, ale i o uložení přístupových údajů uživatele v zařízení.

Analýza

3.1 Existující řešení

V rámci analytické části této práce proběhla rešerše existujících aplikací, které řeší zadanou problematiku. V rámci této rešerše byly prozkoumány existující aplikace sloužící k evidenci optické sítě.

Ve většině případů se jedná o nadstavby nad grafické programy sloužící k projektování technických výkresů. Existuje, ale i několik řešení pouze pro správu optické sítě. Ve většině případů jsou webové stránky těchto projektů stručné a u žádného nebyla nalezena zmínka o mobilní aplikaci.

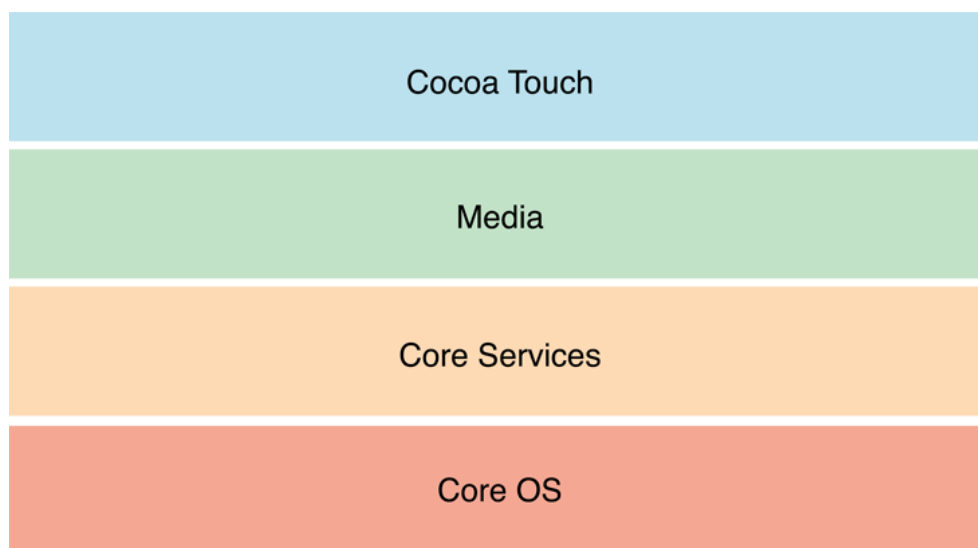
3.2 Platforma

Tato kapitola nejprve popisuje hardware iPadu, poté samotný systém iOS a jeho architekturu a nakonec i prostředky využívané k vývoji aplikací pro systém iOS.

3.2.1 iPad

Apple v historii již několikrát pracoval na vývoji tabletu, žádný z projektů se ovšem nedočkal velkého úspěchu. Projekt Newton, který byl reakcí na nově se objevující segment PDA, se dokonce dostal až do výroby a prodeje, avšak přestože se prodával necelých pět let (v letech 1993-1998) a za tu dobu si získal hodně příznivců, nikdy se nedočkal takového úspěchu, jaký si od něj Apple sliboval. [3] O projektu tabletu od Applu se začalo opět mluvit až s představením iPhone v roce 2007. Apple při vývoji iPhone získal cenné zkušenosti v oblasti zařízení, které je ovládané pouze dotykovým displejem. Netrvalo dlouho a Apple v lednu roku 2010 představil iPad. [4]

Podporovány jsou v současné době všechny iPady, s výjimkou iPadu první generace [5]. Jedná se tedy celkem o 11 podporovaných modelů iPadu. Modelová řada iPadů se skládá ze zařízení tří různých velikostí. Nejmenší iPad



Obrázek 3.1: Vrstvy systému iOS. Převzato z [6]

mini disponuje displejem o velikosti 7,9 palců, poté následuje iPad standardní velikosti 9,7 palců a nejnovějším přírůstkem do rodiny iPadů je iPad Pro s velikostí displeje 12,9 palců.

Z pohledu vývoje ovšem tato skutečnost není velkou komplikací. Až s výjimkou nejnovějšího 12,9 palcového iPadu Pro mají všechny iPady z pohledu vývoje shodné rozlišení displeje. Nejstarší podporované modely tedy iPad 2 a iPad mini mají rozlišení 1024×768 obrazových bodů, novější zařízení pak mají rozlišení displeje dvojnásobné. Při vývoji se ovšem stále pracuje s původním rozlišením a na zařízeních s vyšším rozlišením je obraz jemnější.

3.2.2 iOS

iOS je operační systém vyvíjený firmou Apple. Tento operační systém je vyvíjen přímo pro zařízení společnosti Apple, konkrétně se v současné době jedná o iPhone, iPad a iPod touch. Díky tomu, že Apple vyvíjí hardware i software, je spolupráce hardwaru i softwaru na dobré úrovni.

3.2.2.1 Vrstvená architektura

Operační systém tvoří vrstvu mezi hardwarem a samotnou aplikací. Spravuje hardware a poskytuje rozhraní potřebné pro vývoj aplikací. Architektura systému iOS se dá rozdělit do 4 vrstev (viz obr. 3.1). Vrstvy umístěné výše poskytují objektově orientovaný přístup k základům z nižších vrstev. Pokud je to možné, doporučuje se preferovat frameworky vyšších vrstev před frameworky nižších vrstev. [6]

3.2.2.2 Cocoa Touch

Z výše zmíněných vrstev, je z pohledu této práce nejzásadnější nejvyšší vrstva, kterou je vrstva Cocoa Touch. Tato vrstva obsahuje klíčové frameworky sloužící k vývoji aplikací. Tyto frameworky nejenže definují vzhled aplikace, ale zároveň poskytují podporu pro klíčové technologie, jako je multitasking, interakce aplikace na dotyky uživatele, notifikování uživatele a spoustu dalších. [8] Jedním z klíčových frameworků poskytovaných touto vrstvou je framework UIKit. Tento framework řeší interakci aplikace s uživatelem a řízení běhu aplikace a její interakci se systémem. [9]

3.2.3 Swift

Jazyk Swift je nový programovací jazyk vyvinutý společností Apple pro vývoj aplikací pro mobilní operační systém iOS, počítače Mac a další produkty společnosti Apple. [7] Jazyk Swift by měl do budoucna zcela nahradit jazyk Objective-C, ve kterém se aplikace pro platformy iOS a Mac vyvíjely doposud. Jazyk Swift je rychlejší a bezpečnější než Objective-C. Přesto není problém používat v jedné aplikaci jak kód v jazyku Objective-C, tak i v jazyku Swift. Přestože Apple představil jazyk Swift jako primární jazyk pro psaní iOS aplikací, výše zmíněný framework Cocoa Touch, který zajišťuje běhové prostředí pro aplikace určené pro systém iOS, zůstal i nadále napsán v Objective-C. [10]

3.2.4 Vývojové prostředí

Vývoj aplikací pro iOS probíhá ve vývojovém prostředí od společnosti Apple. Jedná se o vývojové prostředí Xcode. Kromě tvorby aplikací pro iOS zařízení poskytuje možnost tvorby aplikací pro Apple Watch, pro počítače Mac a pro multimediální zařízení Apple TV. Kromě tvorby aplikací pro produkty společnosti Apple je možno využít vývojové prostředí pro vývoj aplikací v jazycích C, C++, Java a dalších [11]. Vývojové prostředí Xcode je dostupné pouze pro operační systém OS X a to prostřednictvím Mac App Store (obchod s aplikacemi) zdarma.

Kromě vývojového prostředí Xcode od společnosti Apple je k dispozici vývojové prostředí AppCode od společnosti JetBrains. Narozdíl od Xcode je AppCode dostupné za poplatek. Oproti Xcode poskytuje pokročilejší editor zdrojových kódů, avšak na druhou stranu má velmi omezené funkce tvorby uživatelského rozhraní pomocí grafického editoru.

3.2.4.1 Tvorba uživatelského rozhraní

Kromě klasické tvorby uživatelského rozhraní pomocí kódu umožňuje Apple snadnější cestu, kdy tvorba probíhá v Xcode pomocí vestavěného nástroje Interface Builder. Ten umožňuje pomocí jednoduchého umístování jednotlivých prvků uživatelského rozhraní v grafickém editoru vytvořit kompletní grafické

3. ANALÝZA

rozhraní aplikace. Takto vytvořené uživatelské rozhraní se poté jednoduše propojí s kódem aplikace a k jednotlivým prvkům je možné přistupovat z kódu. Kromě tvorby jednotlivých obrazovek je možné pomocí grafického editoru vytvořit storyboard, který znázorňuje průchod aplikací a vztahy mezi jednotlivými obrazovkami. Tento nástroj ulehčuje práci s tvorbou uživatelského rozhraní, ale na druhou stranu u složitějších aplikací se spoustou přechodů ztrácí svoji sílu a přestává být přehledný.

3.2.4.2 iOS simulator

Dalším velmi užitečným nástrojem, který je součástí Xcode, je iOS simulátor, který umožňuje běh vyvíjené aplikace na simulátoru iOS zařízení, přičemž je na výběr z několika zařízení. Jedná se o pohodlnější řešení než každou menší změnu aplikace testovat na reálném zařízení. Zároveň je možné díky simulátoru testovat běh aplikace na různých verzích operačního systému a samozřejmě i na více typech zařízení. V současnosti již není ovšem problém testovat běh aplikace i na vlastním zařízení bez omezení. Dříve bylo nutné být pro spuštění na zařízení v placeném Apple developer programu, jinak měl vývojář možnost testovat svoji aplikaci pouze v simulátoru.

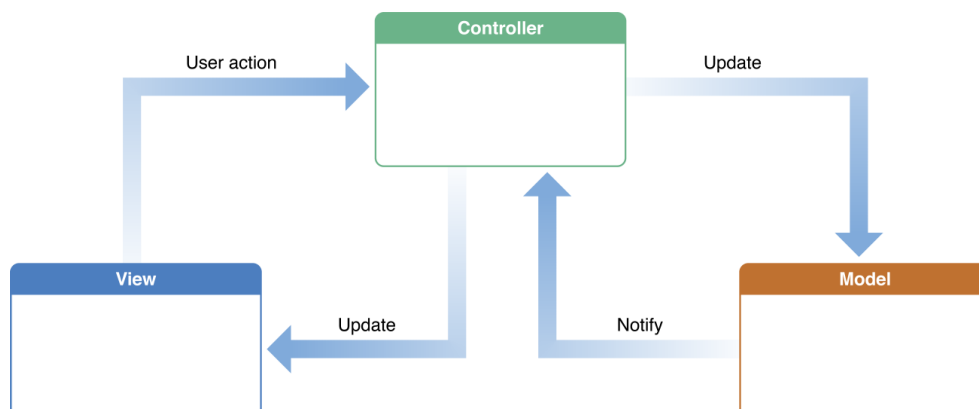
Návrh

4.1 Architektura (MVC)

Návrhový vzor MVC (Model-View-Controller) rozděluje objekty v aplikaci do tří skupin: model, view a controller. Návrhový model určuje nejenom roli jednotlivých objektů, ale i způsob komunikace mezi objekty. (viz obrázek 4.1) Dodržování návrhového vzoru MVC přináší řadu výhod, mezi které patří znovupoužitelnost jednotlivých částí, ale také nahrazení jedné části aplikace částí jinou bez nutnosti zásahu do jiných vrstev. [12]

Model Objekty z vrstvy model reprezentují data aplikace a také slouží k manipulaci s nimi.

View Objekty typu view reprezentují uživatelské rozhraní aplikace. Definují vzhled uživatelského rozhraní, ale i reakce na akce provedené uživatelem. Hlavní funkcí je prezentace dat získaných od Modelu prostřednictvím Controlleru.



Obrázek 4.1: Model MVC. Převzato z [12]

Controller Objekty typu controller slouží jako zprostředkovatelé mezi objekty typu view a objekty typu model.

Příklad komunikace mezi jednotlivými vrstvami může být následující. Objekt typu controller obdrží od objektu typu view požadavek na zpracování uživatelské akce, pokud je potřeba požádá objekt typu model o data. Ve chvíli kdy jsou data k dispozici, předá je objektu view, který je zobrazí.

Zpravidla se složitější aplikace neskládají pouze z jednoho MVC, ale z více navzájem propojených MVC. Tyto MVC pak například sdílí stejnou vrstvu modelu nebo mezi sebou komunikují jednotlivé objekty typu controller.

4.2 Doménový model

Doménový model aplikace vychází z existující webové aplikace a především jejího API. Na obrázku 4.2 je znázorněn vztah mezi jednotlivými entitami. Pro zjednodušení jsou v doménovém modelu vyobrazeny pouze nejdůležitější entity. Například atribut `gps` entity `POP` je samostatnou entitou, která ovšem obsahuje pouze údaj o zeměpisné šířce a délce, a tak je pro zjednodušení uvedena jako atribut.

Entity doménového modelu `POP`, `Cable`, `Fiber`, `EndPoint`, `Organizer` a `ODF` představují evidované položky popsané v první kapitole (konkrétně `POP`, optický kabel, vlákno, koncový bod, organizér a `ODF`).

4.2.1 Fost

Entita `Fost` představuje kazetu, ve které jsou uloženy sváry optických vláken. Kazeta může být součástí organizéru či `ODF`.

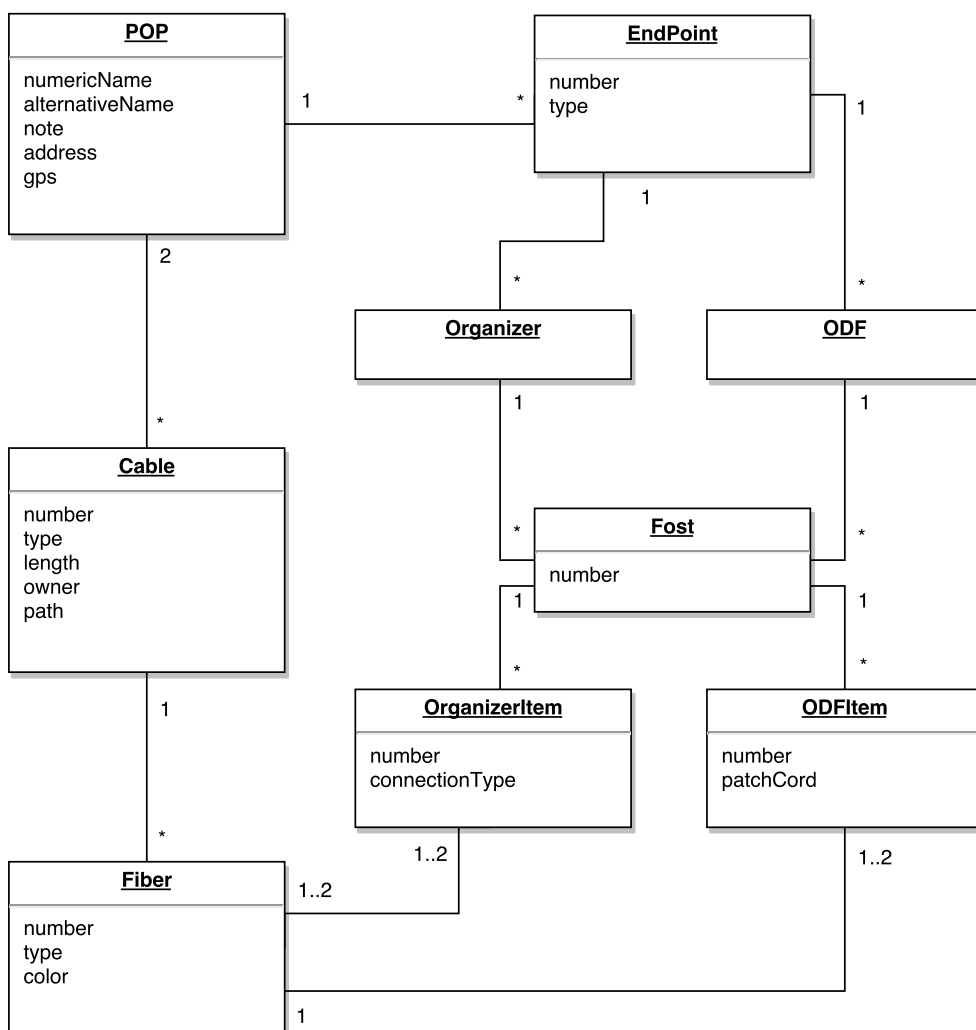
V případě organizéru je jednotlivá položka v kazetě evidována jako entita `OrganizerItem` a v případě, že se kazeta nachází v `ODF` je jednotlivá položka evidována jako entita `ODFItem`.

4.2.2 OrganizerItem

Entita `OrganizerItem` představuje položku kazety v organizéru. Tato položka může obsahovat buď dvě vlákna, jedno na levé a jedno na pravé straně, nebo může obsahovat vlákno pouze na jedné straně či dokonce neobsahovat žádné vlákno.

4.2.3 ODFItem

Entita `ODFItem` představuje položku v `ODF`. Tato položka obsahuje jedno nebo žádné vlákno. Dále pak může obsahovat informaci o připojeném patch kabelu.



Obrázek 4.2: Doménový model

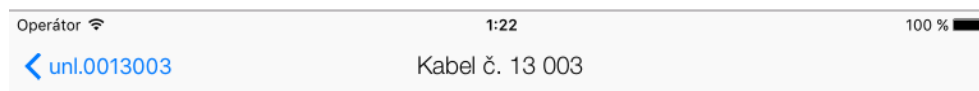
4.3 Návrh uživatelského rozhraní

Návrh uživatelského rozhraní z části vychází z existující aplikace. Její jednotlivé obrazovky i způsob pohybu mezi nimi je uzpůsoben zobrazení na mobilním zařízení. Při návrhu bylo postupováno podle iOS Human Interface Guidelines (HIG), což je dokument vydávaný přímo společností Apple, který popisuje jak navrhovat uživatelské rozhraní aplikace, tak aby aplikace dobře vypadala a intuitivně se ovládala.

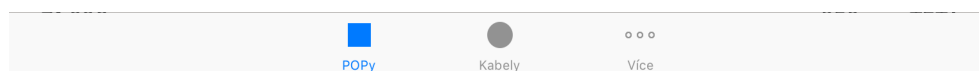
V této části následuje nejprve popis navigace aplikací a dále návrh nejdůležitějších obrazovek aplikace.

Návrh probíhal pomocí vestavěné funkce programu Xcode Interface Builder. Konkrétně pak storyboardu, popsaného v 3.2.4.1. Výhodou tohoto po-

4. NÁVRH



Obrázek 4.3: Ukázka prvku Navigation Bar



Obrázek 4.4: Ukázka prvku Tab Bar

stupu je, že po návrhu získáváme i funkční prototyp aplikace a následně je možná takto vytvořený návrh použít pro tvorbu aplikace samotné. O nevýhodách použití storyboardu se pak zmiňuje kapitola 5.2 zabývající se implementací uživatelského rozhraní.

Vzhledem k použitému způsobu návrhu jednotlivých obrazovek se výsledné obrazovky zásadně neliší od těch navržených ve storyboardu, a tak budou v této kapitole použity ukázky z hotové aplikace.

V této kapitole se kromě popisu jednotlivých obrazovek nachází pouze několik ukázek uživatelského rozhraní. Ukázky všech obrazovek v aplikaci se nachází v příloze B.

4.3.1 Navigace aplikací

První důležitou věcí při návrhu uživatelského rozhraní aplikace je navigace mezi jednotlivými obrazovkami. Obecně se v iOS aplikacích, když pomíneme hry, používají dva základní způsoby navigace aplikací.

Prvním je využití navigační lišty v horní části obrazovky (Navigation Bar), která slouží pro pohyb hierarchickou strukturou aplikace (viz obr. 4.3). Lišta se skládá z titulku, který říká uživateli, v jaké části aplikace se právě nachází. Pokud se uživatel již nachází zanořen hlouběji ve struktuře aplikace, nachází se v levé části lišty tlačítko pro návrat zpět na předchozí obrazovku. Vpravo poté může být umístěno tlačítko sloužící k úpravě obsahu - například vyhledávání, přidávání, editace a podobně. [13]

Druhým způsobem je lišta záložek ve spodní části obrazovky (Tab Bar), která slouží pro pohyb mezi rovnocennými a nezávislými obrazovkami bez ohledu na to, kde se právě uživatel v aplikaci nachází (viz obr. 4.4). Jedná se o lištu ve spodní části obrazovky, ve které se nacházejí ikonky jednotlivých záložek společně s jejich textovým popisem. [13]

V této aplikaci je využíváno obou způsobů, kdy mezi hlavními kategoriemi se přepíná pomocí záložek, tedy Tab Baru a v rámci každé hlavní obrazovky slouží k navigaci hierarchií obrazovek Navigation Bar.

4.3.2 Přihlašovací obrazovka

Při prvním zapnutí aplikace nebo v případě, že se uživatel odhlásí, je po zapnutí jako první zobrazena přihlašovací obrazovka. Apple ve svých dříve zmíněných HIG doporučuje nezobrazovat přihlašovací obrazovku ihned po spuštění aplikace, ale až ve chvíli, kdy je to nezbytné. Vzhledem k tomu, že aplikace zobrazuje výhradně data stažená pomocí REST API, které požaduje přihlášení, je nutné vyžádat přihlašovací údaje od uživatele ihned po prvním zapnutí aplikace.

Tato obrazovka obsahuje logo aplikace, políčka pro zadání uživatelského jména a hesla a tlačítko pro přihlášení. Při úspěšném přihlášení je uživateli zobrazena hlavní obrazovka aplikace. V případě selhání přihlášení je uživateli zobrazena chybová hláška.

4.3.3 Zobrazení mapy POPů a kabelů

První položkou v Tab Baru, která je zároveň zobrazena jako první při spuštění aplikace je obrazovka zobrazující mapu POPů a kabelů. POPy jsou v mapě zakresleny jako body a kabely pomocí křivek. Po kliknutí na bod, který představuje POP je zobrazen štítek s číslem, jménem a adresou POPu. Po kliknutí na tento štítek se přejde na obrazovku s detailem POPu.

V pravé horní části navigační lišty se nachází tlačítko pro vyhledávání. Po jeho stisku se místo této navigační lišty zobrazí políčko pro vyhledávání. Při zadávání znaků jsou postupně hledány výsledky a zobrazovány uživateli. Mapa je přibližována tak, aby pokryla všechny položky odpovídající vyhledávání. V pravé části vyhledávacího pole je tlačítko pro zrušení hledání. Po jeho stisku je opět zobrazen kompletní seznam POPů a mapa přejde do výchozího zobrazení.

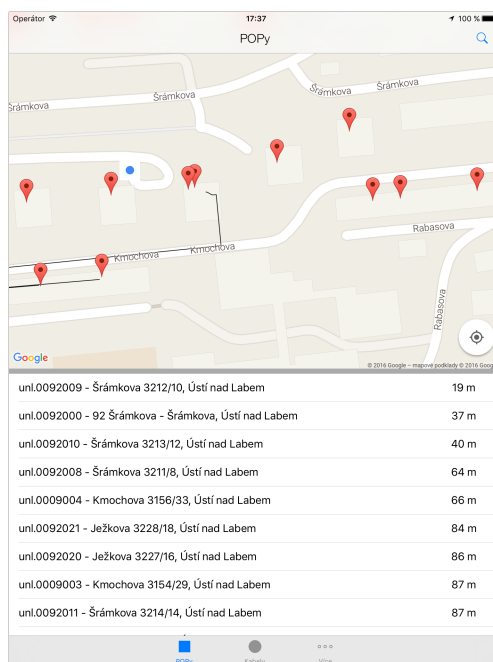
Ve spodní polovině této obrazovky s mapou se nachází tabulka obsahující seznam POPů. U každé položky je zároveň uvedena vzdálenost položky od uživatele. V případě, že je aktivní vyhledávání, obsahuje tabulka seznam POPů, které odpovídají výsledku vyhledávání. V případě že vyhledávání aktivní není, obsahuje seznam všech POPů. V obou případech jsou POPy seřazeny podle vzdálenosti od uživatele a zobrazeny od nejbližších.

Na obrázku 4.5 je zobrazena ukázka této obrazovky.

4.3.4 Zobrazení seznamu kabelů

Druhou položkou v Tab Baru je obrazovka sloužící k zobrazení seznamu všech kabelů. Jedná se o jednoduchou tabulku obsahující seznam kabelů. Každá položka obsahuje číslo kabelu, dva POPy, mezi kterými kabel vede, počet vláken v kabelu, typ vláken v kabelu, délku kabelu a vlastníka kabelu. Kliknutím na příslušný řádek se přejde na obrazovku s detailem kabelu. V navigační liště se nachází tlačítko sloužící k vyhledávání konkrétního kabelu podle čísla. Po

4. NÁVRH



Obrázek 4.5: Zobrazení mapy POPů a kabelů

zadání příslušného čísla kabelu je v tabulce kabelů zobrazen kabel odpovídajícího čísla.

4.3.5 Nástroje

Poslední položkou v Tab Baru je obrazovka s nástroji. Jedná se o jednoduchou tabulku pro přístup k obrazovkám sloužícím k návrhu doporučené trasy a hledání místa závady. V pravé horní části obrazovky se nachází tlačítko pro odhlášení uživatele z aplikace.

4.3.6 Detail POPu

Obrazovka zobrazující detail POPu se skládá ze tří částí. První je záhlaví ve vrchní části obrazovky, po kterém následují dvě tabulky.

První tabulka obsahuje koncové body nacházející se v POPu. U každého koncového bodu je uvedeno jeho číslo, typ, počet kazet a pozic u typu organizér, respektive adaptérů v případě typu ODF a poznámka. Výběrem koncového bodu se uživatel dostane na detail příslušného koncového bodu.

Druhá tabulka reprezentuje optické kabely ukončené v tomto POPu. U kabelů je uvedeno číslo kabelu, POP, do kterého kabel vede, počet a typ vláken v kabelu, délka a vlastník kabelu. Výběrem kabelu se uživatel dostane na detail kabelu.

4.3.7 Detail kabelu

Tato obrazovka slouží k zobrazení informací o kabelu a vláknech, které obsahuje. V navigační liště je uvedeno číslo právě zobrazovaného kabelu, následuje hlavička obsahující informace o kabelu. Tyto informace obsahují i POPy, mezi kterými kabel vede. Při kliku na tlačítko s názvem a adresou POPu se přejde na obrazovku s detailem příslušného POPu.

Další částí obrazovky je tabulka, která obsahuje jednotlivá vlákna nacházející se v kabelu. U každého kabelu je zobrazeno číslo vlákna, číslo bufferu, ve kterém se vlákno nachází, a barva vlákna. Dále řádek obsahuje dvě tlačítka s informací o koncových bodech, ve kterých je vlákno ukončeno. Při kliku na tlačítko koncového bodu se přejde na jeho detail. Poslední informací v řádce je obsazení obou konců vlákna.

4.3.8 Detail koncového bodu

Tato obrazovka se skládá z horní hlavičky, kde jsou souhrnné informace o koncovém bodu. Dále následuje část, ve které se nachází tři typy pohledů. Jedná se o pohled na organizér, pohled na ODF a pohled na vývody.

Mezi jednotlivými pohledy se přepíná pomocí přepínače typu Segmented Control, který se skládá z jednotlivých tlačítek zvaných segmenty. Tento přepínač slouží k přepínání mezi jednotlivými prvky, kdy vybraný prvek je barevně zvýrazněn a vždy může být vybrán pouze jeden segment.

4.3.8.1 Pohled na organizér

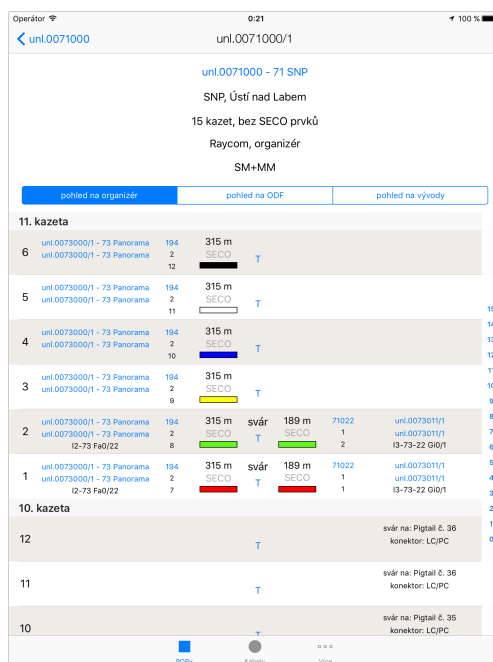
Pohled na organizér je jednou z nejsložitějších obrazovek v aplikaci. Její složitost je dána především množstvím obsahu, který je potřeba zobrazit. Skládá se z tabulky, která je znázorněním reálného uložení vláken v kazetách. Řádek tabulky reprezentuje jednu pozici v optické kazetě. Řádka začíná číslem, které udává pozici vlákna v kazetě. Vlákna jsou číslována od nejvyššího po nejnižší podle zvyklosti techniků. Poté by se dala řádka logicky rozdělit na levou a pravou stranu, mezi těmito stranami je informace o typu propojení těchto dvou stran a tlačítko sloužící k zobrazení trasy vlákna. Levá i pravá strana organizéru může obsahovat buď vlákno, pigtail, ranžír nebo být prázdná. Podle toho se také liší zobrazení v tabulce.

Na obrázku 4.6 je zobrazena ukázka této obrazovky.

4.3.8.2 Pohled na ODF

Pohled na ODF je velmi podobný pohledu na organizér, přičemž obsahuje v levé části informaci o patch kabelu, který je připojen do příslušného konektoru ODF. V pravé části se nachází informace o vláknu známá z pohledu na organizér.

4. NÁVRH



Obrázek 4.6: Pohled na koncový bod - organizér.

4.3.8.3 Pohled na vývody

Data zobrazována v pohledu na vývody jsou úzce spjata s pohledem na ODF i pohledem na organizér. V případě, že obsahuje koncový bod ODF, obsahuje vývody tohoto ODF. Konkrétně se jedná o tzv. pigtaily. Jedná se o konektor s krátkým kusem kabelu, který se navaří na vlákno, protože konektorování kabelů v optických sítích je velmi komplikovanou záležitostí a takto je zaručena vyšší kvalita.

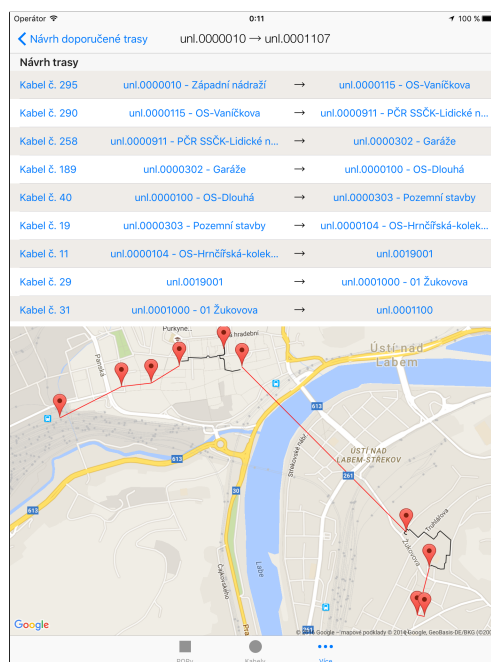
4.3.9 Trasa vlákna

Tato obrazovka v horní části zobrazuje informace o kabelu, následuje tabulka s jednotlivými položkami trasy. Pod tabulkou se nachází mapa trasy, na které se nachází koncové body vlákna a trasa vlákna. V případě, že pro kabel, ve kterém se vlákno nachází, neexistují polohová data, je trasa tohoto vlákna zakreslena červeně jako přímka spojující jeho začátek a konec.

4.3.10 Návrh doporučené trasy

Tato obrazovka obsahuje dvě tabulky se seznamem POPů. V tabulkách je možné vyhledávat pomocí vyhledávacího pole v jejich záhlaví. V obou tabulkách je možné vybrat jeden POP, tyto dva vybrané POPy poté tvoří počáteční a koncový bod hledané trasy (na pořadí bodů samozřejmě nezáleží). Ve spodní

4.3. Návrh uživatelského rozhraní



Obrázek 4.7: Výsledek návrhu doporučené trasy

části se nachází tlačítko, které slouží k přechodu na obrazovku zobrazující výsledek návrhu doporučené trasy.

4.3.11 Výsledek návrhu doporučené trasy

Tato obrazovka slouží k prezentaci návrhu trasy. V horní polovině obrazovky se nachází tabulka obsahující kabely a POPy, mezi kterými tyto kabely vedou. Ve spodní části obrazovky se nachází mapa vyhledané trasy. Shodně jako u zobrazení trasy vláknů jsou kabely, u kterých nejsou k dispozici polohová data, zobrazeny jako červená přímka spojující jejich počáteční a koncový bod.

Na obrázku 4.7 je zobrazena ukázka této obrazovky.

4.3.12 Vyhledávání závady

Obrazovka sloužící k vyhledávání pravděpodobného místa závady je jednoduchou tabulkou, která obsahuje pronajaté spoje a ze které je možné vybrat libovolné množství položek. Ve spodní části obrazovky se nachází tlačítko, které slouží k přechodu na obrazovku zobrazující výsledek tohoto hledání.

4.3.13 Výsledek vyhledávání závady

Výsledek vyhledávání závady je prezentován v tabulce, ve které se na jednotlivých řádkách nachází jednotlivé části sítě, na kterých mohla nastat závada.

4. NÁVRH

Kliknutím na příslušnou položku se přejde na její detail (jedná se o detail kabelu či koncového bodu).

Realizace

5.1 Použité externí knihovny

Pro správu externích knihoven je použit manažer závislostí CocoaPods. Ten umožňuje jednoduchým způsobem, pomocí definice v jednoduchém textovém souboru, přidávat do vlastní aplikace frameworky a knihovny třetích stran. Frameworky a knihovny je samozřejmě možné do projektu přidávat i ručním nakopírováním zdrojových kódů. Využití CocoaPods ovšem ulehčí nejen samotné přidávání a vyhledávání frameworků a knihoven, ale především i jejich aktualizace.

5.1.1 Google Maps SDK

Google Maps SDK je sadou nástrojů sloužící k využití map od společnosti Google v iOS aplikacích. Oproti klasickému SDK pro použití Google map ve webových aplikacích nabízí menší množství funkcí. Zároveň neumožňuje tolik funkcionalit jako MapKit od Applu, který slouží k práci s mapou v iOS aplikacích. Přestože je na tom MapKit co se týče funkcí lépe, nemůže se Google mapám rovnat v kvalitě mapových podkladů.

Kvůli lepším mapovým podkladům byly, i přes zvážení větších funkcionalit MapKitu od Applu, použity pro zobrazování map v aplikaci právě Google mapy. Důvodem byl již zmiňovaný rozdíl v mapových podkladech obou řešení.

5.1.2 Alamofire

Alamofire je ve Swiftu napsaná knihovna sloužící k práci se sítí v iOS aplikacích, ale i v aplikacích pro počítače Mac. Jedná se o nadstavbu nad standardní funkce ze systémového frameworku Foundation, která zjednodušuje práci se síťovým spojením prostřednictvím protokolu HTTP. [14]

5.1.3 SwiftyJSON

SwiftyJSON je knihovna ulehčující parsování dat ve formátu JSON.

5.1.4 SVProgressHUD

SVProgressHUD je jednoduchá knihovna sloužící k zobrazení indikátoru aktivity uživateli. Využívá se například při načítání dat.

5.1.5 KeychainSwift

KeychainSwift je knihovna sloužící ke zjednodušení zápisu a čtení ze zabezpečeného úložiště Keychain, které je poskytováno systémem iOS.

5.1.6 UIColor Hex Swift

Jednoduchá knihovna umožňující použití barev definovaných kódem RGBA (kód definující jednotlivé prvky barevného spektra RGB, tedy červená, zelená a modrá, doplněný o hodnotu alpha, která určuje průhlednost).

5.2 Uživatelské rozhraní

K tvorbě uživatelského rozhraní bylo využito nástroje Interface Builder (viz. kapitola 3.2.4.1). Přičemž jednotlivé obrazovky byly navrženy v rámci storyboardu, který obsahuje nejenom jednotlivé obrazovky, ale i přechody mezi jednotlivými obrazovkami. Kromě definice přechodů ve storyboardu, byly přechody ze složitějších tabulek, jako je například pohled na organizér, definovány ve zdrojovém kódu.

Při zpětném pohledu nebyla volba definice uživatelského rozhraní pomocí storyboardu ideální. Při rozsáhlejších aplikacích již přestává být přehledný a to obzvláště při práci na monitoru s menším rozlišením. Vhodnějším řešením by bylo buď použít definici jednotlivých obrazovek v samostatných souborech, což Interface Builder také umožňuje, a jejich vzájemné propojení a přechody řešit pouze na úrovni kódu, nebo kompletní definice uživatelského rozhraní pomocí zdrojového kódu.

Zásadní nevýhodou využívání storyboardu je roztříštěnost jednotlivých částí, které spravují vzhled aplikace, ale i přechody mezi jednotlivými obrazovkami. Existují parametry prvků uživatelského rozhraní, které je možné upravovat pouze pomocí kódu, a tak se pak stává, že část nastavení prvku je udělána pomocí Interface Builderu a část pomocí kódu. Ve chvíli vytváření takového prvku se tento fakt nejeví jako velký problém, ovšem v případě, že je zapotřebí prvek nějakým způsobem upravit, je potřeba složitě hledat, kde byla která věc nastavena.

Dalším problémem při využití storyboardu je předávání informací mezi controllery, kdy jednotlivý objekt typu controller není vytvářen pomocí kódu,

ale automaticky pomocí storyboardu, který také řídí přechody mezi jednotlivými controllery. K předání dat novému controlleru potom slouží metoda `prepareForSegue`. V této metodě rozlišujeme mezi jednotlivými přechody pomocí jejich identifikátorů nastavených ve storyboardu. U složitějších obrazovek, ze kterých se přechází na několik míst, obsahuje tato metoda velké množství podmínek.

Dále bylo při tvorbě uživatelského rozhraní využito techniky Auto Layout popsané v následující kapitole.

5.2.1 Auto Layout

Auto Layout slouží k dynamickému rozmístění prvků v rámci obrazovky, případně v rámci jejich částí jako je například řádek tabulky. Pozice prvků nejsou definovány absolutně, ale relativně k ostatním prvkům. Tyto pozice jsou definovány pomocí jednotlivých omezení (constraints). [15]

Díky této technice se při změně orientace zařízení přizpůsobí obsah obrazovky nové velikosti. Zároveň je možné díky Auto Layout spouštět aplikaci na zařízeních s různou velikostí displeje při zachování jedné definice uživatelského rozhraní. To je velkou výhodou obzvláště při vývoji aplikací pro iPhone, u kterého je více různých rozlišení displeje napříč verzemi, ale s příchodem většího iPad Pro s vyšším rozlišením je zapotřebí tento fakt zohledňovat i při vývoji aplikací pro iPad.

5.3 Datové třídy

Jednotlivé evidované entity popsané v doménovém modelu jsou v aplikaci prezentovány pomocí datových tříd. Kromě těchto hlavních entit jsou použity i další pomocné třídy jako jsou například `Gps` reprezentující polohu nebo `Address` reprezentující adresu.

Tyto datové třídy obsahují jednotlivé atributy evidovaných entit a konstruktor přijímající jako parametr strukturu JSON, která je definována knihovnou `SwiftyJSON` a vytvořena po přijetí dat ze serveru nebo je částí již zpracovaného JSONu jako například v případě atributů `gps` a `address` v následujícím příkladu.

Využití tohoto způsobu získávání datových objektů z dat ve formátu JSON umožňuje jednoduché přidání atributů do jednotlivých tříd bez nutnosti modifikovat třídy, které je dále využívají.

Příklad definice třídy `Pop`:

```
class Pop {
    var id: Int = 0
    var numericName: String = ""
    var alternativeName: String?
    var note: String?
```

```
var address: Address = Address()
var gps: Gps = Gps()
var distance: Int?

init(json: JSON) {
    id = json["id"].intValue
    numericName = json["popNumericName"].stringValue
    alternativeName = json["alternativeName"].string
    note = json["note"].string
    address = Address(json: json["address"])
    gps = Gps(json: json["gps"])
}

init () {}
}
```

5.4 Získávání dat ze serveru

Pro získání dat ze serveru slouží třída `RestManager`, která pošle dotaz serveru a zpracuje jeho odpověď. Třída obsahuje metody pro získávání jednotlivých druhů dat, jako jsou například `getCables`, `getPops` a tak dále. Kromě specifických parametrů mají tyto metody dva parametry vždy stejné, jedná se o parametry `success` a `failed`, kterými jsou funkce z volajícího controlleru.

Příklad hlavičky funkce sloužící k získání POPů:

```
func getPops(success: ((Array<Pop>) -> Void),
failed: ((Array<String>) -> Void))
```

Tyto metody pomocí knihovny `Alamofire` pošlou GET požadavek na server. V případě úspěšného získání dat se ze získaných dat ve formátu JSON vytvoří objekt či pole objektů (v tomto případě pole POPů), které je následně předáno funkci `success`, která byla zadána jako argument při volání této metody. Stahování dat probíhá asynchronně a po dokončení je díky této funkci upozorněn controller, který inicioval stahování. V opačném případě, tedy při neúspěchu, je volána funkce `failed`, které je předáno pole chybových hlášek, jež jsou controllerem prezentovány uživateli.

5.5 Bezpečnost aplikace

Při návrhu každé aplikace by měl být kladen důraz na bezpečnost. Obzvláště v případě aplikace komunikující skrze internet, která ukládá přihlašovací údaje uživatele.

5.5.1 Přihlašování do aplikace

Při prvním spuštění aplikace je uživateli zobrazena přihlašovací obrazovka. Pokud dojde k úspěšnému ověření proti serveru, jsou přihlašovací údaje uloženy v zařízení.

Pro uložení přihlašovacích údajů je využito služby Keychain, což je zabezpečené úložiště, sloužící k ukládání citlivých dat, poskytované systémem iOS.

5.5.2 Komunikace

Zabezpečení komunikace je další důležitou částí bezpečnosti aplikace. Aplikace v současné době komunikuje se serverem prostřednictvím protokolu HTTP, ale podpora zabezpečeného protokolu HTTPS je samozřejmostí a ve chvíli kdy bude protokol podporován ze strany serveru může dojít k jeho nasazení.

V současné době ovšem skutečnost, že aplikace využívá nezabezpečeného protokolu HTTP, neohrožuje bezpečnost přenášených dat. Připojení k severu je možné pouze skrze zabezpečenou podnikovou síť nebo pomocí VPN.

5.6 Podpora IPv6

Obdobná situace jako v případě podpory HTTPS nastává i v případě protokolu IPv6. Jakmile bude protokol podporován ze strany serveru, bude moci být využit namísto stávajícího protokolu IPv4. Aplikace se k serveru připojuje pomocí doménového jména, a tak po nasazení IPv6 na straně serveru bude moci být tento protokol ihned využíván.

5.7 Přidané funkcionality

Součástí zadání práce bylo doplnění aplikace o nové funkcionality, konkrétně se jedná o:

- hledání pravděpodobného místa závady na základě selhání více spojů,
- návrh doporučené cesty pro vytvoření spojení na základě zadaných koncových bodů.

Při přidávání nových funkcionalit je důležité rozhodnout o místě zpracování dat potřebných pro tyto nové funkcionality. Nabízejí se dvě možnosti a to zpracování dat na straně serveru, přičemž jsou zařízení odeslána data, která je možné bez větších úprav prezentovat uživateli, nebo zpracování dat na straně zařízení.

Zpracování dat na straně serveru je možné v případě, že má vývojář aplikace přístup i k serverové části aplikace. Tato možnost je výhodnější především

z pohledu budoucího využití funkce v jiné klientské aplikaci, kdy bude možné využít již implementované funkcionality serveru.

Varianta zpracování dat na straně zařízení se hodí především pro jednodušší úkony nebo v případě, kdy nemá vývojář přístup k serverové části aplikace.

Zvolené řešení

Z výše uvedených důvodů bylo rozhodnuto, že data budou pro nově přidané funkce zpracovávána na serveru. Bylo tedy zapotřebí doplnit funkce do již existující serverové aplikace. Jedná se o rozsáhlý informační systém, který kromě zde využívané evidence optické sítě slouží k řadě dalších úkonů. Tento informační systém je napsán v jazyku Java a využívá frameworky Maven, Spring, Hibernate a Struts. S využitím těchto prostředků tedy byly implementovány i doplněné funkce.

5.7.1 Hledání pravděpodobného místa závady na základě selhání více spojů

První novou funkcí je hledání pravděpodobného místa závady na základě selhání více spojů. Uživatel zadá dva a více spojů a úkolem aplikace je nalézt společné prvky tras těchto spojů.

Jedná se o průnik dvou a více množin. Samotný průnik dvou množin je v programovacím jazyce Java jednoduchou záležitostí a postačí využít metodu `retainAll`, kterou poskytuje interface `Set` z balíčku `java.util`, který slouží v Javě k reprezentaci množin. Aby bylo možné dělat průnik více množin je tato metoda volána opakovaně a vždy je jí jako parametr předávána další množina.

5.7.2 Hledání nejkratší cesty

Další novou funkcí je návrh doporučené cesty pro vytvoření spojení na základě daných koncových bodů.

Problém hledání nejkratší cesty pro vytvoření spojení v optické síti je problémem hledání nejkratší cesty v grafu.

Pro hledání nejkratší cesty v grafu byl zvolen Dijkstrův algoritmus. Jedná se o algoritmus pro hledání všech nejkratších cest ze zadaného uzlu do ostatních uzlů grafu. Předpokladem pro využití Dijkstrova algoritmu je fakt, že graf neobsahuje záporně ohodnocené hrany. [17]

Pro hledání nejkratší cesty v síti představují hrany grafu kabely a délkou hrany je délka kabelu. Délka kabelu nemůže nabývat záporných hodnot, tím pádem je splněn předpoklad, že graf neobsahuje hrany o záporných délkách. Uzlem grafu je v tomto případě POP.

Při hledání trasy je zapotřebí brát v úvahu pouze kabely, které mají minimálně jedno volné vlákno, aby bylo možné vyhledané spojení realizovat. Dále

je nutné rozlišovat mezi jednovidovými a mnohavidovými vlákny, protože je možné spojovat pouze vlákna stejného typu kvůli odlišnému způsobu přenosu světelných signálů.

Jsou tedy vytvořeny dva grafy, jeden obsahující pouze optické kabely s jednovidovými vlákny a druhý s mnohavidovými. Hledání nejkratší cesty probíhá pro každý graf zvlášť a vrácena je kratší cesta.

Samotná práce s grafem byla realizována pomocí open source knihovny JGraphT. Tato knihovna nabízí třídy sloužící k reprezentaci grafů a grafové algoritmy. Graf je reprezentován jako entita třídy `WightedGraph`, která slouží k reprezentaci neorientovaného grafu s různými hodnotami uzlů. Poté je vytvořen objekt typu `DijkstraShortestPath`, kterému jako parametry konstruktoru předáme vytvořený graf, počáteční a cílový uzel. Při vytvoření tohoto objektu dojde k provedení Dijkstrova algoritmu a výslednou nejkratší cestu je možné získat pomocí metody `getPathEdgeList`, která vrátí hrany nejkratší cesty.

5.8 Zobrazení nejbližších POPů

Na obrazovce, která zobrazuje mapu a tabulku se seznamem POPů, jsou POPy v tabulce seřazeny podle vzdálenosti od uživatele od nejbližších a zároveň je u každé položky zobrazena vzdálenost od uživatele.

Pro práci s polohou uživatele slouží třída `CLLocationManager`. Jedná se o třídu, která slouží k získání polohy uživatele a umožňuje reagovat na její změnu. Aby mohla aplikace získat polohu zařízení, musí mít uživatel povoleno použití polohových služeb v nastavení. Zároveň uživatel může povolit či zakázat použití polohových služeb pro konkrétní aplikaci. [16]

Všechny POPy obsahují informaci o jejich poloze, díky tomu může být pro každý POP vypočítána vzdálenost od polohy zařízení. Tato vzdálenost je pro každý POP uložena do atributu `distance`. Následně proběhne seřazení POPů v poli, ve kterém jsou uloženy, podle této vzdálenosti od nejnižší po nejvyšší.

Tento proces probíhá poprvé po získání dat ze serveru a následně při každé změně uživatelovy polohy o více než 100 metrů. Tohoto chování, lze docílit díky atributu `distanceFilter` třídy `CLLocationManager`, který zajistí, že je notifikována změna přesahující určitou vzdálenost.

Testování

6.1 Uživatelské testování

Uživatelské testování probíhalo zkušebním nasazením aplikace do provozu při běžné práci techniků. Při tomto testování bylo odhaleno několik chyb, které byly nahlášeny a odstraněny.

Zároveň byly na základě reakcí techniků provedeny menší změny v uživatelském rozhraní aplikace.

6.2 Crash reporting

Při uživatelském testování bylo zároveň použito crash reportingu. Ten slouží k odesílání informací o pádu aplikace vývojáři. Crash reporting je nástroj skládající se z knihovny, která je přidána do aplikace, a ze serverové části, která tyto data přijímá a prezentuje vývojáři.

Nástrojů sloužících k těmto účelům existuje několik. Kromě samotného crash reportingu pak některé z nich obsahuje i další nástroje, jako jsou například statistiky používání aplikace. Pro tuto práci byl zvolený bezplatný nástroj Crashlytics, který je součástí platformy Fabric sloužící ke správě vývoje aplikací. [18]

Vývojář se díky reportům dozví o pádu aplikace. Report o pádu obsahuje přesné místo v kódu, ve kterém došlo k pádu aplikace, ale i typ zařízení, verzi operačního systému a další informace. Pro potřeby testování je možné zahrnout do reportu i informace o uživateli aplikace.

Crash reporting se nevyužívá pouze při testování aplikace, ale je možné jej využít i při produkčním nasazení, kdy se díky notifikaci o pádu může vývojář dozvědět o případných chybách, které nebyly odhaleny při testování.

Závěr

Cílem této práce bylo vytvoření aplikace pro mobilní zařízení iPad, která slouží k evidenci optické sítě. Aplikace vychází z existující webové aplikace a využívá jejího REST API.

Při vývoji byly do aplikace doimplementovány nové funkce. Jedná se o hledání pravděpodobného místa závady na základě selhání více spojů a návrh doporučené cesty pro vytvoření spojení na základě zadaných koncových bodů. Tyto funkcionality byly implementovány na straně serveru, a tak je bude možné v budoucnu použít i ve webové aplikaci, ze které tato mobilní aplikace vychází.

Výsledná aplikace splňuje požadavky vytyčené v zadání práce i funkční a nefunkční požadavky definované v kapitole 2 a umožňuje technikům rychlý přístup k informacím v terénu prostřednictvím tabletu iPad.

V budoucnosti by bylo možné rozšířit aplikaci o editační možnosti, ale nabízejí se i další nové funkcionality, jako je například lokalizace závady na optickém kabelu pomocí údajů z měřicího zařízení a další funkce, které by usnadnily technikům řešení potíží v terénu.

Dalším místem pro zlepšení je načítání dat do mapy, které v současné době probíhá načtením všech dat najednou. Po úpravách na straně serveru by bylo možné načítat data pouze pro určitý výřez, tím by se značně zvýšil uživatelský komfort aplikace. Toto zlepšení je plánováno do webové aplikace a při jeho podpoře ze strany serveru nebude problém ho nasadit i v mobilní aplikaci.

Literatura

- [1] Freudenrich, C.: How Fiber Optics Work [online]. 2016, [cit. 2016-02-17]. Dostupné z: <http://computer.howstuffworks.com/fiber-optic.htm>
- [2] Zendulka, J.: Projektování programových systémů - Požadavky a jejich specifikace [online]. 2016, [cit. 2016-03-14]. Dostupné z: http://www.fit.vutbr.cz/study/courses/PPS/public/pdf/5_2.pdf
- [3] Erben, L.: Příchod hackerů: Newtonův vzestup a pád [online]. 2015, [cit. 2016-02-05]. Dostupné z: <http://www.root.cz/clanky/prichod-hackeru-newtonuv-vzestup-a-pad/>
- [4] Apple Inc.: iPad [online]. 2010, [cit. 2016-02-05]. Dostupné z: <http://www.apple.com/pr/library/2010/01/27Apple-Launches-iPad.html>
- [5] Apple Inc.: iOS 9 - What's New [online]. 2016, [cit. 2016-02-05]. Dostupné z: <http://www.apple.com/ios/whats-new/>
- [6] Apple Inc.: About the iOS Technologies [online]. 2014, [cit. 2016-02-07]. Dostupné z: <https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html>
- [7] Apple Inc.: Swift [online]. 2016, [cit. 2016-03-05]. Dostupné z: <http://www.apple.com/swift/>
- [8] Apple Inc.: Cocoa Touch Layer [online]. 2014, [cit. 2016-02-07]. Dostupné z: <https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSTechnologies/iPhoneOSTechnologies.html>
- [9] Apple Inc.: UIKit Framework Reference [online]. 2014, [cit. 2016-02-07]. Dostupné z: https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIKit_Framework/index.html

- [10] McNeish, K.: Swift 101 - Swift Meets the Cocoa Touch Framework [online]. 2015, [cit. 2016-02-12]. Dostupné z: <http://www.iphonelife.com/blog/31369/swift-101-swift-meets-cocoa-touch-framework>
- [11] Apple Inc.: Xcode Essentials - At a Glance [online]. 2016, [cit. 2016-02-07]. Dostupné z: https://developer.apple.com/library/ios/documentation/ToolsLanguages/Conceptual/Xcode_Overview/
- [12] Apple Inc.: Model-View-Controller [online]. 2015, [cit. 2016-03-11]. Dostupné z: <https://developer.apple.com/library/ios/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html>
- [13] Apple Inc.: iOS Human Interface Guidelines: Navigation [online]. 2016, [cit. 2016-03-11]. Dostupné z: <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/Navigation.html>
- [14] Thompson, M.: Alamofire [online]. 2014, [cit. 2016-04-24]. Dostupné z: <http://nshipster.com/alamofire/>
- [15] Apple Inc.: Auto Layout Guide: Understanding Auto Layout [online]. 2016, [cit. 2016-04-24]. Dostupné z: <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/AutolayoutPG/>
- [16] Apple Inc.: CLLocationManager Class Reference [online]. 2016, [cit. 2016-04-28]. Dostupné z: https://developer.apple.com/library/ios/documentation/CoreLocation/Reference/CLLocationManager_Class/
- [17] Kolář, J.: Praha: České vysoké učení technické, 2009, ISBN 978-80-01-04331-8, s. 116–119, [cit. 2016-05-05].
- [18] Rocchi, C.: Overview of iOS Crash Reporting Tools [online]. 2013, [cit. 2016-05-04]. Dostupné z: <https://www.raywenderlich.com/33669/overview-of-ios-crash-reporting-tools>

Seznam použitých zkratk

API Application Programming Interface

GPS Global Positioning System

HIG Human Interface Guidelines

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

HUD Head-Up Display

JSON JavaScript Object Notation

MVC Model-View-Controller

ODF Optical Distribution Frame

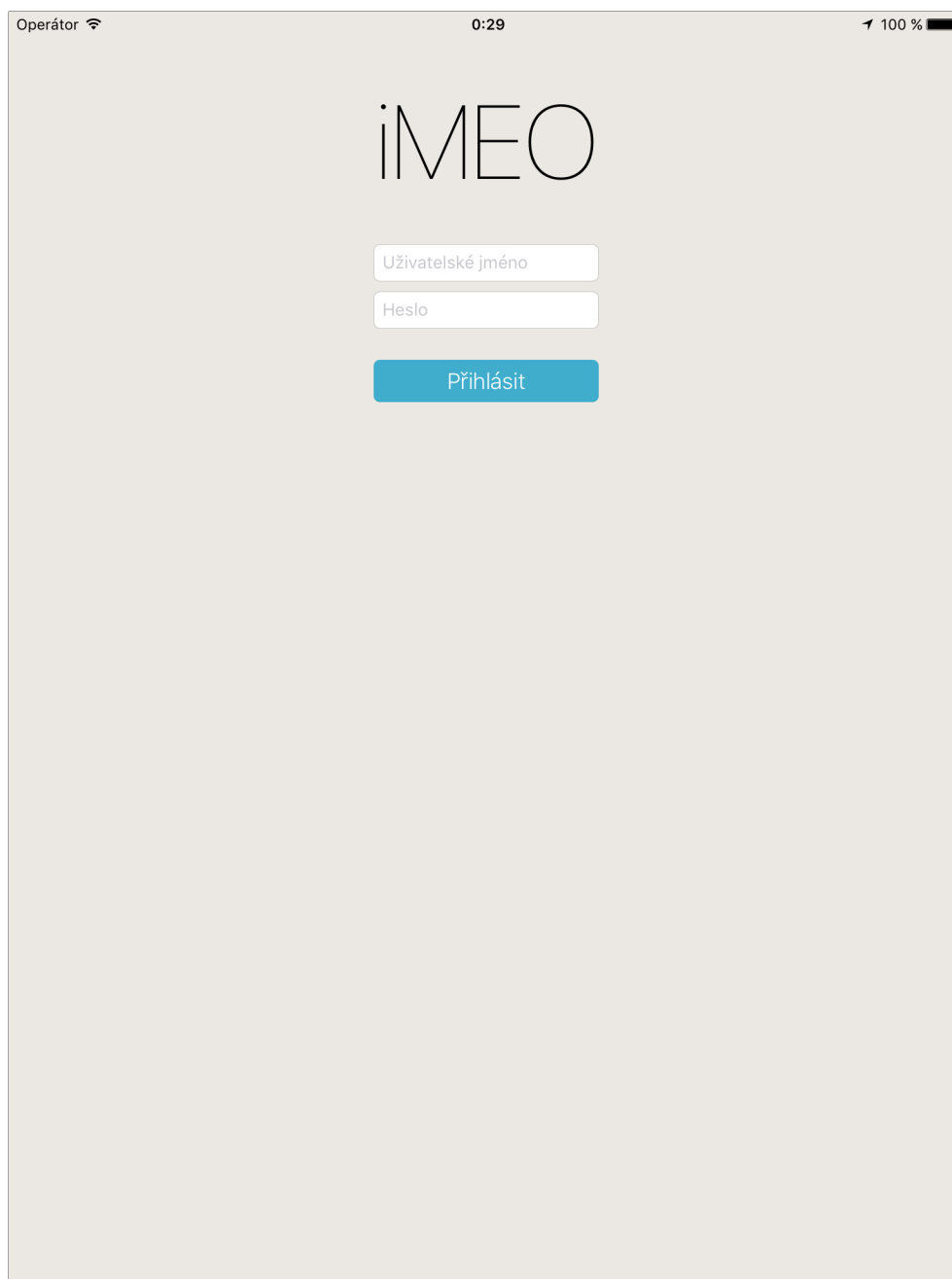
POP Point of Presence

REST Representational State Transfer

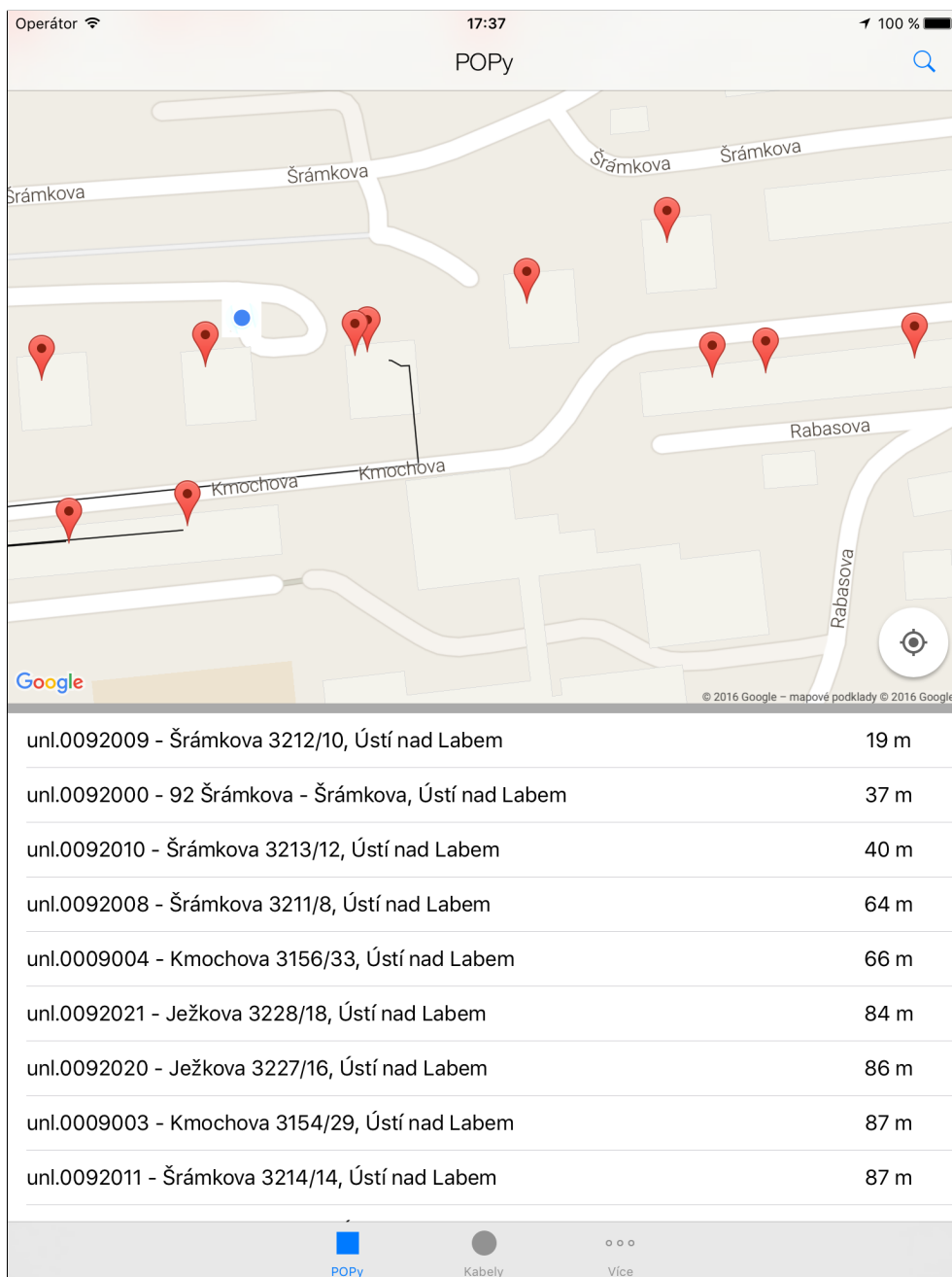
SDK Software Development Kit

Ukázky obrazovek aplikace

B. UKÁZKY OBRAZOVEK APLIKACE

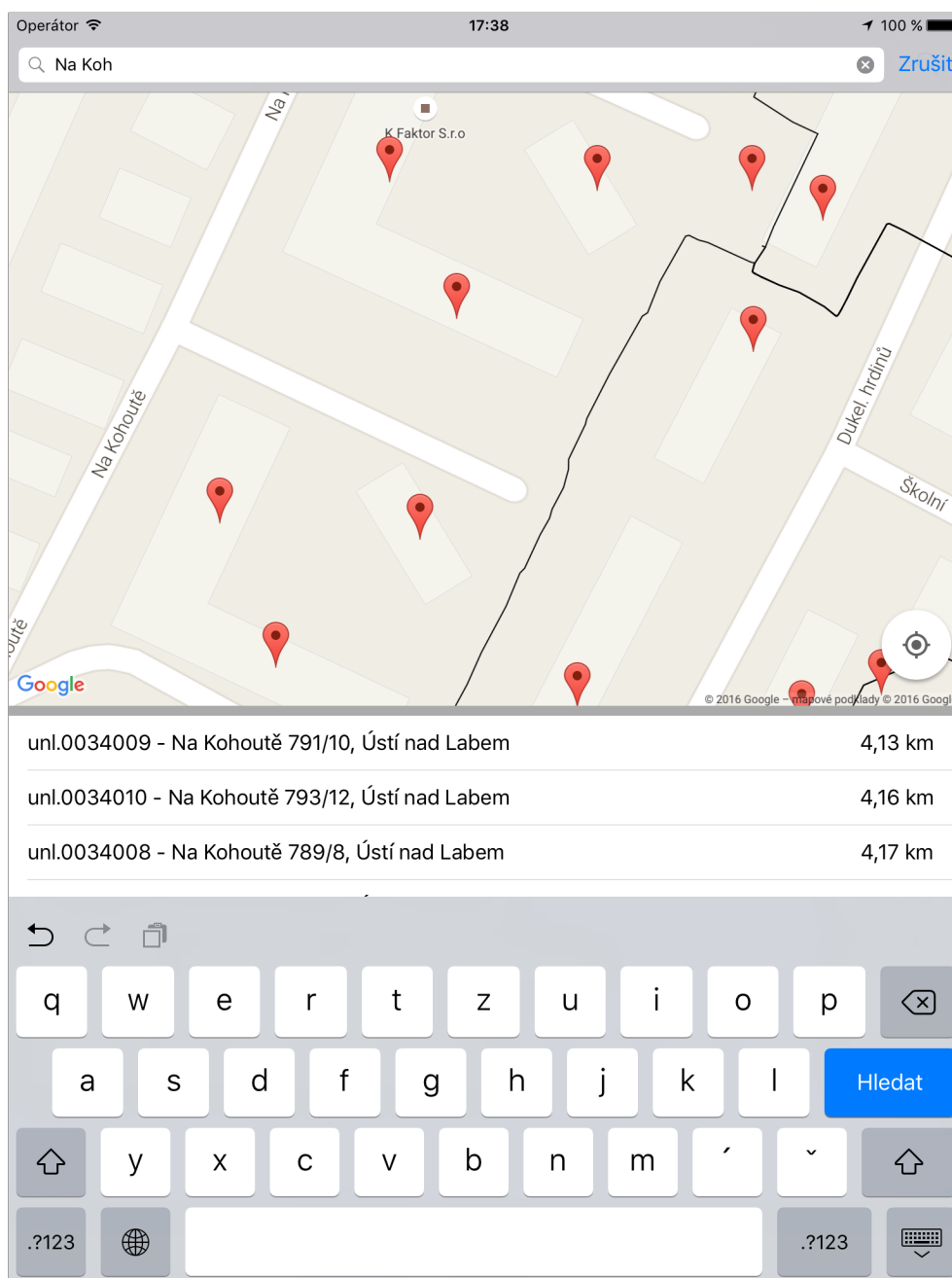


Obrázek B.1: Přihlašovací obrazovka



Obrázek B.2: Zobrazení mapy POPů a kabelů

B. UKÁZKY OBRAZOVEK APLIKACE



Obrázek B.3: Zobrazení mapy POPů a kabelů s vyhledáváním

Operátor		17:39		100 %	
POPy		unl.0082000			
Mírové náměstí 1/1, Ústí nad Labem Palác Zdar					
Koncové body					
1	organizér	10 kazet, 0 pozic	Žádná poznámka		
2	ODF	20 adaptérů	Žádná poznámka		
3	organizér	1 kazeta, 1 pozice	4. NP, pouze TKR		
Optické kabely					
191	unl.0000302 - Garáže Dlouhá 3096/2, Ústí nad La...	unl.0082000 - 82 Zdar Mírové náměstí 1/1, Ústí na...	24 vláken single-mode	348 m	TETA
266	unl.0082000 - 82 Zdar Mírové náměstí 1/1, Ústí na...	unl.0082000 - 82 Zdar Mírové náměstí 1/1, Ústí na...	24 vláken single-mode	10 m	TETA
267	unl.0082000 - 82 Zdar Mírové náměstí 1/1, Ústí na...	unl.0082000 - 82 Zdar Mírové náměstí 1/1, Ústí na...	12 vláken single-mode	108 m	TETA

Obrázek B.4: Detail POPu

B. UKÁZKY OBRAZOVEK APLIKACE

Operátor 0:21 100%

unl.0071000 unl.0071000/1

unl.0071000 - 71 SNP
 SNP, Ústí nad Labem
 15 kazet, bez SECO prvků
 Raycom, organizér
 SM+MM

pohled na organizér pohled na ODF pohled na vývody

11. kazeta

6	unl.0073000/1 - 73 Panorama unl.0073000/1 - 73 Panorama	194 2 12	315 m SECO	T				
5	unl.0073000/1 - 73 Panorama unl.0073000/1 - 73 Panorama	194 2 11	315 m SECO	T			15	
4	unl.0073000/1 - 73 Panorama unl.0073000/1 - 73 Panorama	194 2 10	315 m SECO	T			14 13 12	
3	unl.0073000/1 - 73 Panorama unl.0073000/1 - 73 Panorama	194 2 9	315 m SECO	T			11 10 9	
2	unl.0073000/1 - 73 Panorama unl.0073000/1 - 73 Panorama I2-73 Fa0/22	194 2 8	315 m SECO	svár T	189 m SECO	71022 1 2	unl.0073011/1 unl.0073011/1 I3-73-22 Gi0/1	8 7 6
1	unl.0073000/1 - 73 Panorama unl.0073000/1 - 73 Panorama I2-73 Fa0/22	194 2 7	315 m SECO	svár T	189 m SECO	71022 1 1	unl.0073011/1 unl.0073011/1 I3-73-22 Gi0/1	5 4 3

10. kazeta

12				T			svár na: Pigtail č. 36 konektor: LC/PC	1
11				T			svár na: Pigtail č. 36 konektor: LC/PC	0
10				T			svár na: Pigtail č. 35 konektor: LC/PC	

POPy Kably Více

Obrázek B.5: Pohled na organizér

Operátor 17:43 100%

unl.0071000 unl.0071000/1

unl.0071000 - 71 SNP

SNP, Ústí nad Labem

15 kazet, bez SECO prvků

Raycom, organizér

SM+MM

pohled na organizér | **pohled na ODF** | pohled na vývody

1A	patchkord pro: I2-71 Fa1/0/1 konektory: MT-RJ	adaptér: LC/PC kazeta: 5 pozice: 1	404 m SECO 7	71001 0 7	unl.0071001/1 unl.0071001/1 I3-71-01 Gi0/1
1B	patchkord pro: I2-71 Fa1/0/1 konektory: MT-RJ	adaptér: LC/PC kazeta: 5 pozice: 2	404 m SECO 8	71001 0 8	unl.0071001/1 unl.0071001/1 I3-71-01 Gi0/1
2B	patchkord pro: nezapojený konektory: MT-RJ	adaptér: LC/PC kazeta: 5 pozice: 8	463 m SECO 10	71002 0 10	unl.0071002/1 unl.0071002/1 žádné obsazení
2A	patchkord pro: nezapojený konektory: MT-RJ	adaptér: LC/PC kazeta: 5 pozice: 7	463 m SECO 9	71002 0 9	unl.0071002/1 unl.0071002/1 žádné obsazení
3B	patchkord pro: I2-71 Fa1/0/3 konektory: MT-RJ	adaptér: LC/PC kazeta: 5 pozice: 12	209 m SECO 10	71003 0 10	unl.0071003/1 unl.0071003/1 I3-71-03 Gi0/1
3A	patchkord pro: I2-71 Fa1/0/3 konektory: MT-RJ	adaptér: LC/PC kazeta: 5 pozice: 11	209 m SECO 9	71003 0 9	unl.0071003/1 unl.0071003/1 I3-71-03 Gi0/1
4B	patchkord pro: I2-71 Fa1/0/4 konektory: MT-RJ	adaptér: LC/PC kazeta: 6 pozice: 4	726 m SECO 10	71004 0 10	unl.0071004/1 unl.0071004/1 I3-71-04 Gi0/1
4A	patchkord pro: I2-71 Fa1/0/4 konektory: MT-RJ	adaptér: LC/PC kazeta: 6 pozice: 3	726 m SECO 9	71004 0 9	unl.0071004/1 unl.0071004/1 I3-71-04 Gi0/1
5A	patchkord pro: I2-71 Fa1/0/5 konektory: MT-RJ	adaptér: LC/PC kazeta: 6 pozice: 7	822 m SECO 9	71005 0 9	unl.0071005/1 unl.0071005/1 I3-71-05 FastEthernet 0/12 (MC)
	patchkord pro: I2-71 Fa1/0/5	adaptér: LC/PC	822 m	71005	unl.0071005/1

POPy | Kabely | Více

Obrázek B.6: Pohled na ODF

B. UKÁZKY OBRAZOVEK APLIKACE

Operátor 0:29 100 %

< unl.0071000 unl.0071000/1

unl.0071000 - 71 SNP

SNP, Ústí nad Labem

15 kazet, bez SECO prvků

Raycom, organizér

SM+MM

pohled na organizér pohled na ODF **pohled na vývody**

36	pigtail	kazeta 10 / pozice 12	LC/PC	
36	pigtail	kazeta 10 / pozice 11	LC/PC	
35	pigtail	kazeta 10 / pozice 10	LC/PC	adaptér 18B
35	pigtail	kazeta 10 / pozice 9	LC/PC	adaptér 18A
34	pigtail	kazeta 10 / pozice 8	LC/PC	
34	pigtail	kazeta 10 / pozice 7	LC/PC	
33	pigtail	kazeta 10 / pozice 6	LC/PC	adaptér 17A/nezapojený
33	pigtail	kazeta 10 / pozice 5	LC/PC	adaptér 17B/nezapojený
32	pigtail	kazeta 10 / pozice 4	LC/PC	nezapojený
32	pigtail	kazeta 10 / pozice 3	LC/PC	nezapojený
31	pigtail	kazeta 10 / pozice 2	LC/PC	adaptér 16B/12-71 Fa1/0/16
31	pigtail	kazeta 10 / pozice 1	LC/PC	adaptér 16A/12-71 Fa1/0/16
30	pigtail	kazeta 9 / pozice 12	LC/PC	nezapojený
30	pigtail	kazeta 9 / pozice 11	LC/PC	nezapojený
29	pigtail	kazeta 9 / pozice 10	LC/PC	adaptér 15B/12-71 Fa1/0/15

POPy Kabely Více

Obrázek B.7: Pohled na vývody

Operátor 17:47 100%

unl.0062000/1 unl.0062000/1 - unl.0062013/1

unl.0062000/1 - 62 ELBA
ELBA, Ústí nad Labem
nezapojený

unl.0062013/1
Šumavská 3105/13, Ústí nad Labem
nezapojený

451.0 m, multi-mode
útlum 1310 nm: 1.67785 dB útlum 1550 nm: 1.61922 dB

Trasa

ostatní obsazení	nezapojený	ELBA
pigtail	číslo 26, konektor LC/PC	
organizér	organizér unl.0062000/1	ELBA
optický kabel	kabel č. 62 010 (163 m), vlákno č. 16, barv...	
organizér	organizér unl.0062010/1	Šumavská 2949/1
optický kabel	kabel č. 62 011 (101 m), vlákno č. 16, barva...	
organizér	organizér unl.0062011/1	Šumavská 2951/5
optický kabel	kabel č. 62 012 (101 m), vlákno č. 16, barv...	

Google © 2016 Google - mapové podklady © 2016 Google













POPy Kabely Více

Obrázek B.8: Trasa vlákna

B. UKÁZKY OBRAZOVEK APLIKACE

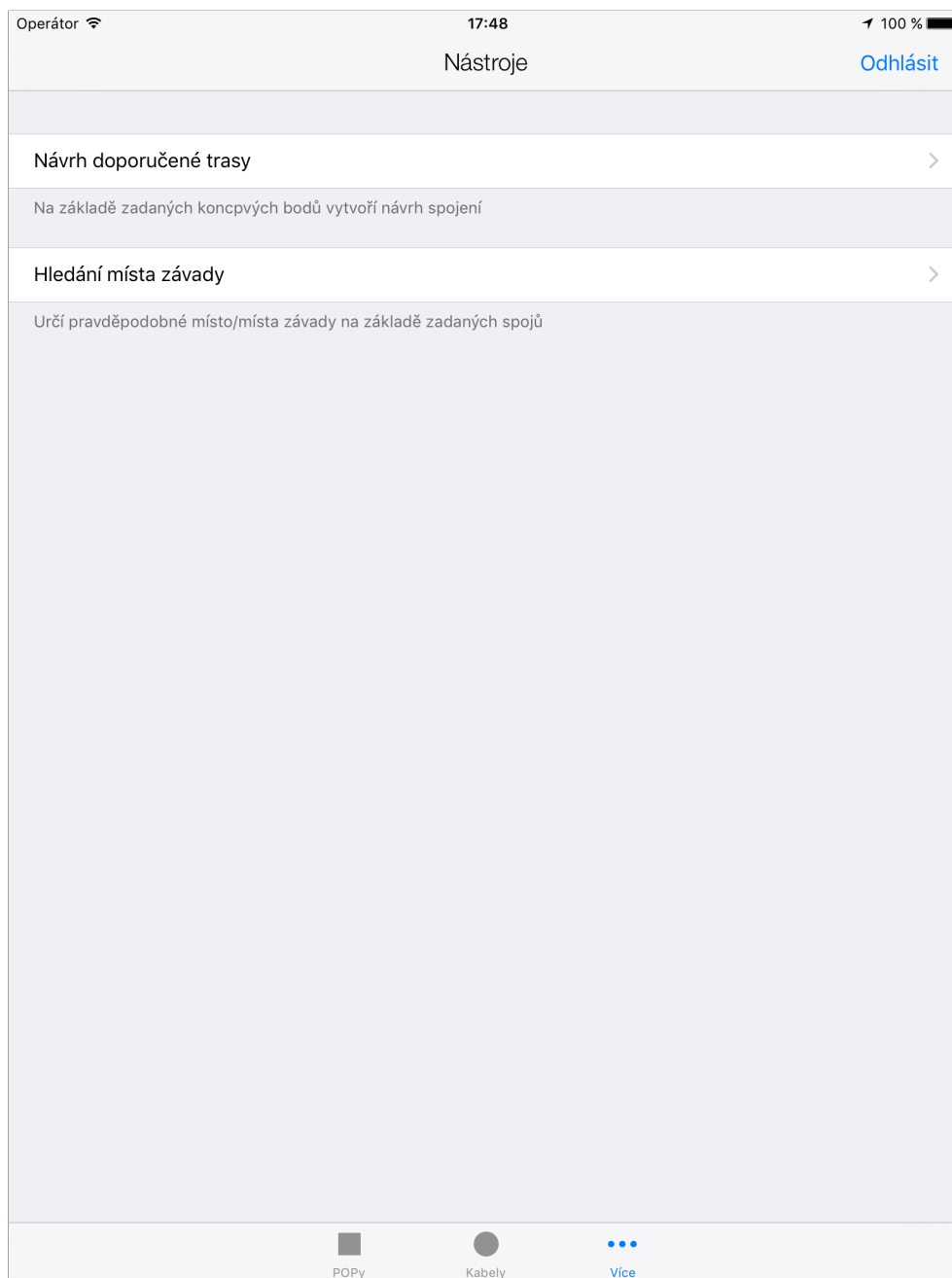
Operátor						17:44	100 %
Kabely							
1	unl.0010192 - Klíše OCEF B Klíšská 977/77, Ústí nad La...	unl.0000301 - KHS Klíšská, Ústí nad Labem	144 vláken single-mode	1 447 m	TETA		
2	unl.0025100 Vojanova, Ústí nad Labem	unl.0025000 - 25 V Oblouku V Oblouku 589/23, Ústí na...	24 vláken single-mode	275 m	TETA		
3	unl.0000301 - KHS Klíšská, Ústí nad Labem	unl.0081000 - 81 Masaryko... Masarykova 3085/12, Ústí...	96 vláken single-mode	545 m	TETA		
4	unl.0081000 - 81 Masaryko... Masarykova 3085/12, Ústí...	unl.0000303 - Pozemní sta... K Národnímu domu, Ústí na...	96 vláken single-mode	559 m	TETA		
5	unl.0000303 - Pozemní sta... K Národnímu domu, Ústí na...	unl.0007000 - 07 Velká Hra... Velká Hradební 2320/82, Ú...	96 vláken single-mode	291 m	TETA		
7	unl.0000104 - OS-Hrnčísk... Hrnčíská, Ústí nad Labem	unl.0000107 - OS-Pincova kolektor, Ústí nad Labem	48 vláken single-mode	2 433 m	TETA		
8	unl.0000102 - OS-Dobětice... Dobětická, Ústí nad Labem	unl.0000307 - T-Mobile Výstupní, Ústí nad Labem	96 vláken single-mode	541 m	TETA		
9	unl.0000101 - OS-Dobětice I. Dobětická, Ústí nad Labem	unl.0000102 - OS-Dobětice... Dobětická, Ústí nad Labem	24 vláken single-mode	348 m	TETA		
10	unl.0000111 - OS-Stříbrníky... Jizerská, Ústí nad Labem	unl.0000977 - Hala SKP Stříbrnická 3131/9, Ústí nad...	12 vláken single-mode	692 m	TETA		
11	unl.0019001 Bratří Mrštíků 83/4, Ústí na...	unl.0000104 - OS-Hrnčísk... Hrnčíská, Ústí nad Labem	72 vláken single-mode	1 792 m	TETA		
12	tel.0001100 - Obecní Úřad Telnice 77, Telnice	tel.0000013 Telnice 139, Telnice	8 vláken single-mode	197 m	TETA		
14	unl.0000306 - Krásné Břez... Vojanova 805, Ústí nad Lab...	unl.0020000 - 20 Dvojdómí Dvojdómí 611/2, Ústí nad La...	12 vláken single-mode	99 m	TETA		
15	unl.0000306 - Krásné Břez... Vojanova 805, Ústí nad Lab...	unl.0026000 - 26 Anežky Č... Anežky České, Ústí nad La...	24 vláken single-mode	596 m	TETA		
16	unl.0000306 - Krásné Břez... Vojanova 805, Ústí nad Lab...	unl.0025100 Vojanova, Ústí nad Labem	12 vláken single-mode	202 m	TETA		
18	unl.0000104 - OS-Hrnčísk... Hrnčíská, Ústí nad Labem	unl.0007000 - 07 Velká Hra... Velká Hradební 2320/82, Ú...	48 vláken single-mode	273 m	TETA		
19	unl.0000303 - Pozemní sta... K Národnímu domu, Ústí na...	unl.0000104 - OS-Hrnčísk... Hrnčíská, Ústí nad Labem	96 vláken single-mode	270 m	TETA		

Obrázek B.9: Zobrazení seznamu kabelů

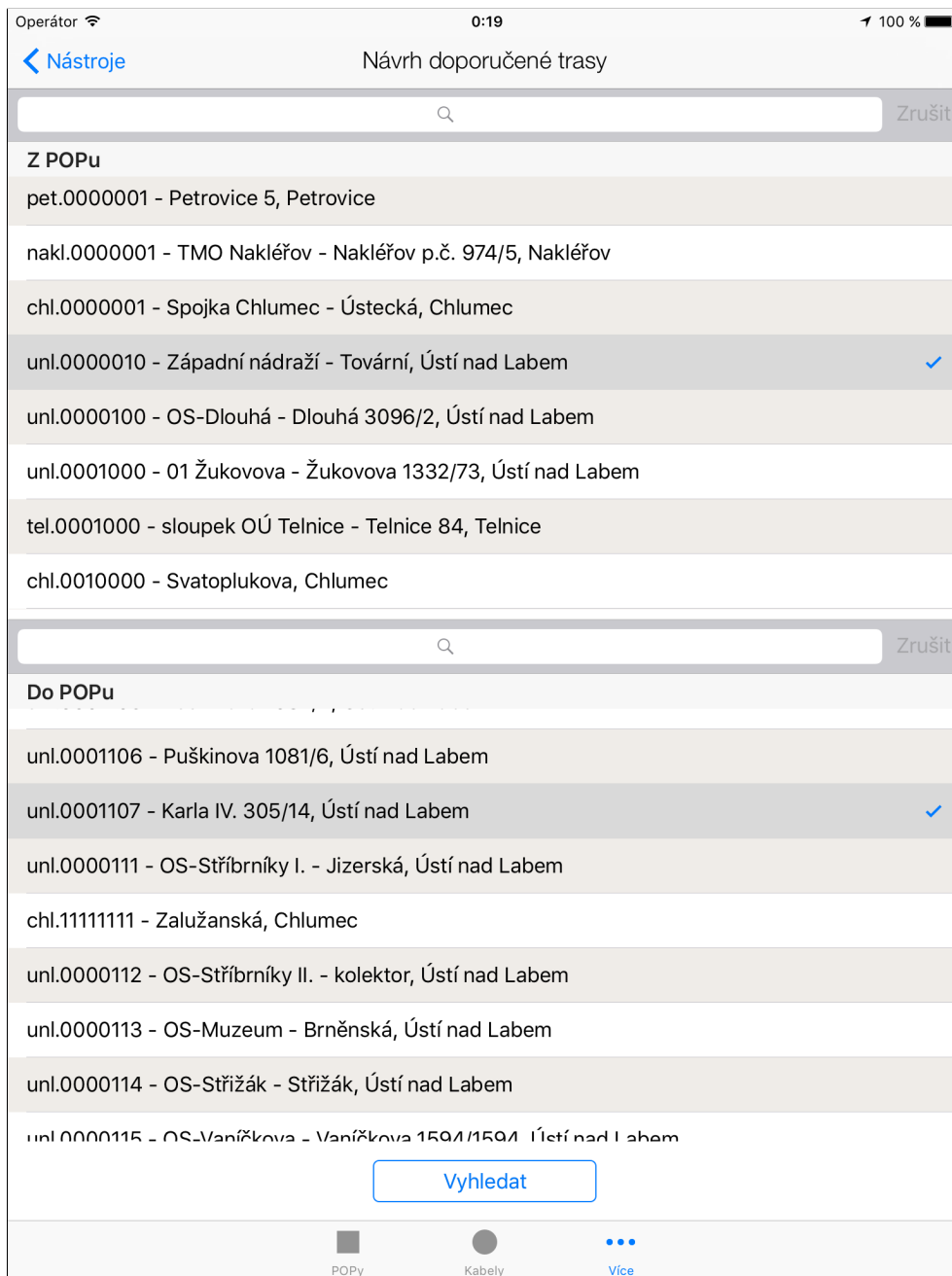
Operátor		17:44	100 %
Kabely		Kabel č. 9	
Samsung 6x4 - 24 vláken (18 volných), single-mode 348 m unl.0000101 - OS-Dobětice I. - Dobětická, Ústí nad Labem unl.0000102 - OS-Dobětice II. - Dobětická, Ústí nad Labem TETA			
Vlákna v kabelu			
1/1 - SM 	unl.0030100/2 - Klíše 301 Klíšská 977/77, Ústí nad Labem adaptér 45	unl.0009000/1 - 09 Kmochova Kmochova 3148/17, Ústí nad Labem vstup splitteru 1	
2/1 - SM 	unl.0010192/2 - Klíše OCEF B Klíšská 977/77, Ústí nad Labem	unl.0009000/1 - 09 Kmochova Kmochova 3148/17, Ústí nad Labem	
3/1 - SM 	unl.0010000/2 - Datové centrum Klíšská 977/77, Ústí nad Labem central2 Gi2/1	unl.0009000/1 - 09 Kmochova Kmochova 3148/17, Ústí nad Labem	
4/1 - SM 	unl.0010000/2 - Datové centrum Klíšská 977/77, Ústí nad Labem central2 Gi2/1	unl.0009000/1 - 09 Kmochova Kmochova 3148/17, Ústí nad Labem	
5/1 - SM 	unl.0010000/2 - Datové centrum Klíšská 977/77, Ústí nad Labem central2 Gi2/10	unl.0093000/1 - 93 Poláčkova Poláčkova, Ústí nad Labem I2-93 Gi0/1 (ukončen)	
6/1 - SM 	unl.0010000/2 - Datové centrum Klíšská 977/77, Ústí nad Labem central2 Gi2/10	unl.0093000/1 - 93 Poláčkova Poláčkova, Ústí nad Labem I2-93 Gi0/1 (ukončen)	
7/2 - SM 	unl.0019000/2 - DC Střekov Bratří Mrštíkú 83/4, Ústí nad Labem central1-brm Te1/29	unl.0092000/1 - 92 Šrámkova Šrámkova, Ústí nad Labem d53 Te0/0/25	
8/2 - SM 	unl.0019000/2 - DC Střekov Bratří Mrštíkú 83/4, Ústí nad Labem central1-brm Te1/29	unl.0092000/1 - 92 Šrámkova Šrámkova, Ústí nad Labem d53 Te0/0/25	
9/2 - SM 	unl.0007000/1 - 07 Velká Hradební Velká Hradební 2320/82, Ústí nad Labem	unl.0009000/1 - 09 Kmochova Kmochova 3148/17, Ústí nad Labem	
10/2 - SM 	unl.0007000/1 - 07 Velká Hradební Velká Hradební 2320/82, Ústí nad Labem	unl.0009000/1 - 09 Kmochova Kmochova 3148/17, Ústí nad Labem	
11/2 - SM 	unl.0007000/1 - 07 Velká Hradební Velká Hradební 2320/82, Ústí nad Labem	unl.0009000/1 - 09 Kmochova Kmochova 3148/17, Ústí nad Labem	
12/2 - SM 	unl.0007000/1 - 07 Velká Hradební Velká Hradební 2320/82, Ústí nad Labem	unl.0009000/1 - 09 Kmochova Kmochova 3148/17, Ústí nad Labem	
	POPy	Kabely	Více

Obrázek B.10: Detail kabelu

B. UKÁZKY OBRAZOVEK APLIKACE



Obrázek B.11: Nástroje



Obrázek B.12: Návrh doporučené trasy

B. UKÁZKY OBRAZOVEK APLIKACE

Operátor 📶 0:11 📶 100%

[← Návrh doporučené trasy](#) unl.0000010 → unl.0001107

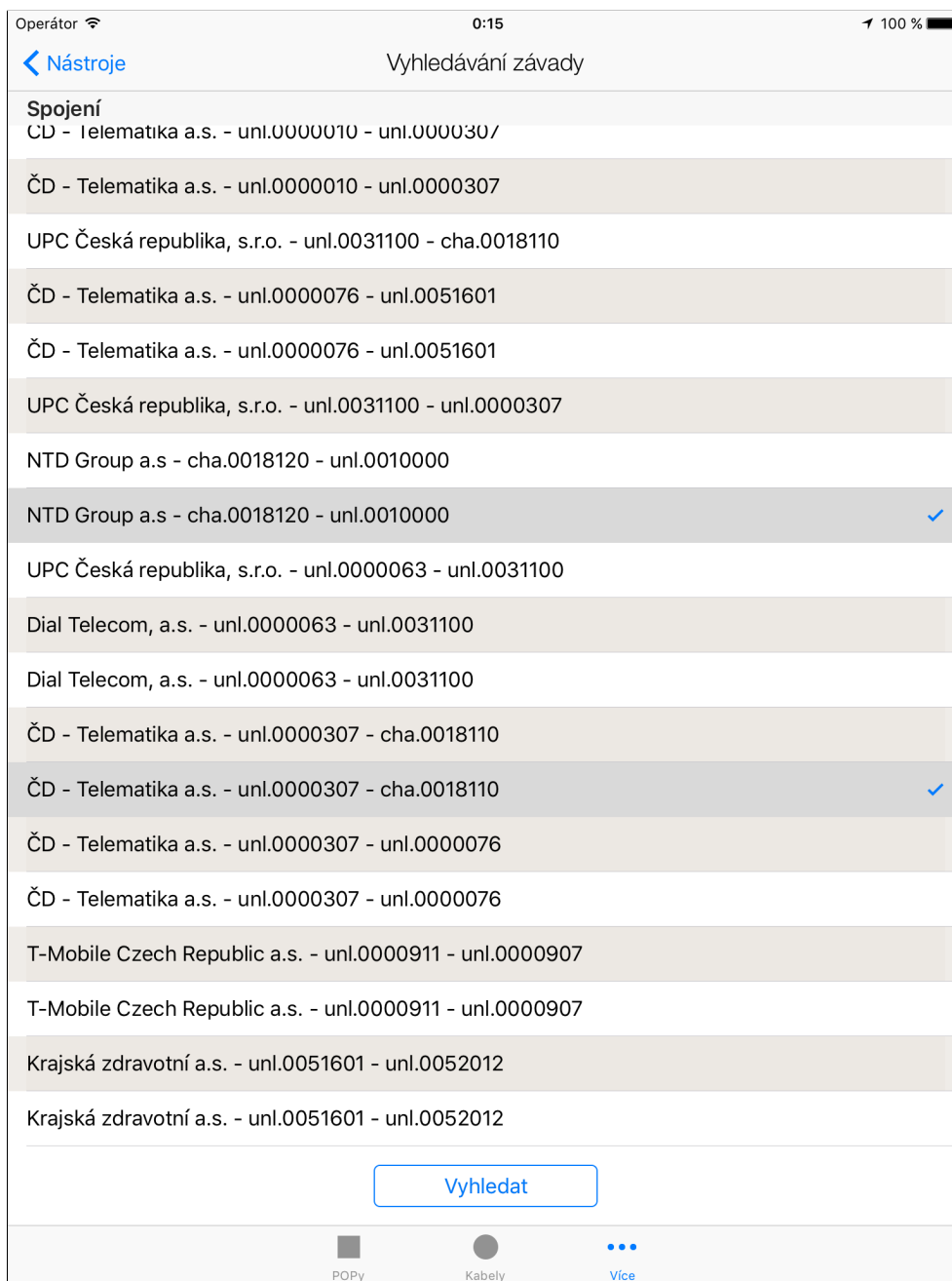
Návrh trasy

Kabel č. 295	unl.0000010 - Západní nádraží	→	unl.0000115 - OS-Vaničkova
Kabel č. 290	unl.0000115 - OS-Vaničkova	→	unl.0000911 - PČR SSČK-Lidické n...
Kabel č. 258	unl.0000911 - PČR SSČK-Lidické n...	→	unl.0000302 - Garáže
Kabel č. 189	unl.0000302 - Garáže	→	unl.0000100 - OS-Dlouhá
Kabel č. 40	unl.0000100 - OS-Dlouhá	→	unl.0000303 - Pozemní stavby
Kabel č. 19	unl.0000303 - Pozemní stavby	→	unl.0000104 - OS-Hrnčářská-kolek...
Kabel č. 11	unl.0000104 - OS-Hrnčářská-kolek...	→	unl.0019001
Kabel č. 29	unl.0019001	→	unl.0001000 - 01 Žukovova
Kabel č. 31	unl.0001000 - 01 Žukovova	→	unl.0001100

Google

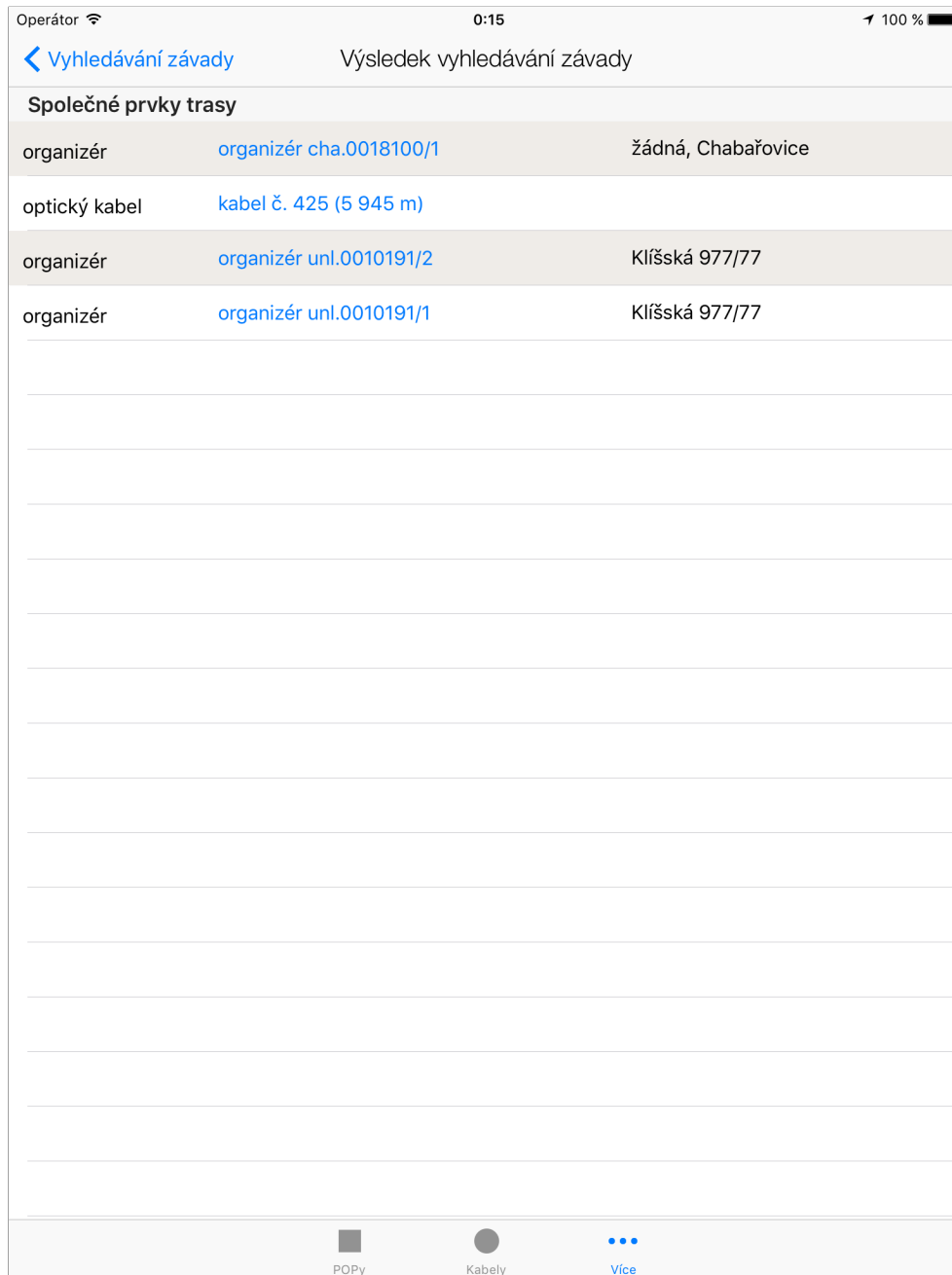
POPpy Kabely Více

Obrázek B.13: Výsledek návrhu doporučené trasy



Obrázek B.14: Vyhledávání závady

B. UKÁZKY OBRAZOVEK APLIKACE



Obrázek B.15: Výsledek vyhledávání závady

Obsah přiloženého flash disku

readme.txt.....	stručný popis obsahu flash disku
src	
├─ BP_Durcik_Jakub_2016.tex..	zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
├─ images/.....	obrázky
└─ *.....	další soubory šablony
text.....	text práce a zadání
├─ BP_Durcik_Jakub_2016.pdf	text práce ve formátu PDF
└─ ZZP_Durcik_Jakub_2016.pdf	zadání práce ve formátu PDF