

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe
Individual Professional Practice in the
Company

Zadání bakalářské práce

Student:

Petr Burdík

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: GIRITON Systems s.r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

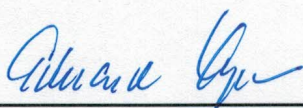
Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Martin Kot, Ph.D.**

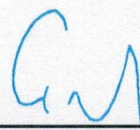
Konzultant bakalářské práce: Ing. Jan Gřeš, MSc.

Datum zadání: 01.09.2015

Datum odevzdání: 29.04.2016



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snašel, CSc.
děkan fakulty

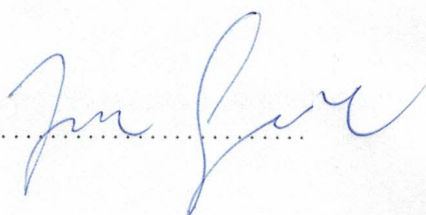
Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 25. dubna 2016

Petr Bunde
.....

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 25. dubna 2016

.....


Na této straně bych chtěl poděkovat všem, díky kterým mohla vzniknout tato bakalářská práce. Děkuji firmě GIRITON Systems s.r.o. za možnost absolvovat odbornou praxi a jejím zaměstnancům za všechnu pomoc v rámci této praxe. Dále bych chtěl poděkovat mému vedoucímu bakalářské práce ing. Martinu Kotovi, Ph.D. za konzultace při vytváření této práce.

Abstrakt

Cílem této bakalářské práce je popsat mé působení ve firmě GIRITON Systems s.r.o., kde jsem absolvoval individuální odbornou praxi v rámci této bakalářské práce. Odbornou praxi jsem absolvoval po dobu 50 dnů, kdy mou hlavní pracovní činností byla implementace GUI a následná implementace funkcí webových aplikací a Android aplikací Píchací hodiny GIRITON a Moje docházka GIRITON.

Klíčová slova: Individuální odborná praxe, GIRITON Systems s.r.o., Java, Apache Cordova, AngularJS, Onsen UI, Android, HTML, CSS, mobilní aplikace, webová aplikace

Abstract

The aim of this work is to describe my career with the company GIRITON Systems s.r.o., where I graduated individual professional practice in the context of this thesis. Professional experience I have completed a period of 50 days where my main job was to implement GUI functionality and the subsequent implementation of web applications and Android applications Píchací hodiny GIRITON and Moje docházka GIRITON.

Key Words: Individual professional practice, GIRITON Systems s.r.o., Java, Apache Cordova, AngularJS, Onsen UI, Android, HTML, CSS, phone application, web application

Obsah

Seznam použitých zkratk a symbolů	8
Seznam obrázků	9
1 Úvod	10
2 O společnosti	11
3 Pracovní zařazení	12
4 Píhací hodiny GIRITON verze pro Android	13
4.1 Agenda Plán směn	13
4.2 Agenda Vlastní docházka	14
4.3 Další úkoly	16
5 Píhací hodiny GIRITON verze pro webový prohlížeč	18
5.1 Agenda Plán směn	18
5.2 Agenda Vlastní docházka	19
5.3 Agenda Žádosti	20
5.4 Další úkoly	22
6 Moje docházka GIRITON verze pro mobilní zařízení	24
6.1 Agenda Přílohy projektu	24
6.2 Agenda Vlastní docházka	26
6.3 Agenda Služební cesta	27
6.4 Agenda Vlastní projekty	28
6.5 Další úkoly	30
7 Individuální úkoly	32
7.1 Vyhledání vhodných tabletů pro aplikaci	32
7.2 Využívání aplikací a návrhy na jejich vylepšení	32
8 Znalostí a dovedností scházející studentovi	33
9 Dosažené výsledky a celkové zhodnocení praxe	34
9.1 Dosažené výsledky	34
9.2 Celkové zhodnocení praxe	34
Literatura	35

Seznam použitých zkratek a symbolů

API	– application programming interface (rozhraní pro programování aplikace)
CSS3	– cascading style sheets 3 (kaskádové styly)
GPS	– global positioning system (globální polohovací systém)
GUI	– graphical user interface (grafické uživatelské rozhraní)
HTML5	– hypertext markup language 5
Ing	– inženýr
iOS	– iPhone operating system
NFC	– near field communication
REST	– representational state transfer
RFID	– radio frequency identification (identifikace na rádiové frekvenci)
RMI	– remote method invocation
sms	– short message service (služba krátkých textových zpráv)
sql	– structured query language (strukturovaný dotazovací jazyk)
s.r.o.	– společnost s ručením omezeným
tzv.	– takzvaný
URL	– uniform resource locator (jednotná adresa zdroje)
USB	– universal serial bus (univerzální sériová sběrnice)
VŠB - TUO	– Vysoká škola báňská - Technická univerzita Ostrava

Seznam obrázků

1	Hlavní screen píchacích hodin (Android)	13
2	Plán směn (Android) - přehled	14
3	Vlastní docházka (Android) - přehled	15
4	Vlastní docházka (Android) - Denní sumy	16
5	Plán Směn (Píchací hodiny) - přehled	19
6	Vlastní Docházka (Píchací hodiny) - přehled	20
7	Žádosti (Píchací hodiny) - seznam s žádostmi ke zpracování	21
8	Žádosti (Píchací hodiny) - detail žádosti	22
9	Přílohy (aplikace Moje Docházka) - přehled	25
10	Vlastní docházka (aplikace Moje Docházka)	27
11	Služební cesta (aplikace Moje Docházka)	28
12	Vlastní projekty (aplikace Moje Docházka)	29
13	Nový projekt a Sumy docházky(aplikace Moje Docházka)	30

1 Úvod

Odbornou individuální praxi ve firmě jsem si zvolil hlavně kvůli získání nových praktických zkušeností jako programátor s lidmi v týmu, kteří se denně věnují vývoji a implementaci aplikací. Chtěl jsem zjistit jaké to je pracovat pod vedením takových zkušených lidí a jak vypadá spolupráce v týmu. Dále jsem se chtěl podílet na vývoji komplexnějšího projektu, než na kterých jsem doposud pracoval při studiu na vysoké škole.

V rámci mé odborné individuální praxe ve firmě GIRITON Systems s.r.o. jsem měl možnost pracovat na informačním systému Docházka GIRITON. Jedná se o cloudový systém primárně určený ke sledování a plánování pracovních činností. Mým hlavním úkolem byla práce na aplikacích pracujících s tímto systémem. Při práci jsem se setkal s programovacím jazykem Java [1] se zaměřením na aplikace pro Android. Dále jsem převážně pracoval s frameworky Apache Cordova [2], AngularJS [3] a Onsen UI [4] při práci na mobilních a webových aplikacích.

Mojí hlavní prací bylo vytvářet grafické uživatelské rozhraní. Hlavní důraz při jeho tvorbě byl kladen na jeho jednoduchost a maximální přehlednost, aby se uživatel dokázal rychle a jednoduše v aplikaci zorientovat. Dalším krokem byla implementace jednotlivých funkcí grafického rozhraní. Jednalo se především o funkce týkající se prohlížení a editace dat cloudového docházkového systému.

2 O společnosti

GIRITON Systems s.r.o. je malá start-up společnost sídlící v podnikatelském inkubátoru, který se nachází v areálu VŠB - TUO. Její tým je tvořen skupinou programátorů, kteří vyvíjejí systém Docházka GIRITON. Jedná se o moderní online cloudový docházkový systém. Tento systém umožňuje jednoduše sledovat, evidovat a vyhodnocovat docházku uživatelů. Ve webové cloudové aplikaci lze pak všechna data přehledně zpracovat díky intuitivnímu rozhraní.

Společnost se v rámci systému Docházka GIRITON zabývá vývojem aplikací týkající se sledování a organizace práce. Aplikace Moje Docházka GIRITON po spárování s chytrým telefonem umožňuje uživateli zaznamenávat jeho docházku i s polohou. Dále aplikace umožňuje zaznamenávat průběh služebních cest, práci na projektech, či podávání žádosti o dovolenou, návštěvy doktora a podobně. Další aplikace, která společnost vyvíjí, jsou elektronické Píchnací hodiny GIRITON. Uživatel se přiložením RFID čipu k zařízení nebo zadáním pinu identifikuje a jeho docházka bude zařízením zaznamenána. Aplikace nesleduje pouze docházku uživatele, ale ten si zde může také zobrazit naplánované směny, odpracované dny, či podávat žádosti na dovolenou a podobně.

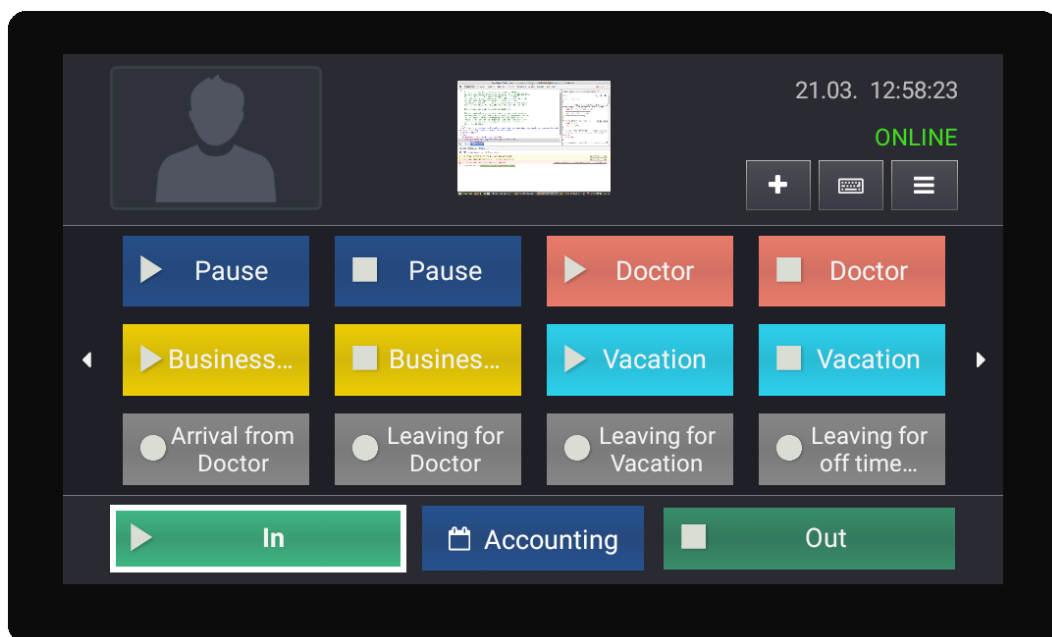
3 Pracovní zařazení

Při absolvování odborné praxe jsem měl možnost podílet se na vývoji mobilních a webových aplikací pod vedením Jana Gřeše. Během této doby jsem pracoval jako programátor a tester. Hlavní náplní bylo programovat funkce nových agend těchto aplikací a dále vytvořit pro ně vhodné a přehledné uživatelské rozhraní. Další fází vždy bylo testování nových nebo upravených agend, které zahrnovalo hledání chyb a jejich následné odstranění. Poslední fází vždy byla optimalizace dané agendy, aby vše správně a plynule pracovalo na všech zařízeních.

Mé působení na praxi se dá rozdělit na tři části, podle aplikací, na kterých jsem pracoval. V první části jsem pracoval na verzi Píchacích hodin GIRITON pro Android, na kterých jsem strávil přibližně dva měsíce. V druhé části mé praxe jsem začal pracovat na webové verzi Píchacích hodin GIRITON, kdy bylo cílem naprogramovat všechny dosavadní agendy z verze píchacích hodin pro Android. V poslední části mé praxe jsem pracoval na aplikaci Moje docházka GIRITON, ve které bylo mou hlavní činností upravovat agendy s cílem optimalizace nebo rozšíření dosavadních agend této aplikace.

4 Píchnací hodiny GIRITON verze pro Android

Tato aplikace slouží jako elektronické píchnací hodiny. Zařízení s touto aplikací bude umístěno na pracoviště, kde si uživatelé budou moci zaznamenávat svou docházku. Aplikace zaznamená zvolenou událost uživatele, až po jeho jednoznačné identifikaci pomocí bezkontaktního RFID nebo NFC čipu, či zadání osobního pinu uživatele. K záznamu může aplikace také pořídit fotografii a zaznamenat GPS souřadnice. Pokud zařízení nemá k dispozici připojení k internetu, všechna data jsou ukládána do paměti, ze které se po připojení k internetu sama odešlou. Na Obrázku 1 můžete vidět hlavní screen aplikace.



Obrázek 1: Hlavní screen píchnacích hodin (Android)

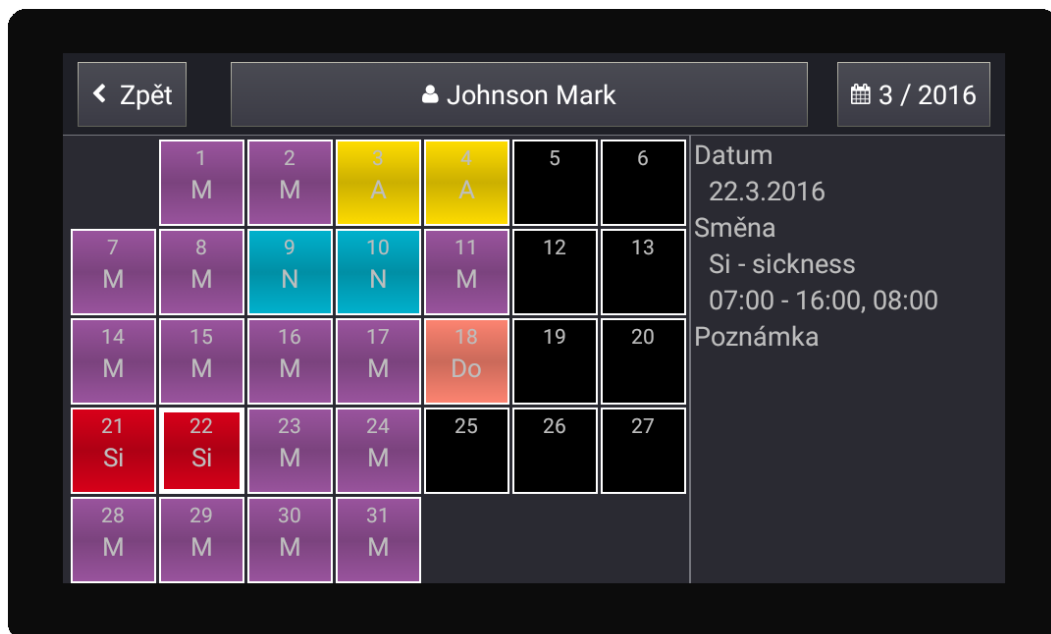
Aplikace také uživateli umožňuje prohlížet vlastní docházku a plán směn, popřípadě i jiných osob na základe přiděleného oprávnění. Další funkcí aplikace je podání žádostí, například o dovolenou, návštěvy doktora a podobně. Tyto žádosti může osoba s dostatečným oprávněním také v aplikaci zpracovávat.

Aplikace je psaná v programovacím jazyce Java [1]. Java je objektově orientovaný programovací jazyk. Vychází z programovacího jazyka C++, ale je jednodušší. Zdrojový kód je zkompilován do bytekódu, který může být poté spuštěn jakémkoliv zařízení, které má virtuální stroj javy.

4.1 Agenda Plán směn

Cílem této agendy je umožnit uživateli prohlížet plány směn. Hledal jsem proto komponentu, která by umožnila zobrazit graficky editovatelný kalendář. Tento úkol se mi nezdařil, proto jsem

se rozhodl vytvořit vlastní kalendář, kde budu moci zobrazovat požadovaná data. Klíčovými požadavky bylo, aby šly jednotlivé dny graficky odlišit a dalo se zde zobrazovat datum a informace o směně. Kalendář je vytvořen pomocí GridView, kdy jednotlivá buňka zastupuje jeden den. Zde jsem musel vyřešit, aby se nezobrazovaly buňky, které již do měsíce nepatří. Zjistím si proto, jaký den je prvního daného měsíce, a na základě této informace začnu vykreslovat dny od dané pozice. Dále si zjistím počet buněk potřebný k zobrazení všech dnu kalendáře. Na základě tohoto počtu určím počet řádků gridu. Na Obrázku 2 můžete vidět jak vypadá GUI této agendy.



Obrázek 2: Plán směn (Android) - přehled

Jednotlivým buňkám se potom přiřadí data získaná ze serveru. K datům přistupuji pomocí RMI API. RMI [5] je technologie programovacího jazyka Java, která nám umožňuje vytvářet objekty na jiném zařízení a náhledně na tyto objekty volat jejich metody. Volaná metoda vrací data uložena v mapě, která je ekvivalentem mapy, kterou vrací REST API na webové aplikaci. Stažená data si uložím do své vlastní struktury, ze které je čtu při vytváření jednotlivých dnů.

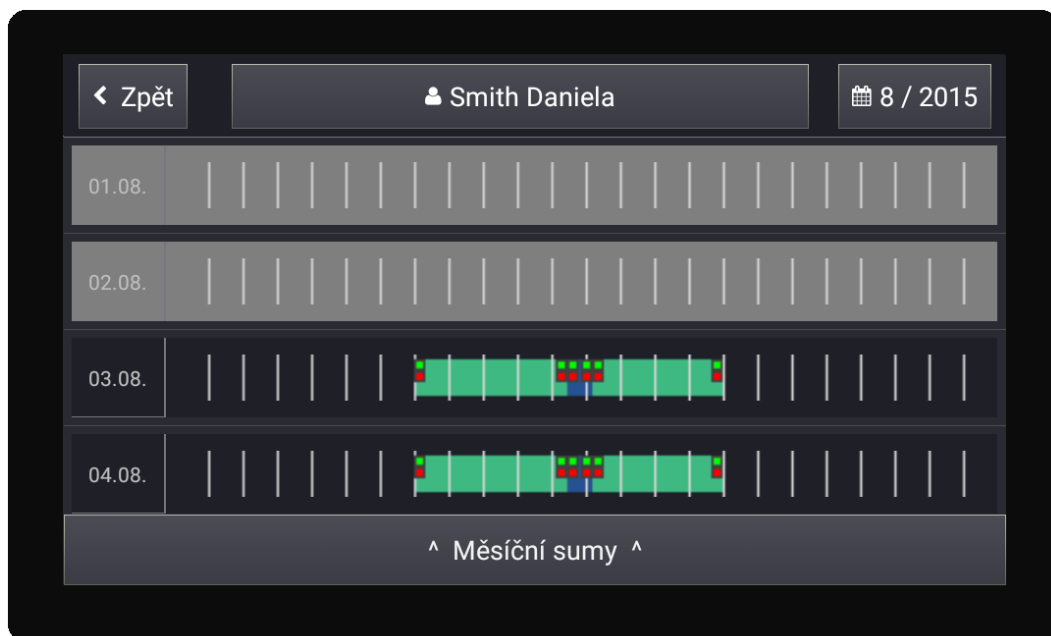
Každý den má přiřazený posluchač reagující na kliknutí. Při vyvolání eventů se zobrazí informace o dané směně v pravé části obrazovky. Při kliku na den se překreslí layout všech dnů, přičemž právě zvolený den má zvýrazněný okraj.

Při práci na této agendě jsem strávil 25 hodin. Nejvíce práce dalo grafické stylování jednotlivých dnů a naprogramování správného zobrazení kalendářních dnů.

4.2 Agenda Vlastní docházka

Cílem této agendy je umožnit uživateli prohlížet informace o docházce. Je zde zobrazený seznam všech dnů v měsíci, kdy se po kliku na den zobrazí doplňující informace. Pomocí RMI API stáhnou

data ze serveru na základě oprávnění uživatele. Data přijdou ve formě mapy, která obsahuje denní a měsíční sumy. Součástí těchto dat je i grafická reprezentace docházky, která se vytváří již na serveru z důvodu ušetření výkonu zařízení. Pro zobrazení dnů používám list, kdy jeden prvek listu je právě jeden den. Na Obrázku 3 můžete vidět jak vypadá GUI této agendy.



Obrázek 3: Vlastní docházka (Android) - přehled

Každý den má přiřazený posluchač. Po kliknutí na daný den se zobrazí element, který obsahuje informace o něm. Tento element je zde obsažen celou dobu, má pouze nastavenou výšku na 0 pixelů. Pro zobrazení se nastaví výška elementu tak, aby byl všechn jeho obsah viditelný. Jsou zde zobrazeny sumy aktivit daného dne a jednotlivé záznamy o docházce uživatele. Na Obrázku 4 můžete vidět jak se GUI změní po kliknutí na den.



Obrázek 4: Vlastní docházka (Android) - Denní sumy

Ve spodní části je umístěné tlačítko, které po stisknutí zobrazí element, obsahující spočtené sumy aktivit za daný měsíc. Zde se pro zobrazení mění poměr výšek elementu obsahujícího seznam dnů a elementu s měsíčními sumami. Po kliknutí se poměr 4:0 změní na 1:3.

Při práci na této agendě jsem se učil základy programování aplikace pro Android, a strávil jsem na ní 40 hodin.

4.3 Další úkoly

- Implementace vlastního tlačítka, kterému by šlo jednoduše nastavovat jeho text a ikonu. Zde jsem pracoval na úkolu společně s kolegou. Tímto tlačítkem jsme poté nahradili všechna dosavadní tlačítka, která obsahovala jak text tak ikonu.
- Hledání řešení, jak zobrazit softwarovou klávesnici nezávisle na tom, zda je či není připojena hardwarová klávesnice. Žádné funkční řešení se mi s kolegou nepodařilo najít.
- Oprava chyby se zobrazením dialogu na výběr data žádosti, kdy na starých verzích Androиду se zobrazoval prázdný dialog. Tento problém jsem vyřešil pomocí nastavení továrního zobrazení dialogu v jeho konstruktoru.
- Nahrazení elementu pro vybrání a zobrazení projektu na naše vlastní tlačítko, které umožňuje zobrazit ikonu tak i text. Přiřazení textu tlačítka na základě, zda je či není vybraný projekt.
- Vytváření automatizovaných testů aplikace pomocí frameworku Robotium [6]. Tento framework je velice jednoduchý na používání. Na daném screenu si získám jednotlivé elementy,

na které chci kliknout pomocí jejich unikátních identifikátorů. Poté jsem na prvky volal metodu, která simulovala klik. Na konci testu Robotium zobrazí jeho výsledky.

5 Píchací hodiny GIRITON verze pro webový prohlížeč

Tato aplikace je webovou verzí aplikace Píchací hodiny GIRITON. Poskytuje stejné funkce jako android aplikace. Mým úkolem při práci na této aplikaci bylo naimplementovat všechny dosavadní agendy z verze pro Android do webové verze aplikace.

Aplikace Píchací hodiny GIRITON verze pro webový prohlížeč a Osobní docházka online GIRITON jsou postaveny na Apache Cordova [2] frameworku sloužící k vývoji mobilních aplikací nezávisle na platformě. Aplikace je spuštěná tak, že je obalená wrapperem pro danou platformu a pracuje pouze s API dané platformy. Tento framework nám umožňuje používat technologie pro tvorbu webu jako jsou HTML5, CSS3 a JavaScript bez nutnosti optimalizace kódu pro danou platformu.

Jako další framework, se kterými jsem se v rámci těchto dvou aplikací setkal je AngularJS [3]. AngularJS je javascriptový webový framework, který umožňuje psát single-page aplikace. AngularJS automaticky synchronizuje data z uživatelského rozhraní (view) s JavaScriptovými objekty a naopak. Toto se nazývá 2 way data binding.

Poslední framework kterým jsem se během práce s těmito aplikacemi setkal, je Onsen UI [4]. Onsen UI je framework, který se zabývá optimalizací HTML5, CSS3 a JavaScriptu pro mobilní zařízení.

5.1 Agenda Plán směn

Tato agenda plní stejnou funkci jako agenda "Plán Směn" v android verzi píchacích hodin. Mým úkolem bylo naimplementovat tuto agendu do webových píchacích hodin. Pokusil jsem se vyhledat nějakou vhodnou komponentu, která by umožnila zobrazit kalendář, ale neúspěšně. Proto jsem začal implementovat vlastní kalendář, kdy jsem postupoval stejně jako v přídatě kalendáře v android verzi píchacích hodin. Výslednou implementaci GUI můžete vidět na Obrázku 5.

Data zde získávám pomocí http požadavku na metodu REST API, které předávám pin uživatele. Metoda vrátí seznam osob, které má uživatel oprávnění sledovat. Poté se na základě vybrané osoby a zvoleného data vytvoří další http požadavek na metodu REST API, která vrátí odpovídající data týkající se plánu směn.

Při práci na této agendě jsem strávil 16 hodin.

po	út	st	čt	pá	so	ne
	1 M	2 M	3 A	4 A	5	6
7 M	8 M	9 N	10 N	11 M	12	13
14 M	15 M	16 M	17 M	18 Do	19	20
21 Si	22 Si	23 M	24 M	25	26	27
28 M	29 M	30 M	31 M			

Datum:
03.03.2016

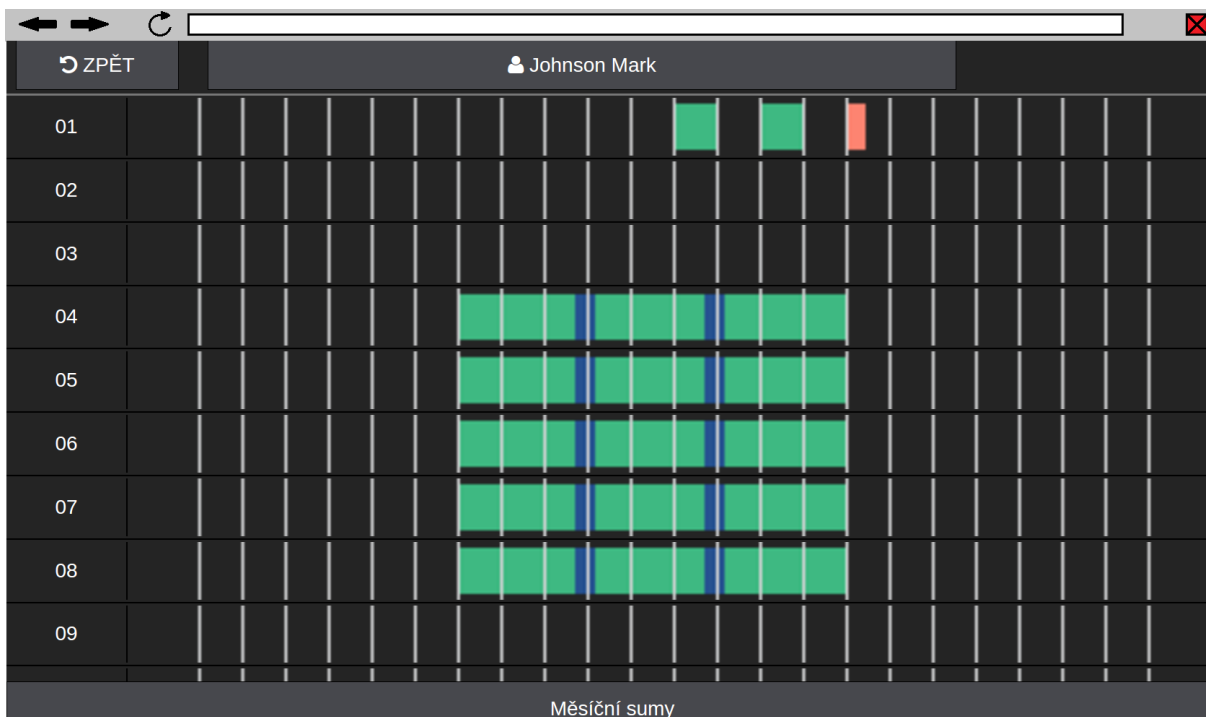
Směna:
A - Afternoon shift
13:00 - 22:00, 08:00

Poznámka:
-

Obrázek 5: Plán Směn (Píchnací hodiny) - přehled

5.2 Agenda Vlastní docházka

Cílem mé práce bylo naimplementovat agendu "Vlastní Docházka" z android verze píchnacích hodin. Přehled všech dnů zůstal stejný. Zde používám list a angularovskou metodu ng-repeat pro procházení pole s daty, kde jednotlivé prvky zastupují jeden den. Pro denní a měsíční sumy jsem vytvořil nové html šablony. Po kliknutí na prvek uchovávající jeden den se pomocí angularovské metody ng-click volá metoda, které se předávají data denních sum. Metoda slouží k vytvoření nové html stránky, kde se předaná data zobrazí. Pro zobrazení měsíčních sum je zde tlačítko. Po kliku na něj se volá metoda, které se předají data tykající se měsíčních sum. Data se poté zobrazí v nové html stránce. Na Obrázku 6 můžete vidět GUI agendy 'Vlastní Docházka'.



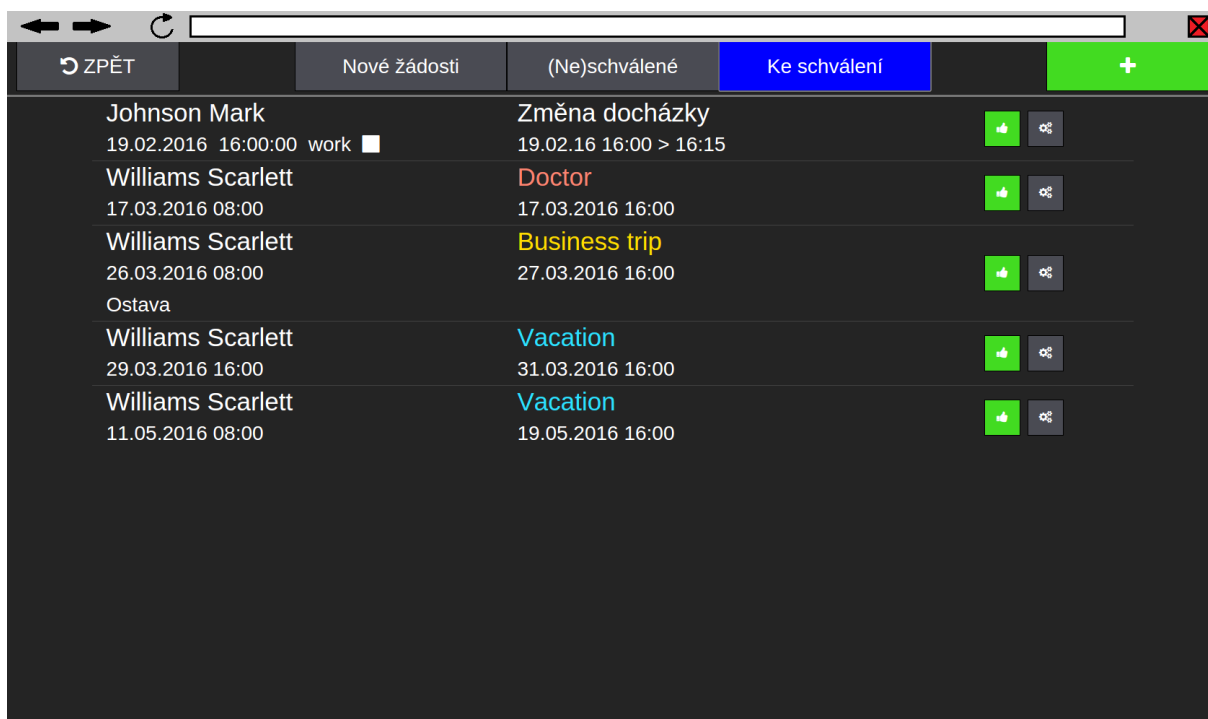
Obrázek 6: Vlastní Docházka (Píchnací hodiny) - přehled

Při práci na této agendě jsem strávil 24 hodin.

5.3 Agenda Žádosti

Cílem této agendy je umožnit uživateli podat si novou žádost a prohlížet si již existující žádosti. Dále můžeme osobám přidat oprávnění, která jim umožní také tyto žádosti zpracovávat, což zahrnuje jejich schválení, zamítnutí či smazání.

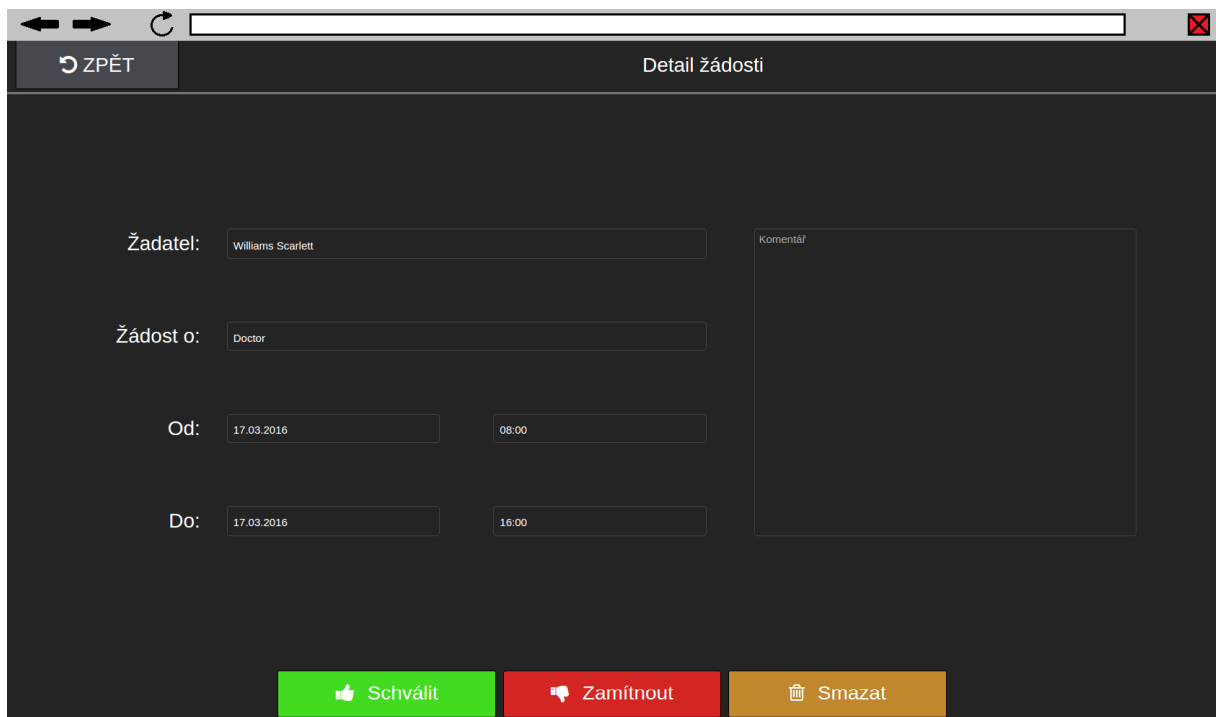
Při navrhování grafického uživatelského rozhraní šlo především o to, rozdělit jednotlivé žádosti do tří hlavních kategorií. Jedná se o nové žádosti, zpracované žádosti a žádosti ke schválení, kdy si uživatel bude moci jednoduše zvolit danou kategorii. Volbu zajišťují tlačítka umístěné v horní části obrazovky. Po vybrání kategorie se z dat vyberou právě dané žádosti. Na základě kategorie žádosti se jednotlivým prvkům nastavuje, zda budou viditelné, pomocí angulárovské funkce ng-show. Jedná se například o tlačítka potvrzení žádosti a zobrazení detailu žádosti, která jsou navíc viditelná v kategorii žádosti ke schválení na Obrázku 7. Pro získání dat pro tuto agendu se vytvoří http požadavek na metody REST API, které ze serveru vrací seznam všech zpracovaných žádostí a seznam žádostí ke zpracování.



Obrázek 7: Žádosti (Píchnací hodiny) - seznam s žádostmi ke zpracování

Na Obrázku 7 si můžete všimnout, že zde jsou dva druhy žádostí. Jeden je požadavek o aktivitu, jako je například dovolená, doktor a podobně. Druhý typ je požadavek o změně záznamu, jako je například příchod nebo odchod z práce. Každý objekt žádosti ovšem obsahuje odlišné argumenty. Proto jsem vytvořil dvě html šablony. Pro výběr šablony používám angulárovskou funkci ng-switch, která na základě typu žádosti vybere správnou šablonu. Šablona je poté vložená do aktuální html stránky pomocí angulárovské funkce ng-include.

Při zobrazení detailu žádosti se opět zohledňuje, o jaký typ žádosti se jedná. Na základě této informace se pomocí angulárovské funkce ng-show rozhoduje, které prvky se v html šabloně zobrazí. Tato šablona se také používá pro vytvoření nové žádosti. Na základě parametru, zda se jedná o novou žádost nebo o detailu žádosti, se nastaví jestli jsou prvky učeny pouze pro čtení. Pokud se jedná o novou žádost, tak se namísto textových polí u parametrů od a do zobrazí prvky, které umožňují zvolit datum a čas. Na Obrázku 8 můžete vidět detail žádosti určené ke zpracování.



Obrázek 8: Žádosti (Píchnací hodiny) - detail žádosti

Pro zpracování žádosti se vytváří http požadavek na metodu REST API, která provede změnu stavu žádosti. Uživatel má možnost žádost potvrdit, zamítnout nebo smazat. Každá z těchto voleb má na serveru přiřazenou konstantu, podle které se určí, o jaký požadavek se jedná. Při úspěšném zpracování tohoto požadavku, se pomocí callbacku volá metoda, která znovu stáhne a aktualizuje seznam žádostí. Při neúspěchu se provádí jiný callback na metodu, která uživateli zobrazí chybovou hlášku o neúspěšném provedení požadavku.

Při práci na této agendě jsem strávil 40 hodin.

5.4 Další úkoly

- Prováděl jsem změny ve stylování aplikace. Jednalo se o vytvoření a následného nahrazení všech záhlaví jednotlivých stránek. Cílem bylo sjednotit stylování celé aplikace, aby byla uživatelsky příjemnější a intuitivní. Záhlaví se proto skládá z tlačítka zpět a názvu aktuálního obsahu. Dalším krokem bylo sjednocení stylování všech tlačítek a úprava velikosti textu na základě rozlišení obrazovky.
- Vytvoření nového dialogu umožňující zobrazení jednotlivých agend aplikace. Jedna se o dialog s tlačítky, kdy každá položka odkáže uživatele na jím zvolenou agendu. Na hlavní stránce aplikace jsem také vytvořil tlačítko pro zobrazení tohoto dialogu.

- V rámci celé aplikace jsem přidával nový efekt na všechny interaktivní elementy. V každé šabloně jsem si proto vyhledal všechny takové elementy a následně jim přidal další parametr, který za pomoci pluginu daný efekt po kliknutí vykreslil.
- Zobrazit logo společnosti v hlavním screenu aplikace, pokud společnost má nějaké logo uložené. Aplikace každých pět minut posílá http požadavek na metodu REST API, který metodě předá aktuální kód loga. Pokud se kód neshoduje s kódem na serveru, logo bylo změněno a metoda vrátí nové logo ve formě base64. Jedná se o obrázek uložený ve formě textu. Pokud logo nebylo změněno nebo firma nemá logo uložené, metoda vrátí příslušnou zprávu. Jakmile je logo jednou staženo, je uloženo do lokální paměti aplikace, odkud je při každém spuštění načítáno.
- V agendě pro registraci nového čipu k uživateli jsem prováděl opravu špatného zobrazení seznamu s uživateli. Dále jsem zde prováděl korekci textů hlášek, které se zobrazují během procesu registrace čipu k uživateli. Cílem bylo jednoznačně oznámit uživateli, co má v dané chvíli dělat.

6 Moje docházka GIRITON verze pro mobilní zařízení

Tato aplikace je zaměřena pro uživatele, kteří se při své práci pohybují v terénu a neustále mění svou pozici. Díky chytrému mobilnímu telefonu má uživatel možnost si kdykoliv zaznamenat svou docházku. Pokud není připojen k internetu, aplikace si všechny záznamy uloží do paměti telefonu a jakmile je internetové připojení k dispozici, začne aplikace sama všechna shromážděna data odesílat.

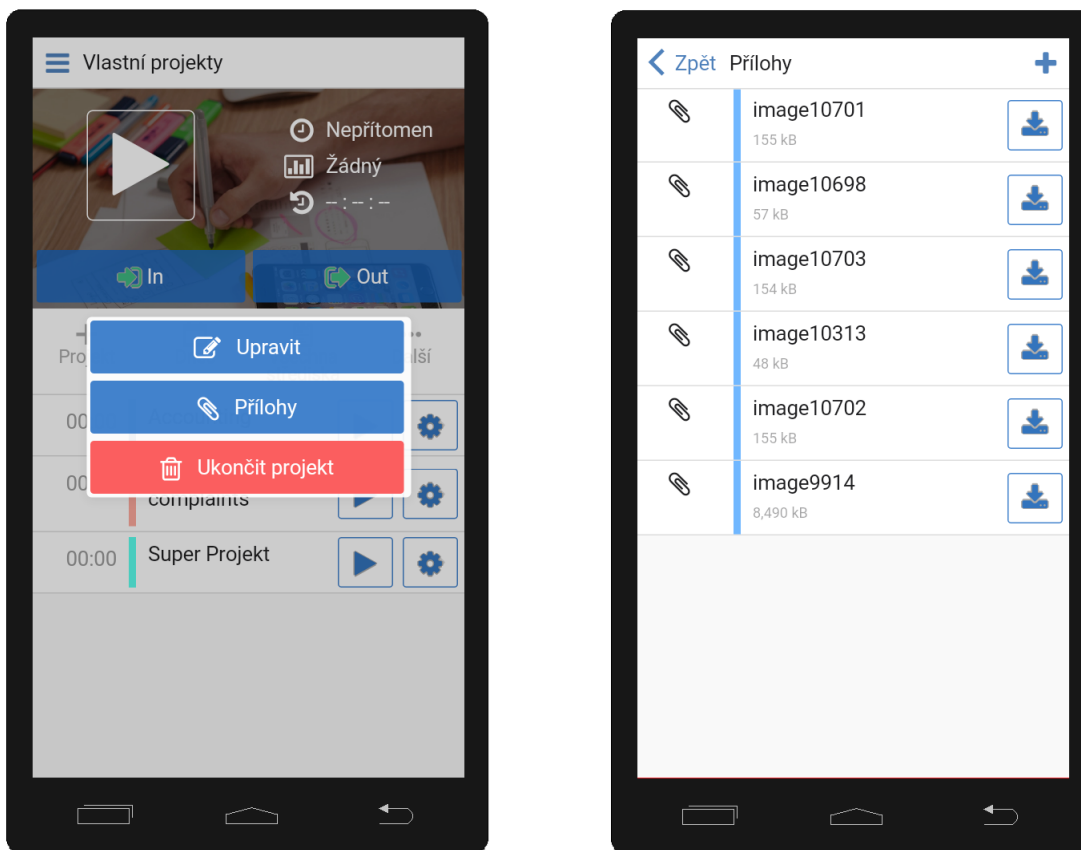
Aplikace umožňuje u jednotlivých záznamů docházky uživatele zaznamenávat aktuální polohu zařízení na základě GPS souřadnic. Dále u jednotlivých záznamů umožňuje pořádkem a následně přiložit fotku, psát uživateli vlastní komentáře a přikládat další přílohy.

Aplikace nabízí také plno dalších užitečných funkcí. Uživatel má možnost zaznamenávat služební cesty, kdy aplikace po celou dobu mapuje pohyb uživatele pomocí GPS. Další funkce se týkají prohlížení a spravování vlastní docházky či docházky jiných uživatelů. Dále uživatel může pracovat na vybraném projektu nebo podávat či schvalovat žádosti a to vše na základě přidělených oprávnění.

Jak už jsem se zmínil u aplikace Píhací hodiny GIRITON verze pro webový prohlížeč, tak aplikace Moje docházka GIRITON také využívá frameworky Apache Cordova, AngularJS a Onsen UI se kterými jsem pracoval.

6.1 Agenda Přílohy projektu

Cílem této agendy je umožnit uživateli si prohlížet seznam všech příloh, které se týkají zvoleného projektu. U jednotlivých příloh jsou vypsány základní informace o souboru. Jakoukoliv z těchto příloh si může uživatel jednoduše stáhnout do svého zařízení. Další funkce, kterou tato agenda nabízí, je nahrání nové přílohy k projektu. Uživatel si zvolí požadovaný soubor ze svého zařízení a odešle ho na server.



Obrázek 9: Přílohy (aplikace Moje Docházka) - přehled

Při navrhování grafického rozraní šlo hlavně o jednoduchost a přehlednost. V horní oblasti obrazovky je zobrazen název aktuální agendy, pro maximální orientaci v aplikaci. Dále je zde vpravo umístěno tlačítko pro nahrání nové přílohy. V seznamu příloh je u každé z nich uveden název, její velikost a komentář, pokud zde nějaký je. Každá příloha má pak své vlastní tlačítko, které umožňuje uživateli jednoduše daný soubor po kliknutí stáhnout do svého zařízení. To vše můžete vidět na Obrázku 9.

Údaje o přílohách jsou obsažena v datech týkajících se projektů, které se stahují v agendě "Vlastní projekty". Každý objekt zastupující projekt má argument attachments, ve kterém je uloženo pole objektů. Každý objekt obsažen v poli zastupuje informace právě o jedné příloze. Mé funkci na zobrazení projektu předám jako argument zvolený projekt.

Pro stažení souboru se vytvoří URL adresa, která se skládá z adresy serveru, cesty k metodě na REST API a následných parametrů pro danou metodu na serveru. Posléze se tato URL adresa otevře v defaultním prohlížeči a soubor se stáhne.

Pro nahrávání používám tři různé pluguny, díky kterým získám URL adresu souboru na zařízení. Musel jsem použít tři různé pluguny, jelikož aplikace běží na více platformách, kdy daný plugin je vybrán v závislosti na platformě. Jedná se o pluguny cordova-filechooser [7] pro Android, FilePicker-Phonegap-iOS-Plugin [8] pro iOS a cordova-imagePicker [9] pro Windows

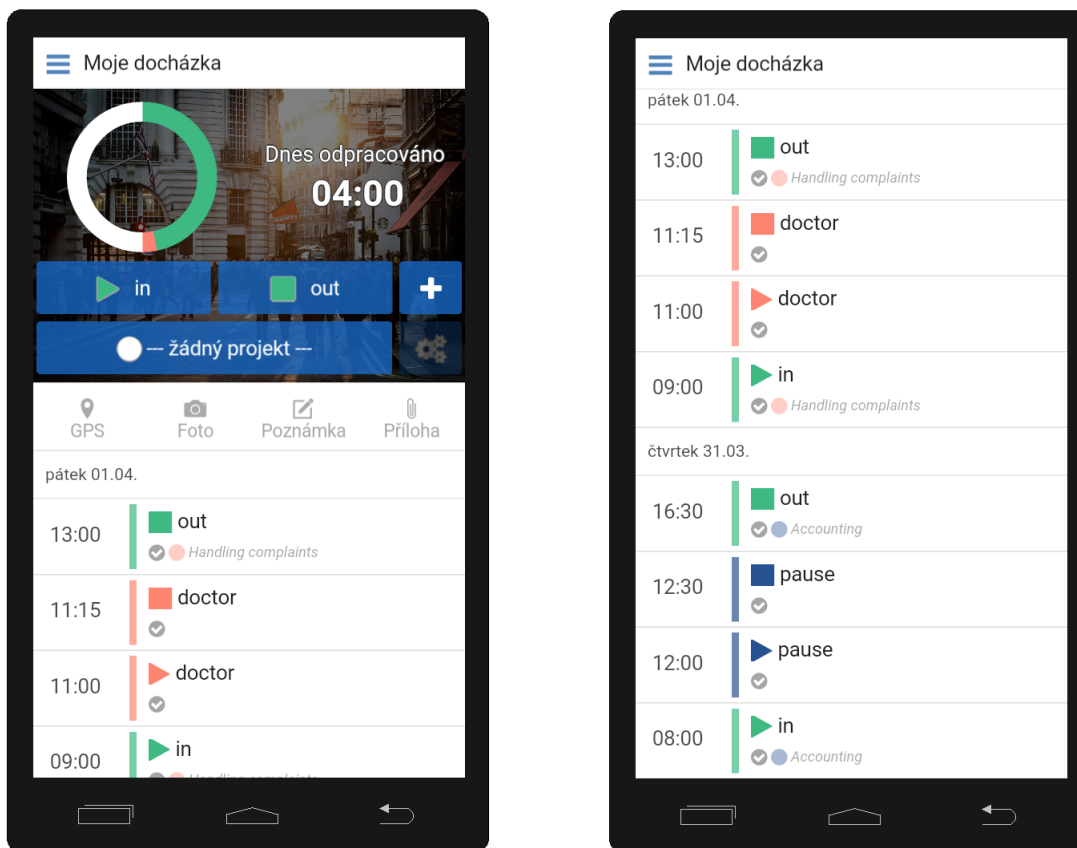
Phony. Pro samotné nahrání souboru používám cordova-plugin-file-transfer [10], jelikož soubor posílá po částech. Z tohoto důvodu je jeho použití lepší u větších souborů, jelikož nezahlcuje paměť telefonu. U tohoto pluginu jsem měl problémy se správným nastavením argumentů. Pro správné fungování jsem musel za pomoci spolupracovníka upravit metodu na REST API, aby podporovala stahování multipart dat.

Při práci na agendě "Přílohy" jsem strávil 35 hodin. Největší část jsem strávil laděním metod pro nahrávání přílohy z různých platforem.

6.2 Agenda Vlastní docházka

Zde jsem dostal za úkol upravit dosavadní metody pro stahování dat se záznamy docházky, aby bylo možno data stahovat na základě časového intervalu. Musel jsem proto upravit dosavadní http požadavek, kdy jsem přidal další parametry určující časový interval. Následně jsem provedl úpravu metody REST API a na serveru, při které mi asistoval spolupracovník. Bylo zapotřebí v dosavadních metodách zohlednit časový interval, na základě kterého se budou filtrovat data.

Stažená data se ukládají pomocí naší cache služby do sql databáze, ze které jsou poté načítána. Načtená data se poté seřadí na základě data pořízení daného záznamu. Seřazení je nutné, aby se v grafickém rozraní mohlo odlišit, kdy se změnil den a následně zobrazit datum nového dne. Data jsou zobrazená pomocí listu, kdy list prochází pole s daty pomocí angularovské funkce ng-repeat a jednotlivé záznamy poté zobrazuje.



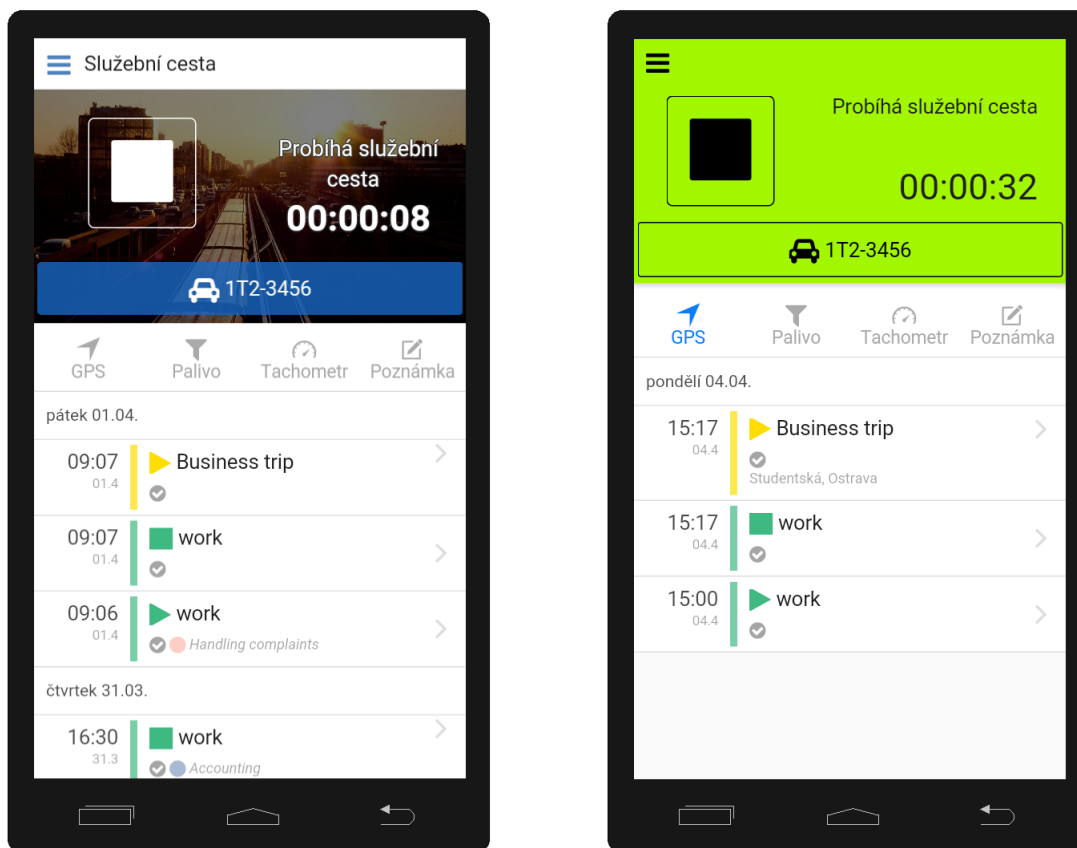
Obrázek 10: Vlastní docházka (aplikace Moje Docházka)

Mým dalším úkolem v této agendě bylo opravit chybu, kdy při odpojení zařízení od internetu aplikace zobrazovala špatně odpracovanou dobu a graf, jelikož nedostávala aktuální data ze serveru. Pokud aplikace není připojena k internetu, vytvoří a zobrazí nový prázdný graf. Údaj odpracované pracovní nahrazují textem s neinicializovaným časem. Jakmile aplikace opět získá spojení se serverem, automaticky aktualizuje graf a odpracovanou dobu.

Dále jsem zde prováděl úpravy týkající se grafického rozhraní. Jednalo se především o stylování jednotlivých elementů, na základě zpětné vazby uživatelů. Výsledný vzhled GUI můžete vidět na Obrázku 10. Při práci na této agendě jsem strávil 15 hodin.

6.3 Agenda Služební cesta

V této agendě jsem měl za úkol upravit dosavadní zobrazení záznamů docházky, aby zde bylo opět možno zobrazovat a dělit starší docházku podle dnů, tak jako tomu je u agendy "Vlastní docházka". Dalším úkolem bylo předělání grafického rozhraní s cílem zpřehlednit tuto agendu. Výsledek můžete vidět na Obrázku 11.



Obrázek 11: Služební cesta (aplikace Moje Docházka)

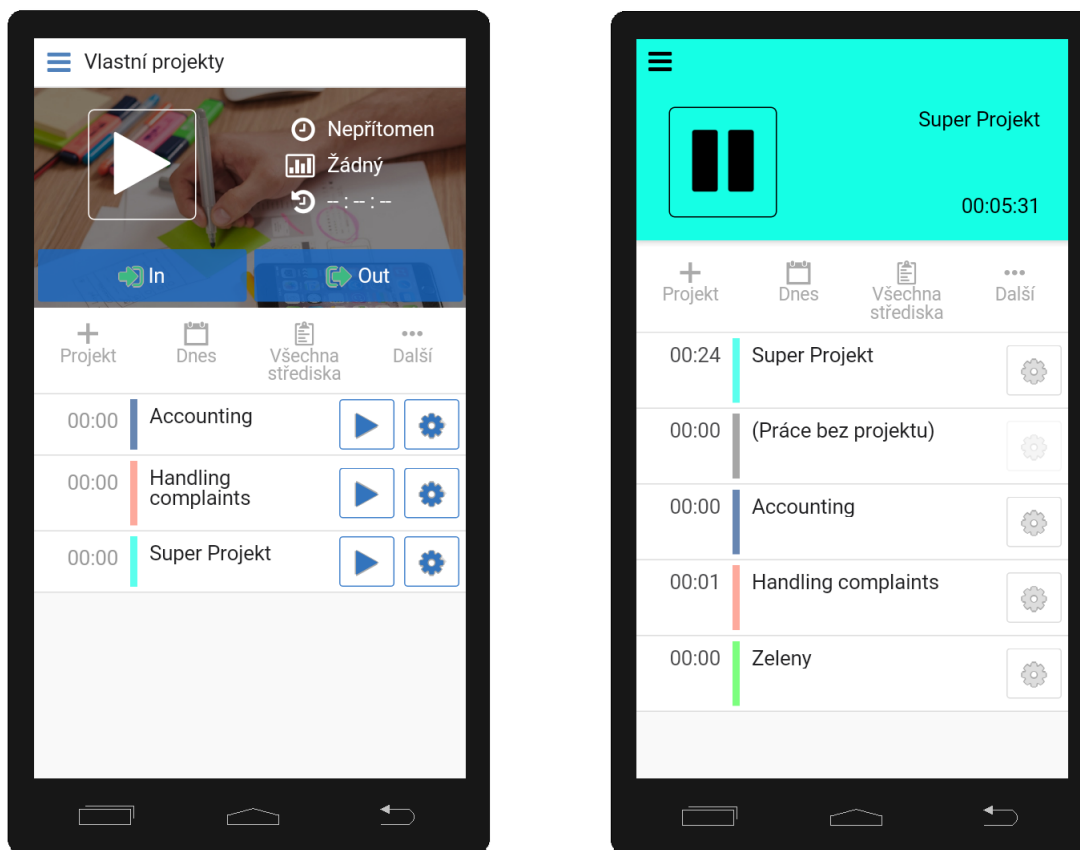
Při testování a zavádění nových efektů na tlačítka, jsem musel vytvořit nový dialog s upozorněním o zapnutí GPS sledování, jelikož současný dialog neumožňoval tento efekt.

Dalším úkolem v této agendě bylo upravení oznámení při počátku služební cesty. Cílem bylo změnit ikonu oznámení v pluginu, který aplikace používá. Jelikož jsem nikde v dokumentaci pluginu nenašel žádné informace o formátu či velikosti ikony, musel jsem se hledat tyto informace na internetu.

Při práci na této agendě jsem strávil 8 hodin.

6.4 Agenda Vlastní projekty

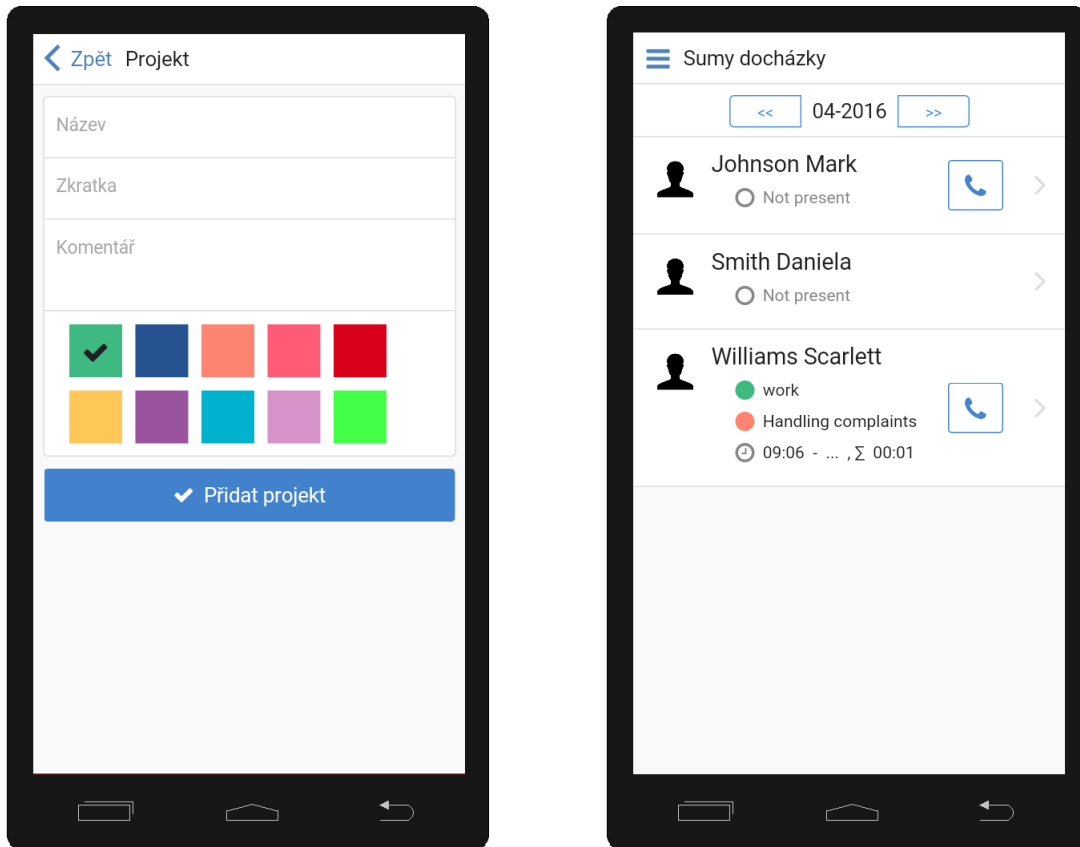
Zde jsem prováděl celkové stylování této agendy. Jednalo se zde hlavně o úpravy týkající se responzivity. Dále jsem přidal nové tlačítko, které zobrazí dialog s možností ukončení projektu a zobrazení agendy "Přílohy" projektu.



Obrázek 12: Vlastní projekty (aplikace Moje Docházka)

V agendě pro vytvoření nových projektů jsem upravoval dosavadní stylování této agendy. Hlavním úkolem bylo odstranit nepotřebné věci a tím zpřehlednit danou agendu. Upravený vzhled GUI můžete vidět na Obrázku 12. Byla odstraněna tlačítka pro příchod a odchod z práce. Ta byla nahrazená položkou "Práce bez projektu". Dále byly odstraněny ikony a informace o tom zda je uživatel v práci. To lze jednoduše poznat podle ubýhajícího času a přenastavení ikony start na ikonu pauzy. Poslední změna se týká pozadí, které se obarví podle aktuálně vybraného projektu.

Dále bylo mým úkolem vytvořit jednoduchý způsob jak zvolit barvu projektu. Vytvořil jsem si proto seznam s barvami, které běžně používáme v ostatních aplikacích. Tyto barvy jsem použil pro pozadí buňky, kdy každá buňka má přiřazený posluchač. Ten reaguje na kliknutí a zvolenou označí ikonou fajfky. Pokud zařízení nemá dostatečně velkou obrazovku, zobrazí se zde pouze takový počet barev, který se vejde na obrazovku. Konečnou podobu můžete vidět na Obrázku 13.



Obrázek 13: Nový projekt a Sumy docházky(aplikace Moje Docházka)

Při práci na této agendě jsem strávil 8 hodin.

6.5 Další úkoly

- V agendě "Sumy docházky" jsem přidával k jednotlivým osobám s vyplněným telefoním číslem tlačítko, které uživateli umožní toto číslo zavolat. Dále jsem k jednotlivým uživatelům přidal textová pole s informacemi o směně a kolik již odpracoval. Výsledek můžete vidět na Obrázku 13.
- Úprava chování hardwarového tlačítka zpět, kdy tlačítko pouze minimalizuje aplikaci. Pokud v zásobníku s historií otevřených agend již nebyly žádné záznamy, tak jsem pomocí pluginu zavolał metodu simulující hardwarové tlačítko Domů, která aplikaci minimalizuje. V hlavním menu aplikace jsem přidal tlačítko pro ukončení aplikace. Tohle chování jsem programoval pouze pro systémy Android, kdy jsem větvil chování programu podle platformy.
- V tutoriálu jsem při prvním spuštění aplikace přidal tlačítko s možností přeskočení tutoriálu. Taky zde sbírám informace, jestli uživatel přeskočil tutoriál či ho prošel celý.

- Při vytváření nebo upravování dosavadních agend jsem musel upravovat soubory s překlady aplikace do dalších jazyků.
- U každého interaktivního elementu v aplikaci jsem přidával nový efekt při kliknutí na daný element. Musel jsem vyhledat všechny tyto elementy a přidat jim další parametr, který daný efekt s využitím pluginu vyvolal.
- Další úkoly se týkaly testování funkcí aplikace. Jednalo se především o replikování hlášených chyb uživateli. Pokud se chyba podařila replikovat, začal jsem po konzultaci se spolupracovníkem pracovat na její opravě.

7 Individuální úkoly

7.1 Vyhledání vhodných tabletů pro aplikaci

Byl jsem pověřen vyhledáním vhodných tabletů pro Android aplikaci Píchnací hodiny GIRITON. Byly mi sděleny požadavky, které tablet musel splňovat a na základě nich jsem poté začal vypracovávat seznam vhodných tabletů.

Vyhledával jsem na základě dvou hlavních požadavků. První požadavek byl napájení před napájecí jack místo micro USB portu. Je to z důvodu, aby se k tabletu dala připojit čtečka čipů. Další požadavek byl na full USB port, aby se nemusely používat redukce z micro USB na full USB k čtečkám čipu.

Po nalezení vhodných tabletů jsem vypracoval seznam, kde jsem jednotlivé tablety rozdělil podle šířky obrazovky, jejich výkonu a pořizovací ceny.

7.2 Využívání aplikací a návrhy na jejich vylepšení

Pro testovací účely jsem začal běžně používat aplikaci Moje docházka GIRITON, kdy bylo mým hlavním cílem zhodnotit aplikaci jako běžný uživatel. Zaměřil jsem se zejména na přehlednost a srozumitelnost aplikace a hledání situací, kdy mohla aplikace začít nesprávně pracovat.

Takto nabyté zkušenosti z pohledu běžného uživatele jsem poté konzultoval se svými spolupracovníky, kdy jsme všichni podali své připomínky a návrhy na jejich řešení. V případě, že tyto následné připomínky byly odůvodněné, bylo následujícím krokem vybrat nebo nalézt nejvhodnější řešení, které by vylepšilo celkový dojem z aplikace.

Dalším úkolem bylo testování nových nebo upravených agend, z důvodů hledání chyb v implementaci. Důležité bylo prověřit funkčnost všech nových věcí a změn a vyzkoušet všechny možné situace, jak se uživatel v dané agendě může zachovat. Hlavní účel tohoto testování bylo zajistit, aby se k uživatelům nedostal update aplikace s chybami.

8 Znalostí a dovedností scházející studentovi

V rámci mé odborné individuální praxe ve firmě GIRITON Systems s.r.o. jsem se setkal s programovacím jazykem Java, se kterým jsem již měl nějaké odborné zkušenosti nabrané během studia. Nicméně jsem se za celou dobu studia nesetkal se složitějšími strukturami kódu nebo komplexnějšími projekty, což mi často činilo problém při orientaci v projektu aplikace Píhací hodiny GIRITON pro Android.

Jako můj další hlavní programovací jazyk, se kterým jsem pracoval v rámci aplikací Moje docházka GIRITON a webové verze aplikace Píhací hodiny GIRITON, byl JavaScript. S tímto jazykem jsem se poprvé setkal až ve třetím ročníku, kdy jsem začal brát jeho základy. Proto jsem se musel v tomto jazyku začít zdokonalovat sám, protože informace sdělené na cvičeních a přednáškách pro mě nebyly dostačující. S tím souvisí i práce s frameworkem AngularJS, který vychází z JavaScriptu. Chybějící znalosti jsem si ovšem doplnil a rychle se naučil s tímto frameworkem pracovat.

9 Dosažené výsledky a celkové zhodnocení praxe

9.1 Dosažené výsledky

Svou odbornou individuální praxi jsem absolvoval ve firmě GIRITON Systems s.r.o., kde jsem celkem pracoval po dobu 50 dnů pod vedením Jana Gřeše.

Během této doby jsem převážně pracoval s frameworky Apache Cordova, AngularJS a Onsen UI. Mou prací bylo navrhovat grafické uživatelské rozhraní jednotlivých agend aplikací Píhací hodiny GIRITON a Moje docházka GIRITON. Následně jsem prováděl implementaci jednotlivých funkcí těchto grafických rozhraní. Díky tomu jsem se naučil pracovat s těmito frameworky na takové úrovni, abych byl schopen sám vytvořit vlastní jednoduchou aplikaci.

V rámci mé praxe jsem měl možnost podílet se na vývoji Píhací hodiny GIRITON pro Android, kde jsem si rozšířil své znalosti programovacího jazyku Java se zaměřením na android aplikace. Zde jsem nabral základní zkušenosti a znalosti o tom, jak fungují aplikace pro systém Android.

Všechny mnou vytvořené agendy a úpravy prováděné na aplikacích postupně vycházely v aktualizacích.

9.2 Celkové zhodnocení praxe

Díky mé odborné praxi ve firmě GIRITON Systems s.r.o. jsem nabyt nových zkušeností a znalostí, které mi pomohly si rozšířit mé dosavadní teoretické a praktické znalosti týkající se navrhnutí a implementace grafického rozhraní a jeho funkcí. Dále jsem prohloubil své znalosti Javy se zaměřením na android aplikace a naučil se pracovat s novými technologiemi, jako jsou AngularJS, Onsen UI a Apache Cordova.

Jako další pozitivní zkušenost do profesního života hodnotím práci v týmu, kdy jsem se naučil komunikovat a pracovat na problému společně se zkušenějšími spolupracovníky. Jedná se hlavně o hledání nejlepšího možného řešení pro moje zadané úkoly. Cílem bylo identifikovat problém a správně ho formulovat spolupracovníkovi, který mě poté mohl nasměrovat na jeho úspěšné vyřešení.

Literatura

- [1] What is Java? - Definition from WhatIs.com [online] [Citace: 9.4.2016.] <http://searchsoa.techtarget.com/definition/Java>
- [2] Overview - Apache Cordova [online] [Citace: 12.3.2016.] <https://cordova.apache.org/docs/en/latest/guide/overview/>
- [3] GitHub - angular/angular.js: HTML enhanced for web apps [online] [Citace: 9.4.2016.] <https://github.com/angular/angular.js>
- [4] Onsen: HTML5 Hybrid App Framework & UI Components [online] [Citace: 12.3.2016.] <https://onsen.io/>
- [5] Remote Method Invocation Home [online] [Citace: 3.4.2016.] <http://www.oracle.com/technetwork/articles/javaee/index-jsp-136424.html>
- [6] GitHub - RobotiumTech/robotium [online] [Citace: 16.3.2016.] <https://github.com/RobotiumTech/robotium>
- [7] GitHub - don/cordova-filechooser: Cordova Plugin that supplies a File Chooser [online] [Citace: 16.3.2016.] <https://github.com/don/cordova-filechooser>
- [8] GitHub - jcesarmobile/FilePicker-Phonegap-iOS-Plugin: This plugin makes possible to pick files from iCloud or other document providers [online] [Citace: 16.3.2016.] <https://github.com/jcesarmobile/FilePicker-Phonegap-iOS-Plugin>
- [9] cordova-imagePicker/README.md at 1ec4242c4f15d83a1842f0439dbb642d5c730773 - wymsee/cordova-imagePicker - GitHub [online] [Citace: 16.3.2016.] <https://github.com/wymsee/cordova-imagePicker/blob/1ec4242c4f15d83a1842f0439dbb642d5c730773/README.md>
- [10] GitHub - apache/cordova-plugin-file-transfer: Mirror of Apache Cordova Plugin file-transfer [online] [Citace: 16.3.2016.] <https://github.com/apache/cordova-plugin-file-transfer>