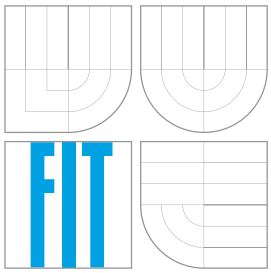
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

VZDÁLENÉ OVLÁDÁNÍ INTELIGENTNÍ DOMÁCNOSTI

SMARTHOME HOME REMOTE CONTROL

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. LEOPOLD PODMOLÍK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAN KOŘENEK, Ph.D.

BRNO 2015

Abstrakt

Hlavním cílem práce bylo zajistit jednoduché a přehledné ovládání pro platformu Android a současně poskytnout uživateli možnost sledovat aktuální a analyzovat historická data ze senzorů v inteligentní domácnosti. Nad rámec zadání jsem rozšířil práci i na platformu Windows phone. Navržené uživatelské rozhraní vychází z požadavků na ovládání inteligentní domácnosti v rámci projektu IoT (Internet of Things), které jsou v této práci sepsány. Výsledné uživatelské rozhraní bylo otestováno pomocí dostupných automatizovaných testů a průběžného testování na úrovni aplha a beta. Dále byla kvalita rozhraní a uživatelská přívětivost vyhodnocena pomocí dotazníků.

Abstract

The aim of the thesis was to ensure simple and clear operation while providing users the ability to monitor and analyze the current historical data from sensors in the smart home. Beyond the task I have extended the work on the Windows Phone platform. Designed user interface based on the requirements for intelligent home control in the project IoT (Internet of Things), which are written in this work. Tests were conducted using the available automated testing and continuous testing at alpha and beta. Finally, the quality of the interface and user friendliness assessed using questionnaires.

Klíčová slova

inteligentní domácnost, uživatelské rozhraní, GUI, android, windows phone

Keywords

intelligent home, smart home, GUI, android, windows phone

Citace

Leopold Podmolík: Vzdálené ovládání inteligentní domácnosti, diplomová práce, Brno, FIT VUT v Brně, 2015

Vzdálené ovládání inteligentní domácnosti

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Jana Kořenka, Ph.D. Další informace mi poskytl Ing. Martin Žádník, Ph.D.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Leopold Podmolík
25. května 2015

Poděkování

Rád bych poděkoval Ing. Janu Kořenkovi, Ph.D. za vedení a konzultace. Dále bych rád poděkoval Ing. Martinu Žádníkovi, Ph.D. za konzultace k inteligentní domácnosti v rámci projektu IoT. Velké díky, také patří rodině a přítelkyni za psychickou podporu.

© Leopold Podmolík, 2015.
Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod **3**

2 Inteligentní domácnost **5**
2.1 Existující systémy inteligentních domácností 5
2.2 Projekt IoT 14
2.3 Podpůrné nástroje pro vývoj 17
2.4 Architektura aplikace 18

3 Ovládání domácnosti mobilním zařízením **20**
3.1 Požadavky na ovládání inteligentní domácnosti v rámci projektu IoT 20

4 Návrh uživatelského rozhraní **22**
4.1 Platforma Android 22
4.2 Platforma Windows phone 8.1 29

5 Implementace **33**
5.1 Vývoj na platformě Android 33
5.2 Vývoj na platformě Windows phone 39
5.3 Problémy při vývoji uživatelského rozhraní 40

6 Vyhodnocení **41**
6.1 Alternativní testování 43
6.2 Dotazník 44

7 Závěr **48**

A Obsah CD **52**

B Ikonky pro inteligentní domácnost **53**

C Obrazovky uživatelského rozhraní – Android **54**

D Obrazovky uživatelského rozhraní – Windows phone **61**

Seznam obrázků

2.1 Mobilní aplikace systému INSTEON. 7
2.2 Mobilní aplikace systému Belkin – WeMo pro platfromu Android. 8
2.3 Mobilní aplikace systému SmartThings. 9
2.4 Mobilní aplikace systému Elkoep – iNELS pro platfromu Android. 10
2.5 Mobilní aplikace systému NetAtmo. 11
2.6 Mobilní aplikace systému Philips HUE pro platfromu Android. 12
2.7 Schéma služby IFTTT. 13
2.8 Struktura inteligentní domácnosti v rámci projektu IoT. 14
2.9 Struktura brány. 15
2.10 Ukázka reálné brány bez obalu. 15
2.11 Struktura serverové části systému. 16
2.12 Ukázka reálného zařízení. 17
2.13 Obecné schéma aplikace pro inteligentní domácnost v rámci projektu IoT. . 18
2.14 Schéma aplikace pro inteligentní domácnost v rámci projektu IoT pro platformu Android. 19

4.1 Struktura obrazovek v aplikaci pro platformu Android. 22
4.2 Přihlašovací a hlavní obrazovka v aplikaci pro platformu Android. 23
4.3 Vysouvací menu v hlavní obrazovce a detail senzoru v aplikaci pro platformu Android. 25
4.4 Obrazovky přidávající HW zařízení – bránu či zařízení. 27
4.5 Ukázka obrazovek sloužící jako nápověda. 28
4.6 Struktura obrazovek v uživatelském rozhraní pro platformu Windows phone. 29
4.7 Ukázka obrazovek aplikace Windows phone. 30
4.8 Obrazovky pro přidání nového HW do domácnosti v aplikaci Windows phone. 31

5.1 Ukázka IDE AndroidStudio s náhledy obrazovky na několika základních typech mobilních zařízení. 34
5.2 Struktura GUI z pohledu implementace základních obrazovek. 37
5.3 Struktura GUI z pohledu implementace nápovědy. 38
5.4 Struktura GUI z pohledu implementace dalších obrazovek. 38
5.5 Struktura GUI z pohledu implementace obrazovek. 39
5.6 Struktura GUI z pohledu implementace dialogů. 40

6.1 Graf průběhu chyb. 43
6.2 Ukázka testování na alternativních systémech. 44

Kapitola 1

Úvod

Internet věcí (Internet of Things, dále jen IoT) si můžeme představit jako všechna zařízení, která jsou připojena k internetu a jsou schopna spolu navzájem komunikovat, ať už to jsou mobilní zařízení, spotřebiče nebo celé domácnosti. V posledních letech narůstá počet připojených zařízení k internetu a tento trend potvrzuje i řada informací z letošního veletrhu CES v Las Vegas, kde bylo jasně řečeno, že IoT je nejžhavějším trendem v oboru IT [25]. Citovaná slova potvrzuje několik existujících prognóz, které mluví o tom, že v roce 2020 bude připojeno k internetu více než 50 mild. zařízení [23], přičemž v současnosti je dle firmy Cisco k internetu připojeno kolem 15 mild. zařízení.

IoT dnes velmi často chápeme ve spojitosti s domácnostmi, kde již dnes existuje několik systémů, jako jsou INSTEON, SmartThings, WeMo Belkin, Philips HUE nebo NetAtmo, které řeší ovládání inteligentní domácnosti. Nevýhodou těchto řešení je drahá pořizovací cena, nemožnost jednoduše rozšířit systém nebo jednoúčelovost systému. Systém BeeeOn vyvíjený na Fakultě informačních technologii VUT v Brně v rámci skupiny IoT se zaměřuje na zmíněné slabiny a klade důraz na vlastnosti, jako jsou jednoduchá rozšiřitelnost, komplexnost (tzn. řešení domácnosti jako celku), bezpečnost a nízké pořizovací náklady. Přesto má BeeeOn s ostatními systémy řadu společných vlastností. Kromě snahy o jednoduché ovládání je to zejména přenos veškerých informací na server neboli cloud, což umožňuje vzdálené ovládání domácnosti z jakéhokoliv místa s internetovým připojením.

Cílem této práce je provést analýzu existujících systémů inteligentních domácností, prozkoumat možnosti ovládání takovéto domácnosti a navrhnout a implementovat jednoduché a co nejvíce intuitivní uživatelské rozhraní (dále jen GUI), které umožní snadno vizualizovat data z jednotlivých senzorů v inteligentní domácnosti BeeeOn.GUI je vyvíjeno jak pro platformu Android, tak pro Windows phone. Původní zadání práce počítalo jen s platformou Android, nicméně v průběhu zpracování zadání vyvstal požadavek i pro platformu Windows phone. Z tohoto důvodu je v mé práci věnován větší prostor platformě Android. V rámci návrhu GUI byly aplikovány jak požadavky pro inteligentní domácnost v rámci projektu IoT, které byly v práci sepsány, tak i pravidla z oficiálního grafického manuálu každé platformy.

V první kapitole tohoto textu je teoreticky rozebrána podstata inteligentní domácnosti. Dále pak jsou popsána existující komerční a open-source řešení inteligentní domácnosti. U komerčních řešení jsou systémy rozděleny na komplexní a jednoúčelové. Kapitola pokračuje popisem inteligentní domácnosti vyvíjené v rámci projektu IoT a to jak základním popisem, tak i charakteristikou celé struktury systému s krátkou specifikací jednotlivých částí. V závěru kapitoly je popsána architektura aplikace, která ovládá inteligentní domácnost v rámci projektu IoT, a to hlavně z důvodu jasného vymezení jednotlivých částí – GUI,

Controller a Network. V další kapitole diplomové práce je teoreticky rozebráno ovládání inteligentní domácnosti z obecného pohledu a soupis specifikace požadavků na uživatelské rozhraní aplikace, které ovládá inteligentní domácnost v rámci projektu IoT. Následující kapitola Návrh se zaobírá grafickým návrhem aplikace a to jak pro softwarovou platformu Android, tak pro platformu Windows phone. Jednotlivé návrhy jsou v kapitole popsány a ilustrovány v podobě ukázek základních obrazovek. Každý z návrhů má svoji specifickou strukturu obrazovek, které vycházejí z možností dané platformy. Dále byla pro potřebu GUI vytvořena sada ikonek charakterizujících jednotlivé typy zařízení a typy místností. Po kapitole Návrh následuje kapitola Implementace, kde je rozebrán vývoj na jednotlivých platformách a použité knihovny. V předposlední kapitole Vyhodnocení je popsáno testování uživatelských rozhraní a vyhodnocení dotazníků. Práce je ukončena kapitolou Závěr.

Kapitola 2

Inteligentní domácnost

Pojem inteligentní domácnost, či dům, se v dnešní době skloňuje v mnoha pádech, aniž by bylo zcela jasné, o co se konkrétně jedná. V podstatě všechny domácnosti, které například disponují senzory pro snímání pohybu pro rozsvícení světel, by se daly považovat za inteligentní domácnosti. Těchto jednoduchých příkladů by bylo možné nalézt celou řadu.

Co tedy vlastně znamená inteligentní domácnost v moderním pojetí? V diplomové práci z Kanady [28] je uvedena definice inteligentní domácnosti ("Home automation") jako integrace elektrických a mechanických zařízení, která spolupracují v jednotném systému jako jedna jednotka. Citovaná definice vystihuje samotný princip inteligentní domácnosti, kde jednotlivá zařízení jako jsou světla, zabezpečovací systém či například mechanismus otevírání garážových dveří, komunikují v rámci jednoho systému.

V minulosti byly systémy již integrovány při stavbě samotného domu, protože se jednalo o "drátové"systémy, kdy byla všechna zařízení připojena k ovládacímu panelu pomocí drátů ve zdi. Samozřejmě, že mluvíme o novodobé historii, tj. pouze posledních 25 let. První komplexní systémy se začaly objevovat na začátku devadesátých let, kdy stávající technologie umožňovala integrování ovládacích mechanismů do domácností. Popsané systémy se vyznačovaly relativně vysokou cenou a jejich ovládání bylo většinou umožněno pomocí zabudovaného panelu ve zdi.

2.1 Existující systémy inteligentních domácností

V dnešní době existuje spousta řešení a možností jak ovládat inteligentní domácnost. Obecně lze řešení inteligentních domácností rozdělit na komerční a open-source. Rozdělení jsem provedl na základě vlastností systémů, kdy open-source systémy nejsou tvořeny pro "laiky", ale spíše pro IT zdatné uživatele. Naopak komerční řešení jsou tvořeny pro uživatele, kteří očekávají "easy-to-use"systém. Komerční řešení se dále dělí na systémy komplexní a jednoúčelové.

Současná řešení se snaží pokrýt nejrůznější funkce požadované v domácnosti. Například se snaží řešit:

- **Ovládání světel** – a to jak automatické ovládání pomocí například pohybových senzorů, tak i dálkové ovládání světel,
- **Zabezpečovací systém**,
- **Ovládání termostatu, kotle či klimatizace** – v zahraniční literatuře je často použita zkratka HVAC, která znamená Heating, ventilation and air conditioning,

- **Audiovizuální zařízení** – často použito při ovládání domácího kina,
- **Interkom** – neboli domácí komunikace.

Ostatní funkce systémů inteligentních domácností je možné najít na webových stránkách [7].

Dalším rysem současných řešení je skutečnost, že se používají pojmy Sensor a Aktor. Pojem sensor znamená pouze snímající zařízení, které posílá informace dále, zatímco pod pojmem aktor se myslí zařízení, které je možné vzdáleně ovládat nebo řídit, a tím provést nějakou akci.

2.1.1 Komerční systémy

U komerčních řešení je potřeba rozdělit systémy na komplexní a jednoúčelové, protože mají většinou jiný segment zákazníků a jiné vlastnosti systému. Například u kamerového systému nám stačí, pokud můžeme snímat určité místo a mít případně zapnutou automatickou detekci pohybu. V případě, že potřebujeme komplexní systém, který nám dokáže zajistit více takovýchto služeb, jako je například ovládání termostatu či kontrolu vlhkosti místnosti, pak musí mít systém jiné parametry. U všech zmíněných řešení jsou popsány získané informace z příslušných webových stránek, jejichž odkazy jsou u každého popisu řešení uvedeny.

INSTEON

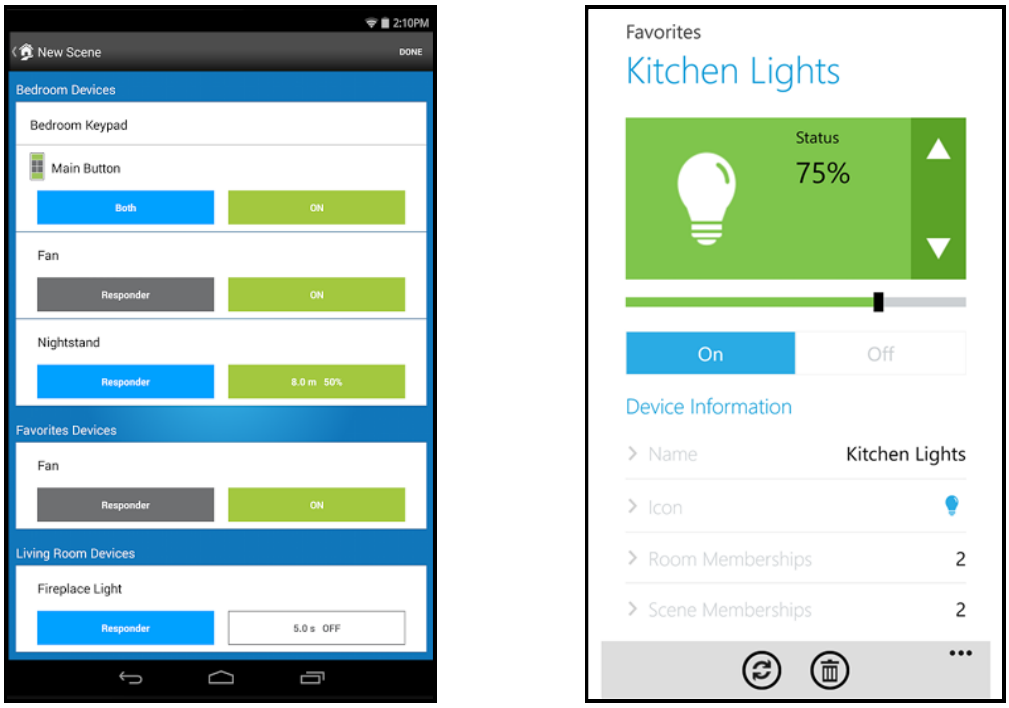
Systém INSTEON [8] zahrnuje zařízení od inteligentních žárovek přes termostat až k různým senzorům. INSTEON je koncipován jako uzavřený, tudíž do něj nelze připojit zařízení třetích stran a skládá se z několika částí:

- Insteon Hub – centrální prvek, ke kterému jsou připojeny všechny ostatní prvky systému,
- Wirelless Sensors – bezdrátové senzory, může se jednat například o sensor vlhkosti nebo senzor u dveří, který monitoruje, zda jsou dveře zavřené,
- Plug-In Devices – zařízení, která mohou ovládat například lampy,
- Wall Switches – vzdálené ovládání vestavěných světel,
- LED Bulbs – inteligentní žárovka, kterou lze ovládat.

Protože je tento systém neustále připojen k internetu a jeho centrální prvek komunikuje se serverem celého sytému, je možné jej ovládat odkudkoliv pomocí moderních zařízení jako jsou smartphony a tablety.

Výrobce na svých stránkách uvádí, že podporuje více jak 200 zařízení a oproti ostatním řešením patří robustnost mezi velké výhody systému. Další výhodou toho systému je možnost použití scén, což znamená, že uživateli je nabídnuto několik variant nastavení dalších zařízení, aby si mohl například dotvořit atmosféru při sledování filmu, případně si scény mohl upravit sám. Tento systém lze ovládat téměř na všech platformách a to buď přes nativní aplikace pro iOS, Android a WindowsPhone, nebo přes webové rozhraní na MacOS X a Linux.

Uvedený systém je navržen pro větší a členitější domácnosti. Aplikace umožňuje rozdělení jednotlivých senzorů do více místností, což můžeme vidět na obrázku 2.1a, který zobrazuje aplikaci pro android. Na obrázku 2.1b je znázorněna verze aplikace pro windows phone a můžeme zde vidět detail daného senzoru, u kterého je možné změnit nastavení příslušného zařízení nebo zjistit jeho podrobnější informace.



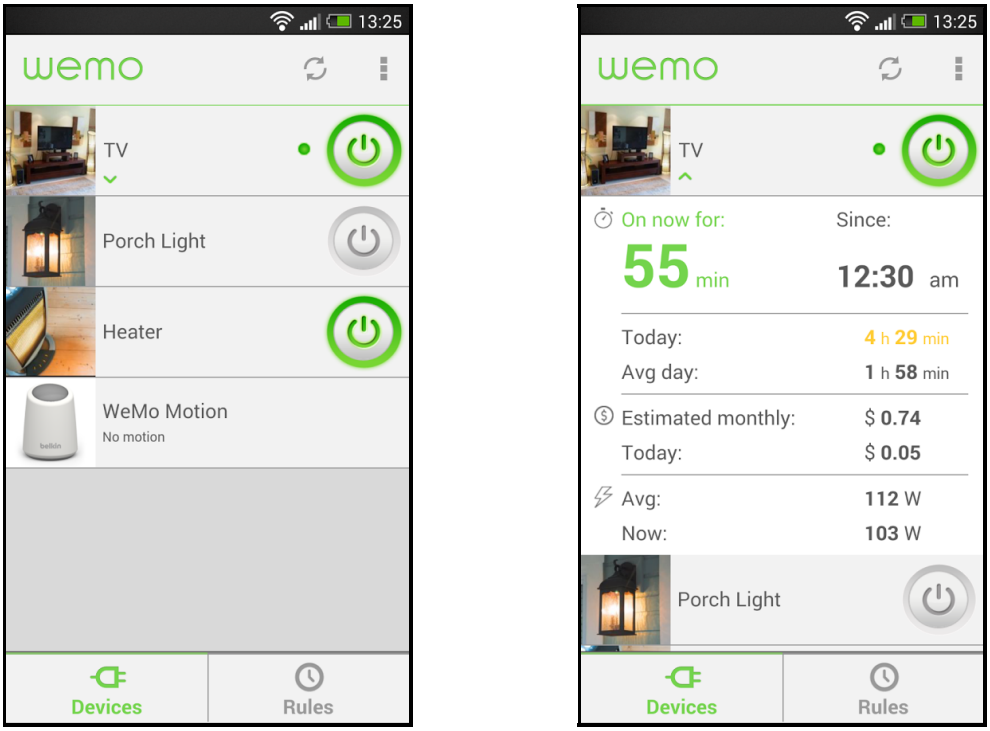
(a) Verze aplikace pro platformu Android. (b) Verze aplikace pro platformu Windows phone.

Obrázek 2.1: Mobilní aplikace systému INSTEON. Zdroj: [8]

BELKIN – WeMo

Systém WeMo [20] pětšiochází od firmy BELKIN, která vyrábí celou řadu elektroniky, jako jsou routery nebo kamery, takže jejich systém využívá všech prvků, které vyrábějí. Tento systém používá pouze zařízení od firmy Belkin, takže jej není možné rozšířit o zařízení třetích stran. Dále tento systém nepoužívá centrální prvek pro správu dalších zařízení, ale každé zařízení se připojí k domácí síti přes WiFi a komunikuje se serverem systému WeMo, takže je opět možné zařízení ovládat vzdáleně pomocí mobilních zařízení. Výhodou uvedeného systému je velká modularita a nezávislost všech zařízení. Zajímavostí systému je napojení na službu IFTTT, která bude více rozebrána v další části textu. Uvedený systém lze ovládat pomocí nativních aplikací pro iOS a Android.

Aplikace pro systém WeMo, která je k vidění na obrázcích 2.2, na rozdíl od jiných aplikací, je navržena tak, že obsahuje pouze výpis všech připojených zařízení a jednotlivá zařízení není možné třídit například do místností. Dále aplikace neumožňuje žádný jiný pohled na data mimo pohledu na historii. Aplikace umožňuje velmi rychlé ovládání domácnosti, respektive jejich aktorů, a to tím, že u každého aktoru je ovládací tlačítko pro zapnutí nebo vypnutí.



(a) Přehled všech zařízení.

(b) Detail zařízení.

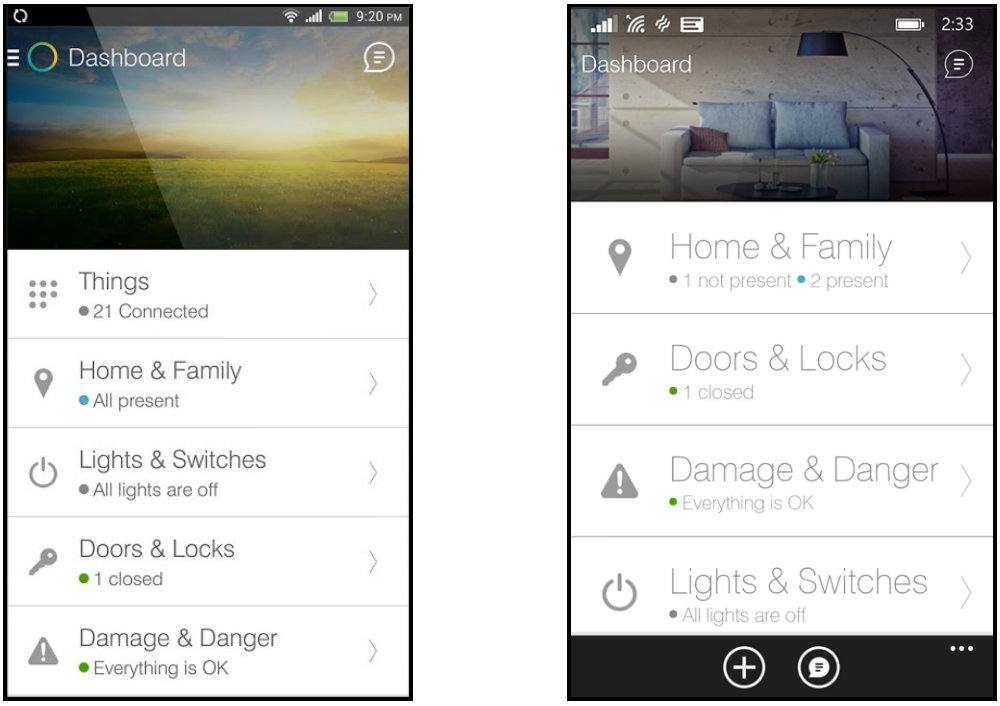
Obrázek 2.2: Mobilní aplikace systému Belkin – WeMo pro platfromu Android. Zdroj: [20]

SmartThings

Systém SmartThings [17] je, opět jako předcházející systémy, připojen online k serveru a lze jej ovládat vzdáleně. Další podobností je centrální prvek SmartThings Hub. Tento systém na rozdíl od systému INSTEON klade více důraz na otevřenost a spolupracuje s větším počtem výrobců, kteří se specializují na konkrétní zařízení, jako je například elektrický zámek dveří apod. U tohoto systému bychom mohli říci, že nabízí platformu, která spojuje zařízení ostatních specializovaných výrobců do jednoho systému, který je ovládán pomocí mobilního zařízení. Zajímavostí systému SmartThings je, že spolupracuje s inteligentním termostatem Nest[1]. Tento systém nabízí nativní aplikace pro všechny platformy tedy Android, iOS a Windows.

Na rozdíl od aplikace INSTEON, používá aplikace SmartThings jiný princip ovládání domácnosti. Jednotlivé senzory jsou rozděleny podle svých typů a není možné jejich rozdělení do jednotlivých místností. Celkově by se dalo říci, že je aplikace navržena s ohledem na co nejjednodušší ovládání pro koncového uživatele, tj. aby nemusel nastavovat další informace, např. jako je konkrétní místnost apod. Na následujících obrázcích 2.3a a 2.3b můžeme vidět verzi aplikace pro Android a dále pak verzi aplikace pro Windows phone.

[1] https://nest.com/



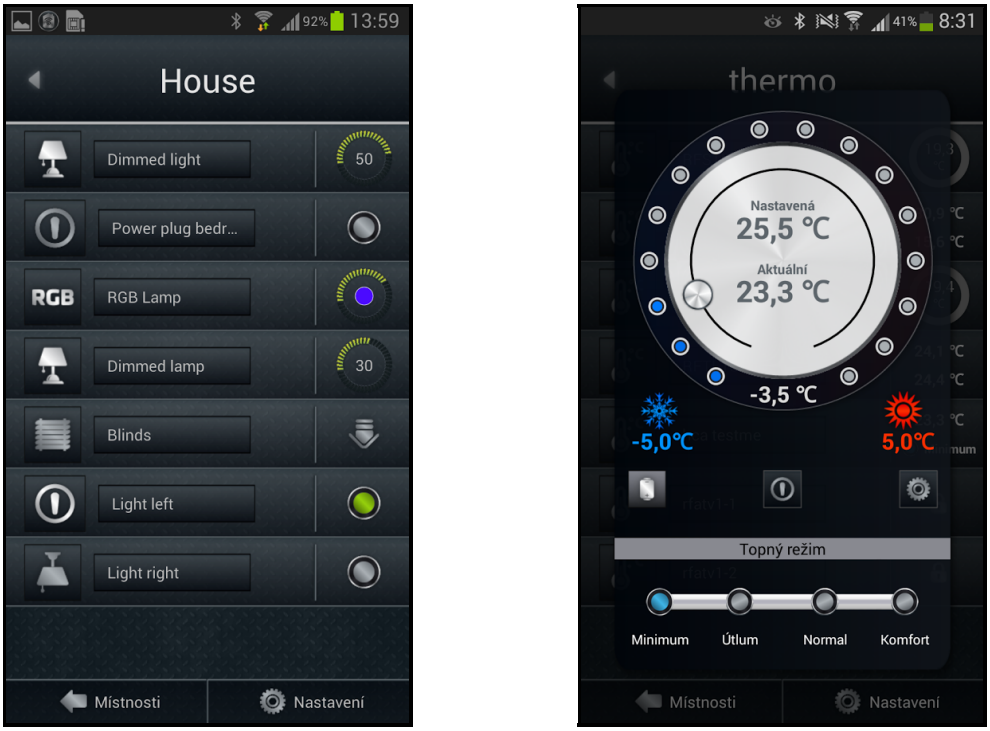
(a) Verze aplikace pro platformu Android. (b) Verze aplikace pro platformu Windows phone.

Obrázek 2.3: Mobilní aplikace systému SmartThings. Zdroj: [17]

iNELS

Česká firma Elko ep s.r.o. vyrábí systém iNELS [9], který má podobné vlastnosti jako předcházející systémy. Základním prvkem je "Chytrá RF krabička", která je připojena k internetu a tím i ke vzdálenému serveru, díky kterému můžeme opět tento systém ovládat odkudkoliv. Uvedená firma vyrábí celou řadu sensorů a aktorů přímo pro systém iNELS. I tento systém umožnuje uživateli nastavení scénáře, což znamená posloupnost úkonů, které se provedou v domácnosti. Další důležitou vlastností systému je jeho široká možnost rozšíření o další senzory. Systém iNELS se dá ovládat pomocí nativních aplikací pro Android a iOS.

Aplikace od společnosti Elkoep je navržena tak, aby se mohly jednotlivé aktory velmi jednoduše zapínat anebo vypínat, a to poklepem na danou ikonku. Protože je systém navržen jako robustní, tak i zde je možnost přidávat jednotlivé senzory nebo aktory do místností, a tím je přehledně členit. Na obrázcích 2.4a a 2.4b jsou vidět ukázky aplikace iNELS.



(a) Přehled všech zařízení.

(b) Nastavení termostatu.

Obrázek 2.4: Mobilní aplikace systému Elkoep – iNELS pro platfromu Android. Zdroj: [9]

2.1.2 Komerční systémy – jednoúčelové systémy

Alkom security a.s. – Evohome

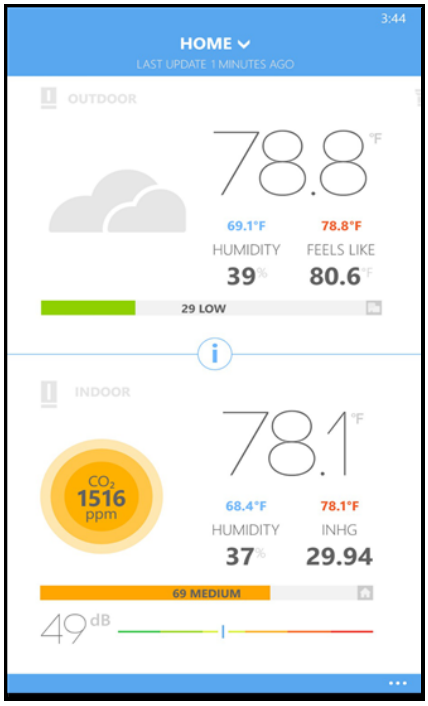
Česká firma Alkom vyrábí systém Evohome [10], který se soustředí na ovládání a regulaci topení. Hlavní výhodou tohoto řešení je možnost připojení k různým kotlům díky RF Bridge (Open Therm Interface). Tento systém není napojen na vzdálený server a umožňuje tedy pouze lokální použití. Ovládání je možné pouze pomocí dotykového displeje na řídící jednotce.

Netatmo

Systém NetAtmo [12] patří mezi inteligentní meteostanice. Nabízí jak sensor pro vnitřní měření, tak i externí sensor pro měření venku. Cílem tohoto systému je nabídnout uživateli přehledné informace o aktuálním stavu, historii a předpovědi počasí. Systém měří teplotu, vlhkost, kvalitu ovzduší, úroveň CO_2 a hluk. Systém Netatmo je možné ovládat pomocí nativních aplikací pro Android, iOS a Windows phone a také pomocí webové aplikace.

Systém Netatmo obsahuje základní funkci, kterou je meteostanice, a tomu je přizpůsobena i aplikace, která má za úkol uživatele co nejvíce informovat o hodnotách, které meteostanice sleduje. Zajímavé u této aplikace je, že nabízí zobrazení grafů s historickými hodnotami. Ukázku aplikace Netatmo pro platformu Android lze vidět na obrázku 2.5a a pro platformu Windows phone na obrázku 2.5b.

(a) Verze aplikace pro platformu Android.



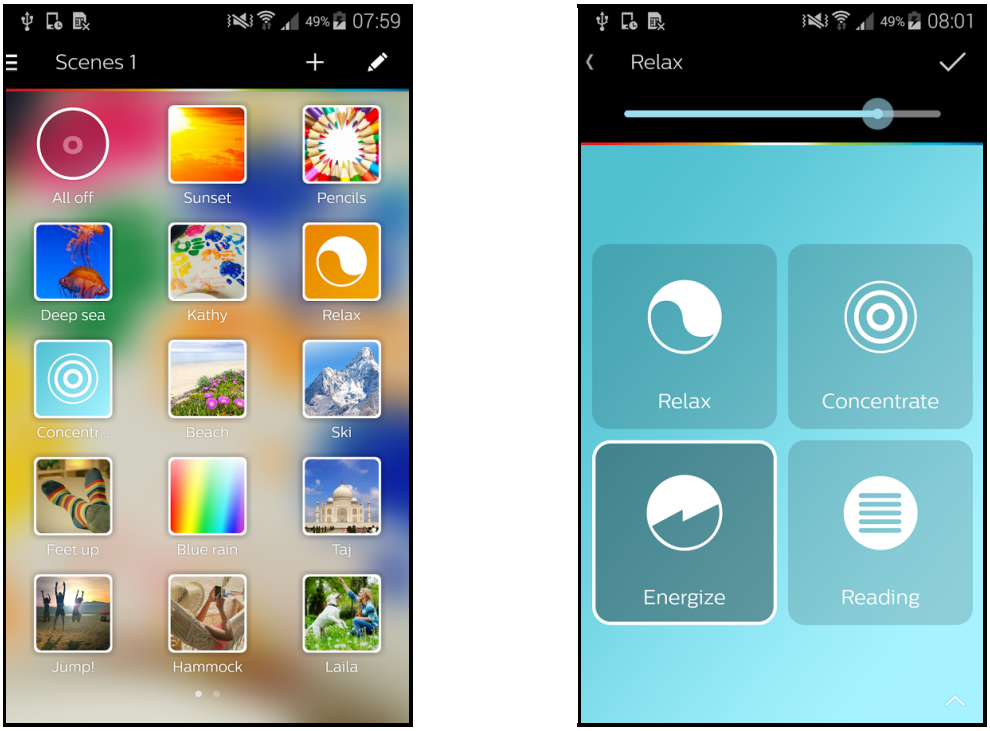
(b) Verze aplikace pro platformu Windows phone.

Obrázek 2.5: Mobilní aplikace systému NetAtmo. Zdroj: [12]

Philips – Hue

Systém Hue [11] od společnosti Philips nabízí ovládání inteligentní žárovek, které umožňuje nejenom ovládání stavu vypnout/zapnout, ale také změnu barvy osvětlení. Inteligentní žárovky jsou připojeny samostatně pomocí wifi, tudíž zde není nutný žádný centrální prvek. Velký důraz je kladen na personalizaci, což se projevuje v možnosti nastavení různých scénářů. Velkou výhodou tohoto systému je možnost napojení na službu IFTTT. Žárovky Hue je možné ovládat pomocí nativních aplikací pro Android a iOS.

Aplikace od firmy Philips pro systém Hue je plně zaměřena na personalizaci inteligentních žárovek. Na hlavní obrazovce najdeme možnost nastavení scény pro danou žárovku. Kromě nastavení scény, což jsou různé sekvence blikání a změny barvy, nabízí také aplikace možnost nastavení stále svícení konkrétní vybrané barvy pro žárovku. Na obrázcích 2.6a a 2.6b lze vidět ukázku aplikace Hue pro platformu Android.



(a) Přehled scén.

(b) Detail scény.

Obrázek 2.6: Mobilní aplikace systému Philips HUE pro platfromu Android. Zdroj: [11]

2.1.3 Volně šiřitelné systémy

OpenRemote

Systém OpenRemote [14] je vytvořen jako softwarová platforma, umožňující ovládat domácnost pomocí několika podporovaných hardwarových zařízení a skládá ze tří základních komponent:

- OpenRemote Controller – ovládá připojené zařízení a umožnuje plánování akcí,
- OpenRemote Designer – jedná se o nástroj, který umožnuje upravovat jak výše uvedený controller, tak i si přizpůsobit uživatelské prostředí aplikace,
- OpenRemote Control Panel – výsledná aplikace ovládající domácnost.

Postup pro zprovoznění tohoto systému není určen pro obyčejného uživatele, takže se jedná o systém, který využijí převážně jen nadšenci a IT odborníci.

FreeDomotic

FreeDomotic [6] je framework, který umožnuje použití hned na několika platformách, a to buď s podporou Javy na jakémkoliv OS nebo na platformě Raspberry PI. Výhodou tohoto systému je, že může být použit s technologiemi, jako jsou KNX, Z-wave a nebo arduino.

OpenDomo

Projekt OpenDomo [13] patří mezi operační systémy pro domácnost. Systém umožňuje zpřístupnění ovládání domácnosti z vnějšku a tím pádem i zvýšení zabezpečení domácnosti. Jediná možnost, jak ovládat daný systém, je webové rozhraní.

2.1.4 Služba IFTTT

Služba IFTTT [15] (celým názvem If this then that) se nezaměřuje na inteligentní domácnosti, ale s mnoha výše uvedenými systémy spolupracuje. Princip této služby je znázorněn na obrázku 2.7 a spočívá v jednoduchém vytváření receptů, kdy je na jedné straně událost, která spouští daný recept, a na druhé straně nějaká akce, která se má provést.

Obrázek 2.7: Schéma služby IFTTT. Zdroj: [15]

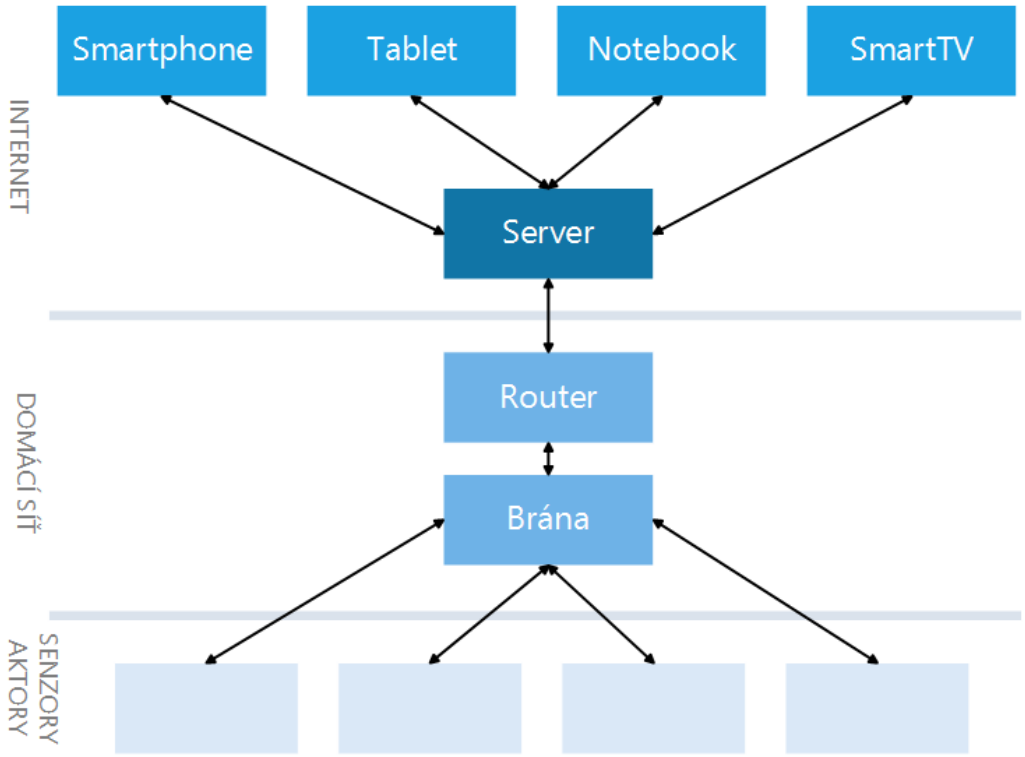
V praxi je často uváděn příklad ve spojení s inteligentní žárovkou Hue, kdy je využito geo-lokace a při příchodu uživatele domů změní žárovka barvu. Jednoduše by se dalo říci, že služba IFTTT přidává věcem automatizovanou závislost na vnějších podmínkách, které si může uživatel sám definovat. Službu můžeme ovládat pomocí aplikací pro Android a iOS a také přes webové rozhraní.

Službu IFTTT využívají následující systémy:

- Nest Thermostat,
- Netatmo Weather,
- Philips hue,
- SmartThings,
- WeMo.

2.2 Projekt IoT

Systém je založen na principu ovládání domácnosti jedním centrálním prvkem (Bránou) a několika sensorů či aktorů, které jsou k němu připojeny pomoci bezdrátové sítě. Brána slouží jako shromažďovací prvek a posílá informace sesbírané od senzorů nebo aktorů na server. Tento systém je navržen jako "cloud" řešení. Veškeré algoritmy, které přidávají tomuto systému inteligenci, jsou implementovány na serveru systému. Poslední částí tohoto systému je aplikace, která ovládá inteligentní domácnost, resp. posílá příkazy serveru, ten požadavek zpracuje a případně předá dál. Pro ilustraci je níže na obrázku 2.8 uvedeno schéma systému. Podrobnější informace jsou popsány dále v textu.



Obrázek 2.8: Struktura inteligentní domácnosti v rámci projektu IoT.

Projekt IoT, jehož systém byl pojmenován BeeeOn, si neklade za cíl vytvořit velkou hardwarovou základnu senzorů či aktorů, ale spíše nabídnout ucelený systém správy jednotlivých prvků tak, aby se do systému dala připojit zařízení třetích stran, a současně aby bylo možné jejich jednotné ovládání pomocí jedné aplikace. Další vlastností tohoto systému je možnost inteligentní práce na serveru, který zajišťuje analýzu dat a poté předává informace dále. Jako typický příklad můžeme uvést senzor s tepelným čidlem v situaci, kdy v domácnosti začne hořet, server stav zachytí a informuje uživatele o této skutečnosti.

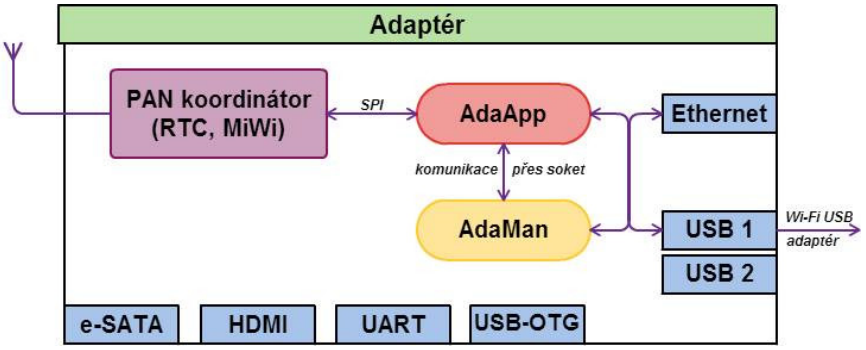
Velmi důležitou vlastností systému je zabezpečení, a to hned v několika rovinách. První rovina se uskutečňuje přímo v komunikaci mezi senzory a bránou, kde je zajištěno protokolem *FITpro* vyvíjeným na FIT VUT v Brně, aby nedocházelo ke ztrátám nebo k falzifikaci zpráv. Druhá rovina zabezpečení komunikace je mezi bránou a serverem, kdy je zabezpečeno pomocí self-signeted certifikátů a následného SSL spojení. Poslední rovina zabezpečení je v komunikaci mezi aplikací a serverem, kde jsou opět použity certifikáty pro šifrovanou komunikaci.

2.2.1 Struktura inteligentní domácnosti IoT

Jak již bylo uvedeno v předchozím textu, celý systém se skládá ze senzorů či aktorů, brány, serveru a ovládací aplikace. Nyní budou blíže popsány jednotlivé prvky systému.

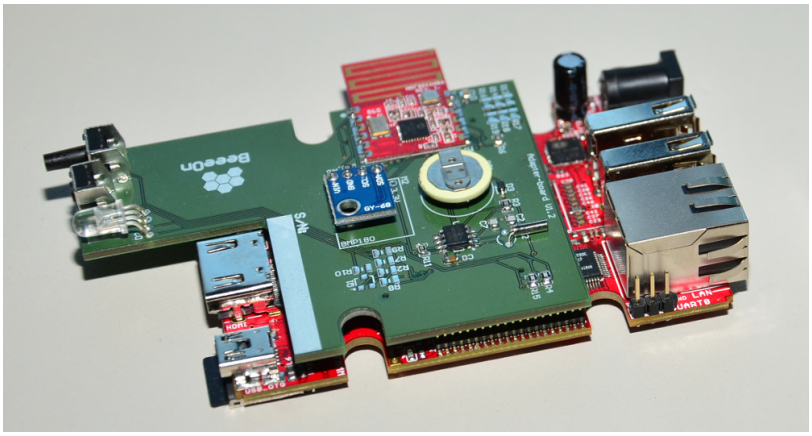
Brána - je HW zařízení, které je centrálním prvkem v domácnosti a komunikuje jak se senzorovou částí systému přes bezdrátovou sít, tak se serverem pomocí internetového připojení v domácnosti.

Jak lze vyčíst z obrázku 2.9, na bráně funguje několik aplikací. Nejdůležitější aplikací je AdaApp, která se stará o sběr informací ze senzorové části a přeposílá je serveru, anebo naopak reaguje na žádost od serveru při požadavku na změnu stavu aktora nebo registraci nového zařízení. Další aplikací, která funguje na bráně, je AdaMan. Jedná se o aplikaci, která se stará o aktualizaci AdaApp a shromažďuje statistiky a předává je na server.



Obrázek 2.9: Struktura brány. Zdroj: Wiki projektu IoT

Pro realizaci brány je z technického hlediska použita open-hardware platforma OLinuXino, na které běží OS Linux. Použitá je konkrétně varianta A10, což znamená, že vývojová deska je osazena procesorem řady Cortex-A8, HDMI a ethernetem. Přesněji byla použita rozšířená varianta A10-OLinuXino-LIME-4GB, která má větší úložiště než standardní verze. Použitá varianta umožňuje svojí výbavou i náročnější operace, tudíž je zde prostor i pro další aplikace. HDMI konektor je sice nyní nevyužitý, ale v blízké budoucnosti se předpokládá vyvinutí nové aplikace, která by vizualizovala data již ze samotné brány, právě pomocí HDMI konektoru (například na televizi). Na obrázku 2.10 lze vidět uvedenou vývojovou desku OLinuXino spolu s modulem pro rádiovou komunikaci se zařízeními (senzory, aktory).

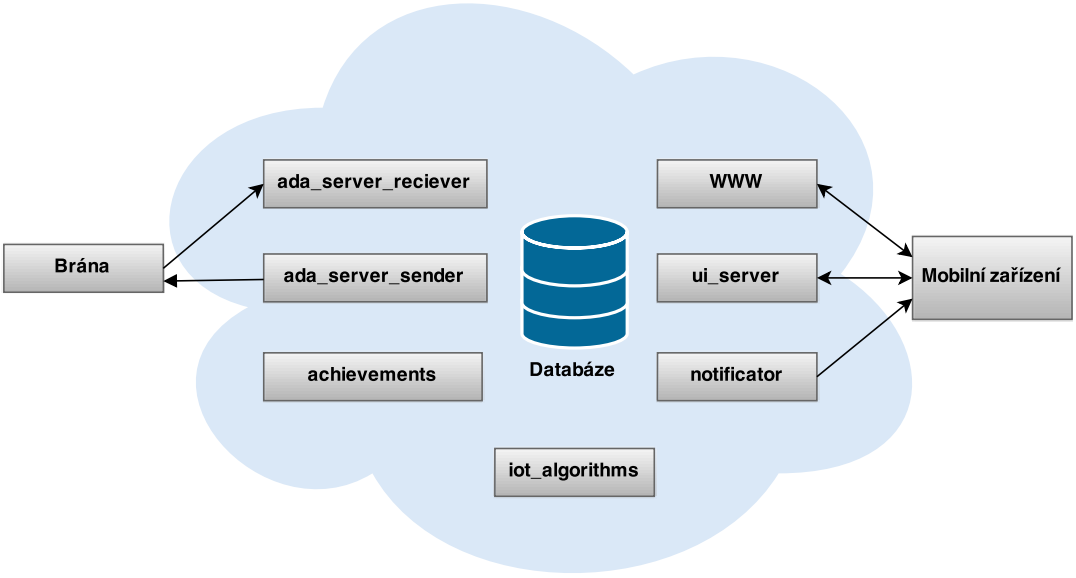


Obrázek 2.10: Ukázka reálné brány bez obalu. Zdroj: Wiki projektu IoT

Server - Veškeré algoritmy, které přidávají celému systému "chytrost", jsou implementovány na serveru. A právě "chytrost"je zajišťována několika aplikacemi, které můžeme vidět na obrázku 2.11, a to Ada_Server_Reciever (ASR), Ada_Server_Sender (ASS), UI_Server (UIS) a Notificator (N). Pro ukládání dat na serveru slouží databáze, do které přistupuje jak ASR, tak UIS. Aplikace UIS komunikuje obousměrně s koncovým zařízením a předává případné zprávy ASR. Aplikace ASS posílá zprávy na bránu a ASR od brány zprávy přijímá. Další zajímavou aplikací je N, která zajištuje u OS Android zaslání notifikační zprávy na mobilní zařízení. Další součástí serveru jsou *iot_algoritmy*, které se skládají ze dvou částí a to z frameworku a samotných algoritmů. Framework pro algoritmy je napojen na databázi a komunikuje s ASR. Konkrétní algoritmy řeší již různé akce, jako může být třeba *Watchdog*. Algoritmus *Watchdog* slouží jako hlídač, kterému je možné nastavit sledování překročení hodnoty senzoru a provedení následné akce. Jako akci lze nastavit zaslání zprávy skrz notifikaci nebo přepnutí stavu aktoru.

Poslední aplikací, která je spuštěna na serveru je aplikace *achivments*, která slouží pro gamifikaci[2] systému. V rámci gamifikace celého systému je potřeba uchovávat informace a případně informovat uživatele o stavu úkolu jako je například počet zařízení v domácnosti. Gamifikace systému má uživateli zpříjemnit práci ze systémem a to "hravou"formou. Provedením určitých akcí, jako například přidání nového zařízení, uživatel získává virtuální body a díky tomu může soutěžit ze svými známými, kteří taktéž využívají inteligentní domácnost BeeeOn.

Server funguje na platformě OS Linux a pro účely databáze zde běží *PostqresSQL*. *PostqresSQL* byl vybrán z toho důvodu, že umožňuje rychlý export dat z tabulek do XML struktury. Jelikož je pro veškerou komunikaci použit XML protokol, tak se jedná o optimální řešení.



Obrázek 2.11: Struktura serverové části systému. Zdroj: Wiki projektu IoT

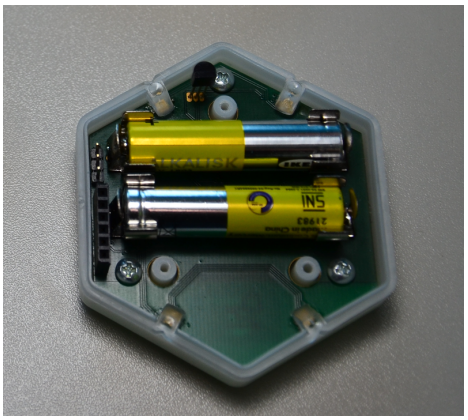
[2]http://cs.wikipedia.org/wiki/Gamifikace – je uplatňování tzv. technik herních designů, herního myšlení a herních principů do neherních oblastí.

Senzory a aktory - Senzorová část systému obsahuje dva typy zařízení, a to senzory a aktory. Senzory jsou RFD (reduced funciton device) [22] a jejich funkcí je zasílání aktuálních informací od všech čidel, která obsahují. Základní čidla, umístěná na senzoru, jsou tepelná, luminiscentní a vlhkostní. Výhodou těchto zařízení je, že se senzory dokáží přepnout do režimu spánku na určitý čas, a tím pádem jsou i velmi úsporné.

Aktory jsou FFD (full function device) [22] a jsou založeny na možnosti měnit svůj stav. Příkladem může být inteligentní zásuvka – zásuvce lze nastavit takový stav, který umožňuje procházení nebo brání průchodu proudu. Zařízení, na rozdíl od senzorů, nemohou být uvedena do režimu spánku, jelikož nelze predikovat, kdy událost vyžadující změnu stavu nastane.

Na obrázcích 2.12a a 2.12b můžeme vidět vyrobené zařízení obsahující dva senzory a to senzor teploty a vlhkosti.

(a) Zařízení ve finálním obalu.



(b) Vnitřek zařízení.

Obrázek 2.12: Ukázka reálného zařízení. Zdroj: Wiki projektu IoT

Koncové uživatelské zařízení - Poslední částí systému je aplikace, která běží na moderních zařízeních, jako jsou smartphony nebo tablety. Aplikace slouží k přihlášení uživatele do systému, čímž mu umožňuje zjistit aktuální stav nebo historii senzoru, případně změnit stav aktoru. Dále aplikace umožňuje přidávání nových zařízení a správu všech zařízení v domácnosti. Detailnější popis aplikace je uveden dále v textu.

2.3 Podpůrné nástroje pro vývoj

Jelikož se na vývoji celé inteligentní domácnosti podílelo několik desítek lidí, kteří měli na starosti určité části systému, bylo potřeba předávat si získané znalosti a informace o problémech napříč celým systémem. Z toho důvodu byl použit software pro podporu řízení Redmine [16]. Pomocí nástroje Redmine byly uchovávány informace týkající se celé domácnosti BeeeOn, které umožňovaly zadávání nových úkolů (*features*) nebo úkolů na opravu (*bugs*) při zjištění chyby. Nástroje Redmine se velmi osvědčil především ve fázi testování, kdy umožnil způsob předávání informací od testerů k nám vývojářům. Více informací o testování je podrobněji popsáno v následující kapitole.

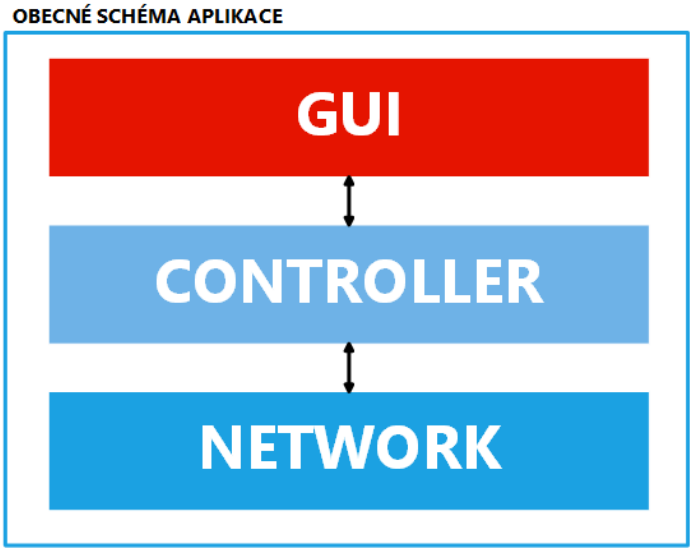
Dalším systémem, který byl při vývoji inteligentní domácnosti použit, je systém správy verzí *git*. Tento systém umožnil paralelní vývoj více vývojářů na jednom projektu. V mém případě se jedná o vývojáře, kteří měli na starosti vývoj sítové vrstvy (*Network*), *Controller*

nebo widgetu tak, aby výsledná aplikace byla konzistentní a funkční. Další nespornou výhodou systému *git* je jeho použití jako archivačního systému, který umožňuje mít uloženou celou historii verzí kódu na serveru, takže odpadá riziko, že by se ztratila data při poruše zařízení, na kterém vývoj probíhá. Jako klientskou aplikaci pro ovládání systému *git* jsem použil aplikaci SmartGit [19] pro nekomerční účely. Výhodou klientské aplikace je její použitelnost jak na OS Linux, tak na OS Windows. Z tohoto důvodu ji bylo možné použít jak při vývoji uživatelského rozhraní pro Android, které probíhalo v OS Linux, tak při vývoji uživatelského rozhraní pro Windows phone, které probíhalo na OS Windows.

2.4 Architektura aplikace

Tato práce se zabývá aplikací, která ovládá inteligentní domácnost v rámci projektu IoT. Cílem je vytvořit uživatelské rozhraní. Vzhledem k této skutečnosti je nutné vysvětlit i samotnou strukturu aplikace. Aplikace se skládá z několika vrstev, které jsou znázorněné i na obrázku 2.13:

- **Network** – vrstva zajišťuje komunikaci se serverem a zpracování nebo vytváření jednotlivých zpráv,
- **Controller** – vrstva částečně zajišťuje logické funkce aplikace a uchovávání hodnot potřebných pro chod aplikace,
- **GUI** – vrstva se stará o grafické zobrazení všech informací a zpracování požadavků od uživatele. Vrstva **GUI** je cílem mojí práce.



Obrázek 2.13: Obecné schéma aplikace pro inteligentní domácnost v rámci projektu IoT.

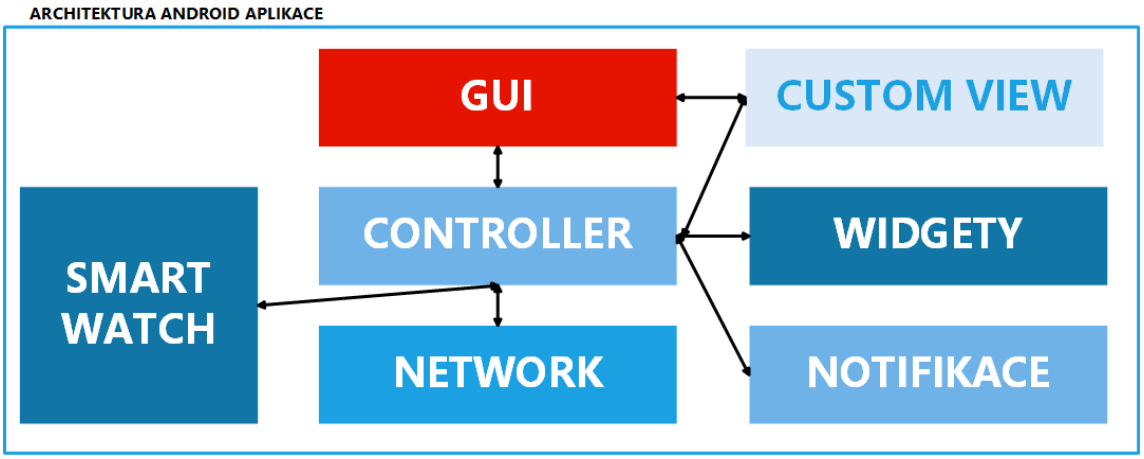
Cílem mé práce bylo vytvořit uživatelské rozhraní aplikace pro platformu Android. V průběhu vývoje aplikace na platformě Android se původní úkol rozšířil i o platformu Windows Phone, čím vznikají mírné odlišnosti od jednotlivých platforem, ale základní struktura aplikace, která byla uvedena výše, je společná pro obě platformy.

Vzhledem k tomu, že platforma Android byla zvolena jako první a byla tedy prioritní, byla struktura aplikace rozšířena o několik částí:

- **SmartWatch** – část *SmartWatch* je rozšířením umožňujícím spustit aplikaci i na chytrých hodinkách firmy Sony, konkrétně Smartwatch sw2,
- **Widget** – část *Widget* implementuje grafické objekty, které si uživatel může přidat na svoji plochu v OS Android,
- **Custom View** – část aplikace, která uživateli umožňuje flexibilně měnit pohledy na sbíraná data ze senzorů v inteligentní domácnosti,
- **Notifikace** – část *Notifikace* zajišťuje komunikaci se službou Google Cloud Messaging (GCM) a zpracovává zprávy, které přijdou právě přes tuto službu.

Celá struktura aplikace pro platformu Android je zobrazena na obrázku 2.14.

U platformy Windows phone nejsou žádná další rozšíření oproti obecnému schématu aplikace.



Obrázek 2.14: Schéma aplikace pro inteligentní domácnost v rámci projektu IoT pro platformu Android.

Kapitola 3

Ovládání domácnosti mobilním zařízením

Protože mluvíme o ovládání inteligentní domácnosti mobilním zařízením, pak je zjevné, že uživatelské rozhraní musí splňovat nejprve všechny základní parametry, které má mít uživatelské rozhraní pro mobilní zařízení, a to intuitivní ovládání přizpůsobené pro dotyková zařízení, a dále pak musí respektovat doporučený design pro danou mobilní platformu.

GUI by mělo dále podporovat celý proces práce se senzory, či aktory, v domácnosti, tedy nejenom její obsluhu jako zapínání, vypínání a získávání informací, ale také přidávání a odebírání jednotlivých zařízení. Pokud je systém navržen jako robustní, u kterého se očekává velké množství senzorů, pak by aplikace měla umožnovat jejich rozdělení, buď dle typů zařízení anebo dle lokace, resp. podle místností.

Přidávání nových zařízení do domácnosti je složitý úkon, v rámci kterého je potřeba zajistit vyřešení i několika dalších podmínek, např. spolupráci hardwarových zařízení apod. Z tohoto důvodu je nutné, aby GUI obsahovalo vždy alespoň základní nápovědu pro jednotlivé úkony.

3.1 Požadavky na ovládání inteligentní domácnosti v rámci projektu IoT

Systém vyvíjený v rámci projektu IoT je koncipován tak, aby mohl pojmout velké množství zařízení. Je tedy nutné umožnit uživateli roztřídění jednotlivých zařízení podle místností, v kterých se fyzicky nacházejí. Tento systém umožňuje uživateli spravovat více míst, jako je například domácnost a kancelář. Na každém místě pak musí být samotná brána, takže je nutné, aby aplikace umožňovala přepínání jednotlivých místností, resp. bran. Dále se předpokládá, že domácnost nebude ovládat pouze jediný uživatel, ale skupina uživatelů s různými přístupovými právy, např. rodina, tudíž je nezbytné, aby aplikace umožňovala správu uživatelů. V systému je zavedeno hned několik rolí, které mají různé úrovně přístupu a to:

- **Guest** – může pouze číst data jemu přidělené,
- **User** – práva Guest + možnost přepínat aktory,
- **Admin** – práva User + změna nastavení zařízení (přejmenování, změna refresh-time,...),
- **Owner** – práva Admin + správa účtů k bráně.

Jelikož systém předpokládá používání aktorů, mělo by GUI umožnit jejich nastavení. Existují tři typy aktorů:

- **zapnuto/vypnuto** – jedná se o přepínač, který přepíná mezi dvěma stavy (typickým příkladem může být žárovka),
- **termostat** – jedná se o speciální aktor, kterému lze nastavit konkrétní teplotní hodnotu,
- **více stavový** – jedná se o aktor, který má více stavů, mezi kterými lze přepínat (například změna módu termostatu).

Dalším požadavkem na GUI je přístup k nápovědě v několika rovinách. V první části bude uživateli ještě před akcemi přidání brány nebo senzoru/aktoru, vysvětlena práce s těmito HW zařízeními. Uživateli se zobrazí ilustrovaná nápověda s několika postupnými kroky. V druhé rovině bude, vzhledem ke složitosti GUI a rozsáhlosti aplikace, uživateli v několika krocích vysvětlena samotná práce s aplikací a její ovládání.

Posledním požadavkem na GUI je možnost při detailním zobrazení zařízení zobrazit graf s naměřenými či agregovanými hodnotami za několik posledních dnů.

GUI bude obsahovat tyto části a možnosti:

- **Správa zařízení** – přidat, odstranit a nastavit jednotlivé senzory,
- **Ovládání aktorů** – změnit stav aktoru,
- **Graf** – zobrazit historické hodnoty zařízení,
- **Místnosti** – možnost přidat či změnit místnost u zařízení,
- **Správa brány** – přidat a odstranit brány,
- **Správa uživatelů** – přidat, odebrat anebo změnit uživatelská práva vzhledem k bráně
- **Nápověda** – nápovědy k úkonům přidání zařízení, přidání brány a dále pak samotné vysvětlení práce v GUI.

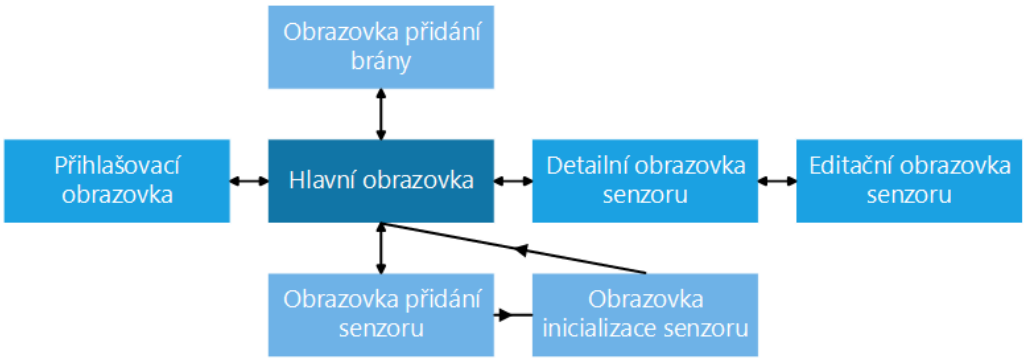
Kapitola 4

Návrh uživatelského rozhraní

Návrhy uživatelských rozhraní vychází jak z požadavků, které byly definovány v předcházející kapitole, tak z oficiálního grafického designu jednotlivých platforem. Každá verze GUI má svoji specifickou strukturu obrazovek vycházející z možností dané platformy. Pro potřeby GUI byla navíc vytvořena sada unikátních ikon pro jednotlivé místnosti, senzory a aktory. Sadu ikon můžete vidět v příloze diplomové práce.

4.1 Platforma Android

Uživatelské rozhraní je navrženo pro OS Android podporující verzi systému API 10, což odpovídá Android verzi 2.3. Podpora již od nízké verze Androidu má za následek, že některé grafické prvky nejsou podporované. Aby byly všechny grafické prvky použitelné až do API 10, je nutné použít několik knihoven (například. *Support library – v7 – Appcompat*). Celé GUI je navrženo dle aktuálního doporučeného grafického stylu a to *material-design* [1], který představila společnost Google.



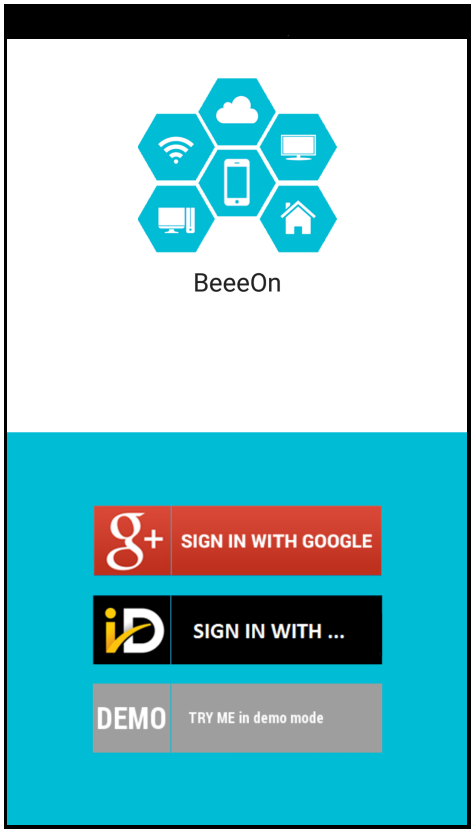
Obrázek 4.1: Struktura obrazovek v aplikaci pro platformu Android.

Na diagramu 4.1 je znázorněna provázanost jednotlivých obrazovek v GUI. První obrazovka je přihlašovací a slouží pro výběr přihlašovací metody. Dále obrazovka umožňuje vstoupit do demo režimu aplikace. Demo režim poskytuje uživateli možnost prohlédnout si celé GUI a vyzkoušet si všechny funkce aplikace. Po přihlášení do systému nebo spuštění demo režimu následuje hlavní obrazovka, která zobrazuje menu a seznam jednotlivých zařízeních. Po krátkém dotyku na položku v seznamu zařízení se objeví obrazovka s detailními

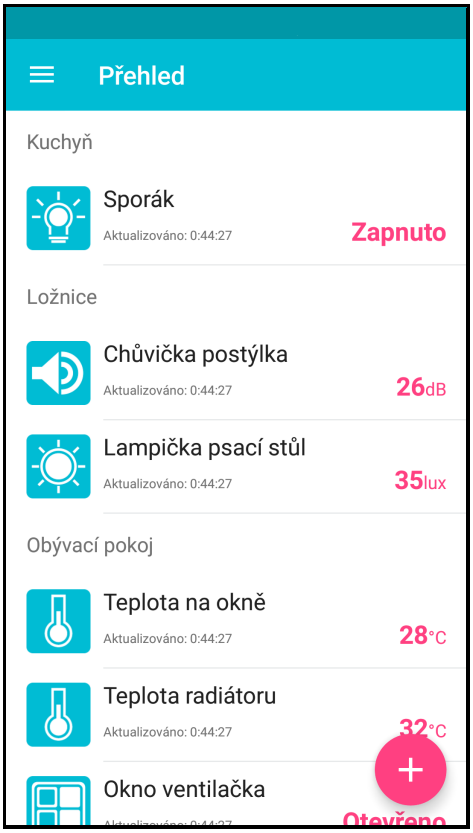
informacemi o senzoru. Menu obsahuje seznamy bran, přehled, grafy, sekci s aplikacemi a další tlačítka, jako je *Nastavení aplikace*, *O aplikaci* a *Odhlášení*. Po kliknutí v menu na tlačítko *Nastavení* se zobrazí obrazovka s dalšími parametry, které lze dále upravit.

Přihlašovací obrazovka

Přihlašovací obrazovka, kterou můžeme vidět na obrázku 4.2a, slouží k výběru přihlašovací metody. Při návrhu GUI byla implementována metoda přihlášení pomocí Google účtu, ale počítá se i s přihlašováním pomocí autority MojeID[1]. Další funkcí obrazovky je možnost vstoupit do aplikace v demo režimu.



(a) Přihlašovací obrazovka.



(b) Seznam zařízení.

Obrázek 4.2: Přihlašovací a hlavní obrazovka v aplikaci pro platformu Android.

Hlavní obrazovka

Hlavní obrazovku můžeme vidět na obrázku č. 4.2b. Základními funkcemi hlavní obrazovky jsou:

- zobrazit přehled zařízení,
- zobrazit vysouvací menu, které umožňuje přepínat mezi bránami a pohledy na domácnost (přehled, grafy) a
- možnost přidat další prvky domácnosti.

[1] www.mojeid.cz

Přehled zařízení je navržen tak, že vždy zobrazuje zařízení seskupené dle místností. Zařízení mohou být buď jednoduchá, tzn. že obsahují jeden senzor či aktor, a nebo složená, tzn. že obsahují více senzorů a nebo aktorů. Problém přehlednosti zobrazení jednotlivých zařízení byl vyřešen tak, že je odděluje jednoduchá linka. Vizualizace konkrétních senzorů nebo aktorů je navržena tak, že je zobrazena ikonka, informace o senzoru/aktoru jako je název a čas poslední aktualizace a jeho aktuální hodnota. Ikonka by měla uživateli pomoci s rychlou orientací v aplikaci.

V horní části Hlavní obrazovky můžeme vidět grafický element – jedná se o *Toolbar*. Tento grafický prvek je přítomný na všech obrazovkách mimo přihlašovací obrazovku. V prvku *Toolbar* se vždy nachází název aktuální obrazovky a případně ovládací prvek, který zobrazuje vysouvací menu (viz obrázek 4.2b) nebo šipku zpět (viz obrázek 4.3b) pro návrat na předcházející obrazovku. Další vlastností *Toolbaru* je možnost zapnutí *action mode*, který můžeme vidět v příloze na obrázku C.1b. Pokud je *action mode* zapnutý, změní se barva *Toolbaru* a zobrazí se další akce. Pro zapnutí *action mode* je použita událost dlouhého stisku na položce senzor či aktor v seznamu zařízení nebo na položce brána v menu.

Dalším důležitým prvkem na obrazovce je kruhové tlačítko vpravo dole, které se nazývá *floating action button* (dále jen fab). Tlačítko *fab* slouží pro přidávání dalších senzorů či aktorů a nebo bran do domácnosti. Jelikož je tento grafický prvek zaveden až v nejnovější verzi OS Android verze 5, lze předpokládat, že někteří uživatelé, nebudou ihned vědět, k čemu tento grafický prvek slouží. Zejména uživatelé využívající OS Android 2.3, by s tím mohli mít problém, proto se při prvním zobrazení objeví i nápověda. Nápovědu pro tlačítko můžete vidět v příloze na obrázcích C.5a a C.5b, které pomáhají uživateli pokud se má *fab* použít pro přidání brány nebo pro přidání zařízení.

Menu

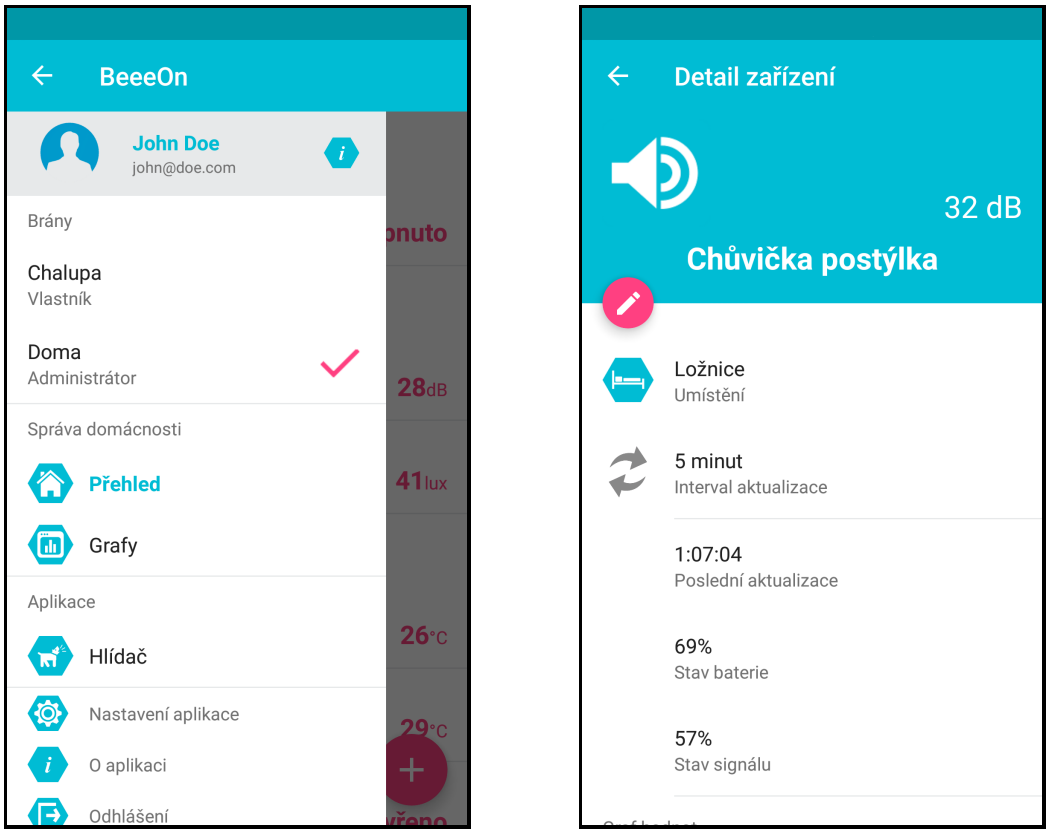
Vysouvací menu, které je zobrazené na obrázku 4.3a, je prvkem pro přechod do jiných částí Android aplikace. Bylo nadefinováno v požadavcích na ovládání systému BeeeOn, že aplikace musí umožnit uživateli přepínat mezi jednotlivými bránami, kde jedna brána symbolizuje jednu domácnost. Zmíněná možnost se nachází hned pod první položkou menu zobrazující informace o aktuálně přihlášeném uživateli, kde je sekce *Brány* pro výběr konkrétní brány. Kromě zvolení brány pomocí krátkého stisku je v aplikaci navržen způsob pro další práci s bránou a to pomocí dlouhého stisku. Po tomto dlouhém stisku na konkrétní bránu se spustí tzv. *action mode*, který umožní uživateli bránu přejmenovat, odstranit nebo zobrazit správu uživatelů dané brány (viz obrázek C.3a v příloze).

Další sekcí v menu je **Správa domácnosti**, která se vztahuje k aktuálně zvolené domácnosti. V této sekci jsou dvě položky a to *Přehled* a *Grafy*. Zvolením možnosti *Přehled* se uživateli zobrazí přehled zařízení. Zvolí-li uživatel *Grafy* zobrazí se jiný pohled na data z domácnosti. Obrazovka *Grafy* nebyla předmětem mé práce, ale v rámci návrhu menu jsem musel zajistit podporu jejího zobrazení.

Název třetí sekce je **Aplikace** a v ní jsou plánovány různé obrazovky, které nejsou řešením mé diplomové práce, ale obdobně, jako v předcházejícím případě, bylo nutné jim poskytnout podporu při návrhu v rámci menu. V současné době je v sekci *Aplikace* pouze jedna položka a to *Hlídač*.

Poslední část menu obsahuje následující položky:

- **Nastavení aplikace** – tlačítko slouží pro přechod do obrazovky nastavení aplikace,
- **O aplikaci** – při kliknutí na toto tlačítko se zobrazí dialogové okno s informacemi o aplikaci,
- **Odhlášení** – tlačítko slouží pro odhlášení uživatele a přesměrování na přihlašovací obrazovku.



(a) Zobrazení vysouvacího menu.

(b) Detail senzoru.

Obrázek 4.3: Vysouvací menu v hlavní obrazovce a detail senzoru v aplikaci pro platformu Android.

Detailní obrazovka senzoru/aktoru

Při návrhu obrazovky zobrazující detailní informace o senzoru či aktoru jsem se držel doporučeného stylu společnosti Google (*material design*). Tedy v detailu senzoru/aktoru, který je zobrazen na obrázku 4.3b, je první část barevně podbarvena stejně jako *Toolbar*. Část obsahuje nejdůležitější informace, a to ikonku senzoru/aktoru, která zobrazuje jeho typ, aktuální hodnotu či stav a nakonec jeho název. Na rozhraní první části je umístěno tlačítko, které slouží pro přechod na editační obrazovku aktuálního senzoru/aktoru. Toto tlačítko je zobrazeno pouze uživatelům, kteří mají právo měnit informace o senzoru/aktoru.

Další část obsahuje informace, které jsou nastavitelné, tzn. že je může uživatel na editační obrazovce přenastavit. Jedná se o nastavení umístění zařízení v místnosti a jeho interval aktualizace. V třetí části jsou zobrazeny informace, které jsou čistě informativní, tedy čas, kdy zařízení poslalo poslední data, a procentuální informace o stavu baterie a síle signálu. V poslední části, která není vidět na tomto obrázku, ale můžeme ji vidět v příloze na obrázku C.2d, je zobrazen graf s historickými hodnotami pro tento senzor/aktor. Pro tento graf byla použita komponenta, která byla předmětem bakalářské práce vytvořené v rámci projektu IoT.

Při návrhu detailu aktoru bylo dále zapotřebí vložit ovládací prvek pro změnu stavu aktoru. Tento ovládací prvek byl umístěn pod aktuální hodnotu v první části obrazovky. Byly navrženy dvě varianty ovládacích prvků. První je přepínač pro aktory, který nabývá pouze dvou stavů a to zapnuto resp. vypnuto (viz obrázek C.2a). Druhá varianta je tlačítko pro aktory, které nabývají více stavů, takže je vyvoláno dialogové okno pro jejich nastavení. V tomto dialogovém okně je buď seznam položek, mezi kterými uživatel přepíná, nebo v případě aktoru nastavujícího teplotu, je zde rolovací prvek s číselnými hodnotami.

Při zobrazení uvedené obrazovky se provede tzv. *pasivní aktualizace* dat, což znamená, že je-li čas poslední aktualizace starší než délka intervalu aktualizace, jsou vyžádána aktuální data ze serveru, jinak se požadavek neposílá. Další způsob aktualizace dat je pomocí gesta zhora-dolů, které provede tzv. *aktivní aktualizaci*, což znamená, že se provede dotaz na aktuální data ze serveru.

Pro jednodušší procházení senzoru či aktoru v rámci místnosti je navržena možnost přechodu na další, resp. předcházející, senzor/aktor pomocí gesta *swipe-right* resp. *swipe-left*. Gesto *swipe-left* znamená, že uživatel provede pohyb prstem po obrazovce zleva doprava a tím dojde k přechodu na předchozí senzor/aktor.

Editační obrazovka senzoru/aktoru

Editační obrazovka slouží pro přenastavení názvu senzoru/aktoru, umístění zařízení a aktualizačního intervalu. U změny umístění zařízení lze volit ze seznamu místností, který obsahuje reálné místnosti spolu s výchozími místnostmi. Uživatel si může také vybrat možnost vytvoření nové místnosti a zvolit si pro ni ikonku včetně vlastního názvu. Seznam výchozích místností je předefinovaným seznamem. Po nastavení všech položek je v *Toolbaru* tlačítko *uložit* pro odeslání dat na server. Obrazovku můžeme vidět v příloze na obrázku C.3c.

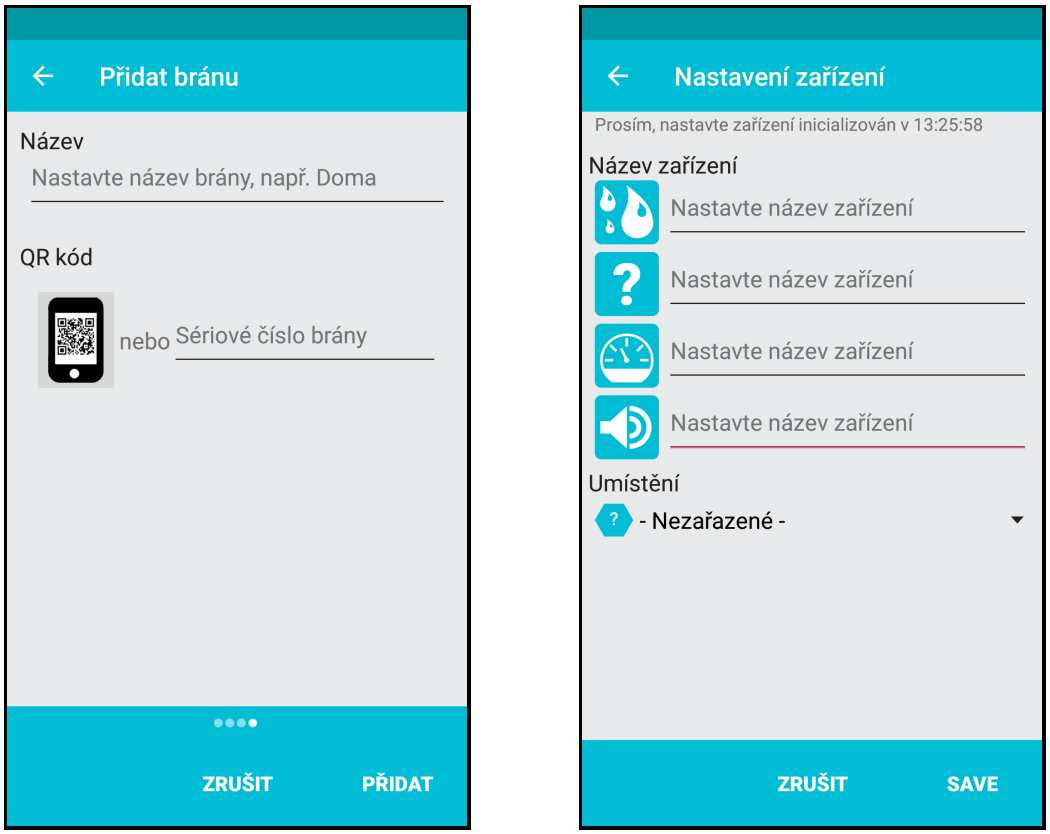
Obrazovka přidání brány

Pro přidání nové brány jsou nutné dva údaje – uživatelem zadané pojmenování brány a její unikátní číslo. Pojmenování musí uživatel zadat ručně, ale číslo brány lze zjistit naskenováním QR kódu, který by měl být umístěn na bráně. QR čtečka je zapnuta po kliknutí na tlačítko s obrázkem QR kódu. Pro případ, že by byl QR kód poškozen, nebo by uživatel nebyl fyzicky u zařízení, je na obrazovce možnost ručního zadání sériového čísla brány. Navrženou obrazovku můžeme vidět na obrázku 4.4a .

Obrazovka přidání zařízení a jeho inicializace

Přidání nového zařízení se skládá z dvou obrazovek a to z obrazovky informující o stavu procesu (viz obrázek C.4a) a obrazovky, která již provádí inicializaci nového zařízení. První obrazovka informuje uživatele o stavu procesu přidávání nového zařízení, resp. spuštění skenovacího režimu na bráně, aby byla schopná spárovat nové zařízení. Pokud se tento proces

provede, pak se na obrazovce zobrazí kruhový odpočet, aby uživatel věděl, kolik času mu zbývá, než se párovací režim na bráně ukončí. Druhá obrazovka se objeví v okamžiku, kdy je nalezeno nové zařízení. Pokud se jedná o složené zařízení, provede se na inicializační obrazovce zadání názvů jednotlivých senzorů a aktorů. Tento případ můžeme vidět na obrázku 4.4b. Dále si uživatel vybere umístění ze seznamu, který obsahuje již existující lokace v domácnosti spolu s předdefinovanými místnostmi. Tento seznam lze vidět na obrázku C.4b. Pokud by uživateli nevyhovovala ani jedna z možností daného seznamu, může si nadefinovat novou vlastní místnost, a to zvolením ikonky a zadáním vlastního názvu místnosti.



(a) Přidání nové brány.

(b) Inicializace nového zařízení.

Obrázek 4.4: Obrazovky přidávající HW zařízení – bránu či zařízení.

Nápověda

Nedílnou součástí každé aplikace je uživatelská nápověda. Jednou z variant nápovědy je vysvětlení samotného ovládání aplikace. Tento typ nápovědy byl použit pouze u zmiňovaného *fab*, ostatní grafické prvky byly navrženy tak, aby byly co nejvíce intuitivní a nevyžadovaly tak další vysvětlení. Pravděpodobně nejdůležitější nápovědou je ta, která se nachází při vyvolání akce přidání brány a přidání zařízení. Tyto nápovědy slouží pro uživatele jako průvodce, co má vykonat s bránou nebo se zařízením tak, aby bylo možné přidat zařízení do domácnosti.

Nápověda pro přidání brány je rozdělena do tří obrazovek. První obrazovka zobrazuje připojení brány k internetu. Druhá obrazovka vysvětluje, že je potřeba bránu připojit také do elektrické sítě. Poslední obrazovka obsahuje postup inicializace brány uživatelem. Jednotlivé obrazovky lze vidět v příloze na obrázcích C.7b až C.7d.

Další nápověda slouží pro přidání zařízení a je rozdělena do dvou kroků. První krok vysvětluje uživateli, jak zprovoznit zařízení, a to buď vytažením proužku oddělujícího baterii u senzoru, nebo zapojením aktoru do elektrické sítě (viz obrázek C.4c). Druhý krok, který můžeme vidět na obrázku 4.5b, vysvětluje uživateli, že po začátku odpočítávání 30 sekund má krátce zatřást senzorem nebo stisknout párovací tlačítko na aktoru.

Součástí aplikace je také úvodní nápověda, která se zobrazí pouze při prvním spuštění aplikace a vysvětluje celý systém domácnosti BeeeOn. Úvodní nápověda je rozdělena do pěti obrazovek, kde první obrazovka (viz obrázek C.6a) uvítá uživatele v systému BeeeOn. Druhá obrazovka, kterou můžeme vidět na obrázku 4.5a, ukazuje všechny komponenty domácnosti (bránu, senzor/aktor a propojení na telefon). Další obrazovky vysvětlují, co která komponenta vlastně znamená. Nejprve je vysvětlena činnost a spolupráce senzoru a aktoru (viz obrázek C.6c). Senzor měří hodnoty a aktor nastavuje požadované hodnoty. Na předposlední obrazovce (viz obrázek C.6d) je popsána brána, jako prvek, který sbírá data z bezdrátových zařízení a posílá je dále na server. Poslední krok (viz obrázek C.7a) nápovědy vysvětlí uživateli, jak může celou domácnost ovládat prostřednictvím svého mobilního zařízení.

(a) Ukázka z úvodní nápovědy.

(b) Ukázka z nápovědy pro přidání zařízení.

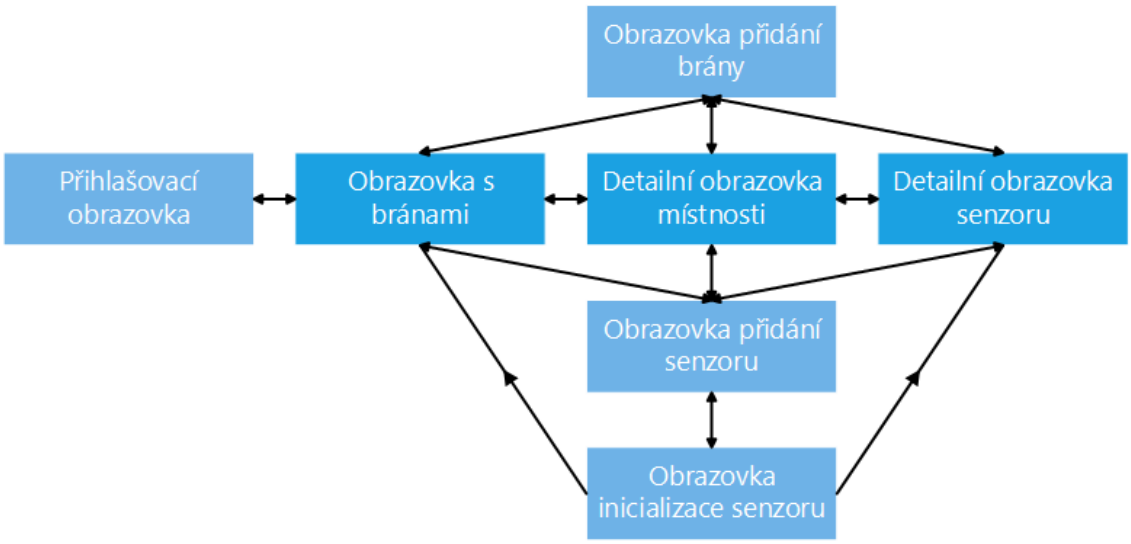
Obrázek 4.5: Ukázka obrazovek sloužící jako nápověda.

Všechny uvedené nápovědy mají stejný princip, kdy v horní části je umístěn vysvětlující popisek, pod ním je ilustrační obrázek a pod obrázkem je indikátor, který ukazuje jednotlivé kroky nápovědy. Ve spodní části obrazovky jsou tlačítka, která umožňují přeskočení celé nápovědy nebo její zrušení, případně přechod na další obrazovku. Dále byl navržen přesun mezi jednotlivými obrazovkami pomocí gest *swipe-left* a *swipe-right*. Všechny použité ilustrační obrázky byly vytvořeny speciálně pro vyvíjené GUI.

4.2 Platforma Windows phone 8.1

Aplikace pro OS Windows phone je vyvíjena jako *univerzální aplikace* [5] pro Windows phone 8.1, což znamená, že má oddělenou logickou a grafickou část. Je tedy možné použít jednu logickou část a dvě grafické části – jednu pro mobilní verzi a druhou verzi pro aplikaci na desktopu.

Na diagramu 4.6 lze vidět strukturu obrazovek uživatelského rozhraní. Jako první je zobrazena přihlašovací obrazovka, která slouží pro vstup do aplikace. Po přihlášení následuje obrazovka se seznamem brán. Po kliknutí na konkrétní bránu se objeví obrazovka se seznamem místností. Po kliknutí na místnost se zobrazí seznam zařízení v dané místnosti.



Obrázek 4.6: Struktura obrazovek v uživatelském rozhraní pro platformu Windows phone.

V následujícím textu jsou postupně popsány jednotlivé obrazovky aplikace. Tento popis je vždy doplněn o obrázek finálního návrhu.

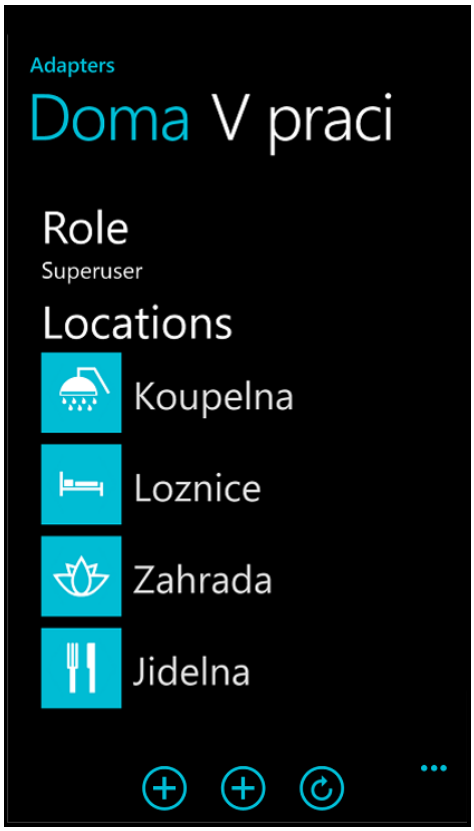
Přihlašovací obrazovka

Přihlašovací obrazovka je velmi jednoduchá a obsahuje pouze jedno tlačítko uprostřed pro přihlášení uživatele pomocí Google účtu. Jelikož platforma Windows phone neumožňuje jednoduché přihlášení přes Google účet, který je již v mobilním zařízení přidán, je uživatel po kliknutí na přihlašovací tlačítko přesměrován na webovou stránku, na které provede samotné zadání přihlašovacích údajů. Přihlašovací obrazovku lze vidět v příloze na obrázku D.1b.

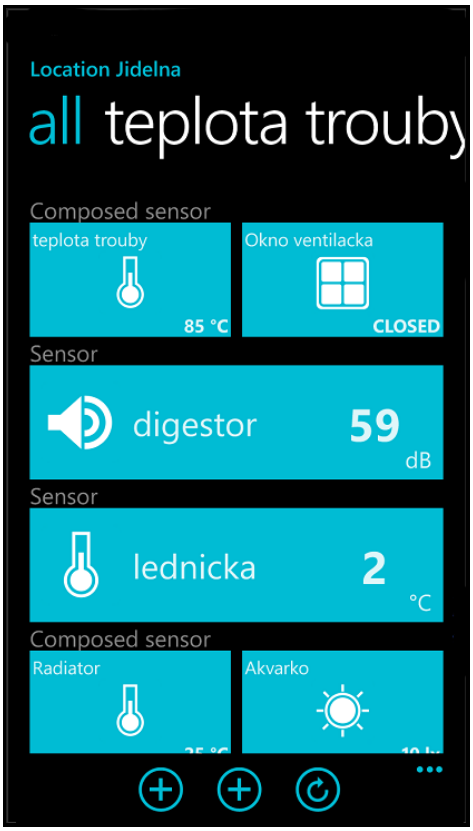
Obrazovka s bránami

Obrazovka obsahuje nejprve seznam všech brán a poté detail každé z nich. Vzhled obrazovky byl navržen dle doporučeného stylu společnosti Microsoft do dlaždicového rozvržení. Všechny elementy jsou navrženy jako čtvercové nebo obdélníkové grafické prvky a všechny obrazovky jsou navrženy tak, že se mezi položkami přesouvá pomocí gest *swipe-left* nebo *swipe-right*. Na první pozici je vždy seznam všech prvků, v tomto případě tedy všech bran. Při kliknutí na konkrétní bránu dochází k přesunu na detail brány. V detailním zobrazení brány je vidět uživatelská role pro danou bránu a seznam místností. Při kliknutí na místnost se přepne aplikace na obrazovku s detailem místnosti. Při dlouhém stisku položky v seznamu brán se zobrazí plovoucí menu, které obsahuje možnost odregistrace brány od uživatele. Obrazovku můžeme vidět na obrázku 4.7a.

Na všech obrazovkách je jeden společný prvek a to spodní vysouvací menu (*appbar*). V tomto menu jsou umístěna tlačítka pro přidání dalších zařízení do místnosti a aktualizace dané obrazovky. Po vysunutí menu se objeví další tlačítka a to pro *Nastavení*, *O aplikaci* a pro *Odhlášení*.



(a) Obrazovka detailu brány.



(b) Obrazovka detailu zařízení.

Obrázek 4.7: Ukázka obrazovek aplikace Windows phone.

Detailní obrazovka místnosti

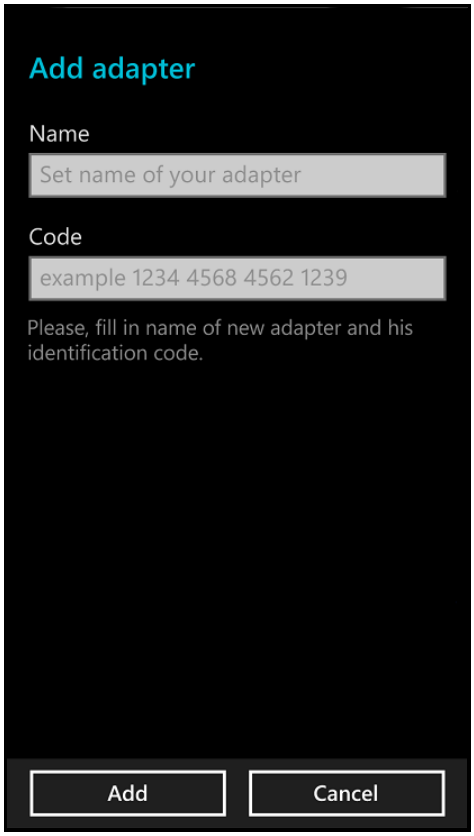
Cílem obrazovky s detailem místnosti je zobrazit všechna zařízení, která se nacházejí v jedné místnosti. Obdobně jako v předcházejícím případě, je první položkou posunovacího seznamu výčet všech místností z aktuální brány. V detailu místnosti je možné najít přehled všech

zařízení, který je rozdělen na jednotlivé senzory. U položky senzoru lze vidět obrázek, určující jeho typ, název a nakonec jeho aktuální hodnotu. Při kliknutí na senzor se aplikace přepne na detail senzoru. Na obrazovce lze opět najít výsuvné menu *appbar* se všemi dříve popsanými položkami. Obrazovku lze vidět v příloze na obrázku D.1a.

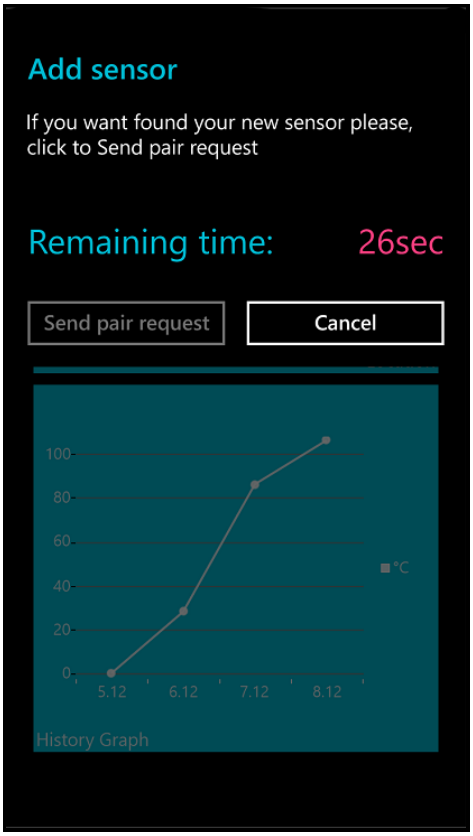
Detailní obrazovka senzoru/aktoru

Detailní obrazovku senzoru/aktoru můžeme vidět na obrázku 4.7b, kde je zobrazen přehled všech zařízení v místnosti. Na obrazovce je trochu jiné zobrazení než v detailu místnosti, i když ukazuje stejné informace. Jedná se o rozvržení, které má uživateli evokovat domovskou stránku samotného OS, kde jsou jednotlivé bloky různě velké. Dále se na obrazovce nachází detail senzoru, který je také rozdělen do různě velkých bloků. Nahoře jsou dva velké bloky, které obsahují ikonku senzoru a aktuální hodnotu. Pod těmito bloky jsou tři menší bloky, které zleva obsahují interval aktualizace, procentuální hodnotu baterie a síly signálu. Dále je zde zobrazena místnost, ve které se senzor nachází. Pod touto místností se nachází graf, který zobrazuje historické hodnoty.

Na obrazovce detailu senzoru/aktoru, lze měnit hodnoty, a to interval aktualizace a místnost, ve které se senzor nachází. Pro změnu těchto hodnot je navržena akce dlouhého stisku nad těmito bloky, poté se zobrazí plovoucí menu.



(a) Obrazovka přidání brány.



(b) Obrazovka přidání zařízení.

Obrázek 4.8: Obrazovky pro přidání nového HW do domácnosti v aplikaci Windows phone.

Obrazovka přidání brány

Obrazovka pro přidání brány, která je k vidění na obrázku 4.8a, je velmi jednoduchá. Jedná se o dialog, na který jsou uživatelé ve Windows aplikací zvyklí. Tento dialog obsahuje dva prostory pro zadání textu - první je název brány a druhý slouží pro zadání sériového čísla brány. Pokud by se vyskytla chyba během zadávání údajů, pak se v tomto dialogovém okně zobrazí chybová hláška. Pokud by nastala chyba až po odeslání dat na server, resp. při obdržení chyby od serveru, pak se informace objeví až v dalším dialogu.

Obrazovka přidání zařízení

Cílem obrazovky, znázorněné na obrázku 4.8b, je přidání nového zařízení do domácnosti BeeeOn. Opět je obrazovka řešena jako dialogové okno, které obsahuje informaci o zbývajícím čase - kolik sekund zbývá, než je ukončena možnost přidat nové zařízení k bráně. Jedná se o opatření proti krádeži zařízení, resp. získání přístupu k zařízení. Po úspěšném nalezení nového senzoru/aktoru se aplikace přepne na inicializaci daného zařízení, pokud ovšem není nalezeno nové zařízení, zaktivuje se tlačítko pro zaslání nového párovacího požadavku.

Obrazovka inicializace zařízení

Na obrazovku inicializace zařízení se uživatel dostane v případě, že bylo nalezeno nové neinicializované zařízení. Obrazovka má za úkol nastavit všem senzorům/aktorům jejich název a lokaci, kde se nacházejí. Uživatel si může vybrat ze seznamu místností, který obsahuje výchozí místnosti spolu s již vytvořenými místnostmi.

Kapitola 5

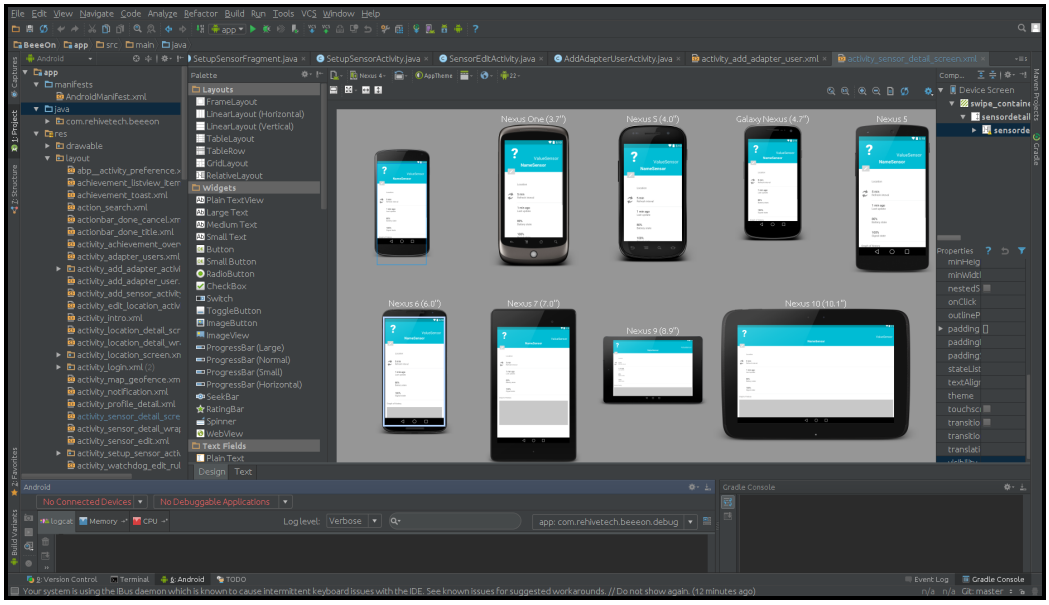
Implementace

V kapitole Implementace se nachází bližší popis vývoje uživatelských rozhraní na jednotlivých platformách – Android a Windows phone. Dále jsou zde popsány podpůrné nástroje, které byly použity při vývoji uživatelských rozhraní. Na závěr jsou shrnuty problémy, které nastaly během vývoje.

5.1 Vývoj na platformě Android

Vývoj uživatelského prostředí započal na začátku roku 2014, kdy bylo standardním a doporučeným vývojovým prostředím IDE Eclipse s rozšíření ADT (*Android Developer Toolkit*), které umožňovalo spuštění emulátorů a několik diagnostických nástrojů jako *ADB logs* nebo *debug*. Ovšem stále se jednalo o obecné IDE, které bylo pouze rozšířené o modul podporující platformu Android. V té době již bylo vyvíjeno nové specializované IDE AndroidStudio [4], které ovšem bylo v brzkém stádiu vývoje a nebylo dostatečně stabilní. A tak první verze uživatelského rozhraní vznikala pod IDE Eclipse. Situace se změnila až koncem roku 2014, kdy byla uvolněna první stabilní verze IDE AndroidStudio a bylo doporučeno ze strany společnosti Google přejít na toto vývojové prostředí.

Vývojové prostředí AndroidStudio je vyvíjeno přímo společností Google, takže se dá očekávat co největší podpora pro vývoj na platformu Android. AndroidStudio je postaveno na jádře InteliJ IDEA, které vyvíjí společnost JetBrain. Mezi základní změny oproti IDE Eclipse patří změna překladového systému na *Gradle*. Díky tomuto překladovému systému je například možné definovat různé varianty generovaných APK (finálních spustitelných souborů) nebo definování knihoven, které se automaticky stahují z centrálního repositáře. Odpadá tak problém ručního stahování knihoven a jejich vkládání do projektu. Další výhodou uvedeného vývojového prostředí je z pohledu uživatelského rozhraní možnost velmi jednoduše vygenerovat náhledy v různých rozlišeních, velikostech obrazovky a nebo v jazycích. Na obrázku 5.1 je vidět ukázka IDE AndroidStudio a vytvořený náhled obrazovky - detail senzoru na několika základních typech mobilních zařízení. Také při samotné práci s obrázky a řetězci přináší AndroidStudio novinky, a to například při najetí myši na odkaz k obrázku nebo k řetězci nabídne ihned náhled dané položky, což velmi zrychlilo vývoj celého uživatelského rozhraní. Jedna z posledních výhod, které byly přidány do AndroidStudia až v poslední verzi, je zobrazení zátěže CPU a paměti RAM, což umožňuje velmi rychle reagovat na neadekvátní náročnost při testování uživatelského rozhraní.



Obrázek 5.1: Ukázka IDE AndroidStudio s náhledy obrazovky na několika základních typech mobilních zařízení.

Nyní budou popsány knihovny, které vyvíjí společnost Google a to odděleně od zdrojových kódů jednotlivých API. Důvod proč jsou knihovny oddělené je jednoduchý. Jedná se o tzv. *support libraries*, které doplňují funkcionalitu či grafické prvky nejnovějších API do starších API. Pokud je aplikace vyvíjena přímo pro nejnovější API není žádný důvod podporující knihovny použít.

Support library – v4

Knihovna *Support library – v4* [18] přidává podporu základních stavebních prvků až do API verze 4 (odtud je označení v4). Mezi základní prvky pro tvorbu uživatelského rozhraní patří:

- *Fragment* – Komponenta *Fragment* umožňuje tvorbu obrazovek nejenom na úrovni *Activity* (což celá obrazovka), ale i na jemnější prvky tak, aby se dala vytvořit obrazovka, která bude obsahovat více fragmentů.
- *ViewPager* – Komponenta *ViewPager* umožňuje vytvořit obrazovku, která obsahuje více fragmentů,mezi kterými se lze pohybovat pomocí gest *swipe-left* a *swipe-right*.
- *DrawerLayout* – Komponenta *DrawerLayout* umožňuje zobrazení vysouvacího menu spolu s tzv. *hamburgrem*, což je ovládací prvek pro zobrazení menu.

Aktuální revize má číslo 22.1 a přidává pouze "kosmetické" úpravy. Uvedená knihovna má více prvků, které pomáhají vývojářům přizpůsobit aplikace dle nejnovějších trendů a elementů, ale já jsem zde jmenoval pouze ty komponenty, které jsem využil při tvorbě uživatelského rozhraní.

Support library – v7 – appcompat

Knihovna *Support library – v7 – appcompat* [18] přidává podporu důležitým prvkům až do API 7. Mezi důležité prvky pro tvorbu uživatelského rozhraní patří:

- *ToolBar* – Dříve se komponenta *ToolBar* jmenovala *ActionBar*. Implementuje většinou horní prostor obrazovky, kde je ovládací prvek pro vysouvací menu (*hamburger*) nebo další ovladací prvky. Uvedená komponenta je také důležitá skrze to, že umožňuje použití *action mode*.
- *ActionBarActivity* – Komponenta *ActionBarActivity* navazuje na *ToolBar* a umožňuje s ním pracovat v aktivitě.

Knihovnu *Support library – v7 – appcompat* využívám v revizi 22, i když je k dispozici novější revize 22.1. Poslední revize však velmi zasahuje do všech grafických elementů, takže ji nebylo možno použít, protože by se v podstatě jednalo o přepsání celého uživatelského rozhraní. Jak bylo popsáno u předcházející knihovny, tak i zde knihovna *Support library – v7 – appcompat* obsahuje další obsah, ale popsal jsem pouze ty prvky, které používám při tvorbě uživatelského rozhraní.

5.1.1 Knihovny třetích stran

Protože je komunita vývojářů pro platformu Android rozsáhlá, vznikají tak stále nové a nové knihovny, které přidávají různé rozšiřující prvky. Nejčastější publikování těchto knihoven je na serveru GitHub, což jsou většinou knihovny šířené pod *Apache 2.0 License*.

FloatingActionButton

Knihovna *FloatingActionButton* [27] implementuje rozšiřující prvek *floating action button*, který byl popsán v *material design*. Tento prvek ale nebyl implementován společností Google přímo do SDK ani do *support* knihoven. Uvedená knihovna, na rozdíl od podobných knihoven, implementovala tzv *autohide*, což znamená, že pokud je tento prvek napojen na seznam, pak se při pohybu směrem dolů tlačítko schová a naopak při pohybu směrem nahoru se tlačítko zobrazí. Pomocí rozšiřujících úprav (*pull-requestu*) od dalších autorů na serveru GitHub se rozšířila knihovna i o podmenu, které se zobrazí po stisknutí na uvedené tlačítko. Knihovnu jsem použil na hlavní obrazovce pro akce přidání dalších HW zařízení do domácnosti.

StickyListHeader

Knihovna *StickyListHeader* [29] implementuje rozšíření nad klasickým seznamem (*ListView*) o funkci, kdy se hlavička sekce v seznamu přichytí při pohybu seznamu směrem dolů k horní části obrazovky do té doby, nežli ji odsune další hlavička seznamu. Knihovnu jsem použil v menu, kde názvy jednotlivých sekcí jsou hlavičky tohoto seznamu.

ViewPagerIndicator

Knihovna *ViewPagerIndicator* [32] implementuje rozšíření nad prvkem *ViewPager*, které přidává do tohoto rolujícího se seznamu obrazovek (*fragments*) grafickou část zobrazující aktuální stav – na které z kolika obrazovek se uživatel nachází. Uvedenou knihovnu jsem velmi využíval při tvorbě nápověd a tutoriálů. Použití knihovny můžeme vidět například na obrázku C.4c zobrazující první krok tutoriálu pro přidání nového zařízení.

CircleProgress

Knihovna *CircleProgress* [26] implementuje grafický prvek kruhové odpočítávání času. Tento prvek je použit čistě z estetických a trendových důvodů. Kruhové odpočítávání času je použito u obrazovky pro přidání nového zařízení a můžeme ho vidět na obrázku C.4a.

ShowCaseView

Knihovna *ShowCaseView* [24] implementuje funkci tzv. "zašednutí a vysvětlení". Tento princip "zašednutí a vysvětlení" využívaly velmi často aplikace od společnosti Google, ale popsaná funkcionalita nikdy nebyla implementována přímo do SDK nebo do rozšiřující knihovny. Uvedenou knihovnu jsem použil pro vysvětlení grafického prvku *floating action button*, který je moderní, ale je vcelku pravděpodobné, že pokud uživatel nepoužívá poslední OS Android verze 5, pak tento grafický prvek uvidí poprvé. Použití knihovny je zobrazeno na obrázcích C.5a a C.5b.

StyledDialogs for Android

Knihovna *StyledDialogs for Android* [31] rozšiřuje klasické dialogy o podporu nového grafického stylu *material desing* pro starší verze OS Android. Další výhodou knihovny je, že ulehčuje vývojáři obsluhu událostí jako je např. otočení obrazovky, kdy v rámci uvedené události dojde k vymazání dat a vývojář se musí postarat o uložení dat a jejich případné znovu-načtení. Ovšem knihovna podobné události obsluhuje sama. Pokud se vybere v dialogu nějaká položka seznamu, pak i při jeho otočení zůstane vybraná položka označená. Použití knihovny lze vidět na obrázku C.5c.

NumberPicker

Knihovna *NumberPicker* [30] implementuje podporu stejnojmenného prvku v SDK platformy Android pro starší verze OS Android než-li 4.2. Jedná se o grafický prvek, který vybírá číslo ze seznamu a to pomocí "*scrolování*". Použití knihovny v kombinaci s použitím předcházející knihovny *StyledDialogs for Android* lze vidět na obrázku C.5d.

Grafová knihovna

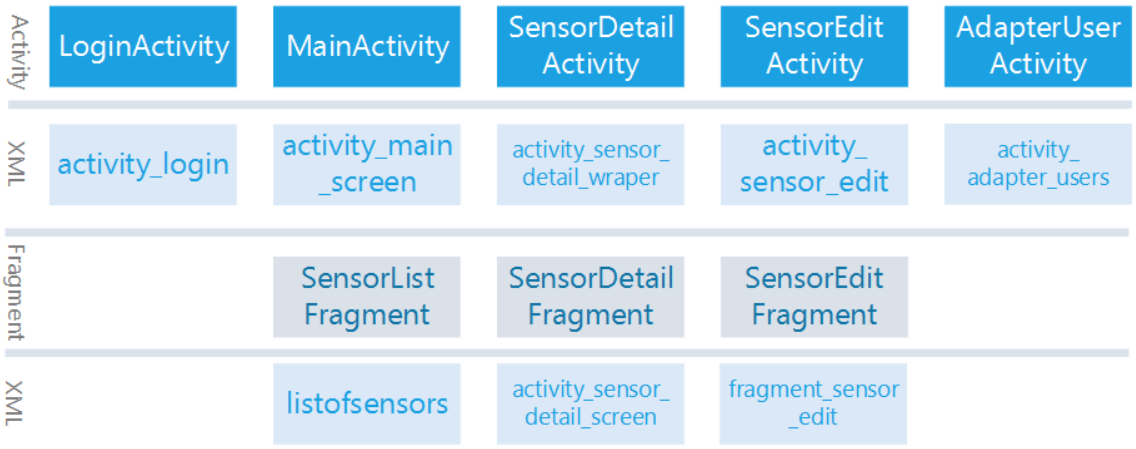
Nakonec byla použita knihovna pro grafy, která je vyvíjena jako bakalářská práce na Fakultě informačních technologii VUT v Brně. Knihovna je uzpůsobena potřebám domácnosti BeeeOn a na obrázku C.2d lze vidět její použití.

5.1.2 Struktura uživatelského rozhraní

Jak byly v kapitole Návrh graficky a funkčně navrženy jednotlivé obrazovky, tak byly i implementovány. Nyní bych se rád věnoval implementační struktuře uživatelského rozhraní pro platformu Android.

Na obrázku 5.2 můžeme vidět strukturu základních obrazovek. Na tomto obrázku jsou obrazovky (vertikální směr) rozděleny dle jednotlivých vrstev, a to *Activity*, *XML (layout activity)*, *Fragment* a *XML (layout fragmentu)*. Všechny *Activity* jsou odvozeny od *ActionBarActivity*, která byla popsána dříve u knihovny *Support library – v7*. Důvodem proč většina obrazovek není implementována pouze pomocí *Activity*, ale i pomocí *fragment*, je ten, že tento způsob umožňuje rozdělit práci s GUI tak, že v *Activity* je implementována

spíše logická stránka GUI a pak samotné *fragment* obsahuje pouze grafickou stránku GUI. Tedy kromě skutečnosti, že rozdělení pomáhá k přehlednosti kódu, umožňuje práce s *fragment* také jednodušeji rozšířit GUI například o rozvržení pro tablety.

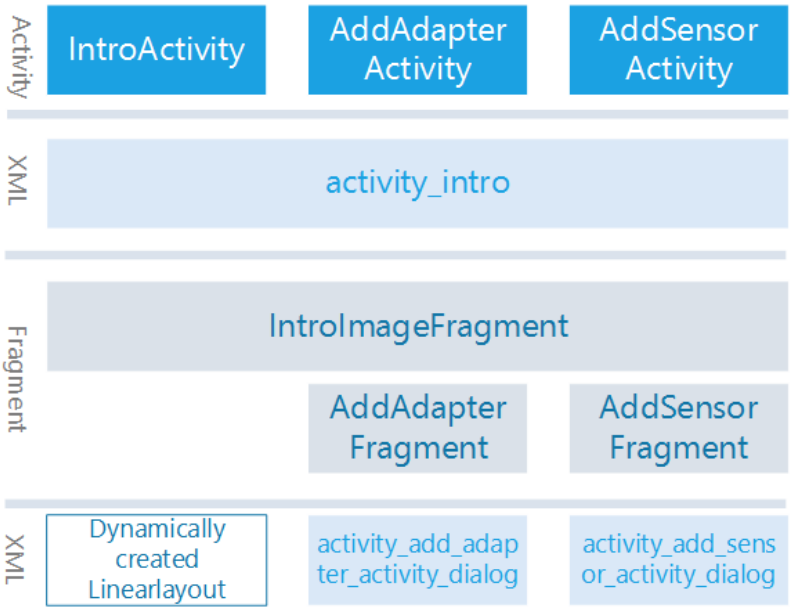


Obrázek 5.2: Struktura GUI z pohledu implementace základních obrazovek.

U obrazovky detailu senzoru či aktoru (*SensorDetailActivity*) má použití *fragementu* ještě další důvod a to, že byla navržena možnost přesouvat se pomocí gest *swipe-left* a *swipe-right* k dalším senzorům či aktorům. Obrazovka detailu senzoru/aktoru pak obsahuje několik takovýchto detailů. Z tohoto důvodu je zde použit *FragmentStatePagerAdapter*, který umožňuje výše popsanou funkcionalitu, tj. přesunout se při uvedených gestech na další detail. Jeden detail senzoru/aktoru pak znamená jeden *fragment*.

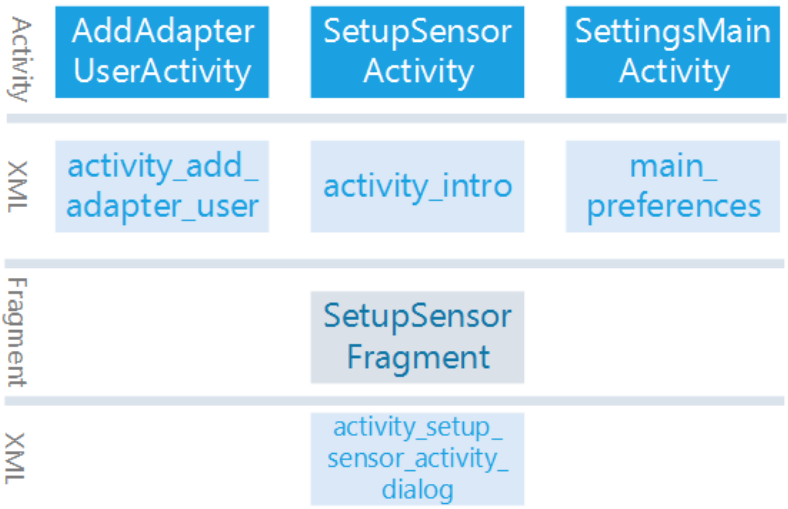
Hlavní obrazovky slouží mimo zobrazování *fragmentu* seznamu zařízení i k zobrazení dalších *fragmentů* (grafy, hlídač), které ale již nejsou součástí mé práce. Z tohoto důvodu se v uvedené struktuře vyskytuje pouze seznam zařízení, která byla dána zadáním mé práce. Dalším prvkem, který můžeme najít na hlavní obrazovce, je vysouvací menu. Jelikož byl obsah menu popsán již v kapitole Návrh, uvedu zde pouze, že menu bylo implementováno pomocí *StickyListHeader* a bylo potřeba implementovat různá rozvržení pro jeho položky. Pro sekci profil v menu šlo o speciální rozvržení *drawer_listview_profile* a pro hlavičky sekcí (Brány, Správa a Aplikace) bylo implementováno rozvržení *drawer_listview_group*. Pro ostatní položky bylo použito jedno rozvržení *drawer_listview_location* a pouze u položek bran nebyl nastaven obrázek.

Na obrázku 5.3 je zobrazena struktura obrazovek obsahující nápovědu. Jedná se o obrazovky úvodní nápověda (*IntroActivity*), přidání brány (*AddAdapterActivity*) a přidání nového zařízení (*AddSensorAcitity*). U těchto obrazovek byl použit *IconPagerAdapter* z knihovny *ViewPagerIndicator*, který obsahuje opět více fragmentů. V tomto případě jeden *fragment* představuje jeden krok nápovědy. Poslední *fragment* obsahuje *fragment* vykonávající požadovanou akci. Díky tomu, že všechny obrazovky pro nápovědu mají stejnou strukturu (vysvětlující text a konkrétní obrázek), mají stejné základní rozložení a pouze obsahují jiné texty a obrázky. Ovšem poslední *fragment* je pro každou obrazovku nápovědy speciální.



Obrázek 5.3: Struktura GUI z pohledu implementace nápovědy.

Zbývající obrazovky, resp. jejich struktura, je zobrazena na obrázku 5.4. Obrazovky nejsou nijak komplikované, takže obrazovky pro přidání uživatele k bráně (*AddAdapterUserActivity*) a nastavení aplikace (*SettingsMainActivity*) jsou implementovány pouze jako *Activity*. Naopak u obrazovky pro inicializaci zařízení (*SetupSensorActivity*) je potenciál pro rozšíření rozložení pro tablet, takže je zde výhodné implementovat i *fragment*.



Obrázek 5.4: Struktura GUI z pohledu implementace dalších obrazovek.

5.2 Vývoj na platformě Windows phone

V podkapitole je rozebrán vývoj uživatelského rozhraní pro platformu Windows phone. Zmíněno je zde jak vývojové prostředí, tak knihovny, které byly použity pro grafy. Dále je v textu rozebrána implementační struktura uživatelského rozhraní pro platformu Windows phone.

5.2.1 Vývojové prostředí

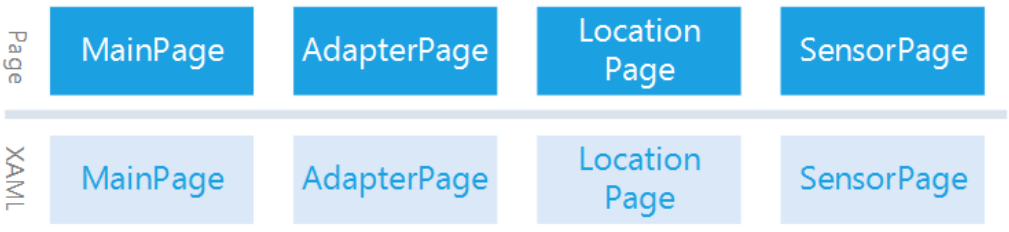
Pro vývoj *univerzální aplikace* je nutné použít IDE VisualStudio 2013, což rozhodně není žádné negativum, ba naopak práce s tímto IDE je velmi příjemná. Při vývoji jsem narazil na problém při použití vestavěného emulátoru pro Windows phone. Tento emulátor lze použít v případě, kdy máte OS Windows 8.1 Pro. Okolnostmi jsem tak byl donucen dokoupit si rozšíření pro OS Windows 8.1 na edici Pro, aby se mohl do OS doinstalovat vizualizační nástroj *Hyper-V*. Po prvotních komplikacích se spuštěním emulátoru s OS Windows phone 8.1 pak probíhaly práce vcelku rychle.

5.2.2 Knihovny třetích stran

Přestože se systém Windows phone resp. *univerzální aplikace* tváří jako celkem uzavřený systém, existuje pro platformu knihovna, kterou vyvíjejí příznivci platformy. Jedná se o *WinRT XAML Toolkit for windows 8.1/windows phone 8.1* [21]. Knihovna má mnoho komponent, které ulehčují vývojářům jejich práci. Já jsem si z knihovny vybral pouze jednu komponentu, a to *Chart* neboli grafy. Komponenta *Chart* byla vcelku obtížně integrována do uživatelského rozhraní aplikace pro platformu Windows phone, ale nakonec se mi ji v uživatelském rozhraní povedlo použít.

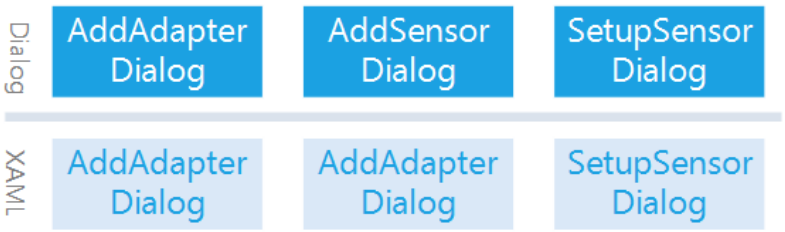
5.2.3 Struktura uživatelského rozhraní

Na rozdíl od uživatelského rozhraní pro platformu Android platforma Windows phone neumožňuje žádné dělení logiky práce v GUI na jemnější části. Proto na platformě Windows phone platí, že jedna obrazovka se rovná dvěma částem, a to soubor, *.cs, v kterém je umístěna celá logika GUI a soubor, *.xaml, v kterém je definované samotné GUI. Na obrázku 5.5 můžeme vidět všechny obrazovky uživatelského rozhraní. Všechny obrazovky, mimo přihlašovací (*MainPage*), jsou implementovány jako *PivotPage*, což znamená něco podobného jako *ViewPager* u platformy Android. Na těchto obrazovkách se můžeme přesouvat k dalším položkám pomocí gest *swipe-left* a *swipe-right*.



Obrázek 5.5: Struktura GUI z pohledu implementace obrazovek.

Na obrázku 5.6 můžeme vidět všechny dialogy, které jsou implementovány v uživatelském rozhraní. U těchto dialogů platí to stejné, jako u *Page*, tj. že jsou rozdělené na soubory *.cs a *.xaml. U dialogu se implementuje pouze obsah dialogu a ovládací tlačítka jsou vždy zobrazena.



Obrázek 5.6: Struktura GUI z pohledu implementace dialogů.

5.3 Problémy při vývoji uživatelského rozhraní

Během práce na uživatelských rozhraních jsem narazil na mnohé problémy, které jsem musel vyřešit. Nyní bych se rád zmínil o několika zásadních problémech. Jeden z největších problémů nastal při přechodu z IDE Eclipse na nové AndroidStudio, kdy bylo potřeba změnit v podstatě celý projekt od základů, včetně použití nového překladového systému Gradle. I když bylo na oficiálních stránkách pro vývojáře na platformě Android jasně popsán postup převodu celého projektu, nebylo možné tento převod ani po několika pokusech zprovoznit. Řešením bylo nakonec vytvoření nového čistého projektu v AndroidStudiu a postupné ruční překopírování zdrojových kódů. Dalším častým problémem byla nekompatibilita knihoven s API 10, kdy jsem hledal nové knihovny, které by mi pomohly při tvoření uživatelského rozhraní. Knihovny byly často implementovány pro nové API a nebo propagovaly ve svých dokumentacích, že podporují i nízké API 7 či 10, ale nebyla to pravda.

Vcelku zásadní problém nastal při vyhledávání informací k platformě Windows phone, protože uživatelské rozhraní bylo vyvíjeno jako univerzální aplikace, což znamená pro platformu Windows phone 8.1. Předcházející verze OS Windows phone byla 8, tudíž při vyhledávání informací, se velmi míchaly informace pro verzi 8 a 8.1. Na první pohled by se mohlo zdát, že uvedené verze budou velmi podobné a budou se lišit pouze v maličkostech, ale opak je pravdou. Z pohledu vývoje se jedná o velký rozdíl, kdy například všechny synchronní metody byly nahrazeny asynchronními. Takových příkladů by bylo možné najít celou řadu, ale ve zkratce se dá říci, že se jedná o velmi odlišné OS.

Kapitola 6

Vyhodnocení

Testování uživatelského rozhraní pro platformu Android probíhalo v několika úrovních. První úroveň testování probíhala na několika různých emulátorech přímo při vývoji, čímž se otestovaly všechny pravděpodobné scénáře práce s uživatelským rozhraním. Další úroveň testování byla již více automatizovaná a náhodná. V uvedené úrovni testování byly použity dostupné automatizované testy, které odhalily celkem 15 chyb a přispěly k celkové stabilitě výsledné aplikace. Poslední úroveň testování byla prováděna jako *alpha* a *beta* testování. Byly vydány *aplha* a *beta* aplikace, ve kterých bylo obsaženo také mnou vytvořené uživatelské rozhraní. *Aplha* aplikace byla testována všemi vývojáři inteligentní domácnosti (skupiny senzory, brána, atd.). *Beta* aplikace byla otestována v reálném nasazení v 5 domácnostech. Během testování v domácnostech bylo zaznamenáno celkem 77 chyb, které byly následně opraveny. Výsledná aplikace pro platformu Android byla úspěšně otestována na desítkách odlišných zařízeních s různou verzí operačního systému.

Testování uživatelského rozhraní pro platformu Windows phone probíhalo pouze ve dvou úrovních. První úrovní bylo opět testování přímo při samotném vývoji na emulátorech, kde se testovaly všechny pravděpodobné scénáře použití uživatelského rozhraní. Druhá úroveň testování spočívala v *aplha* testování a to u několika uživatelů.

Základní scénář testování na emulátorech spočíval v otestování všech funkcionalit uživatelského rozhraní. Postup kroků testování byl vždy použit při přidání nebo opravě funkce. Na začátku testování byla aplikace smazána ze zařízení tak, aby se dalo ověřit uživatelské rozhraní zcela od prvotních úkonů. V průběhu všech kroků byla dynamicky otáčena obrazovka tak, aby se zjistila plynulost reakce uživatelského rozhraní na polohu zařízení. Kroky testování byly následující:

1. Průchod úvodní nápovědy.
2. Přihlášení do demo režimu aplikace a ověření všech zobrazených položek a jejich detailů.
3. Odhlášení z aplikace a znovu přihlášení pomocí Google účtu.
4. Přidání první brány a průchod nápovědy.
5. Přidání prvního zařízení do výchozí místnosti a průchod nápovědy.
6. Přidání druhého zařízení do nově vytvořené místnosti.
7. Zobrazení detailu senzorů a aktorů.

8. Editace aktualizačního intervalu a umístnění zařízení.
9. Přepnutí stavu aktoru.
10. Zobrazení správy uživatelů brány a následné přidání nového uživatele k bráně.
11. Změna role u uživatele brány.
12. Smazání jednoho zařízení.
13. Přidání druhé brány.
14. Smazání první brány.
15. Minimalizace a následná maximalizace aplikace.

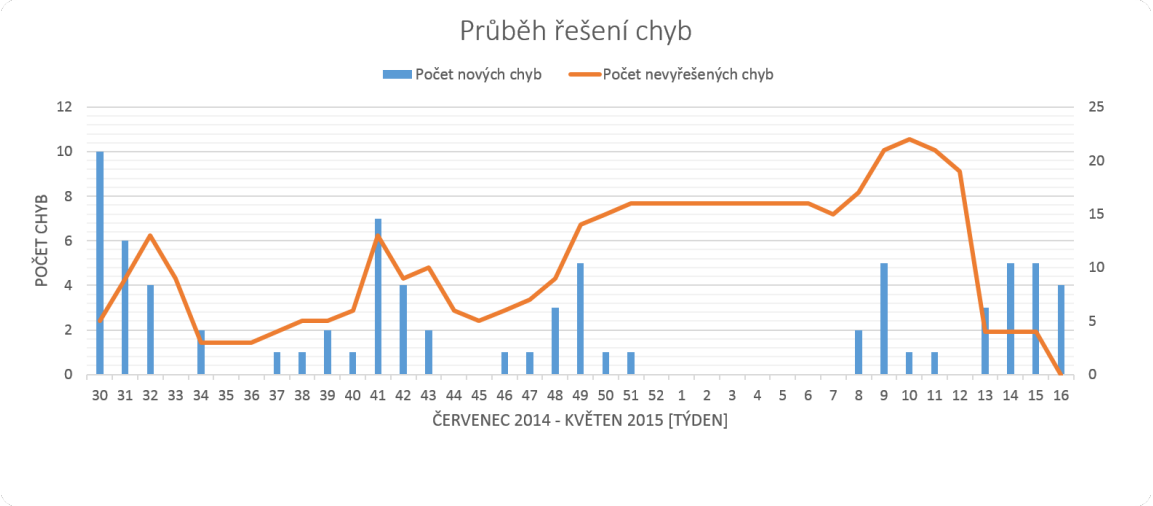
Při testování uživatelského rozhraní jsem využíval dostupného emulátoru brány, což mi umožňovalo velmi flexibilní testování a nebyl jsem tak vázán na fyzické zařízení.

Aplha testování aplikace pro OS Android probíhalo již od brzkého vývoje celé aplikace. Důvodů bylo několik. Jedním z nich bylo umožnění i ostatním vývojářům testovat své části v rámci inteligentní domácnosti. Dalším důvodem bylo, abych měl zpětnou reakci od ostatních vývojářů tak, aby se uživatelské rozhraní nevyvíjelo úplně samostatně. Výhodou testování byla rozmanitost zařízení, na kterých byla aplikace nainstalována. *Alfa* testování se zúčastnilo celkem 20 vývojářů s různými typy zařízení, např. *custom-rom* cyanogenmod nebo mobilní zařízení od společnosti Xiaomi, které obsahuje velmi upravenou verzi OS Android. Samozřejmě zde byla zastoupena i mobilní zařízení, která mají většinu na trhu a to mobilní zařízení od společnosti Samsung. V rámci testování se objevila i velmi stará zařízení s OS Android 2.3, tudíž bylo možné na nich aplikaci také otestovat. Problém nebyl ve stáří zařízení, ale ve velikosti obrazovky, která byla kolem 3". I díky těmto "extrémním"variantám bylo možné uživatelské rozhraní optimalizovat pro uvedené případy tak, aby vždy zobrazení na jakémkoliv podporovaném zařízení bylo korektní. V rámci optimalizace pro velmi malé obrazovky byla do uživatelského rozhraní přidána vlastnost rotování textů v případě malého prostoru tak, aby si uživatel mohl přečíst celý text. Pro účely testování bylo vydáno kolem 40 verzí aplikace.

Beta testování aplikace pro OS Android začalo na začátku roku 2015 v 5 domácnostech. Testování již probíhalo na "standardních"zařízeních, takže získané informace měly charakter zpětné vazby z pohledu běžného uživatele, tj. zda je rozhraní přehledné nebo pochopitelné apod. Jedním z příkladů bylo zobrazení výsledků akcí v aplikaci. Na základě zpětné vazby od uživatelů, podle kterých nebyla informace o úspěšnosti akce dostatečně srozumitelná, pak byly v aplikaci provedeny změny směřující k nápravě. Pro vývoj uživatelských aplikací jsou takové zpětné vazby od "běžných"uživatelů nezbytné.

Aplha i *Beta* aplikace byly šířeny pomocí oficiálního obchodu OS Android *GooglePlay*, kde byla aplikace napojena i na nástroj *Google Analytics*. Tento nástroj měl pro ladění velký přínos. Pokud nastala chyba v uživatelském rozhraní, pak se ke mně dostaly informace o tom, kde přesně v kódu je chyba. Pokud uživatel vyplnil i popis, mohl jsem provést simulaci uvedené akce, abych ověřil správnost provedené opravy.

Na obrázku 6.1 je znázorněn průběh řešení chyb, které byly zaznamenány v rámci *aplha* a *beta* testování. V grafu lze vidět dva typy hodnot, a to počet nových chyb a počet nevyřešených chyb. Uvedené hodnoty jsou agregovány po týdnech. Z grafu lze vyčíst, že na začátku testování se objevilo velké množství chyb, což je vcelku pochopitelné. Po zhruba jednom až jednom a půl měsíci je v grafu vidět zvýšený počet chyb, který byl způsoben



Obrázek 6.1: Graf průběhu chyb.

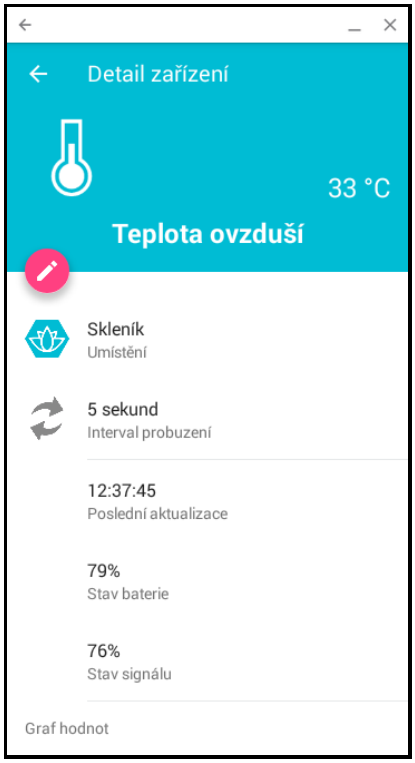
vydáním nové *beta* verze aplikace a tudíž i zvýšeným testováním testerů. V prvních týdnech roku 2015 je v grafu vidět stagnace řešení chyb a to z důvodu zkouškového období. Vývoj uživatelského rozhraní pokračoval až v 7. týdnu.

Aplha testování aplikace pro OS Windows phone probíhalo od začátku roku 2015 a zúčastnili se ho 2 uživatelé, kteří vlastnili mobilní zařízení s OS Windows phone 8.1. Šíření aplikace bylo pomocí oficiálního obchodu *Windows phone Store*. Na rozdíl od platformy Android neexistuje na platformě Windows takové množství nástaveb OS, takže po testování v emulátorech nebyly při běžném používání reálných zařízení zjištěny žádné další chyby.

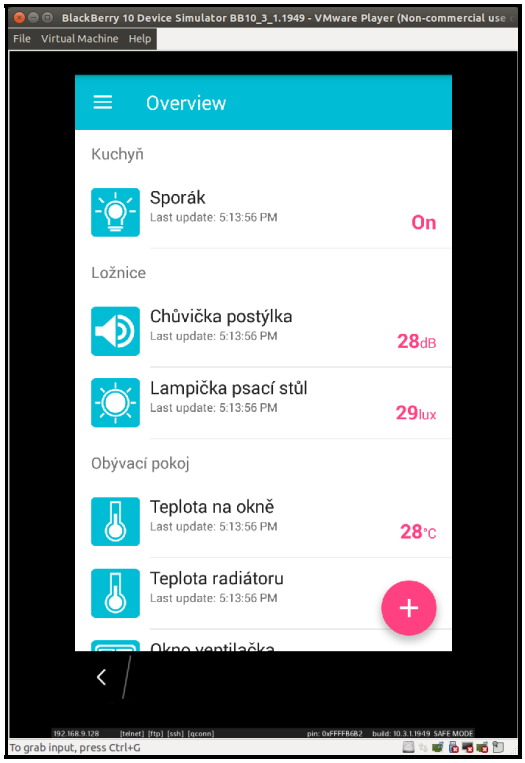
6.1 Alternativní testování

V dnešní době existují i alternativní formy použití aplikací pro platformu Android, tj. mimo OS Android. Jedním z těchto použití je spuštění aplikace pro Android pomocí prohlížeče Google Chrome. Způsob použití aplikace je popsán ve vývojářské dokumentaci [2] a je použit doplněk pro prohlížeč *ARC Welder*. Tento způsob spuštění aplikace spočívá ve vložení APK souboru do doplňku *ARC Welder* a následného spuštění zástupce aplikace v záložce *aplikace* v prohlížeči Google Chrome. Ukázku takto spuštěné aplikace můžeme vidět na obrázku 6.2a. Vzhledem k uživatelskému rozhraní fungovalo GUI naprosto totožně jako v OS Android. Testování pomocí doplňku *ARC Welder* jsem provedl jak na OS Ubuntu 14, tak na OS Windows 8.1. V obou případech testování proběhlo v prohlížeči Google Chrome ver. 42.

Dalším alternativním použitím aplikace pro platformu Android je možnost nainstalování aplikace na OS BlackBerry od verze 10. Celý postup, jak zprovoznit aplikaci pro platformu Android na OS BlackBerry, je popsán na stránkách [3] pro vývojáře. Na uvedených stránkách je popis jak spouštět aplikace přímo z IDE AndroidStudio a možnost stáhnout si emulátor OS BlackBerry. Uživatelské rozhraní jsem úspěšně odzkoušel na emulátoru verze 10.3 OS BlackBerry. Z pohledu uživatelského rozhraní opět vše fungovalo totožně jako na standardním OS Android. Uvedená verze OS BlackBerry se v IDE AndroidStudio tváří jako OS Android API 18. Ukázku spuštěné aplikace na OS BlackBerry verze 10.3 pomocí nástroje VM Player se nachází na obrázku 6.2b.



(a) Ukázka aplikace spuštěné pomocí prohlížeče Google Chrome.



(b) Ukázka aplikace spuštěné na OS BlackBerry 10.3.

Obrázek 6.2: Ukázka testování na alternativních systémech.

6.2 Dotazník

Prostřednictvím dotazníku bylo u uživatelů analyzováno, zda je navržené GUI uživatelsky přívětivé a stabilní. Pro vytvoření dotazníků a ukládání výsledků jsem použil *Formuláře Google*. Tento nástroj mi umožnil velmi jednoduše vytvořit dotazníky pro obě vytvořené platformy a následně ukládat odpovědi respondentů. Dotazník na uživatelské rozhraní pro platformu Android, jehož zprůměrované výsledky jsou uvedeny v tabulce 6.2, vyplnilo 21 respondentů.

Tématická sekce	**Nejlepší hod.**	**Nejhorší hod.**	**Průměrné hod.**
Nápověda	1	3	1,52
Vzhled	1	4	1,63
Funkčnost	1	3	1,68

Tabulka 6.1: Agregovaný výsledek dle tématických okruhů dotazníku pro aplikaci na platformě Android.

#	Otázky	Nejlepší hod.	Nejhorší hod.	Průměrné hod.
1	Přišla Vám úvodní nápověda dostatečně vysvětlující princip inteligentní domácnosti?	1	2	1,6
2	Jak hodnotíte nápovědu pro ovládání aplikace ?	1	3	1,5
3	Tutoriál v akcích přidání brány nebo zařízení byl dostatečně vysvětlující?	1	3	1,45
4	Jaký dojem na Vás aplikace udělala ?	1	2	1,45
5	Jak hodnotíte barevné schéma aplikace?	1	3	1,65
6	Pomáhaly Vám ikonky v rychlejší orientaci v aplikaci?	1	4	1,7
7	Máte pocit, že je GUI přehledné?	1	3	1,7
8	Je vám z GUI jasné, co prezentuje?	1	3	1,65
9	Jak hodnotíte přehled všech senzorů?	1	3	1,5
10	Přijde Vám vysouvací menu přehledné?	1	3	1,5
11	Jak hodnotíte detail senzoru?	1	3	1,95
12	Jsou podle Vás důležité informace dostatečně výrazné?	1	3	1,6
13	Je pro Vás přepínání aktorů dostatečně intuitivní?	1	3	1,8
14	Vyhovuje vám způsob aktualizace?	1	5	1,75

Tabulka 6.2: Výsledek dotazníku pro aplikaci na platformě Android.

Dotazník obsahoval většinou uzavřené otázky, které jsou uvedeny v tabulce 6.2, a pár otevřených otázek, které jsou uvedeny v tabulce 6.5. Uzavřené otázky byly koncipovány jako otázky testové s hodnocením 1 až 5 s tím, že hodnocení 1 znamenalo nejlepší a naopak 5 to nejhorší. Otevřené otázky byly položeny respondentům jako možnost, aby se mohli vyjádřit v případě, že jim něco chybělo nebo měli jinou poznámku k uvedeným uzavřeným otázkám. Dotazník obsahoval celkově 19 otázek a byl rozdělen do tří tématických sekcí – nápověda, vzhled a funkčnost. Průměrné ohodnocení uzavřených otázek se pohybuje mezi $1,5$ a $1,95$, což lze považovat za dobré hodnocení s tím, že se respondentům líbí vytvořené uživatelské rozhraní. V tabulce 6.1 jsou uvedeny agregované výsledky hodnocení dle tématických sekcí. Z uvedené tabulky lze usoudit, že nápověda je hodnocena nejlépe, nicméně oproti ostatním

sekcím není rozdíl příliš velký. V uzavřených otázkách se objevily poznámky a návrhy na vylepšení uživatelského rozhraní. Z poznámek k nápovědě vyplynulo, že je tlačítko *zrušit* pro někoho nadbytečné, nebo že by v obrazovce detailu senzoru/aktoru uživatelé změnili pořadí jednotlivých položek. Názory respondentů jsou zajímavé a budou zohledněny v budoucím vývoji.

Na dotazník o uživatelském rozhraní pro platformu Windows phone odpovědělo 9 respondentů a výsledky průměrného hodnocení jsou uvedeny v tabulce 6.4.

Tématická sekce	**Nejlepší hod.**	**Nejhorší hod.**	**Průměrné hod.**
Vzhled	1	4	1,82
Funkčnost	1	3	1,47

Tabulka 6.3: Agregovaný výsledek dle tématických okruhů dotazníku pro aplikaci na platformě Windows phone.

#	**Otázky**	**Nejlepší hod.**	**Nejhorší hod.**	**Průměrné hod.**
1	Jaký dojem na Vás aplikace udělala?	1	3	1,87
2	Jak hodnotíte barevné schéma aplikace?	1	3	2
3	Pomáhaly Vám ikonky v rychlejší orientaci v aplikaci?	1	2	1,5
4	Máte pocit, že je GUI přehledné?	1	4	2,1
5	Je vám z GUI jasné co prezentuje?	1	3	1,63
6	Jak hodnotíte přehled všech senzorů?	1	4	1,75
7	Přijde Vám vysouvací menu (appbar) přehledné?	1	3	1,37
8	Jak hodnotíte detail senzoru?	1	3	1,5
9	Jsou podle Vás důležité informace dostatečně výrazné?	1	3	1,5
10	Vyhovuje vám způsob aktualizace?	1	2	1,25

Tabulka 6.4: Výsledek dotazníku pro aplikaci na platformě Windows phone.

Dotazník obsahoval stejné otázky jako v případě dotazníku pro platformu Android, pouze byly vynechány ty otázky, které neměly smysl z hlediska uživatelského rozhraní pro platformu Windows phone. Průměrné hodnocení uzavřených otázek se pohybuje mezi $1,25$ a $2,1$. Na první pohled je zřejmé, že hodnocení je o něco horší než v předcházejícím pří-

padě u platformy Android. I tak je ale možné hodnocení považovat za velmi dobré. Dle agregovaných hodnot v tabulce 6.3, lze vidět že mezi hodnocením funkčnosti a vzhledu je velký rozdíl. Horší hodnocení vzhledu může být způsobeno tím, že respondenti byli většinou uživatelé mobilního zařízení s OS Android, kteří jsou zvyklí na jiný grafický styl. Pouze 2 respondenti byli majiteli mobilního zařízení s OS Windows phone.

V otevřených otázkách, které jsou k vidění v tabulce 6.5, se vyskytla například odpověď "Není mi jasný smysl větších a menších senzorů, připadá mi to matoucí". Uvedená odpověď jasně vypovídá o tom, že uživatel nemá větší zkušenosti s OS Windows phone verze 8 a 8.1, v kterých jsou v GUI většinou použité různé velikosti bloků. Protože i systém Windows phone jako samotný používá tento druh zobrazování (menších a větších bloků), tak i vytvořené uživatelské rozhraní se snaží držet zvyklostí OS. Názory respondentů v uzavřených otázkách budou opět zohledněny v budoucím vývoji.

#	**Otázky**	**Android**	**Windows phone**
1	Pokud máte nějaké připomínky k nápovědě, prosím, napište je zde.	X	–
2	Pokud máte nějaké připomínky ke vzhledu, prosím, napište je zde.	X	X
3	Doplnili byste nebo odstranili nějaké informace z přehledu senzorů ?	X	X
4	Doplnili byste nebo odstranili nějaké informace z vysouvacího menu ?	X	X
5	Chybí Vám nějaké informace v detailu senzoru ? Nebo byste nějaké informace odstranili ?	X	X

Tabulka 6.5: Otevřené otázky v dotazníku pro aplikaci na platformě Windows phone a Android.
Pozn. X znamená, že tato otázka byla použita v dotazníku pro danou platformu.

Kapitola 7

Závěr

Cílem práce bylo navrhnout a implementovat uživatelské rozhraní pro platformu Android, které by ovládalo inteligentní domácnost vyvíjenou na FIT VUT v Brně. Tento cíl jsem splnil. Nad rámec zadání jsem rozšířil práci i pro platformu Windows phone.

Nejdříve jsem nastudoval existující řešení inteligentních domácností. Nastudované systémy inteligentních domácností jsem popsal ve druhé kapitole, kde jsem se zaměřil nejprve na celý systém a poté na jejich aplikace. Systémy jsem rozdělil dle zaměření na komplexní a jednoúčelové, a dle možnosti použití na komerční a open-source. Dále jsem sestavil seznam požadavků pro inteligentní domácnost vyvíjenou na FIT VUT v Brně. Na základě získaných informací jsem navrhl uživatelské rozhraní jak pro platformu Android, tak i pro platformu Windows phone.

V kapitole implementace jsem se věnoval popisu použitých knihoven a vývoje na dané platformě. Dále jsem v kapitole popsal strukturu obrazovek uživatelského rozhraní z pohledu implementace.

Následně jsem uživatelská rozhraní testoval a to hned několika způsoby. Testování probíhalo v úrovni *aplha* i *beta*, kdy *aplha* testování se zúčastnilo 20 vývojářů a *beta* testování probíhalo v 5 domácnostech. U platformy Android jsem využil pro testování i alternativní platformy jako jsou Google Chrome či OS BlackBerry. Testování uživatelského rozhraní pro platformu Windows phone probíhalo pouze v úrovni *aplha*. Nakonec byla analyzována uživatelská přívětivost rozhraní pomocí dotazníku a to pro každou platformu zvlášť.

Do budoucna se předpokládá, že se bude celý systém dynamicky rozšiřovat, takže i uživatelské rozhraní se bude muset přizpůsobovat a to ve smyslu nových typů senzorů, tj. nových ikon, případně i nového rozvržení detailu senzoru či aktoru. Další rozšíření Android aplikace by mohlo spočívat v přidání modulu pro Android Wear, což je operační systém pro chytré hodinky, nebo modulu pro AndroidAuto, což je rozšíření pro systémy, které by měly být nasazeny přímo v automobilech. Rozšíření aplikace pro Windows phone spočívá v implementaci pro desktop verzi.

Literatura

[1] Material design – Google design guidelines [online]. Google, November 2014, [cit. 2015-05-10].
URL
http://www.google.com/design/spec/material-design/introduction.html

[2] Getting Started with ARC [Online]. 2014, [cit. 2015-05-12].
URL https://developer.chrome.com/apps/getstarted_arc

[3] Runtime for Android apps [Online]. 2014, [cit. 2015-05-12].
URL https://developer.blackberry.com/android/tools/

[4] Android Studio Overview – Android Developers [online]. 2015, [cit. 2015-05-10].
URL http://developer.android.com/tools/studio/index.html

[5] Building universal Windows apps for all Windows devices [online]. Microsoft, 2015, [cit. 2015-05-10].
URL
https://dev.windows.com/en-us/develop/building-universal-windows-apps

[6] Freedomotic – Open IoT Framework [online]. 2015, [cit. 2015-05-10].
URL http://freedomotic.com/

[7] Home automation [online]. March 2015, [cit. 2015-05-10].
URL http://en.wikipedia.org/wiki/Home_automation

[8] Insteon [online]. 2015, [cit. 2015-05-10].
URL http://www.insteon.com

[9] Inteligentní elektroinstalace – iNELS.cz [online]. 2015, [cit. 2015-05-10].
URL http://www.inels.cz/

[10] Malá domácí regulace topení Evohome [online]. 2015, [cit. 2015-05-10].
URL http:
//www.alkom.cz/sluzby/technologie/mala-automatizace-mereni-a-regulace/mala-domaci-regulace-topeni-evohome/

[11] Meet Hue [online]. 2015, [cit. 2015-05-10].
URL http://www2.meethue.com/en-xx/

[12] Netatmo weather station [online]. 2015, [cit. 2015-05-10].
URL https://www.netatmo.com/en-US/product/weather-station

[13] OpenDomo Services – control systems and advanced energy management [Online]. 2015, [cit. 2015-05-10].
URL http://www.opendomo.com/

[14] OpenRemote – Home of the Digital Home [online]. 2015, [cit. 2015-05-10].
URL http://www.openremote.org/display/HOME/OpenRemote

[15] Put the internet to work for you. [online]. IFTTT Inc., 2015, [cit. 2015-05-10].
URL https://ifttt.com/

[16] Redmine – project management web application [online]. 2015, [cit. 2015-05-10].
URL http://www.redmine.org/

[17] SmartThings [online]. 2015, [cit. 2015-05-10].
URL http://www.smartthings.com/

[18] Support Library – Android Developers [online]. Google Inc., April 2015, [cit. 2015-05-10].
URL http://developer.android.com/tools/support-library/index.html

[19] syntevo: Git client, Hg client, CVS client, File/Directory compare [online]. 2015, [cit. 2015-05-10].
URL http://www.syntevo.com/

[20] WeMo Home automation [online]. 2015, [cit. 2015-05-10].
URL http://www.belkin.com/us/Products/home-automation/c/wemo-home-automation/

[21] WinRT XAML Toolkit [online]. 2015, [cit. 2015-05-10].
URL https://winrtxamltoolkit.codeplex.com/

[22] Brychta, T.: *Bezdrátové senzory pro inteligentní domácnost*. Bakalářské práce, VUT FIT v Brně, Brno, Czech republic, 2014.

[23] Cisco: Connections Counter: The Internet of Everything in Motion [online]. http://newsroom.cisco.com, July 2013, [cit. 2015-05-10].
URL http://newsroom.cisco.com/feature-content?type=webcontent&articleId=1208342

[24] Curran, A.: ShowcaseView [online]. April 2015, [cit. 2015-05-10].
URL https://github.com/amlcurran/ShowcaseView

[25] Dunion, T.; Fried, A.: 2015 International CES to Host Largest Ever Internet of Things Showcase [online]. http://www.cesweb.org, December 2014, [cit. 2015-05-10].
URL http://www.cesweb.org/News/Press-Releases/CES-Press-Release.aspx?NodeID=bc2c6e17-3991-4897-88ab-114f115dc8b9

[26] Lee, B.: CircleProgress [online]. March 2015, [cit. 2015-05-10].
URL https://github.com/lzyzsd/CircleProgress

[27] Melnykov, O.: FloatingActionButton [online]. April 2015, [cit. 2015-05-10].
URL https://github.com/makovkastar/FloatingActionButton/

[28] Ping, Z.: *Smart House: Home Automation and Housing for the Future*. Diplomová práce, School of Architecture, Carleton University, Ottawa,Ontario, Canada, 2003.

[29] Sjölander, E.: StickyListHeaders [online]. April 2015, [cit. 2015-05-10].
URL https://github.com/emilsjolander/StickyListHeaders

[30] Therkildsen, S. V.: Android number picker [online]. June 2013, [cit. 2015-05-10].
URL https://github.com/SimonVT/android-numberpicker

[31] Vavra, D.: StyledDialogs for Android [online]. March 2015, [cit. 2015-05-10].
URL https://github.com/avast/android-styled-dialogs

[32] Wharton, J.: Android ViewPagerIndicator [online]. September 2012, [cit. 2015-05-10].
URL https://github.com/JakeWharton/ViewPagerIndicator

Příloha A

Obsah CD

- **Source-Android** – obsahuje zdrojové kódy vytvořeného uživatelského rozhraní pro platformu Android.
- **APK-Android** – obsahuje spustitelnou aplikaci pro platformu Android.
- **Source-Windows** – obsahuje zdrojvé kódy vytvořeného uživatelského rozhraní pro platformu Windows phone.
- **APPX-Windows** – obsahuje spustitelnou aplikaci pro platformu Windows phone 8.1.
- **Graphics** – obsahuje vytvořené ikony, použité v uživatelských rozhraních, i grafiku z obrazovek nápovědy.
- **ReadMe.txt** – stručný návod jak nainstalovat aplikace.
- **BeeeOn-video.mp4** – Video ukázka funkčnosti vytvořených uživatelských rozhraní.

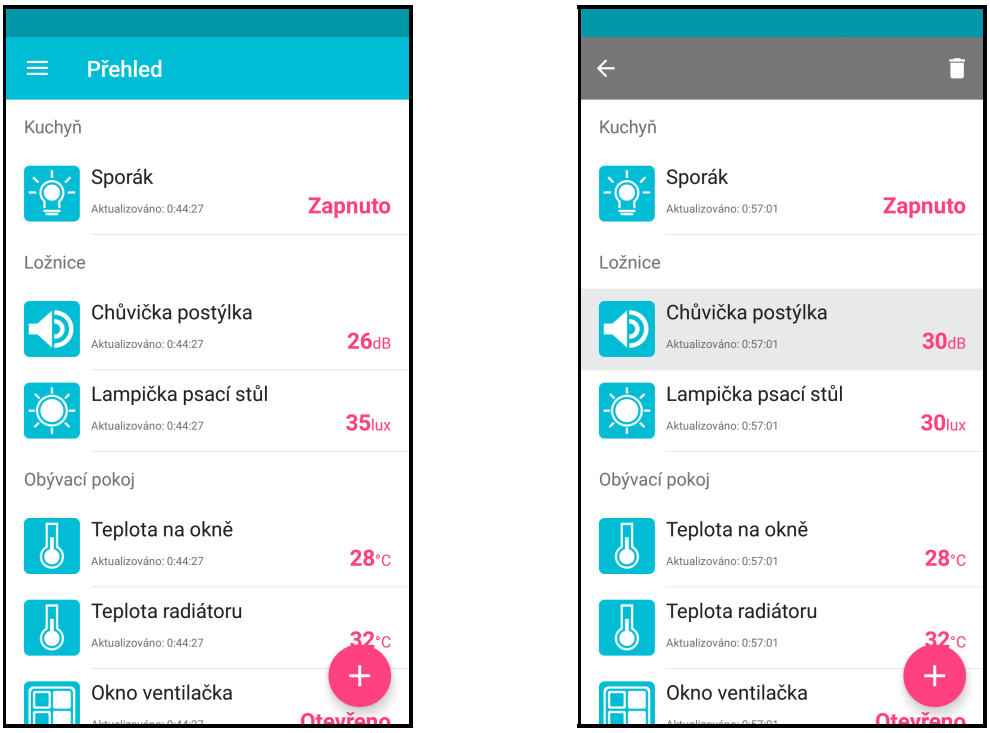
Příloha B

Ikonky pro inteligentní domácnost

Příloha C

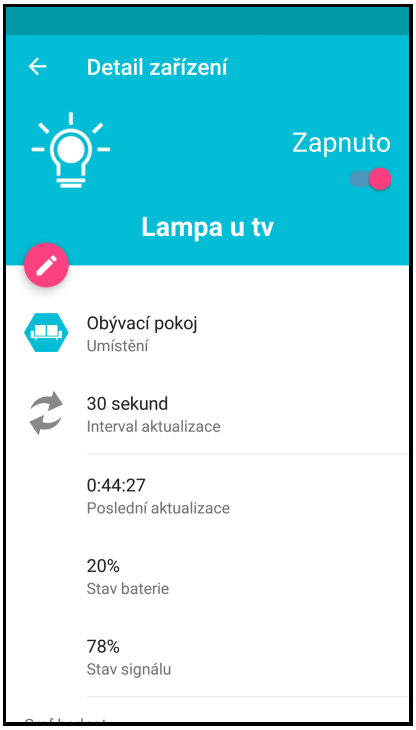
Obrazovky uživatelského rozhraní – Android

V této příloze jsou vloženy všechny obrazovky z aplikace pro platformu Android.

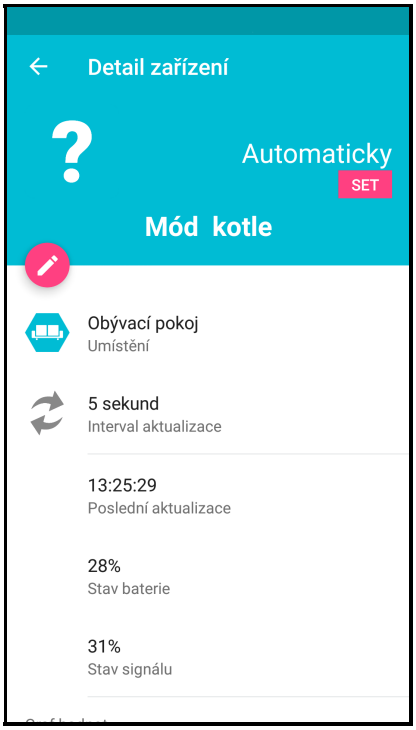


(a) Seznam zařízení. (b) Vybraný jeden sensor.

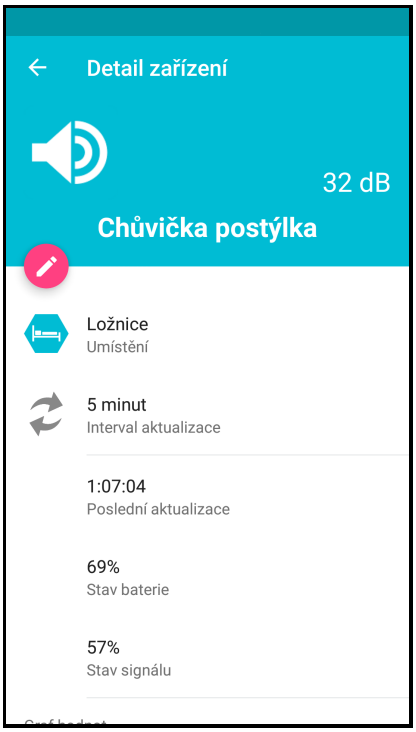
Obrázek C.1



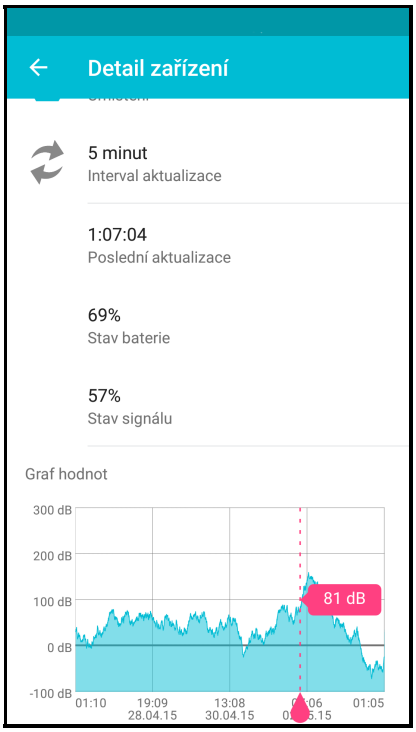
(a) Detail aktoru - typu zapnout/vypnout.



(b) Detail aktoru - typu více stavů.

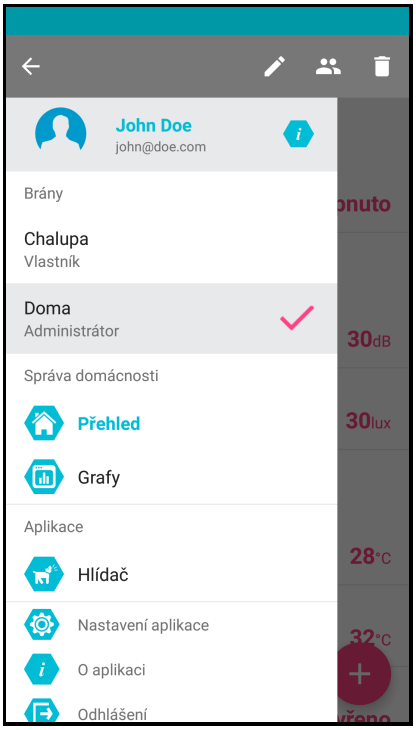


(c) Detail senzoru.

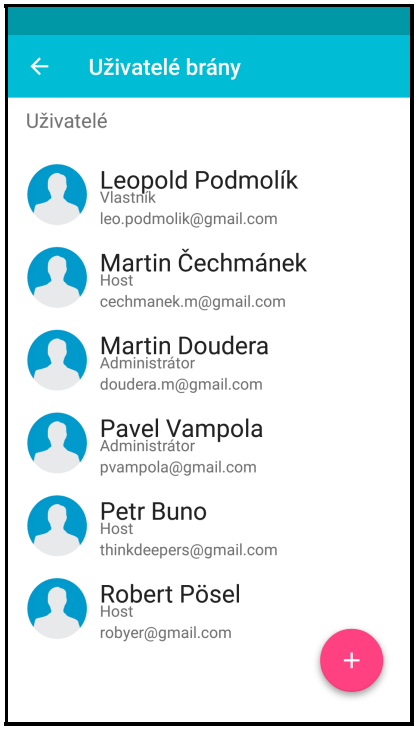


(d) Detail senzoru spolu s grafem.

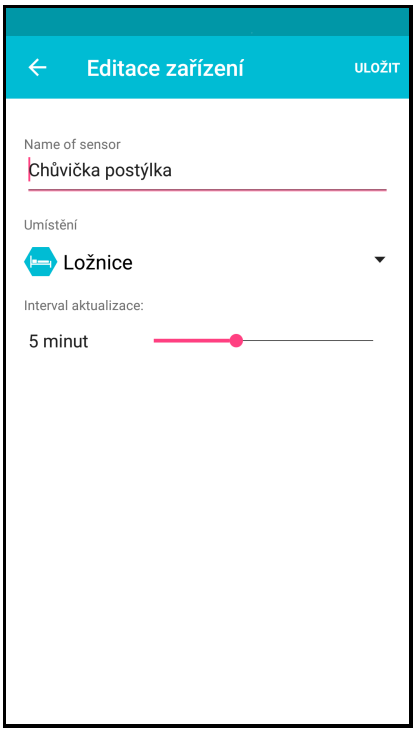
Obrázek C.2



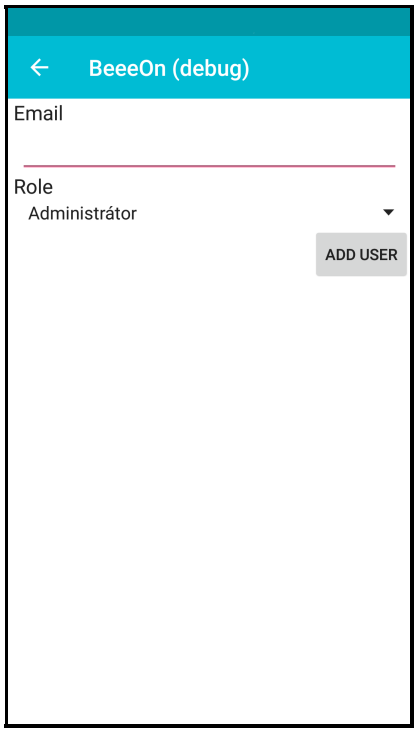
(a) Menu - zobrazený *action mode*.



(b) Správa uživatelů - seznam uživatelů.

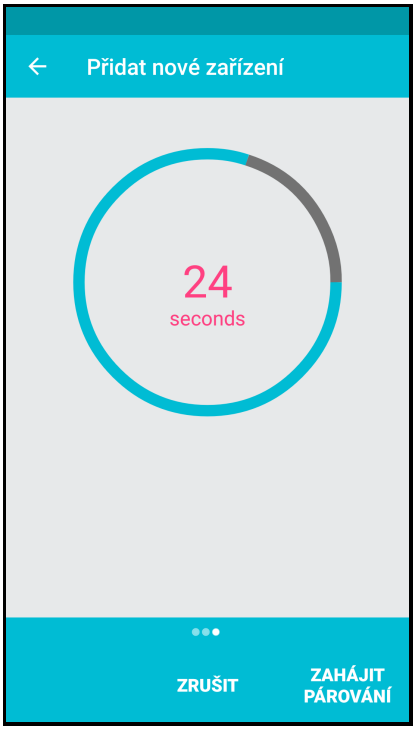


(c) Editační obrazovka senzoru/aktoru.

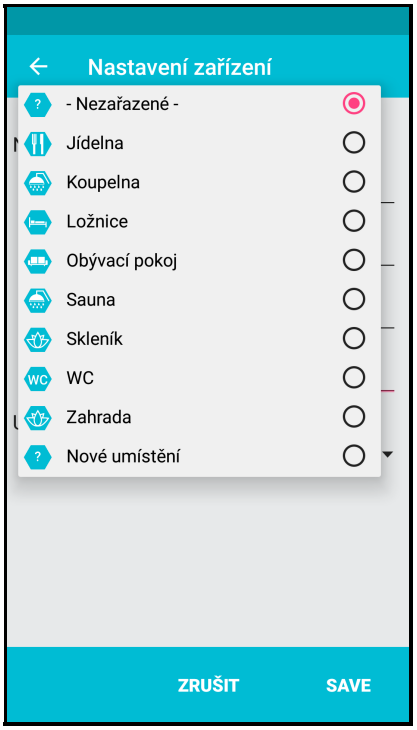


(d) Obrazovka přidání nového uživatele.

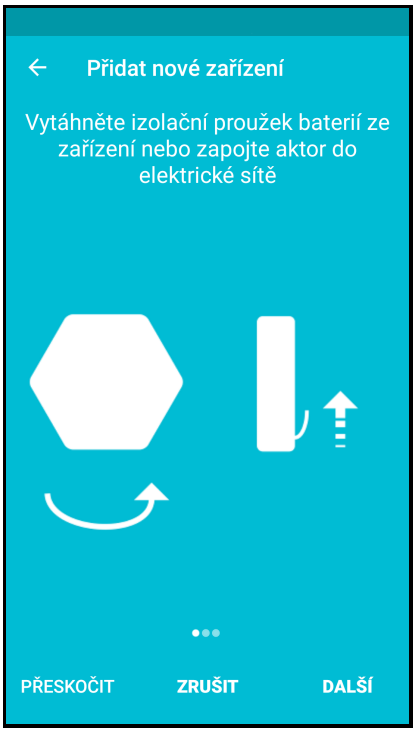
Obrázek C.3



(a) Přidávání zařízení.



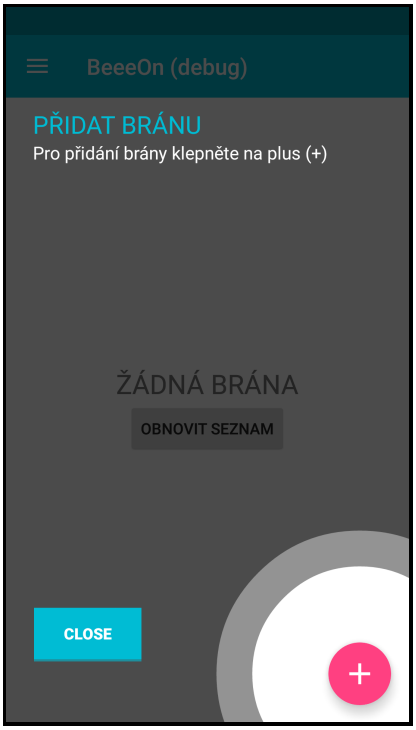
(b) Inicializace zařízení - výběr místnosti.



(c) Nápověda pro přidání zařízení - krok 1.

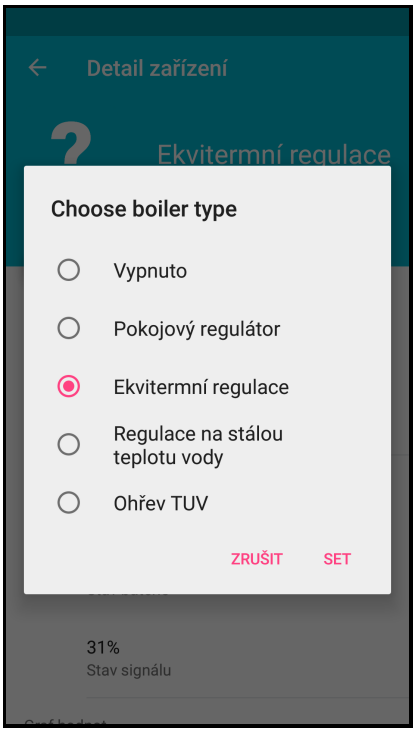
(d) Nápověda pro přidání zařízení - krok 2.

Obrázek C.4

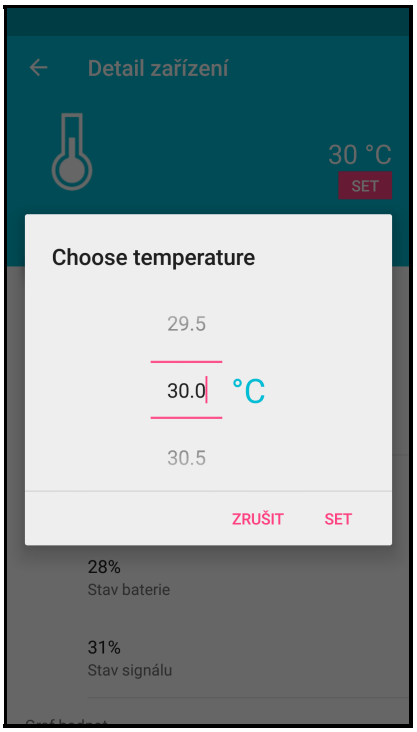


(a) Nápověda pro použití tlačítka "+".

(b) Nápověda pro použití tlačítka "+".

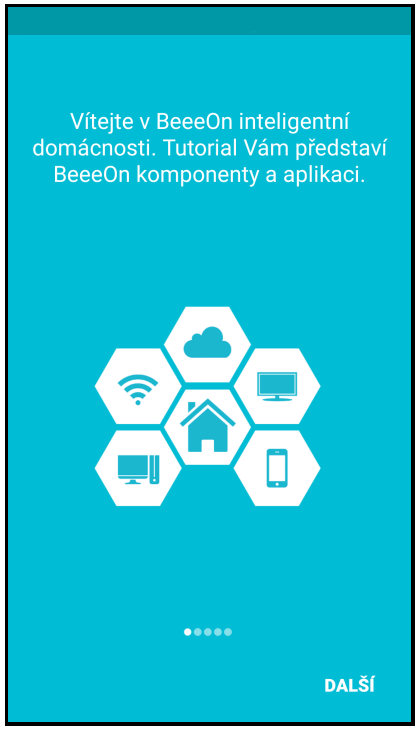


(c) Přepnutí stavu aktoru - seznam.



(d) Nastavení teploty aktoru.

Obrázek C.5



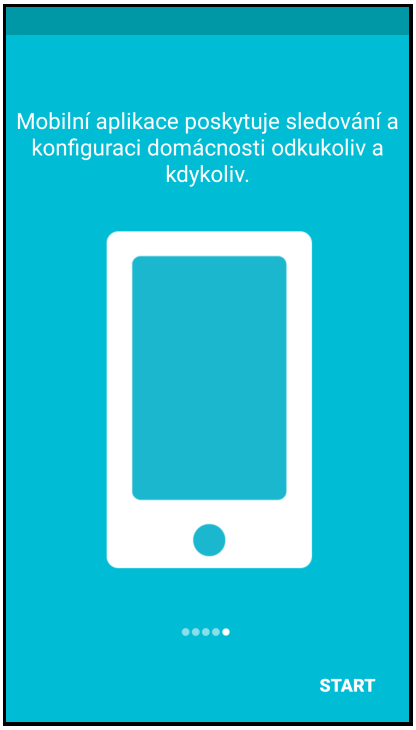
(a) První obrazovka úvodní nápovědy.

(b) Druhá obrazovka úvodní nápovědy.

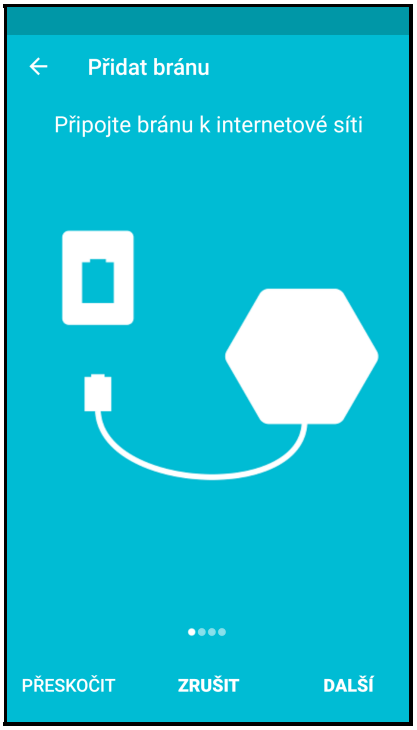
(c) Třetí obrazovka úvodní nápovědy.

(d) Čtvrtá obrazovka úvodní nápovědy.

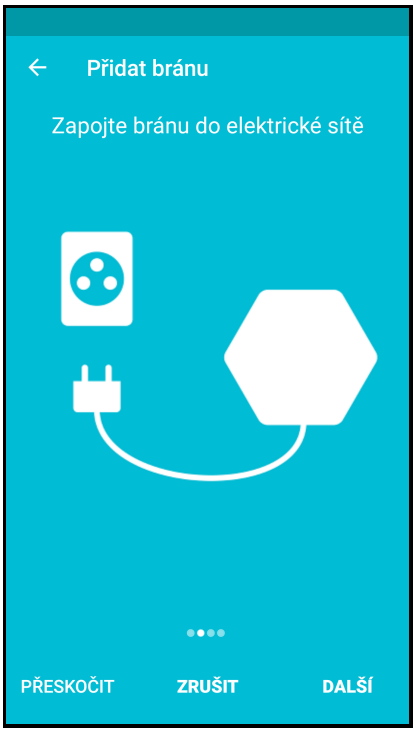
Obrázek C.6



(a) Pátá obrazovka úvodní nápovědy.



(b) První obrazovka nápovědy pro přidání brány.



(c) Druhá obrazovka nápovědy pro přidání brány.

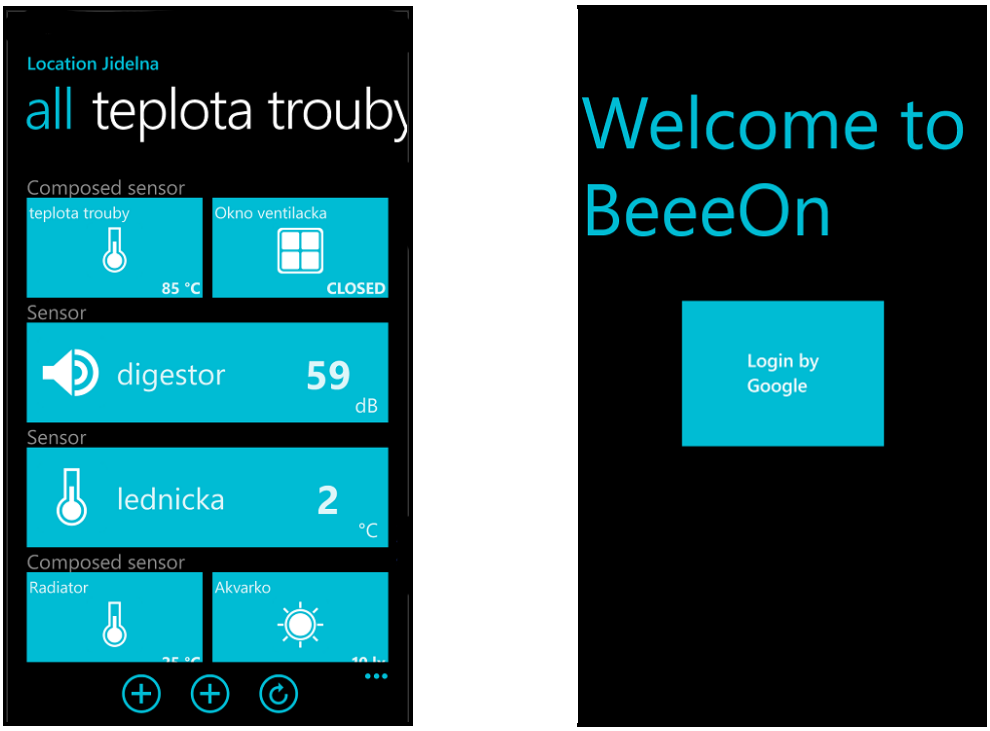
(d) Třetí obrazovka nápovědy pro přidání brány.

Obrázek C.7

Příloha D

Obrazovky uživatelského rozhraní – Windows phone

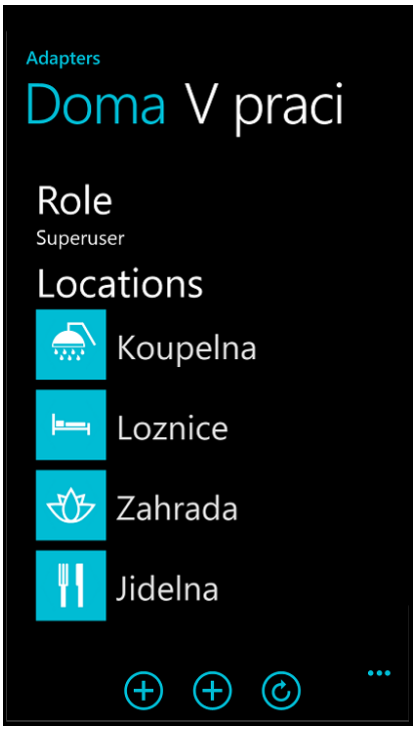
V této příloze jsou vloženy všechny obrazovky z aplikace pro platformu Windows phone.



(a) Seznam zařízení.

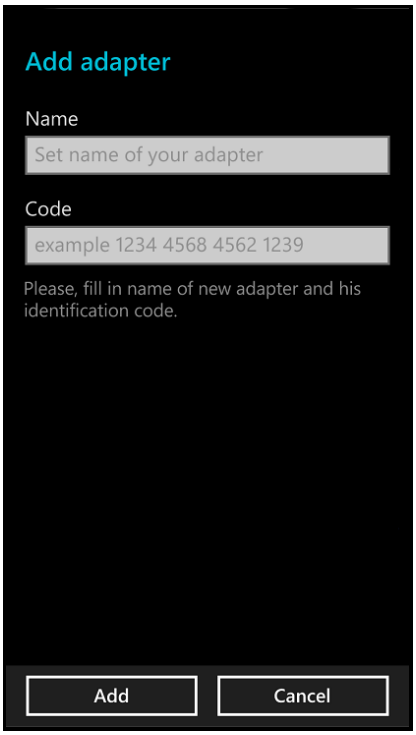
(b) Přihlašovací obrazovka.

Obrázek D.1

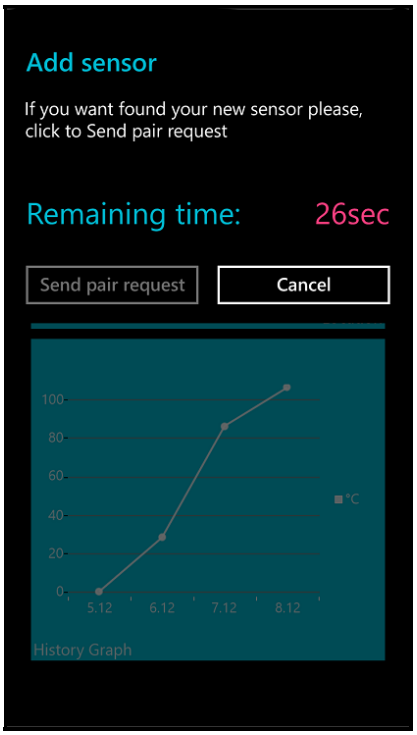


(a) Seznam bran.

(b) Odstranění brány.

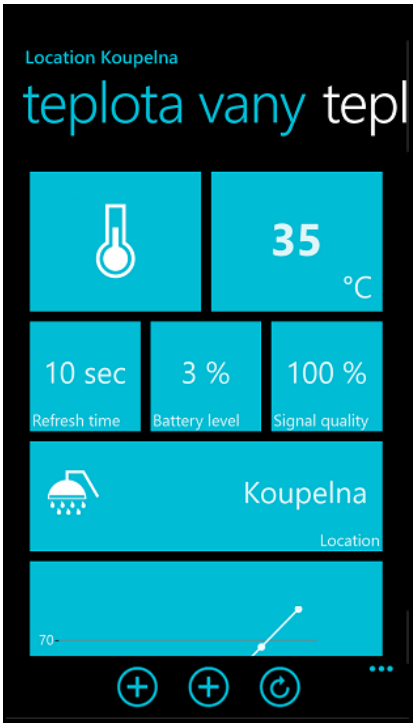


(c) Přidání brány.

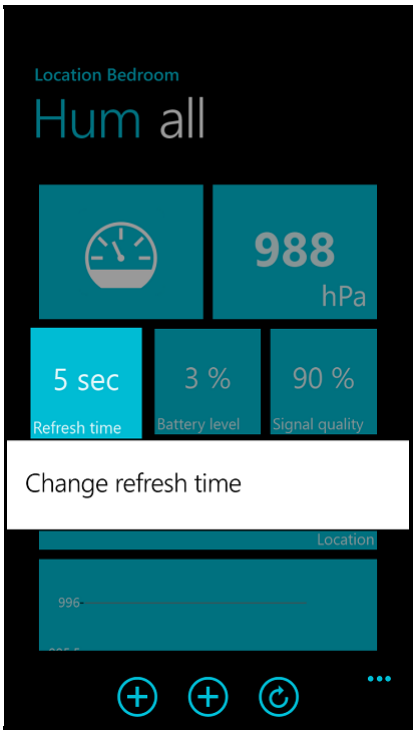


(d) Přidání zařízení.

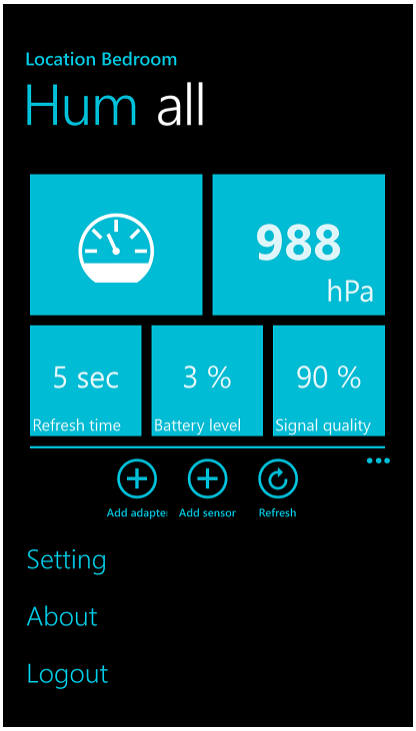
Obrázek D.2



(a) Detail senzoru.



(b) Změna aktualizačního intervalu.



(c) Vysouvací menu.

Obrázek D.3