



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

GRAFICKÉ ROZHRAŇÍ PRO OVLÁDÁNÍ INTELIGENTNÍ DOMÁCNOSTI

GRAPHICAL USER INTERFACE FOR SMARTHOME CONTROL

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN MATEJČÍK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MARTIN ŽÁDNÍK, Ph.D.

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů

Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Matejčík Martin**

Obor: Informační technologie

Téma: **Grafické rozhraní pro ovládání inteligentní domácnosti**
Graphical User Interface for Smarthome Control

Kategorie: Uživatelská rozhraní

Pokyny:

1. Seznamte se s řešením pro inteligentní domácnost, které je vyvíjeno v rámci projektu IOT na FIT VUT v Brně.
2. Na základě různých případů použití sestavte požadavky pro ovládání domácnosti inteligentním telefonem.
3. Navrhněte aplikaci, která bude splňovat Vámi definované požadavky. Při návrhu dbejte důraz zejména na jednoduchost ovládání.
4. Implementujte navržené řešení pro systém Android. Zaměřte se na přehlednou a intuitivní grafickou stránku.
5. Ověřte funkčnost implementace. Na vhodném vzorku uživatelů vyhodnoťte statistiky používání aplikace pomocí nástroje Google Analytics.
6. Zhodnoťte dosažené výsledky a diskutujte možná rozšíření práce.

Literatura:

- Dle pokynů vedoucího.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Žádník Martin, Ing., Ph.D.,** UPSY FIT VUT

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
602 00 Brno, Božetěchova 2



doc. Ing. Zdeněk Kotásek, CSc.
vedoucí ústavu

Abstrakt

Táto práca sa zaoberá vývojom grafického používateľského rozhrania na platformu Android pre systém inteligentnej domácnosti vyvíjanou na FIT VUT v Brne. V práci je predstavený systém BeeeOn, ďalej vysvetlený princíp vytvárania grafických rozhraní na Android. Praktická časť sa zaoberá návrhom a implementáciou obrazoviek rozširujúcich už existujúcu aplikáciu. Na konci práce sú vyhodnotené dosiahnuté výsledky pomocou služby Google Analytics.

Abstract

This thesis deals with development graphical user interface on Android platform for intelligent control system developed at FIT BUT. The thesis introduces a system BeeeOn, further explanation of principles of creating graphical interfaces for Android. The practical part discusses design and implementation of screens which extends an existing application. The concluding section evaluates achieved results using Google Analytics.

Kľúčové slová

chytrá, inteligentná, domácnosť, Android, grafické rozhranie, Google Analytics

Keywords

smart, intelligent, home, Android, graphics interface, Google Analytics

Citácia

MATEJČÍK, Martin. *Grafické rozhraní pro ovládání inteligentní domácnosti*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Žádník Martin.

Grafické rozhraní pro ovládání inteligentní domácnosti

Prehlásenie

Prehlasujem, že som túto prácu vypracoval samostatne pod vedením pána Ing. Martina Žádníka, PhD.

.....

Martin Matejčík

18. mája 2016

Podakovanie

Týmto by som chcel poďakovať svojemu vedúcemu práce, pánovi Ing. Martinovi Žádníkovi, PhD., za odborné vedenie práce. Taktiež ďakujem svojej rodine a priateľom za morálnu podporu a trpezlivosť.

© Martin Matejčík, 2016.

Táto práca vznikla ako školské dielo na FIT VUT v Brně. Práca je chránená autorským zákonom a jej využitie bez poskytnutia oprávnenia autorom je nezákonné, s výnimkou zákonne definovaných prípadov.

Obsah

1 Úvod	3
2 BeeeOn inteligentná domácnosť	4
2.1 Architektúra systému	4
2.2 Senzory a aktuátory	5
2.3 Brána	5
2.4 Server	5
2.5 Koncové ovládacie zariadenia	6
2.5.1 Android Aplikácia	6
3 Android	8
3.1 Architektúra	8
3.2 Verzie systému	9
3.3 Tvorba grafického rozhrania v Androide	10
3.3.1 Typy rozložení (layoutov)	10
3.4 Google material design	11
3.4.1 Popis komponentov material dizajnu	11
3.4.2 Základná definícia farebnej schémy	12
3.4.3 Typografia	13
4 Návrh pohľadov pre inteligentnú domácnosť	14
4.1 Prehľad podobných aplikácií	14
4.2 Návrh grafického rozhrania v BeeeOn aplikácii	16
4.2.1 Návrh detailu zariadenia	16
4.2.2 Návrh grafového detailu senzoru	17
4.2.3 Návrh prehľadu	18
4.2.4 Návrh pridávania položiek do prehľadu	19
4.2.5 Návrh detailu položky prehľadu	21
5 Implementácia	22
5.1 Použité knižnice tretích strán	22
5.2 Implementácia obrazoviek	22
5.2.1 Implementácia prehľadu	24
6 Vyhodnotenie	25
6.1 Testovanie	25
6.2 Štatistiky Google Analytics	25
6.2.1 Počet zobrazení implementovaných obrazoviek	25

6.2.2	Počet zobrazení rozsahov grafu v grafovom detaile	26
6.2.3	Počet pridání položiek do prehľadu	26
6.2.4	Počet kliknutí na detail položky v prehľade	27
7	Záver	28
	Literatura	29
	Přílohy	30
	Seznam příloh	31
A	Obrazovky aplikácie	32

Kapitola 1

Úvod

V dnešnej dobe, keď už je do veľkej miery rozšírený internet a pripája sa čoraz viac zariadení či už mobilných alebo stacionárnych, začína hýbať svetom Internet vecí (z angl. *Internet of Things*). Domácnosti tomu nie sú výnimkou. Dnes už nie je problém si kúpiť „smart“ žiarovku alebo dokonca chladničku, ktorú môžete ovládať z druhého konca sveta pomocou aplikácie.

Cielom tejto práce je rozšíriť už existujúcu aplikáciu pre telefóny s operačným systémom Android, ktorá sa práve stará o ovládanie inteligentnej domácnosti. Výsledok umožní používateľovi poskladať si pohľady na domácnosť podľa svojich potrieb čo zaručí nový prehľad domácnosti. Vylepšený bude aj detail zariadenia, ktorý poskytne používateľovi komplexný pohľad na svoje zariadenie inteligentnej domácnosti. Nakoniec budú nové grafové detaily, ktoré poskytnú rôzne časové intervaly zobrazenia s voľbou prispôsobenia zobrazovaných údajov.

Táto práca popisuje riešenie inteligentnej domácnosti BeeeOn vyvíjanej na FIT VUT v Brne (kapitola 2). V ďalšej kapitole 3 je popísaná platforma Android, jej architektúra a hlavne tvorba grafického rozhrania. V kapitole 4 popisuje existujúce riešenia a návrh jednotlivých obrazoviek tak, aby spĺňali požiadavky používateľov a systému BeeeOn. Ďalšia kapitola 5 sa zaoberá detailami implementácie grafického rozhrania. Posledná kapitola 6 popisuje testovanie a vyhodnocuje dosiahnuté výsledky zozbierané pomocou služby Google Analytics.

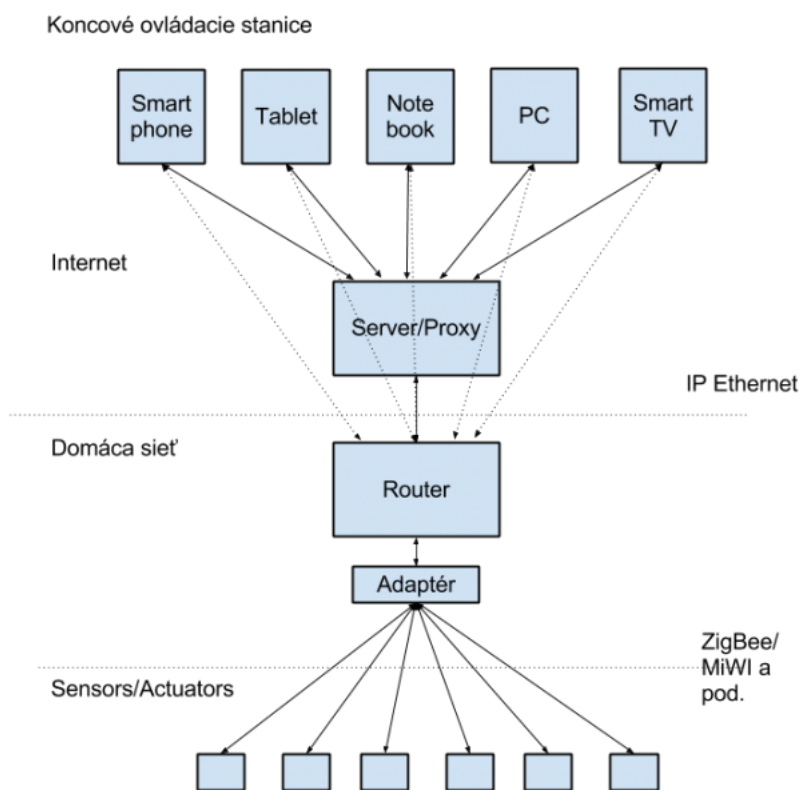
Kapitola 2

BeeeOn inteligentná domácnosť

BeeeOn je open source projekt, ktorý je vyvíjaný v rámci projektu Internet of Things (IoT) na FIT VUT. Jeho cieľom je vytvoriť modulárny, ľahko rozširiteľný a bezpečný systém pre riadenie domov a automatizáciu. Obsah tejto kapitoly je inšpirovaným z [1].

2.1 Architektúra systému

Celý systém sa skladá z niekoľko častí, viď obrázok 2.1, ktoré budú ďalej popísané.



Obr. 2.1: Architektúra systému BeeeOn¹

¹Prevzaté z <https://ant-2.fit.vutbr.cz/projects/iot/wiki/Architektura>

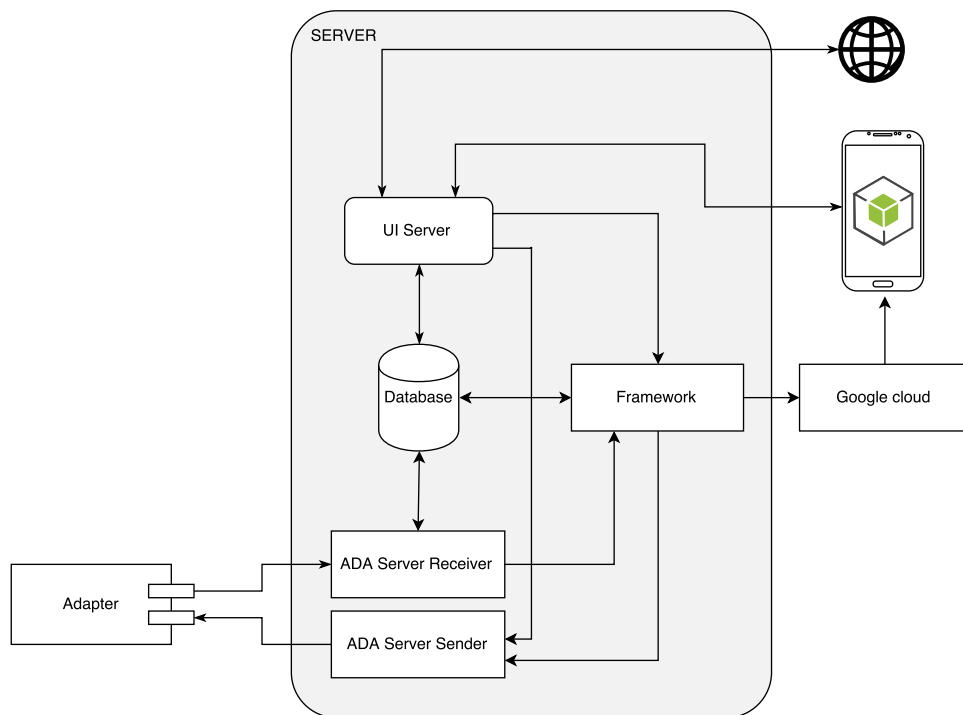
2.2 Senzory a aktuátory

Najnižšiu vrstvu tvoria zariadenia, ktoré merajú dané hodnoty z okolia alebo ovplyňujú svojim stavom nejaké zariadenie. Mali by byť navrhnuté ako veľmi jednoduché zariadenia, ktoré ideálne snímajú iba jednu veličinu. Všetky typy zariadení komunikujú v systéme s bránou (2.3) vlastnou implementáciou RF protokolu s názvom FIT protocol²

2.3 Brána

Brána je centrálnou jednotkou v inteligentnej domácnosti. Svojou konštrukciou a hlavne funkciou je veľmi podobná domácomu routeru. Zariadenie je pripojené k zdroju napájania a do internetu. Ako softwarové jadro je použitý Linux. Jej úlohou je zabezpečiť komunikáciu s napáranými senzorovými zariadeniami a serverom. Môžeme teda povedať, že je len akýmsi premostením medzi bezdrôtovým protokolom a IP protokolom.

2.4 Server



Obr. 2.2: Architektúra serveru³

Server poskytuje niekoľko služieb. Popredne sa stará o registráciu a autentizáciu používateľov a uchováva v databáze ich dáta. Medzi uchovávané dáta patrí zaregistrované brány, spárované zariadenia (senzory alebo aktuátory) a ich namerané dáta, nastavenia domácností a ich používateľov s nastavenými právami. Ďalej komunikuje s bránami a stará sa o príjem

²http://beeeon.org/index.php?title=FIT_protocol

³Prevzaté z <https://ant-2.fit.vutbr.cz/projects/server/wiki>

dát, odpovedá na požiadavky koncových ovládacích zariadení. V súčasnosti sa vyvíja na serveri aj aplikačný framework, ktorý bude zabezpečovať rozširujúce funkcie a algoritmy.

2.5 Koncové ovládacie zariadenia

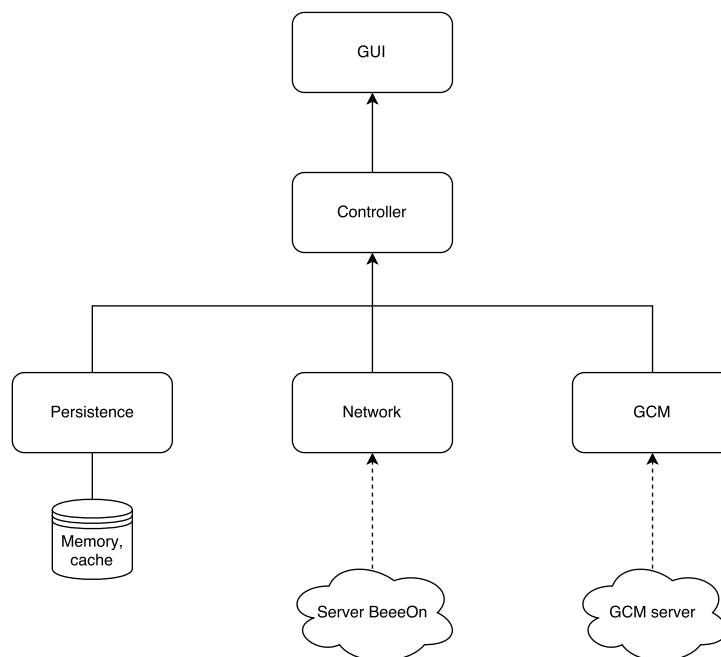
Koncové zariadenia majú za účel ovládať a monitorovať celý systém. Aktuálne celá komunikácia prebieha cez server. Súčasne sa ale vyvíja možnosť pripojenia priamo na Bránu bez nutnosti pripojenia na internet, čím sa zaručí konektivita pri nedostupnosti internetového pripojenia. To ale neplatí pre pokročilé funkcie, ktoré su dostupné iba na serveri.

Po registrácii a prihlásení má používateľ možnosť si spárovať novú bránu (2.3) (pokiaľ už nebol pridaný k existujúcej iným používateľom). zobrazovať historické dáta senzorov, prepínať stavy aktuátorov, meniť interval prebudenia senzoru a zobrazovať historické hodnoty v grafoch. Dostupné funkcie sa líšia na základe prístupových práv používateľa.

2.5.1 Android Aplikácia

Jedným z možných koncových zariadení je práve Android aplikácia BeeeOn. Na jej rozšírení je založená táto bakalárska práca.

Architektúra aplikácie je čiastočne inšpirovaná architektúrou MVC a skladá sa z niekoľkých celkov.



Obr. 2.3: Architektúra Android aplikácie

Controller

Predstavuje jadro aplikácie. Je vytvorený ako Singleton. Pre svoju funkčnosť používa ďalšie moduly Network a Persistence.

Network

Sieťová vrstva aplikácie, ktorá implementuje API pre komunikáciu so serverom využívajúca vlastný XML protokol.

Persistence

Uchováva dáta v modeloch v pamäti a ich cachovanie v zariadení.

GCM

Služba *Google Cloud Messaging*⁴, prostredníctvom ktorej je zabezpečený príjem notifikácií zo serveru.

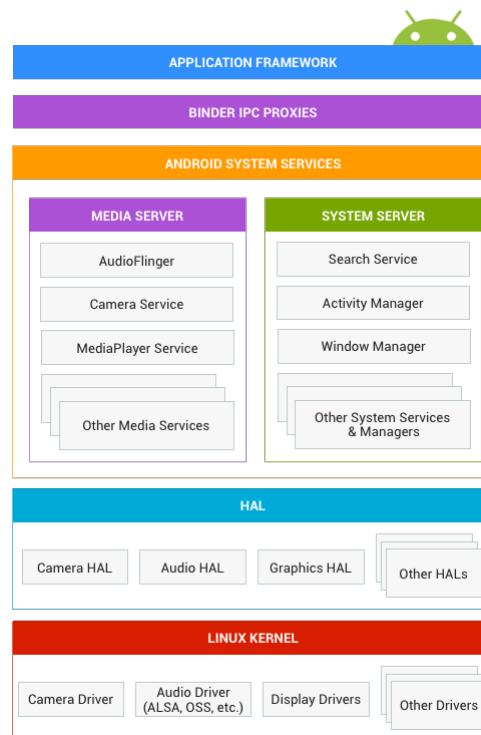
⁴<https://developers.google.com/cloud-messaging/>

Kapitola 3

Android

Android je komplexná open source platforma určená pre mobilné zariadenia. Je presadzovaná spoločnosťou Google a vlastnená zoskupením Open Handset Alliance, čo je skupina 84 technologických a mobilných spoločností, ktoré sa spojili s cieľom urýchliť inovácie v oblasti mobilných technológií a spotrebiteľom ponúknuť bohatší, lacnejší a lepší mobilný zážitok[2].

3.1 Architektúra



Obr. 3.1: Android stack¹

¹Prevzaté z <https://source.android.com/devices/>

Application framework (Aplikačný rámec) – Najvyššia vrstva, ktorá sa používa na vývoj aplikácií.

Binder IPC proxies – Medziprocesný komunikačný mechanizmus, vrstva poskytujúca aplikačnému rámcu volať systémové služby Androidu, z pohľadu aplikačného rámca je táto vrstva neviditeľná.

Android system services (Systémové služby) – Funkčnosť zabezpečujúca pre aplikačný rámec komunikáciu so systémom a základným hardverom. Služby sú modulárne, jednotlivé moduly sú napríklad správca okien, služba vyhľadávania alebo manažér notifikácií. Android zahŕňa dve skupiny služieb: *system* (systémová) a *media* (pre média – napr. nahrávacie).

HAL – Hardware abstraction layer (Abstraktná vrstva pre hardware) definuje štandardné rozhranie pre dodávateľov hardwaru na implementáciu ovládačov pre Android. Poskytuje možnosť implementovať novú funkčnosť bez zásahu do vyšších vrstiev.

Linux kernel (Linuxové jadro) – Android používa ako jadro verziu Linuxového jadra s pár špeciálnymi úpravami ako napríklad *wake locks* (zámok prebudenia systému), *Binder IPC driver* a iné vlastnosti dôležité pre vstavané mobilné systémy.

3.2 Verzie systému

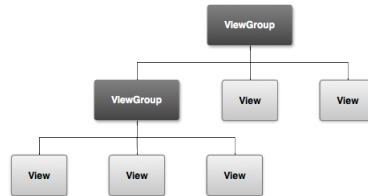
V súčasnosti sa na trhu nachádzajú nasledovné verzie operačného systému Android:

Verzia	Kódový názov	API	Distribúcia
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.2%
4.1.x	Jelly Bean	16	7.8%
4.2.x		17	10.5%
4.3		18	3.0%
4.4	Kitkat	19	33.4%
5.0	Lollipop	21	16.4%
5.1		22	19.4%
6.0	Marshmallow	23	4.6%

Tabuľka 3.1: Verzie systému Android k 04.2016²

3.3 Tvorba grafického rozhrania v Androide

Všetky prvky používateľského rozhrania sú zložené zo základných objektov **View** a **ViewGroup**. **View** je objekt, ktorý kreslí niečo s čím dokáže používateľ interagovať. **ViewGroup** je objekt, ktorý zoskupuje ostatné **View** objekty tak, aby definoval usporiadanie (layout) rozhrania [3].



Obr. 3.2: Ilustrácia hierarchie View objektov³

3.3.1 Typy rozložení (layoutov)

Frame layout

Typ rozloženia, ktoré ukladá každý grafický objekt po vrstvách v poradí akom sú vkladané. Veľmi často sa používa aj na vyhradenie miesta (placeholder) pre iné grafické objekty, ktoré sú pridané neskôr programovo v kóde [4].

Linear layout

Jeden z najjednoduchších a najviac používaných rozložení. Jednotlivé grafické objekty ukladá vedľa druhého buď horizontálne alebo vertikálne. [4]

Relative layout

V tomto prípade sa využívajú relatívne usporiadania prvkov navzájom. Je to užitočné, pretože nie je potrebné zanorovať viac rozložení do hierarchie, čo vo výsledku zvyšuje rýchlosť vykresľovania systémom [4].

Coordinator layout

Používa sa na dva špecifické prípady:

1. Ako základné rozloženie aplikácie
2. Ako kontajner pre špecifickú interakciu s objektami vo vnútri rozloženia.

Zadaním správania (**Behaviors**) pre potomka tohto rozloženia je možné definovať rôzne správanie, napríklad zachytenie k inému elementu [5].

³Prevzaté z <http://developer.android.com/guide/topics/ui/overview.html>

3.4 Google material design

Material design je komplexná príručka pre vizuálny, pohybový a interakčný návrh naprieč rôzne platformy a zariadenia. Android zahŕňa podporu pre aplikácie v Material design od verzie systému 5.0 (API level 21) natívne alebo s príslušnými podpornými knižnicami⁴ zabezpečujúce kompatibilitu až do verzie 2.2 (API level 8) [6].

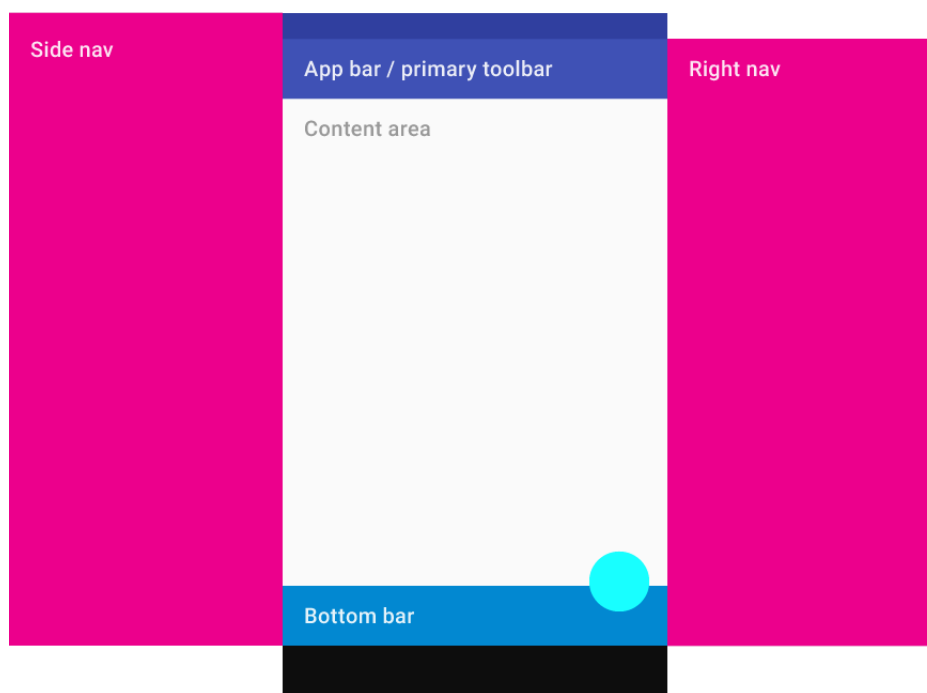
Google prvý krát ukázal Material Design na ich I/O konferencií v lete v roku 2014. Pôvodne formalizácia a rozšírenie pre grafické rozhranie ich Google Now aplikácie⁵, Material Design rástol (a stále rastie) do komplexného a systematického súboru dizajnových filozofií.

Material môže byť považovaný za niečo ako inteligentný papier. Podobne ako papier, má plochy a hrany, ktoré odrážajú svetlo a vrhajú tieň, ale narozdiel od papiera, material má vlastnosti, ktoré v reálnom papieri nie sú, ako napríklad schopnosť sa pohybovať, zmeniť jeho tvar a veľkosť, a zlúčenie sa s iným prvkom.

Z veľkej časti je založený na dlhoročných princípoch prevzatých z teórie farieb, animácie, tradičnom tlačovom dizajne a fyzike. Material dizajn, ktorý umožňuje virtuálny priestor, kde vývojári môžu použiť povrch a svetlo k vytvoreniu zmysluplného rozhrania a pohybu na navrhnutie intuitívnych použiteľských interakcií [7].

3.4.1 Popis komponentov material dizajnu

Základné Rozloženie



Obr. 3.3: Mobilná štruktúra rozloženia aplikácie⁶

⁴<https://developer.android.com/tools/support-library/index.html>

⁵<https://play.google.com/store/apps/details?id=com.google.android.googlequicksearchbox>

⁶Prevzaté z <https://www.google.com/design/spec/layout/structure.html>

- **Side nav a Right nav** – (ľavý a pravý navigačný panel) používajú sa ako navigácia najvyššej úrovne. Zobrazujú sa gestom posunutia zľava, pre ľavý panel alebo posunutia zprava, pre pravý panel. Primárne používaný (aj v BeeOn aplikácii) ľavý panel ako navigačné menu, zatiaľ čo pravý sa používa len ako doplnkové menu.
- **App bar/primary toolbar** – predtým známejší v Androide ako *Action bar*, je panel, ktorý sa používa na logá, navigáciu, vyhľadávanie a rôzne iné akcie, napríklad kontextové menu.
- **Content area** – hlavné miesto pre obsah aplikácie.
- **Bottom bar** – obsahuje väčšinou rýchlu navigáciu medzi obrazovkami aplikácie.

Ďalšie komponenty

- **Bottom sheet** – je alternatíva k menu alebo jednoduchému dialógu. Môžu tiež predstavovať hlboko spojený obsah z iných aplikácií. Predovšetkým sú určené pre mobilné telefóny.
- **Button** – jasne oznamuje, k akej akcii dôjde po dotknutí. Je zložené z textu, obrázku alebo s obidvomi, navrhnuté tak, aby bolo v súlade s grafickým štýlom aplikácie.
- **Floating Action Button** – špeciálny typ tlačidla pre akcie s vyššou dôležitosťou. Majú kruhovú ikonu vznášajúcu sa nad UI⁷, majú pohybové správanie a dokážu sa transformovať na iný grafický objekt.
- **Card** – vhodný nástroj na zobrazenie obsahu zloženého z rôznych prvkov.
- **Dialog** – špeciálny typ modálneho okna, ktorý obsahuje text, ovládacie prvky UI zamerané na konkrétnu úlohu. Jeho hlavnou úlohou je zobraziť používateľovi dôležitú informáciu vyžadujúcu rozhodnutie, alebo môže zahŕňať viac úloh.
- **List, Grid list** – najvhodnejšie na prezentáciu homogénnych dát. Predstavujú viac riadkové položky v zvislom smere (*List*), alebo zobrazenie v mriežke (*Grid*).
- **Snackbar, toast** – poskytujú spätnú väzbu na danú operáciu tým, že ukážu krátku správu v dolnej časti obrazovky. *Snackbar* navyše od *Toast* môže obsahovať akciu.
- **Tabs** – uľahčujú prepínanie medzi rôznymi pohľadmi alebo funkčnými aspektami aplikácie.
- **Text Fields** – umožňujú používateľovi zadať text, vybrať text alebo vyhľadať dáta prostredníctvom automatického dopĺňovania.

3.4.2 Základná definícia farebnej schémy

Rozlišujeme tri základné farby témy aplikácie.

Primary color

Primárna farba témy. Táto farba je najviac používaná v aplikácii.

⁷z angl. User Interface - používateľské rozhranie

Secondary color

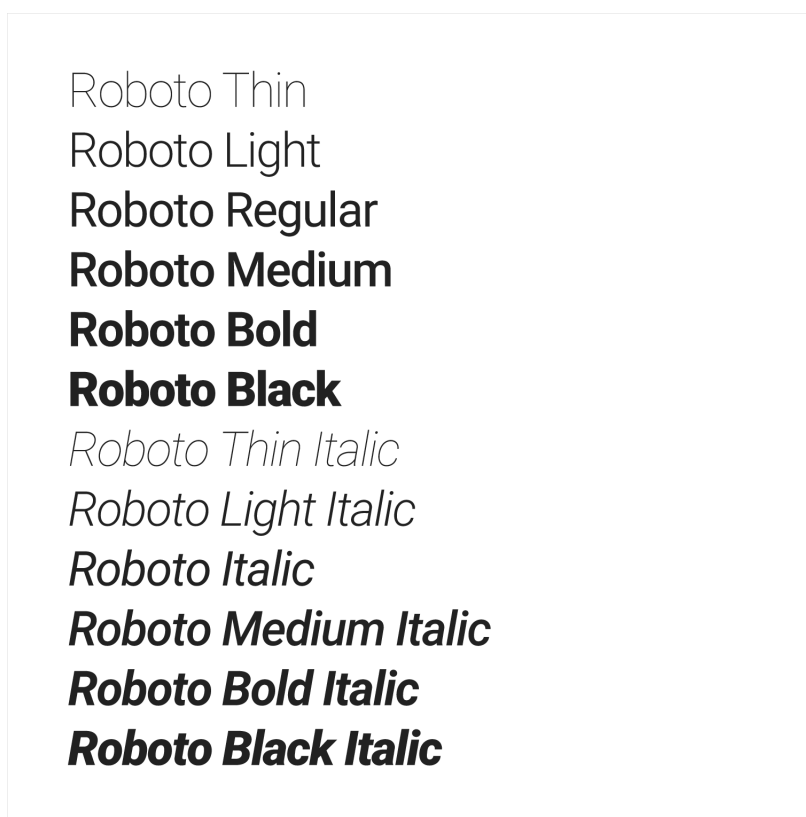
Sekundárna farba témy. Používa sa pre prvky indikujúce súvisiace akcie alebo informácie. Takisto pre podfarbenie horného systémového panelu a dolného navigačného panelu.

Accent color

Zvýrazňovacia farba témy. Mala by byť použitá pre komponenty s akciou (*Buttons, Floating action button, Text field*)

3.4.3 Typografia

Material dizajn odporúča ako hlavný font písma Roboto pre jazyky latinské a pre ostatné jazyky font písma Noto.



Obr. 3.4: Príklad Roboto fontu⁸

⁸Prevzaté z <https://www.google.com/design/spec/style/typography.html#typography-typeface>

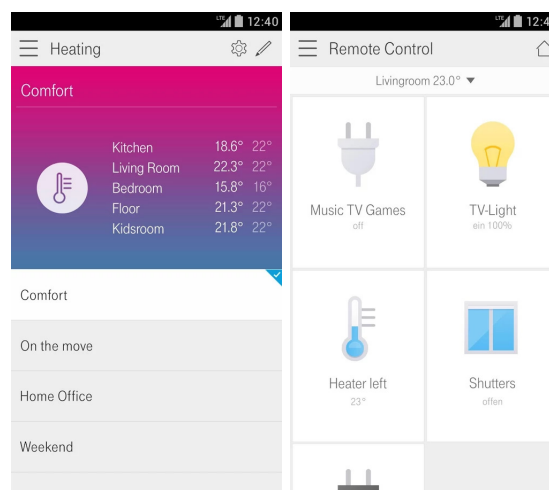
Kapitola 4

Návrh pohľadov pre inteligentnú domácnosť

4.1 Prehľad podobných aplikácií

Telekom Smart Home

Riešenie od spoločnosti Telecom ponúka komplexné ovládanie domácnosti ako napríklad kúrenie, osvetlenie alebo poplašný systém. Aplikácia obsahuje obrazovku prehľadu, kde je zoznam dostupných zariadení v domácnosti s ich aktuálnymi hodnotami. Nevýhodou je absencia zobrazenia akéhokoľvek grafu s historickými hodnotami.

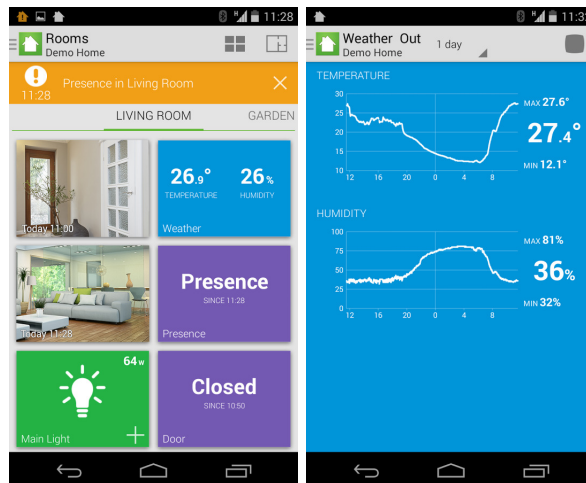


Obr. 4.1: Telekom Smart Home aplikácia¹

¹Prevzaté z <https://play.google.com/store/apps/details?id=de.telekom.smarthomeb2c>

Archos Smart Home

Podobná aplikácia od spoločnosti Archos má ako primárnu obrazovku prehľad delený na skupiny podľa umiestnenia v byte. Z prehľadu je možné zobraziť aj detail s grafmi.



Obr. 4.2: Archos Smart Home aplikácia²

Alfred smart home

Aplikácia vyniká hlavne prepracovaným grafickým dizajnom s využívaním material dizajnu. Rovnako ako u predchádzajúcich príkladoch je aj v tejto aplikácii ako hlavná obrazovka prehľad zariadení rozdelených do skupín. Taktiež je možné zobraziť detail s grafom.



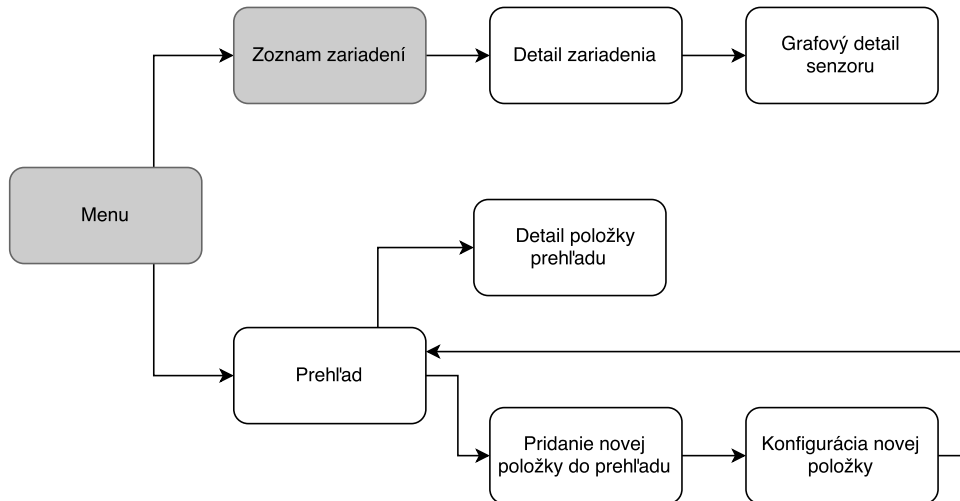
Obr. 4.3: Alfred Smart Home aplikácia³

²Prevzaté z <https://play.google.com/store/apps/details?id=com.archos.athome.center>

³Prevzaté z <https://play.google.com/store/apps/details?id=mobile.alfred.com.alfredmobile>

4.2 Návrh grafického rozhrania v BeeOn aplikácii

V tejto kapitole bude popísaný návrh grafického rozhrania obrazoviek aplikácie. Pri navrhovaní bolo treba nadviazať na dizajn už existujúcej aplikácie a architektúry systému BeeOn. Všetky navrhované obrazovky boli navrhnuté podľa grafického dizajnu popísaného v kapitole (3.4)



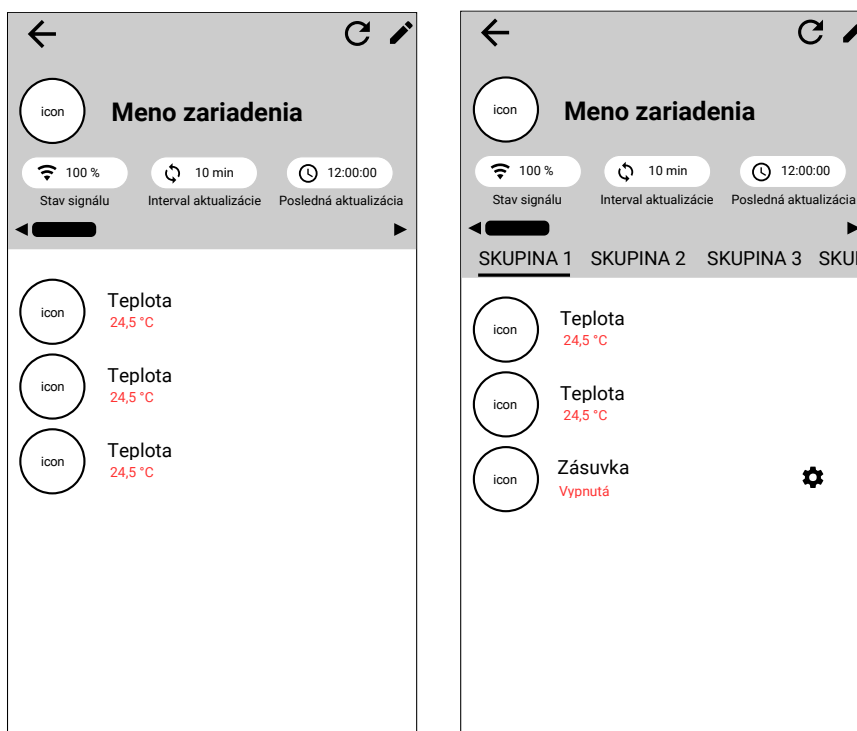
Obr. 4.4: Návrh obrazoviek

Na obrázku 4.4 je možno vidieť usporiadanie navrhovaných obrazoviek v aplikácii. Nadväzujú na už v minulosti implementované *menu* a *zoznam zariadení*, ktoré neboli predmetom tejto práce. Jednotlivé obrazovky budú ďalej popísané v ďalších sekciách.

4.2.1 Návrh detailu zariadenia

Detail zariadenia ponúka používateľovi pohľad na jedno zariadenie popísané v kapitole 2.2. Obrazovka je zložená z dvoch hlavných častí. V hornej časti je umiestnený *Toolbar* s možnosťou vrátenia späť na predchádzajúcu obrazovku, tlačidlom na obnovu údajov a tlačidlom na editáciu zariadenia. V ďalšom riadku je umiestnená ikona a názov zariadenia. Údaje o zariadení – stav signálu, stav batérie, interval aktualizácie, posledná aktualizácia a umiestnenie boli vložené do horizontálneho *ScrollView*, aby nezaberali príliš veľa miesta. Každý údaj predstavuje jednotný grafický tvar s ikonkou, textovou informáciou o aktuálnej hodnote a popisom, ktorý informuje o type. Na rozhraní hornej časti sú zobrazované skupiny, do ktorých môžu byť voliteľne delené senzory/aktuátory.

Ďalšia časť zobrazuje zoznam senzorov/aktuátorov. Každý prvok tohto zoznamu zobrazuje názov, ikonu a aktuálnu hodnotu. Pri aktuátoroch, u ktorých je možné meniť stav je navyše prítomné tlačidlo na zobrazenie dialógu pre zmenu stavu.



Obr. 4.5: Návrh detailu zariadenia

4.2.2 Návrh grafového detailu senzoru

Grafový detail bude zobraziteľný po kliknutí na senzor v detaile zariadenia. V *Toolbare* bude uvedený názov senzoru/aktuátora, tlačidlo na vrátenie späť a tlačidlo na obnovu obsahu. Aktuálna hodnota bude umiestnená v strede pod *Toolbarom*. Časový rozsah grafu bude možné zvoliť pomocou záložiek umiestnených v spodnej časti hlavičky s možnosťou zobraziť poslednú hodinu, deň, týždeň alebo mesiac. Hlavnú časť obrazovky zaberá graf podľa typu senzoru/aktuátora, čiarový pre senzory a stĺpcový pre aktuátory.

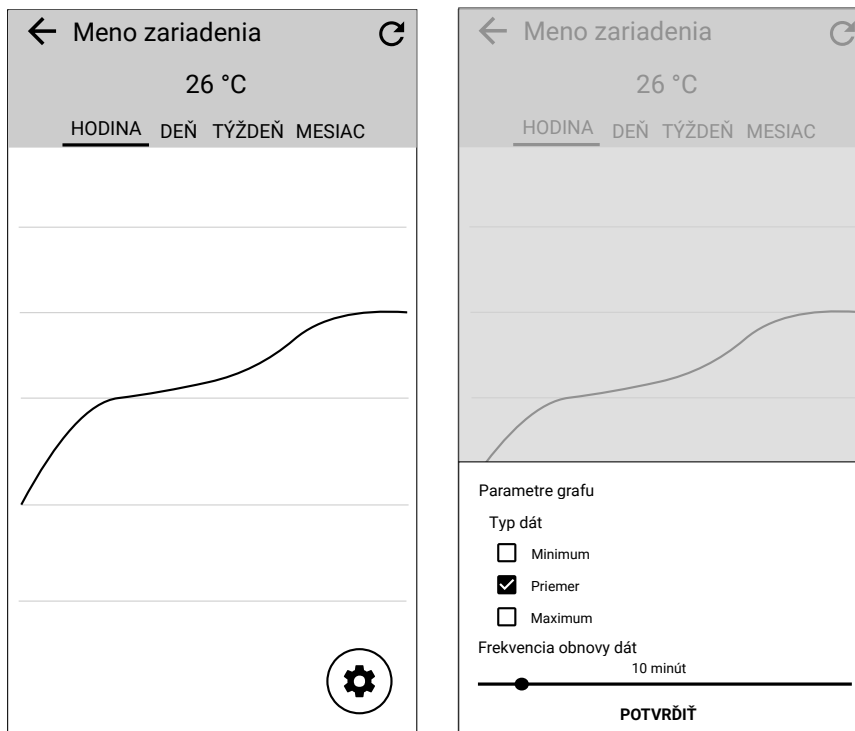
Čiarový graf

Jeho vodorovná os predstavuje čas a ľavá zvislá os predstavuje namerané hodnoty. Graf je možné približovať v horizontálnom aj vertikálnom smere. Aby bol graf prehľadný, po ťuknutí na určitý bod je možné zobraziť „bublinu“ s hodnotou alebo po dostatočnom priblížení sa zvýraznia všetky body v grafe aj s ich hodnotami. V predvolenom nastavení je hustota bodov v grafe nastavená na interval aktualizácie zariadenia. Toto nastavenie je možné zmeniť po rozkliknutí nastavení, ktoré sa zobrazia po stlačení akčného tlačidla v pravej dolnej časti obrazovky. V týchto nastaveniach je ďalej možné zvoliť zobrazenie sérií minima, priemeru a maxima ak si používateľ zvolí väčší interval ako je interval aktualizácie zariadenia.

Stĺpcový graf

Tento typ grafu je jednoduchší ako čiarový graf. Ponúka pohľad na historické stavy aktuátorov. Na vodorovnej osi je takisto ako v predchádzajúcom type zobrazený čas. Rozdiel je ale v ľavej zvislej osi, kde sú zobrazené všetky stavy, ktoré môže aktuátor nadobudnúť.

Kedže tieto názvy stavov sú vo väčšine prípadov veľmi dlhé, na osi sa zobrazia iba ich číselné odkazy. Pre zobrazenie celých názvov spárovaných s odkazmi na osi je k dispozícii tlačidlo pod grafom. Spárované názvy sú zobrazené v dialogu, ktorý je možné zrušiť s tlačidlom OK.



Obr. 4.6: Návrh grafového detailu

4.2.3 Návrh prehľadu

Prehľad umožňuje používateľovi poskladať si pohľad na svoju domácnosť podľa svojich predstáv. Základnú obrazovku tvorí hlavička s *Toolbarom* kde sú tlačidlá na otvorenie menu aplikácie (*Navigation drawer*), tlačidlo na obnovenie obsahu, tlačidlo na zobrazenie zoznamu prijatých notifikácií (nie je predmetom tejto práce) a kontextové menu kde je umiestnená položka na odstránenie aktuálneho pohľadu. Pod *Toolbarom* sa nachádzajú skupiny, ktoré tvoria jednotlivé pohľady. Zvyšok obrazovky tvorí zoznam pridaných komponent s akčným tlačidlom v pravej dolnej časti.

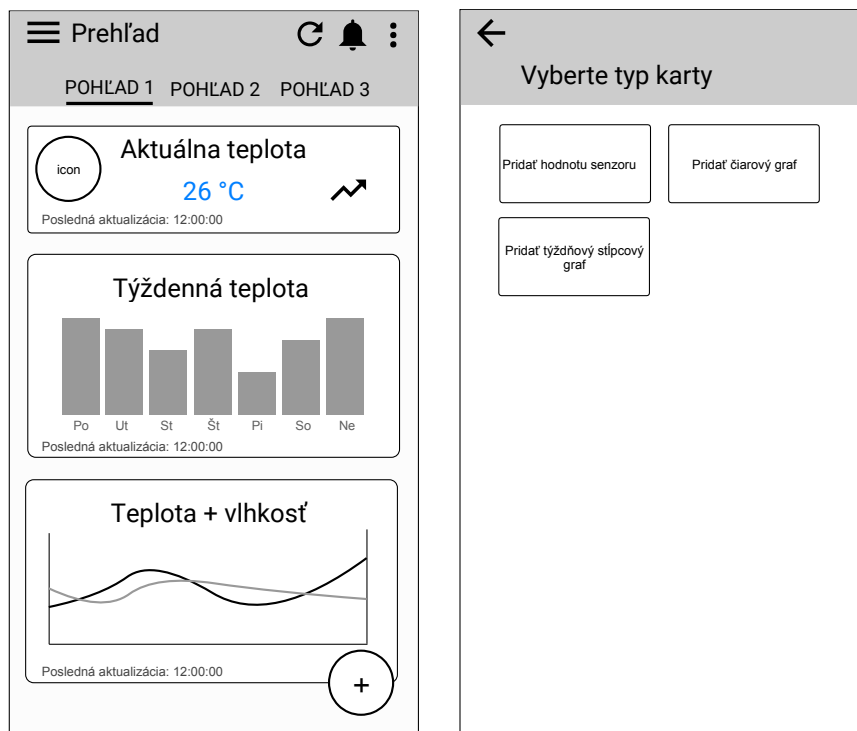
Typy položiek pohľadu:

Aktuálna hodnota senzoru/aktuátoru – Najjednoduchší typ položky, zobrazuje názov, aktuálnu hodnotu a pri senzorech navyše ukazuje trend vývoja hodnoty

Týždenný stĺpcový graf – Poskytuje týždenný prehľad nameraných hodnôt. Jeden stĺpec v grafe predstavuje agregované hodnoty z celého dňa, podľa vlastných nastavení to môže byť minimum, priemer alebo maximum.

čiarový graf – v tomto type používateľ má možnosť si vybrať grafovú sériu pre ľavú os a voliteľne pre pravú os, taktiež rozsah zobrazených hodnôt je možný od jednej hodiny

až po mesiac.



Obr. 4.7: Návrh Prehľadu

Vo všetkých typoch je možné nastaviť vlastný názov, ktorý je primárne predvolený podľa názvu senzoru, prípadne zložený z kombinácie názvu skupiny, z ktorej pochádza a názvu senzoru alebo kombinácie názvov sensorov ak ich je v jednej položke viac.

Postup pridávania a nastavovania jednotlivých položiek možno vidieť na obrázku 4.4. Po rozkliknutí akčného tlačidla + sa zobrazí akčné menu s možnosťou pridať nový pohľad alebo pridať novú položku do aktuálne vybraného pohľadu.

4.2.4 Návrh pridávania položiek do prehľadu

Pridávanie nových položiek začína s výberom aký typ položky chce používateľ pridať. Typy položiek sú zobrazené do mriežky, aby zbytočne nezaberali veľa miesta. Na zprehľadnenie výberu je k dispozícii aj ilustratívny náhľad položky. Návrh možno vidieť na obrázku 4.7 vpravo.

Po vybratí typu položky sa používateľovi zobrazí konfigurácia špecifická pre každý typ.

Aktuálna hodnota senzoru/aktuátoru

1. krok – výber senzoru, pre ktorý sa bude aktuálna hodnota zobrazovať. Dostupné senzory/aktuátory sú zobrazené v mriežke zoskupované podľa zariadení a ich skupín. Výber položky spočíva v označení ťuknutím na položku v mriežke. Pre pokračovanie na krok č. 2 slúži akčné tlačidlo umiestnené vpravo dole. Návrh obrazovky možno vidieť na obrázku 4.8 vľavo.

2. krok – Prispôsobenie názvu položky. Ako preddefinovaný názov je názov vybratej položky, ale používateľ si ho môže prispôsobiť podľa vlastných predstáv. Pre uloženie konfigurácie slúži podobne ako v predchádzajúcom kroku akčné tlačidlo umiestnené vpravo dole.



Obr. 4.8: Príklad pridávania položiek do prehľadu

Týždenný stĺcový graf

1. krok – výber senzoru, rovnaký ako pri aktuálnej hodnote.
2. krok – ďalšie nastavenia. Prispôsobenie názvu položky rovnaké ako pri aktuálnej hodnote a výber typu hodnôt, ktoré sa budú zobrazovať v grafe.

Čiarový graf

1. krok – výber senzoru pre ľavú os grafu, taktiež rovnaký ako pri aktuálnej hodnote.
2. krok – voliteľný výber senzoru pre pravú os grafu. V tomto prípade nie je predvybraný žiadny senzor. Pokiaľ používateľ tento krok preskočí, zobrazí sa v grafe iba jedna séria s ľavou osou.
3. krok – ďalšie nastavenia. Prispôsobenie názvu položky rovnaké ako pri aktuálnej hodnote a výber rozsahu zobrazovaných dát v grafe. Výber je riešený pomocou komponenty *Seekbar*.

4.2.5 Návrh detailu položky prehľadu

U všetkých položiek je možné zobraziť aj ich detail, ktorý sa líši podľa typu. Pri rozkliknutí detailu aktuálnej hodnoty sa zobrazí Grafový detail senzoru navrhovaný v podsekcii 4.2.2. Ostatné typy detailov, ktoré sú znázornené na obrázku 4.9 majú spoločný *Toolbar* kde sú umiestnené tlačidlo pre vrátenie na predchádzajúcu obrazovku, tlačidlo na obnovu obsahu a názov položky prehľadu. Zvyšok obrazovky je tvorený grafom, aký je zobrazený aj pri náhľade v zozname ale s možnosťou približovania a posúvania. Detail čiarového grafu má v spodnej časti zobrazenú legendu, pretože môže obsahovať viac sérií v grafe a nebolo by ich tak možné rozlíšiť.



Obr. 4.9: Návrh detailov položiek prehľadu

Kapitola 5

Implementácia

V tejto kapitole bude bližšie popísaný postup implementácie jednotlivých obrazoviek navrhnutých v predchádzajúcej kapitole. Na implementáciu bolo použité oficiálne vývojové prostredie pre Android – Android Studio vyvíjané spoločnosťou Google [8].

Grafické rozhranie sa pre Android implementuje v jazyku Java s definíciou grafického rozhrania v XML. Cieľová verzia API androidu je 23 a kompatibilita je zaručená až do API 15 čo zaručuje možnosť nainštalovať aplikáciu na 97,3 % všetkých zariadení na trhu, viď tabuľka 3.1. Túto kompatibilitu do veľkej miery zaručujú podporné knižnice tzv. *Android support libraries* a niektoré knižnice tretích strán.

5.1 Použité knižnice tretích strán

StyledDialogs for Android – knižnica[9], ktorá zaručuje vzhľad dialógov kompatibilným s material dizajnom popísaným v kapitole 3.4.

MP Android Chart – knižnica[10], ktorá implementuje vykresľovanie grafov. Podporuje až 8 typov grafov ale v aplikáciách boli využité iba čiarový a stĺpcový typ grafu. Medzi jej hlavné vlasti patrí široká možnosť prispôsobenia vzhľadu, schopnosť približovania a posúvania grafov v oboch smeroch a vyznačovanie bodov po dotyku s taktiež prispôbovaným vzhľadom.

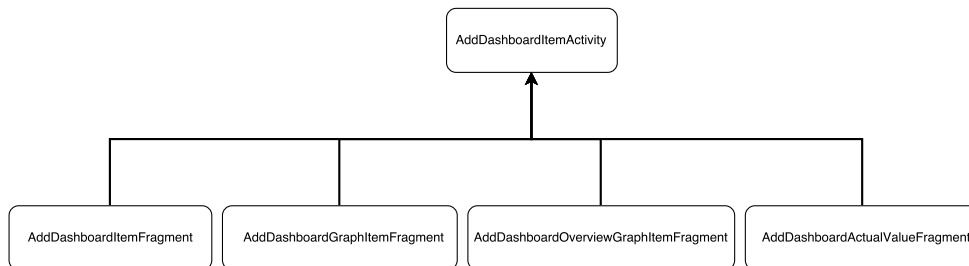
FloatingActionButton – knižnica[11], ktorá implementuje *Floating Action Button* z material dizajnu popísané v kapitole 3.4.1.

BottomSheet – knižnica [12], ktorá implementuje princíp *Bottom sheet* z material dizajnu popísaný v kapitole 3.4.1.

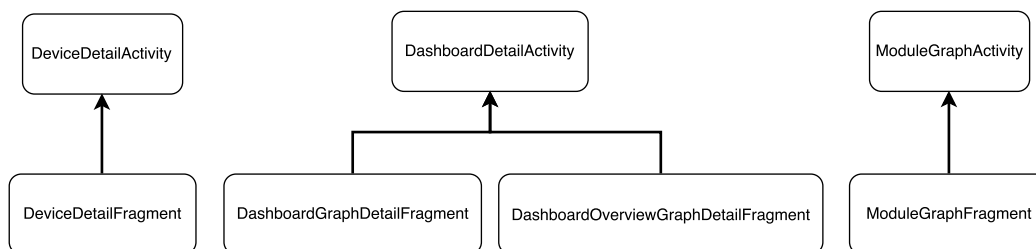
5.2 Implementácia obrazoviek

Implementácia prebehla v súlade s návrhom popísaným v predchádzajúcej kapitole. Aby nedošlo k zbytočnému kopírovaniu kódu, boli využité pri implementáciách *Fragmenty*, ktoré väčšinou zdieľajú rovnakú *Activitu*.

Pri implementáciách bol z veľkej časti využívaný *Coordinator layout* (3.3.1), ktorý umožňuje definovať pokročilé závislosti na grafických prvkoch. Napríklad v detaile zariadenia bola implementovaná schopnosť transformácie hlavičky do *Toolbaru* pomocou kombinácie s *Collapsing Toolbar Layout*. Ako možno vidieť na obrázku 5.3, aby sme zaručili schovávanie



Obr. 5.1: Schéma implementácie obrazoviek pridávania do prehľadu



Obr. 5.2: Schéma implementácie ostatných obrazoviek

pri scrollovaní obrazovky, je nutné definovať `layout_scrollFlags` a `layout_collapseMode` a o ostatné sa už postará samotný *Coordinator Layout*.

```

<android.support.design.widget.CoordinatorLayout>
  <android.support.design.widget.AppBarLayout>
    <android.support.design.widget.CollapsingToolbarLayout
      app:layout_scrollFlags="scroll|exitUntilCollapsed">
      <ImageView
        android:id="@+id/device_detail_icon"/>
      <TextView
        android:id="@+id/device_detail_device_name"
        app:layout_collapseMode="parallax"/>
      <HorizontalScrollView
        android:id="@+id/device_detail_features_scrollview"
        app:layout_collapseMode="parallax">
        <LinearLayout
          android:id="@+id/device_detail_features_layout"
          android:orientation="horizontal"/>
        </HorizontalScrollView>
      <include layout="@layout/layout_beeon_toolbar"/>
    </android.support.design.widget.CollapsingToolbarLayout>
  </android.support.design.widget.AppBarLayout>
  <android.support.v7.widget.RecyclerView
    app:layout_behavior="@string/appbar_scrolling_view_behavior"/>
</android.support.design.widget.CoordinatorLayout>
  
```

Obr. 5.3: Ukážka využitia *Coordinator layoutu*

5.2.1 Implementácia prehľadu

V prehľade bola využitá možnosť *RecyclerView* adaptéru, že dokáže mať viac typov prvkov v jeho zozname. Na to slúži metóda `getItemViewType`, kde sa zisťuje typ položky podľa aktuálnej pozície v zozname. Následne túto metódu využívajú aj `onCreateViewHolder` a `onBindViewHolder`, aby vytvorili potrebný layout podľa pozície.

Kapitola 6

Vyhodnotenie

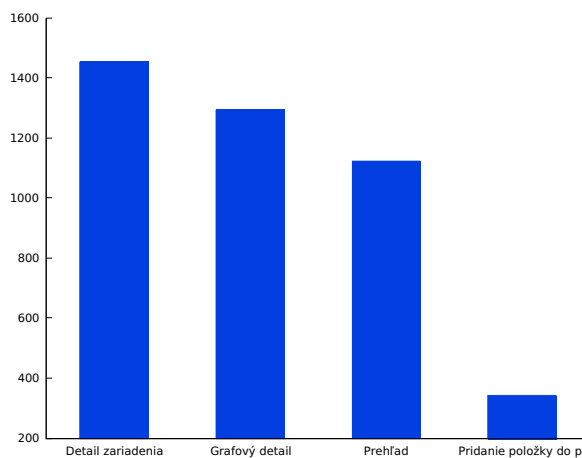
6.1 Testovanie

Testovanie prebiehalo vo viacerých úrovniach. Počas implementácie prebiehalo na emulátoroch od verzie API 15 až po verziu API 23 tak, aby pokrylo všetky podporované verzie Androidu. Ďalej testy pokračovali aj na fyzických zariadeniach s API 15, 19, 22 a 23. V ďalšej úrovni sa na testovaní podieľali ostatní vývojári BeeeOn projektu. Najaktuálnejšie zosťavenie aplikácie bolo k dispozícii pomocou nástroja Jenkins, ktorý sa staral o automatický preklad zdrojových súborov.

6.2 Štatistiky Google Analytics

V aplikácii bol implementovaný nástroj Google Analytics[13] od spoločnosti Google na analyzovanie interakcie s aplikáciou. Implementácia prebehla pomocou dodávanej knižnice, ktorá dokáže jednoducho poslať zobrazenia obrazoviek alebo definovaných vlastných udalostí ako napríklad stlačenie určitého tlačidla. Zozbierané dáta pochádzajú z mesačného intervalu.

6.2.1 Počet zobrazení implementovaných obrazoviek

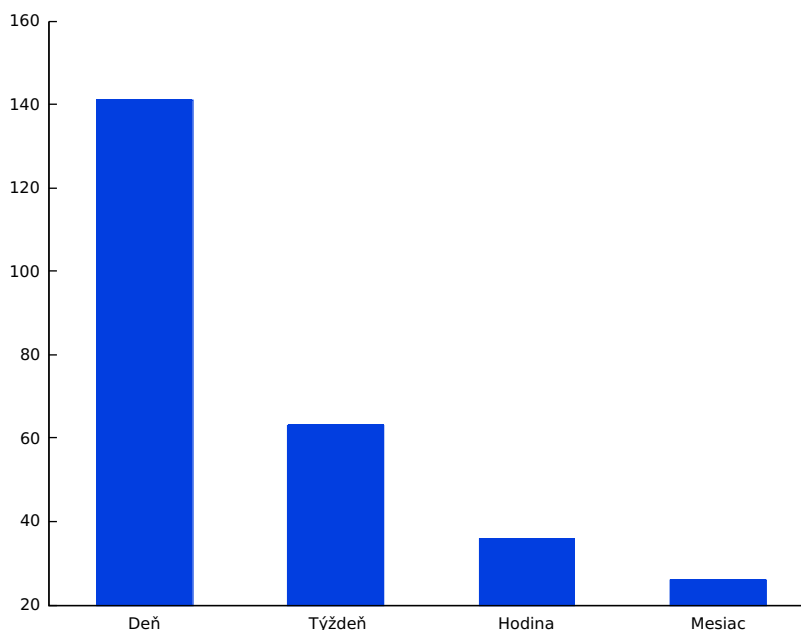


Obr. 6.1: Počet zobrazení implementovaných obrazoviek

Ako možno vidieť v grafe na obrázku 6.1, najviac používaná bola obrazovka detailu zariadenia, kde je možné vidieť najviac údajov o konkrétnom zariadení a na druhu stranu používatelia boli s ňou najviac zoznámení, pretože bola implementovaná ako prvá.

6.2.2 Počet zobrazení rozsahov grafu v grafovom detaile

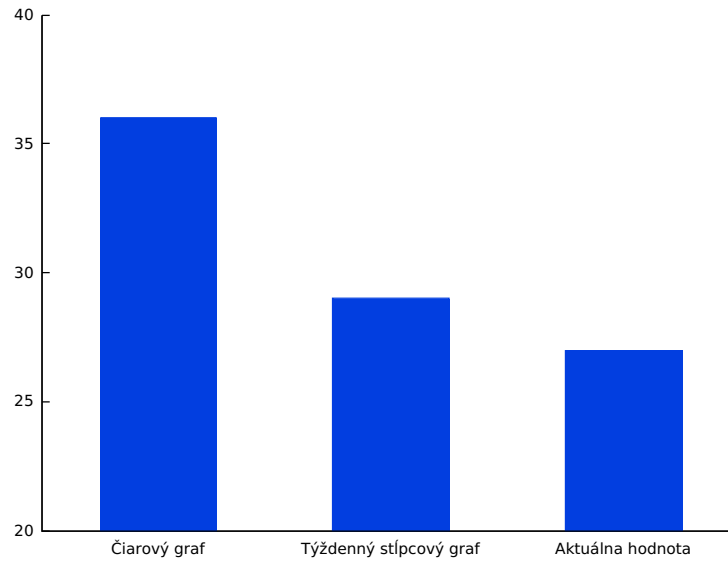
Najoblúbenejším rozsahom grafu sa stal rozsah jedného dňa, z čoho vyplíva, že používateľov zaujíma najviac vývoj počas celého dňa. Naopak mesačný rozsah nebol zobrazovaný skoro vôbec, čo môže byť príčinou príliš dlhého čakania na odpoveď zo serveru z dôsledku veľkého objemu prichádzajúcich dát.



Obr. 6.2: Počet zobrazení jednotlivých rozsahov grafu v grafovom detaile

6.2.3 Počet pridaní položiek do prehľadu

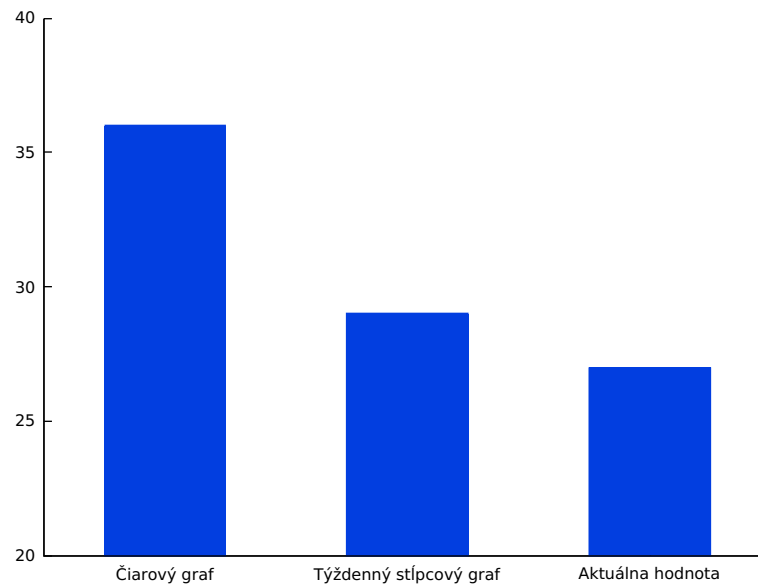
Najpridávannejším typom položky do prehľadu sa stala položka čiarového grafu poskytujúca najviac možností prispôsobenia, čo oslovilo najväčší počet používateľov aplikácie. Ďalším dôvodom môže byť, že ako jediná poskytovala možnosť zobraziť hodnoty z viac ako jedného senzoru, čo poskytovalo používateľovi napríklad závislosti teploty a vlhkosti.



Obr. 6.3: Počet pridaní položiek do prehľadu podľa typu

6.2.4 Počet kliknutí na detail položky v prehľade

Najviac zobrazovaný detail v prehľade bol detail čiarového grafu. Tento výsledok priamo súvisí aj s prechádzajúcim výsledkom pretože platí úmera počtu pridaní danej položky s počtom zobrazenia jej detailu. Ďalej ak si používateľ definoval rozsah grafu jedného mesiaca, tak mu „mini“ graf v položke určite nevyhovoval a bol nútený zobraziť detail.



Obr. 6.4: Počet kliknutí na detail položky prehľadu podľa typu

Kapitola 7

Záver

V rámci riešenia práce som sa zoznámil so systémom inteligentnej domácnosti BeeeOn, naštudoval princípy implementácie grafického rozhrania pre systém Android a navrhol jednotlivé obrazovky v súlade s požiadavkami systému. Implementácia prebehla tak, aby v budúcnosti bola jednoducho rozšíriteľná napríklad o ďalšie typy pohľadov domácnosti.

V budúcnosti sa môže práca rozšíriť o ďalšie typy grafových pohľadov, hlavne koláčový, ktorý sa hodí pre zobrazovanie najčastejších stavov aktuátorov. Takisto je možné rozšíriť už existujúce pohľady aby poskytovali používateľovi viac nastavení a prispôbení.

Keďže celý systém BeeeOn je vyvíjaný ako open source riešenie, výsledok mojej práce je dostupný v beeeon repozitári na Githube alebo aj na Google Play.

Literatúra

- [1] Architektúra domácnosti. *IOT wiki* [online]. 2014 [cit. 2016-04-10]. Dostupné z: <<https://ant-2.fit.vutbr.cz/projects/iot/wiki/Architektura>>.
- [2] Open Handset Alliance. [online]. 2007 [cit. 2016-04-09]. Dostupné z: <http://www.openhandsetalliance.com/oha_faq.html>.
- [3] Layouts. *Android developers* [online]. 2016 [cit. 2016-04-10]. Dostupné z: <<http://developer.android.com/guide/topics/ui/declaring-layout.html>>.
- [4] GARENTA, M. a NAKAMURA, M. *Learning Android, Second Edition*. 2. vyd. USA: O'Reilly Media, Inc., 2014. ISBN 978-1-449-31923-6.
- [5] Coordinatorlayout. *Android Developers API* [online]. [cit. 2016-04-10]. Dostupné z: <<http://developer.android.com/reference/android/support/design/widget/CoordinatorLayout.html>>.
- [6] Google Material Design. [online]. 2016 [cit. 2016-04-10]. Dostupné z: <http://www.openhandsetalliance.com/oha_faq.html>.
- [7] MEW, K. *Learning Material Design*. 1. vyd. Birmingham: Packt Publishing, 2015. ISBN 1785289810.
- [8] Android Studio Overview. *Android Developers* [online]. 2016 [cit. 2016-05-10]. Dostupné z: <<http://developer.android.com/tools/studio/index.html>>.
- [9] AVAST. StyledDialogs for Android. [online]. 2016 [cit. 2016-05-15]. Dostupné z: <<http://developer.android.com/tools/studio/index.html>>.
- [10] JAHODA, P. MP Android Chart. [online]. 2016 [cit. 2016-05-15]. Dostupné z: <<https://github.com/PhilJay/MPAndroidChart>>.
- [11] TARIANYK, D. Floating Action Button. [online]. 2016 [cit. 2016-05-15]. Dostupné z: <<https://github.com/Clans/FloatingActionButton>>.
- [12] FLIPBOARD. BottomSheet. [online]. 2016 [cit. 2016-05-15]. Dostupné z: <<https://github.com/Flipboard/bottomsheet>>.
- [13] GOOGLE. Google Analytics. [online]. 2016 [cit. 2016-05-16]. Dostupné z: <<https://www.google.com/analytics/>>.

Prílohy

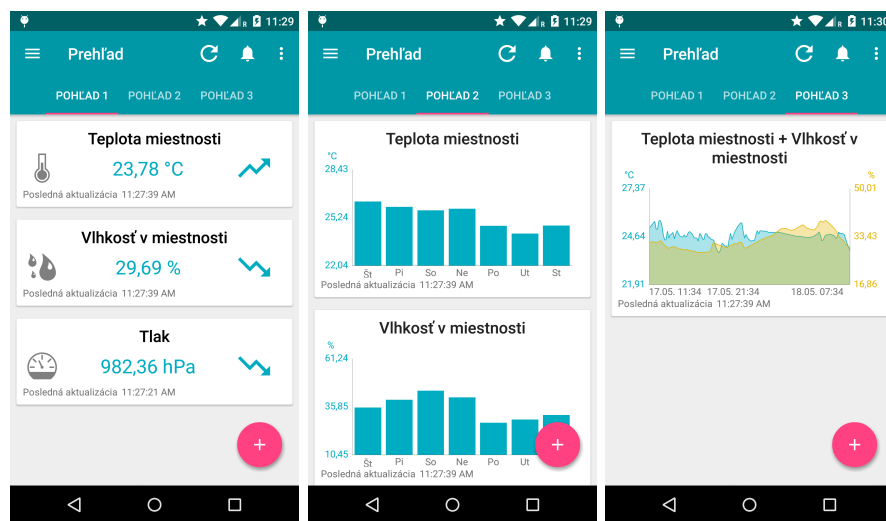
Zoznam príloh

A **Obrazovky aplikácie**

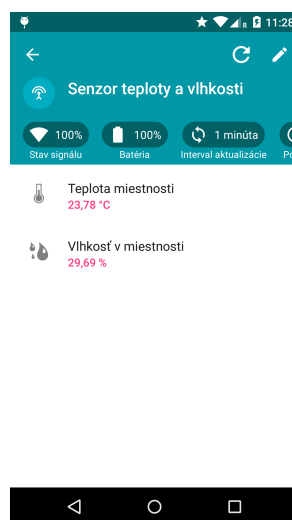
32

Príloha A

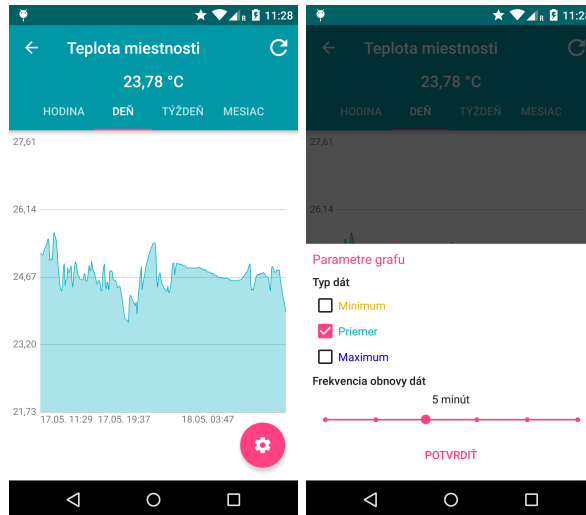
Obrazovky aplikácie



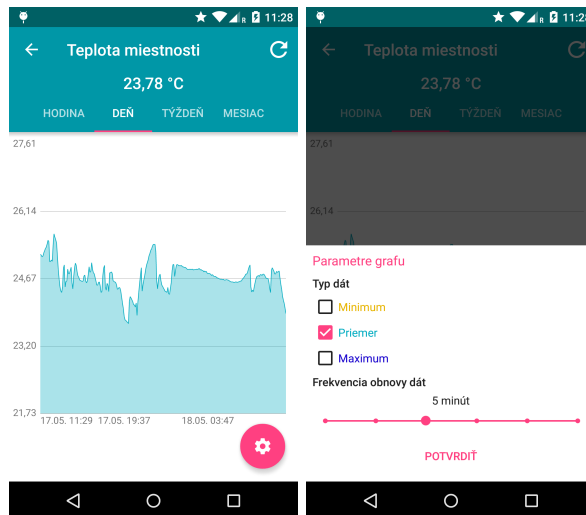
Obr. A.1: Obrazovky prehľadu



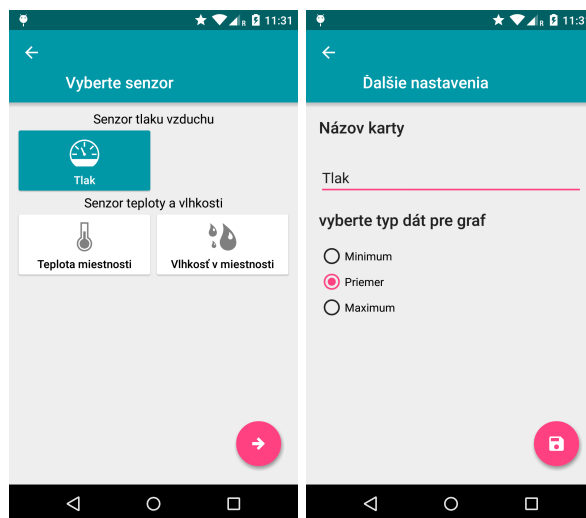
Obr. A.2: Obrazovka detailu zariadenia



Obr. A.3: Obrazovky grafového detailu



Obr. A.4: Obrazovky grafového detailu



Obr. A.5: Obrazovky pridávania položky do prehľadu