



**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

F3

**Fakulta elektrotechnická
Katedra kybernetiky**

Bakalářská práce

Jednoduchý elektronický domácí spoločník

Maroš Pechník
Kybernetika a robotika

Máj 2019

Vedúci práce: Ing. Petr Novák Ph.D.



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Pechník** Jméno: **Maroš** Osobní číslo: **465991**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra kybernetiky**
Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Jednoduchý elektronický domácí společník

Název bakalářské práce anglicky:

Simple Electronic Home Companion

Pokyny pro vypracování:

1. Prostudujte existující návrhy a projekty zabývající se tvorbou tzv. elektronických domácích společníků pro starší a osamělé lidi. Zaměřte se zejména na tzv. stacionární (umístěné na pevném místě) nikoli mobilní (pohyblivé) zařízení.
2. Rozdělte schopnosti a dovednosti těchto el. domácích společníků do několika skupin / kategorií podle jejich využitelnosti, vhodnosti a nutnosti v běžném domácím prostředí.
3. Navrhněte a vytvořte prototyp jednoduchého el. domácího společníka jako aplikaci na tablet / PC, která bude poskytovat následující možnosti:
 - rady pro ovládání různých běžných domácích spotřebičů a zařízení,
 - připomínání důležitých (zejména) denních akcí a činností,
 - schopnost kontroly domácího prostředí pomocí externích senzorů,
 - tvorba záznamu akcí a základní zobrazení jako přehled činností uživatele.

Seznam doporučené literatury:

- [1] Webové stránky projektů obdobných zařízení
- [2] Matthew MacDonald, Pro WPF in C# 2010, Apress, 2013, ISBN 978-1-4302-7205-2
- [3] Matthew MacDonald, Pro WPF: Windows Presentation Foundation in .NET 3.0, Apress, 2007, ISBN 1-59059-782-6
- [4] Xamarin Cross-platform Application Development (WWW, dokumentace, knížky)

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Petr Novák, Ph.D., Analýza a interpretace biomedicínských dat FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **09.01.2019** Termín odevzdání bakalářské práce: **24.05.2019**

Platnost zadání bakalářské práce: **20.09.2020**

Ing. Petr Novák, Ph.D.
podpis vedoucí(ho) práce

doc. Ing. Tomáš Svoboda, Ph.D.
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Pod'akovanie / Prehlásenie

Chcel by som sa podakovať vedúce-
mu práce Petrovi Novákovi za podporu a
pomoc pri spracovávaní tejto práce. Ta-
kisto by som sa chcel podakovať aj rodi-
ne, priateľke a priateľom za podporu.

Prehlasujem, že som predloženú prá-
cu vypracoval samostatne a že som uvie-
dol všetky použité informačné zdroje v
súlade s Metodickým pokynom o dodr-
žiavaní etických princípov pri príprave
vysokoškolských záverečných prác.

V Prahe dňa 24.5.2019

.....

Abstrakt / Abstract

V teoretickej časti je prehľad súčasných hlasových aj robotických asistentov určených do domácnosti. Následuje zhrnutie súčasného stavu v tejto oblasti a vyslovenie potreby na jednoducho konfigurovateľnú aplikáciu s jednotným ovládaním určenú priamo pre seniorov.

V návrhu je odôvodnený výber programovacieho prostredia a programovacieho jazyka pre tvorbu aplikácie. Ďalej sú popísané jednotlivé časti aplikácie, ich funkcia a prípadná implementácia.

Záverečná časť je určená na pochopenie konfigurácie aplikácie spolu s ukázkami a popisom konfiguračných súborov pre obrazovky, deklaráciu senzorov, konštant, premenných a pravidiel a takisto aj s ukázkami tvorby pravidlových súborov.

Kľúčové slová: Aplikácia pre seniorov, konfiguračný súbor, senzor, pravidlový súbor, domáci asistent, domáci spoločník

In theoretical part there is overview of existing voice and robotic home assistants designed for the household. Then there is a summary of the current situation in the field of home assistants and stated need for simple configurable application with uniform control and designed specially for seniors.

In section design there is explained selection of programming framework and programming language. Then there are explained parts of application, their function and implementation.

Final section is intended to understand how to configure the application with description of configuration files for screens, declaration of sensors, constants, variables and rules. There is also shown how to create rules file.

Keywords: Application for seniors, configuration file, sensor, rules file, home assistant, home companion

Title translation: Simple electronic home companion

Obsah /

1 Úvod	1	5 Záver	31
1.1 Súčasný stav	1	Literatúra	33
1.1.1 Hlasoví asistenti.....	1	A Skratky	35
1.1.2 Hlasový asistent Elliq	1	B Obsah priloženého CD	36
1.1.3 Robot Buddy	2		
1.1.4 Robot Aeolus.....	2		
1.1.5 Tablet GrandPad	3		
1.2 Ďalšie aplikácie	3		
1.3 Zhrnutie	3		
2 Ciele tejto práce	5		
3 Návrh a tvorba aplikácie	6		
3.1 Zvolená technológia	6		
3.2 Princíp činnosti aplikácie	7		
3.3 Obrazovky a ich prepínanie	7		
3.4 Nápoveda k obsluhu domácich spotrebičov	8		
3.5 Pripomínanie informácií	8		
3.6 Časti používajúce internet.....	9		
3.6.1 Funkcionalita zábava	9		
3.6.2 Predpoveď počasia	10		
3.6.3 Volanie cez Skype	12		
3.7 Využitie externých senzorov ...	12		
3.7.1 Sensory pre demonštráciu	12		
3.7.2 Príjem a uloženie prijatých dát zo senzorov ...	13		
3.7.3 Pravidlá	14		
3.8 Hlasový výstup.....	15		
3.9 Záznam akcií aplikácie a užívateľa a ich zobrazenie.....	15		
4 Vzorová aplikácia, jej konfigurácia a činnosť	17		
4.1 Základná konfigurácia	17		
4.1.1 Kľúčové slová.....	17		
4.1.2 Príkazy	18		
4.1.3 Zobrazenie viacerých spotrebičov súčasne	22		
4.2 Umiestnenie adresárov a ich význam	23		
4.3 Práca so senzormi.....	23		
4.3.1 Súbor senzorov.....	23		
4.3.2 Súbor konštánt.....	24		
4.3.3 Súbor premenných	24		
4.3.4 Súbory pravidiel	24		
4.4 Ukážky varovaní na základe pravidiel	25		

Kapitola 1

Úvod

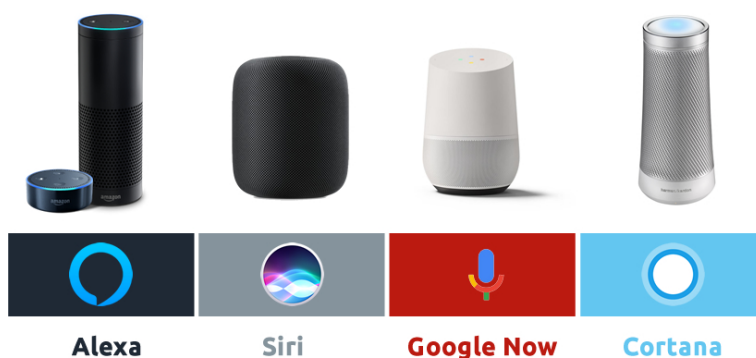
V súčasnosti dochádza v našej populácii k jej starnutiu a v najbližších rokoch to bude len horšie. Podľa údajov eurostatu [1], bolo v roku 2016 v EÚ 19,2% obyvateľstva starších ako 65 rokov, v Českej Republike to bolo 18,3% a na Slovensku 14,4%. Trend je ale stúpajúci a populácia starne, takže v najbližších rokoch bude dôchodcov stále viac. Rozdiel v priemernej dĺžke života medzi ženami a mužmi v Česku a na Slovensku sa pohyboval v roku 2016 okolo 4 rokov. Až 32,4% ľudí v Česku a 30,4% ľudí na Slovensku nad 65 rokov žije osamote. Teda dôchodcov pribúda a mnoho z nich žije bez ľudskej spoločnosti čo sa prejavuje hlavne na ich zdraví.

1.1 Súčasný stav

V dnešnej dobe sa domáci asistenti dostávajú čoraz viac do povedomia ľudí. Dokážu odpovedať na otázky, pripomínať dôležité udalosti ako napríklad užívanie liekov, môžu uskutočniť hovor, niektoré vedia komunikovať s inými tzv. smart zariadeniami v domácnosti.

1.1.1 Hlasoví asistenti

Hlasoví asistenti ako napríklad Amazon Alexa, Google Home alebo Apple Siri dokážu viesť rozhovor, ale iba po anglicky, avšak niekedy až na nepoznanie od človeka. Pomocou hlasu im môžete zadávať príkazy na ovládanie rôznych aplikácií aj na iných spárovaných zariadeniach. Hlasoví asistenti nedokážu fungovať samostatne, musia sa spárovať s tabletom alebo chytrým telefónom. [2]



Obrázok 1.1. Hlasoví asistenti [3]

1.1.2 Hlasový asistent ElliQ

Existujú aj hlasoví asistenti disponujúci aj obrazovkou, napríklad ElliQ. Tento systém s Vami nadviaže aj „očný kontakt“ a nie je to iba hlasový asistent. Dokáže si s Vami zahrať spoločenskú hru, ukáže Vám prijaté správy a umožní na ne aj odpovedať. Vie uskutočniť aj videohovor, pripomenúť, kedy si treba vziať lieky alebo zacvičiť.



Obrázok 1.2. Domáci asistent Elliq disponujúci obrazovkou a „hlavou“ [4]

■ 1.1.3 Robot Buddy

Okrem statických asistentov sú na trhu aj mobilné roboty určené na robenie spoločnosti. Príkladom je robot Buddy, ktorý využíva umelú inteligenciu, podľa tváre rozpozná s kým komunikuje a je založený na open-source platforme. Tento robot dokáže vyjadriť svoje emócie, pripomenie udalosti z kalendáru, počasie alebo správy. Zahrá si hru s deťmi, sprostredkuje videohovor. Aj v neprítomnosti osôb v domácnosti chodí po dome/byte a kontroluje, či napríklad neuniká plyn alebo nedošlo k požiaru. V prípade neobvyklej situácie odošle upozornenie. Ovláda sa hlasom alebo aj vzdialene, pomocou spárovaných zariadení dokáže ovládať prístroje v celej domácnosti. Ako aj ostatné spomenuté zariadenia, aj Buddy reaguje a komunikuje v anglickom jazyku. Jeho cena je momentálne okolo 600\$.



Obrázok 1.3. Robot Buddy [5]

■ 1.1.4 Robot Aeolus

Robot Aeolus sa od ostatných spomenutých robotov sa líši tým, že je vybavený ramienom, vďaka ktorému vie manipulovať s predmetmi. Robot rozumie domácejmu prostrediu v zmysle, že vie, kam dané predmety patria. Dokáže sa prispôsobiť aj zmenám v usporiadaní predmetov. Vie rozoznať množstvo predmetov v domácnosti, zdvihnúť ich napríklad zo zeme a vrátiť ich na pôvodné miesto. Zapamätá si, kde videl predmet naposledy a v prípade potreby ho tam vráti. Vďaka zabudovanému vysávaču alebo mopu dokáže samostatne vyčistiť domácnosť. Navyše má robot zabudovaného hlasového asistenta ako Amazon Alexa alebo Google Home, takže dokáže viesť aj konverzáciu. Robot

má množstvo senzorov a vyvinuté biometrické technológie, vďaka čomu vie rozoznávať rodinných príslušníkov, priestor na žitie alebo domáce spotrebiče. Na základe polohy tela vie v prípade núdze zavolať pomoc. Aeolus je stále iba prototyp a jeho cena nie je známa, ale samotná firma tvrdí že bude lacnejší, ako rodinná dovolenka pri mori.



Obrázok 1.4. Robot Aeolus [6]

1.1.5 Tablet GrandPad

Tablet vytvorený konkrétne pre seniorov sa volá GrandPad. Je to tablet s jednoduchým ovládaním. Dá sa s ním telefonovať, posilať mailom hlasové správy, zistiť počasie, prijímať a posilať fotky rodinným príslušníkom. Grandpad sa dá napojiť aj na sociálne siete. Dajú sa na ňom hrať aj hry a k dispozícii je pre seniorov online podpora, kde cez videohovor konzultujú svoj problém so zamestnancom podpory. Samotný tablet stojí okolo 200\$, ale je k tomu nutné každý mesiac platiť 40\$.



Obrázok 1.5. Tablet GrandPad [7]

1.2 Ďalšie aplikácie

Existujú aj rôzne aplikácie, ktoré sa snažia seniorom uľahčiť život, tým že mobil nastaví do režimu zjednodušeného ovládania, pripomenú čas užívania liekov alebo stimulujú mozog jednoduchými hrami, krížovkami, prípadne sudoku.

1.3 Zhrnutie

Nevýhodou hlasových asistentov pre českého alebo slovenského (celkovo dôchodcu z nie anglicky rozprávajúcej krajiny) je okrem ceny aj to, že sa s ním väčšina nedokáže dohodovať. Kvôli tejto jazykovej bariére strácajú pre tuzemských dôchodcov zmysel.

Rovnako je to aj s domácim asistentom Elliq. Roboty taktiež komunikujú v anglickom jazyku, niektoré využívajú už spomenutých hlasových asistentov. Ich cena je na príjmy väčšiny dôchodcov privysoká. Aplikácie na telefón sú užitočné, ale akjed' bežia na jednom zariadení, tak sú často veľmi rozdielne v ovládaní, čo je pre starsích ľudí veľmi nepríjemné.

Hlasoví asistenti ako Amazon Alexa, Apple Siri alebo Google Home sú zariadenia určené pre bežného spotrebiteľa, nie sú vyrobené konkrétne pre seniora. Na súčasnom (českom a slovenskom) trhu sa mi nepodarilo nájsť jednoducho konfigurovateľné zariadenie určené priamo pre dôchodcov. Asi nejzávažnejším nedostatkom je, že väčšina aplikácií, asistentov a robotov komunikuje v anglickom jazyku. Ďalší problém spočíva v tom, že starší ľudia niektoré slová nevyslovujú zreteľne a ťažšie sa im rozumie, čo môže spôsobovať problémy pri hlasovej komunikácii s technikou. Hlasových asistentov treba párovať s chytrým telefónom alebo počítačom, čo je pre bežného dôchodcu veľmi náročné.

Funkcionality robotov by život dôchodcov zjednodušili, ale ich cena je vysoká. Na trhu chýba napríklad aj základná aplikácia jednoduchá na ovládanie, ktorá by spájala základné funkcionality, akými sú predpoveď počasia, pripomienkovač, návod na obsluhu prístrojov v domácnosti do jedného komplexného celku.

V súčasnej dobe je dostupných veľmi málo tzv. domácich spoločníkov, preto ich hodnotenie formou rozdelenia do niekoľkých kategórií nie je úplne na mieste. Pri ich hodnotení sa teda zameriam na parametre z hľadiska tejto práce:

- **Hlasový výstup** - na jednu stranu úplná väčšina existujúcich zariadení / projektov obsahuje hlasový výstup, ale na druhú stranu je tento hlasový výstup (snáď) vždy v anglickom jazyku, čo je veľká prekážka pre použitie v Česku.
- **Konfigurovateľnosť** - väčšina domácich spoločníkov sú sofistikované zariadenia bez možnosti vlastnej konfigurácie, teda úpravy chovania podľa cieľovej osoby.
- **Cena** - cena je veľkou prekážkou, na trhu chýba jednoduché a lacné zariadenie, ktoré by si dôchodcovia mohli dovoliť a dokázali s ním pracovať.

Kapitola 2

Ciele tejto práce

Cielom tejto bakalárskej práce je vytvoriť jednoduchého domáceho spoločníka a čiastočne aj strážcu ako aplikáciu na tablet alebo počítač. Pri tvorbe aplikácie sa bude dbať na tieto hlavné požiadavky:

- jednoduchosť ovládania, rovnaký spôsob ovládania celej aplikácie
- internetové pripojenie (aspoň občas) pre aktualizáciu informácií v aplikácii
- aplikácia bude jednoducho konfigurovateľná (nepredpokladá sa, že samotným dôchodcom, ale napríklad rodinným príslušníkom)
- prehľadné, hlavne dotykové ovládanie

Vytvorená aplikácia bude teda zahŕňať nasledujúce funkcie:

Rady pre ovládanie rôznych bežných domácich spotrebičov

Bude možno jednoducho nakonfigurovať s akými konkrétnymi spotrebičmi v domácnosti potrebuje senior poradiť. Po kliknutí na spotrebič z výberu spotrebičov na poradenie sa po česky prečíta text vyzývajúci na kliknutie na konkrétnu časť prístroja a následné poradenie. Po kliknutí na konkrétnu časť spotrebiča (napríklad tlačidlo *volume +* na rádiu) aplikácia povie, ako sa tlačidlo ovláda a k čomu slúži.

Pripomínanie dôležitých denných akcií a činností

V konfiguračnom súbore bude možno napísať konkrétne dni, časy a texty pripomienok. Keď nastane čas na pripomenutie, aplikácia na obrazovke zobrazí nakonfigurovaný text pripomienky, prípadne obrázok a čaká na potvrdenie od dôchodcu, že pripomienku zobral na vedomie.

Pripomienkovač môže slúžiť ako na pripomínanie užívania liekov tak aj na pripomenutie obľúbeného programu v televízii alebo pripomenutie narodenín/menín rodinných príslušníkov alebo priateľov.

Schopnosť kontroly domáceho prostedia pomocou externých senzorov

Aplikácia bude prijímať dáta z domácich senzorov, ktoré budú mať predovšetkým bezpečnostný účel. Budú kontrolovať napríklad, či nedochádza k úniku plynu alebo či nie je príliš plná vaňa. V prípade nízkej teploty v byte a otvorených oknách aplikácia dôchodcu upozorní, aby okná zatvoril.

Senzory budú mať aj monitorovaciu úlohu, napríklad ak bude dôchodca dlho sedieť na gauči, aplikácia ho upozorní, aby si zacvičil alebo sa poprechádzal.

Tvorba záznamu akcií a základné zobrazenie ako prehľad činností užívateľa

Dáta so senzorov budú spracovávané na vytvorenie grafov a prehľadu činností, akými sú napríklad čas vstávania, čas ako dlho bola zapnutá televízia, doba strávená v posteli a podobne.

Taktiež je nutné zdôrazniť, čím sa táto práca nezaoberá:

- ovládaniu aplikácie hlasovými povelmi
- výrobe senzorov a komunikačných protokoloch
- zabezpečeniu aplikácie a akejkoľvek komunikácie
- jedná sa iba o experimentálny prototyp, nie finálne zariadenie, za jeho využívanie nenesie tvorca žiadnu zodpovednosť

Kapitola 3

Návrh a tvorba aplikácie

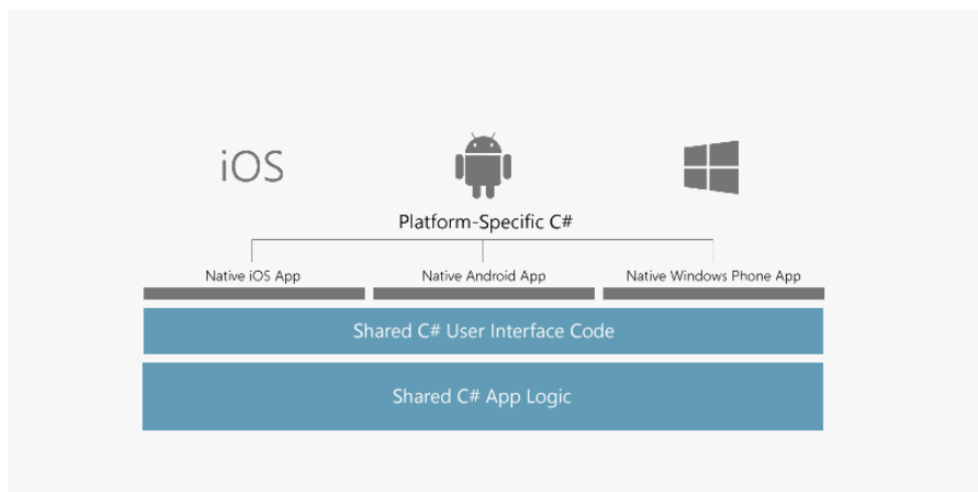
Ako základné požiadavky pri návrhu tohto riešenia boli zvolené tieto:

- dostupnosť vývojových nástrojov, prípadne možná portácia aj na iné operačné systémy
- dostatočne vysokoúrovňový a rozšírený programovací jazyk
- konfigurácia najmä pomocou textových súborov, ich umiestnenie v adresároch
- pevne daná štruktúra/pravidlá pre tvorbu konfigurácie

3.1 Zvolená technológia

Na vývoj aplikácie som zvolil platformu .NET a jej jazyk C#. Príbuzný projekt Xamarin, ktorý využíva taktiež jazyk C#, je nástroj pre vývoj najmä mobilných aplikácií pre platformy Android, IOS a Windows a teda umožňuje písať väčšinu kódu tak, aby bežala na všetkých troch platformách. Aplikácia sa dá priamo písať vo voľne prístupnom vývojovom prostredí Visual Studiu Community, ktoré je zadarmo. To bol ďalší dôvod výberu programovacieho jazyku C#. Konkrétne sa pre vývoj aplikácie použil framework WPF [8] [9], čo je nástroj na tvorbu GUI. Pri tvorbe aplikácie som používal aj [10], [11], [12].

Hlavnou výhodou platformy NET teda je, že sa napíše jeden kód použiteľný pre 3 platformy a nie je potrebné písať celý kód pre každú platformu zvlášť. Programový kód netvoriaci GUI je cez uvedené platformy v podstate rovnaký, iba kód tvoriaci GUI aplikácie sa v niektorých smeroch líši.



[13]

Obrázok 3.1. Štruktúra kódu pre rôzne operačné systémy vďaka nástroju Xamarin

3.2 Princíp činnosti aplikácie

Cielom je, aby bola aplikácia čo najviac prispôsobiteľná seniorovi. To je zabezpečené konfiguračnými súbormi, ktorými sa nastavujú jednotlivé obrazovky. Aplikácia taktiež prijíma dáta zo senzorov, ktoré sú pomocou nakonfigurovaných pravidiel spracovávané a následne aj vykreslené do grafu.

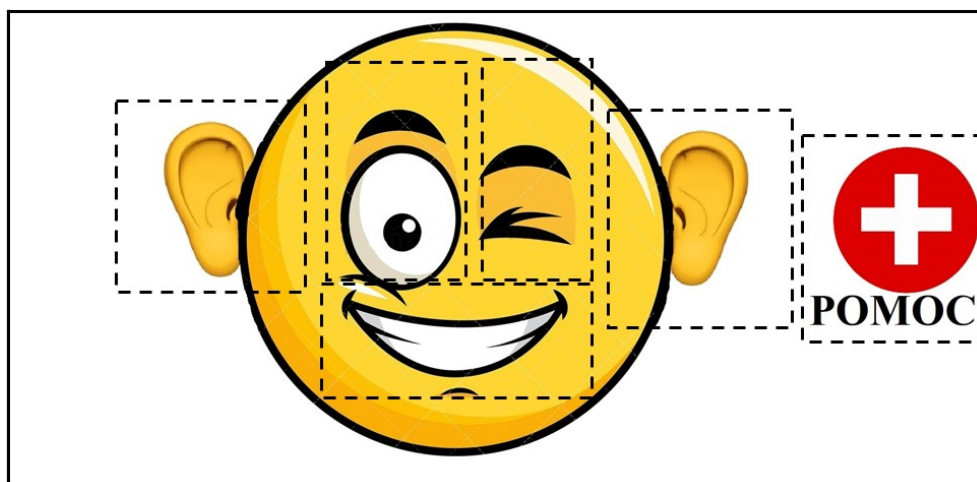
Celá aplikácia je rozdelená do niekoľkých aj paralelne pracujúcich blokov. Základné časti aplikácie, ktoré sú popísané v nasledujúcich kapitolách sú:

- Obrazovky a ich prepínanie
- Návod na obsluhu domácich spotrebičov
- Pripomínanie informácií
- Získavanie dát z webu
- Príjem a uloženie prijatých dát zo senzorov
- Vykonávanie pravidiel
- Grafické zobrazenie stavu a zvukový výstup
- Záznam akcií aplikácie a užívateľa a ich zobrazenie

3.3 Obrazovky a ich prepínanie

Aplikácia sa skladá z množstva obrazoviek a každá z nich slúži na určitý účel (zobrazenie počasia, spotrebičov...). Chovanie aplikácie je založené na prechode medzi týmito obrazovkami. Obrazovka je určená konfiguračným súborom. V ňom sa okrem iného nachádza aj obrázok, ktorý sa zobrazí s definovanými oblasťami. Po kliknutí na definovanú oblasť sa vykonáva príslušná akcia. Zoznam postupne načítavaných konfiguračných súborov sa ukladá do *Listu*, čo sa využíva pri prepínaní späť. Na načítanie novej obrazovky sa používa príkaz *move*. Príklad použitia je:

```
area: 0.50, 0.46; 0.63, 0.24
action: move - spotrebice/radio
```



Obrázok 3.2. Úvodná obrazovka s ilustračne zvýraznenými oblasťami. Po kliknutí na rôzne oblasti sa vyvolajú odlišné akcie, ktorým sa budem venovať v kapitole 4.1.1

Konkrétne príkazy budú podrobne popísané v kapitole 4.1.1. Tieto dva príkazy z konfiguračného súboru zabezpečujú, že po kliknutí do definovanej oblasti za príkazom *area* sa načíta nový konfiguračný súbor v adresári ¹ *spotrebice, radio.txt* ².

Po kliknutí na tlačidlo *ZPĚT* sa posledný súbor z *Listu* odstráni a načíta sa aktuálne posledný, čím sa načíta predchádzajúci konfiguračný súbor. Pri stlačení tlačidla *DOMŮ* sa List načítaných konfiguračných súborov vymaže a načíta sa pôvodná domovská obrazovka, ktorú je možno vidieť na obrázku 3.2.

3.4 Nápoveda k obsluhu domácich spotrebičov

Pre mnoho starsích ľudí je problematické zapamätať si, ako sa ovládajú rôzne prístroje v domácnosti. Preto aplikácia obsahuje nápovedu k vybranám spotrebičom.

V momentálne nakonfigurovanej aplikácii sa po kliknutí na pravé oko smajlíka zobrazí matica obrázkov jedlotlivých domácich spotrebičov. Načíta sa špeciálny konfiguračný súbor *Panel.txt*, popísaný v kapitole 4.1.1, v ktorom sú definované obrázky, ktoré sa zobrazia v matici spotrebičov. Oblasti kliknutia netreba konfigurovať manuálne, podľa počtu adresárov s domácimi spotrebičmi sa vypočítajú samé. Po kliknutí na konkrétny spotrebič sa načíta konfiguračný súbor.³

Adresár spotrebiče typicky obsahuje súbor *Panel.txt* na zobrazenie všetkých spotrebičov, aby si z nich senior vybral ten, s ktorým potrebuje poradiť.

Po kliknutí na spotrebič z matice obrázkov sa daný spotrebič zobrazí na celú obrazovku a aplikácia užívateľa vyzve, aby klikol na nejakú časť spotrebiča. Ak napríklad klikne na tlačidlo *volume +* na obrázku rádia, aplikácia vysvetlí ako a na čo sa dané tlačidlo používa. Jednotlivé oblasti spotrebičov a texty, ktoré sa prečítajú sa konfigurujú v súboroch príslušných k daným spotrebičom.

3.5 Pripomínanie informácií

Starší ľudia môžu mať aj problém so zapamätávaním si rôznych vecí, ktoré by cez deň mali splniť. Preto aplikácia obsahuje funkciu pripomienkovač. Pripomienkovač sa môže využívať napríklad na pripomínanie užívania liekov alebo upozornenie, že v televízii beží nejaký program. Dá sa použiť aj na pripomenutie narodenín či menín.

Na pripomínanie informácií sa využíva konfiguračný súbor *alarm.txt*. V tomto konfiguračnom súbore každý riadok definuje jednu pripomienku. Štruktúra konfiguračného súboru na alarm je nasledovná:

- deň v týždni alebo konkrétny dátum, v ktorý alarm bude pripomínať
- čas alarmu
- text pripomienky

V prípade, že k alarmu sa má zobraziť aj nejaký obrázok (napríklad pri pripomienke televíznej relácie obrázkov televízie) použije sa príkaz *showImage*, ktorý je popísaný v kapitole 4.1.1. Príklad konfigurácie alarmu je:

```
alarm: streda - 13:40 - dej si prasek
```

¹ Adresáre písané v konfiguračnom súbore za príkazom *move* neobsahujú celú cestu k súboru, iba časť cesty odlišnú od tej, na ktorej je uložená samotná aplikácia.

² Koncovka *.txt* sa pri príkaze *move* nepíše

³ Ten musí mať rovnaké meno ako obrázok v *Panel.txt*, teda ak sa zobrazuje obrázok s cestou *spotrebice/vysavac/vysavac.png*, tak po kliknutí na vysávač sa vykoná príkaz *move* a cieľ je konfiguračný súbor *spotrebice/vysavac/vysavac.txt*

3.6 Časti používajúce internet

Viacero informácií sa sťahuje z webových stránok z dôvodu, že dané dáta sa menia.

3.6.1 Funkcionalita zábava

Funkcionalita zábava obsahuje viacero možností oddychu a zabavenia sa pre seni-ora. V časti zábava si môže vypočúť vtipy, pozrieť obrázkové vtipy, zistiť stručný obsah práve hraných filmov v kine. Taktiež si bude môcť pozrieť recepty alebo pustiť prehrávač hudby. Princíp fungovania všetkých spomenutých funkcií je z veľkej časti rovnaký.

Z webovej adresy¹ sa v programe stiahne súbor *Info.txt*. Tento súbor obsahuje texty, ktoré sa zobrazia v aplikácii a mená ďalších adresárov. Info súbor v hlavnom adresári zábava vyzerá nasledovne:

```
Vtipy a príhody|Vtipy
Obrázkové Vtipy|VtipyObrázky
Kino a diavadlo|KinoDivadlo
Recepty na jídlo|Recepty
Hudba a piesne|HudbaPisne
```

Časť pred oddeľovacím znakom (zvislá čiara) určuje text, ktorý sa zobrazí v apli-kácii na tlačidlách. Časť za oddeľovacím znakom definuje ďalšie adresáre na serveri. Rovnako pomenované adresáre sa vytvoria aj v zariadení, kde je aplikácia. Z daných adresárov na serveri sa stiahnu Info súbory, v ktorých sú nasledujúce údaje:

```
UPDATE 2019-03-23 15:16:17
NUMBERS 3001-3005
```

Za kľúčovým slovom *UPDATE* sa nachádza dátum a čas, kedy boli dáta v adresári aktualizované. V prípade, že dátum a čas aktualizácie v Info súbore na serveri je aktuálnejší ako v stiahnutom Info súbore v zariadení, jednotlivé súbory sa stiahnu opäť.

Za kľúčovým slovom *NUMBERS* sa nachádza interval čísel definujúci mená súbo-rov.

Teda v tomto adresári sa na serveri nachádzajú súbory *3001.inf*, *3002.inf*, ..., *3005.inf*.

Všetky tieto súbory sa aj stiahnu. Súbory s koncovkou *.inf* obsahujú na prvom riadku text, ktorý sa zobrazí v aplikácii na tlačidlách. Na druhom riadku je napísaný prislúchajúci súbor k danému textu. Napríklad obsah súboru *3004.inf* je:

```
So Will I - 100 Billion X
3004.mp3
```

Po kliknutí na definovanú časť smajlíka sa zobrazí panel tlačidiel s textami defino-vanými v Info súbore. Aplikácia po kliknutí napríklad na tlačidlo s textom *Hudba a Písne* vyzerá obrazovka v aplikácii nasledovne:

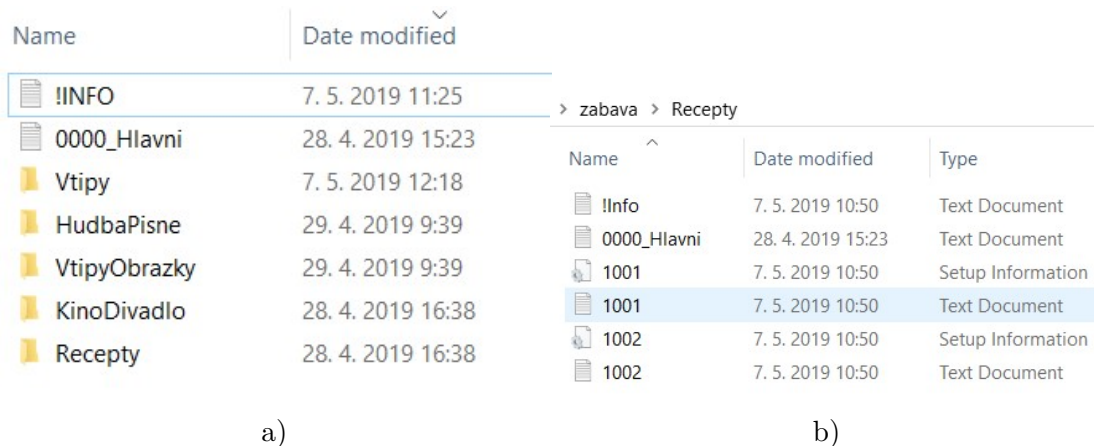
Pri sťahovaní sa do zariadenia stiahnu aj súbory z druhého riadku *inf* súborov, v tomto prípade *mp3* súbory. Keby sa na druhom riadku *inf* súboru nachádzali obrázky alebo texty, stiahli by sa a zobrazili by sa v aplikácii.

V každom adresári zábavy v počítači musí byť aj súbor *0000_Hlavni.txt*, v ktorom sa definuje pomocou príkazov popísaných v Manuáli v kapitole 4.1.1 pozadie alebo text, ktorý sa vysloví po kliknutí na jednotlivé tlačidlá. Tento súbor v jednotlivých adresároch musí byť.

¹ <https://nit.felk.cvut.cz/~dark/Projekty/HomeComp/Zabava/>



Obrázok 3.3. Na tlačidlách sa zobrazia texty z prvých riadkov jednotlivých *inf* súborov. Po kliknutí na konkrétnu pieseň sa pesnička spustí.



Obrázok 3.4. Siahnuté súbory a vytvorené adresáre v počítači

3.6.2 Predpoveď počasia

Ďalšou užitočnou vlastnosťou aplikácie je predpoveď počasia (spolu s grafickou ilustráciou počasia) na 3 dni, vrátane aktuálneho dňa.

Mesto, pre ktoré sa sťahuje počasie sa vyčíta z konfiguračného súboru pre počasie. Počasie sa sťahuje zo servera `OpenWeatherMap`¹, poskytovaná predpoveď je na 5 dní, v trojhodinových intervaloch. Stiahnuté dáta sú vo formáte *json*. Príklad dát stiahnutých z `OpenWeatherMap`² je:

```
{
  "cod": "200", //interný parameter
  "message": 0.0118, //interný parameter
  "cnt": 40, //počet predpovedí
  "list": [ //predpoveď je uložená v liste
    {
      "dt": 1557684000, //čas a deň predpovede
      "main": {
        "temp": 8.08, //teplota [st. C]
        "temp_min": 8.08, //minimálna teplota [st. C]
        "temp_max": 9.17, //maximálna teplota [st. C]
        "pressure": 1024.52, //atmosfer. tlak na úrovni mora [hPa]
        "sea_level": 1024.52, //atmosfer. tlak na úrovni mora [hPa]
        "grnd_level": 1002.58, //atmosfer. tlak na úrovni zeme [hPa]
        "humidity": 91, //vlhkosť vzduchu [%]
      }
    }
  ]
}
```

¹ <https://home.openweathermap.org/>

² <http://api.openweathermap.org/data/2.5/forecast?q=Bratislava&appid=1df76ae57bedf19bd5540a2b4b9cd0df&units=metric&lang=cz>

```

    "temp_kf": -1.09          //interný parameter
  },
  "weather": [
    {
      "id": 501,              //id počasia
      "main": "Rain",        //parameter počasia
      "description": "déšť", //popis počasia
      "icon": "10d"          //ikonka k danému počasiu
    }
  ],
  "clouds": {
    "all": 99                //oblačnosť [%]
  },
  "wind": {
    "speed": 5.76,           //rýchlosť vetra [m/s]
    "deg": 338.125          //smer vetra [stupne]
  },
  "rain": {
    "3h": 3.875              //spadnuté zrážky za 3 hodiny [mm]
  },
  "sys": {
    "pod": "d"               //interný parameter
  },
  "dt_txt": "2019-05-12 18:00:00" //dátum a čas predpovede
  ...

```

Pri sťahovaní dát sa pri kľúčovom slove *units* dá nastaviť, v akých jednotkách sa majú dáta stiahnuť. V tomto prípade sa sťahujú v metrických jednotkách. Pri kľúčovom slove *lang* sa nastaví, v akom jazyku sa majú stiahnuť popisy k počasiu. Pri kľúčovom slove *appid* je API kľúč, ktorý som získal po vytvorení konta na OpenWeatherMap¹, predpoveď, ktorá sa sťahuje do aplikácie je zadarmo.

Vyčítanie dát z json súboru má nastarosti funkcia `GetForecast(string city)`, ktorá vyzerá nasledovne:

```

public weatherForecast GetForecast(string city)
{
    using (WebClient web = new WebClient())
    {
        //adresa, aj s mestom, odkiaľ sa sťahuje predpoveď
        string url = string.Format
            ("http://api.openweathermap.org/data/2.5/forecast?
             " + "q={0}&appid=1df76ae57bedf19bd5540a2b4b9
             cd0df&units=metric&lang=cz", city);
        var json = web.DownloadString(url);
        //stiahnuté dáta sú vo formáte json, treba ich správne
        //interpretovať
        var Object =
            JsonConvert.DeserializeObject<weatherForecast>(json);
        weatherForecast forecast = Object;
        //funkcia vráti instanciu triedy WeatherForecast
        return forecast;
    }
}

```

¹ openweathermap.org

}

3.6.3 Volanie cez Skype

Ako bolo spomenuté v úvode 1.1, až 32,4% ľudí v Česku nad 65 rokov žije bez ľudskej spoločnosti. Komunikácia s ostantými ľuďmi je pre nich veľmi cenená. Pre komunikáciu s možnosťou videohovoru je v aplikácii použitý Skype.

Na použitie Skypu je potrebná knižnica *SKYPE4COMLib*. Pomocou funkcií z nej sa uskutoční hovor cez Skype. Je dôležité poznamenať, že pomocou C# už nie je možné ovládať najnovší skype od Microsoftu. Preto bolo potrebné stiahnuť starú a funkčnú verziu 7.41 a odmietiť každý pokus skypu o aktualizáciu. Hovor sa uskutoční nasledujúcim kódom:

```
Skype skype = new Skype();
//ak skype nie je zapnutý, zapni ho
if (!skype.Client.IsRunning)
{
    skype.Client.Start(Minimized: true, Nosplash: true);
}
//zavolaj kontaktu, contact je užívateľske meno zo skypu
Call call = skype.PlaceCall(contact);
```

3.7 Využitie externých senzorov

Senzory majú za úlohu monitorovať domáce prostredie. Okrem monitorovacej úlohy majú aj bezpečnostnú funkciu. Senzory slúžia na meranie teploty, meranie hodnoty oxidu uhličitého v ovzduší alebo ako senzor zatvorených dverí či okien. V budúcnosti sa môže použiť aj senzor tlaku na detekciu prítomnosti osoby v kresle alebo pohybový senzor na detekciu pohybu.

3.7.1 Senzory pre demonštráciu

Pre demonštráciu bol vedúcim práce vytvorený prototypový HW obsahujúci niekoľko senzorov.

Senzory sú vytvorené ako dve samostatné zariadenia s názvami *Prvý* a *Druhý*. Pre experiment sú do PC pripojené pomocou USB. Jednotlivé zariadenia obsahujú tieto senzory:

■ Prvý

- Čítačka NFC kariet - umožňuje po priložení NFC karty rozpoznať o akú kartu sa jedná a na základe pravidiel sa vykoná daná akcia.
- Pohybový senzor - monitoruje pohyb dôchodcu
- Výstupná zvuková signalizácia - pri vyvolaní určitej akcie v pravidlách sa môže pre upozornenie dôchodcu zapnúť zvuková signalizácia

■ Druhý

- Dva teplomery - slúžia na meranie vnútornej a vonkajšej teploty
- Tlačítko slúžiace na simuláciu otvorených dverí alebo okna
- Senzor koncentrácie CO_2 v ovzduší - v prípade zvýšenej hladiny je dôchodca upozornený na túto skutočnosť

3.7.2 Príjem a uloženie prijatých dát zo senzorov

Aplikácia je vytvorená tak, aby bola schopná prijímať dáta zo senzorov pripojených k počítaču ľubovoľným spôsobom. Aj keď sa ako hlavný spôsob príjmu dát zo senzorov predpokladá Bluetooth a WiFi, tak v tejto prototypovej aplikácii bolo (hlavne pre ladenie a demonštráciu) využité pripojenie pomocou USB portu. Jednoduchý experimentálny HW obsahujúci niekoľko senzorov (popísaných v 3.7.1) poskytol vedúci práce. Pre príjem dát z aplikácie boli vytvorené triedy, kam sa tieto dáta ukladajú. Súčasťou práce sú iba triedy pre uloženie týchto dát, nie na ich prenos a príjem. K tomuto účelu slúžia príslušné konfiguračné súbory.

Konfiguračný súbor senzorov obsahuje:

- štvormiestne číslo identifikujúce senzor
- definícia zariadenia
- názov senzoru pre ďalšie použitie
- prípadne definícia typu a hodnoty premennej, ktorú vysiela senzor
- môže obsahovať riadok s komentárom začínajúci výkričníkom

Príklad konfiguračného súboru senzorov je:

```
! ctecka NFC karet
1234;Prvy;KartaNFC#EVENT
! pohybove PIR cidlo
1234;Prvy;Pohyb
```

Vlastná časť aplikácie zahŕňajúca spracovanie a ukladanie dát zo senzorov zahŕňa tieto časti:

- Aplikácia obsahuje triedy *HWDevice* obsahujúce informácie a dáta o jednom pripojenom zariadení obsahujúcom aj niekoľko senzorov a triedu *HWsensor* nesúcu informácie a dáta o jednom senzore obsiahnutom v skôr spomenutom zariadení. Tieto sú pri spustení aplikácie vytvorené podľa vstupných konfiguračných súborov
- Triedy pre vyzdvihnutie prijatých dát z externých senzorov. V súčasnosti je implementovaná trieda *InputUSBHID* pre vyzdvihnutie dát zo senzorov pripojených pomocou USB a emulujúcich klasické HID zariadenie (nie sú potrebné špeciálne ovládače).
- Dáta prijaté z externých senzorov sú vkladané do fronty v aplikácii. Dáta nie sú okamžite pri prijatí spracovávané, aby nemohlo prísť k zahlteniu aplikácie. Zatiaľ je implementovaný veľmi jednoduchý algoritmus zabráňujúci viacnásobnému uloženiu rovnakých dát do tejto vstupnej fronty.
- V dobe medzi vykonávaním pravidiel sú dáta z externých senzorov z tejto internej fronty aplikácie vyzdvihované a spracovávané. Každý paket obsahuje: identifikáciu projektu, sériové číslo zariadenia, zoznam dát zo senzorov a kontrolný súčet.
- Postup spracovania dát je nasledujúci
 - Test, či sa jedná o dáta pre tento projekt. Ak nie, dáta sú zahodené.
 - Test, či je zariadenie (podľa sériového čísla) povolené v konfigurácii. Ak nie, dáta sú zahodené
 - Postupné vyzdvihnutie dát všetkých senzorov z paketu a ich uloženie do príslušnej triedy *HWsensor*. Dáta zo senzorov obsiahnuté v pakete sú úmyselne komprimované.
 - Pokiaľ senzor obsahuje príznak *EVENT*, tak je tento príznak uchovaný. Vďaka tomuto príznaku sa pravidlá spustia oveľa rýchlejšie ako je ich interval a tým sa zabezpečí rýchla obsluha hodnoty z daného senzoru.

- Pokiaľ je v senzore nastavený príznak *OUT* znamenajúci, že senzor je výstupný, tak príslušná trieda prijímajúca dáta zo senzoru zaistí vysielanie tejto hodnoty do zariadenia obsahujúceho senzor.

V nasledujúcej kapitole bude popísané spracovanie pravidiel. V pravidlách sú použité nielen konštanty ale aj premenné. Pravidlá sú spúšťané periodicky, obdobne ako riadiaci automat. Vďaka tomu sa môžu vytvárať zložité pravidlá a sofistikované chovanie aplikácie. Konštanty a premenné sú vytvorené v podstate rovnako ako zariadenie so senzormi. Jediný rozdiel spočíva v tom, že ich hodnoty nie sú nastavované z externých senzorov ale z konfiguračných súborov pre konštanty a premenné, ktoré budú popísané neskôr. Z hľadiska interpretera pravidiel je jedno, či sa hodnota berie zo skutočného senzoru alebo nejakého pomyselného (umelo vytvoreného senzoru) obsahujúceho v podstate konštanty alebo všeobecné premenné. Takto bolo možné kód tvoriaci interpreter pravidiel veľmi zjednodušiť. Zariadenia predstavujúce konštanty alebo premenné možno pomenovať ľubovoľne, je však nutné dodržať, aby sériové číslo bolo *0000h*.

Konfiguračný súbor pre konfiguráciu konštánt obsahuje:

- sériový identifikátor *0000h*
- kľúčové slovo *Konstanty*, aby bolo jasné, že sa jedná o konštanty
- názov konštanty
- dátový typ
- hodnota konštanty

Konfiguračný súbor pre konfiguráciu premenných obsahuje:

- sériový identifikátor *0000h*
- kľúčové slovo *Premenne*
- názov premennej
- dátový typ

Príklady konfigurácie sú v 4.1.1

3.7.3 Pravidlá

Jednou z najviac prínosných častí aplikácie je možnosť tvorby a vykonávanie tzv. pravidiel. Tieto pravidlá v podstate vytvárajú chovanie systému ako odozvu na hodnoty prichádzajúce zo senzorov a rovnako reflektujú odozvy užívateľa. Pravidlá sú napísané v ľubovoľnom počte súborov, rozdelených napr. podľa účelu a skladajú sa z nasledujúcich blokov:

- Podmienka - obsahuje súbor pravidiel, ktoré sú testované a na vykonanie nasledujúceho bloku *Vykonane* musia byť splnené. Každé pravidlo je na jednom riadku a je medzi nimi logická operácia AND.
- Vykonanie - obsahuje súbor pravidiel, ktoré sa vykonajú po splnení pravidiel v bloku *Podmienka*

Takýchto blokov môže pravidlový súbor obsahovať ľubovoľný počet.

Povolené operácie sú =, <>, <, >, <=, >=. Na ľavej strane porovnania môžu byť veličiny senzorov a premenné. Na pravej strane môžu byť veličiny senzorov, premenné a konštanty.

Blok *#THEN* obsahuje na samostatných riadkoch akcie. Môže obsahovať akciu priradenia, sčítania alebo odčítania.

Za špeciálnym znakom @ sa definujú špeciálne povely, ktoré sa spracujú v aplikácii. Štruktúra takového príkazu je:

- špeciálny znak @
- následuje kľúčové slovo (napríklad *INFO* - zobraz informáciu)
- stredníkom oddelené parametre pre povel

Momentálne sú definované tieto špeciálne povely:

- **INFO** - spracovanie informácie pre zobrazenie
 - **ADD** - pridanie informácie pre zobrazenie, parametre sú kľúč, úroveň (dôležitosť informácie), text do záhlavia a text informácie
 - **OUT** - odstránenie informácie zo zobrazenia, parameter je kľúč
 - **CLR** - vymazanie všetkých informácií zo zobrazenia
- **INTERNET-RADIO** - ovládanie internetového rádia
 - **PLAY** - spustenie internetového rádia, parameter je názov stanice
 - **STOP** - vypnutie internetového rádia, bez parametrov

Príklad pravidlového súboru je:

```
! začalo zvýšené CO2
#IF
  ! ak je koncentrácia vyššia než stanovená hodnota
  Druhy.VzduchCO2.Value > 60
  ! ešte nie je aktívne varovanie
  Promenne.VzduchCO2VarovanieAno.Value = FALSE
#THEN
  ! aktívne varovanie
  Promenne.VzduchCO2VarovanieAno.Value = TRUE
  ! nastavenie času pre budúce varovanie - hneď
  Promenne.VzduchCO2VarovanieCas.Value = [DateTimeNow]
#END
```

3.8 Hlasový výstup

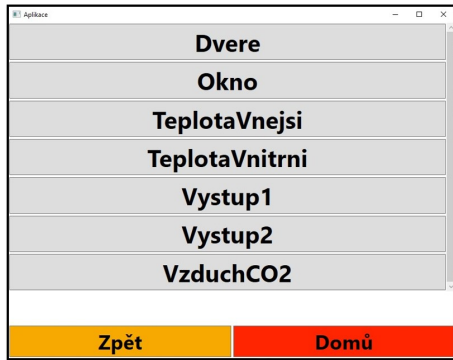
Je vhodné, aby aplikácia na užívateľa aj rozprávala v jeho materinskom jazyku.

Na hlasový výstup je využitá knižnica *System.Speech*. V C# však český hlas nebolo možné vybrať, aj keď bol nastavený ako jazyk systému. Pre odblokvanie hlasu bola použitá aplikácia stiahnutá z [14]. Po odblokovaní sa už v C# český hlas dal nastaviť. Konfigurácia textov, ktoré má aplikácia prečítať je popísaná v kapitole 4.1.1.

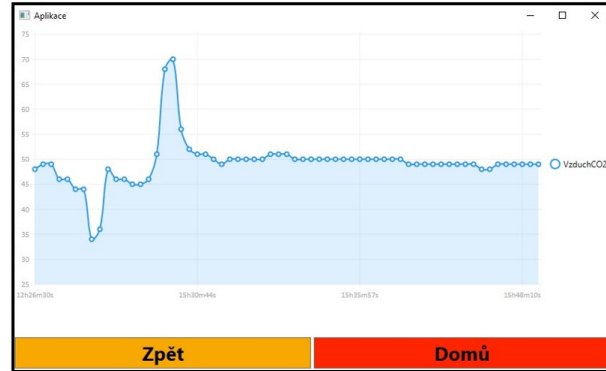
3.9 Záznam akcií aplikácie a užívateľa a ich zobrazenie

Veľmi vhodnou požiadavkou je záznam činnosti. V aplikácii je implementovaný ako graf, kde sa na časovej ose zobrazujú dáta prijaté z externých senzorov.

V adresárovej štruktúre je adresár, do ktorého sa ukladajú prichádzajúce dáta zo senzorov. Ďalej sa tam nachádza konfiguračný súbor *grafy.txt*, ktorý nastaví pozadie a prípadný text pre danú obrazovku. Po presmerovaní na obrazovku *grafy* sa podľa [15] zobrazia grafy zobrazujúce hodnoty pre vybraný senzor. Tieto dáta by mali slúžiť prevažne pre obsluhu zariadenia.



Obrázok 3.5. Výber nakonfigurovaného senzoru



Obrázok 3.6. Zobrazenie dát pre vybraný senzor

Kapitola 4

Vzorová aplikácia, jej konfigurácia a činnosť

V rámci bakalárskej práce bola vytvorená vzorová aplikácia, ktorá slúži ako východzí bod pre neskoršiu vlastnú konfiguráciu. Práve na tejto aplikácii sa dá vysvetliť ako sa tvoria jednotlivé konfiguračné súbory, senzory, príslušné pravidlá atď.

4.1 Základná konfigurácia

Prispôsobenie aplikácie seniorovi sa skladá zo viacerých častí: konfigurácia obrazoviek, konfigurácia senzorov, konštant, premenných a vytvorenie pravidiel. Všetky tieto časti budú postupne popísané v tejto kapitole.

Väčšina konfiguračných súborov obrazoviek až na výnimky, obsahuje definovanú oblasť (kľúčové slovo `area`). Za riadkom s kľúčovým slovom `area` môže nasledovať ľubovoľné množstvo akcií, ktoré sa vykonajú po kliknutí na príslušnú oblasť.

Je dôležité dodržiavať oddelujúce znaky ako dvojbodka (:), pomlčka (-) v správnom poradí ako je ukázané v tomto manuáli.

4.1.1 Kľúčové slová

Kľúčové slová sa používajú pri konfigurácii obrazoviek. V tejto podkapitole je prehľad kľúčových slov aj s príkladmi použitia.

- **img:** Za kľúčovým slovom sa napíše cesta (iba cesta od adresy, kde je uložená aplikácia, nie absolútna cesta), kde je obrázok uložený s jeho menom a aj koncovkou definujúcou typ súboru. Príklad použitia je:

```
img: spotrebice/radio/TlacidloHlasitosti.jpg
```

- **area:** Obsahuje 4 čísla ako 2 usporiadané dvojice. Bod kliknutia je definovaný x-ovou a y-ovou súradnicou. Kvôli rôznym veľkostiam okna, v ktorom je aplikácia zobrazená, je v konfiguračnom súbore tento bod vyjadrený pomerom k celkovej veľkosti okna, teda v percentách. Vďaka tomu nedochádza k deformáciám oblastí kliknutia. Prvé dve čísla oddelené čiarkou (,) definujú ľavý krajný bod ohraničujúci obdĺžnikovú oblasť. Dvojica je oddelená od druhej dvojice bodkočiarkou (stredníkom ;). Druhá dvojica desatinných čísel (desatinné čísla sú oddelené bodkou(!)) definuje pravý horný bod ohraničujúci oblasť. Príklad správne napísanej oblasti je:

```
area: 36, 54; 50, 18
```

Na získanie čísel definujúcich oblasti bol vytvorený demo program, kde po zadaní obrázka do konfiguračného súboru sa obrázok zobrazí, a po kliknutí naň sa zobrazia aj čísla definujúce kliknutý bod. Tieto čísla potom môžu byť použité v konfigurácii aplikácie. Je dôležité, aby sa oblasti napísané v konfiguračnom súbore neprekrývali, pretože by nebolo možné rozoznať, do ktorej sa kliklo.

- **action:** Toto kľúčové slovo znamená, že sa bude vykonávať určitá akcia. Príklad použitia kľúčového slova akcia je

action: príkaz - destinácia

Príkazy sú vysvetlené v kapitole 4.1.2. Destinácia je adresa alebo dáta konkretizujúce príkaz. Aj destinácia je vysvetlená v kapitole 4.1.2.

- **default_action:** Štruktúra príkazu je rovnaká ako príkazu action. Rozdiel je v tom, že príkaz *action* sa spravidla vykonáva po kliknutí do danej oblasti. Príkaz *default_action* sa vykonáva hneď po načítaní konfiguračného súboru.
- **end:** Definuje koniec konfiguračného súboru.
- **alarm:** Toto kľúčové slovo sa vzťahuje ku konfiguračnému súboru pre alarmy a slúži na pripomenutie pravidelne sa opakujúcich udalostí. Každý riadok definuje jeden alarm. Štruktúra konfiguračného súboru na alarm je nasledovná:
 - deň v týždni alebo konkrétny dátum, v ktorý alarm bude pripomínať
 - čas alarmu
 - text pripomienky

V prípade, že k alarmu sa má zobrazíť aj nejaký obrázok (napríklad pri pripomienke televíznej relácie obrázkov televízie) použije sa príkaz `showImage`, ktorý je popísaný v 4.1.3. Príklad konfiguračného súboru pre alarmy:

```
alarm: pondeli - 13:30 - Televize
alarm: 15.4. - 10:00 - Dcera narodeniny, zavolej ji
action: showImage - fotky/Dcera.jpg
```

4.1.2 Príkazy

Príkazy slúžia na špecifikáciu akcie. Vďaka príkazom aplikácia vie, akú akciu má vykonať. V nasledujúcom texte je vysvetlenie a ukážka používania jednotlivých príkazov rozdelených podľa účelu.

Všeobecné príkazy

- **move:** Tento príkaz načíta nový konfiguračný súbor so všetkými jeho akciami. Za príkazom `move` sa napíše cesta k textovému súboru (ktorý sa píše bez koncovky `.txt`). Teda podľa príkladu sa načíta súbor z adresára `skype` pomenovaný `skype.txt`

```
action: move - skype/skype
```

- **read:** Aplikácia prečíta obsah textového súboru napísaného za týmto príkazom. K textovému súboru treba napísať cestu od adresy, kde je uložená aplikácia. Príklad použitia:

```
default_action: read - spotrebice/radio/info.txt
```

- **readThis:** Aplikácia prečíta text napísaný za týmto príkazom. Príklad použitia:

```
action: readThis - Venku je hezky, nechceš se jít trochu projít?
```

- **showImage:** príkaz využívajúci sa v konkrétnych situáciách. Môže sa použiť:
 - V konfiguračnom súbore `skype` ako obrázok (fotografia kontaktu), ktorá sa zobrazí na obrazovke
 - V konfiguračnom súbore pre alarmy ako obrázok zobrazujúci sa pri texte k alarmu. (Pri pripomenutí televízneho program sa zobrazí obrázok televízie.) Príkaz `showImage` sa napíše na nasledujúci riadok po konfigurácii alarmu

Ako adresa obrázku sa píše cesta k obrázku od adresy, v ktorej je uložená aplikácia. Príklad použitia:

```
action: showImage - skype/phone.png
```

Príklad konfiguračného súboru pre spotrebič rádio je:

```
! obrázok danej obrazovky
img: spotrebice/radio/Hlavni.jpg
! hneď pri načítaní tohto konfig. súboru sa prečíta daný text
default_action : readThis - Klikni na nejakej knoflík a poradím ti.
! oblasť kliknutia
area: 60, 85; 74, 70
! po kliknutí do oblasti sa spracuje iný konfiguračný súbor
action: move - spotrebice/radio/tuning
area: 55, 74; 60, 68
action : move - spotrebice/radio/vol_plus
area: 48, 74; 54, 66
...
end:
```

Po kliknutí do oblasti, v ktorej leží tlačidlo *volume +* sa načíta konfiguračný súbor *vol_plus.txt*:

```
!obrázok danej obrazovky
img: spotrebice/radio/vol_plus.jpg
!hneď po načítaní konfig. súboru sa prečíta text v definovanom súbore
default_action: read-spotrebice/radio/info.txt
end:
```



Obrázok 4.1. Ilustračné zobrazenie oblastí na obrazovke rádia



Obrázok 4.2. Obrazovka vyššie popísaného konfiguračného súboru, načítaná po kliknutí na jedno z tlačidiel

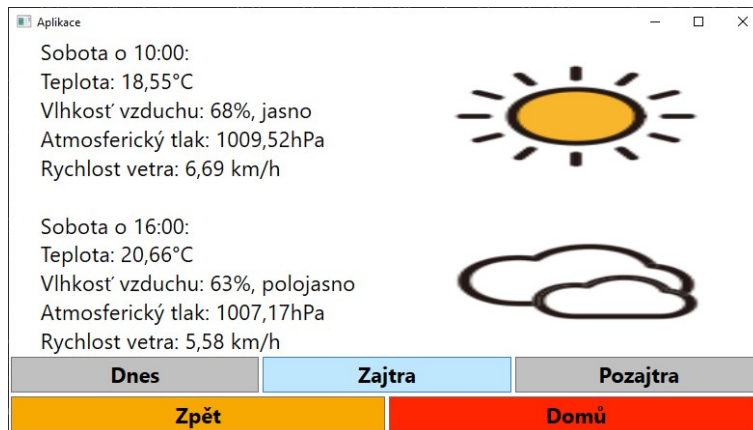
Príkazy vzťahujúce sa k funkcii predpoved' počasia

Predpoveď počasia sa v aplikácii zobrazuje na 3 dni. Na správne fungovanie predpovede stačí nakonfigurovať iba jeden príkaz *weatherCity*, a to následovne.

- **weatherCity** Za príkazom sa napíše mesto, pre ktoré sa bude v aplikácii zobrazovať predpoveď. Či pre dané mesto existuje predpoveď na webe odporúčam overiť na stránke¹. Ak sa pre požadovanú obec predpoveď neposkytuje, treba skúsiť okresné mesto. Príklad použitia:

```
action: weatherCity - Praha
```

¹ <https://openweathermap.org/>



Obrázok 4.3. Obrazovka počasia s predpoveďou na 3 dni

Príkazy vzťahujúce sa k funkcii privolania pomoci

Na domovskej obrazovke sa nachádza záchranársky kríž. Po kliknutí naň a potvrdení, že pomoc naozaj potrebuje sa odošle email požadujúci pomoc. Aby sa email skutočne odoslal, je potrebné použiť všetky nasledujúce príkazy (príkaz *confirm* je dobrovoľný ale doporučený). V tomto konfiguračnom súbore sa nepoužíva kľúčové slovo *area*. Všetky dáta sa použijú v programe po načítaní tohto konfiguračného súboru.

- **confirm:** Príkaz slúži, ak sa užívateľ napríklad preklikne a v skutočnosti pomoc nepotrebuje. Príkaz sa použije s kľúčovým slovom *default.action*. Za príkazom sa napíše text, ktorý sa zobrazí spolu s možnosťami Áno a Nie. Po kliknutí na áno sa pomoc zavolá, po kliknutí na nie sa nezavolá. Príklad použitia:

```
default_action: confirm -Skutočne chceš|zavolať|pomoc?
```

Oddelovací znak zvislej čiary slúži na odriadkovanie.

- **odosielatel_mail:** Za kľúčové slovo sa napíše emailová adresa, z ktorej sa mail odošle. Príklad:

```
action: odosielatel_mail - testbak@zoznam.sk
```

- **odosielatel_host:** Za týmto príkazom sa definuje takzvaný server host name pre odchádzajúcu poštu, SMTP Mail server, ktorý si obsluha musí vyhľadať (napríklad na stránke¹), prípadne inde. Príklad použitia:

```
action: odosielatel_host - smtp.zoznam.sk
```

- **odosielatel_port:** Definuje sa port SMTP Mail servera, rovnako ako server host name ho treba vyhľadať na internete.

```
action: odosielatel_port - 587
```

- **odosielatel_heslo:** Za týmto príkazom sa napíše heslo do emailovej schránky, z ktorej odchádza mail. Príklad použitia:

```
action: odosielatel_heslo - TajneHeslo123
```

- **prijimatel:** Mailová adresa príjemcu. Príklad použitia:

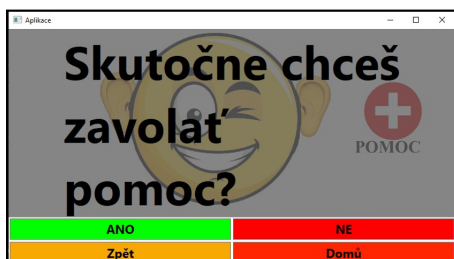
```
action: prijimatel - Janka@gmail.com
```

¹ <https://www.serversettings.email/index.html>

- **text:** Text, ktorý sa napíše ako telo emailu. Príklad použitia:

```
action: text - Potřebuji pomoc, přijdi nebo zavolej.
```

Po kliknutí na záchranársky kríž na domovskej obrazovke (obrázok 3.2), sa zobrazí obrazovka 4.4, ktorá sa, kvôli možnému omylu musí potvrdiť. Po potvrdení sa podľa konfiguračného súboru odošle email a zobrazí sa obrazovka 4.5



Obrázok 4.4. Potvrdzovacia obrazovka pre volanie pomoci



Obrázok 4.5. Obrazovka po potvrdení úmyslu zavolať pomoc.

Príkazy vzťahujúce sa k funkcii volania cez skype

Ak by dôchodca chcel zatelefonovať svojim blízkym alebo priateľom, môže na to využiť aplikáciu ovládanú skype. Na konfiguráciu skypu sa používajú nasledovné príkazy.

- **name:** Meno, ktoré sa zobrazí v aplikácii Príklad použitia:

```
action: name - Marekovi
```

- **confirm:** Na potvrdenie akcie sa môže (ale nemusí) použiť príkaz *confirm*. Po kliknutí na fotku užívateľa, ktorému chce senior telefonovať sa zobrazí potvrdzovacia obrazovka s definovaným textom. Po potvrdení sa kontaktu zavolá, po nepotvrdení sa prejde späť na obrazovku s kontaktami. Za príkazom *confirm* sa napíše text, ktorý sa zobrazí v potvrdzovacom okne. Pre odriadkovanie sa používa symbol zvislej čiary. Príklad použitia:

```
action : confirm - Chceš|zavolať?
```

- **skype_call:** Užívateľské meno používateľa skypu, zistí sa priamo z aplikácie skype ako užívateľské meno. Príklad použitia:

```
action: skype_call - lmarek55
```

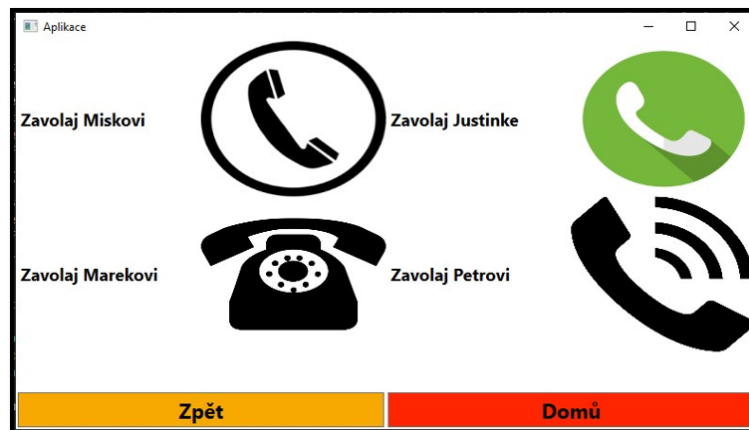
Konfiguračný súbor pre skype môže vyzeráť:

```
!pozadie domovskej obrazovky (čisto biele)
img: white.png
!oblasť vľavo hore
area: 0, 50; 50, 0
meno, ktoré sa zobrazí na obrazovke
action: name - Miškovi
!obrázok daného kontaktu
action: showImage - skype/phone.png
!voliteľný príkaz vyžadujúci potvrdenie úmyslu volať danému kontaktu
action : confirm -Skutočne|chceš|zavolať|cez|skype
!užívateľské meno v skype
action: skype_call - michal.pechnik
!vpravo dole
```

```

area: 50, 100; 100, 50
action: name - Petrovi
action: showImage - skype/phone4.png
action: skype_call - darkagentpetr
...
end:

```



Obrázok 4.6. Obrazovka pre skype. Po kliknutí do definovanej oblasti sa kontakt pomocou aplikácie skype vytočí. Miesto aktuálnych obrázkov telefónov sa môžu nakonfigurovať napríklad fotografie kontaktov.

4.1.3 Zobrazenie viacerých spotrebičov súčasne

Pre tento účel slúži špeciálny konfiguračný súbor *Panel.txt*. Tento konfiguračný súbor sa od ostatných líši, má fixne dané meno a je uložený v adresári spotrebiče. Príklad konfiguračného súboru *Panel.txt*:

```

default_action: read - spotrebice/info.txt
img: spotrebice/mikrovlanka/mikrovlanka.jpeg
img: spotrebice/truba/truba.jpg
...
end:

```

V tomto súbore sú definované iba *default_action* a obrázky, ktoré sa zobrazia v Paneli spotrebičov. Oblasti kliknutia netreba konfigurovať manuálne, podľa počtu adresárov s priečkami sa vypočítajú samé. Po kliknutí na konkrétny spotrebič sa načíta konfiguračný súbor (ktorý musí mať rovnaké meno ako obrázok v *Panel.txt*, teda ak sa zobrazuje obrázok *spotrebice/vysavac/vysavac.png*, tak po kliknutí na vysávač sa vykoná príkaz *move* a cieľ je konfiguračný súbor *spotrebice/vysavac/vysavac.txt*) Adresár spotrebiče typicky obsahuje súbor *Panel.txt* na zobrazenie všetkých spotrebičov, aby si z nich senior vybral ten, s ktorým potrebuje poradiť.



Obrázok 4.7. Zobrazenie všetkých nakonfigurovaných spotrebičov.

4.2 Umiestnenie adresárov a ich význam

Konfiguračné súbory sú pre prehľadnosť uložené v adresároch. Adresáre sú uložené tam, kde je uložený aj priečinok s aplikáciou. Vytvorenie a pomenovanie adresárov je ľubovoľné až na pár výnimiek.

Konfiguračný súbor určujúci pripomienky *alarm.txt* má fixné meno. Jeho uloženie vo vzorovej aplikácii je v adresári *alarm*.

Vo vzorovej aplikácii každý adresár konfiguruje jednu obrazovku v aplikácii. V každom adresári je teda aspoň hlavný konfiguračný súbor, ktorý konfiguruje základnú obrazovku. Konfiguruje sa v ňom najmä obrázok, ktorý sa zobrazí, text, ktorý sa prečíta, prípadne oblasti a akcie.

V nasledujúcej časti sú vymenované adresáre vo vzorovej aplikácii a obsah, čo sa v nich nachádza:

- alarm - konfiguračný súbor pre alarmy
- pomoc - konfiguračný súborov vzťahujúci sa k volaniu pomoci
- skype - súbory k obrazovke skype a konfiguračný súbor na ovládanie skypu
- spotrebiče - obsahuje podadresáre spotrebičov, ktoré obsahujú súbory pre dané obrazovky, adresár spotrebiče obsahuje aj konfiguračný súbor *Panel.txt*
- počasie - súbory nutné pre funkciu predpovedi počasia
- zábava - stiahnuté adresáre z internetu

4.3 Práca so senzormi

V demonštračnej aplikácii sú použité dve zariadenia so senzormi a s nimi niekoľko vzorových ukážok popísaných v nasledujúcich podkapitolách

4.3.1 Súbor senzorov

Senzory, ktoré komunikujú s aplikáciou sa definujú v príslušnom konfiguračnom súbore. Poznámka *#EVENT* znamená, že dáta zo senzoru sa spracujú čé najskôr po zmene hodnoty. Poznámka *#OUT* znamená, že dáta sa posielajú do zariadenia. Napríklad pre zvukový výstup sa posiela príkaz k jeho zapnutiu a vypnutiu. Príklad definície je:

```

!Sériové číslo; Názov zariadenia; Názov senzoru
! prvá skupina senzorov
! čítačka NFC kariet
1234;Prvy;KartaNFC#EVENT
! zvukový výstup
1234;Prvy;Buzzer#OUT&Bool&FALSE
! druhá skupina senzorov
! otváranie dverí a okna
5678;Druhy;Dvere#EVENT
5678;Druhy;Okno
! vnútorná a vonkajšia teplota
5678;Druhy;TeplotaVnutorna
5678;Druhy;TeplotaVonkajsia
! koncentrácia CO2
5678;Druhy;VzduchCO2
...

```

4.3.2 Súbor konštánt

Konštanty majú pevnú hodnotu po celý čas behu programu, nedajú sa meniť a slúžia na porovnávanie s inými dátami. Príklad konfigurácie konštánt je:

```

! Názov konštanty; Typ hodnoty; Hodnota
! Konštanty pre NFC karty
0000;Konstanty;NFCRadioOlympic&Bytes&95D2A843AC0000000000
0000;Konstanty;NFCRadioKiss&Bytes&0594A043720000000000
0000;Konstanty;NFCRadioEvropa2&Bytes&12345678901234567890
...

```

4.3.3 Súbor premenných

Premenné menia svoju hodnotu na základe dát alebo splnených podmienkach. Následne sa podľa premennej môže taktiež vykonať akcia. Spoločne s konštantami sa využívajú v pravidlách. Príklad konfigurácie premenných je:

```

! Názov premennej; Typ hodnoty; Východzia hodnota
! posledný známy stav dverí (otvorené = TRUE)
0000;Promenne;DverePoslednyStav&Bool&False
! či už prebieha varovanie o otvorených dverách
0000;Promenne;DvereVarovanieAno&Bool&False
! čas aktívneho zvukového signálu
0000;Promenne;DvereBuzzerCas&DateTime
! či hrá nejaké internetové rádio
0000;Promenne;IntRadio&Bool&False
...

```

4.3.4 Súbor pravidiel

Vďaka pravidlám sa dá reagovať na hodnoty a zmeny dát zo senzorov. Následujúci príklad definuje udalosť po priložení konkrétnej NFC karty. Po priložení NFC karty sa porovná jej hodnota s hodnotou uloženou v konštantách. Ak sa nájde zhoda, teda sa identifikuje priložená karta, vykoná sa akcia. V tomto prípade akcia je, že sa spustí dané rádio a premenná indikujúca hrajúce rádio sa nastaví ako pravdivá.


```

#IF
  ! prázdna podmienka je vždy splnená
#THEN
  ! predpoklad, že rádio má byť vypnuté
  Promenne.IntrRadio.Value = FALSE
#END

#IF
  ! pokiaľ je to táto karta
  Prvni.KartaNFC.Value = Konstanty.NFCRadioOlympic.Value
#THEN
  ! zapne sa definované rádio
  @INTERNET-RADIO Impuls
  ! premenná indikujúca hrajúce rádio
  Promenne.IntrRadio.Value = TRUE
#END

#IF
  ! pokiaľ je to táto karta
  Prvni.KartaNFC.Value = Konstanty.NFCRadioKiss.Value
#THEN
  ! zapne sa definované rádio
  @INTERNET-RADIO Kiss
  ! premenná indikujúca hrajúce rádio
  Promenne.IntrRadio.Value = TRUE
#END

#IF
  ! pokiaľ nie je priložená karta definujúca rádio
  Promenne.IntrRadio.Value = FALSE
#THEN
  ! ak hrá, vypne sa
  @INTERNET-RADIO STOP
#END
...

```

4.4 Ukážky varovaní na základe pravidiel

V pravidlách sa definujú podmienky, ktoré keď sú splnené dôjde k vyvolaniu udalosti. Následujúci pravidlový súbor kontroluje, či sú otvorené dvere. Pokiaľ sú zatvorené a aj boli zatvorené, pravidlový súbor sa ďalej nevykonáva. V tomto súbore sa počíta aj počet upozornení, aby sa po viacerých upozorneniach zobrazilo iné Info. Na počítanie slúži premenná *Promenne.DvereVarovaniePocet.Value*. Pokiaľ sú dvere zatvorené a predtým neboli tak sa vypne zvuková signalizácia a už sa v aplikácii nezobrazí Info. Pokiaľ sa dvere otvorili, nastaví sa čas upozornenia. Takisto sa nastavuje aj čas pre spustenie zvukovej signalizácie. Následne sa podľa počtu varovaní pridá príslušné Info, ktoré sa zobrazí v aplikácii.

```

! dlhodobo zatvorené dvere
#IF
  ! pokiaľ sú dvere zatvorené
  Druhy.Dvere.Value = FALSE
  ! a boli zatvorené

```

```

    Promenne.DverePoslednyStav.Value = FALSE
#THEN
    ! koniec vykonávania pravidiel pre tento súbor
    @RULES-END
#END

! dvere sa práve zatvorili
#IF
    ! dvere sú zatvorené
    Druhy.Dvere.Value = FALSE
    ! a predtým boli otvorené
    Promenne.DverePoslednyStav.Value = TRUE
#THEN
    ! tak posledný stav sa nastaví na zatvorené
    Promenne.DverePoslednyStav.Value = FALSE
    ! varovanie nie je
    Promenne.DvereVarovanieAno.Value = FALSE
    ! počet varovaní sa vynuluje
    Promenne.DvereVarovaniPocet.Value = 0
    ! vymažú sa hlásenia o dverách
    @INFO OUT;DVERE-VCHOD-A
    @INFO OUT;DVERE-VCHOD-B
    ! už netreba zvukovú signalizáciu
    Promenne.DvereBuzzerAno.Value = FALSE
    ! vypne sa zvuková signalizácia
    Prvy.Buzzer.Value = FALSE
#END

! dvere sa práve otvorili
#IF
    ! pokiaľ sú otvorené
    Druhy.Dvere.Value = TRUE
    ! a predtým boli zatvorené
    Promenne.DverePoslednyStav.Value = FALSE
#THEN
    ! posledný stav sa nastaví ako otvorené
    Promenne.DverePoslednyStav.Value = TRUE
    ! čas ďalšieho varovania
    Promenne.DvereVarovanieCas.Value = [DateTimeNow] + 00h00m30s
#END

! uplynul čas pre varovanie o otvorených dverách
#IF
    ! iba pokiaľ sú dvere otvorené
    Promenne.DverePoslednyStav.Value = TRUE
    ! a je čas pre varovanie
    Promenne.DvereVarovanieCas.Value < [DateTimeNow]
#THEN
    ! musí sa vykonať varovanie o otvorených dverách
    Promenne.DvereVarovanieAno.Value = TRUE
    ! musí byť aj zvukový signál
    Promenne.DvereBuzzerAno.Value = TRUE
    ! zapne se zvuková signalizácia

```

```

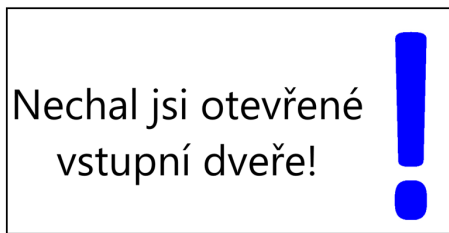
Hlavny.Buzzer.Value = TRUE
! čas pripomienky
Promenne.DvereBuzzerCas.Value = [DateTimeNow] + 00h00m10s
! zvýšenie počtu varovaní o otvorených dverách
Promenne.DvereVarovaniPocet.Value=
                                Promenne.DvereVarovaniPocet.Value+1
! čas budúceho varovania
Promenne.DvereVarovanieCas.Value = [DateTimeNow] + 00h00m30s
#END

! varovanie A že sú dvere otvorené
#IF
! ak je potreba varovať o otvorených dverách
Promenne.DvereVarovanieAno.Value = TRUE
! jedná sa o nízke varovanie
Promenne.DvereVarovaniPocet.Value < 3
#THEN
! vloženie varovania
@INFO ADD;DVERE-VCHOD-A;110;Dvere vchod;Nechal jsi otvorené|
    vstupní dveře!
! varovanie vykonané
Promenne.DvereVarovanieAno.Value = FALSE
#END

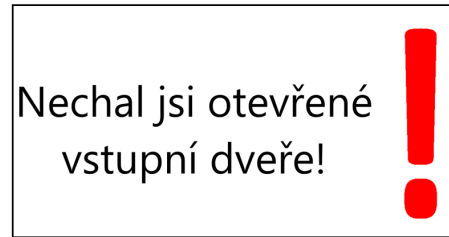
! varovanie B že sú otvorené dvere
#IF
! ak je potreba varovať o otvorených dverách
Promenne.DvereVarovanieAno.Value = TRUE
! jedná sa o vysoké varovanie
Promenne.DvereVarovaniPocet.Value >= 4
#THEN
! vloženie nového typu varovania
@INFO OUT;DVERE-VCHOD-A
@INFO ADD;DVERE-VCHOD-B;210;Dvere vchod;Nechal jsi otvorené|
    vstupní dveře!
! varovanie vykonané
Promenne.DvereVarovanieAno.Value = FALSE
#END

! zvuková varovná signalizácia
#IF
! pokiaľ je zvuková signalizácia
Promenne.DvereBuzzerAno.Value = TRUE
! a má sa vypnúť
Promenne.DvereBuzzerCas.Value < [DateTimeNow]
#THEN
! nastavenie premennej indikujúcej zvuk. signal.
Promenne.DvereBuzzerAno.Value = FALSE
! vypne sa zvuková signalizácia
Prvy.Buzzer.Value = FALSE
#END

```



Obrázok 4.8. Upozornenie po 30 sekundách otvorených dverí



Obrázok 4.9. Upozornenie po 90 sekundách otvorených dverí

O kontrolu hodnoty CO_2 sa stará následujúce pravidlo. Pokiaľ je hodnota získaná zo senzoru menšia ako 58, pravidlový súbor sa už ďalej nevykonáva. Pokiaľ je hodnota vyššia ako 100 a nie je aktívne varovanie, tak sa nastaví premenná indikujúca varovanie a taktiež sa nastaví aj čas pre varovanie. O samotné varovanie sa stará posledný blok. Ak je aktívna premenná indikujúca varovanie a uplynul čas od predchádzajúceho varovania, pridá sa Info, ktoré sa zobrazí v aplikácii a nastaví sa čas pre ďalšie upozornenie.

```

! kontrola vydýchaného CO2

! nie je zvýšené CO2 a ani nebolo
#IF
  ! pokiaľ koncentrácia CO2 je pod hodnotou
  Druhy.VzduchCO2.Value < 58
  ! a nie je varovanie
  Promenne.VzduchCO2VarovanieAno.Value = FALSE
#THEN
  ! koniec vykonávania pravidiel pre tento súbor
  @RULES-END
#END

! skončilo zvýšené CO2
#IF
  ! pokiaľ koncentrácia CO2 klesla pod hodnotu
  Druhy.VzduchCO2.Value < 58
#THEN
  ! tak už nie je varovanie
  Promenne.VzduchCO2VarovanieAno.Value = FALSE
  ! odstránenie hlásenia o varovaní
  @INFO OUT;VZDUCH-CO2
#END

! začalo zvýšene CO2
#IF
  ! pokiaľ je koncentrácia CO2 vyššia než
  Druhy.VzduchCO2.Value > 100
  ! ešte nie je aktívne varovanie
  Promenne.VzduchCO2VarovanieAno.Value = FALSE

#THEN
  ! tak je varovanie aktívne
  Promenne.VzduchCO2VarovanieAno.Value = TRUE
  ! nastavenie času pre ďalšie varovanie (hneď)

```

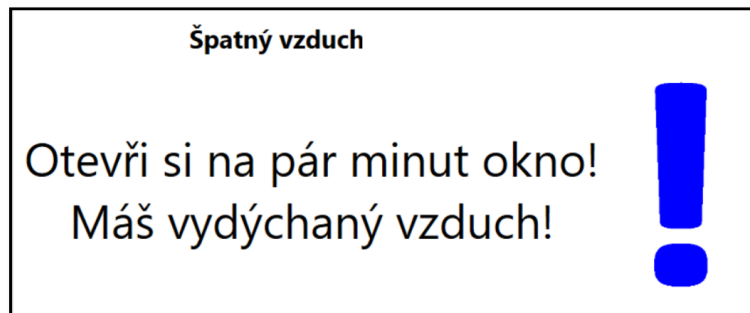
```

Promenne.VzduchCO2VarovanieCas.Value = [DateTimeNow]
#END

! periodické varovanie o zvýšenom CO2
#IF
! pokiaľ je varovanie aktívne
Promenne.VzduchCO2VarovanieAno.Value = TRUE
! a uplynul čas medzi varovaniami
Promenne.VzduchCO2VarovanieCas.Value < [DateTimeNow]
#THEN
! vlož do 'Info' hlásenie o nutnosti otvoriť okno
@INFO ADD;VZDUCH-CO2;150;Špatný vzduch;Otvri si na pár minut okno!
|Máš vydýchaný vzduch!
! nastavnie času pre ďalšie varovanie
Promenne.VzduchCO2VarovanieCas.Value = [DateTimeNow] + 00h01m00s
#END

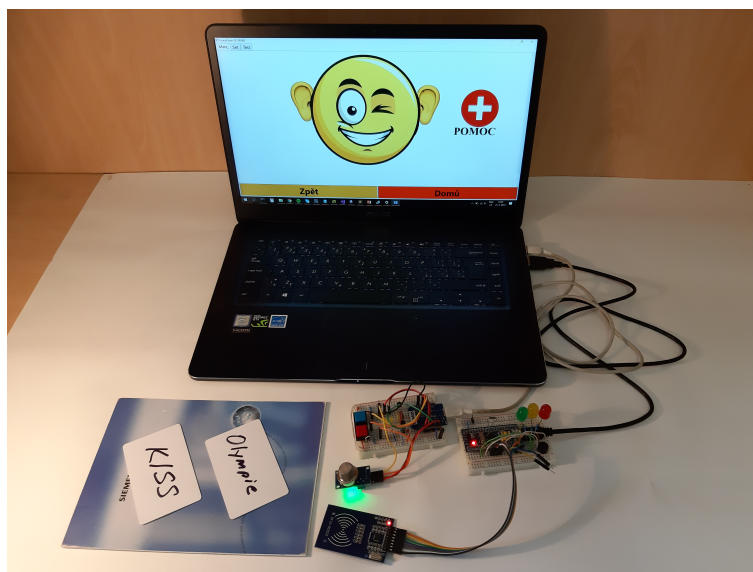
```

Pri prekročení hodnoty sa zobrazí Info definované v pravidlovom súbore:



Obrázok 4.10. Info upozorňujúce na zvýšenú koncentráciu CO_2

Na nasledujúcom obrázku je vidieť spustenú aplikáciu na počítači s pripojenými dvoma zariadeniami, ktoré obsahujú už popísané senzory. Vedľa nich sú NFC karty, po ktorých priložení k senzoru sa dá spustiť internetové rádio.



Obrázok 4.11. Vzorová aplikácia s demonstračným HW

Na zobrazenie aktuálnych hodnôt premenných, konštánt a dát prichádzajúcich zo senzorov z jednotlivých zariadení slúži v aplikácii záložka zobrazená na obrázku.

Device(s) / Variable(s) / Constant(s)
Name: Prvni / SerNum: 1234 Time(s) Update: 00h00m00s (01d01m0001r) / Lost: 12h00m00s Name: KartaNFC / Value: --- (None) / Changed: 00h00m00s (01d01m0001r) / Update: 00h00m00s (01d01m0001r) Name: Pohyb / Value: --- (None) / Changed: 00h00m00s (01d01m0001r) / Update: 00h00m00s (01d01m0001r) Name: Buzzer / Value: False (Bool) / Changed: 00h00m00s (01d01m0001r) / Update: 00h00m00s (01d01m0001r) Name: Green / Value: False (Bool) / Changed: 00h00m00s (01d01m0001r) / Update: 00h00m00s (01d01m0001r) Name: Orange / Value: False (Bool) / Changed: 00h00m00s (01d01m0001r) / Update: 00h00m00s (01d01m0001r) Name: Red / Value: False (Bool) / Changed: 00h00m00s (01d01m0001r) / Update: 00h00m00s (01d01m0001r)
Name: Druhý / SerNum: 5678 Time(s) Update: 16h23m24s (21d05m2019r) / Lost: 12h00m00s 15-48-43-56-78-01-00-01-00-01-00-01-00-10-55-10-FF-10-3F-FF-34 Name: Dvere / Value: False (Bool) / Changed: 16h22m53s (21d05m2019r) / Update: 16h23m24s (21d05m2019r) Name: Okno / Value: False (Bool) / Changed: 16h22m53s (21d05m2019r) / Update: 16h23m24s (21d05m2019r) Name: Vystup1 / Value: False (Bool) / Changed: 16h22m53s (21d05m2019r) / Update: 16h23m24s (21d05m2019r) Name: Vystup2 / Value: False (Bool) / Changed: 16h22m53s (21d05m2019r) / Update: 16h23m24s (21d05m2019r) Name: TeplotaVnitri / Value: 85 (Integer) / Changed: 16h22m53s (21d05m2019r) / Update: 16h23m24s (21d05m2019r) Name: TeplotaVnejsi / Value: 255 (Integer) / Changed: 16h23m09s (21d05m2019r) / Update: 16h23m24s (21d05m2019r) Name: VzduchCO2 / Value: 63 (Integer) / Changed: 16h23m24s (21d05m2019r) / Update: 16h23m24s (21d05m2019r)
Name: Promenne / SerNum: 0000 Time(s) Update: 00h00m00s (01d01m0001r) / Lost: 12h00m00s Name: DverePosledniStav / Value: False (Bool) / Changed: 00h00m00s (01d01m0001r) / Update: 00h00m00s (01d01m0001r)

Obrázok 4.12. Záložka pre zobrazenie hodnôt

V tejto kapitole bola popísaná vzorová aplikácia, ktorá je východzím bodom pre vlastnú konfiguráciu.

Kapitola 5

Záver

Táto bakalárska práca sa zaoberala návrhom a tvorbou prototypu tzv. jednoduchého domáceho spoločníka. Najprv som hľadal podobné už existujúce projekty. Ajkeď existuje niekoľko projektov pre rôznych domácich spoločníkov, tak väčšinou sú to veľmi sofistikované zariadenia alebo komplexné systémy. Tieto existujúce systémy majú dve spoločné nevýhody.

- Cieľovou skupinou väčšiny zariadení nie je dôchodca, týmpádom nie je dbaný ohľad na ich špecifické potreby
- Komunikácia je v anglickom jazyku

Na základe zistených informácií bol vytvorený návrh jednoduchého domáceho spoločníka spĺňajúceho tieto hlavné požiadavky:

- jednoduchosť, jednotnosť a prehľadnosť ovládacieho prostredia
- možnosť ovládania iba dotykcom
- možnosť konfigurácie podľa cieľovej osoby
- možnosť pripojenia domácich senzorov

Celkový návrh aplikácie je podrobne popísaný v kapitole 3.1 Ako vývojové prostredie pre tvorbu tejto experimentálnej aplikácie bol zvolený MS.NET Framework. Jeho výhodou je jednoduchosť tvorby aplikácie pre MS Windows a do veľkej miery aj jej neskoršia portácia na platformu ako je Android.

Celá aplikácia sa skladá z troch základných častí

- Obrazové GUI s jednotným ovládaním pre všetky vytvárané úlohy
- Dátové položky pre uchovávanie dát z ľubovoľných typov senzorov
- Nástroj interpretujúci vytvorené konfiguračné a skriptovacie súbory

Práve posledná menovaná časť aplikácie je veľkým prínosom pre jej všestranné použitie. Konfiguráciu možno vytvoriť podľa potrieb užívateľa. V tejto časti bol tiež naprogramovaný nástroj spracovávajúci tzv. pravidlá, pomocou ktorých sa dá vytvoriť komplexné chovanie aplikácie (reakcie na hodnoty zo senzorov, reakcie na úkony užívateľa atď...) Niektoré príklady sú uvedené v kapitole 4.8.

Jedná sa o experimentálnu a teda prototypovú aplikáciu, takže niektoré jej časti sú len prvotným návrhom a zaslúžia si dôkladnejšie prepracovanie, čo by však presiahlo rozsah tejto práce. Ale vzhľadom k súčasnému stavu vytvorenej aplikácie sa dá túto prácu považovať za vydarenú a ako východzí bod pre aplikáciu do skutočne reálneho prostredia.

Vytvorená aplikácia je určená najmä pre starsích ľudí pohybujúcich sa samostatne po byte/dome a potrebuje často nejakú malú pomoc alebo nápovedu. Aplikácia je taktiež pomocou externých senzorov schopná kontrolovať niektoré základné veci, ako napríklad otvorené dvere do bytu, otvorené okno, prípadne zvýšenú koncentráciu CO_2 v ovzduší.

Rovnako je schopná pripomínať udalosti a sťahovať z webu rôzne informačné súbory. Takáto aplikácia obsahuje veľký potenciál a závisí na jej konfigurácii aké schopnosti cieľovému užívateľovi poskytne.



Literatúra

- [1] *A look at the lives of the elderly in the EU today [online]. [cit. 2019-03-29].*
<https://ec.europa.eu/eurostat/cache/infographs/elderly/index.html>.
- [2] *Voice Assistants: How Artificial Intelligence Assistants Are Changing Our Lives Every Day [online]. [cit. 2019-04-02].*
<https://www.smartsheet.com/voice-assistants-artificial-intelligence>.
- [3] *Choosing the best voice assistant for your home [online]. [cit. 2019-04-02].*
<https://geeksfl.com/blog/best-voice-assistant/>.
- [4] *Elliq [online]. [cit. 2019-04-02].*
<https://elliq.com/>.
- [5] *GrandPad [online]. [cit. 2019-04-02].*
<https://www.grandpad.net/>.
- [6] *Robot Aeolus [online]. [cit. 2019-04-02].*
<https://aeolusbot.com>.
- [7] *Buddy, the emotional robot [online]. [cit. 2019-04-02].*
<https://buddytherobot.com/en/buddy-the-emotional-robot/>.
- [8] Matthew MacDonald. *Pro WPF in C# 2010*. Apress, 2013. ISBN 978-1-4302-7205-2.
- [9] Matthew MacDonald. *Pro WPF: Windows Presentation Foundation in .NET 3.0*. Apress, 2007. ISBN 1-59059-782-6.
- [10] *StackOverflow [online]. [cit. 2019-02-25].*
<https://stackoverflow.com/>.
- [11] *Getting Started (WPF) [online]. [cit. 2019-03-04].*
<https://docs.microsoft.com/en-us/dotnet/framework/wpf/getting-started/>.
- [12] *C# Guide [online]. [cit. 2019-03-04].*
<https://docs.microsoft.com/en-us/dotnet/csharp/>.
- [13] *Visual Studio Tools for Xamarin [online]. [cit. 2019-04-17].*
<https://visualstudio.microsoft.com/xamarin/>.
- [14] *Adaptech [online]. [cit. 2019-12-03].*
<https://www.adaptech.cz/novinka-99.php>.
- [15] *LiveCharts [online]. [cit. 2019-05-11].*
<https://lvcharts.net/App/examples/wpf/start>.



Príloha **A**

Skratky

- .NET .NET je platforma pre vývoj rôznych typov aplikácií (webové, mobilné...). .NET framework vie pracovať s viacerými programovacími jazykmi, ako napríklad C#, C++, F# alebo VB.NET.
- API Application programming interface - rozhranie, vďaka ktorému môžu komunikovať dve aplikácie a vymieňať si dáta.
- GUI Graphical user interface - grafické používateľské rozhranie.
- HW Hardware - fyzicky existujúce technické vybavenie aplikácie.
- JSON JavaScript Object Notation - formát pre uloženie a prenos dát, používa sa napr. pri posielaní dát zo servera na web. Účel je, aby dáta z jsonu boli ľahko pochopiteľné.
- MS Microsoft - firma, ktorá vyvíja, vyrába a licencuje mnoho produktov pre rôzne zariadenia. Najznámejšie sú operačný systém Microsoft Windows alebo produkty rady Microsoft Office.
- NFC Near-field communication - súbor komunikačných protokolov umožňujúcich dvom zariadeniam, z ktorých zväčša jedno býva prenosné (napr. mobil, smartwatch...) zabezpečiť komunikáciu priblížením jedného k druhému do vzdialenosti max 4 cm.
- SW Software - programové vybavenie počítača.
- WPF Windows Presentation Form - nástroj na tvorbu GUI používa, sa v .NET frameworku.



Príloha **B**

Obsah priloženého CD

aplikacia.exe	Vytvorená aplikácia
BP_PechnikMaros.pdf	Text bakalárskej práce