ŽILINSKÁ UNIVERZITA V ŽILINE

FAKULTA RIADENIA A INFORMATIKY

Softvérové vybavenie pre raspberry pi implementujúce funkcionality inteligentného domu

Software house for raspberry pi

Diplomová práca

Roman Masarovič

Vedúci práce: Ing. Katarína Zábovská, PhD.

Registračné číslo 442/2013 **Ministerské číslo práce 28360520142442**

ŽILINA 2014

ŽILINSKÁ UNIVERZITA V ŽILINE, FAKULTA RIADENIA A INFORMATIKY.

ZADANIE TÉMY DIPLOMOVEJ PRÁCE.

Študijný program : Informačné systémy

Zameranie: Aplikovaná informatika

Meno a priezvisko
Roman Masarovič

Osobné číslo
554241

Názov práce v slovenskom aj anglickom jazyku

Softvérové vybavenie pre raspberry pi implementujúce funkcionality inteligentného domu

Software house for raspberry pi

Zadanie úlohy, ciele, pokyny pre vypracovanie (Ak je málo miesta, použite opačnú stranu)

Cieľ diplomovej práce:
Cieľom diplomovej práce je navrhnúť a implementovať programový systém pre raspberry pi, ktorý zabezpečí funkcie inteligentného domu. Navrhnúť a vytvoriť používateľské prostredie pre návrh a správu inteligentného domu.

Obsah:

1. Analýza potrieb pre návrh vybavenia inteligentného domu
2. Základné úlohy riadiaceho systému
3. Identifikácia a návrh funkcionalít inteligentného domu
4. Realizácia navrhnutého riešenia
5. Overenie funkčnosti riešenia

Témy z predmetov študijného zamerania 1, 2, 3
Meno a pracovisko vedúceho DP: Ing. Katarína Zábovská, PhD., KMME, ŽU
Meno a pracovisko tútora DP:

vedúci DP (dátum a podpis)	tútor (dátum a podpis)	vedúci katedry (dátum a podpis)	garant (dátum a podpis)
30.10.2013		30.10.2013	

Zadanie zaregistrované dňa 30.10.2013 pod číslom 442/2013 podpis

ABSTRAKT

MASAROVIČ, Roman: *Softvérové vybavenie pre raspberry pi implementujúce funkcionality inteligentného domu*. [diplomová práca]. – Žilinská univerzita v Žiline. Fakulta riadenia a informatiky; Katedra makro a mikroekonomiky. – Vedúci: Ing. Katarína Zábovská, PhD. – Stupeň odbornej kvalifikácie: Inžinier v študijnom programe Informatika. Žilina: FRI ŽU, 2014. – 75 s.

Diplomová práca sa venuje problematike návrhu a implementácie programového systému pre Raspberry Pi, ktorý zabezpečí funkcie inteligentného domu. Teoretická časť práce predstavuje a objasňuje základné teoretické pojmy v problematike inteligentných domov, bližšie popisuje požiadavky na inteligentný dom a funkcie inteligentného domu. Predstavuje teoretické možnosti využitia Raspberry Pi pri návrhu na zabezpečenie funkcií inteligentného domu. V praktickej časti práce prináša vlastný projekt návrhu a implementácie programového systému pre Raspberry Pi, ktorý zabezpečuje vybrané funkcie inteligentného domu. Vytvorené programové vybavenie bolo vyvíjané pomocou modifikovanej agilnej metodiky Scrum; ako vývojové prostredie bolo zvolené NetBeans, ako ďalšie vývojové nástroje boli použité PuTTY, WinSCP, Eclipse. Softvér pre Raspberry Pi bol primárne naprogramovaný v programovacom jazyku JAVA. Navrhnuté riešenie vybraných funkcionalít inteligentného domu bolo v praxi zrealizované a funkčnosť riešenia bola úspešne overená.

Kľúčové slová: Inteligentný dom. Funkcie inteligentného domu. Raspberry Pi (RPi). Návrh softvéru na zabezpečenie funkcií inteligentného domu pre RPi. Implementácia softvéru na zabezpečenie funkcií inteligentného domu pre RPi.

ABSTRACT

MASAROVIČ, Roman: *Software house for raspberry pi.* [diploma thesis]. – The University of Žilina. Faculty of Management Science and Informatics; Departement of macro and microeconomic. – Tutor: Ing. Katarína Zábovská, PhD. – Qualification level: Engineer in study program Informatics. Žilina: FRI ŽU in Žilina, 2014. – 75 p.

The submitted thesis deals with design and implementation of software system for Raspberry Pi, which ensures the basic functions of a smart house.
The theoretical part of thesis presents and explains the basic concept of a smart house and its features, also further describes the additional requirements for a smart house. There for presents the theoretical possibilities how to use Raspeberry Pi during the designing of a smart house to ensure its required functions. In the practical part is introduced own project , which design and implement software system for Raspberry Pi, which provides the selected functions of a smart home. The created software was developed by using a modified agile method Scrum, as a development environment was chosen NetBeans, and as other development tools were used PuTTY, WinSCP, Eclipse. Software for Raspberry Pi was initially programmed in language JAVA. The proposed solution of selected functions for a smart house was implemented in the practise and the functionality of solution was successfully verified.

Key words: smart house, functions of a smart house, Raspberry Pi (RPi), the proposed software for ensuring the basic functions of a smart house for RPi. Implementation of software, which ensures the basic functions of a smart house.

Prehlásenie

Prehlasujem, že som túto prácu napísal samostatne, a že som uviedol všetky použité pramene a literatúru, z ktorých som čerpal.

V Žiline, dňa 25.4.2014

Roman Masarovič

OBSAH

ZOZNAM OBRÁZKOV A TABULIEK

ÚVOD

Vývoj informačných technológií založených na elektronickom spracovaní a prenose informácií sa premieta do všetkých oblastí ľudského života. Stále progresívnejšie prenikanie moderných elektronických technológií do všetkých oblastí nášho života sa výrazne prejavuje i pri systémových riešeniach technicko-bezpečnostných funkcií budov (domov).V súčasnej informačnej spoločnosti s tým, ako rastú nároky na pohodlie, bezpečnosť a hospodárnosť v stavebníctve, narastá i podiel automatizácie tak v bytovej výstavbe, ako i v účelových stavbách (Kabele, 2007).

Trend nástupu inteligentných budov (domov) je badateľný už aj u nás. Tento rastový trend zasiahne mnoho segmentov stavebníctva a s ním súvisiacich odborov. Kľúčovými subjektmi na poli realizácie inteligentných domov v praxi sa stávajú výrobcovia riešení pre inteligentné domy. Za posledných niekoľko rokov vzniklo mnoho riešení, ktoré sú neustále inovované, vznikajú celé produktové rady, ktoré poskytujú zákazníkom širokú škálu možností výberu na zabezpečenie funkcionalít inteligentného domu. Pri rozhodovaní užívateľov o miere inteligencie v dome zohrávajú kľúčovú úlohu najmä tieto požiadavky: užívateľ požaduje za určité zriaďovacie náklady adekvátnu mieru komfortu, bezpečnosti a úspor. Výrobcovia ponúkajú okrem sofistikovaných, rozsiahlych a implementačne náročných riešení aj riešenia jednoduchšie, s cieľom osloviť čo najširší segment potenciálnych zákazníkov. Tieto riešenia sa zameriavajú predovšetkým na poskytovanie najpoužívanejších funkcií inteligentného domu a vyznačujú sa jednoduchým ovládaním a inštaláciou, ako aj nízkymi zriaďovateľskými nákladmi (Skyva, 2013).

Jednou z nízko nákladových možností realizácie funkcií inteligentného domu je ich implementácia s pomocou miniatúrneho počítača Raspberry Pi, ktorého najväčšou výhodou je nízka nákupná cena a vysoká variabilita možných riešení pri návrhoch na tvorbu používateľského prostredia na správu a riadenie inteligentného domu. Táto možnosť ma zaujala a z tohto dôvodu som si vybral vývoj softvérového riešenia na implementáciu funkcionalít inteligentného domu s využitím Raspberry Pi ako tému mojej diplomovej práce.

Cieľom predkladanej diplomovej práce je navrhnúť a implementovať programový systém pre Raspberry Pi, ktorý zabezpečí funkcie inteligentného domu, ako aj navrhnúť a vytvoriť používateľské prostredie pre návrh a správu inteligentného domu.

Práca je rozdelená na teoretickú a praktickú časť. Obsahuje tri samostatné kapitoly. Prvé dve kapitoly sú teoretické – prvá kapitola je zameraná na objasnenie základných teoretických pojmov súvisiacich s problematikou inteligentných budov (domov). Druhá kapitola je zameraná na popísanie funkcií inteligentného domu, približuje niektoré existujúce systémy pre inteligentné domy v súčasnosti a predstavuje teoretické možnosti využitia Raspberry Pi pri návrhu na zabezpečenie funkcií inteligentného domu.

V praktickej časti práce (tretia kapitola) predstavujeme náš projekt návrhu a implementácie softvérového riešenia pre Raspberry Pi, ktorý zabezpečí vybrané funkcie inteligentného domu. Programový systém sme vytvorili a implementovali pre majiteľov víkendového domu, ktorí požadovali navrhnúť a implementovať taký systém, ktorý bude pre nich cenovo dostupný a ktorý zároveň adekvátne uspokojí ich požiadavky na úsporu nákladov, bezpečnosť a požadovanú mieru komfortu. Prílohou práce je nami vytvorené kompletné programové vybavenie na zabezpečenie vybraných funkcií inteligentného domu pomocou Raspberry Pi na CD nosiči.

1 **PROBLEMATIKA INTELIGENTNÉHO DOMU**

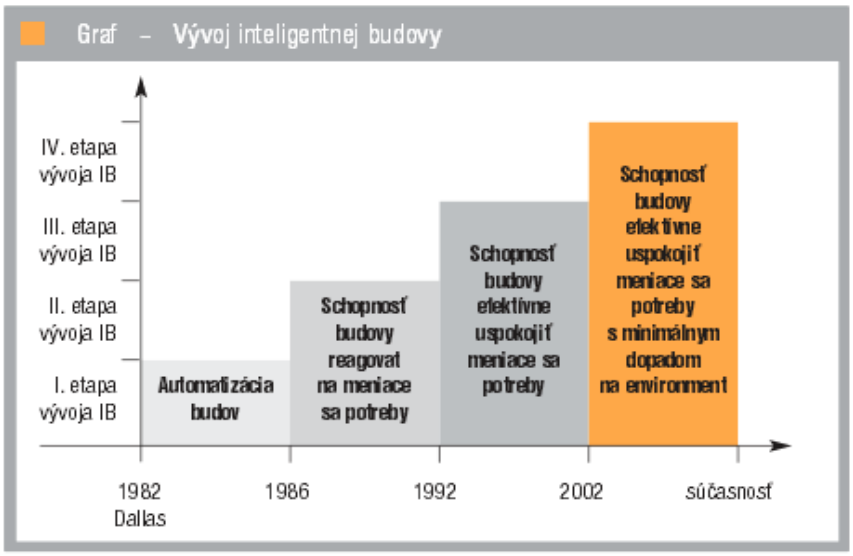
V oblasti súkromnej bytovej výstavby bolo ešte donedávna použitie moderných informačných technológií a automatizovaných systémov finančne veľmi náročné. Súčasný rýchly rozvoj v tomto odvetví však postupne vedie k ich pomerne dobrej finančnej dostupnosti a viaceré systémy sa stávajú štandardom pri výstavbe, rekonštrukcii alebo modernizácii v oblasti súkromných obytných stavieb. Príkladom toho sú napr. vykurovacie systémy, systémy pre riadenie osvetlenia alebo systémy zabezpečenia obytnej budovy. Okrem šetrenia energiami a znižovania nákladov na ne, v popredí je tu aj zabezpečenie komfortu a pohodlia obyvateľov domu. Oproti tomu pri účelových stavbách je snaha pomocou moderných integrovaných riadiacich systémov dosiahnuť čo najvhodnejšie prostredie pre činnosti, na ktoré je budova určená. Toto smeruje k možnosti dosiahnuť čo najefektívnejšie využitie budovy, čo umožňuje dosiahnuť vyššie zisky. Okrem optimalizácie efektivity a nákladov sú moderné systémy výhodné aj preto, že vďaka nim sa účelová budova stáva flexibilná. Pri zmene koncepcie využívania budovy stačí preprogramovať integrované inteligentné komponenty tak, aby ich prevádzka zodpovedala novej koncepcii.

Ak hovoríme o inteligentných budovách (domoch) je potrebné aj vyšpecifikovať, čo tento pojem znamená. Inteligentný dom (inteligentná budova, ďalej aj IB) sú v súčasnosti už bežne používané pojmy, neexistuje však ich jednoznačná definícia. V odbornej literatúre, ale aj v praxi, sa v oblasti obytných domov stretávame aj s anglickým termínom smart home (chytrý dom) alebo aj s termínom digitálny dom, či i-dom. Medzi odbornou aj laickou verejnosťou existuje viacero definícií a predstáv, čo vlastne inteligentné budovy (domy) sú, aké atribúty by mali spĺňať, aby mohli byť označené za inteligentné.

1.1 Vývoj definície inteligentnej budovy

Vývoj definície inteligentnej budovy je možné rozdeliť do štyroch etáp(obr. 1). V prvej etape bola inteligencia budov chápaná ako ich automatizácia, v druhej etape to bola schopnosť budovy reagovať na meniace sa potreby jej užívateľov. V tretej etape sa kladie dôraz na schopnosť budovy efektívne uspokojovať meniace sa potreby jej

užívateľov a vo štvrtej etape sa kladie dôraz na schopnosť budovy efektívne uspokojovať meniace sa potreby jej užívateľov s dopadom na environment (Števo, 2012).



Obrázok 1 Vývoj inteligencie budov od ich vzniku po súčasnosť
Zdroj: Števo (2012, s. 11)

1.2 Rôznorodosť pohľadov na definíciu inteligentnej budovy

V rôznych krajinách a regiónoch, ako aj v rôznych disciplínach môžu prevládať rôznorodé preferencie a koncepty pri definovaní inteligentnej budovy. Prístupy k definovaniu IB možno rozdeliť do troch kategórií, ktoré sú uvedené nižšie:

1) Performance-based definitions - dôraz kladený na výkonnosť budovy
2) Services-based definitions – dôraz kladený na služby
3) System-based definitions – dôraz kladený na systém (Wang, 2009).

Performance-based definitions

Definície založené na výkonnosti definujú aké výkony by inteligentné budovy mali mať. Typickými definíciami pre tento prístup ku definovaniu inteligentnej budovy sú definície americkej inštitúcie IBI (Intelligent Building Institute of USA) a európskej inštitúcie EIBG (European Intelligent Building Group).

Americký ústav pre inteligentné budovy IBI ich definuje ako „racionálne navrhnuté budovy, ktoré poskytujú nákladov efektívne, komfortné a príjemné prostredie pomocou optimalizácie štyroch základných prvkov, a to: stavebnej konštrukcie, technických zariadení, služieb a manažmentu a ich vzájomných vzťahov (Puškár, 2007). Americký ústav pre inteligentné budovy charakterizoval budovu ako inteligentnú vtedy, „keď zabezpečuje produktívne a nákladovo efektívne prostredie pomocou optimalizácie štyroch základných prvkov – stavebnej konštrukcie, technických zariadení, služieb a manažmentu a ich vzájomných vzťahov“. Táto definícia sa zameriavala najmä na dostupnosť použitých technológií pri vytváraní konštrukcie (Števo, 2013).

Podľa EIBG je inteligentná budova taká budova, ktorá v sebe integruje rozličné systémy, schopné v maximálnej miere efektívne riadiť zdroje a koordinovať možnosti v podobe technických zariadení, investícií a znižovania nákladov(Puškár, 2007).Oproti americkému chápaniu je európske odlišné tým, že myslí viac na užívateľa inteligentnej budovy ako na samotnú budovu. Budova je len obalom kvalitného prostredia pre jej obyvateľov.

Services-based definitions

Definícia inteligentnej budovy z tohto pohľadu dáva dôraz na množstvo a kvalitu služieb, ktoré budova užívateľom poskytuje. Príklad takéhoto pohľadu na inteligentnú budovu dáva definícia JIBI (Japan Intelligent Building Institute), ktorá definuje ako inteligentnú takú budovu, ktorá je vybavená komunikačnými službami a automatizáciou a je vhodná na inteligentné aktivity, zároveň sú zdôraznené potreby užívateľov. Dôležité pritom je, aby boli splnené tieto štyri body: inteligentná budova slúži ako miesto pre prijímanie a vysielanie informácií, ktoré podporuje ich efektívne spravovanie; zabezpečuje spokojnosť a pohodlie jej užívateľov; racionalizuje správu budovy a znižuje náklady na prevádzku; zabezpečuje rýchlu, flexibilnú a ekonomickú odpoveď voči meniacemu sa sociologickému prostrediu, voči rôznorodým pracovným požiadavkám či obchodným stratégiám.

System-based definitions

Táto definícia definuje inteligentnú budovu ako budovu poskytujúcu automatizáciu budovy, automatizáciu komunikácie a pracovných priestorov. Túto definíciu navrhol čínsky IB Design Standard (GB / T50314-2000) a je označovaná aj

ako definícia „3 A": building automation (BA), communication automation (CA) and Office automation (OA) (Wang, 2009).

Každá z definícií pristupuje k inteligentnej budove z inej perspektívy. V Ázijských krajinách sa prikladá dôraz na technológie a využitie high-tech. V Európe sa zameriava na spolupôsobenie človeka a konštrukcie. Dôraz je kladený na maximalizovanie pohody pre užívateľa. Na druhej strane v USA je na prvom mieste znižovanie nákladov a až potom sa stavia užívateľ (Strigáčová, 2012).

Inteligentný dom alebo objekt by mal:

- integrovať ovládanie, teda aj riadenie všetkých technológií (technického vybavenia) objektu pod jeden centrálny riadiaci systém, ktorého úlohou je sledovanie všetkých parametrov a funkcií objektu a následné prispôsobovanie nameraných hodnôt na požadované hodnoty,
- optimalizovať správanie sa objektu a jeho súčastí z hľadiska spotreby energií a súčasne zachovať požiadavky na primeraný komfort užívateľov(obyvateľov),
- zabezpečovať v spolupráci so zabezpečovacím systémom maximálnu dosiahnuteľnú bezpečnosť užívateľov (obyvateľov),
- prostredníctvom príjemného užívateľského rozhrania umožniť užívateľom (obyvateľom) jednoduché používanie a využívanie všetkých funkcií a vlastností objektu bez nutnosti štúdia manuálov.

Tieto vlastnosti platia nielen pri rodinných domoch, ale aj pri komerčných objektoch. Rozdiel je len v rozsahu a typoch funkcií v objektoch – napr. bazén, sauna, či domáce kino nie sú bežnou výbavou pri komerčných objektoch. Situácia v SR je oproti krajinám v západnej Európe odlišná v tom, že tam je používanie integrovaných riadiacich systémov bežné aj pri strednej kategórii rezidenčných a komerčných objektov, kým u nás je to stále výsada skôr tých luxusných objektov. Podľa Lelovského (2013) je to dané kúpyschopnosťou obyvateľstva, ale aj konzervativizmom Slovákov, ktorí ešte nie sú zvyknutí používať moderné systémy.

1.3 Požiadavky na inteligentnú budovu

Vzhľadom na zvyšujúce sa nároky užívateľov obytných domov na komfort a pohodlie, je v súčasnosti hlavným zámerom investorov zabezpečiť komfortné bývanie s maximálnym pohodlím, ale s nízkymi prevádzkovými nákladmi. Riešením je inteligentný dom, ktorý je vybavený modernými technológiami a spĺňa požiadavky investora na komfort a bezpečnosť. Podľa Pilipovej (2011) je „*hlavným cieľom inteligentného domu uľahčiť a spríjemniť užívateľom bývanie. Inteligentný dom dokáže všetku techniku medzi sebou prepojiť, zjednotiť ovládanie a predovšetkým poskytnúť jednotný spôsob ovládania, ktorý je prispôsobený na mieru pre konkrétny dom a jeho užívateľov*“. Miera použitia inteligentných technológií v obytných budovách musí obyvateľovi prinášať viac benefitu v komforte, než ho zväzovať v jeho správaní a nútiť ho neustále sa zaoberať ovládaním techniky svojho bytu (Puškár , 2007)

Tvorba kvalitného obytného prostredia v inteligentnej budove má byť orientovaná na uspokojenie potrieb a na vytvorenie atmosféry bez stresových stimulov pre užívateľa tak, aby moderné technológie neznamenali pre užívateľa (obyvateľa) v konečnom dôsledku diskomfort. K príčinám vzniku diskomfortu v inteligentných obytných budovách patrí prekročenie optimálnej miery inteligencie, vplyv psychickej mikroklímy vnútorného prostredia, ako aj neschopnosť vytvoriť pohodu pre užívateľa. Dosiahnutie potrieb užívateľa umožňuje taká inteligentná budova, ktorá podľa miery inteligencie môže, ale aj nemusí obsahovať systém automatizácie budov. Je to z dôvodu, že ani najvýkonnejšie a najvyspelejšie automatizované riadenie samotné neurobí budovu efektívnejšou, ak použité technológie nebudú plne využívať potenciál stavebnej konštrukcie, okolitého prostredia, ako i individualitu jej užívateľov (Pilipová, 2011).

Požiadavky na inteligentné budovy môžeme rozdeliť na dve kategórie, a to: legislatívne požiadavky a požiadavky investorov, prevádzkovateľov a užívateľov IB.

1.3.1 Požiadavky investorov, prevádzkovateľov a užívateľov inteligentnej budovy

Požiadavky na IB závisia na druhu stavby a na individuálnych rozhodnutiach investorov, prevádzkovateľov a užívateľov IB.

Investor – Zámerom investora je poskytnúť užívateľovi maximálny komfort pri čo najnižšej cene a pri dosiahnutí maximálneho zisku.

Projektant – Hlavným poslaním projektanta je vytvoriť plne funkčnú konštrukciu so zameraním na dizajn.

Správca – V súčasnej dobe sa veľké nároky kladú na maximálne zníženie prevádzkových nákladov. Práve správca má na zodpovednosti sledovanie a čo najefektívnejšie riadenie a prevádzkovanie budovy počas jej užívania.

Budúci užívateľ – Pre užívateľa je okrem nákupnej ceny a nízkych prevádzkových nákladov veľmi dôležitá aj obsluha domu. Tá musí byť jednoduchá, aby dochádzalo k minimalizácii obsluhujúcej techniky a k dosiahnutiu celkovej pohody.

Pod inteligenciou v budove vnímame jednak zložité technické zariadenia, ale aj prvotný návrh, ktorý ovplyvňuje ďalšie náklady. (Strigáčová, 2012)

Požiadavky najviac ovplyvňujú bude IB vyzerať a ako bude fungovať. Určujú aké systémy a technológie budú v IB použité a a celkovú architektonickú koncepciu. Ide najmä o požiadavky, napr. spotreba energií, automatizácia, ovládanie systémov v budove, bezpečnosť, správa budovy, režimy a scény, zdravé a príjemné vnútorné prostredie a pod.

1.3.2 Požiadavky na inteligentný dom

V kategórii budov (domov) určených na bývanie sú základné požiadavky na takýto typ domu zväčša tri, a to: dosiahnutie úspor, zaistenie bezpečnosti a komfortu jeho obyvateľov (Skyva, 2013).

Dosiahnutie úspor- energeticky najnáročnejšou technológiou v dome je vykurovanie, preto jeho riadenie inteligentnými technológiami môže pomôcť vytvárať najvyššie prevádzkové úspory. Ďalšie úspory je možné dosiahnuť vhodne nastaveným používaním roliet (žalúzií) v dome. V lete sa tým zabráni prehrievaniu domu (nie je nutné používať klimatizáciu počas celého dňa), naopak počas zimy môžu spustené vonkajšie rolety zabrániť prílišnému vychladnutiu domu cez noc. Ďalšou funkciou, ktorú inteligentné domy poskytujú, je meranie a správa energií. Niektoré inteligentné domy už vedia na rôznu výšku spotreby aj reagovať. Najvyššiu úroveň úspor môžu priniesť pre obyvateľov inteligentné domy s predikciou spotreby a tak jeho obyvateľ môže adekvátne zareagovať a upraviť svoje správanie.

Zaistenie bezpečnosti – zaistenie bezpečnosti obyvateľov inteligentného domu je poňaté komplexne a spadá tu nielen maximálna možná ochrana obyvateľov, ale aj samotného domu. Je to napr. ochrana proti vlámaniu, aktívna simulácia prítomnosti v čase neprítomnosti obyvateľov domu, ďalej je to ochrana proti haváriám ako únik vody alebo plynu, proti povodniam či proti požiaru.

Komfort obyvateľov inteligentného domu – vhodným výberom a implementovaním inteligentných funkcií môžeme dosiahnuť maximálny komfort pre obyvateľov domu, presiahnutie optimálnej miery inteligencie v inteligentnom dome však paradoxne môže viesť ku vzniku diskomfortu pre jeho obyvateľov. Inteligencia v budove sa stáva kontraproduktívnou. Užívateľovi už nezvyšuje kvalitu bývania, ale začína zasahovať do jeho intímneho prostredia alebo mu spôsobovať stres.

2 FUNKCIE INTELIGENTNÉHO DOMU

V súvislosti s funkciami inteligentného domu je potrebné najprv uvažovať o tom, do akej miery bude dom inteligentný. Podľa Valeša (2006) a Harpera (2003) je možné rozlišovať päť stupňov inteligencie inteligentného domu.

I. stupeň

Dom s inteligentnými zariadeniami a systémami

V dome sú použité samostatné, inteligentne fungujúce zariadenia a systémy, ktoré pracujú nezávisle od seba. Napr.: osvetlenie riadené systémom, ktorý pomocou rôznych snímačov a časovačov rozsvieti svetlá v miestnostiach.

II. stupeň

Dom s inteligentnými komunikačnými zariadeniami a systémami

Dom obsahuje inteligentne fungujúce zariadenia a systémy, ktoré si z dôvodu zdokonalenia svojej činnosti vymieňajú informácie a správy medzi sebou. Napr.: po uzamknutí vchodových dverí sa automaticky zapne bezpečnostný systém domu a vyšle príkaz na zhasnutie svetiel, stiahnutie žalúzií, vypnutie hudby a TV prijímača.

III. stupeň

Prepojený dom (connected home)

Dom je prepojený pomocou vonkajšej a vnútornej siete. To umožňuje vzdialenú správu domu z akéhokoľvek miesta pripojeného na internet. Napr.: v prípade alarmu bezpečnostný systém rozsvieti všetky svetlá v dome a na záhrade, vytiahne žalúzie, privolá bezpečnostnú službu a umožní vzdialený prístup k záznamom z bezpečnostných kamier.

IV. stupeň

Učiaci sa dom

Zaznamenáva aktivity užívateľov v dome, ktoré sú neustále vyhodnocované. Tieto informácie sa následne využívajú podľa potrieb na riadenie zariadení v dome. Napr.: ovládanie svetiel a kúrenia podľa zvyklostí užívateľov domu a ich najčastejšieho používania.

V. stupeň

Pozorný dom

Aktivita a poloha ľudí a predmetov v dome sa neustále vyhodnocuje a na základe týchto informácií sú technológie automaticky ovládané podľa predvídaných potrieb. Napr.:

možnosť využitia špeciálnej podlahy snímajúcej kroky osôb na identifikáciu rôznych osôb a miesta, kde sa nachádzajú.

2.1 Klasifikácia funkcií v inteligentnom dome

Odborníci funkcie inteligentných budov (domov) rozdeľujú viacerými spôsobmi. Kabele (2007) základné skupiny funkcií inteligentných budov nazýva službami a radí medzi ne energetické a ekologické služby, komfortné, bezpečnostné, dopravné, sociálne, zábavné a komerčné služby.

Autori Hamerník, Tanuška a Mudrončík (2012) navrhli rozdeliť funkcie inteligentného domu do niekoľkých skupín.

A) Kategórie funkcií podľa typu užívateľov inteligentného domu

Podľa typu užívateľov inteligentného domu ich rozdelili do troch skupín.

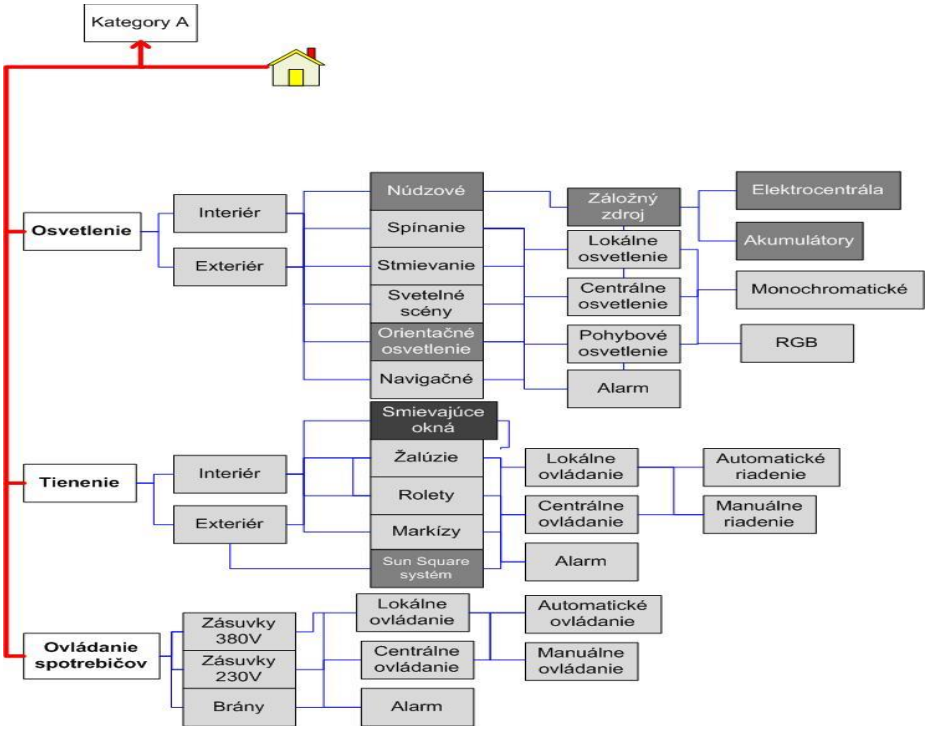
Základná skupina: funkcie v tejto skupine sú nevyhnutnou súčasťou inteligentného domu. Každý inteligentný dom by ich mal obsahovať. Kritériami pri výbere funkcií do tejto skupiny sú možnosti zaistenia štandardných potrieb obyvateľov a automatizácie rutinných činností.

I. úroveň: funkcie v tejto skupine priamo nadväzujú na základnú skupinu. Tieto funkcie sú určené pre špecifických obyvateľov a tvoria akoby nadstavbu funkcií v základnej skupine.

II. úroveň: funkcie umiestnené v tejto úrovni patria medzi maximálne dostupné funkcie pre vybavenosť inteligentného domu.

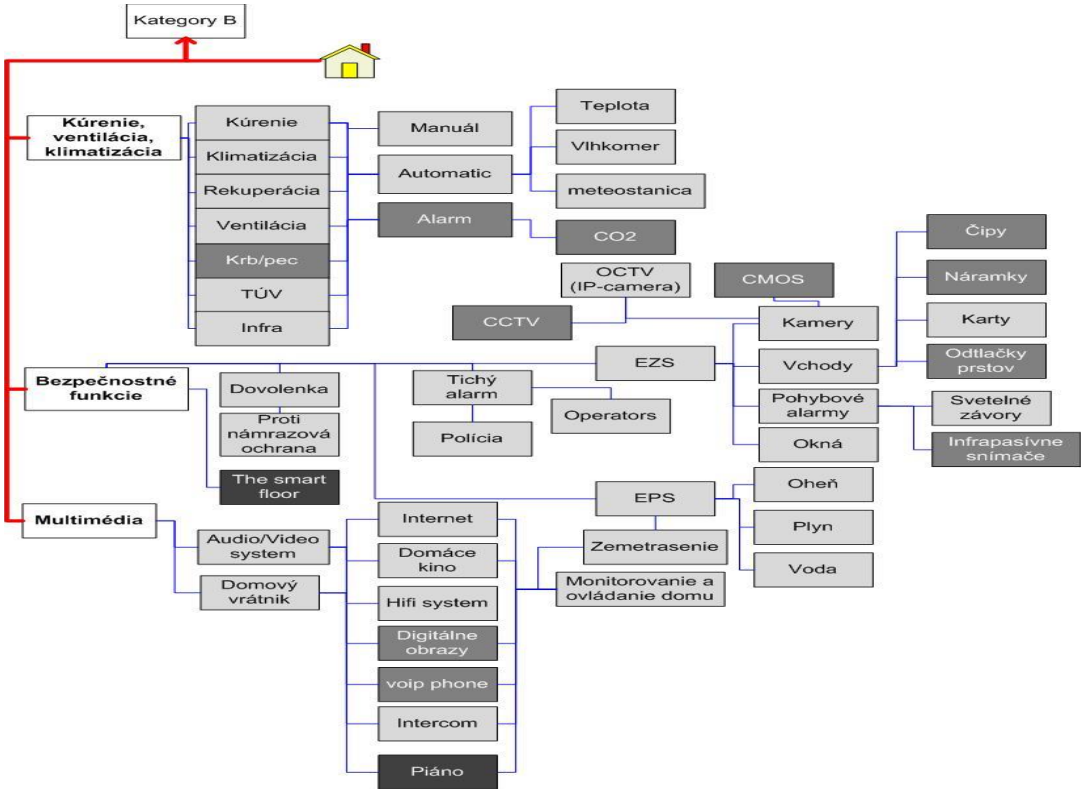
Inteligenciu domu zabezpečujú špecifikované funkcie. Tie možno kategorizovať do hierarchickej štruktúry; v horizontálnej rovine sú príbuzné funkcie, ktoré sa vyberajú podľa špecifických požiadaviek budúceho obyvateľa inteligentného domu. Z toho teda vyplýva, že výber funkcií inteligentného domu sa vykoná podľa typu obyvateľa a následne podľa jeho individuálnej špecifikácie funkcií. Dôsledná kategorizácia typov užívateľov inteligentného domu a špecifický výber funkcií podstatne zjednodušuje verifikáciu a validáciu realizovaného riešenia. Užívateľov domu môžeme kategorizovať napríklad podľa počítačovej gramotnosti, veku, zdravotného stavu, finančných možností či záujmov.

Možné rozdelenie funkcií inteligentného domu znázorňujú obr. 2 a obr. 3.



Obrázok 2 Možné rozdelenie funkcií inteligentného domu – kategória A

Zdroj: Hamerník, Tanuška (2012)



Obrázok 3 Možné rozdelenie funkcií inteligentného domu – kategória B

Zdroj: Hamerník, Tanuška (2012)

Hamerník, Tanuška a Mudrončík (2012) ďalej kategorizovali funkcie inteligentného domu podľa ich aplikácie v inteligentnom dome do jednotlivých skupín nasledovne:

B) Skupiny funkcií inteligentného domu podľa ich aplikácie v dome

B1 Osvetlenie (Lighting)

B2 Systém tienenia (Shading system)

B3 Ovládanie spotrebičov (Controlling of appliances)

B4 Kúrenie, ventilácia a klimatizácia (Heating, ventilation and air conditioning)

B5 Bezpečnostné funkcie (Safety functions)

B6 Multimédiá (Multimedia)

B7 Zdravie (Health)

B8 Kuchyňa (Kitchen)

B9 Zavlažovanie (Irrigation)

B10 Čistenie, údržba (Clean)

B11 Ovládanie (Control)

B1 Osvetlenie (Lighting)

Funkcie v tejto skupine majú optimalizovať osvetlenie v dome podľa požiadaviek jeho užívateľa – napr. miestne osvetlenie, centrálne osvetlenie, navigačné alebo núdzové svetlá, ďalej tiež na navodenie určitej nálady, tzv. svetelné scény. Pri definovaní požiadaviek na osvetlenie inteligentného domu pomáha norma EN 12464-1 a EN 12464-2 (Svetlo a osvetlenie).

B2 Systém tienenia (Shading system)

Tieniace prvky interiéru zabezpečujú nielen príjemnú teplotu obyvateľom inteligentného domu, ale zabezpečujú aj ich bezpečnosť a ochranu súkromia. Konkrétne prepočty, ktoré pri rovnakých podmienkach porovnávajú výsledky prehriatia a tienenia interiéru, určujú štandardizované skúšky a výpočty v súlade s normami EN 12567-2, EN 13363-2 a ISO 15099.

B3 Ovládanie spotrebičov (Controlling of appliances)

Všetky spotrebiče v dome sú pripojené cez sieť 400 V a 230 V pre európsku sieť. Ich ovládanie môže byť:

- automatické alebo manuálne cez lokálne alebo centrálne ovládanie

- špecifické funkcie a nastavenia pre jednotlivé zariadenia sú možné cez sériový port (RS-232), ktorý zabezpečuje obojsmernú komunikáciu so zariadením (napr. spätná informácia o vypnutí spotrebiča)
- cez infračervený port – používa sa predovšetkým pre A/V (audiovizuálne) technológie (nevýhodou je jednostranný spôsob komunikácie).

V najmodernejších spotrebičoch sa začína používať ethernetový port, ak je spotrebič súčasťou počítačovej siete, čo umožňuje široký rozsah regulácie.

B4 Kúrenie, ventilácia a klimatizácia (Heating, ventilation and air conditioning)

Funkcie z tejto skupiny tvoria podstatnú časť funkcií inteligentného domu a zabezpečujú tepelnú pohodu spojenú s dostatočným prívodom teplého alebo chladného vzduchu. Do tejto skupiny funkcií patria kúrenie, klimatizácia, rekuperácia, ventilácia, príprava teplej vody riadenie zdrojov tepla, ako infra vykurovanie, kozub, pec. Umožňuje tiež kombináciu zdrojov tepla, napr.: kotol na tuhé palivá pripojený s elektrickým kotlom, alebo systémy, ktoré používajú obnoviteľné zdroje energie . Ako ďalší príklad možno uviesť kombináciu kozub pec, ktorá obsahuje funkcie regulácie výkonu kozuba, optimálneho horenia, signalizácie prísunu paliva, signalizácie a regulácie prívodu vzduchu, signalizácie plného zásobníka popola, kontroly alarmom hladiny CO_2 v miestnosti a vo vetracej šachte.

B5 Bezpečnostné funkcie (Safety functions)

Zabezpečujú nielen ochranu majetku pred vlámaním, ale aj kontrolu uzatvorenia domu pri odchode, odstavenie chladenia alebo kúrenia po otvorení okien, kontrolu CO, CO2 alebo nedostatok kyslíka. Systém je možné rozdeliť podľa typu na:

- elektronický zabezpečovací systém,
- elektronický požiarny poplachový systém,
- systém na ochranu pred povodňou,
- kamerový monitorovací systém.

B6 Multimédiá (Multimedia)

Tieto funkcie sú rozdelené do dvoch základných skupín: audio video systémy a domový vrátnik. Funkcie zaradené do týchto skupín (napr. domáce kino, hifi systém, digitálne obrazy, voipphone, intercom alebo piano) umožňujú ovládať a monitorovať inteligentný dom.

B7 Zdravie (Health)

Neoddeliteľnou súčasťou inteligentného domu je starostlivosť o zdravie jeho obyvateľov. Zdravie je rozdelené na tri základné podskupiny, a to: výťah, lekár a GPS. Výťah využívajú väčšinou imobilní obyvatelia. GPS systém kontroluje polohu obyvateľa, počet krokov a spolupracuje s podskupinou lekár. Podskupina lekár obsahuje funkcie ako napr. SOS – privolanie rýchlej zdravotnej služby, možnosť dávkovania liekov, EKG, krvný obraz, meranie tlaku, výšky a hmotnosti, kontrola zraku, USG (sono vyšetrenie), kožné vyšetrenie, kontrola kĺbov, meranie teploty, kontrola stability tela človeka. Pri programe cvičenia inteligentný dom po konzultácii s lekárom naordinuje a kontroluje rehabilitáciu, odporučí cviky spolu s ich dávkovaním, príp. odporučí masáž. Najlepšie hodnoty na diagnostikovanie zdravotného stavu sú získavané počas spánku obyvateľa inteligentného domu.

B8 Kuchyňa (Kitchen)

Žiadna domácnosť sa nezaobíde bez kuchyne, preto pri funkciách inteligentného domu sa kladie veľký dôraz na vývoj tých funkcií, ktoré uľahčujú prípravu, kontrolu a objednávku potravín. Medzi najdôležitejšie funkcie využívané v kuchyni patria kontrola pečenia, donáška tovaru do domu, domáca pekáreň, sťahovanie receptov na varenie, kontrola bezpečnosti vody, riadenie filtračného zariadenia, ovládanie mixéra, kontrola dátumu trvanlivosti potravín, ovládanie mikrovlnnej rúry, regulácia hriankovača a kávovaru, identifikácia potravín, ovládanie umývačky, prepojenie s komunikačným systémom, babysitter, kontrola výskytu dymu v kuchyni, box na programovanú prípravu nápojov, predpríprava surovín na vopred zvolené jedlo, samočistiaca varná doska. Neoddeliteľnou súčasťou kuchyne v inteligentnom dome je nastaviteľný nábytok.

B9 Zavlažovanie (Irrigation)

Zavlažovacím systémom sa zabezpečuje pre flóru okolo inteligentného domu dostatočný prísun vody a živín. Zavlažovanie pomocou vonkajšej klimatizácie možno využiť aj pri horúcich dňoch v lete na terase.

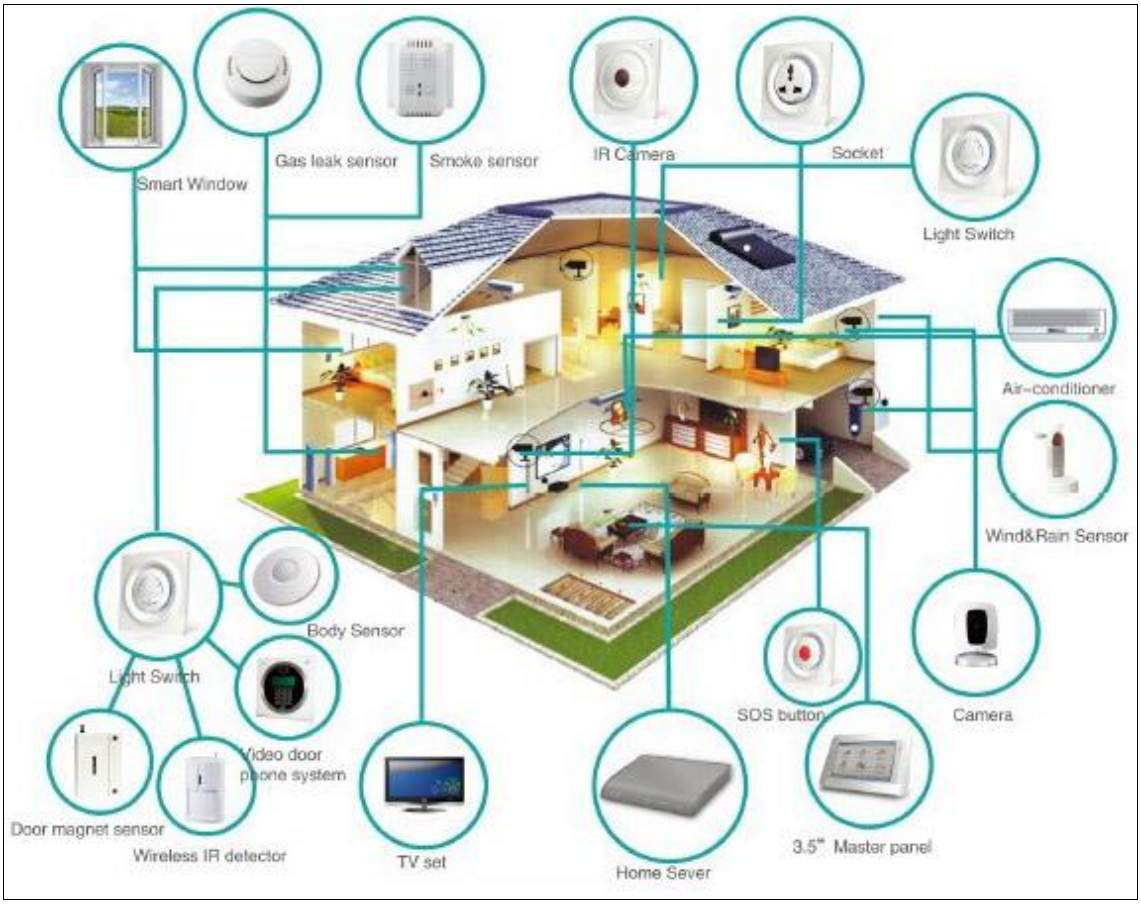
B10 Clean – čistenie a údržba inteligentného domu

Inteligentný dom by mal svojmu užívateľovi zabezpečiť určitý komfort aj tým, že by mu mal zabezpečiť čo najmenej starostí s údržbou a čistotou v dome. Obsahuje preto napr. termokáble, ktoré v zime zabezpečujú odmrazený chodník, príjazdovú cestu,

strechu a odkvapy. Čistotu v dome zabezpečujú rôzne robotické zariadenia, centrálny vysávač prachu, ako aj centrálne čističe bazénov a kúpeľní.

B11 Control – ovládateľnosť inteligentného domu

Inteligentný dom musí byť monitorovaný a ovládaný pomocou zariadení, pričom tu zohráva dôležitú úlohu aj ergonómia a viditeľnosť regulačnej techniky. Ovládanie je možné pomocou telefónu, dotykových panelov, diaľkových ovládačov a spínačov, pripojením sa cez internet alebo hlasom.



Obrázok 4 Inteligentný dom a jeho funkcie

Zdroj: Smart Home Energy In: http://smarthomeenergy.co.uk/what-smart-home

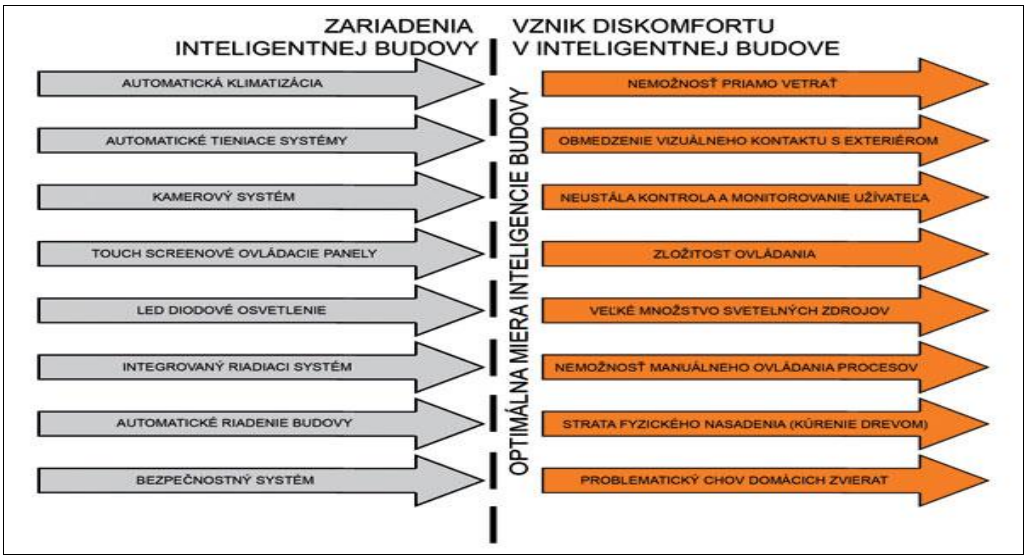
Poslednou veľkou skupinou funkcií inteligentného domu je podľa autorov Hamerníka, Tanušku a Mudrončíka (2012) skupina asistenčných funkcií pre osoby so zdravotným postihnutím.

C) Asistenčné funkcie inteligentného domu pre osoby so zdravotným postihnutím

Aj pre osoby so zdravotným postihnutím navrhujú ich rozdelenie do skupín tak, aby funkcie inteligentného domu boli prispôsobené konkrétnemu typu postihnutia a účinnej pomoci pre obyvateľa domu. Sú to napr. kategórie osôb so zdravotným postihnutím:

- typ športovca,
- typ slepý,
- typ hluchonemý,
- typ invalidný vozík.

Pri výbere funkcií, ktoré bude inteligentný dom zabezpečovať, musí byť dodržaná optimálna miera inteligencie. V prípade prekročenia tejto miery môže byť znížená kvalita bývania obyvateľov domu alebo môže zasahovať do ich života tak, že im miesto komfortu a pohodlia začne vytvárať stres (obr. 5).



Obrázok 5 Optimálna miera inteligencie budovy

Zdroj: Hamerník, Tanuška (2012)

Dôjde tu aj k tomu problému, že zodpovednosť za riadenie domu úplne preberá systém, čoho následkom je vznik tzv. syndrómu master-slave. Znamená to, že majiteľ (pán) domu stráca pocit dominancie vo svojom vlastnom dome, ktorý úplne riadia IT (Hamerník, Tanuška, 2012).

2.2 Existujúce systémy pre inteligentné domy na trhu

V súčasnosti je už na trhu dostatok dostupných systémov pre inteligentné domy. Ich schopnosti, variabilita a flexibilita je rôzna a závisí aj od toho z akých krajín sa na trh dostanú. Napríklad systémy, ktoré sa k nám prichádzajú spoza oceánu, prednostne riešia audiovideo vybavenie, domáce kino a zábavu, pričom majú dokonalé ovládanie pomocou dotykových panelov alebo prostredníctvom TV prijímača. Rozšírením týchto funkcií je ovládanie osvetlenia a zadávanie požadovanej teploty v miestnostiach domu, nastavenie kamier a zabezpečenia domu. Pre konkrétny dom a konkrétne audio video sa jednotlivé funkcionality nakonfigurujú z jednotlivých vopred pripravených funkcií, ktoré sú dostupné v príslušných grafických konfiguračných nástrojoch. Na európskom trhu a u nás sa vyskytuje pri týchto systémoch problém s odlišnými normami pre elektrické inštalácie, dochádza aj k odlišným predstavám projektantov aj koncových užívateľov o ich využití. Často sú potom doplňované cez komunikačný kanál ďalšími zariadeniami, ktoré sú naprogramované na špeciálne doplňujúce úlohy ako napr. bezdrôtové ovládanie vykurovania alebo koordinovanie vykurovacích systémov v dome.

Niekedy sú prvkami domácej automatizácie doplňované zabezpečovacie ústredne a prístupové systémy, kde po pridaní niekoľkých výstupných relé, vypínačov alebo teplomerov je možné nastaviť jednoduchú logiku ich spínania. Existujú tiež systémy, ktoré vychádzajú z vlastnej prepracovanej regulácie vykurovania rozšírenej o niekoľko jednoduchých aktorov a senzorov. Tieto systémy sú voľne programovateľné, keďže však majú obmedzený počet funkcií, môže byť problematické ich ďalšie rozširovanie v budúcnosti či prípadný softwarový upgrade.

Na trhu sú dostupné systémy, ktoré sú pre automatizáciu budov a obytných budov priamo určené. Odlišujú sa tým, že každý z nich je postavený na iných základoch, pričom koncový užívateľ tieto odlišnosti ani nemusí postrehnúť. Rozdiely sú viditeľné pri ich montáži a nastavovaní podľa užívateľských požiadaviek. Niektoré zo systémov sú aj štandardizované normami – ide najmä o systém KNX, ktorý má pôvod v Európe a LON, ktorý má pôvod v USA (Klaban, 2012).

2.2.1 Najčastejšie používané systémy pre inteligentné domy v súčasnosti

Systém KNX (EIB Instabus)

Otvorený zbernicový systém KNX s decentralizovanými inteligentnými prvkami je pokračovateľom európskej inštalačnej zbernice EIB instabus. V tomto systéme má každé pripojené zariadenie svoju vlastnú riadiacu jednotku v podobe mikroprocesora. Všetky zariadenia predstavujú rovnocenné zbernicové prístroje (tzv. multimaster systém).Prenos informácií sa uskutočňuje prostredníctvom zbernice priamo medzi jednotlivými zariadeniami bez nutnosti existencie a aplikácie centrálnej riadiacej jednotky, ako je to v prípade centralizovaných systémov. V systéme KNX sú k dispozícii mnohé zariadenia na ovládanie takmer všetkých funkcií pre moderný inteligentný dom ako sú napr. rôzne snímače, tlačidlové panely, riadiace moduly, zariadenia pre ovládanie svetla, žalúzií, kúrenia, vetrania, a ďalšie. Je podporovaný veľkými výrobcami, ako napr. ABB, Jung, Gira, Merten, SchneiderElectric, Siemens a ďalšími (http://www.knx.org/knx-members/list/).

Systém LON

Systém je postavený na technológii LON firmy Echelon z USA, ktorý našiel široké uplatnenie hlavne v automatizácii budov. LON je skratka pre Local Operating Network a má tú zaujímavú vlastnosť, že nemá centrálnu riadiacu jednotku a všetky pripojené LON zariadenia môžu byť od rôznych výrobcov. Každý prvok na LON zbernici má tzv. neurónový čip s jednoznačnou identifikáciou, v ktorom beží osobitný program určený len pre toto zariadenie. LON predstavuje otvorený štandard a výrobcovia si vyvíjajú vlastné LON zariadenia. (Stanek, In : Bložoň , 2008).

Systém INELS

Systém iNELS (Intelligent Electroinstallation System)od českej firmy ELKO EP predstavuje základ pre ovládanie elektroinštalácie v inteligentnom dome. Jadro tohto systému tvorí centrálna jednotka CU2-01M s rozhraním do ethernetovej siete (pre pripojenie PC, NB, PDA, ovládanie cez internet), ďalej napájacia tlmivka BPS2-O2M pre pripojenie zdroja 27V a v prípade potreby externý master modul zbernice MI2-O2M. Všetky senzory (napr. vypínače, tlačidlá, teplotné, či pohybové senzory) a aktory (napr. spínače svetiel a vykurovania, ovládače žalúzií, stmievania) sú prepojené

prostredníctvom zbernice medzi sebou a sú pripojené do centrálnej jednotky. Všetky funkcie sa potom nastavujú pomocou PC práve v centrálnej jednotke. Tieto funkcie sú veľmi variabilné a je možné ich v budúcnosti kedykoľvek meniť (http://www.inels.sk/).

Systém CYBRO

Základom systému CyBro je voľne programovateľná riadiaca jednotka CyBro, na ktorú možno prostredníctvom zbernice CAN cez protokol IEX pripojiť rôzne moduly. Samotná jednotka CyBro môže mať vlastné vstupy a výstupy a komunikačné rozhrania, z nich najdôležitejším je Ethernet. A-bus protokol umožňuje cenovo efektívne riešenie pre prepojenie riadiacich jednotiek CyBro, dotykových panelov a PC s vizualizáciou. Kompatibilita na iné systémy automatizácie a zabezpečenia (KNX, BACnet, LON, ZigBee) je dosiahnutá cez príslušné brány. Systém sa využíva predovšetkým ako riešenie na automatizáciu elektrických zariadení v IB a v oblasti priemyselnej automatizácie, môže však byť využívaný aj v obytných budovách (http://www.arisys.sk/sk/Produkty-a-riesenia.alej, http://www.cybrotech.co.uk/)

Ako sme už uviedli na trhu existuje množstvo riešení od mnohých výrobkov, ktoré sa od seba odlišujú rozsahom a najmä mierou ponúkaného komfortu. V závislosti od toho v súčasnosti existujú riešenia so zabezpečením základných funkcií inteligentného domu začínajúce na cene cca 1.000,- Eur. V prípade požiadavky zabezpečenia viacerých funkcií sú to sumy okolo 3.000,- až 5.000,- Eur, čo je v našich podmienkach ešte stále pomerene finančne náročná položka pre väčšinu užívateľov (obyvateľov) domu (bytu) (Skyva, 2013).

Z tohto dôvodu nás zaujala možnosť využitia miniatúrneho počítača Raspberry Pi návrhu na vytvorenie vlastného otvoreného systému pre inteligentný dom so zabezpečením niektorých základných funkcií s možnosťou jeho rozširovania do budúcnosti.

2.3 Využitie Raspberry Pi pri návrhu na zabezpečenie funkcií inteligentného domu

Raspberry Pi (ďalej aj RPi) je miniatúrny počítač, ktorý má veľkosť ako kreditná karta (t.j. 85x55 mm), má minimálnu spotrebu (príkon 2,5 až 3,5 W) a je cenovo prístupný širokej verejnosti. Jeho cena sa v slovenských e-shopoch pohybuje od cca 42 do 56 Eur (12/2013). Vyvinutý bol britskou nadáciou Raspberry Pi Foundation s cieľom podpory výučby informatiky na školách. Ako základný prvok technologickej stavebnice predstavuje relatívne výkonný počítač, ktorý je možné ďalej rozširovať o rôzne periférie a vytvárať tak množstvo rôznych projektov.

Obrázok 6 Mikropočítač Raspberry Pi
Zdroj: Honek(2012)

RPi predstavuje riadiaci mikropočítač, ktorý je potrebné vždy prvotne nastaviť a naprogramovať na konkrétnu aplikáciu. Jeho jadro predstavuje Broadcom BCM2835 systém na jednom čipe, ktorý zahŕňa ARM1176JZF-S 700 MHz procesor a grafické jadro VideoCore IV. Keďže je použitý ARM procesor (používaný pre mobilné telefóny a tablety), ako systém nie je možné využiť Windows, k dispozícii je však niekoľko linuxovských distribúcií, z ktorých je doporučovaný systém Raspbian. Ide o špeciálne upravenú distribúciu Debianu, ktorá je optimalizovaná pre tento malý počítač a zvládne aj pretaktovanie (Janeček, 2012). RPi neobsahuje žiadne pohyblivé časti, čo zvyšuje jeho odolnosť voči vibráciám.

Systém sa inštaluje a ukladá na SD kartu, delí sa o 256 MB alebo 512 MB RAM s grafickou kartou. Napájanie 5V s maximálnym nutným príkonom 3,5 W je dodávané cez mikro USB alebo cez piny (GPIO) na doske. Cez GPIO konektor a ďalšie I/0

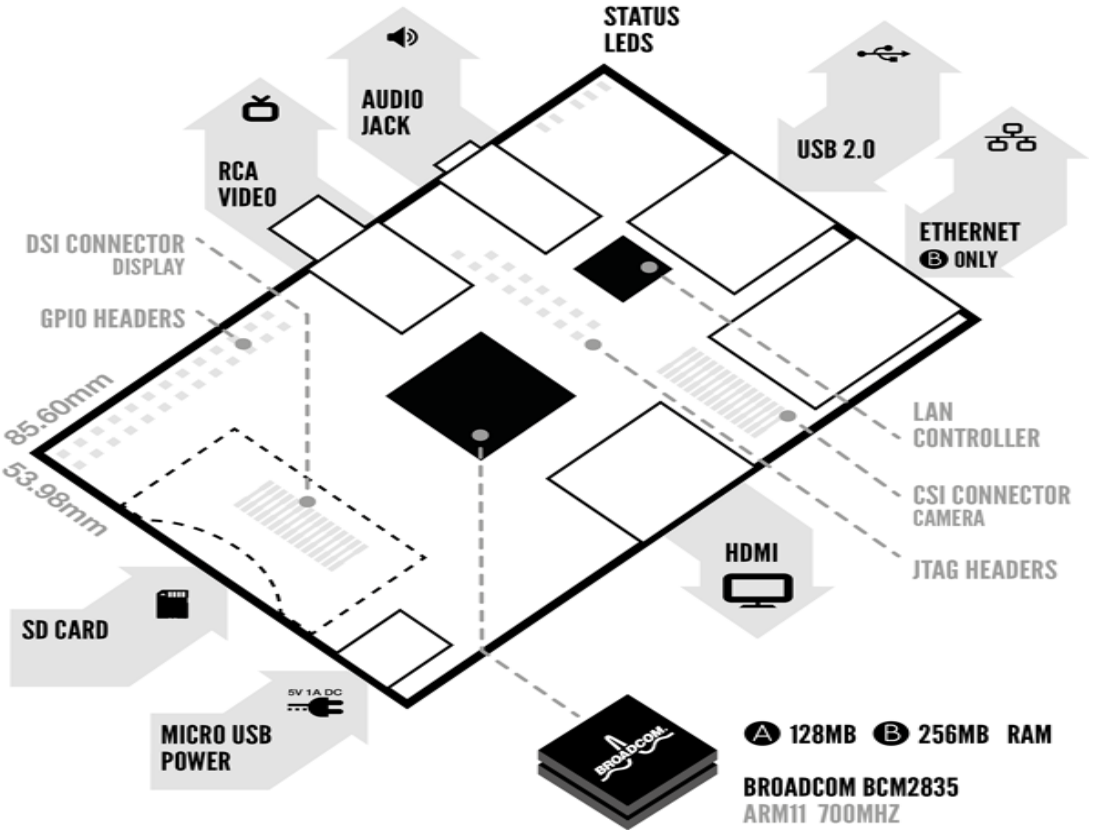
rozhrania je možné RPi prepájať na rôzne ďalšie periférie a komunikovať s nimi. Z najbežnejších rozhraní tu nájdeme HDMI, 3,5 jack audio out a RCA video výstup, dva USB 2.0 porty a Ethernet.

RPi neobsahuje žiadny vypínač, stačí ho zapojiť cez microUSB kábel buď do adaptéra (minimálne sa odporúča 1000mA) prípadne do akéhokoľvek USB. Ihneď po pripojení napájania sa začne z vloženej SD karty zavádzať operačný systém. Užívateľ je o tom informovaný prostredníctvom niekoľkých svetelných indikátorov, z ktorých jeden hlási, že sa úspešne zavádza operačný systém a ostatné indikujú rýchlosť pripojenia sieťovej karty a jej komunikáciu. V prípade poškodenia systému je potrebné SD kartu z RPi vybrať, opakovane vložiť do čítačky (alebo priamo do notebooku) a systém nahrať odznova.

Ak je k RPi pripojená klávesnica, myš a monitor môže slúžiť ako plnohodnotné PC. Výrobca uvádza, že jeho výkon je dostatočný aj pre sledovanie videa vo Full HD kvalite.

2.3.1 Technická špecifikácia Raspberry Pi Model B

- SoCBroadcom BCM2835 (CPU, GPU, DSP, and SDRAM)
- CPU: 700 MHz ARM1176JZF-S core (ARM11 family)
- GPU: BroadcomVideoCore IV, OpenGL ES 2.0, 1080p30 h.264/MPEG-4 AVC high-profile decoder
- Memory (SDRAM): 256 Megabytes (MiB)
- Video outputs: Composite RCA, HDMI
- Audio outputs: 3.5 mm jack, HDMI
- On board storage: SD, MMC, SDIO card slot
- 10/100 Ethernet RJ45 on board network
- Storage via SD/ MMC/SDIO card slot (Barták, 2014).



Obrázok 7 Raspberry Pi – schéma

Zdroj: Dostupné na: <http://www.raspi.cz/2011/12/co-je-raspberry-pi/raspberry-pi-ab/>

2.3.2 RPi GPIO konektor

RPi GPIO je vstupno/výstupný konektor, ktorý je možné ovládať programom a prostredníctvom ktorého RPi komunikuje s okolím. de o 26 pinový port tvorený dvoma radmi po 13 pinov, ktoré sa dajú ovládať pomocou log 0 (LOW) alebo log 1 (HIGH). Piny sa nastavujú na vstup (IN) alebo výstup (OUT).

Tieto porty teda môžu byť vstupné alebo výstupné, t.j. buď môžu snímať nejaký stav (napr. kontakt či fotobunku) alebo je možné pomocou niečo ovládať (napr. osvetlenie, spotrebiče, tienenie a pod.). Každý pin RPi GPIO konektora má vlastný účel, niektoré piny sa používajú spoločne pri vytváraní súčastí obvodov. Je možné ich rôzne

konfigurovať a meniť funkcie pinov portu. (HalfaCree, Upton, 2013) Rozloženie RPi GPIO konektora zobrazuje obr. 8.

Na RPi GPIO konektor je možné tiež pripojiť rôzne zariadenia, ktoré komunikujú cez I^2C, SPI alebo cez UART (RS232) – čo znamená, že je možné pripojiť rôzne sériové AD prevodníky, displeje, senzory (snímače), ktoré posielajú zistenú hodnotu priamo cez zbernicu (napr. teplomery) alebo je možné pomocou RPi ovládať ďalšie zariadenia, ktoré sú riadené pomocou týchto komunikačných liniek.

Obrázok 8RPi GPIO konektor

Zdroj: Dostupné na: <http://elinux.org/RPi_Low-level_peripherals>

3 PRAKTICKÁ ČASŤ

V predchádzajúcej kapitole sme popísali teoretické východiská, na základe ktorých sme navrhli riešenie na tvorbu cenovo prístupného funkčného systému, ktorý spĺňa základné požiadavky na správu a riadenie inteligentného domu. Systém pre správu a riadenie inteligentného domu sme vyvíjali pre majiteľov rodinného domu, ktorý je využívaný ako víkendový dom, t.j. nie je trvalo obývaný. Našou úlohou bolo pre nich navrhnúť a implementovať taký systém, ktorý bude cenovo dostupný a ktorý zároveň disponuje možnosťami ako uspokojiť požiadavky majiteľov domu na úsporu nákladov, bezpečnosť a požadovanú mieru komfortu.

Hlavný cieľ

Navrhnúť a implementovať programový systém pre Raspberry Pi, ktorý zabezpečí funkcie inteligentného domu. Navrhnúť a vytvoriť používateľské prostredie pre návrh a správu inteligentného domu.

Čiastkové ciele

1. Návrh programového systému pre Raspberry Pi na zabezpečenie funkcií inteligentného domu
2. Implementácia programového systému pre Raspberry Pi na zabezpečenie funkcií inteligentného domu
3. Návrh používateľského prostredia pre návrh a správu inteligentného domu
4. Implementácia používateľského prostredia pre návrh a správu inteligentného domu

3.1 Analýza potrieb pre návrh vybavenia inteligentného domu

Pre vytvorenie návrhu vybavenia inteligentného domu sme najprv analyzovali potreby a požiadavky majiteľov domu, ktorí chceli pre svoj dom zabezpečiť základné inteligentné funkcie na vzdialenú správu a ovládanie niektorých zariadení a systémov (napr. osvetlenie, kúrenie, ap.) bez toho, aby bola potrebná ich prítomnosť vo víkendovom dome.

3.1.1 Popis objektu

Systém na správu a riadenie inteligentného domu sme navrhli pre obytný dom, ktorý nie je trvalo obývaný. Ide o jednopodlažný rodinný dom, ktorý má dve miestnosti, kuchyňu a príslušenstvo. K domu patrí aj garáž a hospodárska budova. Pre jeho majiteľov slúži ako víkendový dom – nachádza sa v menšej podtatranskej obci, ktorá má cca do 2 tis. obyvateľov. Nevýhodou pre majiteľov je to, že v okolí objektu sú tiež neobývané domy, ktoré ich majitelia navštevujú len sporadicky. Nakoľko bol dom už viackrát vykradnutý, jeho majitelia potrebovali riešiť predovšetkým prevenciu proti ďalšiemu možnému vykradnutiu objektu. Okrem škôd, ktoré im pri vlámaní do objektu vznikli, súvisí s týmto aj riešenie vlámania na polícii, ktoré z ich doterajších skúseností neviedlo k žiadnym pozitívnym výsledkom, naopak to bola zbytočná strata času pre nich. Ďalšou potrebou, ktorú musia zabezpečiť pred každou zimou je nutnosť odstavenia vody v objekte, aby nedošlo k zamrznutiu vodovodného potrubia v objekte a následným škodám. Riešený objekt sa nachádza cca 16 km od bydliska majiteľov, preto je pre nich problematické objekt pravidelne navštevovať. V objekte je zriadené internetové pripojenie.

3.2 Identifikácia a návrh funkcionalít inteligentného domu

Pre potrebu analýzy funkcionalít inteligentného domu sme analyzovali doménu víkendového domu a jeho majiteľa, ktorá je popísaná v nasledujúcom scenári.

3.2.1 Scenár

Pre popis scenára nášho projektu z pohľadu užívateľov uvažujeme o niekoľkých variantoch riešenia podľa ročných období, a to v letnom a zimnom období.

SITUÁCIA 1 - ročné obdobie LETO

Opis situácie

Majitelia do svojho víkendového domu chodievajú v letnom období pravidelne počas voľných dní, v pracovných dňoch je dom prázdny, objekt je nestrážený a pretože bol už viackrát vykradnutý, je potrebné jeho zabezpečenie.

1. scenár

1. A Plánovaný pobyt vo víkendovom dome

- majitelia odchádzajú z bytu v meste,

- cesta do víkendového domu na dedine trvá autom cca 20 minút, autobusom cca 45 minút,
- po príchode majitelia vojdú do objektu (otvoria dvere do domu, následne do garáže), skontrolujú elektrické spotrebiče a osvetlenie, kúrenie, prívod vody,
- počas voľných dní vykonávajú práce okolo domu, prípadne relaxujú,
- na konci voľného dňa (víkendu) odchádzajú domov – vypnú elektriku, zavrú prívodný ventil vody, uzavrú okná a dvere na dome, zamknú objekty (dom a garáž) a odchádzajú preč

1. B Plánovaný pobyt vo víkendovom dome s nepredpokladanými komplikáciami

- majitelia odchádzajú z bytu v meste,
- cesta do víkendového domu na dedine trvá autom cca 20 minút, autobusom cca 45 minút,
- po príchode majitelia prídu k objektu – zistia vypáčené dvere do domu, skontrolujú objekt, zistia odcudzenie väčšieho rozsahu rôznych vecí z domu – privolanie polície,
- príchod polície – vyšetrovanie na mieste, po troch mesiacoch trestné stíhanie v prečine krádeže políciou zastavené (páchateľ neznámy),
- oprava vypáčených dverí, spevnenie výstuže zárubní, výmena zámkov na dverách,
- predčasný odchod z plánovaného pobytu vo víkendovom dome – vypnú elektriku, zavrú prívodný ventil vody, uzavrú okná a dvere na dome, zamknú objekty (dom a garáž) a odchádzajú preč,
- spisovanie zápisu o krádeži na polícii v sídle obvodu PZ (policajného zboru) – cca 2 hodiny.

1. C Neplánovaná cesta do víkendového domu, ktorú si vyžiadali okolnosti

- majiteľ(ka) odchádza z bytu v meste,
- cesta do víkendového domu na dedine trvá autom cca 20 minút, autobusom cca 45 minút,
- po príchode majiteľ(ka) vojde do objektu (otvorí dvere do domu, následne do garáže), skontroluje objekt – okná, dvere, elektrické spotrebiče a osvetlenie, kúrenie, prívod vody,
- uzavrie okná a dvere na dome, uzamkne objekty (dom a garáž) a odchádza preč.

SITUÁCIA 2 – ročné obdobie ZIMA

Opis situácie

Majitelia do svojho víkendového domu v zimnom období chodievajú nepravidelne, zväčša počas voľných dní, počas zimného obdobia je dom väčšinou prázdny, objekt je nestrážený a pretože bol už viackrát vykradnutý, je potrebné jeho zabezpečenie. V zimnom období je navyše potrebné zabezpečiť po príchode mrazov uzavretie hlavného prívodu vody do objektu, vypustenie vody z vodovodného potrubia a temperovanie objektu.

2. scenár

2. A Plánovaný pobyt vo víkendovom dome

- majitelia pred plánovaným pobytom vo víkendovom dome zapnú kúrenie, aby bola v dome požadovaná teplota,
- majitelia odchádzajú z domu,
- cesta do víkendového domu na dedine trvá autom cca 20 minút, autobusom cca 45 minút,
- po príchode majitelia vojdú do objektu (otvoria dvere do domu, následne do garáže), skontrolujú elektrické spotrebiče a osvetlenie, kúrenie, prívod vody,
- počas plánovaného pobytu vykonávajú práce okolo domu, prípadne relaxujú,
- na konci pobytu odchádzajú domov – vypnú elektriku, resp. v prípade potreby elektriku nevypínajú, ale nastavia elektrické kúrenie na temperovanie objektu, zavrú prívodný ventil vody, vypustia vodu z vodovodného potrubia, uzavrú okná a dvere na dome, zamknú objekty (dom a garáž) a odchádzajú preč

2. B Plánovaná cesta do víkendového domu, ktorú si vyžiadali okolnosti

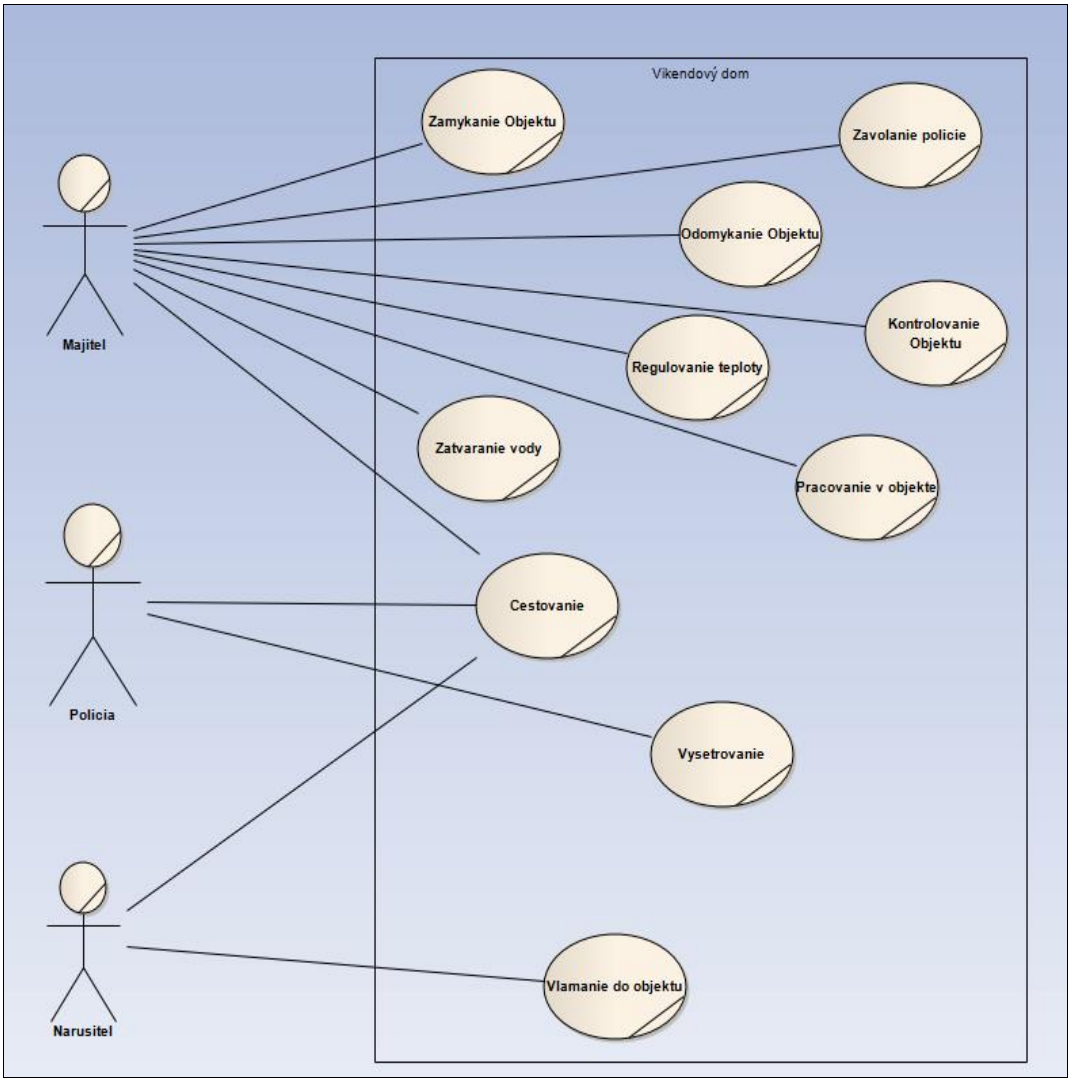
- majitelia v zimnom období nemajú plánovaný pobyt v objekte,
- v období mrazov majiteľ(ka) odchádza z bytu v meste,
- cesta do víkendového domu na dedine trvá autom cca 20 minút, autobusom cca 45 minút,
- po príchode majiteľ(ka) vojde do objektu (otvorí dvere do domu, následne do garáže), skontroluje objekt – okná, dvere, elektrické spotrebiče a osvetlenie, kúrenie, prívod vody,
- nastaví elektrické kúrenie na temperovanie objektu, zavrie prívodný ventil vody, vypustí vodu z vodovodného potrubia,

- uzavrie okná a dvere na dome, uzamkne objekty (dom a garáž) a odchádza preč.

2. C Neplánovaná cesta do víkendového domu, ktorú si vyžiadali okolnosti

- majiteľ(ka) odchádza z bytu v meste,
- cesta do víkendového domu na dedine trvá autom cca 20 minút, autobusom cca 45 minút,
- po príchode majiteľ(ka) vojde do objektu (otvorí dvere do domu, následne do garáže), skontroluje objekt – okná, dvere, elektrické spotrebiče a osvetlenie, kúrenie, prívod vody,
- uzavrie okná a dvere na dome, uzamkne objekty (dom a garáž) a odchádza preč.

Na základe týchto scenárov sme vypracovali Business Use Case domény. V doméne sme určili troch aktorov (majiteľ, polícia a narušiteľ), ktorí vykonávajú hlavne činnosti, ktoré sú v Business Use Case znázornené – pozri obr. 9.



Obrázok 9 Business Use Case domény

Našou úlohou bude eliminovať výskyt aktorov „polícia“ a „narušiteľ“, ako aj odstrániť činnosti „cestovanie“ a niektoré činnosti umožniť realizovať pomocou elektronického systému.

3.2.2 Požiadavky na inteligentný systém riadenia a správu domu a návrh riešenia

Základné požiadavky majiteľov na inteligentný systém riadenia a správu ich víkendového rodinného domu boli:

- náklady na jeho realizáciu
- cenová výhodnosť – rýchla návratnosť investície,
- vzdialená správa objektu s možnosťou obrazového monitoringu objektu,
- úspora energií,
- pasívna bezpečnosť objektu,
- stabilita systému,
- užívateľský komfort.

Na zabezpečenie týchto požiadaviek sme navrhli tieto funkcionality systému:

- monitoring nežiaduceho pohybu v objekte,
- možnosť monitoringu objektu s prenosom obrazu (web kamery),
- zapnutie a vypnutie alarmu (sirény) na diaľku,
- monitoring teploty v dome a upozornenie na kritickú teplotu – možnosť rozšírenia o zavretie ventilu na prívod vody do objektu,
- vypnutie a zapnutie systému pomocou miniatúrneho wifi zariadenia (kľúčenky), príp. cez RFID kartu,
- spôsoby upozornenia pri narušení – SMS, email a upozornenie na klienta,
- systém je ďalej variovateľný a je možné jeho rozšírenie o ďalšie funkcionality podľa prípadných nových požiadaviek užívateľov v budúcnosti.

Cenová náročnosť

Ako sme už spomínali riešenie, ktoré sme navrhli, je s použitím RPi nákladovo veľmi priaznivé v porovnaní s riešeniami, ktoré sú v súčasnosti na trhu. Predpokladaná cenová náročnosť predstavovaného programového riešenia, ktoré je súčasťou systému sa pohybuje v sume cca 200 Eur (ide len o sumu na nákup potrebných komponentov systému).

3.2.3 Návrh riešenia

V navrhovanom systéme inteligentného riadenia a správy domu s funkcionalitami uvádzanými v podkapitole 3.2.2 bude musieť spolupracovať niekoľko druhov zariadení. Ich činnosti sa od seba rozlišujú – v návrhu riešenia sme ich rozdelili do nasledujúcich kategórií:

1. **typ zariadení**

zariadenia nadobúdajúce hodnoty I/0; Z/V – pohybové, svetelné senzory

2. **typ zariadení**

zariadenia 220 V, ktoré zapína a vypína užívateľ – motory, svetlá, rolety, žalúzie

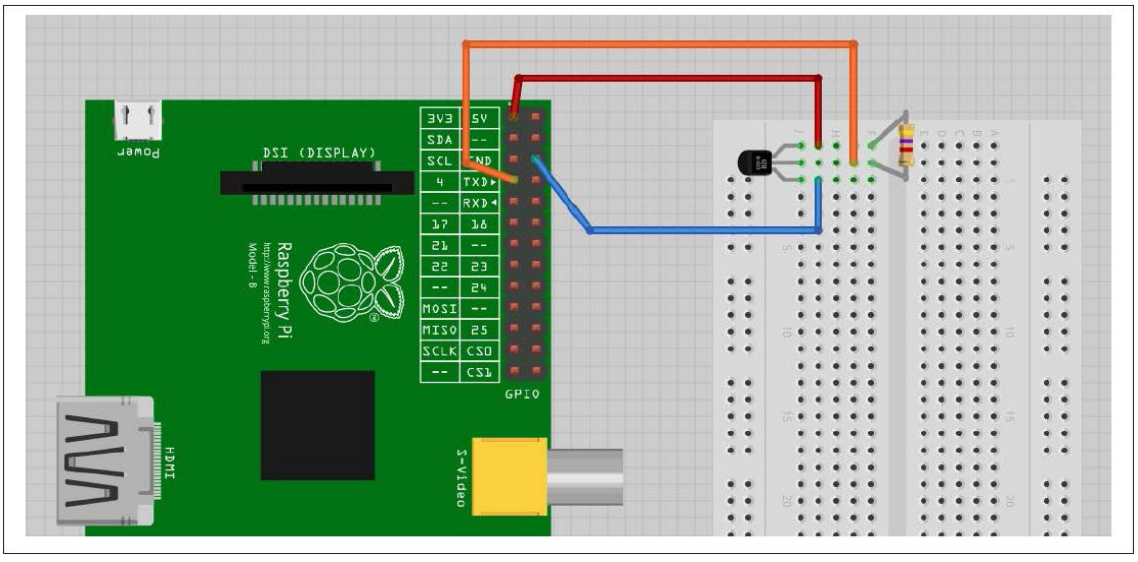
3. **typ zariadení**

zariadenia nadobúdajúce hodnoty I/0; Z/V + zabezpečujú prenos dát – plynové, teplotné senzory

4. **typ zariadení**

zariadenia s komunikačným protokolom – web kamera, RSID čítačka, GSM modul

Tieto zariadenia sú napojené na centrálu (centrálnu riadiacu jednotku), ktorú v našom návrhu predstavuje Raspberry Pi – bližšie sme ho popísali v teoretickej časti práce (str. 29 až 32)



Obrázok 10 Pripojenie teplotného senzora na RPi d

Zdroj: Monk (2014, str.318)

Hodnoty zo zapojených zariadení sa pokúsime načítať v programovacom jazyku JAVA.

Komunikácia

Komunikácia v sieti prebieha pomocou správ. Tieto užívateľské správy poskytujú užívateľovi informácie pre riadenie celej siete.

Formát správ

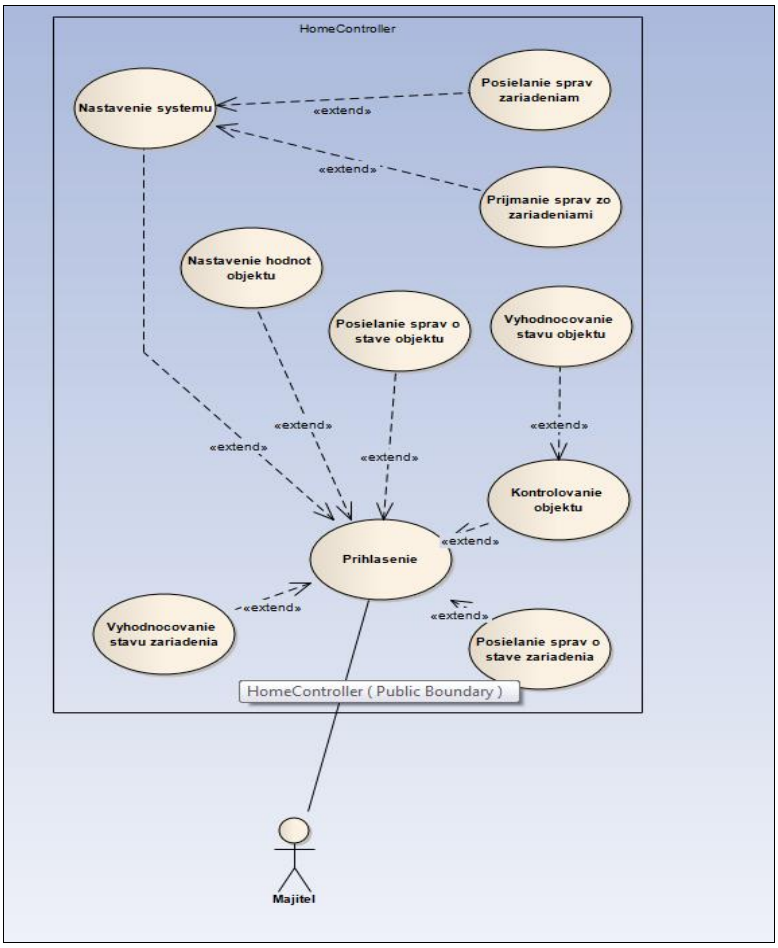
Správy medzi užívateľom a centrálou majú formát serializovateľných objektov JAVA. Správy medzi centrálou a zariadeniami typu 1. a typu 2. majú formát boolean hodnôt (I/0). Pre správy medzi centrálou a zariadeniami typu 3. a typu 4. sa používajú špeciálne komunikačné protokoly pre konkrétny typ zariadenia, napr. GSM modul – komunikačný protokol RXTX.

Prenos správ

Prenos správ medzi systémom a zariadeniami prebieha pomocou zberníc Raspberry Pi – USB, L2C, UART, SPI, GPIO. Užívateľ na komunikáciu so systémom používa ethernetové pripojenie na JAVA server (SSLSocket), ktorý beží na centrálnom zariadení (Raspberry Pi).

Use Case navrhovaného systému

Po analýze súčasného stavu navrhovaný systém pre inteligentný dom by mal spĺňať tieto funkcionality (obr. 11):



Obrázok 11 Use Case Home Controller

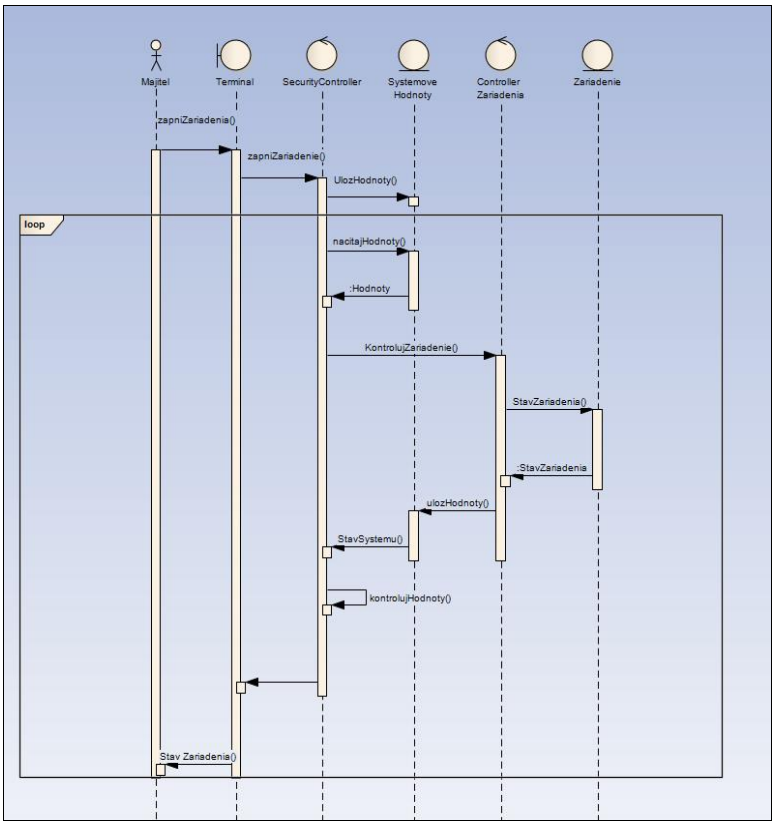
3.2.4 Základné funkcionality systému

Z prezentovaného Use Case systému vyplynuli tri základné požiadavky na systém:

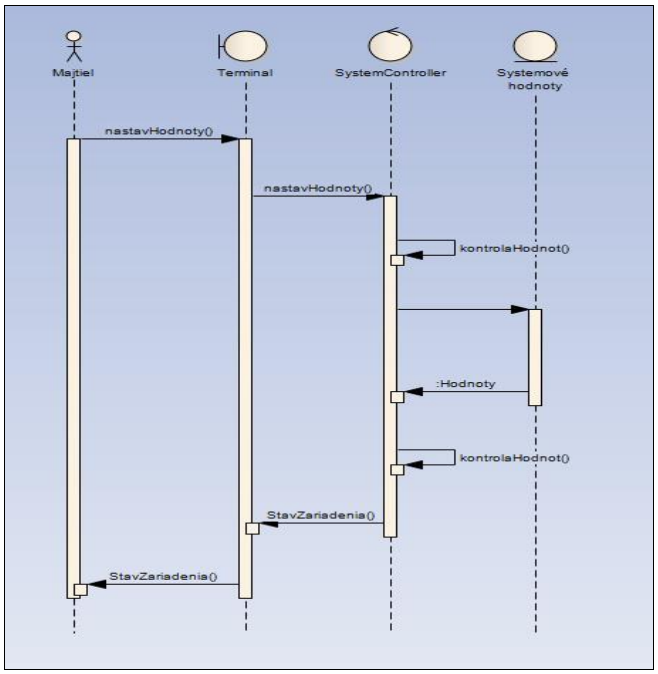
1.) komunikácia: majiteľ – systém (obr. 12)

2.) komunikácia: systém – zariadenia (obr. 12)

3.) kontrola systému podľa nastavených hodnôt (obr. 13)



Obrázok 12 Sekvenčný diagram komunikácie majiteľ – systém a systém – zariadenia



Obrázok 13 Sekvenčný diagram zmeny systému

3.3 Realizácia navrhnutého riešenia

Systém sme vyvíjali pomocou modifikovanej agilnej metodiky Scrum. Ide o iteratívny a inkrementačný spôsob vývoja, pri ktorom je vývoj rozdelený do jednotlivých cyklov, tzv. šprintov, počas ktorých by mali byť dokončené stanovené úlohy (splnené ciele) v tom ktorom časovom úseku. Jednotlivé šprinty nasledujú jeden za druhým bez prestávky, kým nie je vývoj systému (softvérového projektu)dokončený. Ako vývojové prostredie bolo zvolené NetBeans, ako ďalšie vývojové nástroje boli použité PuTTY, WinSCP, Eclipse.

V zmysle spomínanej metodiky Scrum sme systém rozdelili na jednotlivé šprinty, ktorých výsledkom bola funkčná časť projektu. Spojenie viacerých častí (šprintov) projektu sme nazvali etapou vývoja systému.

Na začiatku každého šprintu sme analyzovali požiadavky na časť systému a určili sme cieľ, ktorý má daný šprint splniť. Na začiatku bol tiež určený aj spôsob testu, ktorým sa bude overovať, či bol určený cieľ splnený.

Beh šprintu (priebeh)

Podľa analýzy a požiadaviek sme vytvorili daný systém. Počas šprintu sme zapisovali ďalšie možné požiadavky. Šprint mohol byť kedykoľvek ukončený z dôvodu potreby novej analýzy. Jednotlivé šprinty mohli byť vyhlásené za úspešné a ukončené až vtedy, keď úspešne prešiel určený test. Na konci šprintu boli spísané nové požiadavky a pripomienky k systému. Systém sa vyvíjal naraz aj po softvérovej aj po hardvérovej stránke, čo značne sťažovalo návrh riešenia pre jednotlivé časti analýzy.

3.3.1 Prvá etapa vývoja systému pomocou modifikovanej agilnej metodiky Scrum

V prvej etape sa pokúsime pripraviť RPi ako hlavný Controller a overiť možnosti potrebných funkcionalít RPi.

1. **Šprint**

Cieľ

Pripraviť RPI ako hlavný controller pre smart home.

Priebeh

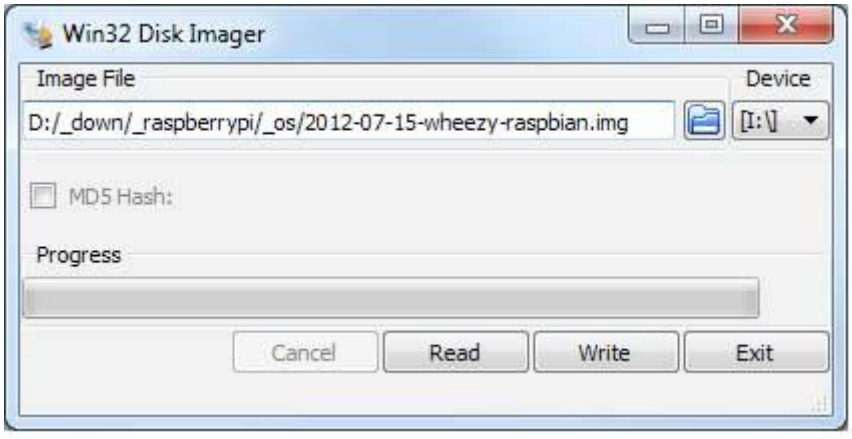
Inštalácia a príprava OS pre Raspberry PI

Pre Raspberry Pi je k dispozícii viacero linuxových distribúcií, pre zoznámenie sa s Raspberry Pi je doporučovaná distribúcia Raspbian (Horák, 2012).Raspbian

predstavuje distribúciu, ktorá je založená na testovacej vetve Debianu a ako desktopové prostredie používa LXDE (Lightweight X11 Desktop Environment). Ide o rýchle a hardvérovo nenáročné prostredie, ktoré sa funkcionalitou podobá IceWm, navyše je však nastaviteľné prostredníctvom GUI. Na správu okien používa OpexBox.

V nasledujúcich krokoch popisujeme základný postup oživenia systému až po nastavenia potrebných knižníc.
Z internetu stiahneme najnovšiu verziu operačného systému Rasbian pre Raspberry Pi (http://www.raspberrypi.org/downloads/). Tento OS nahráme na SD kartu cez PC a následne ho vložíme do Raspberry Pi. Postupnosť možno zhrnúť do nasledovných krokov:

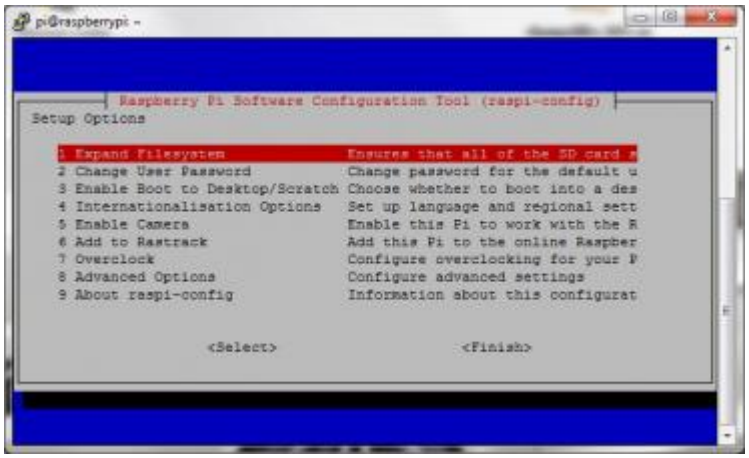
• stiahneme image RaspbianWheezy
• rozbalí sa súbor
• vložíme SD kartu PC a• stiahneme Win32DiskImager
• spustíme nástroj Win32DiskImager
• vo Win32DiskImager (obr. 13) vyberieme stiahnutý image súboru, ktorý sa predtým rozbalil
• vyberieme písmeno jednotky SD karty
• klikneme na tlačidlo Zapísať a čakáme na zápis
• kartu vložíme do Raspberry Pi a nainštalujeme



Obrázok 14win32diskImager užívateľské rozhranie
Zdroj: Vachálek (2014)

Po vložení SD karty s pripravenou inštaláciou sa spustí inštalácia operačného systému. Dôležitý je výber správnej SD karty. Na trhu je veľké množstvo SD kariet, ale

nie všetky sú vhodné pre Raspberry Pi. SD karty niektorých výrobcov nedokážu bootovať, čím nie je možné korektné spustenie operačného systému.



Obrázok 15Raspi-config konfiguračné prostredie

Zdroj: Vachálek (2014)

Po dokončení inštalácie je potrebné príkazom sudoraspi-config nastaviť parametre operačného systému (obr. 14). Je tu možnosť pretaktovania, nastavenia klávesnice, časového pásma, k dispozícii sú aj ďalšie nastavenia.

Softvér pre RaspberryPi bol v našom prípade primárne naprogramovaný v programovacom jazyku JAVA.

Inštalácia JAVY

sudoapt-getupdate
sudoapt-getinstall openjdk-7-jdk

Test

Spustiť Java mainclass na Raspberry Pi.

```
pi@raspberrypi ~ $ java -version
java version "1.7.0_25"
OpenJDK Runtime Environment (IcedTea 2.3.10) (7u25-2.3.10-1~deb7u1+rpi1)
OpenJDK Zero VM (build 22.0-b10, mixed mode)
pi@raspberrypi ~ $ sudo java -jar hellou.jar
test
```

Záver

Test bol splnený, Raspberry Pi podporuje JAVA 7.

2. Šprint

Cieľ

Vytvoriť JAVA triedu, ktorá bude ovládať zariadenia pomocou GPIO I/O.

Do vytvoreného projektu sme importovali knižnicu Pi4J, ktorá umožnila prístup k zberniciam prostredníctvom programovacieho jazyka JAVA.

Priebeh

```
final GpioController gpio = GpioFactory.getInstance();
final GpioPinDigitalOutput pin = gpio.provisionDigitalOutputPin(RaspiPin.GPIO_01, "MY Device", PinState.HIGH);
System.out.println("--> GPIO state should be: ON");
```

Vytvorili sme JAVA triedu, ktorá obsahovala vyššie uvedený kód.

Test

Po spustení triedy sa LED dióda, ktorá bola pripojená na zodpovedajúcu zbernicu rozsvietila a výsledok sa zapísal na terminál.

Záver

Raspberry PI pomocou knižnice Pi4J umožní v Jave pristupovať na zbernicu, test bol úspešný.

3. Šprint

Cieľ

Vytvoriť triedu, ktorá bude sledovať 6 pohybových senzorov a zaznamenaný pohyb vypíše na terminál.

Priebeh

```
String adr = i.getAddress();
for (int j = 0; j < adr.length(); j++) {
    char a = adr.charAt(j);
    if (a == '0') {
        outPinList[j].low();
    }
    if (a == '1') {
        outPinList[j].high();
    }
}
```

Senzory sme adresovali pomocou štvormiestneho binárneho čísla, ktoré určovalo zapojenie v multiplexore a po nastavení adresy sme načítali stav daného senzoru.

Toto adresovanie sme opakovali v cykle.

Test

Senzory sa rozmiestnia do jednotlivých miestnosti a figurant bude pohybom zapínať pohybové senzory. Pohyb bude vypisovaný na terminál.

Záver

Zariadenia môžu byť zapojené v multiplexore a postupným prechádzaním v cykle dokážeme načítať hodnoty zo senzorov.

4. Šprint

Cieľ

Vytvoriť triedu, ktorá bude odosielať SMS cez USB 3G modem.

Priebeh

Vytvorili sme triedu SMSsender, ktorá pomocou knižnice SMSLib (http://smslib.org/) odosiela SMS správy.

Test

Zariadenie pošle SMS správu na určené telefónne číslo

Záver

SMS bola úspešne odoslaná pomocou knižnice SMSLibpre Javu pomocou RXTX seriálového portu.

5. Šprint

Cieľ

Vytvoriť triedu, ktorá bude posielať email.

Priebeh

Vytvorili sme triedu MailSender, ktorá používa JavaMail API (https://java.net/projects/javamail/pages/Home)a odosiela email na stanovenú emailovú adresu.

Test

Odoslať email na určenú emailovú adresu.

Záver

Test bol úspešný. Email bol odoslaný na určenú emailovú adresu pomocou JavaMail API.

6. Šprint

Cieľ

Vytvoriť systém, ktorý bude kontrolovať pohyb v miestnostiach. V prípade, ak bude zaznamenaný pohyb, pošle email na určenú adresu a SMS na určené telefónne číslo.

Priebeh

Vytvorili sme novú triedu SecurityController, ktorá využíva triedy MailSender a SMSSender na odoslanie správ o pohybe v miestnostiach objektu.

Test

Senzory sa rozmiestnia do jednotlivých miestností a figurant bude pohybom zapínať pohybové senzory. Pohyb bude vypisovaný na terminál a pri treťom zaznamenaní pohybu sa odošle email a SMS určeným adresátom – ukážku kódu pozri príloha 9.

Záver

Test bol úspešne vykonaný. Pohyb sa vypisuje na terminál a pri treťom zaznamenaní odošle email a SMS určeným adresátom.

7. Šprint

Cieľ

Vytvorený systém bude možné obsluhovať a kontrolovať prostredníctvom správ cez terminál.

Priebeh

Vytvorili sme nové vlákno ControllerKeyboard, ktoré umožňuje vstup údajov cez terminál a umožňuje modifikovať hodnoty v predchádzajúcej triede SecurityController.

Test

Senzory sa rozmiestnia do jednotlivých miestností a figurant bude pohybom zapínať pohybové senzory. Pohyb bude vypisovaný na terminál a pri treťom zaznamenaní pohybu sa odošle email a SMS určeným adresátom. Možnosť vypnúť a zapnúť posielanie emailu a SMS podľa potreby.

Záver

Test bol úspešne vykonaný.

3.3.1.1 Zhrnutie výsledkov po prvej etape vývoja

RPI dokáže zariadenia I/O monitorovať a kontrolovať ich zapnutie alebo vypnutie a na základe týchto aktivít posielať správy (výstup správ: text v terminály, SMS, Email) – ukážku kódu pozri príloha 8.

3.3.2 Druhá etapa vývoja pomocou modifikovanej agilnej metodiky Scrum

V tejto etape sme nadviazali na prvú etapu a pokúsime sa dokončiť komunikáciu so zariadeniami, ktoré sú potrebné pre funkcionality domu.

8. Šprint

Cieľ

Poslať na RPI cez sieťovú komunikáciu (socket na ethernetovej sieti) obojstranne správu z osobného PC.

Priebeh

Na RPi sme vytvorili triedu Server, ktorá obsahuje ServerSocketa na strane PC sme vytvorili triedu Klient, ktorá obsahuje triedu Socket a pripája sa na server pomocou IP adresy a portu.

Test

Na terminál výpis správy o prijatí a na PC výpis odpovede.

Záver

Test bol úspešný.

9. Šprint

Cieľ

Modifikovať sieťovú komunikáciu na SSLSocket.

Priebeh

Predchádzajúce triedy sme modifikovali a použili sme SSLServerSocket a SSLSocket.

Na generovanie kľúčov sme použili nástroj Keytool[1]:

sudokeytool –genkey –aliasduke –keystoreke.rs –keyalg RSA –validity 7
keytool –export –aliasmojcert –keystoremoj –rfc –filemoj.cer
sudokeytool –import –aliasmojcert –filemoj.cer –keystoretruststore

Test

Poslať šifrovanú správu - ukážku kódu pozri príloha 3.

Záver

Test úspešne splnený. Správa bola odoslaná v šifrovanej podobe.

10.Šprint

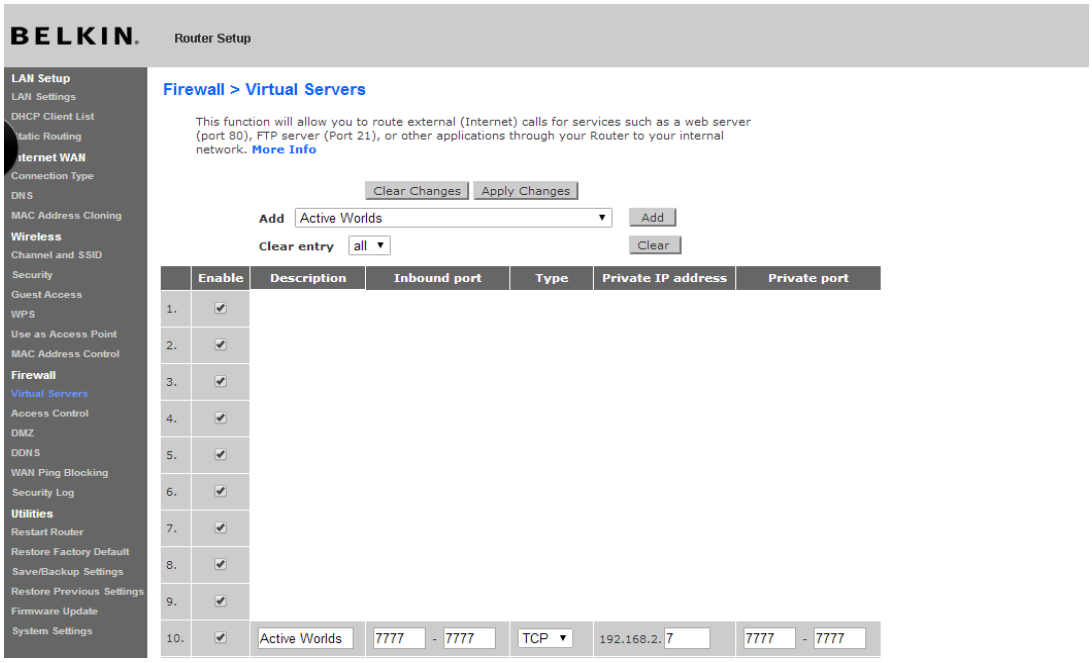
Cieľ

Zopakovať 9.šprint zmenou sieťového prostredia z ethernetu na internetovú komunikáciu.

Priebeh

[1]http://docs.oracle.com/javase/6/docs/technotes/guides/security/jsse/JSSERefGuide.html#Features

Upraviť firewall na strane servera priradiť porty na RPi



Obrázok 16 Úprava firewallu

Test

Poslať šifrovanú správu.

Záver

Test splnený.

11.Šprint

Cieľ

Inštalácia na RPI servise typu dyndns

Priebeh

Vytvorenie konta na www.dynds.org

http://dyn.com/dns/

apt-getinstallddclient

 # /etc/ddclient/ddclient.conf
protocol=dyndns2
use=web
login=mylogin
password=mypassword
myhost.dyndns.org

/usr/sbin/ddclient -daemon 300 –syslog

Test

Zariadenie bude po spustení servisu posielať správu v určenom intervale na adresu ****.dyndns.org

Záver

Zariadenie je možné ovládať pomocou SSH terminálu po zadaní IP adresy, ktorú najprv majiteľ musí zistiť na uvedenej adrese.

12. Šprint

Cieľ

Pripojenie USB web kamery do RPi.

Priebeh

Vytvorili sme triedu CamServer, ktorá inicializuje web kameru a sprístupní streamovanie videa.

Test

Zobrazenie live obrazu pomocou dostupných prehliadačov na PC.

Záver

Test bol úspešný.

13.šprint

Cieľ: Pripojenie 220 V sirény na Raspberry Pi.

Priebeh

Pomocou ovládania relátka cez GPIO sme dokázali cez terminál vypnúť a zapnúť sirénu.

Test: Spustenie sirény cez terminál.

Záver: Test bol úspešný.

14.šprint

Cieľ: Načítanie hodnôt z teplotného senzora.

Priebeh

K senzoru sme museli pristupovať cez C knižnicu, ktorá bola dodaná so zariadením. Knižnicu sme museli upraviť a následne spustiť v JAVE cez JNI framework.

Test: Zobrazenie teploty na terminály – ukážku kódu pozri príloha 11 a príloha 12.

Záver: Test splnený.

3.3.2.1Zhrnutie výsledkov po druhej etape vývoja

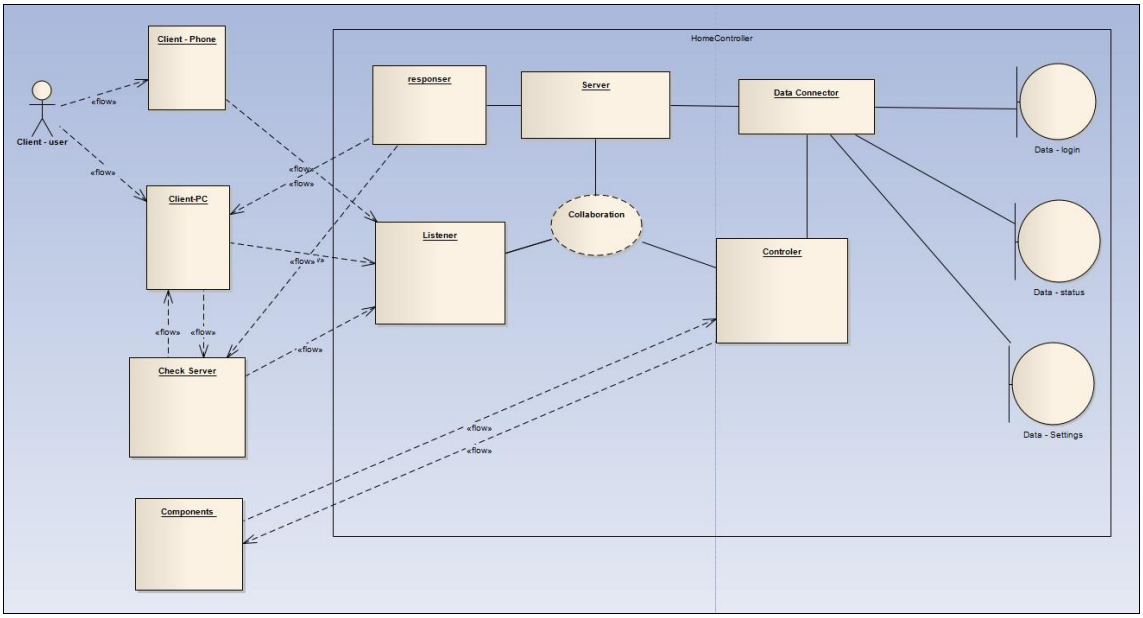
Dosiahnuté výsledky sme po štrnástom šprinte prezentovali majiteľom domu, ktorí boli s nimi spokojní. Zároveň pri tejto prezentácii majitelia vyslovili ďalšie požiadavky na systém:

- nedostatočné ovládanie cez PuTTY konzolu,
- alternatíva k vypínaniu a zapínaniu alarmu cez PC klienta; majiteľom sa pozdávala možnosť vypínať a zapínať alarm cez telefón.

Po tejto etape vývoja systému bola potrebná nová analýza, ktorú ďalej popisujeme.

3.3.3Nová analýza

Po prezentácii výsledkov vývoja systému po jedenástom šprinte sme pre splnenie požiadaviek majiteľov domu navrhli nový doménový model (obr. 17).



Obrázok 17 Nový doménový model na splnenie požiadaviek po 11. šprinte

V novom doménovom modeli vznikli nové časti celého systému, a to:

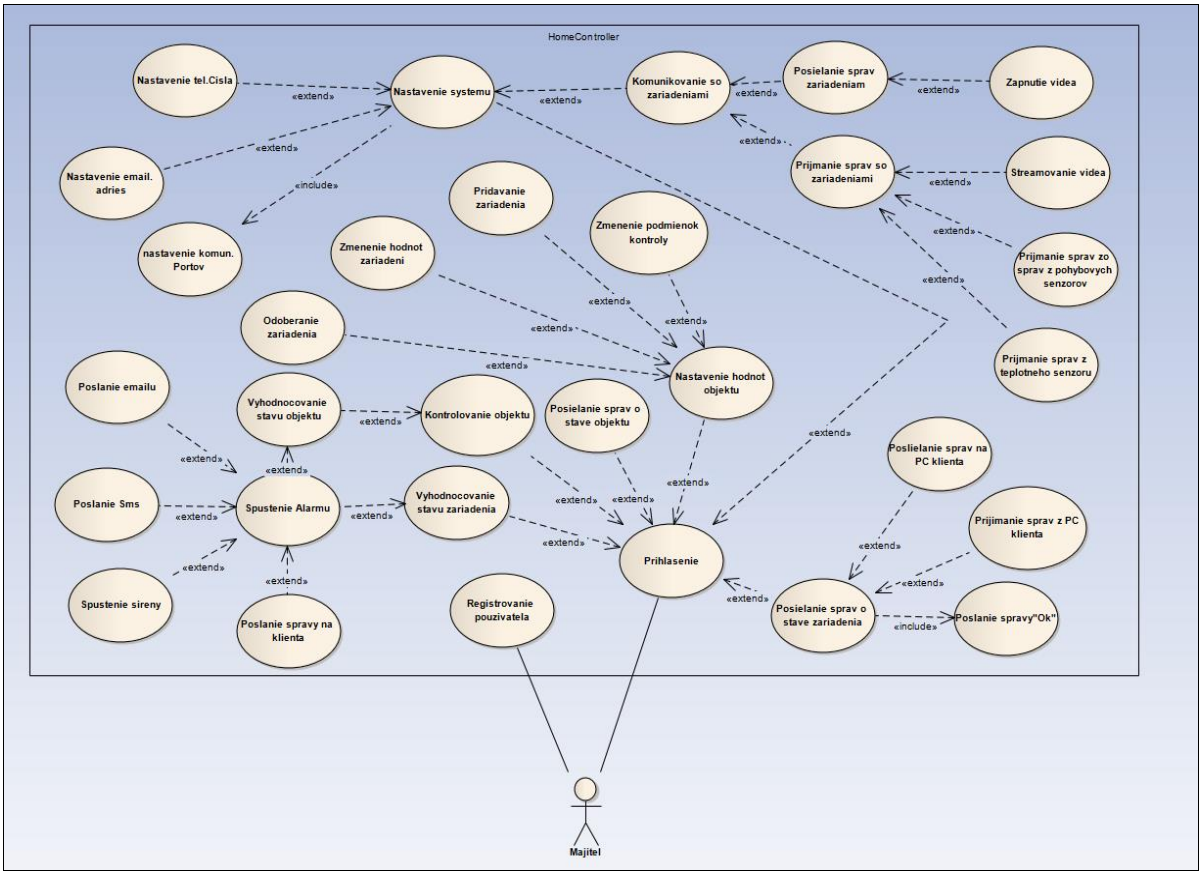
1.) grafický klient na PC, ktorý vizualizuje riadenie a obsluhuje RPi,

2.) Check Server – Server zaznamenáva IP adresu RPi a následne na požiadanie ju odovzdá klientovi na PC. Check Server pri neobdržaní IP adresy kontaktuje majiteľa upozorní na chybu v komunikácii medzi RPi a Check Serverom,

3.) Android klient, ktorý uľahčuje vypínanie a zapínanie alarmu.

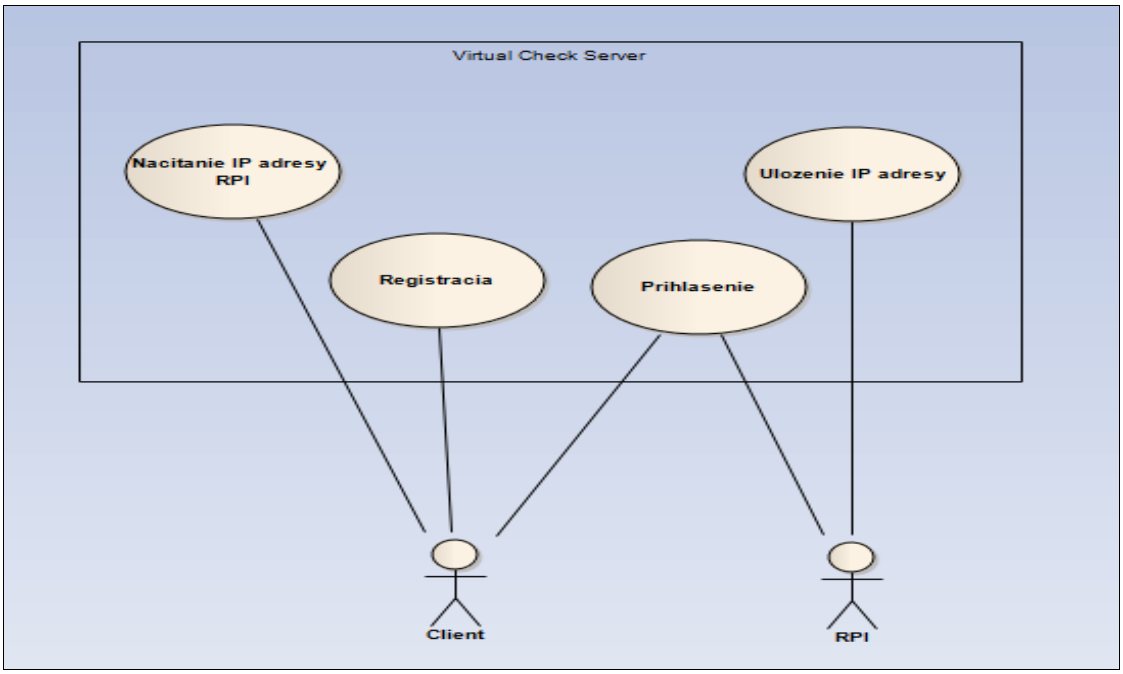
Use Case pre objekty z nového doménového modulu

Pre nové objekty z nového doménového modelu sme vypracovali nové Use Case diagramy (obr. 18 až 21).



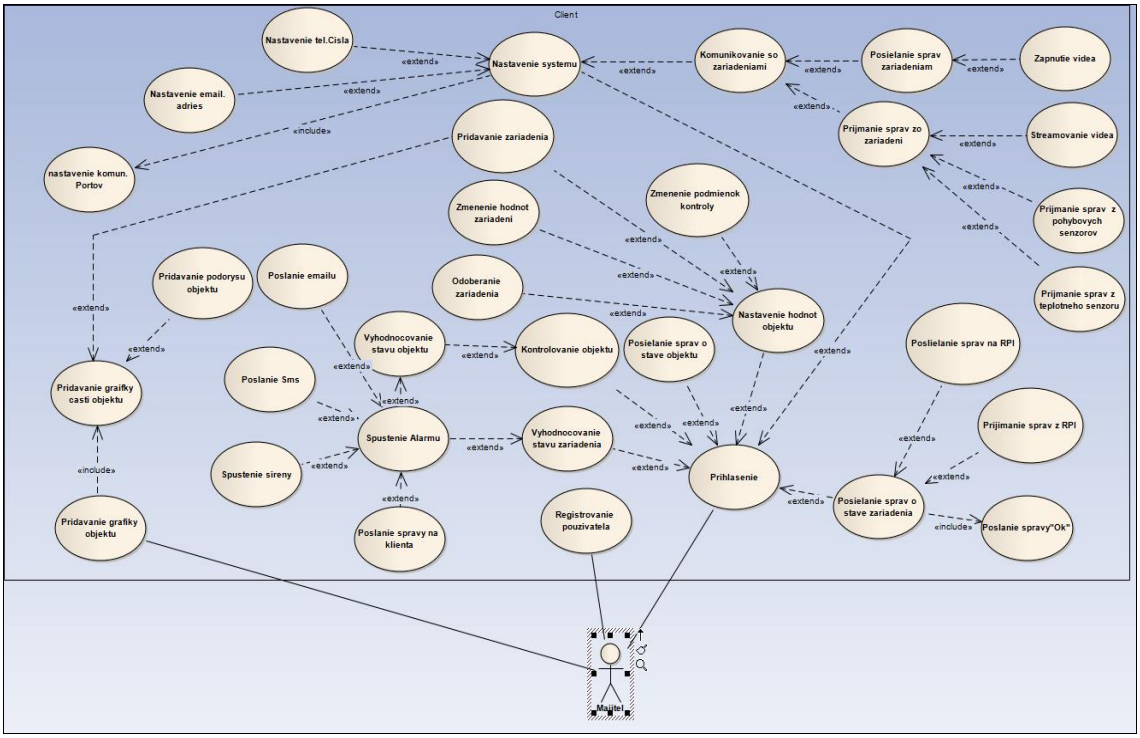
Obrázok 18Nový Use Case pre RPi

Use Case pre RPi zachytáva nové prípady použitia systému pre inteligentný dom.



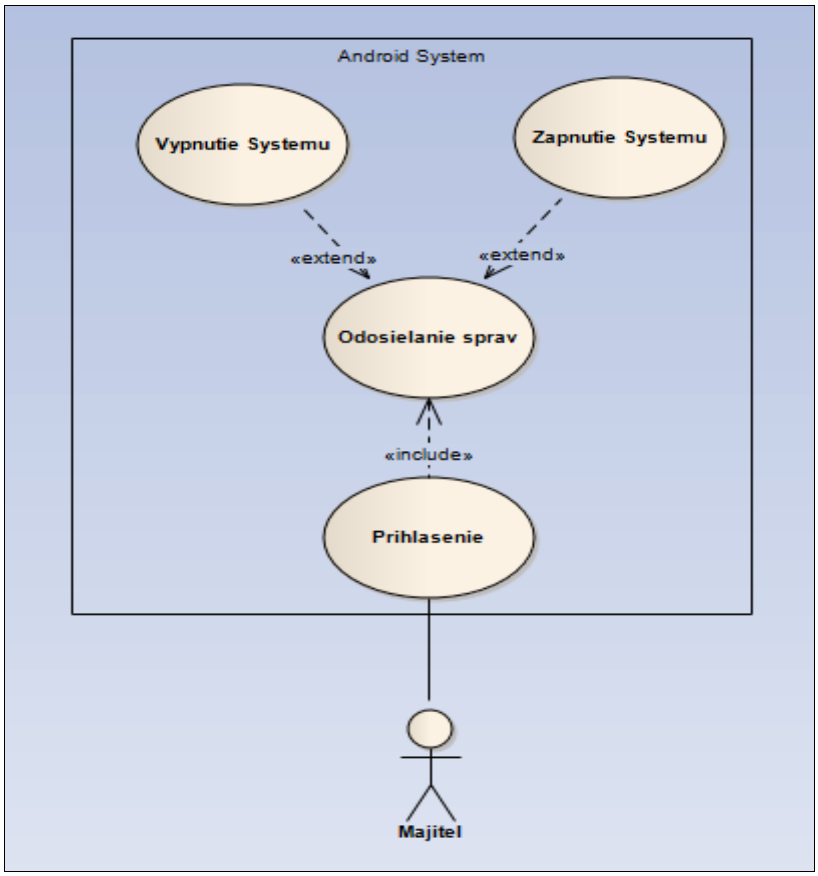
Obrázok 19 Nový Use Case Virtual Check Server

Virtual Check Server bude slúžiť na ukladanie IP adresy RPi a následne na požiadanie klienta túto IP adresu odošle.



Obrázok20 Nový Use CaseClient

Client na PC bude obsahovať grafické ovládanie RPi . Majiteľ bude mať takto zlepšený prístup na obsluhovanie funkcionalít inteligentného domu.



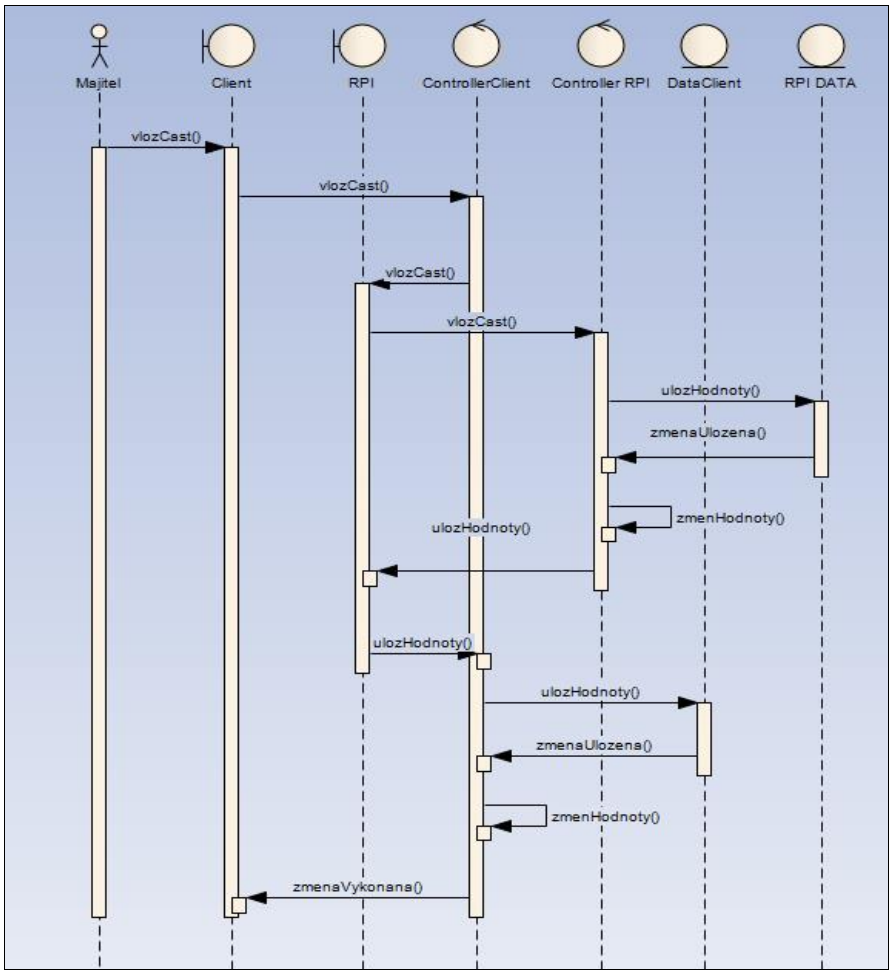
Obrázok 21 Nový Use Case Android Client

Android Client bude slúžiť na automatické zapínanie a vypínanie RPi.

3.3.4 Nové scenáre použitia systému

Systém budú tvoriť štyri samostatné časti, ktoré budú navzájom spolupracovať. Ide o tieto časti:

1. Raspberry Pi Home Controller
2. Client pre PC
3. Check Virtual server
4. Client pre Android



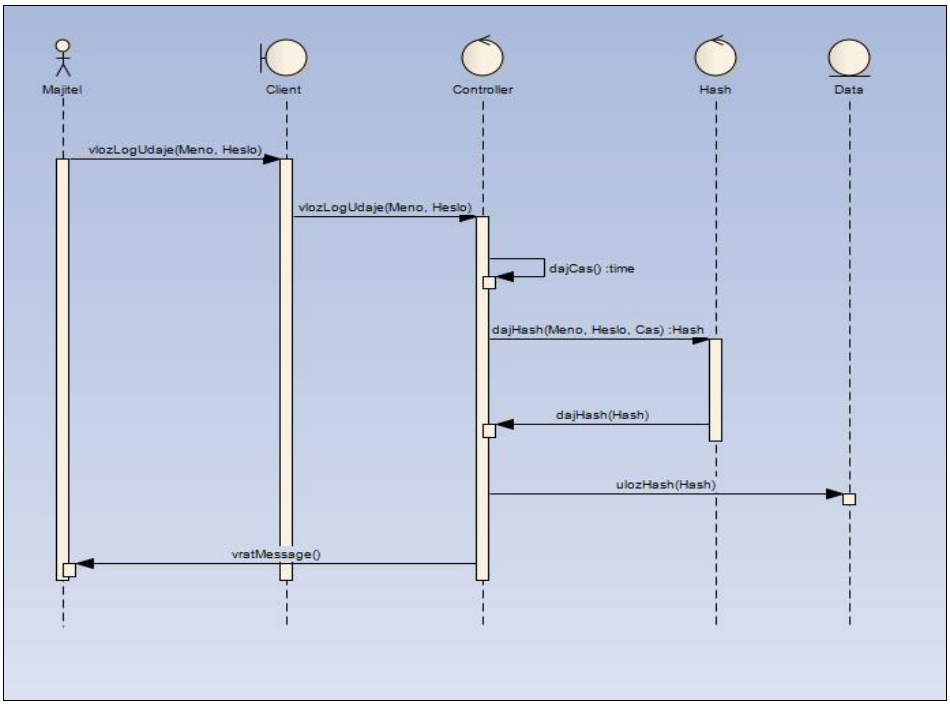
Obrázok 22 Use Case – vkladanie hodnôt do systému

Majiteľ bude pristupovať k zariadeniu pomocou klienta pre PC, ktorý mu bude umožňovať vytvárať, modifikovať a odstraňovať hodnoty objektu v RPi HomeController. Hodnoty sú rozdelené na objektové a systémové. Objektová hodnota sa vzťahuje k reálnemu objektu (napr. počet pohybových senzorov). Systémové hodnoty sa vzťahujú k systémovým nastaveniam (napr. komunikačný port, veľkosť písma, mailová adresa) .

3.3.4.1 Scenár – štart systému

Majiteľ si v klientovi vytvorí účet, ktorý je uložený do súboru. Súbor sa skopíruje cez Total Comander na RPi. Následne majiteľ pomocou príkazu spustí HomeController v termináli. Majiteľ sa prihlási cez logovacie okno na RPi, ktoré overí heslo a vytvorí sa spojenie medzi RPi a klientom. Majiteľ vidí v hlavnom okne, ktoré slúži na prezeranie objektu, prázdny objekt. Majiteľ musí najprv nastaviť objekt cez druhé okno, v ktorom má možnosť nakresliť pôdorys objektu, pridávať a odoberať

zariadenia. Majiteľ musí nastaviť adresy k zariadeniam. Majiteľ môže vypnúť a zapnúť zariadenia. Niektoré zariadenia sa fyzicky nevypínajú, ale systém ich nebude čítať – napr. pohybový senzor. Druhý typ zariadení sa fyzicky vypína – napr. siréna. Majiteľ uloží nastavenia a opustí editovacie okno. V hlavnom okne, ktoré slúži na vizualizáciu synchronizuje vykonané zmeny z RPi, ktoré po obdržaní súboru načíta uložené hodnoty. RPi po tejto synchronizácii začne objekt kontrolovať podľa nastavení.



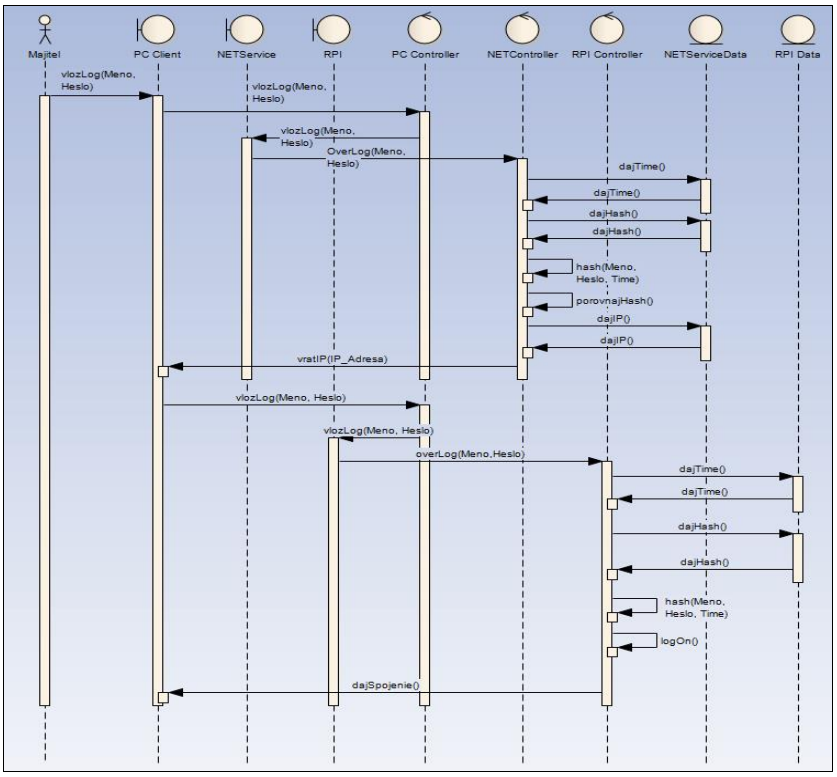
Obrázok 23UseCase - registrácia

3.3.4.2 Scenár – pripojenie klienta z internetu na RPi

Majiteľ v prihlasovacom okne zapne službu Net Servis, zadá meno a heslo, ktoré dostane od poskytovateľa služby. Tento server po pripojení služby poskytne klientovi IP adresu RPi, následne klient znova zadáva meno a heslo, ktoré si zvolil pri registrácii.

3.3.4.3 Scenár – pripojenie klienta z ethernetu na RPi

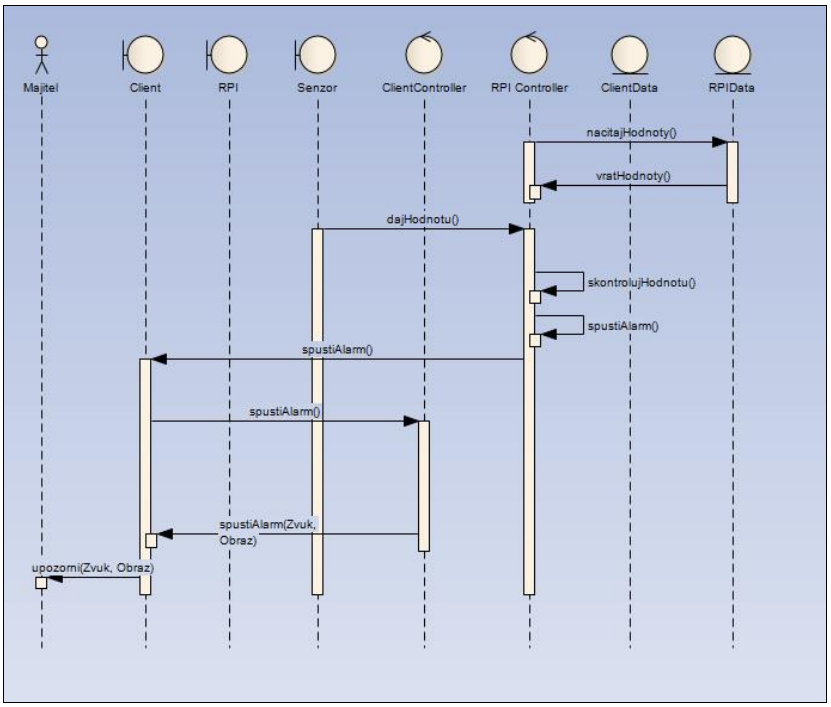
Klient v prihlasovacom okne zadá lokálnu adresu zariadenia, zadá meno a heslo a pripojí sa na zariadenie.



Obrázok 24 Use Case– prihlásenie

3.3.4.4 Scenár – vyvolanie alarmu

RPi v cykle vykonáva kontrolu zariadení.



Obrázok 25 Use Case – vyvolanie alarmu

Pri zistení stavu, kde hodnoty zodpovedajú nastaveniam pre vyvolanie alarmu, RPi spustí posielanie správ v tomto poradí: email, ak je dostupný klient na PC, tak

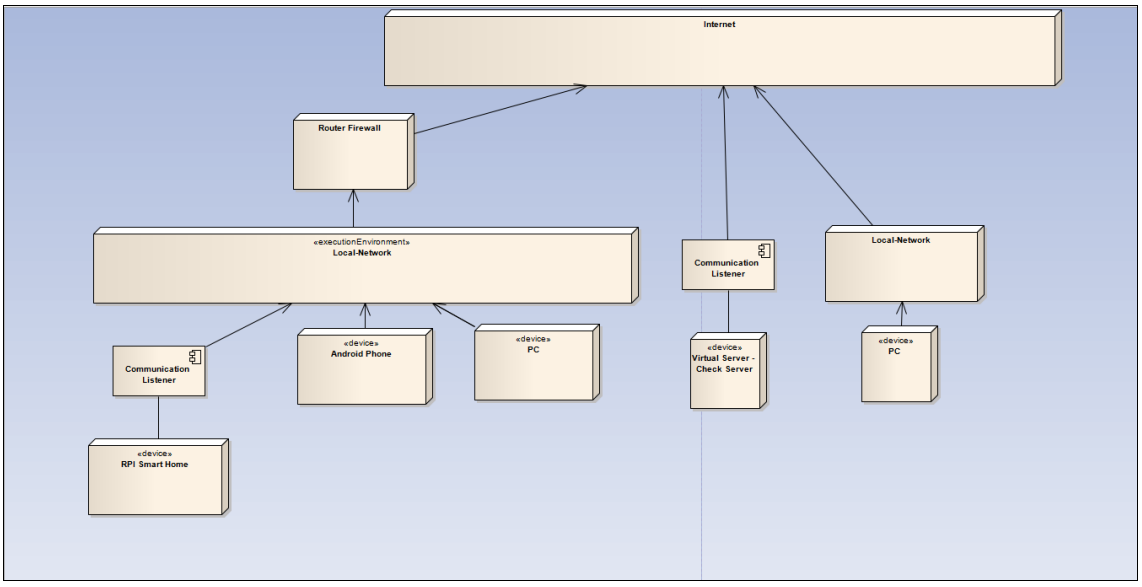
odošle správu na PC klienta, následne sa odošle SMS správa. Majiteľ sa môže pripojiť na web kameru a môže ručne spustiť sirénu.

3.3.4.5 Scenár – vypnutie a zapnutie alarmu prostredníctvom Android Clienta

Majiteľ na telefóne na prihlasovacej obrazovke zadá ethernetovú IP adresu RPi, ktoré automaticky vypne alarm, pokiaľ je telefón pripojený v ethernetovej sieti.

3.3.5 Diagram nasadenia systému

PC klient sa bude môcť pripájať z ethernetovej siete pomocou lokálnej IP adresy RPi zariadenia. Pripojenie PC klienta z internetu bude umožňovať virtuálny server, ktorý bude poskytovať internetovú adresu RPi zariadenia. AndroidClient sa bude môcť pripájať iba z ethernetovej siete.

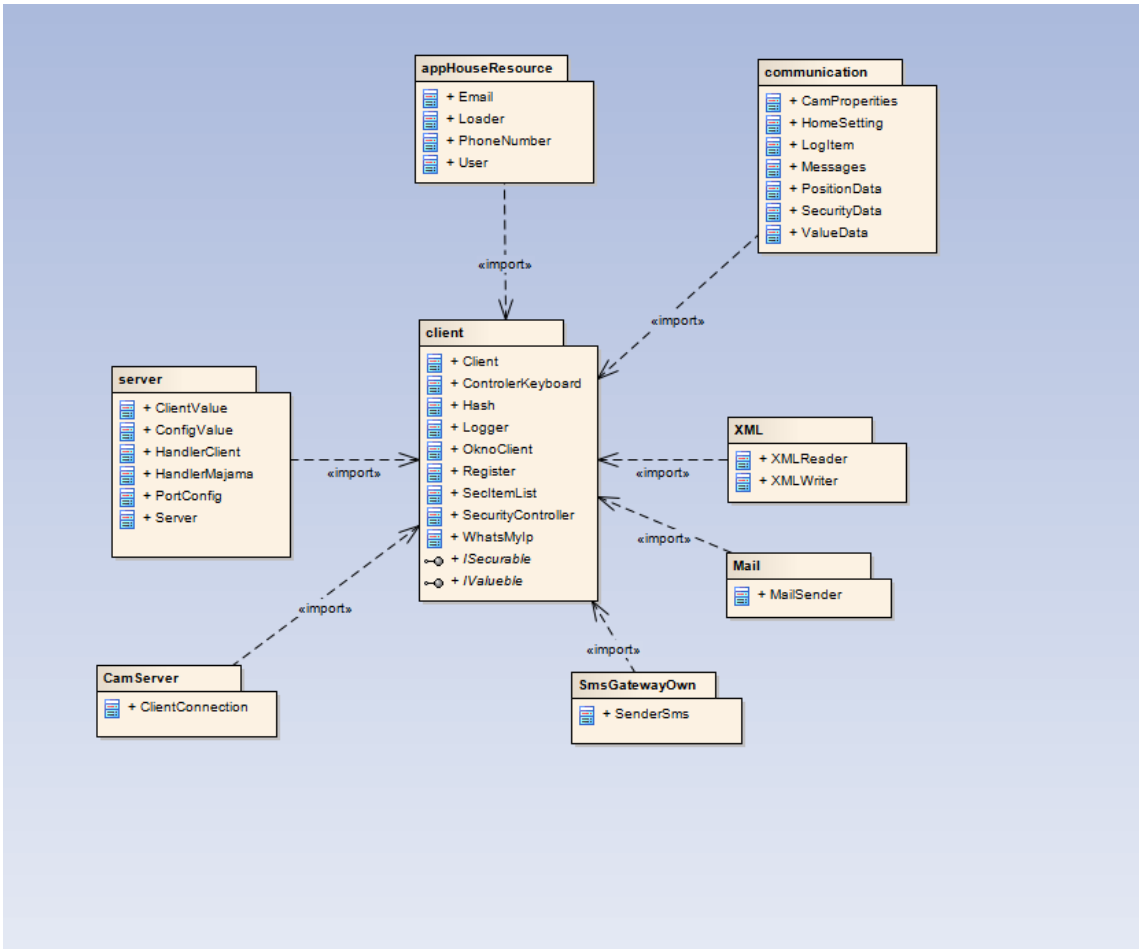


Obrázok 26 Diagram nasadenia systému

3.4 Implementácia návrhu riešenia

3.4.1 Implementácia Raspberry Pi Controllera

Systém bol naimplementovaný v programovacom jazyku JAVA, ktorý používa aj C knižnicu. Na kompilovanie knižnice sme použili GCC kompiler Systém používa na ukladanie dát xml súbory. Systém sa skladá z balíkov, ktoré sme sa snažili rozdeliť podľa OOP princípov (obr. 27).



Obrázok 27 Class diagram inteligentného domu

Popis jednotlivých balíkov

Balík Client

Ide o hlavný balík celého systému, ktorý obsahuje hlavný Controller celého systému, ktorý riadi jednotlivé funkcie.

Balík Server

Balík obsahuje komunikačné rozhranie pre PC a Android Clienta. Taktiež obsahuje rozhranie pre komunikáciu s Check Virtual Serverom – ukážku kódu pozri príloha 10.

Balík CamServer

Balík sprístupňuje web kameru.

Balík SMSGatewayOwn

Balík slúži na odosielanie SMS správ.

Balík Mail

Balík slúži na odosielanie e-mailov.

Balík XML

Balík slúži na ukladanie a načítanie xml súborov. Na to používa knižnicu XStreamJava v (http://xstream.codehaus.org/), ktorá mapuje JAVA objekty do XML jazyka.

Balík Communication

Balík obsahuje serializovateľné triedy, ktoré sú posielané alebo prijímané zo sieťovej komunikácie.

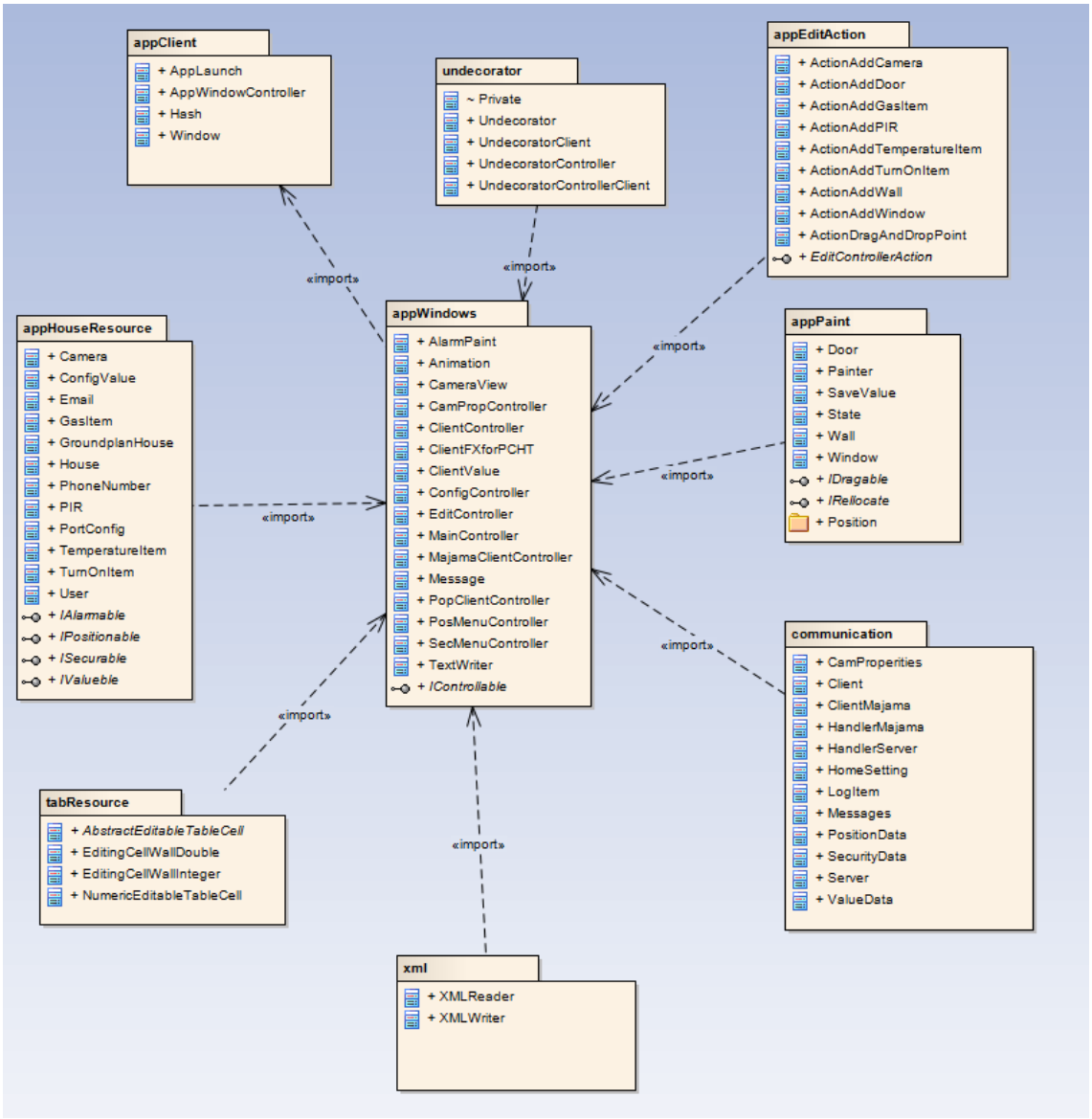
Balík appHouseResource

Balík obsahuje informácie o majiteľovi objektu.

Systém sa spúšťa z konzoly príkazom: *sudoJava –jar Client.jar.* Projekt ešte obsahuje kľúče ke.rs a pistore, ktoré slúžia na SSL komunikáciu. Projekt ďalej obsahuje xml súbory, ktoré obsahujú nastavenia systému. Všetky tieto súbory sa automaticky načítavajú pri spustení systému.

3.4.2 Implementácia PC klienta

PC klienta sme implementovali v programovacom jazyku JAVA. Použili sme rozšírenie JAVA FX, ktorá je určená na vývoj Rich Internet Aplication (bohaté internetové aplikácie). Vývoj grafického prostredia nám uľahčil nástroj SceneBuilder 2. Klient bol vyvíjaný primárne ako desktopová aplikácia. V budúcnosti po úprave bezpečnostných certifikátov by mala byť táto aplikácia dostupná aj webový prehliadač. Klient sa spúšťa súborom *ClientFXforPCHT.jar*. PC klient má nasledovnú štruktúru (obr. 28):



Obrázok 28 Class diagram pre klienta na PC

Balík appClient

Balík slúži na spustenie aplikácie a obsluhuje beh celej aplikácie

Balík appWindows

Balík obsahuje jednotlivé okná a ich Controllery - ukážku kódu pozri prílohy4, 5, 7.

Balík appPaint

Balík slúži na grafické vykreslenie pôdorysu domu a vykreslenie zariadení (napr. steny domu, pohybové senzory, ap.) – ukážku kódu pozri prílohy 1, 13,14.

Balík appHouseResource

Balík obsahuje triedy, ktoré reprezentujú objekty domu

Communication balík

Komunikačný balík obsahuje sérializovateľné triedy, ktoré sa používajú v sieťovej komunikácii.

Balík XML

Balík XML sa používa na čítanie a uloženie dát do xml súboru – ukážku kódu pozri príloha 15.

Balík undecorator

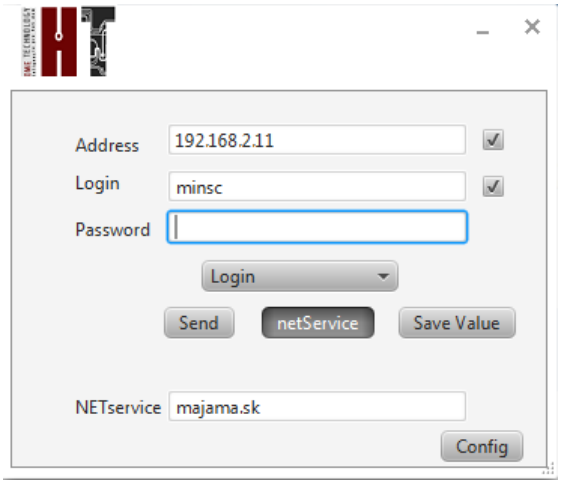
Balík undecorator obsahuje triedy, ktoré modifikujú vzhľad okna (arnaudnouard.wordpress.com/2013/02/02/undecorator-add-a-better-look-to-your-javafx-stages-part-i/).

Balík appEditAction

Balík obsahuje triedy, ktoré reprezentujú jednotlivé akcie v editovacom režime.

PC klient obsahuje viacero okien (obrazoviek), ktoré ďalej bližšie popisujeme.

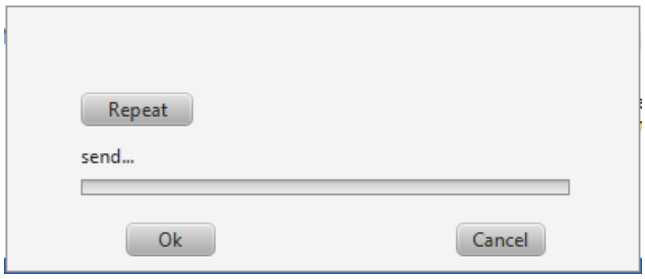
1.) **Prihlasovacie okno** (obr. 29)



Obrázok 29 Prihlasovacie okno PC klienta

Majiteľ v tomto okne vyplní meno, heslo a adresu zariadenia. Tieto hodnoty si môže uložiť, aby ich nemusel vypisovať opakovane. Toto okno slúži aj na registráciu užívateľa. Cez toto okno sa používateľ dokáže prihlásiť aj na Check Server, ktorý mu vráti IP adresu zariadenia. Používateľ sa cez toto okno dostane aj do konfiguračného okna.

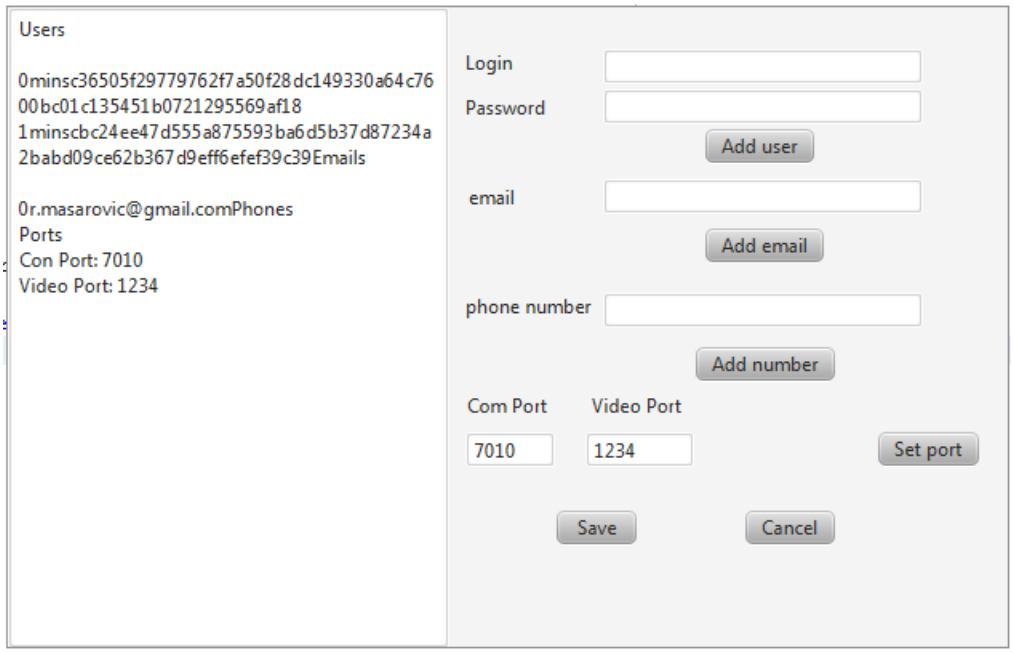
2.) **Komunikačné okno** (obr. 30)



Obrázok 30 Komunikačné okno PC klienta

Okno komunikácie zobrazuje majiteľovi priebeh prihlasovania na RPi.

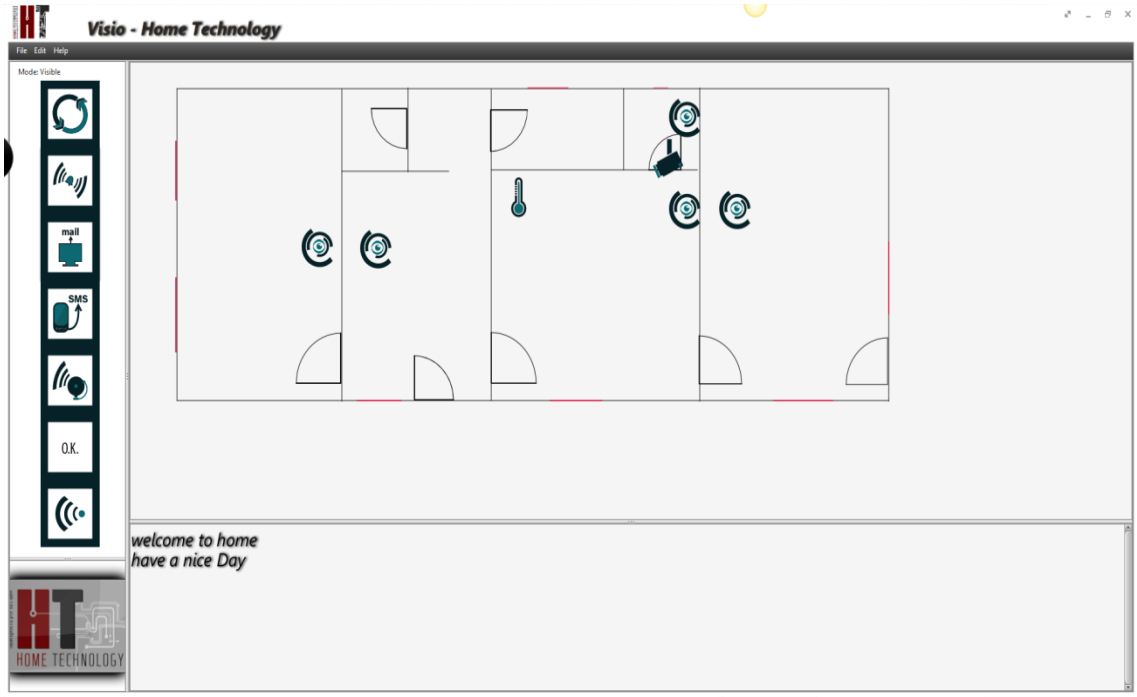
3.) **konfiguračné okno** (obr. 31)



Obrázok 31 Konfiguračné okno PC klienta

Konfiguračné okno umožňuje pridať aj odstrániť užívateľa, zadať e-mail a telefónne číslo, nastaviť komunikačné porty.

4.) **prehliadacie okno**(obr. 32)



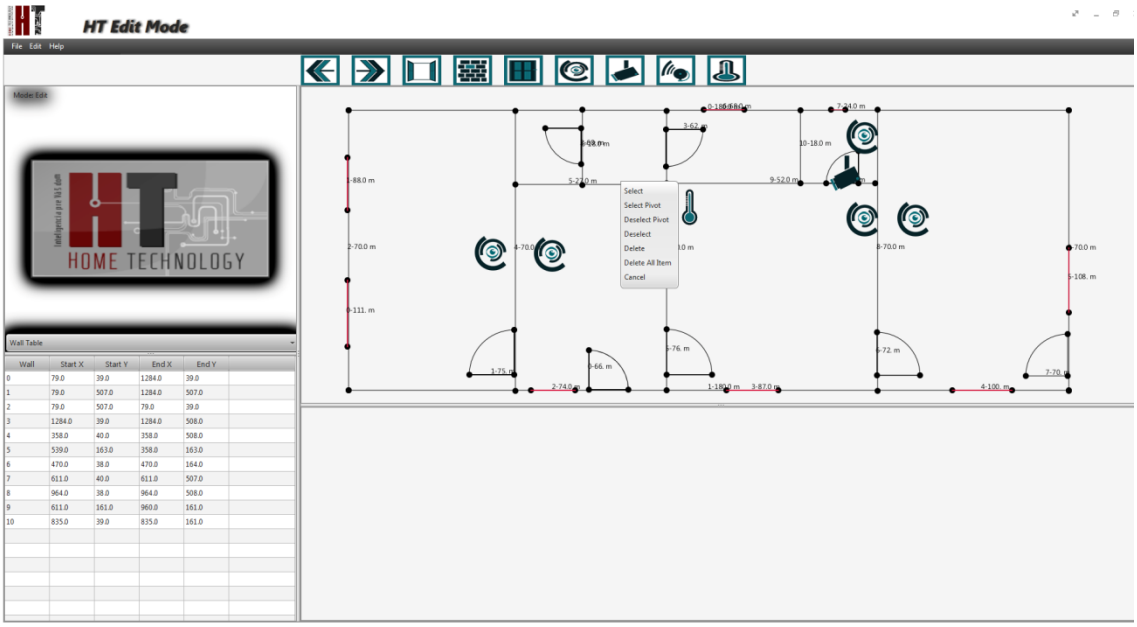
Obrázok 32 Prehliadacie okno

Prehliadacie okno je hlavné okno PC klienta, kde v ľavej časti obrazovky sú tlačidlá obsluhy. Tlačidlá sa ovládajú ľavým klikom myši. Vypínajú a zapínajú funkčnosť znázornenú na obrázku.

V strede sa nachádza pôdorys domu a zariadenia. Zariadenia reagujú na pravý klik myši, kde sa ukáže menu príslušného zariadenia. Po kliknutí na kameru ľavým tlačidlom myši, sa spustí stream videa. Musí byť nainštalovaný program VLCJ. Na používanie tohto programu cez JAVU sme použili knižnicu vlcj (http://www.java2s.com/Code/Jar/v/vlcj.htm).

V dolnej časti sa nachádza obrazovka na výpis správ zo systému. Cez menu sa používateľ dostane do editovacieho okna. Alarm spustí zvukové aj animačné upozornenie na pôdoryse domu.

5.) **editovacie okno** (obr. 33)

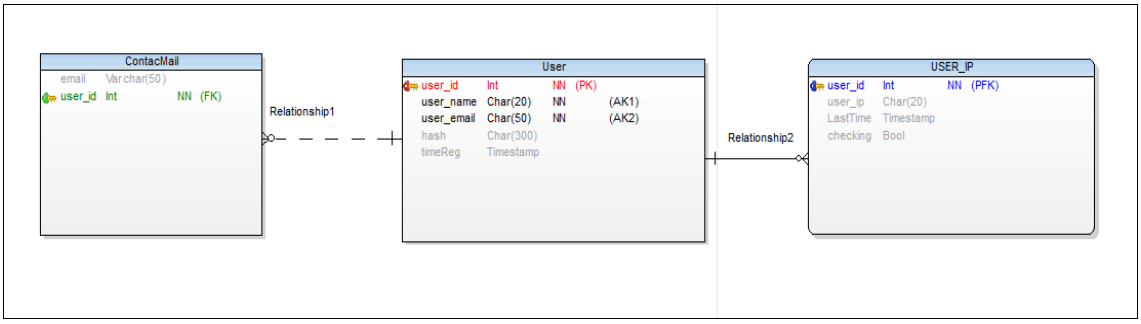


Obrázok 33Editovacie okno

Editovacie okno umožňuje majiteľovi nakresliť v hlavnom okne pôdorys domu, pridávať a odoberať zariadenia a meniť ich konfiguračné hodnoty. V hornej obrazovke sa nachádzajú ikonky, ktoré aktivujú činnosť, ktorú znázorňujú. Ľavým klikom na zariadenie sa inicializuje menu daného zariadenia. V ľavej časti sa nachádza tabuľka, v ktorej sú zobrazené hodnoty pôdorysu. Tieto hodnoty sa tu dajú meniť, čím pomáhajú nakresliť pôdorys domu. Po úprave majiteľ musí uložiť zmeny.

3.4.3 Implementácia Check Servera

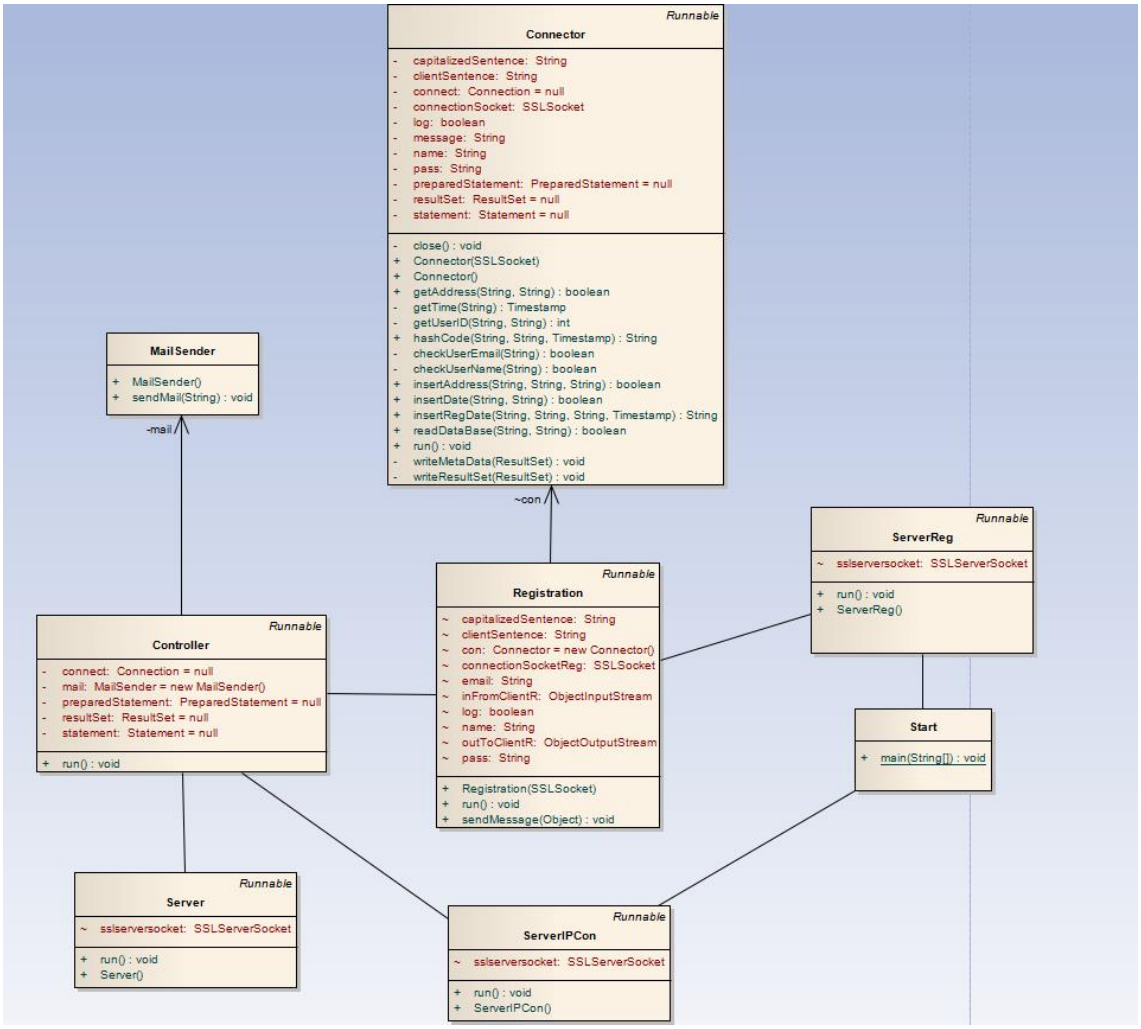
Check Server beží na virtuálnom serveri na doméne majama.sk. Na serveri je nainštalovaná JAVA 7 a databáza MYSQL. Na ukladanie dát preto používame túto databázu. Navrhli sme nasledovnú štruktúru (obr. 34), ktorá obsahuje 3 tabuľky:



Obrázok 34 Štruktúra Check Servera

JAVA Server ukladá IP adresu prihláseného RPi, ktorý mu v pravidelných časových intervaloch posiela správu. Ak RPi nepošle správu v určitý stanovený časový interval, Server pošle e-mail majiteľovi o chybe v komunikácii.

Server má nasledovnú štruktúru (obr. 35):



Obrázok 35 Class diagram štruktúry Check Servera

Trieda Registration slúži na registráciu užívateľa.

Trieda Controller kontroluje posledný čas komunikácie s RPi.

Trieda Conector slúži na obsluhu databázy – ukážku kódu pozri príloha 6.

Triedy ServerIPCon, ServerReg a Server slúžia na sieťovú komunikáciu.

3.4.4 Implementácia Android Clienta

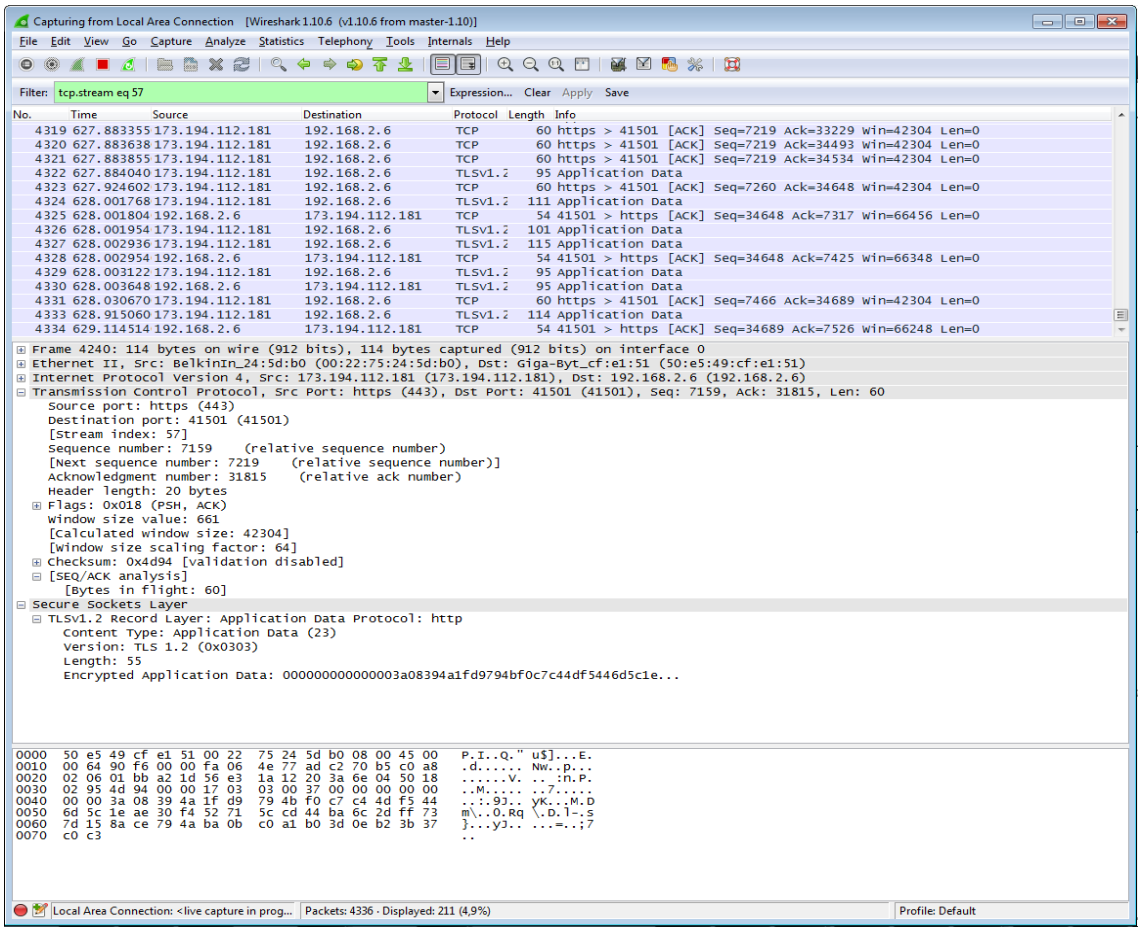
Android Client po pripojení na RPi automaticky vypne alarm. Keď sa pripojenie preruší, alarm sa zapne. Klient obsahuje jednu triedu FullscreenActivity, cez ktoré sa prihlási na zariadenie (obr. 36) - ukážku kódu pozri príloha 2.

Obrázok 36 Prihlasovacie menu Android Clienta

3.5Overenie funkčnosti riešenia

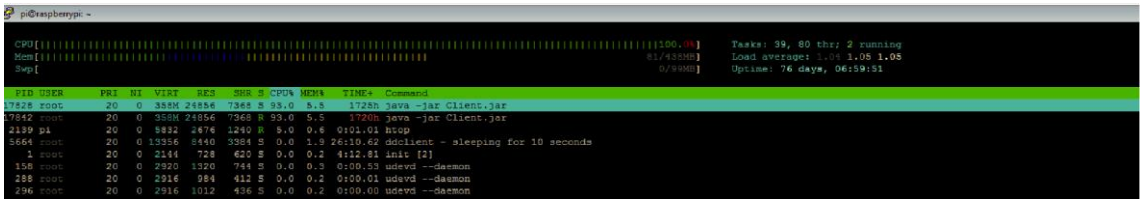
Pri overovaní funkčnosti riešenia sme postupovali podľa scenárov, ktoré boli vypracované na začiatku vývoja softvéru. Celý systém nevykazoval chybovosť a na dané postupy reagoval správne.

Bezpečnosť systému sme sledovali pomocou programu WireShark, kde sme sledovali, či je komunikácia zašifrovaná (obr. 37). Toto kritérium bolo splnené. Bezpečnosť však môže byť narušená v prípade zverejnenia prístupového hesla do systému neoprávneným osobám.



Obrázok 37 Bezpečnosť komunikácie sledovaná programom WireShark

Stabilitu systému sme sledovali počas 70 dní v mesiacoch 12/2013 až 02/2014, kedy nedošlo k pádu aplikácie, systém fungoval bez prerušenia. Na tento účel sme využili monitorovacie prostredie HTop stiahnuté z depozitáru Debian https: //packages.debian.org/search?keywords=htop, ktoré slúži na monitorovanie procesov a systému (obr. 38).



Obrázok 38 HTop monitorovacie prostredie v Raspbiane

Pre ďalšie dlhodobejšie testovanie a ladenie systému zapojíme do procesu aj majiteľov domu.

ZÁVER

S rýchlym vývojom v oblasti informačných technológií je už aj nás viditeľný trend nástupu inteligentných domov, ktoré sa postupne stávajú dostupnejšie aj pre širšiu škálu potenciálnych záujemcov. V priebehu posledných niekoľko rokov vzniklo mnoho riešení, ktoré ponúkajú komplexné systémy na zabezpečenie funkcionalít inteligentného domu. Pri rozhodovaní užívateľov o miere inteligencie v dome sa užívatelia orientujú najmä na tri oblasti, a to: komfort, bezpečnosť a úspory. Jednou z kľúčových požiadaviek je tiež aj cenová náročnosť systému na zabezpečenie funkcionalít inteligentného domu. Jednou z nízko nákladových možností realizácie funkcií inteligentného domu je ich implementácia s pomocou mikro PC Raspberry Pi, ktorého najväčšou výhodou je nízka nákupná cena a vysoká variabilita možných riešení pri návrhoch na tvorbu používateľského prostredia na správu a riadenie inteligentného domu. Tieto výhody takéhoto riešenia ma zaujali, preto som si vybral vývoj softvérového riešenia na implementáciu funkcionalít inteligentného domu s využitím Raspberry Pi ako tému mojej diplomovej práce.

Cieľom predkladanej diplomovej práce bolo navrhnúť a implementovať programový systém pre Raspberry Pi, ktorý zabezpečí funkcie inteligentného domu, ako aj navrhnúť a vytvoriť používateľské prostredie pre návrh a správu inteligentného domu.

Systém pre správu a riadenie inteligentného domu sme vyvíjali pre majiteľov víkendového domu, ktorý nie je trvalo obývaný. Našou úlohou bolo pre nich navrhnúť a implementovať taký systém, ktorý bude cenovo dostupný a ktorý zároveň disponuje možnosťami ako uspokojiť ich požiadavky na úsporu nákladov, bezpečnosť a požadovanú mieru komfortu.

Náš projekt návrhu a implementácie softvérového riešenia pre Raspberry Pi, ktorý zabezpečí vybrané funkcie inteligentného domu sme predstavili v tretej kapitole práce. Vytvorený programový systém bol vyvíjaný pomocou modifikovanej agilnej metodiky Scrum; ako vývojové prostredie bolo zvolené NetBeans, ako ďalšie vývojové nástroje boli použité PuTTY, WinSCP, Eclipse. Softvér pre RaspberryPi bol primárne naprogramovaný v programovacom jazyku JAVA.

Navrhnuté riešenie vybraných funkcionalít inteligentného domu bolo v praxi zrealizované a funkčnosť riešenia bola úspešne overená. Pri overovaní funkčnosti riešenia sme testovali predovšetkým bezpečnosť a stabilitu systému. Systém sme sledovali v priebehu 70 dní, počas ktorých nedošlo k pádu aplikácie, systém nevykazoval chybovosť a fungoval bez prerušenia. Pre ďalšie dlhodobejšie testovanie a ladenie systému zapojíme do tohto procesu aj majiteľov domu. Kompletné programové vybavenie na zabezpečenie niektorých vybraných funkcií inteligentného domu pomocou Raspberry Pi na CD nosiči tvorí prílohu našej diplomovej práce.

Na základe tohto zhrnutia chceme na záver vysloviť presvedčenie, že sme stanovený cieľ práce naplnili a naše programové vybavenie na zabezpečenie vybraných funkcií inteligentného domu bude slúžiť k spokojnosti jeho majiteľov, pre ktorých sme toto riešenie navrhli a v praxi aj implementovali.

LITERATÚRA

BARTÁK, J. 2014. *Raspberry pi alebo počítač neobmedzených možností.* [online]. 2014. [cit.2014-03-22] Dostupné na: <http://www.the-programing.6f.sk/2014/03/02/raspberry-pi-alebo-pocitac-neobmedzenych-moznosti/>

HALFACREE, G., UPTON, E. 2013. *Raspberry Pi. Užívateľská příručka*. Brno : Computer Press. 2013. 232 s. ISBN 978-80-251-4116-8.

HAMERNÍK, P., TANUŠKA, P., MUDRONČÍK, D. 2012. Classification of Functions in Smart Home. In: *International Journal of Information and Education Technology*. ISSN 2010-3689. Vol. 2, No. 2, 2012, s. 149–15.

HAMERNÍK P., TANUŠKA, P. 2012. *Kategorizovanie funkcií inteligentnéhodomu*. [online]. 2012.[cit.2013-11-04] Dostupné na:<http://www.idbjournal.sk/rubriky/prehladove-clanky/kategorizovanie-funkcii-inteligentnehodomu.html?page_id=14446>

HARPER, R. 2003. *Insidethe Smart Home*. Springer-Verlag. London, 2003. ISBN 1852336889.

HONEK, M. *Test RaspberryPpijako HTPC: raději ne*. [online]. 2012.[cit.2013-11-17] Dostupné na: <http://www.digilidi.cz/test-raspberry-pi-jako-htpc-radeji-ne>

HORÁK, M. 2012. *Raspberry Pi: Instalujeme systém (Raspbian)*. [online]. 2012.[cit.2013-11-26] Dostupné na: <http://www.abclinuxu.cz/clanky/raspberry-pi-instalujeme-system-raspbian>

JANEČEK,V. 2012. *Vyzkoušelij sme mikropočítač Raspberry Pi*. [online]. 2012. [cit.2013-12-11] Dostupné na: <http://www.zive.cz/clanky/vyzkouseli-jsme-mikropocitac-raspberry-pi/sc-3-a-165391/>

KABELE, K. 2007. *Vnitřní prostředí inteligentních budov.* [online]. 2007. [cit.2013-11-09] Dostupné na:<http://www.asb-portal.cz/tzb/osvetleni-a-elektroinstalace/vnitrni-prostredi-inteligentnich-budov

KLABAN, J. 2012. *Inteligentní dům a jeho možnosti*.[online]. 2012.[cit.2013-11-07] Dostupné na: <http://www.asb-portal.cz/tzb/osvetleni-a-elektroinstalace/inteligentni-dum-a-jeho-moznosti>

LELOVSKÝ, M. 2013. *Viete aký je to inteligentný dom?* [online]. 2013. [cit.2013-11-10] Dostupné na: <http://www.asb.sk/tzb/osvetlenie-a-elektroinstalacie/viete-aky-je-to-inteligentny-dom>

MONK, S. 2013. *Raspberry Pi. CookBook. Software and Hardware. Problems and Solutions*. Verlag : O´REILLY MEDIA, 2013. 414 p. ISBN 978-1-44936-522-6.

PILIPOVÁ, I. 2011.*Úvod to teórie inteligentných budov a vzniku diskomfortu v inteligentných obytných budovách.* [online]. 2011. [cit.2013-11-10] Dostupné na: <http://www.tzbportal.sk/stavebnictvo/uvod-teorie-inteligentnych-budov-vzniku-diskomfortu-v-inteligentnych-obytnych-budovach>

PUŠKÁR, B. 2007. *Inteligentné budovy na bývanie: Komfort či obmedzovanie?*.[online]. 2007.[cit.2013-11-07]
Dostupné na: <http://www.asb.sk/architektura/stavby/inteligentny-dom/inteligentne-budovy-na-byvanie-komfort-ci-obmedzovanie>

STANEK, J. 2008. *Inteligentné domy sú žiadané čoraz viac*. In Bložoň, 2008. [online]. 2013.[cit.2013-12-04]
Dostupné na: <http://www.atpjournal.sk/rubriky/rozhovory/inteligentne-domy-su-ziadane-coraz-viac.html?page_id=7045&month=1>

SKYVA, M. 2013. *Príležitosť menom inteligentné domy.*[online]. 2013.[cit.2013-11-09]
Dostupné na: <http://www.asb.sk/architektura/stavby/inteligentny-dom/prilezitost-menom-inteligentne-domy>

SMART HOME ENERGY [online] 2013.[cit.2013-11-18] Dostupné na: <http://smarthomeenergy.co.uk/what-smart-home>

STRIGÁČOVÁ, L., 2012. *Optimalizace návrhu inteligentních budov podle potřeb budoucích uživatelů.* [online]. 2012.[cit.2013-11-15] Dostupné na:
<http://elektro.tzb-info.cz/inteligentni-budovy/8935-optimalizace-navrhu-inteligentnich-budov-podle-potreb-budoucich-uzivatelu>

ŠTEVO, S. 2012. Čo tvorí inteligenciu budov? In *Eurostav ,* 18.roč., 2012, č 3, 10-14, ISSN 1335-1249.

ŠTEVO, S. 2013. *Inteligentná budova - čo to v skutočnosti znamená?*[online]. 2013.[cit.2014-01-04] Dostupné na:
<http://www.asb.sk/architektura/stavby/inteligentny-dom/inteligentna-budova-co-to-v-skutocnosti-znamena>

VALEŠ, M. 2008. *Inteligentní dům*. Brno : Vydavatelství ERA. 2008. 136 s. ISBN 978-80-7366-137-3.

VACHÁLEK, J. 2014. *Využitie Raspberry PI pri návrhu zabezpečenia inteligentnej domácnosti (3)* In iDBJournal. ISSN 1338-333, 2014, roč. IV, č. 1, s. 20-21. [online] 2014. [cit.2014-01-17] Dostupné na:
<http://www.idbjournal.sk/buxus/docs/casopisy_cele/iDB_1_2014_web.pdf>

WANG, S. 2009. *Intelligent Buildings and Building Automation*. Hardcover, Published by Spon Press, 2009.264 p. ISBN-13978-0-415-47570-9.

http://www.arisys.sk/sk/Produkty-a-riesenia.alej

http://www.cybrotech.co.uk

http://docs.oracle.com/javase/6/docs/technotes/guides/security/jsse/JSSERefGuide.html#Features
http://dyn.com/dns/
http://elinux.org/RPi_Low-level_peripherals
http://www.inels.sk/
http://www.java2s.com/Code/Jar/v/vlcj.htm
https://packages.debian.org/search?keywords=htop
http://www.raspi.cz/2011/12/co-je-raspberry-pi/raspberry-pi-ab/
http://www.raspberrypi.org/downloads
http://www.rpiblog.com/search/label/GPIO
http://smslib.org/

PRÍLOHY

Príloha 1 Kód vkladania pohybového senzora v klientskej PC aplikácii

```
public void addPIR(double x, double y, Canvas canvas) {
    PIR c = new PIR(appController.getCounter());
    appController.plusCounter();
    c.getPositionData().setX(x);
    c.getPositionData().setY(y);
    c.getPositionData().setRotate(0);
    c.getPositionData().setName("PIR");
    Image image = new Image(appController.getPathPIRImg());
    c.setImg(image);
    ImageView iv = new ImageView(c.getImg());
    iv.setRotate(c.getPositionData().getRotate());
    iv.setX(c.getPositionData().getX());
    iv.setY(c.getPositionData().getY());
    iv.setFitHeight(image.getHeight());
    iv.setFitWidth(image.getWidth());

    c.setIv(iv);
    iv.autosize();

    c.getIv().setOnMouseClicked(new EventHandler<MouseEvent>() {
        @Override
        public void handle(MouseEvent t) {

            appController.checkItemMenu(t.getX(), t.getY());
        }
    });
    itemGroup.getChildren().add(c.getIv());
    pirs.add(c);
}
```

Príloha 2 Kód pripojenia Android Clienta

```
TrustManager[] tmgr = null;

   System.out.println("Using trust file "
           + " with pwd: ");
   TrustManagerFactory tmf;
   try {
       tmf = TrustManagerFactory.getInstance("X509");
       KeyStore ts  = KeyStore.getInstance("BKS");
       ContextWrapper context;
       AssetManager am = getAssets();
       InputStream instream = am.open("rp");

       try {
           ts.load(instream, [redacted].toCharArray());
       } finally {
           try {
              instream.close();
           } catch(Exception ignore) {
           }
       }

           tmf.init(ts);
           tmgr = tmf.getTrustManagers();

           SSLContext ctx = SSLContext.getInstance("TLS");
           ctx.init(null, tmgr, null);
           sslsocketfactory = (SSLSocketFactory) ctx
                   .getSocketFactory();
           sslsocket = (SSLSocket) sslsocketfactory.createSocket(address.getText()+"", 7010);

           outputStream = new ObjectOutputStream(sslsocket.getOutputStream());
           inputStream = new ObjectInputStream(sslsocket.getInputStream());
           outputStream.writeObject(new LogItem(login.getText()+"", pass.getText()+""));
           outputStream.writeObject("Android");
```

Príloha 3 Kód hashovacej funkcie

```
public Hash() {
}
public String hashCode(String s, String st, String tm) throws NoSuchAlgorithmException {
   MessageDigest messageDigest = MessageDigest.getInstance("SHA-256");
   String t = s + " " + st + " " + tm;
   messageDigest.update(t.getBytes());
   byte[] mdbytes = messageDigest.digest();

   //convert the byte to hex format method 1
   StringBuffer sb = new StringBuffer();
   for (int i = 0; i < mdbytes.length; i++) {
       sb.append(Integer.toString((mdbytes[i] & 0xff) + 0x100, 16).substring(1));
   }
   String encryptedString = sb.toString();
   return encryptedString;
}
```

Príloha 4 Kód inicializácie okna

```
@Override
public void initialize(URL url, ResourceBundle rb) {
    try {
        File f = new File("clientValue.xml");
        if (f.exists()) {
            XMLReader xmlReader = new XMLReader();
            xmlReader.startParseClient("clientValue.xml");
            ClientValue clientV = xmlReader.getClientValue();
            addCheckBox.setSelected(clientV.isSaveAddress());
            logCheckBox.setSelected(clientV.isSaveLogin());
            addressInput.setText(clientV.getAddress());
            loginInput.setText(clientV.getLogin());
            btNetService.setSelected(clientV.isNetServiceOn());
            netServiceInput.setText(clientV.getNetService());
        }

    } catch (ParserConfigurationException ex) {
        Logger.getLogger(ClientController.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SAXException ex) {
        Logger.getLogger(ClientController.class.getName()).log(Level.SEVERE, null, ex);
    } catch (IOException ex) {
        Logger.getLogger(ClientController.class.getName()).log(Level.SEVERE, null, ex);
    }

    choiceBox.setItems(FXCollections.observableArrayList(
            "Login",
            new Separator(), "Register")
            );
    choiceBox.getSelectionModel().selectFirst();
}
```

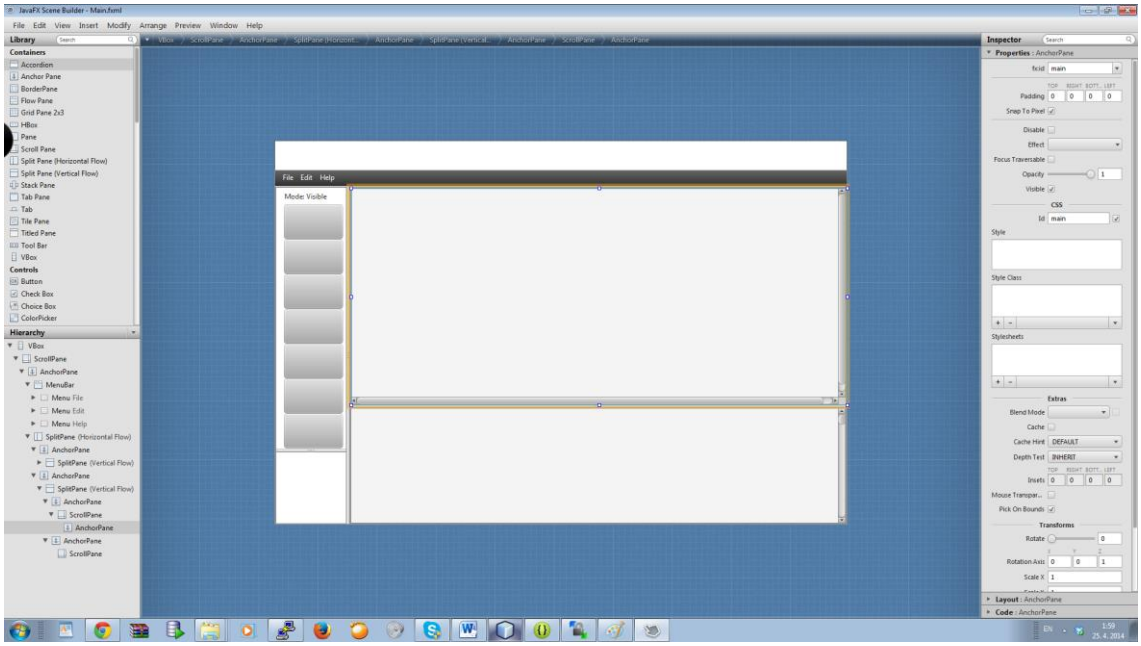
Príloha 5 Kód potrebná inicializácia premenných pred inicializáciou okna

```
@FXML
private ChoiceBox choiceBox;

@FXML
private TextField addressInput;
@FXML
private TextField loginInput;
@FXML
private TextField passwordInput;
@FXML
private TextField netServiceInput;
@FXML
private CheckBox addCheckBox;
@FXML
private CheckBox logCheckBox;
@FXML
private ToggleButton btNetService;


@FXML
private void handleSaveButtonAction(ActionEvent event) throws IOException, ParserConfigurationException, TransformerException {
    ClientValue v = new ClientValue();
    if (addCheckBox.isSelected()) {
        v.setSaveAddress(true);
        v.setAddress(addressInput.getText());
    }
    if (logCheckBox.isSelected()) {
        v.setSaveLogin(true);
        v.setLogin(loginInput.getText());
    }
    v.setNetServiceOn(btNetService.isSelected());
    v.setNetService(netServiceInput.getText());
    XMLWriter xmlWriter = new XMLWriter();
    xmlWriter.writeClientValue(v);
}
```

Príloha 6 Kód registrácie používateľa na Check Server

```
public String insertRegDate(String name, String email, String p, Timestamp t) throws SQLException, NoSuchAlgorithmException {
    connect = DriverManager.getConnection("jdbc:mysql://localhost/rpiService?" + "user=[redacted]");
    statement = connect.createStatement();
    if (checkUserName(name)) {
        return "Name exists choose other";
    }
    if (checkUserEmail(email)) {
        return "Email exists choose other";
    }
    p = this.hashCode(name, p, t);
    System.out.println(p);
    statement.executeUpdate("INSERT INTO rpiService.User (user_name,user_email,hash,timeReg) VALUES ('" + name + "','" + email + "','" + p + "','" + t + "')");
    statement = connect.createStatement();
    int i = this.getUserID(p, name);
    statement.executeUpdate("INSERT INTO rpiService.User_IP (user_id) VALUES ('" + i + "')");
    connect.close();
    return "Registration done";
}
```

Príloha 7 Ukážka vývojového prostredia na tvorbu okien v javaFX



Príloha 8 Kód kontroly pohybových senzorov

```
@Override
public void run() {
    while (runApp) {
        while (runCheckAlarm) {
            for (int i = 0; i < secList.size(); i++) {
                SecurityData item = secList.get(i);
                if (!this.checkItem(item)) {
                    System.out.println(item.toString() + " alarm");
                    alarm = true;
                    if (!alarmOff){
                        alarmOff=true;
                        try {
                            if (connected) {
                                handlerClient.sendMessageAlarm(item.getId(), item.isState());
                            }
                            if (sendMail) {
                                mailSender.sendMail(item);
                            }
                            if (sendSms) {
                                sender.sendMessage(item.getName());
                            }
                        } catch (IOException ex) {
                            Logger.getLogger(SecurityController.class.getName()).log(Level.SEVERE, null, ex);
                        } catch (ClassNotFoundException ex) {
                            Logger.getLogger(SecurityController.class.getName()).log(Level.SEVERE, null, ex);
                        } catch (Exception ex) {
                            Logger.getLogger(SecurityController.class.getName()).log(Level.SEVERE, null, ex);
                        }
                        try {
                            Thread.sleep(1000);
                        } catch (InterruptedException ex) {
                            java.util.logging.Logger.getLogger(Client.class.getName()).log(Level.SEVERE, null, ex);
```

Príloha 9 Kód poslania SMS

```
public void sendMessage(String name) throws Exception {
    OutboundNotification outboundNotification = new OutboundNotification();
    SerialModemGateway gateway = new SerialModemGateway("modem.ttyUSB0", "/dev/ttyUSB0", 57600, "Huawei", "E173");
    gateway.setInbound(false);
    gateway.setOutbound(true);
    Service.getInstance().setOutboundMessageNotification(outboundNotification);
    Service.getInstance().addGateway(gateway);
    Service.getInstance().startService();
    gateway.setSimPin("0000");
    gateway.setSmscNumber("+421905303303");
    OutboundMessage msg = new OutboundMessage("+421905608243",
            "Alarm pohyb nastal v:" + name);
    gateway.sendMessage(msg);
    System.out.println(msg);
    Service.getInstance().stopService();
}
```

Príloha 10 Kód Servera RPi

```
KeyManager[] kmgr = null;
KeyManagerFactory kmf = KeyManagerFactory
        .getInstance("SunX509");
KeyStore ks = KeyStore.getInstance("JKS");
File f = new File("ke.rs");
ks.load(new FileInputStream(f), null);
kmf.init(ks, [redacted].toCharArray());
kmgr = kmf.getKeyManagers();

SSLContext ctx = SSLContext.getInstance("TLSV1.2");
ctx.init(kmgr, null, null);
SSLServerSocketFactory sslserversocketfactory = (SSLServerSocketFactory) ctx
        .getServerSocketFactory();
sslserversocket = (SSLServerSocket) sslserversocketfactory.createServerSocket(c.getComPort());
```

Príloha 11 Kód načítania teplotného senzora

```
class Temp{
public native int readDHT(int c,int g);
static{
System.load("/home/pi/libTemp.so");
}

public int getTemperature(){

int  a= new Temp().readDHT(2302,25);

System.out.println(a);
return a;
}
}
```

Príloha 12 Príkaz na vytvorenie C knižnice

```
pi@raspberrypi ~ $ sudo gcc -shared -o libTemp.so  Adafruit_DHT.c -l bcm2835 -st
d=gnu99 -fPIC -I /usr/lib/jvm/java-1.7.0-openjdk-armhf/include/ -I /usr/lib/jvm/
java-1.7.0-openjdk-armhf/include/linux/
```

Príloha 13 Kód vykreslenia pôdorysu domu v hlavnom okne klientskej aplikácie

```
public void paintScene() {
    GraphicsContext gc = canvas.getGraphicsContext2D();
    gc.clearRect(0, 0, 12000, 12000);
    gc.beginPath();
    gc.setStroke(Color.BLACK);

    for (Wall l : listWall) {
        gc.beginPath();
        gc.moveTo(l.getStartX(), l.getStartY());
        if (l.isSelectStar()) {
            gc.setFill(Color.RED);
        } else {

            gc.setFill(Color.BLACK);
        }
        if (l.isPivotPointS()) {
            gc.setFill(Color.BLUE);
        }
        gc.fillOval(l.getStartX() - 5, l.getStartY() - 5, 10, 10);
        gc.lineTo(l.getEndX(), l.getEndY());

        if (l.isFinished() && !l.isSelectEnd()) {
            gc.setFill(Color.BLACK);
            gc.fillOval(l.getEndX() - 5, l.getEndY() - 5, 10, 10);
        }
        if (l.isFinished() && l.isSelectEnd()) {
            gc.setFill(Color.RED);
            gc.fillOval(l.getEndX() - 5, l.getEndY() - 5, 10, 10);
        }
        if (l.isFinished() && l.isPivotPointE()) {
            gc.setFill(Color.BLUE);
            gc.fillOval(l.getEndX() - 5, l.getEndY() - 5, 10, 10);
        }
```

Príloha 14 Kód vkladania okna v klientskej PC aplikácii

```
public class ClientFXforPCHT extends Application {

    @Override
    public void start(Stage stage) throws Exception {
        FXMLLoader fxmlLoader = new FXMLLoader();
        Parent root = (Parent) fxmlLoader.load(getClass().getResource("Client.fxml").openStream());

        Undecorator undecorator = new Undecorator(stage, (Region) root);
        undecorator.getStylesheets().add("undecorator.css");
        Scene scene = new Scene(undecorator);
        stage.setTitle("HT");
        stage.setScene(scene);
        scene.setFill(Color.TRANSPARENT);
        stage.initStyle(StageStyle.TRANSPARENT);
        stage.setScene(scene);

        ClientController myController = (ClientController) fxmlLoader.getController();
        stage.show();
    }
```

Príloha 15 Kód inicializácie tried v xstream knižnici

```
public void writeConfigValue(ConfigValue conf) throws TransformerException {
    xstream.alias("configValue", ConfigValue.class);
    xstream.alias("user", User.class);
    xstream.alias("number", PhoneNumber.class);
    xstream.alias("email", Email.class);
    xstream.alias("portConf", PortConfig.class);
    rootElement = doc.createElement("config");
    doc.appendChild(rootElement);
    Element node = doc.createElement("ConfigValue");
    rootElement.appendChild(node);
    node.appendChild(doc.createTextNode(this.addNodeConfigValue(conf)));
    this.saveFile("configValue.xml");
}
```

Príloha 16 CD nosič s kompletným softvérom