PROIECT DE DIPLOMA
Sistem IoT pentru controlul accesului in clădire

Alexandru Cristian IONESCU

2021
Noiembrie

Cuprins

Capitolul 1

Introducere

1.1 Obiectivele lucrării de licența

1.1.1 Realizarea unui studiu de piata pentru determinarea fezabilitatii solutiei

In continuare voi realiza un scurt studiu de piata pe nisa sistemelor Internet of Things (IoT) destinate uzului casnic. Un caz particular de astfel de dispozitive sunt cele care indeplinesc functia de interfon sau ofera contrulul accesului intr-o incinta de la distanta.

In momentul de fata exista pe piata o multitudine de produse de tip incuietoare inteligenta sau sisteme tip interfon GSM, atat de la producatori cunoscuti cat si de la branduri nou infiintate. Aceasta lucrare va analiza trei tipuri de solutii existente, cu implementari diferite, incercand sa identifice functionalitati comune, avantaje si dezvantaje dintr-o plaja cat mai mare de dispozitive.

Solutiile prezentate mai sus au dezavantaljul ca nu au fost concepute sa fie integrate cu un sistem existent, intr-un bloc mai vechi. Prin urmare exista un segment de piata de utilizatori care ar dori sa benefecieze de functiile intefonului inteligent, dar nu pot deoarece asta ar presupune schimbarea sistemului din intreaga cladire.

1.1.2 Dezvoltarea unui sistem compatibil POTS pentru interfatarea in reteaua IoT

Pentru a putea oferi functiile inteligente unei audiente cat mai large, sistemul propus in aceasta lucrare se poate conecta la reteaua Plain Old Telephone Service (POTS) printr-o simpla mufa RJ11.

1.2 Descrierea domeniului din care face parte tema de licența

Aceasta lucrare face parte dintr-un domeniu mai vechi, dar care a prins amploare recent, domeniul automatizarilor casnice si IoT.

1.2.1 Istoric

Interesul in conectarea locuintelor pentru a obtine functionalitate aditionala dateaza inca din anii 60, majoritatea fiind concepte prototipate de entuziasti cu inclinatii spre electronica.

Jim Sutherland, inginer la Westinghouse a creat primul sistem de automatizare a domiciliului in anul 1964, ECHO IV. Acesta era capabil sa controleze temperatura, alte aparate casnice cat si sa permita retinerea de mementouri sau liste de cumparaturi. Cu introducerea retelei Advanced Research Projects Agency Network (ARPANet) in 1969, un precursor al Internetului, universul dispozitivelor casnice conectate a cunoscut o perioada rapida de dezvoltare in anii urmatori [7].

Trecerea de la o noutate scumpa la un sistem ce ofera functii cu adevarat practice a venit sub forma proiectului ”X10 Home Automation”. Acesata se putea integra cu sistemul de climatizare existent al cladirii, controla electrocasnice mici, cat si corpuri de iluminat.

In anul 1984, Asociatia Nationala a Constructorilor din Statele Unite a creat un grup de control numit ”Smart House” pentru a accelera includerea tehnologiei in proiectele viitoare [1].

Pentru consumatori, dezvoltarile din urmatorii ani au adus usi automate pentru garaje, termostate programabile si sisteme de securitate in cadrul monden, concomitent reducand preturile solutiilor oferite. In ciuda acestor semne, sociologii au concluzionat la vremea respectiva ca nu exista un interes real in conceptul ”Smart House”.

1.2.2 Curent

Solutiile de tip ”Smart Home” din prezent se integreaza in general cu o retea precum Espressif, Apple HomeKit sau Google Home. Aceasta permite controlul dispozitvelor conectate prin intermediul telefonului mobil.

Apple HomeKit/Google Home

Nest TC

1.3 Prezentare pe scurt a capitolelor

//To do

Capitolul 2

Descrierea problemei abordate

2.1 Formularea problemei

In orase precum Bucuresti, majoritatea blocurilor au fost contruite inainte de anul 1990 si prin urmare interfoanele lor se bazeaza pe POTS. Traind in era digitala, utilizatorul ideal (e corporatist) isi doreste augmentarea functionalitatilor sistemului existent, pentru a nu trebui sa isi convinga toti vecinii sa investeasca in modernizarea sistemului de acces. Pentru a putea adresa cat mai multi utilizatori, solutia acestei probleme trebuie sa fie agnostica de smartphoneul si interfonul existent a utilizatorului, dar sa ofere integrari cu alte solutii de tip ”Smart Home”.

2.2 Studiu asupra realizărilor similare din domeniu

2.2.1 Videx UK

Interfoanele GSM de la Videx sunt conectate la reteaua mobila de telefonie si permit operarea unei porti prin intermediul unui releu. Ele necesita doar o sursa de curent externa, o antena si o cartela Subscriber Identity Module (SIM) pentru a opera.

Figura 2.1: Sistem interfon Videx GSM [5]

Printre functionalitatile principale se numara:

- Poate include un cititor de carduri RFID si cheie

- Versiune rezistenta la vandalism
- Pana la 4 numere de telefon per apartament, pentru redundanta. In cazul in care primul numar nu se poate apela sau nu raspunde, se va incerca urmatorul numar programat
- Ofera aplicatie Android si iOS pentru programat unitatea

Dezavantaje:

- Nu ofera integrare cu servicii din reteaua IoT

2.2.2 Google Nest x Yale Lock

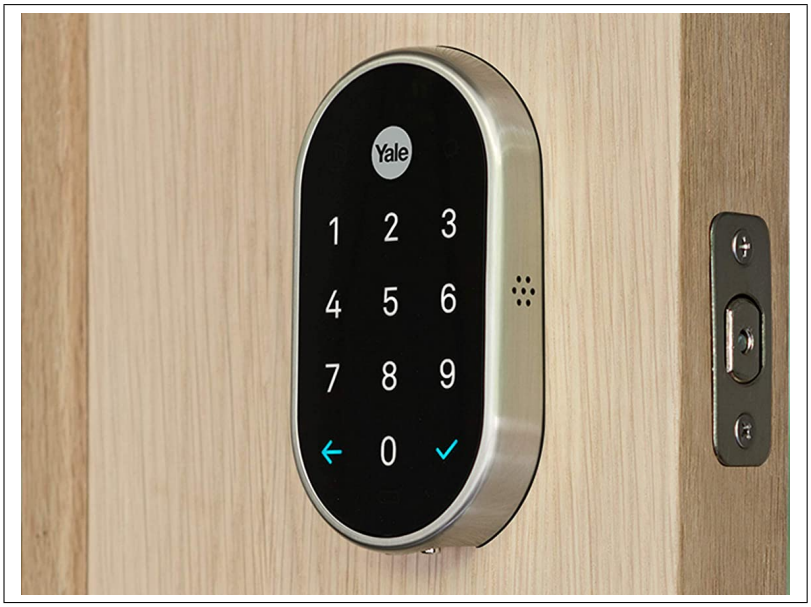

Figura 2.2: Next x Yale Lock [6]

Avantaje:

- Permite accesul prin intermediul unui PIN ales de utilizator
- Ofera alerte cand cineva inchide sau deschide usa
- Ofera integrare cu Google Home si Nest Home

Dezavantaje:

- Are nevoie de 4 baterii tip AA pentru a functiona
- Nu are acces cu cheie sau cartela
- Nu are versiune rezistenta

Figura 2.3: Level Lock [3]

2.2.3 Level Lock - Touch Edition

Level Lock este o incuietoare inteligenta de tip zavor. Are un design minimalist si ascunde partea electronica in interiorul usii pentru mai multa securitate.

Avantaje:

- Multiple modalitati de acces, printre care: amprenta, PIN
- Ofera alerte cand cineva inchide sau deschide usa
- Ofera integrare cu Google Home si Nest Home

2.2.4 Comparatii

Produsele de mai sus adreseaza probleme usor diferite, dar incearca sa ofere functionalitati similare. Sistemul oferit de Videx Security prezinta un design rezistent, dar familiar tuturor utilizatorilor si este destinat cladirilor cu mai multi locatari. In contrast, cele doua incuietori inteligente ofera o integrare avansata in reteaua IoT si multiple cai de acces, dar sunt destinate unei singure locuinte.

Incuietoarea de la Yale prezinta cea mai inovativa abordare a acestul design prin decizia deliberata de a nu oferi posibilitatea de acces cu cheie. Astfel, simplifica partea mecanica eliminand singura cale de acces din exterior catre mecanismul incuietorii.

Produsul celor de la Videx Security se bazeaza pe o tehnologie utilizata la scara larga si prin urmare beneficiaza de robustetea unui sistem matur. Spre deosebire de celelalte doua produse analizate, solutia celor de la Videx Security este agnostica de sistemul de operare al telefonului mobil, avand nevoie doar de o conexiune GSM.

Din lipsa unor standarde in domeniu, dispozitivele noi sufera de alte tipuri de probleme: [2]

2.3 Stabilirea cerințelor funcționale si nefuncționale ale sistemului

2.3.1 Controlul accesului intr-un apartament

Scopul principal al acestui sistem este de a oferi sau nu acces intr-o incinta, prin urmare consider aceasta cea mai importanta cerinta functionala.

2.3.2 Expunerea unui serviciu REST pentru interfatarea cu alte sisteme

Expunerea si abstractizarea terminalului POTS este realizata printr-un set de servicii Representational State Transfer (REST) care controleaza starea sa. Acest lucru ne permite interfatarea cu aplicatia mobila, interfata de administrare web si alte servicii precum Google Home/Google Assistant/Apple HomeKit.

2.3.3 Implementarea unei functii pentru raspuns automat

Aceasta functie va permite utilizatorului sa stabileasca o perioada de timp pentru care sistemul va oferi accesul neconditionat.

2.3.4 Dezvoltarea unui client mobil Android

Principalul client care va interactiona cu serviciile REST va fi aplicatia mobila ce va avea rolul de a notifica userul cand ii suna interfonul si de a controla starea sistemului.

2.3.5 Control granular asupra datelor stocate

Arhitectura aplicatiei necesita interactiunea cu o baza de date, care poate fi tinuta in cloud, pentru convenabilitate sau local. Folosind tehnologii de containerizare precum Docker, putem stoca baza de date local, informatiile fiind stocate intr-un mediu controlat.

2.3.6 Criptarea comunicatiilor cu serviciile web

Avand in vedere nivelul de acces pe care l-ar oferi un exploit al acestei solutii, comunicatiile intre server si clienti trebuie realizate printr-un canal criptat de tip Secure Sockets Layer (SSL). Credentialele userului si ulterior tokenul de acces trebuie trimise doar dupa verificarea autenticitatii serverului si a pachetelor trimise.

2.3.7 Oferirea si revocarea accesului la sistem

Dorim de exemplu sa oferim acces neconditionat unui prieten apropiat pentru a intra in bloc fara a mai suna la interfon. De asemenea ar trebui sa putem realiza si inversul acestei operatii.

2.3.8 Expunerea unui flux duplex audio prin tehnologia VoIP

Pasul final in dezvoltarea acestui sistem ar fi interfatarea cu un Analog to Digital Convertor (ADC) si un Digital to Analog Convertor (DAC) si expunerea streamurilor de date prin Voice Over IP (VoIP)

Capitolul 3

Stadiul actual in domeniu si selectarea soluției tehnice

3.1 Stadiul actual al tehnologiilor utilizate pentru dezvoltarea soluției

3.1.1 Hardware

Deoarece proiectul necesita atat interactiunea cu sisteme electrice cat si cu sisteme digitale precum stiva IP, am ales placa de dezvoltare ”Raspberry Pi 3 Model B Rev 1.2”. Aceasta ofera un procesor quad core cu arhitectura armv7 de 1.2 Ghz, 1 GB RAM si 26 de pini General-Purpose Input/Output (GPIO) pentru interactiunea cu terminalul POTS.

3.1.2 Backend

Intr-un studiu anual realizat de Stack Overflow, peste $80,000$ de dezvoltatori au ales JavaScript ca cel mai folosit limbaj de programare pentru al noualea an consecutiv. NodeJS a urcat pe locul 5 in popularitate, in timp ce Typescript este pe locul 6. Datorita cerintei de portabilitate am ales NodeJS ca limbaj pentru implementarea serverului aplicatiei. [4]

Printre alternative viabile pentru acest tip de proiect se numara Java, C# sau Python, limbaje aflate in primele 10 in topul celor de la Stack Overflow.

Ca framework de dezvoltare a serverului am ales NestJS, oferind o arhitectura Model View Controller (MVC) si multe functionalitati convenabile precum:

- Framework de injectare a dependintelor: graful (aciclic) de dependinte al aplicatiei este calculat la pornire, fiecarui modul ii sunt satisfacute dependintele, instantiindu-se obiectele necesare o singura data. Daca sunt detectati cicli in graful de dependinte sau nu exista informatii despre cum se poate instantia o clasa, atunci se va arunca o eroare de runtime si aplicatia va iesi cu un status code de eroare.
- Separarea logicii de control a aplicatiei de interfata si de date. Utilizatorul interactioneaza cu interfata, care notifica controllerul de actiunile utilizatorului, controllerul executa logica aplicatiei si actualizeaza modelul corespunzator, schimbari ce se vor reflecta in interfata.

- Imbina elemente de Object Oriented Programming (OOP), Functional Programming (FP) si Functional Reactive Programming (FRP). De exemplu: modulele si serviciile sunt clase, iar decoratorii claselor sunt practic functii cu care se compune functia tinta.

3.1.3 Baza de date

Din punct de vedere al scalabilitatii, pradigma relationala scaleaza vartical (putine servere puternice), pe cand cea nerelationala este orizontala (multe servere mici). Prin urmare am ales MongoDB, o solutie de tip NoSQL rulata in modul ”cluster” pentru a oferi redundanta datelor prin replicarea lor de 3 ori pe noduri diferite fizic.

Deoarece MongoDB are nevoie de suport pentru 64 biti, nu poate fi instalata pe acelasi Raspberry Pi unde va rula si serverul. Pentru simplitudine, am ales un serviciu online de hosting gratis, numit Mongo Atlas. Asadar, serverul NodeJS trebuie sa tina cont de eventuala latenta mai ridicata in comunicarea cu baza de date si retransmiterea comenzilor in cazul in care niciunul din nodurile clusterului nu este disponibil.

Object Document Mapping

Pentru transformarea si validarea obiectelor de JavaScript in documente Mongo, am ales Mongoose.

3.1.4 Client

3.2 Prezentarea tehnologiilor si platformelor de dezvoltare alese

Capitolul 4

Considerente legate de implementarea soluției tehnice

4.1 Arhitectura sistemului

Sistemul prezentat presupune atat o partare hardware, cat si una software. Hardwareul realizeaza adaptarea dintre terminalul analog POTS si placa digitala de dezvoltare Raspberry Pi, iar ca software am folosit NodeJS pentru server si Android pentru a implementa un client al serverului

4.1.1 Raspberry Pi HUT

Pentru a proiecta un Printed Circuit Board (PCB) am folosit softwareul Fritzing. Acesta permite proiectarea schemei electrice si ulterior trasarea conexiunilor pe layoutul fizic al placii.

Actionarea butoanelor terminalului POTS se realizeaza cu ajutorul unor opto-cuploare, izoland circuitul interfonului care este proiectat pentru a functiona cu spike-uri de pana la $90V$ de circuitul Raspberry Pi.

Detectarea unui apel este realizata prin legarea unui Metal Oxide Semiconductor Field Effect Transistor (MOSFET) la bornele difuzorului terminalului POTS si inserierea cu un amplificator operational in regim de comparator cu referinta de $0.1V$. Am folosit de asemenea si un Filtru Trece Jos deoarece terminalul este sensibil la zgomote, declansand accidental notificarea.

4.1.2 Webserver NodeJS

NodeJS este un

4.1.3 Android

Android este o platforma mobile care s-a maturizat pe parcusul a 12 versiuni majore si principalul competitor de piata al iOS.

4.2 Implementarea sistemului

4.3 Testarea sistemului

Capitolul 5

Studiu de caz

5.1 Raspuns automat

Mi-am comandat pizza si ajunge in timp ce folosesc pistolul de lipit. Prin urmare voi programa interfonul sa raspunda si sa deschida automat usa la urmatorul apel.

5.2 Raspuns de la distanta

Mi-am pierdut cartela standard Radio-Frequency Identification (RFID) pentru a intra in bloc. Asadar, voi suna la numerul apartamentului meu si voi folosi aplicatia mobila pentru a raspunde la propriul apel.

Capitolul 6

Concluzii

concluzia domnuleee?

Acronime

ADC Analog to Digital Convertor. 9

ARPANet Advanced Research Projects Agency Network. 4

DAC Digital to Analog Convertor. 9

FP Functional Programming. 11

FRP Functional Reactive Programming. 11

GPIO General-Purpose Input/Output. 10

GSM Global System for Mobile Communications. 5, 7

IoT Internet of Things. 3, 6, 7

IP Internet Protocol. 10

MOSFET Metal Oxide Semiconductor Field Effect Transistor. 12

MVC Model View Controller. 10

OOP Object Oriented Programming. 11

PCB Printed Circuit Board. 12

POTS Plain Old Telephone Service. 3, 5, 8, 10, 12

REST Representational State Transfer. 8

RFID Radio-Frequency Identification. 5, 14

SIM Subscriber Identity Module. 5

SSL Secure Sockets Layer. 8

VoIP Voice Over IP. 9

Capitolul 7

Bibliografie

[1] Frances K Aldrich. “Smart homes: past, present and future”. **in***Inside the smart home*: Springer, 2003, **pages** 18–18.

[2] Alexandra Gheorghe. *The Internet of Things: Risks in the connected home*. Research Paper BD-NGZ-86847. Bitdefender, 2016.

[3] Level Home. *Level Lock: The Smallest and Most Advanced Smart Lock Ever*. 2019. URL: https://level.co/products/lock (**urlseen** 29/04/2022).

[4] Stack Overflow. *Stack Overflow Developer Survey 2021*. 2021. URL: https://insights.stackoverflow.com/survey/2021#section-most-popular-technologies-programming-scripting-and-markup-languages (**urlseen** 05/06/2022).

[5] Videx Security. *GSM Intercoms*. 2016. URL: https://www.videxuk.com/system/gsm-intercoms (**urlseen** 28/04/2022).

[6] Google Store. *Nest x Yale Lock*. 2018. URL: https://store.google.com/us/product/nest_x_yale_lock?hl=en-US (**urlseen** 28/04/2022).

[7] Zeus Integrated Systems. *A Brief History of Smart Home Automation*. 2019. URL: https://zeusintegrated.com/blog/item/a-brief-history-of-smart-home-automation (**urlseen** 02/06/2022).