

Abstract

Lista figurilor

Figura 1: Schemă planificare design	8
Figura 2: Structura unei componente angular	10
Figura 3: Structura aplicației web	11
Figura 4: Codul de rutare	12

Cuprins

Abstract i

Lista figurilor.....ii

Cuprins iii

1.	Introducere.....	1
1.1.	Context.....	1
1.2.	Motivație.....	1
1.3.	Ce este „Smart Home”?.....	2
1.4.	Facilitățile traiului într-o casă inteligentă.....	3
1.5.	Provocări	5
2.	Proiectare	7
3.	Arhitectura sistemului.....	9
3.1.	Modulul 1 – Aplicația web.....	9
3.1.1.	Structura proiectului	10
3.1.2.	Rutarea	11
3.1.3.	AuthGuard.....	12
3.1.4.	Librării folosite.....	13
3.2.	Modulul 2 – Server Rest Java	14
3.3.	Modulul 3 – Dispozitivele inteligente.....	15
4.	Direcții de viitor	16
5.	Concluzii	17
6.	Bibliografie	18

1. Introducere

1.1. Context

Persoanele își petrec cea mai mare parte a timpului la domiciliu sau la locul de muncă; pentru mulți, aceste locuri sunt sanctuarele lor. Pe parcursul secolului al XX-lea, progresele tehnologice au contribuit la sporirea confortului și protecției oferite de casele noastre. Observațiile asupra mediului casnic și modelarea comportamentului său sunt utile în a face mediile mai inteligente și mai receptive la nevoile noastre. Progresele recente au adus o astfel de „intelligență ambientală” mai aproape de realitate. Chiar dacă ideea de casă inteligentă există de ceva vreme, case inteligente reale există doar de puțin timp.

De la miniaturizarea microprocesoarelor, puterea de calcul a fost încorporată în obiecte familiare, cum ar fi aparatele de uz casnic și dispozitivele mobile, regăsindu-se aproape la toate nivelurile societății. Inteligența ambientală extinde noțiunea de calcul pentru a oferi suport personalizat și automatizat în viețile noastre.

Ideea [de casă inteligentă] este următoarea: software-ul de calculator joacă rolul unui agent inteligent ce percepe starea mediului fizic și a rezidenților utilizând senzori, raționează în legătură cu starea acestora folosind tehnici de inteligență artificială și apoi ia măsuri pentru a atinge obiectivele specificate, cum ar fi maximizarea confortului locuitorilor, minimizarea consumul de resurse și menținerea sănătății și siguranței locuinței și a locuitorilor. [1]

1.2. Motivație

Primele case inteligente au fost idei, nu construcții reale. Timp de decenii, literatura științifico-fantastică a explorat ideea automatizării locuințelor. Producători prolifici, cum ar fi Ray Bradbury¹, și-au imaginat un viitor în care casele vor fi interactive. În scurta povestire a lui Bradbury, „There Will Come Soft Rain”², se descrie o casă automată care continuă să funcționeze chiar și după ce oamenii au dispărut. Sună un pic înfricoșător, până când luați în considerare beneficiile reale ale automatizării locuințelor, iar apoi ideea devine mai mult confortabilă decât descurajatoare.

¹ Ray Douglas Bradbury (n. 22 August 1920 – d. 5 Iunie 2012) – scriitor de romane științifico-fantastice, fantezie, horror și mister.

² „Va veni ploaie ușoară” – povestire scurtă publicată pe data de 6 Mai 1950.

Atunci când nu sunteți acasă, aveți tot felul de suspiciuni care vă îngrijorează? Am oprit cafetiera? Am închis ușa de la intrare? Oare am lăsat apă aprinsă în baie? Cu o locuință inteligentă, aceste îngrijorări nu-și mai au locul. Puteți afla informații despre locuința dumneavoastră folosindu-vă de tabletă, telefon sau calculator personal. Puteți conecta dispozitivele și aparatele de la domiciliu pentru a comunica cu ele. De altfel, dispozitivele pot fi programate pentru a comunica între ele.

Lucrare de față își propune implementarea ideii de casă inteligentă și aducerea la cunoștiință, prin exemple practice, a avantajelor folosirii unui astfel de sistem. În exemplele oferite, voi folosi drept locuință inteligentă un Raspberry Pi 2, Model B, iar dispozitivele controlate de utilizator vor fi module arduino³.

Inovația acestei lucrări constă în implementarea unui sistem de la zero, cu o arhitectură unică, folosind tehnologii de ultimă generație. Arhitectura sistemului este formată din 3 componente: partea de front-end⁴ care are drept scop interacțiunea eficientă și facilă cu utilizatorul; partea de back-end⁵ ce constă într-un server java care se ocupă de managementul utilizatorilor și al dispozitivelor, oferind siguranță, încredere și eficiență; partea de dispozitive ale utilizatorului ce folosește un protocol de comunicare securizată cu partea de back-end. Protocolul de comunicare oferă un management al erorilor foarte strict pentru a ține utilizatorul la curent cu starea dispozitivelor chiar și în cazuri de excepție.

1.3. Ce este „Smart Home”?

„Smart Home” este termenul utilizat în mod obișnuit pentru a defini o reședință care are iluminat, încălzire, aer condiționat, televizoare, calculatoare, sisteme audio și video de divertisment, sisteme de securitate și camere de luat vederi capabile să comunice între ele și pot fi controlate de la distanță: din orice cameră din casă, precum și din orice locație din lume, prin telefon sau prin internet.

Smart home⁶ este un sistem care oferă deținătorilor locuinței confort, securitate, eficiență energetică (costuri de operare scăzute) și comoditate în orice moment, indiferent dacă este cineva acasă sau nu.

³ Un modul arduino este un circuit hardware complex, adus la o formă compactă pentru utilizare, cu o interfață de conexiune simplă, ce necesită doar atasarea unor fire între placă și circuit.

⁴ Acea parte a site-ului pe care o putem vedea și cu care interacționează vizitatorii.

⁵ Este locul unde se administrează informația. De obicei este format din server și bază de date.

⁶ Echivalentul acestui termen în limba română este „Casă Inteligentă”.

Instalarea de produse inteligente oferă locuinței și ocupanților săi diverse beneficii - aceleași avantaje pe care tehnologia și computerele personale le-au adus în ultimii 30 de ani - confort și economii de timp, bani și energie. [2]

1.4. Facilitățile traiului într-o casă inteligentă

Multora nu le place ideea de „Smart Home” considerând că va fi prea costisitoare, greu de folosit, sau nu vor avea suficient control asupra propriului mediu. Cu toate acestea, casele inteligente devin foarte accesibile, datorită opțiunii de a construi puțin câte puțin, înglobând părțile într-un proiect mare, lăsându-se loc pentru îmbunătățiri viitoare. Fiecare dispozitiv vă stă la dispoziție pentru a vă face viața mai confortabilă.

În continuare voi enumera câteva motive pentru care ar trebui luați în considerare un smart home: [3]

1. Confort

Una dintre cele mai bune părți ale locuinței inteligente este că o puteți configura pentru a corespunde cu exactitate necesităților dumneavoastră. Lampa trebuie să fie configurată la o luminozitate corectă, nu prea luminoasă, dar nici prea întunecată. Puteți chiar să setați aprinderea luminilor treptat, evitând astfel blițul brusc de lumină orbitoare.

Puteți să setați o melodie liniștită atunci când alarmele se sting dimineața, astfel încât să vă treziți într-o atmosferă plăcută. Puteți amplasa difuzoarele în mai multe încăperi pentru a fi programate să difuzeze diferite melodii la diferite volume, astfel încât să puteți asculta mereu ceea ce doriți în orice loc din casă.

De asemenea, aveți posibilitatea ca temperatura casei să fie menținută la valoarea optimă. Instalația de încălzire și climatizare poate fi controlată de la distanță printr-un termostat, astfel încât în casă să nu fie niciodată prea rece sau prea cald.

2. Securitate

Există nenumărate motive pentru care o casă inteligentă vă poate menține în siguranță. Puteți avea senzori specifici pentru a detecta imediat mișcarea, cum ar fi noaptea sau în timp ce vă aflați la serviciu. Dacă detectorii de mișcare sunt avertizați, vi se poate trimite o notificare telefonului dumneavoastră și/sau se pot aprinde luminile pentru a da impresia că sunteți treaz, acest lucru sperându-l pe potențialul spărgător.

Puteți avea senzori pe ferestre și uși, astfel încât să știți dacă au fost deschise sau dacă au fost distruse.

O mulțime de case inteligente au camere video instalate care vă permit să monitorizați orice cameră utilizând dispozitivul inteligent sau terminalul de perete. Puteți seta secțiuni ale ecranului pentru a detecta mișcarea, astfel încât un animal de companie sau un copac în vânt să nu declanșeze alarma.

Dacă vă montați alarme „inteligente” de incendiu, puteți primi notificări despre potențiale incendii, indiferent dacă vă aflați în casă sau la locul de muncă. Dacă știți că este o alarmă falsă, puteți opri alarma printr-o simplă atingere de ecran, în loc să urcați pe un scaun și să vă întindeți pentru a ajunge la butonul de alarmă.

Mai mult, dacă plecați vreodată în vacanță puteți alege între a obține personal notificări despre cazurile excepționale sau a fi transmise unui vecin ori unui membru al familiei.

3. Ușor de utilizat

Partea cea mai convenabilă a unei locuințe inteligente este că fiecare parte a casei ar putea fi doar la o atingere de ecran(sau apăsarea unei taste) distanță. Fie că este vorba de telefonul tău inteligent, tabletă, calculator personal sau un terminal montat în perete.

Când te pregătești să dormi, puteți opri orice lumină din casă doar prin apăsarea unui buton. Puteți să vă uitați la ecranul dumneavoastră și să vedeți că toate ușile și fereastrele sunt blocate, astfel încât să nu trebuiască să mergeți și să le verificați pe fiecare în parte.

Puteți chiar crea setări pentru grupuri de dispozitive, de exemplu, când porniți televizorul, luminile se diminuează automat sau sunetul se aprinde odată cu televizorul. Dimineata puteți seta încălzirea, luminile, muzica pentru a se aprinde când vă ridicați din pat, fără să trebuiască să faceți nimic.

4. Accesibilitate

Casele inteligente oferă un ajutor imens pentru persoanele cu diverse dizabilități. Oamenii care nu văd pot avea interfață vocală cu care pot controla televizoarele, luminile, încălzirea, orice dispozitiv ce este conectat la electricitate și internet. Pentru persoanele cu dizabilități, precum deficiențe musculare, activarea comutatoarelor de lumină sau alarmei de incendiu se poate dovedi a fi un lucru dificil, dacă nu chiar imposibil. Acum, astfel de dispozitive, se pot controla folosind un telefon sau o tabletă. Pentru cineva într-un scaun cu roțile se poate configura deschiderea și închiderea ușilor automat folosind senzori.

Programele pot avea setări personalizate în funcție de utilizatori, astfel scutindu-i de preocupări triviale, cum ar fi setarea zilnică a încălzirii sau luminii.

Familiile care au un membru în vârstă sau poate pe cineva ce suferă de o boală precum Alzheimer⁷ pot avea senzori instalați în casă pentru a fi mai simplă monitorizarea de la distanță a persoanelor în nevoie. Se pot crea alerte care să informeze în caz că ușa frontală este deschisă pe timp de noapte, pot fi atașați senzori la chei pentru a afla dacă se rătăcesc și nu sunt unde vă așteptați să fie. De altfel, se pot crea alerte de inundații în camerele de bucătărie, baie, chiar și monitorizarea casetelor de pastile pentru a vă asigura că sunt luate în fiecare zi.

De asemenea, pot fi instalate cu ușurință butoane de panică, în cazul în care a existat un accident în casă, iar familia sau personalul de îngrijire poate fi anunțat imediat, scutind persoana în nevoie să telefoneze pentru ajutor.

1.5. Provocări

O casă inteligentă poate fi un coșmar pentru acei oameni ce nu se simt confortabil cu tehnologia de ultimă generație.

Unul dintre blocajele esențiale în instalarea unui sistem smart home este menținerea echilibrului dintre complexitatea și gradul de dificultate al utilizării lui. Dacă este exasperant de utilizat, atunci o să vă facă viața mai grea în loc să o facă mai ușoară. Când planificați sistemul, este important să luați în considerare câțiva factori:

- Ce tipuri de componente fac parte din sistem? Sunt de bază, de dimensiuni mici sau impunătoare, cum ar fi un sistem de alarmă sau o cameră video?
- Cât de intuitiv va fi sistemul pentru un non-utilizator?
- Dispozitivul îndeplinește o nevoie sau este doar o fantezie și potențial o jucărie frustrantă?
- Câți oameni vor fi nevoiți să utilizeze sistemul?
- Cine va ști cum să opereze sistemul? Cine va ști cum să mențină sistemul și să remedieze eșecurile?
- Cât de ușor este să faceți schimbări în interfață? De exemplu, dacă casa ta este programată să te trezească la 7 dimineața, cum o vei lăsa să știe că ești deplasat peste noapte la birou sau că dorești să dormi mai mult într-o sâmbătă?

Din aceste motive, ar putea fi mai ușor să începeți cu o rețea foarte simplă și apoi să o extindeți atunci când sunt necesare sau dorite îmbunătățiri. Ca multe dintre noile tehnologii, casele inteligente necesită o investiție semnificativă atât în bani, cât și în timp. În caz contrar,

⁷ Alzheimer este un tip de demență care cauzează probleme cu memoria, gândirea și comportamentul.

dacă nu aveți nici bani, nici timp, ați putea dori să rămâneți la casa dumneavoastră „veche” și „neinteligentă”.

Înainte de a cumpăra, verificați recenziile despre produse. Există o mulțime de produse care fac promisiuni înalte, dar în lumea reală nu au succes. Și dacă sunteți un utilizator de smartphone, luați în considerare produsele care apar și care au construite pentru ele o aplicație smartphone bine revizuită. Unele aplicații sunt atât de greoaie sau complicate încât provoacă mai multe dureri de cap decât vă ușurează viața.

Casele inteligente vin, de asemenea, cu unele probleme de securitate. Hackerii care găsesc o modalitate de a accesa rețeaua internă pot să dezactiveze sistemele de alarmă, luminile, lăsând locuința vulnerabilă la o spargere. De asemenea, ar putea provoca neplăceri, cum ar fi aprinderea și închiderea rapidă a dispozitivelor electronice, ceea ce ar putea dăuna funcționării sau, într-un caz extrem, ar putea provoca un incendiu.

Producătorii produselor electronice de uz casnic își îmbunătățesc liniile de producție, în speranța că automatizarea locuințelor va ajunge în sfârșit să se realizeze în masă. Mulțumită smartphone-urilor, tabletelor și numeroaselor aplicații de automatizare a locuințelor, disponibile acum, există o șansă că trendul va atrage mai mulți utilizatori.

Asta pentru că, în ciuda atâtor progrese tehnologice, nu există încă un sistem standard pentru automatizarea tuturor acestor gadgeturi⁸. Fără un astfel de standard, mulți consumatori sunt supuși riscului de a cheltui sute sau mii de dolari pe produse care vor sfârși depășite sau inutilizabile într-un scurt timp.

Desigur, se pune problema dacă o persoană are nevoie de toată această tehnologie. Societatea noastră este într-adevăr atât de leneșă și comodă încât nu putem apăsa pe un comutator de lumină? Indiferent de răspunsul acestei întrebări, această tehnologie are drept scop satisfacerea altelor puncte de interes, și anume: datorită timpului pe care îl vom economisi din automatizarea locuinței, vom avea timp de mai multe activități mai puțin triviale. [4]

⁸ Un gadget este un obiect tehnologic mic care îndeplinește o anumită funcție, de obicei fiind ceva nou.

2. Proiectare

În ceea ce urmează o să descriu planul abordat de către mine pentru a pune în practică ideea de smart home.

Sistemul trebuie să fie accesibil din orice locație din lume și din această cauză am ales ca interfața cu utilizatorul să fie o aplicație web. Aplicația poate fi accesată de pe telefon, tabletă, calculator personal atât timp cât există conexiune la internet; ceea ce nu ar fi fost la fel de accesibil cu o aplicație Android sau IOS.

Pentru ca dispozitivele din casa dumneavoastră să nu fie accesate de persoane neautorizate, fiecare utilizator va avea un cont personal, ce va fi accesat pe baza unor credențiale alese.

Partea de back-end, mai precis spus serverul de back-end, trebuie să ofere:

- un mediu persistent de stocare a datelor;
- înregistrarea, autentificarea⁹ și managementul utilizatorilor;
- înregistrarea și controlul dispozitivelor expuse de utilizator;
- managementul conexiunilor cu dispozitivele: inițiere, comunicare propriu zisă, finalizare;
- stabilirea unui protocol de comunicare cu dispozitivele „smart”.

Odată stabilite responsabilitățile serverului de back-end, mai rămâne de stabilit modalitatea de conectare a dispozitivelor smart cu partea de back-end și implementarea lor propriu zisă.

Ce limbaj de programare trebuie folosit pentru a programa cât mai multe tipuri de clienți, dacă nu chiar toți clienții posibili? Am ales java deoarece este un limbaj independent de platforma de lucru, aceeași aplicație rulând fără nicio modificare și fără a necesita recompilarea ei pe sisteme de operare diferite cum ar fi Windows, Linux, Mac OS, după cum se înțelege și din sloganul Java „Write once, run everywhere”. Alte calități sunt: simplitate, robustețe, garbage collector¹⁰, portabilitate.

Prin urmare clienții trebuie să poată rula cod sursă scris în java și ar trebui să aibă o conexiune la internet, având în vedere că dispozitivele pot fi controlate de la distanță.

Comunicarea cu serverul de back-end ar trebui să fie criptată astfel încât un posibil atacator să nu poată descifra nimic din ceea ce se transmite pe rețea. În plus, se pune și problema

⁹ Autentificarea are drept scop stabilirea identității actorilor care doresc să comunice sigur în rețea.

¹⁰ Este un program care face managementul memoriei interne automat.

de certificare a serverului și evitare a unui posibil atac de tip „Man-in-the-Middle”¹¹. O să descriu în continuare acest tip de atac aplicat pe acest sistem. Un atacator a aflat protocolul de comunicare a serverului cu dispozitivele. Un dispozitiv al unui utilizator este conectat la internet și primește o cerere de conectare de la atacator, care pretinde că este serverul de back-end. Dacă nu ar fi nici o metodă de identificare a serverului real, atunci întreg sistemul este compromis, atacatorul având posibilitatea sa controleze orice dispozitiv disponibil.

Prin urmare, pentru a evita o un astfel de problemă care dovedește sistemul a fi inutilizabil, am să folosesc comunicare SSL¹² între server și client. SSL este un protocol de securitate care poate asigura confidențialitate, integritatea mesajelor și autentificarea părților. SSL acționează asupra unui flux TCP¹³ și oferă servicii nivelurilor superioare.

În Figura 1 de mai jos se poate observa o schiță a design-ului sistemului.

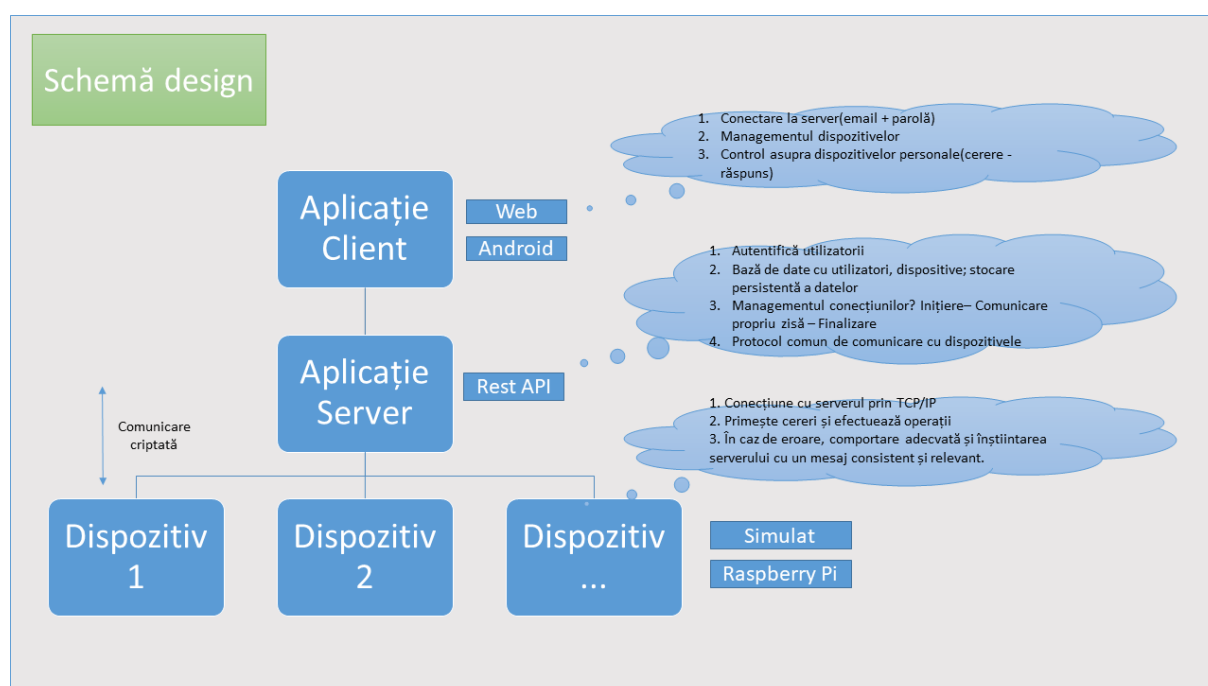


Figura 1: Schemă planificare design

¹¹ Man-in-the-middle – detalii aici: <https://arxiv.org/ftp/arxiv/papers/1504/1504.02115.pdf>

¹² SSL - Secure Sockets Layer

¹³ TCP – Transport Control Protocol

3. Arhitectura sistemului

Rezultatul proiectării a dus la o arhitectură formată din trei module principale. Primul modul constă într-o aplicație web care are drept scop principal comunicarea și interacțiunea cu utilizatorul. Al doilea modul este o aplicație java, ce servește pe post de server, fiind „creierul sistemului”. Am numit acest modul astfel deoarece face legătura între ceea ce vede utilizatorul și cum se desfășoară firul logic al acțiunilor în spate. Al treilea modul este format din aplicații client ce vor juca rolul de dispozitive din cadrul unei case inteligente.

DETALIERE

Modulul 1 – Aplicația web

Modulul de aplicație web este foarte important, deoarece utilizatorul va interacționa cu sistemul folosind interfața web.

DETALIERE

De aceea, am hotărât să folosesc pentru acest modul două dintre cele mai populare tehnologii web: Angular¹⁴ și Bootstrap¹⁵.

Am ales Angular pentru că este o bibliotecă ce ajută la crearea aplicațiilor SPA¹⁶. Datorită acestui fapt, pagina este fragmentată în mai multe componente ce se încarcă dinamic atunci când utilizatorul interacționează cu aplicația. Această abordare bazată pe componente face ca aplicația să se încarce foarte rapid, să fie modularizată, evitând duplicarea de cod.

Apoi, am ales Bootstrap deoarece oferă un suport foarte bun în crearea de site-uri responsive, o documentație foarte structurată cu exemple, ceea ce face developmentul foarte rapid.

DETALII RESPONSIVE

¹⁴ Angular: <https://cli.angular.io/>

¹⁵ Bootstrap: <https://getbootstrap.com/>

¹⁶ Single-page application: o aplicație ce interacționează cu utilizatorul rescriind dinamic pagina curentă, în loc să încarce pagini întregi primite de la server.

Structura proiectului

În Figura 2 se poate observa structura unei componente Angular. O componentă conține un fișier .css¹⁷, unul .html¹⁸ și altul .ts¹⁹. Acest mod de structurare al unei componente îi permite să nu depindă de altă componentă, astfel reducând cuplajul din interiorul aplicației.

DETALII CUPLAJ

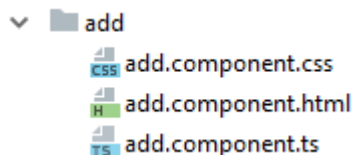


Figura 2: Structura unei componente angular

În Figura 3 se poate observa structura aplicației web, ce conține următoarele module:

- *backendservice*: conține un serviciu responsabil de cererile către serverul java și alt serviciu ce este responsabil de autentificarea utilizatorului când se încearcă accesarea anumitor funcționalități.
- *device*: conține componente ce descriu operațiile ce pot fi aplicate asupra dispozitivelor: add – adăugare, manage – management, view – vizualizare.
- *deviceTypes*: conține mai multe componente ce descriu interfețele specifice în funcție de dispozitive: lampă, laser de securitate, încuietoearea ușii, bec, etc.
- *footer*: definește subsolul site-ului.
- *header*: definește antetul site-ului. Antetul conține meniul principal.
- *home*: componentă responsabilă de pagina principală.
- *login*: componentă ce se ocupă de logarea utilizatorului.
- *notfound*: componentă ce va apare de fiecare dată când utilizatorul va încerca să acceseze o resursă ce nu este disponibilă.
- *profile*: componentă ce se ocupă cu managementul profilului utilizatorului.
- *register*: componenta cu care interacționează utilizatorul când se înregistrează.

Fișierele care încep cu prefixul app sunt folosite pentru a configura pornirea aplicației.

¹⁷ Cascade Style Sheets(foaie de stiluri CSS): standard pentru formatarea elementelor unui document html.

¹⁸ Hyper Text Markup Language: limbajul principal al Web-ului pentru crearea de conținut ce poate fi utilizat oriunde.

¹⁹ TypeScript: limbaj puternic tipizat, orientat obiect, compilat, superset al lui JavaScript.

Rutarea

Pentru a explica ce este rutarea, mai întâi trebuie să explic ce este un view. O grupare de elemente prezentate utilizatorului ce se creează și distrug împreună formează un view.

Rutarea constă în navigarea de pe un view pe următorul view, pe măsură ce utilizatorii execută operații în aplicație.

Figura 4 conține codul corespunzător care efectuează rutarea. Se exportă un router²⁰ care va fi inclus în modulul aplicației. Odată adăugat, el va efectua rutarea pe componenta specificată.

Obiectul ce conține rutele este o listă, iar în interior conține dicționare ce au următorul format: path, component, canActivate. Path conține un string cu url²¹-ul relativ la aplicație. Component definește ce clasă o să fie încărcată, iar canActivate definește dacă se poate accesa sau nu ruta respectivă. Dacă canActivate lipsește, atunci se poate accesa ruta fără nici o restricție, altfel se va decide dacă se poate sau nu accesa ruta.

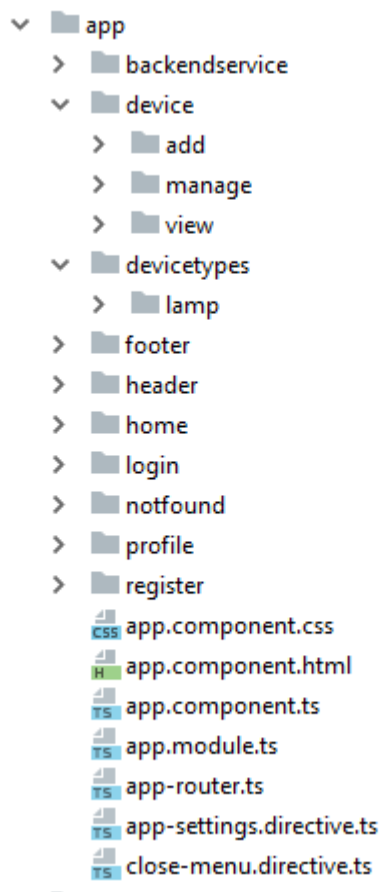


Figura 3: Structura aplicației web

```
export const router: Routes = [  
  {path: '', component: HomeComponent},
```

²⁰ Obiect responsabil cu navigarea de pe un view pe următorul view.

²¹ URL (Uniform Resource Locator): localizator uniform de resurse.

```

    {path: 'home', component: HomeComponent},
    {path: 'login', component: LoginComponent},
    {path: 'register', component: RegisterComponent},
    {path: 'profile', component: ProfileComponent, canActivate: [AuthGuard]},
    {path: 'devices', component: ViewComponent, canActivate: [AuthGuard]},
    {path: 'devices/add', component: AddComponent, canActivate: [AuthGuard]},
    {path: 'devices/manage/:id', component: ManageComponent, canActivate:
[AuthGuard]},
    {path: '**', component: NotFoundComponent},
];

export const RouterGlobal: ModuleWithProviders =
RouterModule.forRoot(router);

```

Figura 4: Codul de rutare

Consider oportună explicarea path-urilor „**”, „devices/manage/:id” întrucât sunt path-uri speciale. „**” este folosit pentru a controla rutele invalide, ceea ce o să declanșeze componenta de NotFound, indicându-i, într-o manieră elegantă, utilizatorului că a introdus un link greșit. „devices/manage/:id” să capteze rutele de forma: <http://localhost:4200/devices/manage/7018dea9-8cb6-4677-a5ea-cfb743ee4dca>, unde „7018dea9-8cb6-4677-a5ea-cfb743ee4dca” este id-ul unui dispozitiv. Se folosește „:id” pentru a avea acces direct la id, evitând astfel parsarea url-ului.

De exemplu, dacă aplicația rulează pe hostul local, la portul 4200, și utilizatorul introduce în browser următorul url: <http://localhost:4200/home>, atunci, conform ruter-ului definit, se va încărca componenta HomeComponent, fără nicio restricție de accesare. Dacă se introduce în browser url-ul: <http://localhost:4200/profile>, atunci ar trebui să se încarce ProfileComponent, dar AuthGuard va decide dacă se poate sau nu accesa. O să explic în următoarea secțiune ce este AuthGuard și care este rolul lui.

AuthGuard

Într-o aplicație web, există rute publice și rute ce pot fi accesate dacă se îndeplinesc anumite condiții. Una dintre aceste condiții este starea de logare a utilizatorului. Acesta este și cazul de față, AuthGuard fiind un serviciu care verifică starea de logare a utilizatorului când accesează anumite rute.

Paginile publice din această aplicație sunt: pagina principală, pagina de login, pagina de register. Paginile verificate de serviciu sunt: pagina de profil și paginile ce țin de dispozitive. Este necesară verificarea stării de logare, deoarece utilizatorul vizitator (ne-logat) nu are acces la principalele funcționalități oferite de aplicație, pentru aceasta fiind nevoie de un profil înregistrat, însoțit de logarea propriu-zisă.

Librării folosite

În crearea aplicației am folosit următoarele librării:

- ngx-toastr [5]
- highcharts [6]
- fontawesome: [7]

REFERENCE

Ngx-toastr este o librărie ce am folosit-o pentru a afișa mesaje toast utilizatorilor. Un mesaj toast este un mesaj scurt într-un popup. Toast-ul dispare dacă se apasă pe el sau automat, după un anumit timp.

DETALIERE: AVANTAJ TOAST DE CE AM FOLOSIT TOAST

HighCharts oferă suport pentru grafice scrise în javascript. Am folosit un grafic la design-ul interfeței obiectului lampă, atunci când se setează culoarea.

DETALIERE

FontAwesome l-am folosit pentru diverse pictograme pe care le oferă.

DETALIERE

3.1. Modulul 2 – Server Rest Java

După cum am zis și la începutul acestui capitol, acest modul este „creierul sistemului”.

Am ales să folosesc pentru acest server Spring Boot²², un framework de java foarte popular în construirea de aplicatii web și enterprise. Nu este prima dată când interacționez cu acest framework, rămânând cu un gust plăcut după fiecare proiect scris în acest framework.

²² Spring: <https://spring.io/projects/spring-boot>

3.2. Modulul 3 – Dispozitivele inteligente

4. Direcții de viitor

5. Concluzii

6. Bibliografie

- [1] D. J. Cook, „How Smart Is Your Home?,” *Science (New York, NY)*, vol. 335, nr. 6076, pp. 1579-1581, 2012.
- [2] ***, „What is a Smart Home,” n.d. n.d. c2018. [Interactiv]. Available: <https://www.smarthomeusa.com/smarthome/>. [Accesat 3 06 2018].
- [3] Homeoftech, „7 Benefits of living in a smart home,” 24 05 2016. [Interactiv]. Available: <http://homeoftechnologies.co.uk/7-benefits-of-living-in-a-smart-home/>. [Accesat 03 06 2018].
- [4] N. C. Molly Edmonds, „How Smart Homes Work,” 25 03 2008. [Interactiv]. Available: <https://home.howstuffworks.com/smart-home6.htm>. [Accesat 03 07 2018].
- [5] S. Cooper, „ngx-toastr,” [Interactiv]. Available: <https://github.com/scttcper/ngx-toastr>. [Accesat 04 05 2018].
- [6] „Highcharts” [Interactiv]. Available: <https://www.highcharts.com/>.
- [7] Komei, „Angular5 Fontawesome,” [Interactiv]. Available: <https://github.com/travelist/angular2-fontawesome>. [Accesat 25 05 2018].