

UNIVERSITATEA POLITEHNICA BUCUREȘTI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE



PROIECT DE DIPLOMĂ

Portal de căutare anunțuri imobiliare

Smeadă Ion Cosmin

Coordonator științific:

Ș.L. dr. ing. Carmen Odubășteanu

BUCUREȘTI

2018

CUPRINS

1	INTRODUCERE	4
1.1	Context	4
1.2	Problema	4
1.3	Obiective	5
1.4	Soluția propusă.....	5
1.5	Rezultatele obținute	6
2	ANALIZA CERINȚELOR	6
3	STUDIU DE PIAȚĂ	13
3.1	Analiza platformelor existente pentru anunțuri imobiliare.....	13
3.2	Tehnologii folosite in dezvoltarea aplicației	17
3.2.1	Ionic framework	17
3.2.2	HTML si CSS.....	18
3.2.3	Angular si TypeScript.....	20
3.2.4	Java	20
3.2.5	Spring Framework	22
3.2.6	Maven	22
3.2.7	Hibernate	23
3.2.8	POSTGRESQL.....	24
4	SOLUȚIA PROPUȘĂ	25
5	DETALII DE IMPLEMENTARE	31
6	CONCLUZII	36
7	BIBLIOGRAFIE	38

Platformele de căutare pentru vânzări sau închirieri imobiliare sunt de mult timp pe piață și sunt folosite de cât mai mulți utilizatori pentru că în ultimii ani această industrie a cunoscut o dezvoltare destul de mare. Astfel numărul vânzătorilor, cât și cel al cumpărătorilor, a crescut semnificativ iar nevoia de astfel de platforme este din ce în ce mai mare fiind una o modalitate ușoară de informare și , eventual, cumpărare pentru o persoană aflată în căutarea unei locuințe.

Scopul acestui proiect este acela de a realiza o platformă ușor de utilizat, atât pentru cumpărători dar și pentru vânzători , prin oferirea unei aplicații pentru telefoanele mobile indiferent de sistemul de operare pe care funcționează acestea, prin care utilizatorii au acces la o mulțime de anunțuri imobiliare, fiecare având posibilitatea de a cauta ce i se potrivește cel mai bine. Utilizatorii pot adăuga anunțuri, pot cauta prin anunțurile deja existente pe platformă, iar dacă dețin un cont de administrator pot modifica sau chiar șterge anunțuri.

1 INTRODUCERE

1.1 Context

Piață imobiliară este una dintre cele mai bine dezvoltate industrii din România, fiind într-o continuă creștere datorată ascensiunii economice pe care a cunoscut-o țara noastră în ultimii ani. Conform studiilor efectuate de specialiștii din domeniu a rezultat faptul că și în anul 2018 această industrie este în continuă creștere , cererea de apartamente și garsoniere fiind tot mai mare, iar acest lucru duce la atragerea cât mai multor dezvoltatori imobiliari.

În perioada anilor 90 prețul unui automobil depășea prețul unei case, dar 10 ani mai târziu prețurile locuințelor atingeau sume greu de imaginat pentru eventualii cumpărători din acea perioadă. Vârful pieței imobiliare a fost atins în anul 2008 când prețul pe metrul pătrat depășea suma de 2500 de euro. În acea perioadă, cunoscută și ca boom-ul imobiliar, prețurile locuințelor din capitala României erau mult mai ridicate decât prețurile din orașele mari ale Europei cum ar fi Paris, Roma sau Viena. La finele anului 2008 criza economică mondială a dus prăbușirea pieței imobiliare astfel încât prețurile de achiziție pentru locuințe au scăzut cu mai mult de 50%. Dacă înainte de criză o garsonieră ajunsese să coste în jur 80.000 Euro, în momentul de față aceeași garsoniera costă undeva la 35000-37000 Euro[1].

1.2 Problema

Prima problemă care se pune atunci când vine vorba de cumpărarea unei locuințe este prețul de achiziție al acesteia. Majoritatea oamenilor care își cumpără o locuință nu au puterea financiară necesară pentru a plăti integral și de aceea optează la credite

bancare. Chiar și pentru închiriere prețul reprezintă principalul criteriu de căutare după care se orientează majoritatea clienților.

Cea de-a doua problemă care apare este locația unde se află locuința. Prețurile diferă foarte mult în funcție de zonă de amplasare și de aceea aceste două criterii reprezintă principalele filtre după care să face căutarea anunțului potrivit pentru fiecare. Conform unor studii de piață realizate de specialiști în domeniul imobiliar, primele locuri în topul celor mai căutate cartiere din capitala României sunt Militari și Titan[2]. Acest lucru este o consecință a raportului preț-zonă foarte bun dar și a existenței diversificate a mijloacelor de transport în comun(metrou, ratb).

1.3 Obiective

Proiectul propune dezvoltarea unui portal de căutare anunțuri imobiliare sub forma unei aplicații mobile, ușor de înțeles și de utilizat prin intermediul căreia se pot căuta, dar și adăuga oferte. De la an la an numărul tranzacțiilor a crescut astfel încât în anul 2017 a ajuns la aproximativ 580.000, chiar dacă prețurile au crescut în medie cu 10%[3]. Toate aceste unități vândute sunt promovate pe diferite site-uri înainte de a-și găsi noul proprietar și de aici apare nevoia de a crea astfel platforme.

Prin utilizarea acestei aplicații clientul are posibilitatea de a-și căuta locuința în funcție de nevoile sale , de bugetul deținut dar și de zonă în care dorește să locuiască. Tot aici poate să găsească și informații legate de modul în care poate lua legătura cu vânzătorul sau dezvoltatorul. Pe de altă parte, vânzătorii au posibilitatea de a-și prezenta oferta cât mai multor persoane care pot deveni ulterior clienți. Aceștia își promovează anunțul, oferă detalii legate de preț, zona în care se află amplasamentul și oferă un număr de telefon prin care pot fi contactați.

1.4 Soluția propusă

Pentru a promova cât mai simplu anunțurile de pe piața imobiliară și pentru a le aduce cât mai aproape de eventualii clienți am implementat o aplicație mobilă prin intermediul căreia se oferă accesul la un portal de anunțuri. Aplicația este disponibilă pe

orice sistem de operare adresându-se unei game largi de utilizatori aflați în căutarea unei locuințe.

Căutarea anunțurilor se face pe baza unor filtre pe care fiecare potențial client le completează în funcție de necesități. Interfața este una simplistă, ușor de înțeles și de utilizat, iar informațiile prezentate sunt scurte și la obiect. În urma selectării filtrelor aplicația va întoarce o serie de anunțuri care respectă dorințele clientului , oferindu-i posibilitatea de a alege casa perfectă pentru el. Și din punct de vedere al vânzătorului aplicația este la fel ușor de utilizat. Publicarea anunțului se face prin completarea unui formular în care sunt prezentate câteva informații de bază cum ar fi comunicarea prețului de vânzare sau de închiriere, zona de amplasare a imobilului. Trebuie completată și o scurtă descriere a locuinței în care vânzătorul trebuie să își prezinte cât mai concis bunul dar și un număr de telefon la care poate fi contactat de eventualii cumpărători.

1.5 Rezultatele obținute

În urma dezvoltării aplicației am obținut un portal pentru anunțuri imobiliare disponibil pentru două categorii de utilizatori, cumpărători și vânzători. Prin intermediul telefonului mobil aceștia pot efectua operațiile de care au nevoie, căutare de locuințe cu scopul cumpărării sau închirierii , sau dat spre vânzare sau închiriere a unui imobil.

Interfața este simplistă și ușor de folosit astfel încât prin completarea unor formulare specifice pentru fiecare operație se pot căuta sau adăuga anunțuri.

2 ANALIZA CERINȚELOR

În ultimii ani piața imobiliară a cunoscut o creștere semnificativă iar atenția cumpărătorilor s-a concentrat asupra complexurilor rezidențiale noi , atât în București cât și în restul țării. Dezvoltarea României pe plan economic reprezintă principalul motiv pentru care piața imobiliară este în continuă ascensiune atrăgând o mulțime de investitori

atât români cât și străini. Nevoia de promovare a acestor complexuri a dus la apariția mai multor platforme online prin intermediul cărora se realizează promovarea anunțurilor indiferent că este vorba de vânzare sau de închiriere.

Apariția acestor complexuri rezidențiale a dus la orientarea cumpărătorilor spre locuințe cât mai noi în detrimentul celor construite înaintea anilor 2000. Chiar dacă amplasamentele noi se află într-o zonă mai îndepărtată sau nu așa bună ca și cele vechi , acestea sunt totuși preferate de către cumpărători. Acest lucru este datorat și guvernului care în anul 2009 a lăsat programul Prima Casă prin care vine în ajutorul persoanelor care doresc să își achiziționează pentru prima dată o casă. Programul a fost lansat într-un moment de criză al pieței imobiliare și avea ca obiectiv ridicarea numărului de locuințe vândute în România în acea perioadă. Guvernul garanta 80% din valoarea unui împrumut pe care beneficiarii trebuiau să îl investească în achiziția unei locuințe, dar valoarea maximă a creditului nu trebuia să depășească 60.000 Euro[4]. Sprijinul se poate acorda tuturor categoriilor de vârstă singură condiții fiind ca în trecut să nu fi avut alt credit ipotecar.

În primele 5 luni ale anului 2018 ponderea imobilelor cumpărate cu programul Prima Casa a ajuns la pragul de 60 de procente. Conform unei analize a pieței s-a rezultat faptul ca beneficiarii prefera apartamentele cu 2 camere, 6 din 10 comparatori optând pentru aceasta varianta , 3 din 10 alegând varianta cu 3 camere. Capitala României deține locul întâi la numărul de locuințe achiziționate, urmata apoi de Brașov, Iași și Cluj[5].

Evoluția prețurilor față de trimestrul anterior

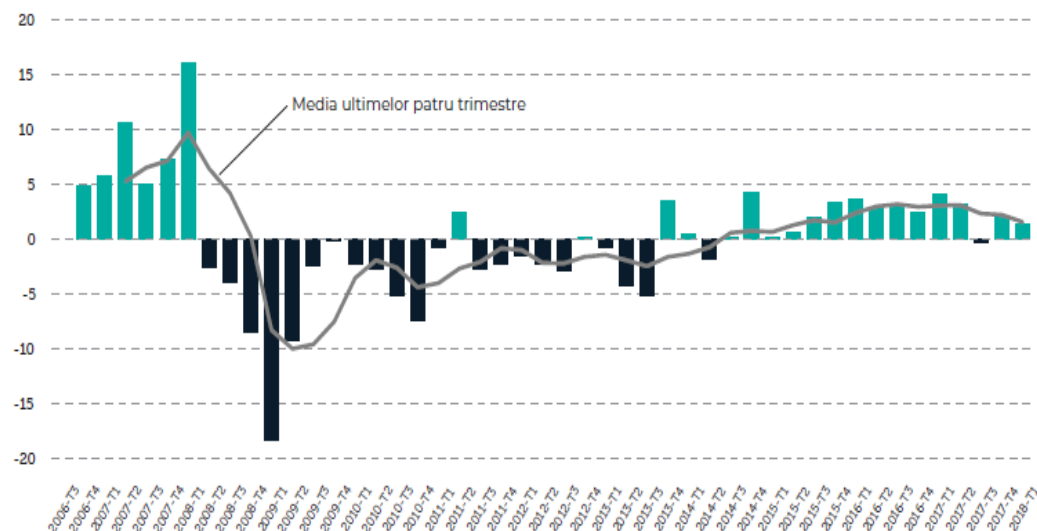


Fig 1 [6] - Evoluția preturilor pe trimestre începând cu anul 2006 pana in prezent

Conform unui studiu realizat de platforma Analize Imobiliare, la începutul anului 2018, se observă o marjă de creștere a prețurilor cu 1,4% spre deosebire de aceeași perioadă a anului trecut unde marja atingea pragul de 4%. De obicei în primul trimestru se înregistrează cea mai mare creștere a prețurilor , în următoarele trei pătrare creșterea fiind vizibil mai redusă. De remarcat este faptul că în 2018 creșterea din prima parte a anului a fost relativ mică rezultând faptul că e posibil că prețul imobiliarelor să intre într-o perioadă de stagnare sau de creștere mai lentă față de anii anteriori. În figura prezentată mai sus (fig[1]) este prezentata o analiză a evoluției prețurilor locuințelor începând cu anul 2006 și până în prezent[6].

Unul dintre cel mai bine dezvoltate orașe din România din punct de vedere al pieței imobiliare este Cluj-Napoca acesta fiind singurul oraș din țară al cărui preț pe metrul pătrat este mai mare decât cel înregistrat în perioadă boom-ului din anul 2008. Media prețurilor este mai ridicată cu 15,2%. Ceea ce este surprinzător este faptul că prețul înregistrat în București este mai mic decât cel din Cluj-Napoca cu toate că numărul de vânzări de imobile ești mult mai mare. Făcând o comparație a prețurilor din capitală înregistrate în primul trimestru din 2008 si

cel din aceeași perioadă a anului curent se observă faptul că evoluția este negativă având valoarea de -41,8% . În fig2 este prezentată o analiză a prețurilor pentru orașele cu cele mai mici și cele mai mari scăderi de preț din România făcându-se o comparație între primul trimestru din 2008 și primul trimestru din 2018.

	2018-T1	2008-T1	Evoluția		2018-T1	2008-T1	Evoluția
Cluj-Napoca	1.480	1.280	+15,2%	Piatra-Neamț	670	1.220	-45,3%
Oradea	930	940	-0,7%	București	1.260	2.160	-41,8%
Timișoara	1.150	1.220	-6,5%	Târgoviște	590	980	-39,7%
Arad	670	720	-6,5%	Reșița	470	690	-31,5%
Alba Iulia	850	940	-9,3%	Alexandria	530	780	-31,2%
Satu Mare	650	720	-10,5%	Galați	810	1.150	-30,1%

fig 2[6] – Comparație preturi între primul trimestru din 2008 și cel din 2018 pentru diferite orașe din țara

Se poate observa faptul că în general orașele din vestul țării sunt cele care au înregistrat cele mai mici scăderi de prețuri, spre deosebire de zona de sud și de est unde majoritatea nu depășesc pragul de -30%. Conform studiului efectuat de cei de la Analize Imobiliare orașul Cluj este cel mai scump atunci când vine vorba de prețul mediu pe metru pătrat, valoarea fiind de 1480 Euro. Este urmat de București al cărui preț mediu pe metrul pătrat atinge pragul de 1260 Euro.

Făcând o analiză asupra celor mai scumpe cartiere din Cluj se observa faptul că prețurile se mențin la același nivel ca și anul trecut, nefiind scumpiri majore. Zona Central/Ultracentral își păstrează titlul de cel mai scump cartier al orașului prețul pe metrul pătrat atingând suma de 1690 Euro. În figura de mai jos sunt prezentate cartierele cu cele mai mari prețuri pe metrul pătrat ale orașului situat în centrul Transilvaniei. Cartierul Zorilor este unul dintre cele mai scumpe zone ale Clujului însă în prima parte a acestui an este sigurul cartier de interes major care înregistrează o scădere de preț cu un procent de 1,3. De cealaltă parte zona Plopilor este în continuă creștere ajungând la o scumpire cu 2,1% procente la începutul lui 2018.



Sursa: Analize Imobiliare și Imobiliare.ro

fig 3[6] – Analiza preturi pe metrul pătrat in cele mai scumpe cartiere din Cluj-Napoca in primul trimestru al anului 2018

În capitala României nivelul prețurilor în cele mai scumpe cartiere este mult mai ridicat decât în Cluj-Napoca. Cea mai scumpă zonă este reprezentată de Kiseleff-Aviatorilor unde prețul pe metrul pătrat a ajuns la valoarea de 2560 Euro și a înregistrat o creștere cu 2,8% procente în primul trimestru din 2018. Al doilea cartier important este Herăstrău-Nordului unde prețul pe metrul pătrat este de aproximativ 2490 Euro. Spre deosebire de aceeași perioadă a anului trecut când scăderea atingea pragul de 4%, anul acesta zonă începe să fie din nou apreciată crescând cu 0,9% procente. Singura zonă care înregistrează o scădere este Dorobanți-Floreasca, -0,3%, dar acesta este relativ mică și nu prezintă motiv de îngrijorare pentru dezvoltatorii imobiliari.



Sursa: Analize Imobiliare și Imobiliare.ro

Fig 4[6] - Analiza preturi pe metrul pătrat in cele mai scumpe cartiere din București in primul trimestru al anului 2018

Mai devreme am menționat faptul că orașul Cluj deține locul 1 la prețul mediu pe metrul pătrat din România cu toate că, dacă ne uităm la valorile înregistrate de cartierele scumpe ale capitalei, acestea sunt cu aproape 1000 Euro mai scumpe. În ultimii ani dezvoltatorii imobiliari din București și-au îndreptat atenția spre zonele din periferia orașului investind în locații ca de exemplu Militari, Titan sau Berceni unde au construit o mulțime de cartiere rezidențiale, acestea fiind din ce în ce mai căutate de către potențialii cumpărători. Chiar dacă clădirile sunt noi, prețul locuitorilor nu poate fi ridicat datorită zonei periferice și sunt vândute la prețuri din ce în ce mai accesibile. Acesta este și motivul pentru care prețul mediu din București a scăzut în ultimii ani, în ciuda faptului că vinde cele mai multe imobile pe parcursul unui an.

Cu toate acestea prețul mediu pe metrul pătrat, în București, se află în creștere iar interesul pentru achiziția de apartamente devine din ce în ce mai mare. Acest lucru se datorează și faptului că multe companii internaționale și-au deschis puncte de lucru în capitală atrăgând o mulțime de tineri, care după terminarea studiilor preferă să nu se mai întoarcă la locurile natale și își creează un viitor aici. Astfel crește cererea pe piață imobiliară, sunt construite tot mai multe apartamente iar prețul înregistrează la rândul lui creșteri semnificative. În figura de mai jos se poate observa evoluția prețului general în București între anii 2008-2018.



Fig 5[7] – Evoluția evoluția prețului general in București între anii 2008-2018

Anul 2008 reprezintă anul de cumpănă al pieței imobiliare după cum se observa și în graficul de mai sus. De remarcat este faptul că scăderea a continuat până anul 2014 și de abia de atunci creșterile au început să apară. În momentul de față este destul de vizibilă panta ascendentă pe care a luat-o prețul general.

În urma analizei pieței imobiliare efectuate mai sus, sunt de părere că nevoia de platforme de promovare pentru anunțuri din acest domeniu este din ce în ce mai mare, cererea apariției lor crescând o dată cu dezvoltarea acestui domeniu. Site-urile de acest gen urmăresc promovarea ofertei imobiliare și distribuirea informației unei mase cât mai mare de oameni aflați în căutarea unei locuințe.

3 STUDIU DE PIAȚĂ

3.1 Analiza platformelor existente pentru anunțuri imobiliare

Piața imobiliară din România a cunoscut o creștere semnificativă în ultimii ani și acest lucru a dus la apariția multor platforme de anunțuri prin intermediul cărora interacțiunea dintre vânzător și cumpărător se realizează foarte ușor, iar drumul către tranzacția finală este mai aproape. Cea mai mare platformă online dedicată vânzătorilor dar și cumpărătorilor de imobiliare din țară este imobiliare.ro[8]. Portalul a fost lansat în anul 2000 și de atunci este într-o continuă expansiune ajungând în momentul de față să depășească 1,5 milioane de vizitatori unici pe luna, conform site-ului de specialitate trafic.ro[9]. Pe site își pot publica anunțurile atât agențiile imobiliare, dar și persoanele particulare. O dată cu creșterea în popularitate au început să vină și contractele iar în ziua de astăzi cei de la imobiliare.ro au peste 1000 de agenții care își promovează anunțurile pe site-ul lor[8].



fig. 6[8] – Interfața aplicației web imobiliare.ro

Platforma este foarte complexă oferind posibilitatea de a filtra anunțurile după o mulțime de criterii începând de la numărul de camere și compartimentare ,până la suprafața utilă și multe altele. Ceea ce este interesant este faptul că pe site se găsesc analize imobiliare , statistici și știri legate de acest domeniu. Dezvoltatorii de complexe imobiliare au o secțiune dedicată prin care pot să își prezinte întreg ansamblul fără a fi nevoiți să își administreze o pagină personală de promovare.

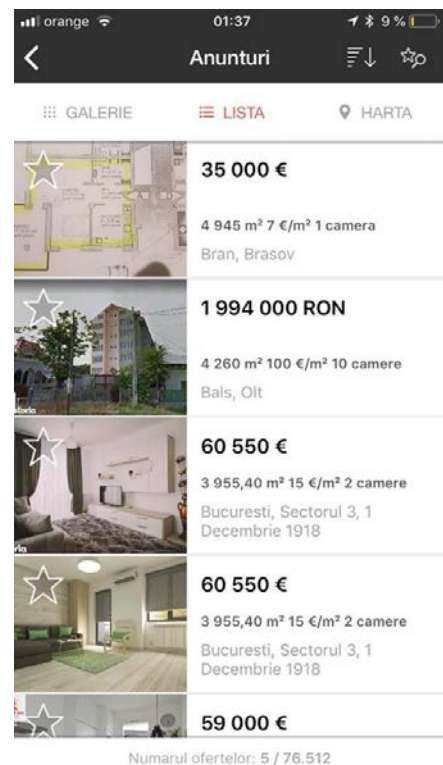
În fig. 6 este afișată pagină de start a aplicației web. În antetul paginii se observă categoriile în care se încadrează fiecare anunț vânzare/ închiriere dar și paginile speciale dedicate ansamblurilor imobiliare sau agențiilor. Secțiunea imoExpert este pagina de știri unde se pot găsi analizele financiare, știri dar și sfaturi utile atunci când urmărești achiziționarea sau închirierea unei locuite.

După cum se poate vedea în imaginea de mai sus prima căutare a anunțurilor pe imobiliare.ro se face pe baza a trei criterii, locație, categorie imobil și operațiune iar de abia după această prima filtrare se pot adăuga și alte criterii că de exemplu număr de camere sau structura locuința(semidecomandat, comandat). Acest lucru poate fi considerat un dezavantaj al platformei prezentate fiind nevoie de două filtrări pentru a aduce în interfață lista de anunțuri favorabile utilizatorului.

O altă platformă importantă în publicarea de anunțuri imobiliare este storia[10]. Face parte din grupul OLX și reprezintă soluția propusă de dezvoltatori pentru publicarea de anunțuri imobiliare. Spre deosebire de soluția propusă de OLX Imobiliare, storia aduce în plus informații suplimentare față de cele din pagină de olx, procesul de căutare devine mai exact astfel încât utilizatorului îi sunt aduse doar anunțurile relevante pentru el[11]. Toate anunțurile publicate pe storia sunt publicate în mod automat și pe olx. Astfel posibilitatea de promovare este mai mare și pot fi văzute de un număr cât mai mare de clienți. Atât olx cât și storia au aplicații web pentru cei care preferă căutarea din față unui laptop, dar și aplicații mobile pentru persoanele care folosesc telefonul pentru astfel de activități. În momentul de față storia este portalul de deschidere pentru aproximativ 75.000 de anunțuri.



Fig[7] – Interfața mobilă Olx



Fig[8] – Interfața mobilă storia

În imaginile de mai sus sunt prezentate cele două interfețe mobile ale aplicațiilor olx și storia. Cea din stânga este olx Imobiliare iar cea din dreapta este platforma storia. Se poate observa că primele anunțuri din pagina de olx sunt marcate cu semnul PROMOVAT. Olx le oferă clienților săi posibilitatea de a le promova anunțul pe site în schimbul unei sume de bani. Astfel anunțurile apar pe pagină indiferent de filtrele pe care le adaugă utilizatorul oferindu-le posibilitatea de a fi văzute o masă mare de oameni care folosesc platforma de căutare. Pentru vânzătorii privați olx este gratuit în limită a 3 anunțuri la fiecare 30 de zile, iar pentru agenții sau promovare de ansambluri rezidențiale se oferă pachete promoționale în funcție de preferințele fiecăruia. Olx cere în schimbul unui anunț sumă de 0,95 Euro iar Storia 3,99 Euro. Dacă se optează pentru ambele platforme , prețul de pornire începe de la 0.30 Euro pe anunț[11].

Pagina de start a celor de la storia.ro este aproximativ asemănătoare cu cea a celor de la imobiliare.ro abordând același stil de căutare a anunțurilor cu o prima filtrare după 3 categorii de bază și abia apoi adăugarea și altor criterii pe care se bazează utilizatorul. Ca și avantaje majore pe care le au ambele platforme prezentate mai sus este faptul că ambele dețin versiuni pentru desktop dar și pentru mobil. Astfel aceștia se adresează unei game largi de utilizatori în funcție modalitatea prin care preferă aceștia să caute locuința dorită.



Fig 7[10] – Complex rezidențial promovat pe storia.ro

În figura 7 se prezintă promovarea unui ansamblu rezidențial pe storia.ro. În primul rând se observă imaginea de ansamblu a întregului complex, aflat în construcție la data publicării anunțului. Se oferă informații referitoare la zona de amplasare, posibilitatea de a vedea pe harta locația, prețul pe metrul pătrat și existența unui formular de completare cu câteva date personale prin care se poate lua legătura direct cu dezvoltatorul pentru mai multe informații.

3.2 Tehnologii folosite in dezvoltarea aplicației

Primul pas care trebuie făcut înainte de a începe dezvoltarea aplicației este alegerea tehnologiilor necesare pentru crearea acesteia. Aplicația trebuie să fie cât mai responsive având în vedere că gestionează un număr mare de date care trebuie filtrate în cel mai scurt timp posibil. Ca orice aplicație mobilă, dezvoltarea se bazează pe implementarea a două niveluri, front-end și back-end. Partea de front-end a fost realizată folosind ionic framework ,dezvoltat pe Angular dar care folosește și câteva tehnologii web cum ar fii HTML și CSS. Back-endul a fost realizat folosind Java , Maven și framework-ul Spring. Ca și bază de date am folosit PostgreSQL iar pentru a putea sincroniza câmpurile din tabela cu obiectele din back-end am folosit Hibernate.

3.2.1 Ionic framework

Interfața aplicației am creat-o folosind ionic, un framework destul de recent apărut, creat în anul 2013 și folosit pentru dezvoltarea de aplicații mobile[12]. Framework-ul este construit pe bază limbajului Angular și folosește pentru implementare cele mai cunoscute tehnologii web precum CSS și HTML5. Folosind Angular, ionic oferă componente și metode personalizate pentru a putea interacționa cu ele. O astfel de componentă este „ion-content” în interiorul căreia se implementează metode utile pentru zona de scrolling[13].

```
<ion-content>
  Add your content here!
</ion-content>
```

Fig 8 [13] – Exemplu de bloc content

3.2.2 HTML si CSS

HTML (Hypertext Markup Language) este cel mai utilizat limbaj de marcare pentru dezvoltarea unei pagini web. Fișierele de tip HTML pot fi recunoscute după una din extensiile .htm sau .html. Un HTML este compus din elemente iar fiecare bloc de elemente se marchează folosind etichetele.[14]. Aceste etichete sunt marcate prin parantezele unghiulare precum: <etichete>.. </eticheta. In figura de mai jos este prezentata structura de baza a unei pagini HTML.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Fig 9[14] –Structura de baza a unei pagini HTML

Prima linie din codul prezentat , <!DOCTYPE html>, reprezintă versiunea HTML a documentului, in cazul de fata fiind vorba despre HTML 5. A doua linie, <html>, deschide un bloc pe care îl încheie in ultima linie, </html> , are rolul de a descrie un document HTML încadrând întreaga pagina web. In blocul descris de <head> si </head> este prezentat antetul paginii web si sunt introduse diferite informații legate de aceasta. In interiorul zonei de head se afla blocul <title> </title> care reprezintă titlul paginii afișat in bara de titlu a browserului. Corpul paginii este marcat de tag-urile <body> si </body> intre care este încadrat conținutul paginii web. Intre <h1> si </h1> se introduce titlul afișat in corpul paginii iar tag-urile <p> </p> este reprezentat un paragraf[14].

CSS (Cascade Style Sheets) este un standard de formatare folosit pentru a descrie felul in care sunt afișate elementele HTML in cadrul paginii web. Există două modalități prin care elementele pot si modificate prin CSS. Una este folosirea unui fișier separat iar cealaltă este folosirea etichetei <style> </style> direct in documentul HTML[15]. In figura de mai jos avem o prezentare a unui cod CSS prin intermediul căruia se modifica modul de afișare al unei pagini web. Ceea ce se afla intre { } reprezintă o „regula” aplicata pe o anumita zona a paginii.

```
body {  
    background-color: lightblue;  
}  
  
h1 {  
    color: white;  
    text-align: center;  
}  
  
p {  
    font-family: verdana;  
    font-size: 20px;  
}
```

Fig 10[15] - Set reguli CSS

In figurile de mai jos se poate observa modificările pe care le poate aduce CSS unui document HTML. In fig 11[15] este o pagina scrisa doar in HTML iar in fig 12 este aceeași pagina scrisa in HTML peste care s-au adăugat elemente de design de tip CSS.

My First CSS Example

This is a paragraph.

Fig 11[15] –HTML

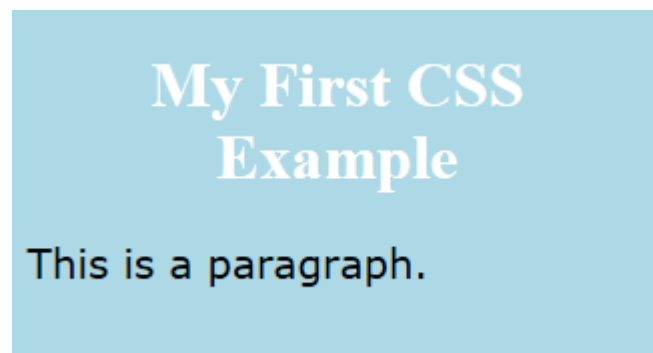


Fig 12[15] – HTML + CSS

3.2.3 Angular si TypeScript

Angular este o platformă de dezvoltare pentru construirea de aplicații web și mobile utilizând TypeScript dar și alte limbaje de programare. De fapt acest TypeScript este o îmbunătățire a limbajul JavaScript ceea ce înseamnă că orice cod scris în JavaScript este valabil și în TypeScript . Totuși TypeScript a început să aibă prioritate în rândul programatorilor deoarece o data cu apariția acestui limbaj au apărut și câteva caracteristici speciale. Pentru ca browserele să poată înțelege codul scris în TypeScript trebuie să îl transforme în JavaScript lucru de care se ocupă compilatorul de tip TypeScript[16].

3.2.4 Java

Java este cel mai răspândit limbaj de programare din lume cu ajutorul căruia au fost dezvoltate cele mai multe aplicații distribuite. Conceptul pe care se bazează limbajul este programarea orientată pe obiecte[17]. Clasă este unul dintre principiile de bază ale programării orientate pe obiect. Cu ajutorul acesteia programatorul poate descrie caracteristici și comportamente ale unui obiect de un anumit tip. Astfel obiectul devine principalul instrument de lucru în Java. În figura de mai jos voi prezenta cum arată o clasă în Java.

```
1  class Carte{
2
3      float pret;
4      String titlu;
5      String autor;
6
7      Carte(int a, String b, String c){
8          pret = a;
9          titlu = b;
10         autor = c;
11     }
12
13     public float getAutor(){
14         return autor;
15     }
16     public String afiseaza(){
17         return "Cartea "+titlu+" are ca autor "+autor;
18     }
19 }
```

Fig 13 – Exemplu clasa Java

Mai sus este descrisă clasa Carte care are 3 caracteristici(preț, titlu, autor) și 2 comportamente (getAutor() și afișează()). Un obiect este un tip de dată care încapsulează toate aceste 5 atribute. Obiectul reprezintă instantierea clasei. Pentru a instanția un obiect de tip Carte se folosește operatorul new : carte1 = new Carte();. Instantierea se realizează cu ajutorul constructorului care are rolul de a inițializa obiectele în momentul în care este apelat. Constructorii trebuie să aibă același nume cu cel al clasei din care face parte. Pot exista mai mulți constructori într-o clasă cu condiția ca semnătura acestora să fie diferită.

Exemplu:

```
class Complex {  
    double real;  
    double imag;  
    public Complex(double x, double y)  
        {real = x; imag = y;}  
    public Complex ( )  
        {real = 0; imag = 0;}  
}
```

Exemple de instanțiere:

```
Complex z= new Complex();      // z = 0 + 0*i  
Complex z1= new Complex(1,0); // z1 = 1 + 0*i
```

Fig 14[18] – Exemplu de clasa cu 2 constructori

Un alt principiu important pe care se bazează Java este Moștenirea. O clasă are proprietatea de a putea fi moștenită de către altă clasă. Astfel clasă moștenitoare preia toate caracteristicile primei clase dar poate să adauge altele noi sau să le modifice pe cele preluate.

3.2.5 Spring Framework

Spring Framework se folosește pentru a simplifica aplicațiile scrise în Java. Frameworkul se remarcă prin folosirea Dependency Injection. Așa cum am spus și mai sus, aplicațiile care se bazează pe Java se folosesc de obiecte care trebuie să lucreze cât mai eficient împreună pentru a oferi utilizatorului un răspuns rapid. Într-o aplicație Java cât mai complexă clasele aplicației trebuie să fie cât mai independente față de restul claselor Java utilizate pentru a le putea testa separat. Dependency Injection ajută la comunicarea dintre acestea reușind să le păstreze în același timp independente[19]. Procesul de injecție se realizează prin folosirea de Bean-uri , obiecte injectate în codul aplicației .

```
1 public class FooService {  
2  
3     private FooFormatter fooFormatter;  
4  
5     @Autowired  
6     public void setFooFormatter(FooFormatter fooFormatter) {  
7         this.fooFormatter = fooFormatter;  
8     }  
9 }
```

Fig 15[20] – Exemplu Dependency Injection

3.2.6 Maven

Maven este un sistem prin intermediul căruia se face procesul de build al aplicațiilor scrise în Java. Prin intermediul acestuia se descrie procesul de build, se descriu dependențele aplicației, dar mai are și alte proprietăți. Toate aceste funcționalități sunt scrise într-un fișier .xml[21]. În Maven acest fișier poartă denumirea de „pom”. În pom.xml se află toate detaliile legate de cum se face build , în ce ordine se realizează acest lucru , toate dependențele dar și biblioteci și plug-in-uri necesare.

3.2.7 Hibernate

Hibernate este un ORM(Object Relational Mapping) folosit de Java prin intermediul căruia are loc maparea claselor java pe tabelele din baza de date folosita. Maparea claselor Java se face folosind fișierul de configurație XML sau folosind adnotări. Prin intermediul fișierului XML se păstrează schema bazei de date si se fac legăturile necesare in clase si tabelele din baza.

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping>
  <class name = "Employee" table = "EMPLOYEE">

    <meta attribute = "class-description">
      This class contains the employee detail.
    </meta>

    <id name = "id" type = "int" column = "id">
      <generator class="native"/>
    </id>

    <property name = "firstName" column = "first_name" type = "string"/>
    <property name = "lastName" column = "last_name" type = "string"/>
    <property name = "salary" column = "salary" type = "int"/>

  </class>
</hibernate-mapping>
```

Fig 15[22] - Exemplu de fișier folosit pentru mapare

<hibernate-mapping> este tag-ul prin care se specifica faptul ca este vorba de un fișier de mapare. In interiorul tag-ului <class> se definește clasa din java si tabela pe care este mapata. Maparea unui element din clasa java si o coloana din tabela bazei de date se face prin tag-ul <property>. Atributul „name” face referire la proprietatea din clasa java si „column” asociază aceasta proprietate cu o coloana din baza de date. Conversia din Java in SQL se face folosind atributul <type>.

3.2.8 POSTGRESQL

PostgreSQL este un sistem de gestiune al bazelor de date, open-source, bazat pe limbajul SQL. Serviciul folosește pgAdmin ca si interfața de lucru , modul de funcționare al acestuia fiind ca si al unui server, conexiunea realizându-se printr-un port stabilit de utilizator.

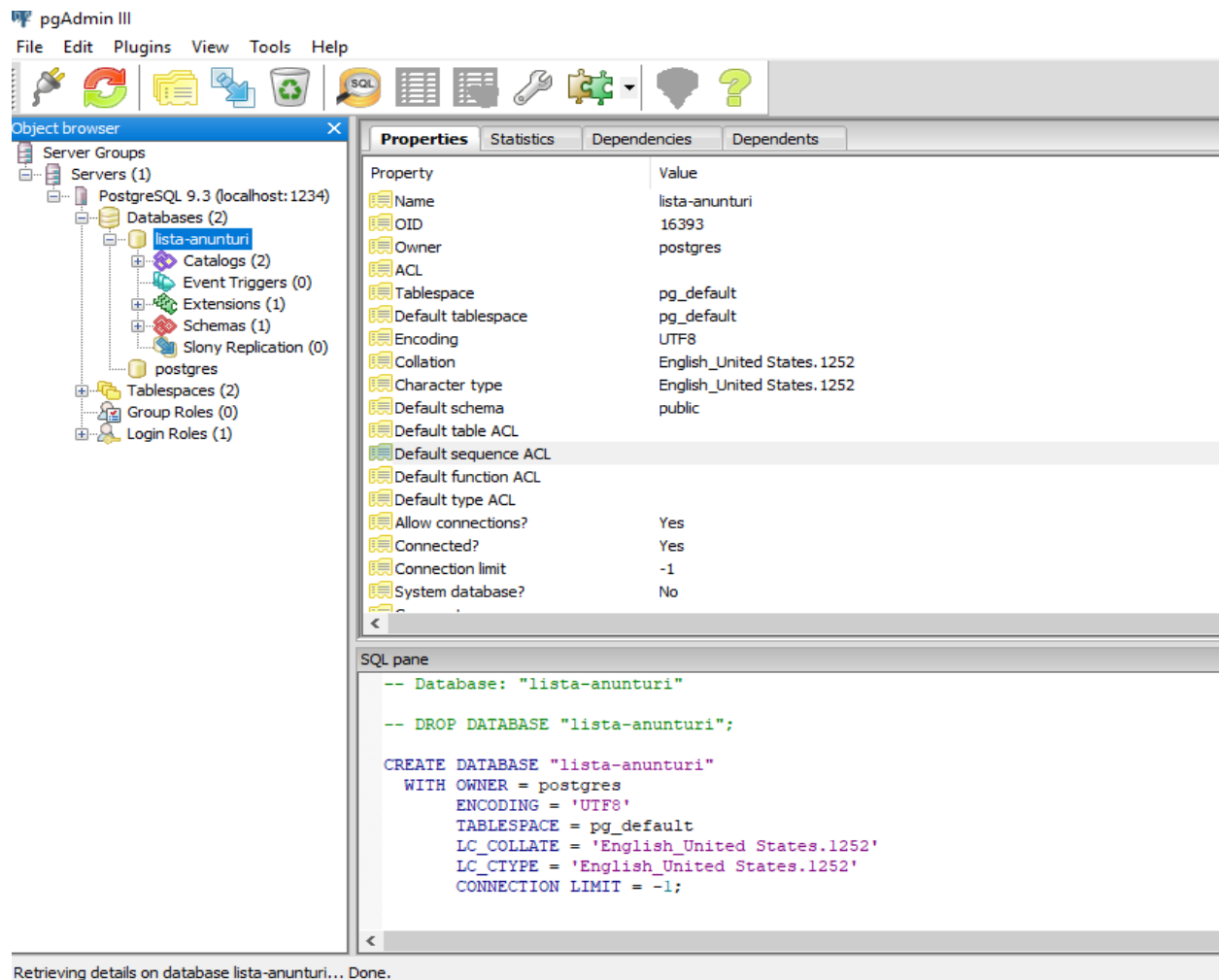


Fig 16 – Interfața pg Admin pentru PostgreSQL

În figura de mai sus se observă faptul că serverul bazei de date funcționează pe portul 1234. Baza de date poate fi creată direct din interfața folosind SQL pane , de altfel acolo se pot scrie și cereri de tip SQL sau orice alte funcționalități ale limbajului.

4 SOLUȚIA PROPUȘĂ

Soluția propusă este dezvoltarea unei aplicații mobile prin intermediul căreia utilizatorii pot căuta un imobil în funcție de nevoile fiecăruia sau pot adăuga anunțuri prin care să scoată locuințe spre vânzare sau spre închiriere.

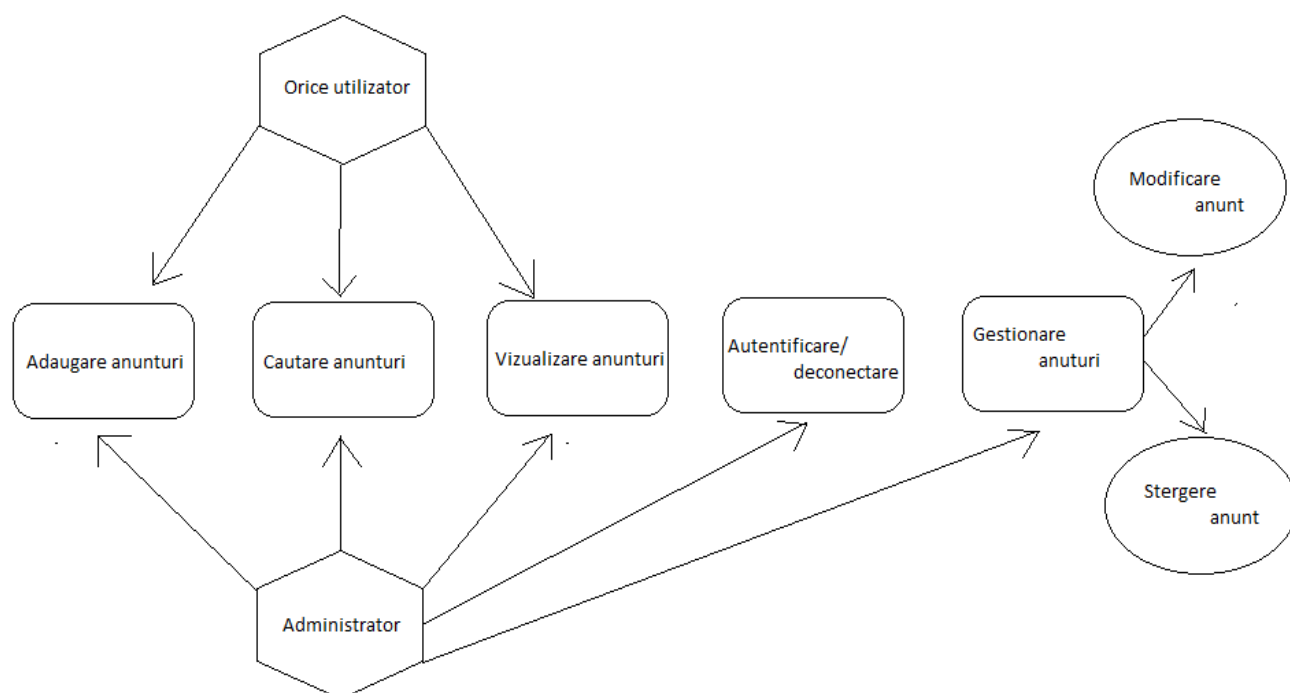


Fig 17 - Cazurile de utilizare ale aplicației propuse

În figura de mai sus este prezentată diagrama cazurilor de utilizare ale aplicației în funcție de persoana care o folosește: orice utilizator sau administratorul. Se observă faptul că doar utilizatorul care deține contul de administrator are dreptul de a modifica sau de a șterge un anunț. Restul pot efectua doar operațiile de bază precum adăugare anunțuri, căutare și vizualizare. În continuare voi prezenta interfața aplicației și voi descrie funcționalitățile pentru fiecare pagină în parte.



Fig 18 – Pagina de start a aplicației.

Pe prima pagină sunt expuse opțiunile pe care le are un utilizator atunci când deschide aplicația. De menționat este faptul că cele 3 butoane din partea de jos nu dispar indiferent de pagina pe care te afli acestea reprezentând scurtături pentru paginile sursă. În centrul paginii este logo-ul site-ului, iar în partea de sus sunt butoanele pentru căutare anunț (Filtre) și pentru pagina de contact a administratorului (Contact Admin).

Adăugarea anunțurilor se face foarte ușor prin apăsarea butonului „Adaugare anunt” și completarea câmpurilor din pagina respectivă. Vânzătorul trebuie să ofere informații despre categoria din care face parte imobilul (Apartament, Garsonieră sau Casă), operațiunea pe care pe care dorește să o facă și aici este vorba despre vânzare sau închiriere dar și locul unde se află poziționat acesta. Acest lucru se realizează foarte ușor prin selectarea categoriilor deja completate în aplicație corespunzătoare fiecărui filtru, așa cum se prezintă în imaginea de mai jos.

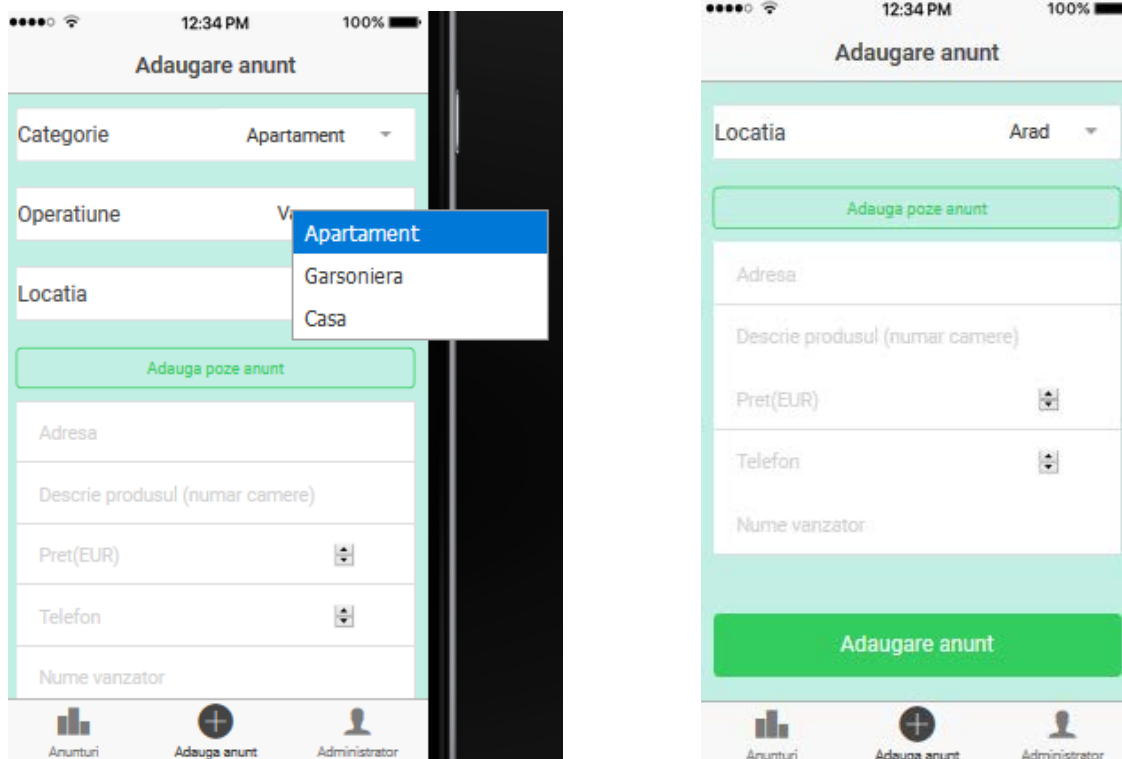


Fig 19 – Pagina de adăugare anunțuri

După completarea primelor 3 select-uri urmează alte 6 câmpuri de completat pentru a putea adăuga anunțul. În primul rând anunțul trebuie să aibă niște poze de prezentare pentru că potențialii comparatori să aibă o prima imagine a imobilului. Adresă completă trebuie adăugată de vânzător deoarece locația trimite referință doar către orașul în care se află imobilul. În câmpul „Adresa” se completează strada, numărul, blocul și numărul apartamentului. Numărul de camere, tipul (decomandat, semidecomandat) și alte informații pe care utilizatorul dorește să le ofere se completează în „Descrierea produsului”. După o prezentare de ansamblu a locuinței se adăugă prețul urmat de numărul de telefon și numele vânzătorului. După ce toate câmpurile formularului au fost completate se apasă butonul „Adauga anunt” iar vânzătorul va fi retrimis automat în pagina de start a aplicației.

Procesul este unul simplu, ușor de înțeles și intuitiv astfel încât adăugarea unui anunț imobiliar poate dura numai câteva minute iar faptul că acest lucru se poate realiza cu ajutorul unui telefon mobil conectat la internet este un avantaj major.

Principiul căutării unui anunț este la fel de simplu ca și adăugarea acestuia. Utilizatorul completează formularul cu propriile criterii de căutare. După ce toate câmpurile au fost completate se apasă butonul „Vezi anunțuri” și astfel sunt aduse din baza de date toate anunțurile care corespund criteriilor introduse. Se va deschide o nouă pagină în care este returnată lista de imobile corespunzătoare cerințelor utilizatorului. Câmpurile care trebuie completate în acest formular sunt în număr mai mic față de cele din formularul de adăugare. Există patru categorii mari după care se face căutarea. Primul este categoria din care face anunțul. Dacă utilizatorul își dorește un apartament filtrarea din baza de date se va face după „Apartament”. Următorul câmp este operațiunea , vânzare sau închiriere, urmată de locație și apoi de preț. Locația este dată de orașul în care se află apartamentul iar prețul este format din două valori care reprezintă limitele între care se face căutarea. Dacă se dorește închirierea unui apartament în orașul București cu prețul între 200 și 250 Euro formularul se va completa că în figură de mai jos.

The image shows a mobile application interface for filtering search results. The top status bar displays signal strength, time (12:34 PM), and battery level (100%). The app header has a back arrow and the text 'Anunturi Filtre'. The filter form is divided into four sections, each with a title and a dropdown menu:

- CATEGORIA**: Dropdown menu with 'Apartament' selected.
- Operatiune**: Dropdown menu with 'Inchiriere' selected.
- LOCATIA**: Dropdown menu with 'Bucuresti' selected.
- PRET(EUR)**: Two input fields with values '200' and '250'.

The bottom navigation bar contains three icons: a bar chart for 'Anunturi', a plus sign for 'Adauga anunt', and a person icon for 'Administrator'.

Fig 20 - Exemplu completare formular de căutare după criteriile dorite de utilizator

În figura 12 se poate observa lista de anunțuri generată în urma căutării după filtrele prezentate mai sus. Au fost găsite în baza de date trei anunțuri care respectă cerințele formularului de căutare. Acum clientul are posibilitatea de a alege care dintre acestea trei pe cel care se potrivește cel mai bine cu nevoile sale. Se observă faptul ca sunt vizibile doar câteva informații legate despre fiecare anunț cum ar fii prețul , locația și o parte a descrierii. Pentru a afla mai multe detalii despre anunț utilizatorul trebuie să apese pe el și va fi trimis către pagina de prezentare a acestuia. De exemplu să presupunem că dorește să afle și restul detaliilor despre primul imobil care apare în lista de căutări. Se face “click” pe el și va fi trimis pe o altă pagină unde va afla și restul detaliilor așa cum se vede în figura 13.

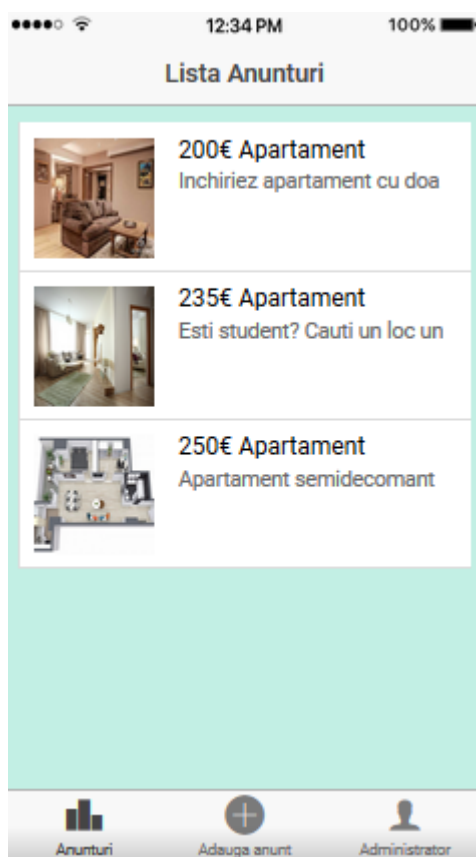


Fig 21 - Lista anuturi dupa cautare

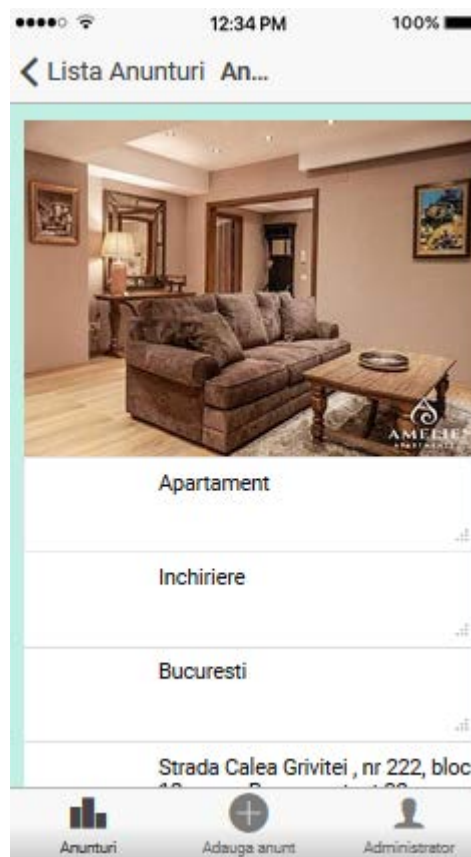


Fig 22 – Pagina vizualizare anunt

După cum am menționat si mai sus administratorul este sigurul care are posibilitatea de a edita sau de a șterge anunțuri. Daca un utilizator dorește sa facă aceste doua operații trebuie sa ia legătura cu adminul pentru ai cere acestuia sa facă un update la anunțul sau. Pe platforma

exista o pagina accesibila tuturor cu informații legate de datele de contact ale detinatorului site-ului. Acesta oferă numele si numărul de telefon la care poate fi contactat dar si o adresa de email unde utilizatorii ii pot trimite mesaje.

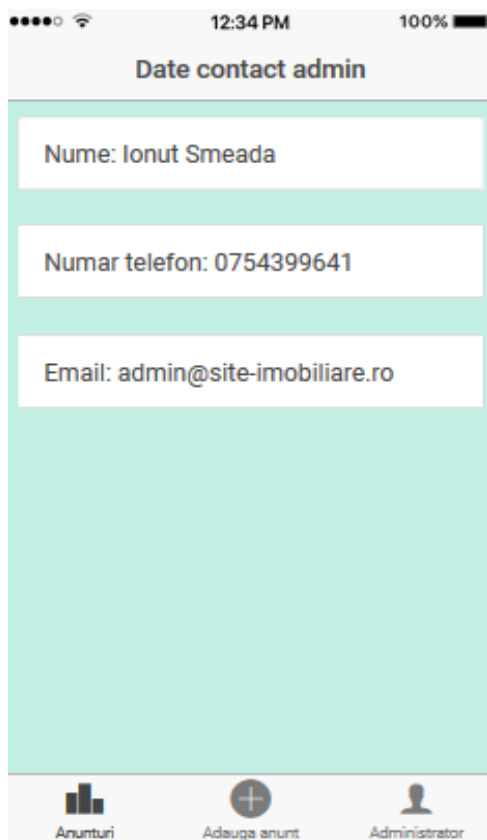


Fig 23 – Pagina date contact administrator

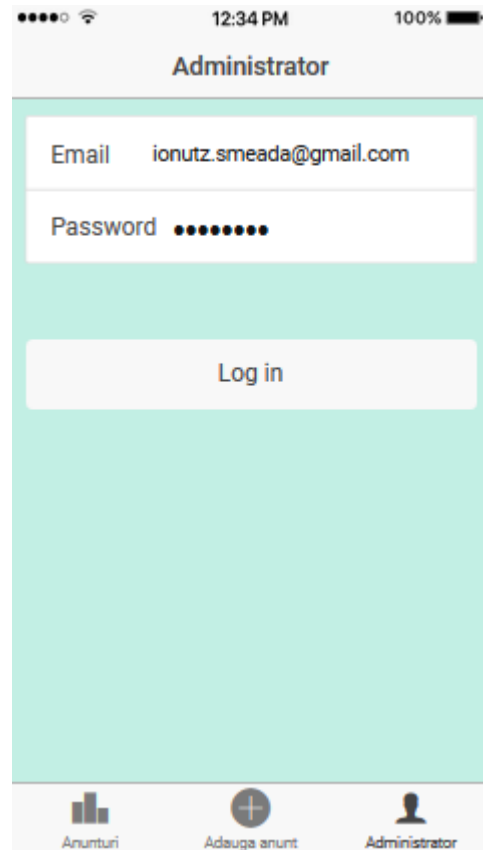


Fig 24 – Pagina de autentificare administrator

Administratorul este singurul utilizator care are nevoie de un cont pentru a se loga pe platformă. Acest lucru este necesar pentru situațiile în care dorește să facă anumite operații asupra anunțurilor. Dacă procesul de logare nu are loc atunci el nu poate șterge sau modifica fiind un utilizator obișnuit.

După ce are loc autentificarea administratorului se deschide o nouă pagină de opțiuni pe care le are acesta. Poate să vadă toată lista de anunțuri disponibilă pe site sau poate să filtreze și el rezultatele după anumite criterii pentru a găsi anunțul căutat.

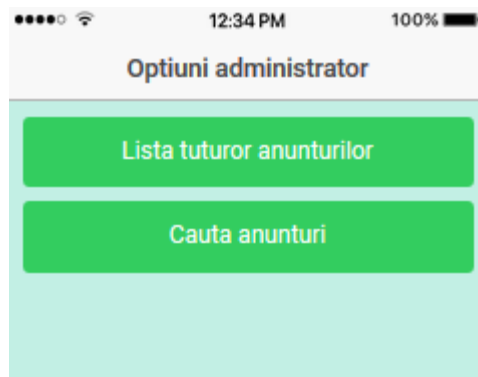


Fig 25 – Pagina opțiuni disponibile pentru administrator

Un utilizator postează un anunț despre închirierea unei garsoniere. În urma promovării pe site reușește să își găsească un client cu care stabilește un contract pe o anumită perioadă. Pentru a nu fi deranjat de către alți posibili clienți care găsesc anunțul pe site , după ce el a închiriat imobilul, utilizatorul trebuie să ia legătura cu administratorul și să îi ceară acestuia ștergerea anunțului adăugat de el. Administratorul are două posibilități prin care poate face acest lucru. Prima este să caute în anunțul în pagina cu toate anunțurile dar acest lucru nu este foarte eficient și cere prea mult timp iar a doua este să filtreze după detaliile primite de la utilizatorul care a postat pentru al găsi într-un mod cât mai rapid. După ce îl găsește îl șterge de pe platformă.

5 DETALII DE IMPLEMENTARE

Aplicația implementată de mine este împărțită în două mari niveluri a căror comunicare se realizează prin HTTP. Hypertext transfer protocol(HTTP) este metodă prin care se pot aduce în front-end informațiile stocate pe un server. Aceste informații sau date pot fi trimise către destinație sub forma unui fișier HTML , un fișier grafic sau video chiar și un fișier de sunet. Datele sunt cerute prin intermediul unui request făcut de destinație către server de la care primește un response.

În figura de mai jos se prezintă comunicarea dintre client și server prin schimbul de mesaje request/response. Se observă faptul că se trimite un HTTP request către server de la care se primește răspuns printr-un HTTP response.



Fig 26[23] – Comunicarea client-server realizata prin http

Pentru ca serverul să știe lucrurile pe care clientul le cere se folosesc adresele URL. Practic fiecare adresă URL se identifică cu o resursă. De exemplu în aplicația creată de mine, dacă utilizatorul dorește să caute un anunț va completa formularul de căutare anunțuri din pagina de front-end, serverul va primi URL-ul compus din fiecare resursă pe care clientul a completat-o și va returna o lista de JSON-uri așa cum este prezentat în figura de mai jos.

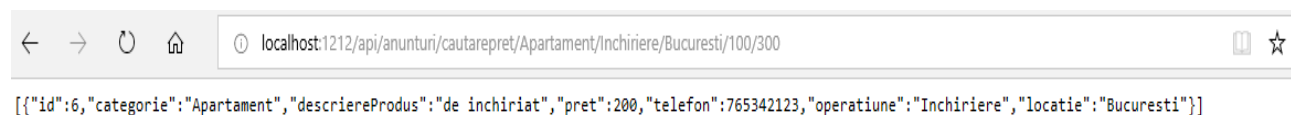


Fig 27 – Response HTTP către client în funcție de cererea acestuia

Fiecare request trimis de client conține un header și un body. În interiorul header-ului se află metoda iar body-ul poate conține informații adiționale. Atunci când clientul trimite o cerere de căutare din front-end, aceasta este transformată în JSON pentru ca mai apoi să fie trimisă în acest format către server de unde va primi un respons cu rezultatele dorite , tot în format JSON și transformat mai apoi în front-end. Cum am spus mai sus, în header-ul requestului făcut de client se află o metodă HTTP. În cazul prezentat este vorba de metoda GET.

. Acesta este cel mai simplu tip de solicitare a protocolului. Este apelată de fiecare data când se face click pe un link sau este scris un URL în bara de adrese. Această metoda este doar o solicitare a clientului trimisă către server realizată doar pentru citire, nu presupune nicio modificare pe partea de server[24].

Pentru a adăuga un anunț în baza de date se realizează același traseu ca și cel prezentat mai sus. Request trimis de către client și response primit de la server. De această dată o să se schimbe metoda din header iar body-ul va conține informațiile pe care le are un anunț. Metoda HTTP folosită în acest caz este POST. Ea este implementată în așa fel încât să îi solicite serverului să accepte datele pe care le conține body-ul. Atunci când se încarcă un anunț nou în baza de date, în body se găsesc toate câmpurile completate de utilizator în pagina de adăugare anunț. În această situație prin POST se trimite formularul către server și apoi este încărcat în baza de date.

Pentru a face update la un anunț aflat în deja pe site, requestul se face cu metoda PUT și cu url-ul anunțului respectiv. Anunțul deja se afla în zona front-end astfel încât clientul face modificările dorite. Apoi cu metoda PUT trimite prin HTTP anunțul deja modificat către server și este înlocuit în baza de date. Pentru a-l șterge se folosește DELETE care este inversul metodei PUT. Anunțul identificat prin URL nu se updatează ci se va șterge definitiv de pe site.

Toate procesele prezentate mai sus se bazează pe folosirea arhitecturii REST. Prin intermediul acestui serviciu se definesc metode bazate pe operațiile HTTP prezentate mai sus. Atunci când se creează un server bazat pe serviciul REST, cererile HTTP sunt gestionate de către un controler. Controller-ul este identificat prin adnotarea `@RestController` ca în figura de mai jos.

```
1 @RestController
2 @RequestMapping("/anunturi")
3 public class AnunturiController {
```

Fig 28 – Definire Controller

După adnotarea specifică controlerului se afla adnotarea `@RequestMapping` prin intermediul căreia se asigură ca cererile pe `/anunturi` sunt mapate pe clasa `AnunturiController`.

În interiorul controlerului este injectat o un obiect de tip service anunțuri așa cum este prezentat mai jos. În interiorul clase AnunturiService se retine o entitate de repository.

```
@Autowired
public AnunturiController(AnunturiService anunturiService) {
    this.anunturiService = anunturiService;
}
```

Fig 29 – Injectare AnunturiService

În momentul în care se efectuează o adăugare de anunț, se primește prin protocolul HTTP un request în al cărui header se află metoda POST și un body în interiorul căruia este un JSON de tipul AnunturiDTO în care este descris anunțul în funcție de coloanele din baza de date. În service entitatea DTO este transformată într-o entitate de tip Anunțuri. Abia după tot acest proces are loc adăugarea în baza de date. Întreg procesul descris mai sus are loc pentru orice operație cerută de client și reprezintă partea de back-end a aplicației.

Pentru partea am front-end am folosit IONIC, un framework a cărui funcționalitate se bazează pe scrierea codului cu ajutorul HTML, CSS și TypeScript. În figura de mai jos este prezentată pagina de start a aplicației

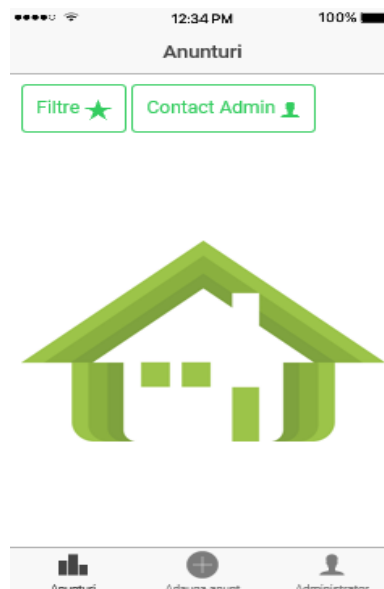


Fig 30 – Pagina de start

Dacă un utilizator dorește să caute un anunț, acesta va apăsa pe butonul „Filtre” și va fi redirecționat către pagina de completare a formularului de căutare. Designul butonului este făcut cu ajutorul HTML-ului + CSS.

```
<button id="anunturi-button1" ion-button outline color="balanced" icon-right on-click="goToFiltre()">
  Filtre
  <ion-icon name="star"></ion-icon>
```

Fig 31 – Exemplu creare buton „Filtre” HTML + CSS

În figura de mai sus este prezentat modul în care am creat butonul. Acesta conține un id, câteva elemente de design cum ar fi „color” sau „icon”. Se observă faptul că în momentul în care se face click pe el, se apelează funcția „goToFiltre()”. Acesta este scrisă în pagina .ts a paginii de start. Fiecare pagină este compusă din fișiere cu același nume dar cu extensii diferite în funcție de funcționalitatea pe care o are fiecare, .html, .scss, .ts.

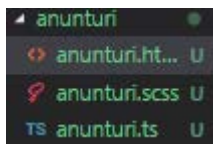


Fig 32 – Structura unei pagini.

După cum am spus mai sus, se apelează funcția „goToFiltre()” din fișierul .ts al paginii. Cu ajutorul acestei funcții se deschide o nouă pagină „Filtre” în care se utilizează formularul de căutare. Această funcționalitate este aplicată pentru toate butoanele aplicației, din HTML se apelează funcția scrisă în TypeScript care la rândul său apelează metoda de care are nevoie pentru a funcționa corespunzător.

```
goToFiltre(params){
  if (!params) params = {};
  this.navCtrl.push(FiltrePage);
```

Fig 33 - Funcția TypeScript pe care o apelează butonul

Cum am prezentat si mai sus pentru baza de date am folosit PostGreSQL care este un sistem de gestionare open-source bazat pe limbajul SQL. Baza de date este una simpla si conține o singura tabela , cea a anunțurilor disponibile.

Data Output	Explain	Messages	History							
	id integer	categorie character varying(40)	pret integer	telefon integer	operatiune character varying(40)	locatie character varying(40)	adresa character varying(100)	numevanzator character varying(50)	imagini character varying(50)	descriereprodos character varying(200)
1	1	Apartment	60000	431009	Vanzare	Bucuresti	Calea Grivitei , nr	Ionel	img1	Vand apartament cu
2	2	Apartment	65000	231009	Vanzare	Bucuresti	Calea Rahova , nr 1	Ghita	img2	Vand apartament cu
3	2	Apartment	350	231009	Inchiriere	Bucuresti	Bulevardul Ghencea	Ghita	img3	Inchiriez apartamen
4	1	Apartment	250	431009	Inchiriere	Arad	Calea Plevnei , nr	Vasile	img3	Apartment cu 2 cam

Fig 34 – Exemplu anunțuri postate in baza de date

Tabela anunțuri conține 9 câmpuri:

- Id- conține id-ul anunțului
- Categorie: conține categoria din care face anunțul
- Pret: prețul imobilului in funcție de operațiunea dorita
- Telefon: numărul de telefon al vânzătorului
- Operatiune: Operațiunea dorita(vânzare sau închiriere)
- Adresa: Adresa completa a locuinței
- Numavanzator: Numele vânzătorului
- Imagini: pozele cu imobilul
- Descriere produs: descrierea produsului prin care se oferă mai multe detalii referitoare la anunț

6 CONCLUZII

Așa cum am prezentat și în primele capitolele ale lucrării, piața imobiliară este într-o continuă dezvoltare iar platformele pentru vânzări de imobile sunt din ce în ce mai căutate. Cu toate că aceste site-uri nu reprezintă o noutate , numărul lor este în creștere iar acest lucru este datorat cererii de cumpărare sau închiriere de locuințe foarte mare. Concurența între aceste platforme imobiliare este destul de intensă și este greu să aduci ceva în plus față de ceilalți dar aplicațiile nou create se pot face remarcate printr-o interfață simplistă , cât mai ușor de înțeles și de folosit.

Aplicația dezvoltată de mine este relativ simplă , implementând câteva funcționalități de bază ale unei platforme de acest fel. Dezvoltarea aplicației nu a implicat niciun cost , toate limbajele folosite fiind gratuite. Principalele funcționalități ale aplicației sunt:

- orice utilizator poate adăuga anunțuri pe platforma prin completarea unui formular specific pentru aceasta operație
- orice utilizator are dreptul de a căuta în lista de anunțuri disponibilă prin completarea formularului de căutare cu criteriile dorite
- doar administratorul are posibilitatea de a șterge sau de a modifica anunțurile de pe platforma

Faptul că platforma are un singur cont de administrator care o poate gestiona reprezintă de dezavantaj major și cea mai bună variantă în care ar putea fi utilizată este aceea de a promova cartiere rezidențiale construite de un anumit constructor sau anunțuri ale unei agenții imobiliare. O altă funcționalitate care ar putea fi adăugată aplicației este aceea că fiecare utilizator să poată să își facă un cont pe platformă prin intermediul căruia să poată să își salveze anunțurile favorite pentru a reveni mai târziu asupra lui sau să gestioneze mai ușor anunțurile postate de el pe site.

Din punctul meu de vedere, aplicația prezentată în lucrare și-a atins scopul principal și anume acela de a crea o platformă de anunțuri imobiliare disponibilă pe telefonul mobil prin intermediul căreia un utilizator poate să caute sau să adauge anunțuri. Interfața este una prietenoasă și ușor de înțeles astfel încât orice utilizator care știe să folosească un telefon mobil o poate folosi fără a întâmpina dificultăți prea mari. Aplicația poate fi îmbunătățită în viitor și poate deveni unul dintre portalurile principale de promovare a anunțurilor imobiliare.

7 BIBLIOGRAFIE

- [1]. <https://www.cotidianul.ro/piata-imobiliara-ca-o-arena-de-lupta/>
- [2]. <https://media.imopedia.ro/stiri-imobiliare/care-sunt-cele-mai-cautate-zone-din-bucuresti-15944-print.html>
- [3]. <http://www.mediafax.ro/economic/retrospectiva-2017-piata-imobiliara-si-a-revenit-pe-crestere-iar-preturile-au-urcat-cu-10-16913895>
- [4]. https://ro.wikipedia.org/wiki/Prima_Casă
- [5]. <https://www.agendaconstructiilor.ro/files/oportunitati-comunicate/fngcimm-60-este-ponderea-imobilelor-noi-achizitionate-prin-programul-prima-casa.html>
- [6]. http://nou.anevar.ro/sites/default/files/page-files/analizeimobiliare_raport_t1_2018.pdf
- [7]. <https://www.imobiliare.ro/indicele-imobiliare-ro>
- [8]. <https://www.imobiliare.ro>
- [9]. www.trafic.ro
- [10]. www.storia.ro
- [11]. <http://blog.storia.ro/2016/08/01/bun-venit-pe-storia>
- [12]. [https://en.wikipedia.org/wiki/Ionic_\(mobile_app_framework\)](https://en.wikipedia.org/wiki/Ionic_(mobile_app_framework))
- [13]. <https://ionicframework.com/docs/>
- [14]. https://www.w3schools.com/html/html_intro.asp
- [15]. <https://www.w3schools.com/Css/>
- [16]. <https://programmingwithmosh.com/angular/angular-4-tutorial/>
- [17]. [https://ro.wikipedia.org/wiki/Java_\(limbaj_de_programare\)](https://ro.wikipedia.org/wiki/Java_(limbaj_de_programare))

[18]. http://cs.curs.pub.ro/2015/pluginfile.php/21331/mod_resource/content/1/curs2_principiiPOO.pdf

[19]. https://www.tutorialspoint.com/spring/spring_dependency_injection.htm

[20]. <http://www.baeldung.com/spring-autowire>

[21]. https://en.wikipedia.org/wiki/Apache_Maven

[22]. https://www.tutorialspoint.com/hibernate/hibernate_mapping_files.htm

[23]. <https://www.webnotes.com/what-is-http/>

[24]. <https://code.tutsplus.com/tutorials/a-beginners-guide-to-http-and-rest--net-16340>