

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI  
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

***CarLovers***

propusă de

***Popa Ștefan-Dorin***

**Sesiunea:** *iunie, 2021*

Coordonator științific

**Lect.dr. Vidrașcu Cristian**

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI  
FACULTATEA DE INFORMATICĂ

# ***CarLovers***

***Popa Ștefan-Dorin***

**Sesiunea:** *iunie, 2021*

Coordonator științific

***Lect.dr. Vidrașcu Cristian***

Anexa 4

Avizat,

Îndrumător Lucrare de Licență

Titlul, Numele și prenumele Lect.dr. Vidrașcu Cristian

Data 22/06/2021 Semnătura 

**DECLARAȚIE privind originalitatea conținutului lucrării de licență**

Subsemnatul Popa Ștefan-Dorin cu domiciliul în strada Revoluției, nr. 19, bloc P3, scara 6, etaj 2, apartament 112, localitatea Focșani, județul Vrancea, născut la data de 02/04/1999, identificat prin CNP 1990402394431, absolvent al Universității „Alexandru Ioan Cuza” din Iași, Facultatea de Informatică specializarea Română, promoția 2021, declar pe propria răspundere, cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art.143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul: CarLovers, elaborată sub îndrumarea dl. Lect.dr. Vidrașcu Cristian, pe care urmează să o susțină în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diploma sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data azi, 22/06/2021

Semnătură student 

## DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „CarLovers.”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, 23/06/2021

Absolvent *Popa Ștefan-Dorin*

A handwritten signature in blue ink, appearing to read 'Popa', is written over a horizontal line.

(semnătura în original)

# Introducere

## Motivație

Achiziționarea unui autoturism reprezintă o problemă din ce în ce mai mare în ultima vreme. După cum se poate observa în *Figura 1*, în Aprilie 2021 avem o creștere de +218.6% la nivelul Uniunii Europene față de anul precedent în aceeași lună.

19/05/2021

	April 2021	April 2020	%Change 21/20	Jan-Apr 2021	Jan-Apr 2020	%Change 21/20
AUSTRIA	22,296	11,220	+98.7	85,818	65,900	+30.2
BELGIUM	37,192	5,296	+602.3	156,429	132,712	+17.9
BULGARIA	2,400	824	+191.3	7,984	6,751	+18.3
CROATIA	4,147	1,077	+285.1	14,621	11,078	+32.0
CYPRUS	925	234	+295.3	3,671	3,175	+15.6
CZECH REPUBLIC	18,976	10,679	+77.7	68,510	60,873	+12.5
DENMARK	15,525	10,198	+52.2	59,418	59,045	+0.6
ESTONIA	2,395	824	+190.7	8,123	6,471	+25.5
FINLAND	8,840	5,981	+47.8	36,819	34,253	+7.5
FRANCE	140,426	20,997	+568.8	582,217	385,676	+51.0
GERMANY	229,650	120,840	+90.0	886,102	822,202	+7.8
GREECE	10,198	2,434	+319.0	33,637	23,824	+41.2
HUNGARY	10,521	6,176	+70.4	42,714	38,847	+10.0
IRELAND	7,164	343	+1,988.6	55,205	50,138	+10.1
ITALY	145,033	4,295	+3,276.8	592,181	351,703	+68.4
LATVIA	1,396	683	+104.4	4,838	4,553	+6.3
LITHUANIA	3,613	1,378	+162.2	11,227	12,106	-7.3
LUXEMBOURG	4,327	1,192	+263.0	16,783	13,104	+28.1
NETHERLANDS	23,846	15,322	+55.6	104,497	118,375	-11.7
POLAND	40,149	15,239	+163.5	158,014	122,875	+28.6
PORTUGAL	14,809	2,749	+438.7	45,848	48,031	-4.5
ROMANIA	6,862	4,321	+58.8	27,624	32,300	-14.5
SLOVAKIA	6,787	3,424	+98.2	22,189	22,390	-0.9
SLOVENIA	4,281	1,846	+131.9	20,572	16,354	+25.8
SPAIN	78,595	4,163	+1,787.9	264,655	222,866	+18.8
SWEDEN	21,873	18,916	+15.6	112,743	85,057	+32.5
<b>EUROPEAN UNION (EU)</b>	<b>862,226</b>	<b>270,651</b>	<b>+218.6</b>	<b>3,422,439</b>	<b>2,750,659</b>	<b>+24.4</b>
<b>EU14<sup>2</sup></b>	<b>759,774</b>	<b>223,946</b>	<b>+239.3</b>	<b>3,032,352</b>	<b>2,412,886</b>	<b>+25.7</b>
<b>EU12<sup>3</sup></b>	<b>102,452</b>	<b>46,705</b>	<b>+119.4</b>	<b>390,087</b>	<b>337,773</b>	<b>+15.5</b>
ICELAND	781	374	+108.8	2,870	2,856	+0.5
NORWAY	13,166	7,425	+77.3	49,475	39,783	+24.4
SWITZERLAND	22,054	9,382	+135.1	78,551	64,834	+21.2
<b>EFTA</b>	<b>36,001</b>	<b>17,181</b>	<b>+109.5</b>	<b>130,896</b>	<b>107,473</b>	<b>+21.8</b>
UNITED KINGDOM	141,583	4,321	+3,176.6	567,108	487,878	+16.2
<b>TOTAL (EU + EFTA + UK)</b>	<b>1,039,810</b>	<b>292,153</b>	<b>+255.9</b>	<b>4,120,443</b>	<b>3,346,010</b>	<b>+23.1</b>
<b>WESTERN EUROPE (EU14 + EFTA + UK)</b>	<b>937,358</b>	<b>245,448</b>	<b>+281.9</b>	<b>3,730,356</b>	<b>3,008,237</b>	<b>+24.0</b>

SOURCE: NATIONAL AUTOMOBILE MANUFACTURERS' ASSOCIATIONS

<sup>1</sup> Data for Malta n.a.

<sup>2</sup> Member States before the 2004 enlargement

<sup>3</sup> Member States having joined the EU since 2004

Figura 1: Numărul de autoturisme înmatriculate în Europa

Având în vedere că există peste 50 de producători în industria automobilistică, alegerea unui autoturism potrivit pentru a satisface nevoile unei persoane poate fi una de durată. Mai mult, unele autoturisme ajung la un preț de cumpărare foarte ridicat, marea majoritate a clienților fiind nevoiți să facă un credit, consecințele acestui fapt sporesc factorul de risc în achiziționarea unui autoturism indiferent dacă acesta este nou sau utilizat.

De asemenea, marea majoritate a persoanelor nu țin pasul cu ultimele noutăți din domeniul auto, iar alții nu au experiență deloc în acesta. Având atât de multe mărci auto la dispoziție, persoanele neavizate tind să apeleze la specialiști, dar acest lucru nu reduce complet șansa de a achiziționa un bun de care aceștia să nu fie mulțumiți.

În egală măsură, este foarte important ca de îndată ce am intrat în posesia autoturismului să îi oferim o mentenanță corespunzătoare pentru a prelungi viața acestuia. Acest lucru se poate realiza prin inspecții periodice la un centru autorizat, dar cu toate acestea tot nu putem fi 100% siguri că nu vom avea nicio problemă, deoarece unele autoturisme au diverse probleme de proiectare, unele fiind destul de grave. Cu toate acestea, ca proprietarul să își poată proteja bunul ar trebui să își creeze legături cu alți posesori ai aceleasi mărci auto pentru a fi la curent cu diferitele probleme ale acestora.

Am creat aplicația *CarLovers* pentru a rezolva problemele enumerate mai sus.

## **Ideea aplicației**

*CarLovers*, după cum îi spune și numele este o aplicație mobilă cu și despre pasionații de mașini. Ideea principală este îmbinarea funcționalităților forumurilor destinate mărcilor de mașini, grupurilor de pe rețelele de socializare, site-urilor auto ș.a.m.d cu scopul rezolvării de probleme și nelămuriri despre anumite mărci auto, dar și în egală măsură de a crea comunități care să unească și să întărească prietenii.

## **Lista funcționalităților principale**

Pagina principală constă dintr-un flux de știri din domeniul auto. Știrile sunt preluate de pe diferite site-uri de specialitate, fiecare utilizator fiind liber să aleagă ce site-uri îi vor fi afișate pentru vizualizare direct în aplicație. Așadar, ținta acestei caracteristici este să ofere utilizatorului șansa de a fi la curent cu ultimele noutăți auto, iar în egala măsură să aibă parte și de o notă de divertisment.

Caracteristica principală este reprezentată de apartenența la cluburi. Utilizatorii au posibilitatea de a-și înregistra autoturismele, făcând o descriere ierarhică exactă a acestora din punct de vedere a mărcii, modelului, versiunii, anului și motorizării. În continuare, se poate opta pentru înscrierea în cluburi în funcție de marcă și model. Motivul principal pentru care s-a ales această ierarhizare este dat de faptul că marea majoritate a producătorilor păstrează modelul de baza al autoturismului, iar pe parcursul anilor acesta primește diferite îmbunătățiri pe partea

estetică, dar și tehnică. Aceste îmbunătățiri poartă denumirea de “*facelift*” în industria automobilistică. Așadar, unele elemente ale unui autoturism de actualitate se pot potrivi cu altele ale unui autoturism mai vechi. Paginile de cluburi sunt destinate cu scopul de a crea o interacțiune între membrii acestuia în vederea clarificării anumitor aspecte fie ele de natură tehnică sau nu. Aceste lucruri se pot desfășura prin posibilitatea utilizatorului de a pune întrebări fiind capabil să atașeze atât fotografii cât și videoclipuri pentru o mai bună descriere a problemei, ca ulterior acesta să primească răspunsuri din partea comunității formate din membrii acelui club, această trăsătură purtând numele de postare. De asemenea, cluburile dispun și de o secțiune pentru anunțuri, scopul principal al acestora este de a semnaliza diferite evenimente ce vor avea loc, ori pentru a descrie regulamente de ordine interioară, în această secțiune doar administratorii clubului pot adăuga date.

De asemenea, altă caracteristică importantă o constituie posibilitatea de a crea întâlniri între pasionații de mașini. Rolul principal al acestor întâlniri este divertismentul, dar de asemenea se pot alătura și alți iubitori ai mărcii pentru a vedea în detaliu autoturismele, eventual pentru a le putea conduce, cu acordul proprietarilor, bineînțeles. Pentru o ușoară desfășurare a acestor evenimente, aplicația *CarLovers* înregistrează și afișează poziția fiecărui utilizator pe harta integrată în aplicație, acest lucru întâmplându-se în timp real. Prin acest mod, fiecare utilizator poate vedea câte persoane participă la un eveniment, dar și care sunt acestea.

## **Gradul de noutate**

În urma realizării studiului de piață, am găsit doar două aplicații mobile, “*CarMeets*” și “*Throdle*” care au funcționalități asemănătoare cu cele pe care le-am enumerat anterior.

La o investigație amănunțită a celor două aplicații am observat următoarele diferențe față de aplicația creată de mine:

Atât “*CarMeets*” cât și “*Throdle*” sunt în stadiul de dezvoltare, așadar multe dintre funcționalitățile lor nu sunt finalizate ori au probleme. De exemplu, nu am reușit să-mi înregistrez niciun autoturism în aplicația “*CarMeets*”, iar acest lucru a fost semnalat și de alți utilizatori ai aplicației așa cum se poate vedea și în Figura 2 de mai jos. De asemenea, “*Throdle*” abia a fost lansat în acest an(2021), iar numărul de instalări este mic(în jur de 500), în schimb “*CarMeets*” este pe piață din anul 2019 și are peste 100.000 de instalări.



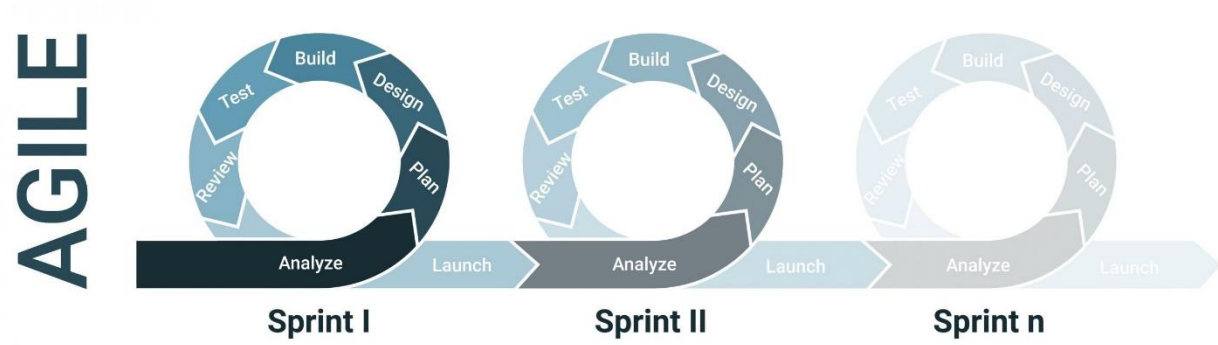
Figura 2: Comentarii ale utilizatorilor "CarMeets" de pe „Magazin Google Play”

Plusurile pe care aplicația dezvoltată de mine le are față de concurență sunt baza de date completă incluzând atât anul de fabricație cât și motorizarea împreună cu caracteristici specifice fiecărei versiuni, posibilitatea de a activa în mai multe cluburi în funcție de marcă și model și nu în ultimul rând ușurința cu care se pot urmări evenimentele cu ajutorul hărții în timp real.

## Metodologia

Metodologia folosită pentru dezvoltarea aplicației a fost “*Agile Scrum*”. Profesorul coordonator a jucat rolul de client care a făcut parte din echipa de dezvoltare. S-au organizat întâlniri frecvente pentru a monitoriza progresul, părerile domnului profesor m-au ajutat să înțeleg necesitățile și să le ajustez în consecință. Aplicația a fost împărțită în mai multe module independente considerate “mini-proiecte”. Fiecare iterație este bazată pe munca de dinainte, făcând produsul gradual mai bun. Prin abordarea acestei metodologii, adaptările s-au putut realiza ușor la modificările apărute pe parcurs.





*Figura 3: Metodologia Agile*

## Cuprins

<b>1</b>	<b>Aplicații mobile</b>	<b>12</b>
1.1	De ce aplicații mobile?	12
1.2	De ce Android?	12
1.3	Tipuri de aplicații mobile	13
<b>2</b>	<b>Proiectarea arhitecturală</b>	<b>14</b>
2.1	Mediul de dezvoltare a programelor	14
2.2	Arhitectura sistemului	15
2.3	Baza de date	16
2.3.1	Firebase	16
2.3.2	Modelarea bazei de date	17
2.4	Identificarea cerințelor	22
2.4.1	Diagrame specifice scenariilor de utilizare	23
2.4.2	Diagrame de clasă	28
<b>3</b>	<b>Implementare</b>	<b>32</b>
3.1	Activitatea principală	33
3.2	Înregistrare și autentificare	34
3.3	Știri din domeniul auto	35
3.4	Cluburi auto	36
3.5	Întâlniri auto	38
3.6	Profil utilizator	39
3.6.1	Adăugare de noi autoturisme	41
<b>4</b>	<b>Testare</b>	<b>42</b>
4.1	Interfața grafică	42
4.2	Funcționalitățile aplicației	42
4.2.1	Înregistrare și autentificare	42
4.2.2	Serviciul de localizare	43

## Lista Figurilor

Figura 1: Numărul de autoturisme înmatriculate în Europa .....	5
Figura 2: Comentarii ale utilizatorilor “CarMeets” de pe „Magazin Google Play” .....	8
Figura 3: Metodologia Agile .....	9
Figura 4: Statistici ale sistemelor de operare pentru telefoanele mobile (StatCounter, 2021) .....	13
Figura 5: Nivelul “API” suportat de fiecare versiune Android .....	14
Figura 6: Planul de bază al sistemului .....	15
Figura 7: Modelul bazei de date pentru colecția “Users” .....	18
Figura 8: Modelul bazei de date pentru colecția “UserCars” .....	19
Figura 9: Web Crawler (By Vector version by dnet based on image by User:ChaTo - self-made, based on image from PhD. Thesis of Carlos Castillo, image released to public domain by the original author., CC BY-SA 3.0, <a href="https://commons.wikimedia.org/w/index.php?curid=3374592">https://commons.wikimedia.org/w/index.php?curid=3374592</a> ) .....	20
Figura 10: Exemplu de intrare în urma prelucrării datelor de către “Crawler” .....	21
Figura 11: Modelul bazei de date pentru colecția “CarsData” .....	21
Figura 12: Modelul bazei de date pentru colecția “Posts” .....	22
Figura 13: Scenariul de utilizare principal al aplicației .....	23
Figura 14: Scenariul de utilizare al activității “Știri din doemniul auto” .....	24
Figura 15: Scenariul de utilizare al activității “Explorare cluburi” .....	25
Figura 16: Scenariul de utilizare al activității “Întâlniri” .....	26
Figura 17: Scenariul de utilizare al activității “Vizualizare profil utilizator” .....	27
Figura 18: Diagrama de clase specifică autentificării .....	28
Figura 19: Diagrama de clase specifică activității de întâmpinare .....	29
Figura 20: Diagrama de clase specifică fragmentului cu știri din domeniul auto .....	29
Figura 21: Diagrama de clase specifică fragmentului cu pagina de cluburi .....	30
Figura 22: Diagrama de clase specifică fragmentului cu întâlniri auto .....	30
Figura 23: Diagrama de clase specifică fragmentului cu profilul utilizatorului .....	31
Figura 24: Schițele paginilor principale .....	32
Figura 25: Interfața grafică și șablonul activității principale .....	33
Figura 26: Interfața grafică pentru activitățile de înregistrare și autentificare .....	34
Figura 27: Interfața grafică a fragmentului “Știri din domeniul auto” și preferințele site-urilor de știri .....	35
Figura 28: Interfața grafică a fragmentului “Cluburi auto” și activitățile conexe .....	37
Figura 29: Interfața grafică a fragmentului “Întâlniri auto” și activitățile conexe .....	38
Figura 30: Interfața grafică a fragmentului “Profil utilizator” și activitățile conexe .....	40
Figura 31: Interfața grafică specifică activităților de adăugare a unui autoturism .....	41

# 1 Aplicații mobile

## 1.1 De ce aplicații mobile?

În ultima vreme, telefoanele mobile sunt un obiect de care din ce în ce mai multe persoane devin dependente, fie că este vorba de socializare, trafic de date pe internet sau rezolvarea problemelor la locul de muncă, acest gadget fiind nelipsit în buzunarele oamenilor. Îl găsim util în rezolvarea majorității sarcinilor de zi cu zi, acest fapt a rezultat din evoluția pieței care a adus telefoane din ce în ce mai performante atât pe partea software cât și hardware.

Chintesența dispozitivelor mobile este dată de faptul că indiferent de locație un utilizator poate folosi aplicația. Faptul că acest dispozitiv este de o dimensiune și greutate redusă și mai ales că poate combina caracteristicile unui computer plus periferice îl plasează pe locul întâi, fiind cea mai buna alegere pentru dezvoltarea aplicației alese. Acest gadget are de obicei măcar o cameră, GPS, Wi-Fi sau poate folosi date mobile, lucruri ce sunt esențiale pentru a putea folosi aplicația.

## 1.2 De ce Android?

Platforma software adoptată pentru dezvoltarea aplicației este Android.

Android este un sistem de operare pentru dispozitive și telefoane mobile bazat pe nucleul Linux, dezvoltat inițial de compania Google, iar mai târziu de consorțiul comercial Open Handset Alliance.(Wikipedia)

Un factor important care a hotărât decizia de a alege dezvoltarea aplicației în Android este dat de popularitatea acestuia, așa cum se poate observa și în Figura 4. Având o aplicație dependentă de conținut generat de utilizatori, cu cât aceștia sunt mai mulți, cu atât aplicația are mai mult de câștigat.

Alt factor care a contribuit la această alegere este ușurința cu care se poate crea o aplicație mobilă indiferent de dispozitivul pe care dezvoltatorul îl folosește. De exemplu, pentru a crea o aplicație în sistemul de operare iOS, trebuie folosit neapărat un dispozitiv Mac, pe de altă parte, dezvoltarea de aplicații Android se poate realiza fie pe Mac, Windows sau Linux.

De asemenea, flexibilitatea este îmbunătățită dat de faptul că Android este “*open-source*”, așadar este foarte ușor de personalizat.

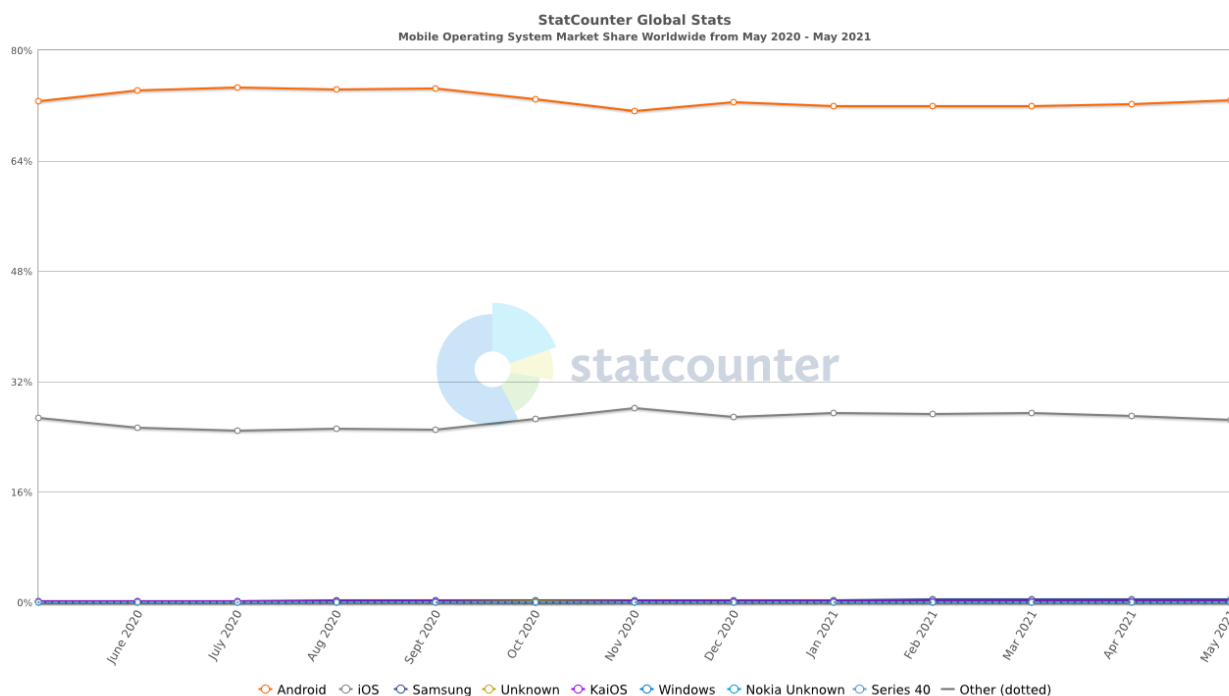


Figura 4: Statistici ale sistemelor de operare pentru telefoanele mobile (StatCounter, 2021)

### 1.3 Tipuri de aplicații mobile

În general aplicațiile mobile pot fi clasificate în două categorii: nativ sau web. O aplicație mobilă nativă este o aplicație destinată să ruleze pe un sistem de operare specific al telefonului mobil. De exemplu, o aplicație nativă dezvoltată pentru iPhone o să funcționeze doar pe platforma proprietară, iOS.

Pe de altă parte, o aplicație mobilă web constă într-un site obișnuit care este găzduit pe un server și poate fi accesat printr-un browser web. De fiecare dată când aplicația este executată, toate fișierele sunt descărcate și procesate local.

În urma analizării posibilităților s-a ales varianta nativă de dezvoltare din următoarele motive: aplicațiile native sunt mult mai rapide, deoarece aceasta este creată și optimizată pentru platforma specifică (Android), un plus ar fi dat de faptul că marea majoritate a datelor sunt stocate direct în telefon, deci nu trebuie descărcate toate datele la fiecare rulare a aplicației. De asemenea, aplicația poate accesa cu ușurință partea hardware a dispozitivului cum ar fi GPS sau camera.

## 2 Proiectarea arhitecturală

### 2.1 Mediul de dezvoltare a programelor

Un „*SDK*” (abr. engl. „*Software Development Kit*” - Trusă de dezvoltare a programelor) este un set de unelte folosite de un programator pentru a scrie programe pentru un anumit sistem de operare, platformă hardware, sau pachet de programe existent. “*SDK*”-urile conțin de regulă biblioteci sau „*APP*”-uri apelabile de către programatori. (Wikipedia, 2018)

Alegerea celui mai mic „*APP*” joacă un rol important în dezvoltarea de aplicații, deoarece determină ce utilizatori o pot instala de pe “Magazin Google Play”.

	Name	API Level	Revision
<input checked="" type="checkbox"/>	Android 11.0 (R)	30	3
<input type="checkbox"/>	Android 10.0 (Q)	29	5
<input checked="" type="checkbox"/>	Android 9.0 (Pie)	28	6
<input type="checkbox"/>	Android 8.1 (Oreo)	27	3
<input type="checkbox"/>	Android 8.0 (Oreo)	26	2
<input type="checkbox"/>	Android 7.1.1 (Nougat)	25	3
<input type="checkbox"/>	Android 7.0 (Nougat)	24	2
<input type="checkbox"/>	Android 6.0 (Marshmallow)	23	3
<input type="checkbox"/>	Android 5.1 (Lollipop)	22	2
<input type="checkbox"/>	Android 5.0 (Lollipop)	21	2
<input type="checkbox"/>	Android 4.4W (KitKat Wear)	20	2
<input type="checkbox"/>	Android 4.4 (KitKat)	19	4
<input type="checkbox"/>	Android 4.3 (Jelly Bean)	18	3
<input type="checkbox"/>	Android 4.2 (Jelly Bean)	17	3
<input type="checkbox"/>	Android 4.1 (Jelly Bean)	16	5
<input type="checkbox"/>	Android 4.0.3 (IceCreamSandwich)	15	5
<input type="checkbox"/>	Android 4.0 (IceCreamSandwich)	14	4
<input type="checkbox"/>	Android 3.2 (Honeycomb)	13	1
<input type="checkbox"/>	Android 3.1 (Honeycomb)	12	3
<input type="checkbox"/>	Android 3.0 (Honeycomb)	11	2
<input type="checkbox"/>	Android 2.3.3 (Gingerbread)	10	2
<input type="checkbox"/>	Android 2.3 (Gingerbread)	9	2
<input type="checkbox"/>	Android 2.2 (Froyo)	8	3
<input type="checkbox"/>	Android 2.1 (Eclair)	7	3

Figura 5: Nivelul “API” suportat de fiecare versiune Android

În Figura 5 se poate observa nivelul “API” specific fiecărei versiuni Android. În ultimii ani, Google nu mai oferă statistici cu vedere la numărul de utilizatori în funcție de versiunea de Android, cu toate acestea aplicația rulează pe orice versiune de Android mai mare decât 7.0(Nougat), această versiune fiind lansată pe 22 August 2016. Așadar, putem trage concluzia că marea majoritate a utilizatorilor Android dețin un dispozitiv cu minim această versiune, deci răspândirea acestei aplicații nu va fi o problemă din acest punct de vedere.

## 2.2 Arhitectura sistemului

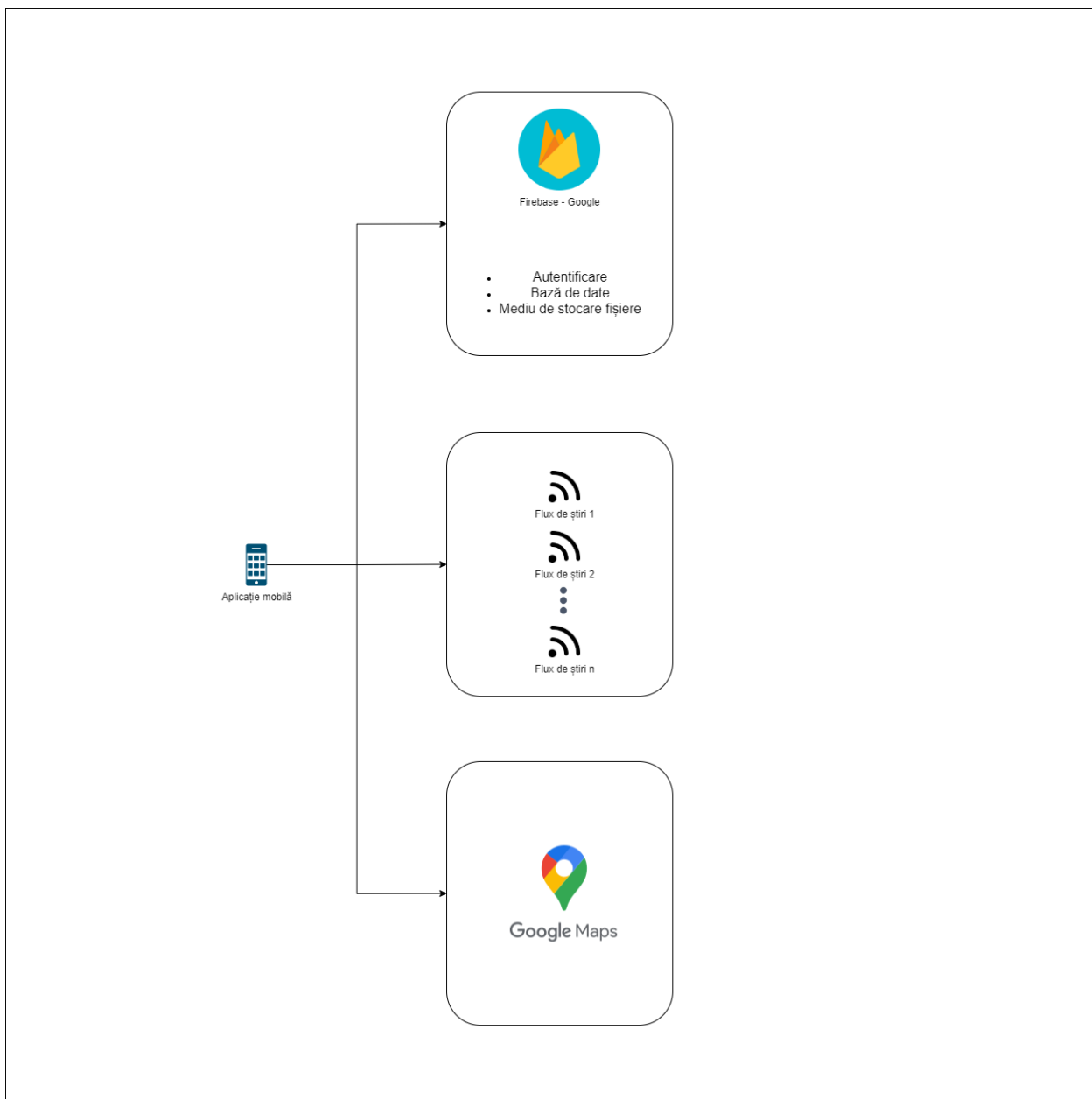


Figura 6: Planul de bază al sistemului

Dupa cum se poate observa în Figura 6, sistemul este format la bază dintr-o aplicație mobilă care folosește trei surse externe diferite prin care utilizatorului îi sunt oferite funcționalitățile menționate mai sus. Mai multe informații despre cele trei componente de bază sunt descrise în capitolele următoare.

## 2.3 Baza de date

Unul dintre cele mai importante aspecte ale acestei aplicații este un server cu care aceasta să poată comunica bidirecțional. Serverul trebuie în principal să trimită și să recepționeze date de la utilizator, astfel încât acestuia i se pot prezenta informațiile cerute. Pentru alegerea unui server se pot lua mai multe decizii, unul dintre acestea ar fi crearea unui server de la zero care să poată comunica cu aplicație printr-un set de reguli personalizat (“APP”). Problema cu această abordare o constituie timpul și resursele limitate pentru finalizarea aplicației. Abordarea favorabilă pentru această problemă este folosirea unui server deja existent care poate satisface toate nevoile aplicației.

În urma analizării posibilităților, s-a decis folosirea platformei “Firebase” de la Google. Firebase este o platformă dezvoltată de Google încă din 2014 pentru crearea de aplicații mobile și web prin care dezvoltatorii se pot concentra mai mult pe utilizatori decât să își creeze propriul “APP” sau să gestioneze serverul. De asemenea, această platformă vine cu o multitudine de funcții prestabilite prin care se pot dezvolta aplicațiile mult mai ușor făcând operațiile de bază cum ar fi autentificarea, stocarea și prelucrarea de date de orice tip mult mai ușoară.

### 2.3.1 Firebase

Pentru a putea folosi serviciile Firebase procedura este foarte simplă, tot ce trebuie să facem este să înregistrăm aplicația în consola Firebase pentru crearea unei asocieri cu aceasta, mai apoi putem adăuga orice “SDK”, chiar cu ajutorul editorului Android Studio. Câteva exemple sunt: “Cloud Firestore”, “Authentication”, “Cloud Storage” etc.

În momentul în care creăm un proiect Firebase, de fapt se creează în spate și un proiect pentru “Google Cloud”. Așadar, deoarece un proiect Firebase este și unul Google Cloud, cele 2 entități împart același plan de facturare al serviciilor. Se pot alege două opțiuni: “free Spark plan” sau “price scaling Blaze plan”. Pentru dezvoltarea aplicației și testare a fost de ajuns să aleg “free Spark plan” care este gratuit, dar are câteva limitări. Ulterior, în momentul lansării aplicației la piața largă se poate face foarte ușor conversia la planul “price scaling Blaze plan”, de îndată ce aplicația își poate genera fonduri. În continuare voi enumera restricțiile pentru planul gratis din cadrul Firebase:

- **Cloud Firestore**
  - Stocarea datelor – maxim 1Gb
  - Scrierea documentelor – 20,000/zi
  - Ștergerea documentelor – 20,000/zi



- Citirea documentelor - 50,000/zi
- **Serviciul de autentificare**
  - Autentificare pe telefonul mobil – 10,000/lună
  - Alte servicii de autentificare – gratis
- **Cloud Storage**
  - Stocarea datelor – maxim 5Gb
  - Descărcarea datelor – maxim 1Gb/zi
  - Operații de încărcare – 20,000/zi
  - Operații de descărcare – 50,000/zi
- **Test Lab**
  - Dispozitive virtuale – 10 teste/zi
  - Dispozitive fizice – 5 teste/zi

Mai multe despre acestea se pot vedea la adresa: <https://firebase.google.com/pricing/>.

Cele mai importante servicii care au fost punctul de interes în dezvoltarea aplicației sunt Cloud Firestore, Authentication și Cloud Storage.

Cloud Firestore reprezintă implementarea firebase a unei baze de date nerelațională care se bazează pe documente folosind un format JSON. Documentele nu trebuie să aibă o structură uniformă, astfel încât se pot stoca diferite tipuri de date cum ar fi: șiruri de caractere, numere, dar și unele mai complexe cum ar fi obiecte, timp, subcolecții de documente. Prin caracteristicile prezentate, această bază de date s-a dovedit a fi scalabilă pe parcursul dezvoltării aplicației.

Cloud Storage a fost folosit cu scopul stocării fotografiilor și videoclipurilor utilizatorilor, dar acesta poate fi folosit pentru stocarea oricărui tip de fișier.

### 2.3.2 Modelarea bazei de date

După cum am menționat anterior, baza de date este structurată pe documente, aceasta fiind nerelațională. Creând un model pentru acest tip de bază de date poate fi o sarcină mai dificilă, deoarece nu este necesar să avem relații între tabele. Pentru realizarea aplicației s-au folosit următoarele colecții principale: Users, UserCars, UserLocation, CarsData, CarClubs, Posts, Likes, Comments, Announces.

#### Users

Această colecție conține date despre utilizatori. În Figura 7 se poate vedea schema acestei colecții. Dupa cum se poate observa, nu este necesară stocarea datelor confidențiale cum ar fi parolele, deoarece aceste documente sunt create de îndată ce utilizatorul este înregistrat, toate datele confidențiale fiind gestionate de Firebase Authentication pentru conectare, aceste date fiind direct relaționate cu “UUID” care reprezintă identificatorul unic pentru fiecare utilizator generat de Firebase.

- **Bio:** biografia utilizatorului care va fi vizibilă în pagina de profil a acestuia
- **Email:** emailul utilizatorului – acesta nu este vizibil nicaieri în aplicație
- **firstName:** Numele utilizatorului
- **lastName:** Prenumele utilizatorului
- **followingClubs:** Un “array” unde se păstrează ID-urile unice specifice paginilor de cluburi
- **imageUrl:** O legătură către fotografia de profil a utilizatorului, aceasta fiind stocată în Cloud Storage
- **userType:** Tipul utilizatorului: admin sau user
- **username:** Nume unic de identificare a utilizatorului

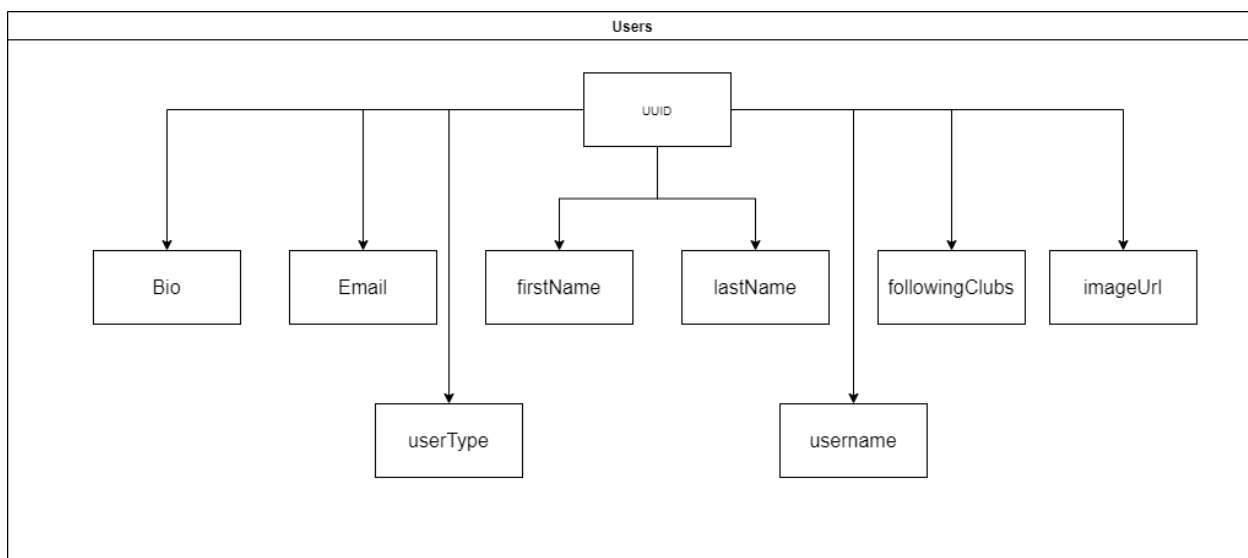


Figura 7: Modelul bazei de date pentru colecția “Users”

## UserCars

Această colecție conține date despre autoturismele fiecărui utilizator. În Figura 8 se poate vedea schema acestei colecții. Toate autoturismele sunt adăugate după un șablon stabilit așa cum se regăsesc în **CarsData**, colecție care este detaliată mai jos.

- **Brand, model, year, version, engine:** date prestabilite despre un anumit autoturism conform colecției carsData
- **brandLogo:** o legătura către imaginea logo-ului specific brandului ales stocat în Cloud Storage
- **carPhoto:** o legătura către imaginea introdusă de utilizator în momentul înregistrării autoturismului, imagine care este stocată de asemenea în Cloud Storage
- **timeAdded:** data și ora când a fost făcut documentul i.e. a fost introdus autoturismul
- **userId:** UUID generat de Firebase specific utilizatorului

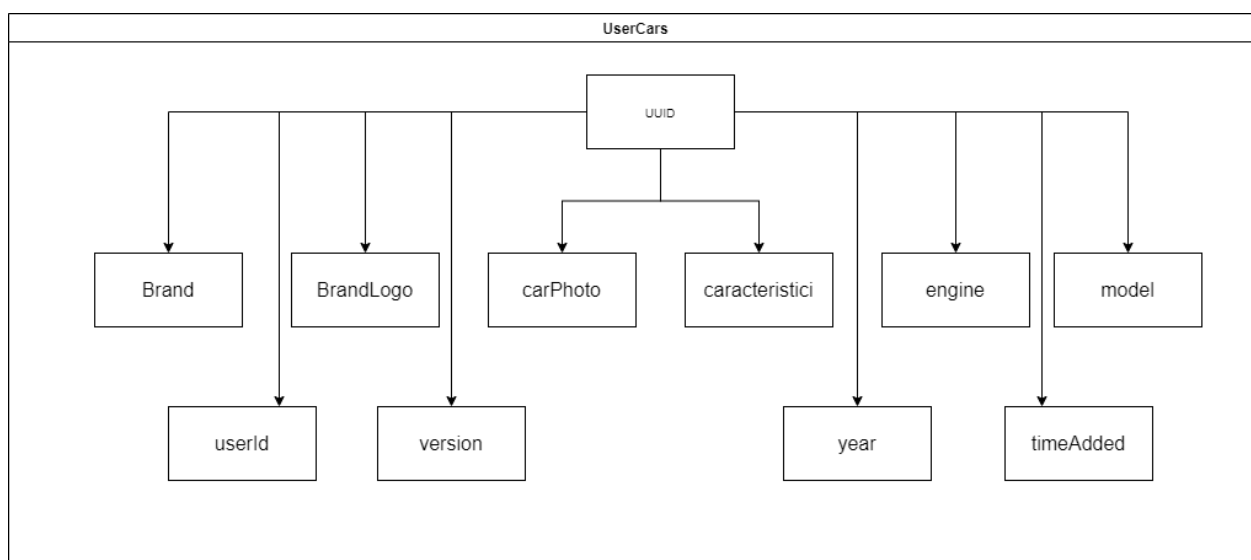


Figura 8: Modelul bazei de date pentru colecția “UserCars”

## CarsData

Această colecție conține toate datele necesare pentru configurarea ierarhică a autoturismelor în funcție de brand, model, an, versiune și motorizare. Pentru obținerea acestor date s-a creat un “Crawler”. În Figura 9 este reprezentată arhitectura acestuia, dar pentru a clarifica orice nelămurire voi explica pe scurt întrebunțările sale. Un “Crawler” are ca scop stocarea datelor de pe anumite site-uri web, lucru care ar dura foarte mult dacă s-ar face manual. Ținta lui este să descarce recursiv diferite pagini web pentru a le prelucra ulterior păstrând doar datele folositoare. În cazul aplicației mele, am avut ca pagină web țintă <https://www.auto-data.net/ro/>. La o primă vedere putem observa cum datele despre autoturisme sunt structurate ierarhic, lucru ce a facilitat prelucrarea datelor. În vederea creării unui “Crawler”, am folosit limbajul de programare Python, deoarece acesta are două librării esențiale dezvoltării acestui program: “Requests” și „BeautifulSoup4”.

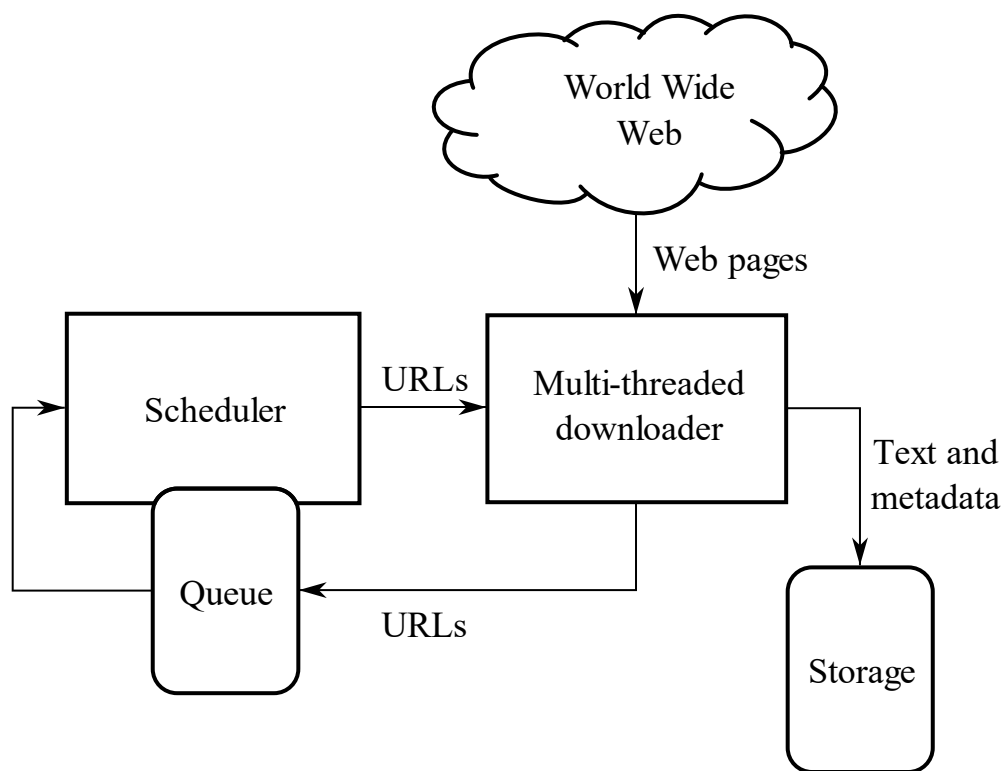


Figura 9: Web Crawler (By Vector version by dnet based on image by User:ChaTo - self-made, based on image from PhD. Thesis of Carlos Castillo, image released to public domain by the original author., CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=3374592>)

Am folosit librăria “*Requests*” pentru a putea face cereri “HTTP” cu ușurință, iar mai apoi prelucrarea fișierelor html s-a făcut cu ajutorul librăriei “*BeautifulSoup4*”. BeautifulSoup4 creează un arbore de analiză specific documentului, fapt care a facilitat la găsirea și prelucrarea datelor.

Am ales să stochez datele în format JSON, fapt ce m-a ajutat ulterior pentru a introduce datele în Cloud Firestore folosind Cloud Functions.

Cloud Functions reprezintă alt modul al Firebase prin care se pot modifica datele atât din baza de date Cloud Firestore, mediul de stocare Cloud Storage, dar și altele. Aceste funcții sunt suportate doar în limbajele de programare JavaScript sau TypeScript, codul fiind stocat și rulat în “Google’s cloud”.

În Figura 10 de mai jos am dat un exemplu de intrare în urma prelucrării datelor.

```

{"Brand": "Abarth", "ModelURL": "https://www.auto-data.net/ro/abarth-brand-200",
"LogoImgURL": "https://www.auto-data.net/img/logos/Abarth.png", "Models": [{"Model":
"124 Spider", "VersionURL": "https://www.auto-data.net/ro/abarth-124-spider-model-
2152", "Versions": [{"Version": "Abarth 124 GT", "EnginesURL": "https://www.auto-

```

```
data.net/ro/abarth-124-gt-generation-6774", "VersionImgURL": "https://www.auto-
data.net/images/f49/Abarth-124-GT_thumb.jpg", "Year": "2018 - ", "Engines":
[{"Engine": "1.4 MultiAir (170 CP) Automatic", "Caracteristici": "Viteza maximă : 229
km/h | 142.29 mph\n0-100 km/h: 6.9 sec, 0-60 mph: 6.6 sec\nConsumul de combustibil:
6.6 l/100 km | 36 US mpg | 43 UK mpg\n"}, {"Engine": "1.4 MultiAir (170 CP)",
"Caracteristici": "Viteza maximă : 232 km/h | 144.16 mph\n0-100 km/h: 6.8 sec, 0-
60 mph: 6.5 sec\nConsumul de combustibil: 6.4 l/100 km | 37 US mpg | 44 UK mpg\n"}]]
```

Figura 10: Exemplu de intrare în urma prelucrării datelor de către “Crawler”

Așadar, în urma inserării datelor în Cloud Firestore am ajuns la următoarea ierahie care se poate vedea în Figura 11.

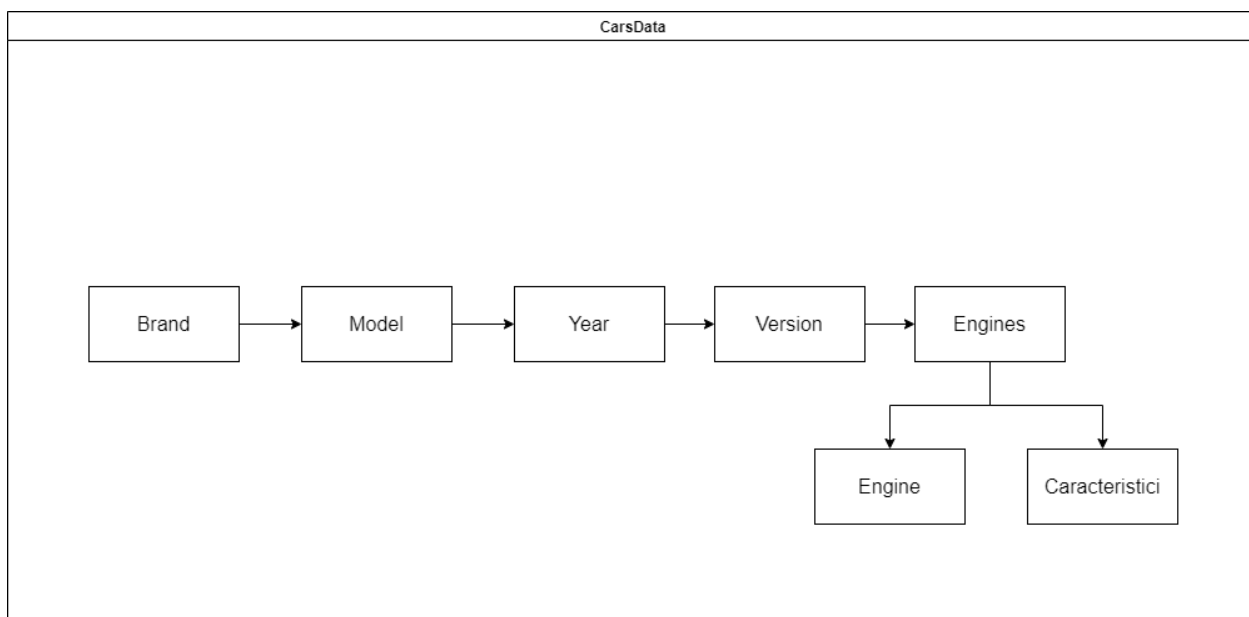


Figura 11: Modelul bazei de date pentru colecția “CarsData”

## CarClubs

Această colecție este responsabilă cu definirea totalităților cluburilor. Există câte un club predefinit pentru fiecare marcă și model de autoturism (de exemplu: pagina clubului Peugeot 308). Aceste cluburi s-au format tot cu ajutorul Cloud Functions folosind **CarsData**, așadar pentru orice model de autoturism există și un club specific, prin acest mod suntem siguri că nu vom avea utilizatori care și-au adăugat autoturisme în aplicație și nu au un club la care să adere.

## Posts

Această colecție este responsabilă cu gestionarea postărilor din paginile de cluburi. În Figura 12 se poate vedea schema acestei colecții.

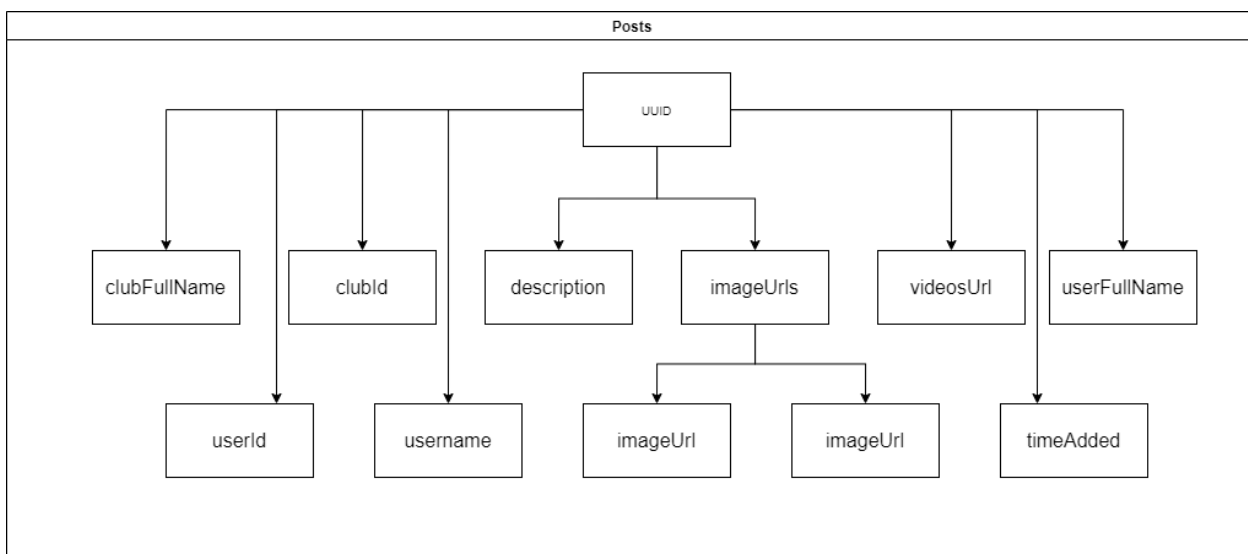


Figura 12: Modelul bazei de date pentru colecția “Posts”

Conceptul de postare este abstractizat astfel încât utilizatorul i se pot afișa postări din mai multe cluburi dat de beneficiile structurării nerelaționale.

Despre celelalte colecții modelarea este trivială, astfel “UserLocation” conține informații despre locația utilizatorului, date care sunt actualizate în timpul utilizării aplicației, “Likes” și “Comments” se folosesc pentru a stoca date extra despre o postare, în cazul acesta numărul de aprecieri și comentarii, iar “Announces” este folosit pentru a stoca date despre anunțuri specifice unui club anume.

## 2.4 Identificarea cerințelor

Pe parcursul dezvoltării aplicației este foarte important ca cerințele să fie identificate, astfel încât scopul aplicației cât și limitele ei să fie clare atât pentru mine cât și pentru coordonator. Așadar, prin această abordare se clarifică o mare parte din integrare a modulelor implementate separat încă de la începutul dezvoltării. Pentru o bună comunicare s-au folosit două tipuri de diagrame UML: diagrame pentru scenarii de utilizare și diagrame de clase.

Diagrama scenariilor de utilizare este o diagramă comportamentală care captează cerințele sistemului. Aceasta este folosită și pentru a arăta relațiile dintre elementele interfeței grafice în legătură cu o acțiune făcută de actori<sup>1</sup>.

Diagrama de clase reprezintă structura generală a aplicației. Aceasta conține clasele specifice aplicației, relațiile dintre ele, dar și cele mai importante metode care sunt folosite în cadrul

<sup>1</sup> Entitățile din afara unui program

acestora. Ca scop principal, diagramele de clasă sunt folosite pentru a modela structura unui program și pentru a surprinde conexiunile semantice sau interacțiunile care se stabilesc între elementele componente.

## 2.4.1 Diagrame specifice scenariilor de utilizare

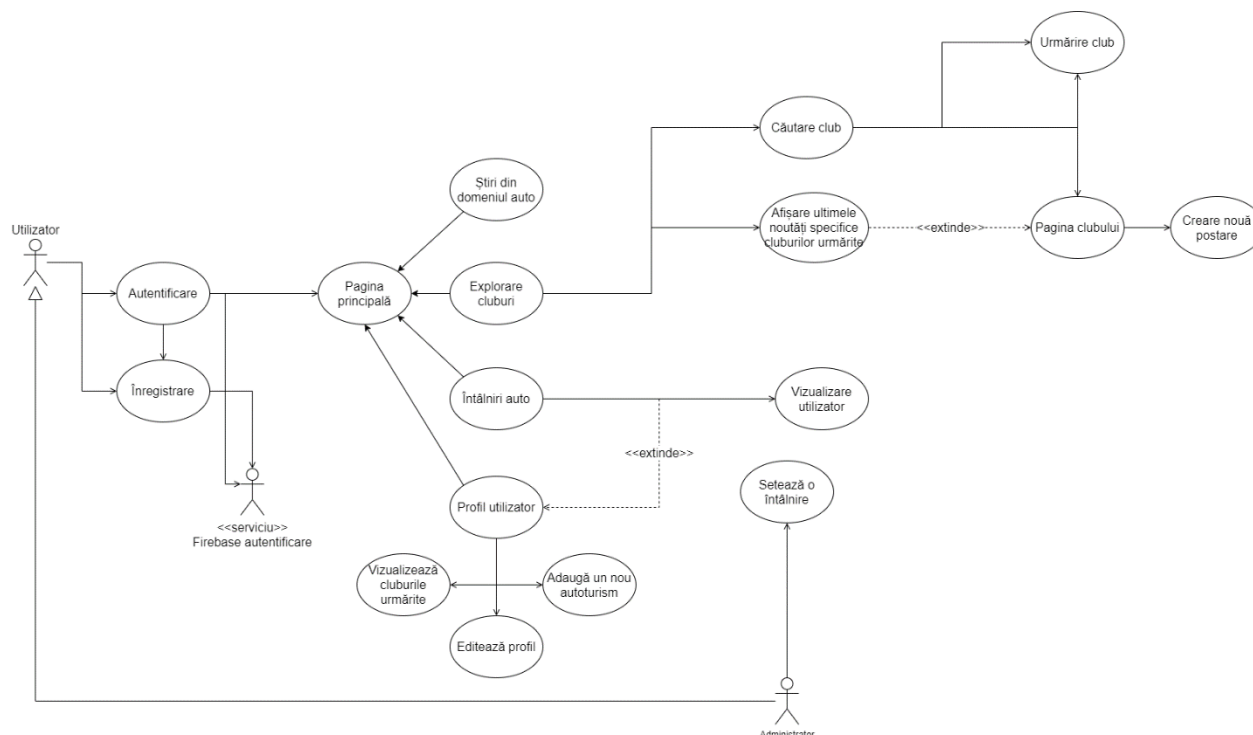


Figura 13: Scenariul de utilizare principal al aplicației

În Figura 13 de mai sus este reprezentat scenariul de utilizare principal al aplicației. După ce utilizatorul se autentifică cu succes la serviciul de autentificare Firebase acesta va fi redirecționat către pagina principală în care se poate afișa una din cele patru activități de bază: **Știri din domeniul auto**, **Explorare cluburi**, **Întâlniri auto**, **Vizualizare profil utilizator**.

### Știri din domeniul auto

În Figura 14 de mai jos este afișat scenariul de utilizare al activității “Știri din domeniul auto”. Funcționalitatea acestei activități este destul de simplă, fiecare utilizator are posibilitatea să afle ultimele noutăți din domeniul auto conform unor site-uri de specialitate. Structura paginii este intuitivă, aceasta fiind la îndemâna oricui, fapt care duce la o navigare accesibilă prin opțiunile afișate. Un utilizator poate să-și seteze preferințele site-urilor de specialitate de pe care va primi noutăți, fiecare știre putând fi citită în totalitate pe pagina web eminentă.

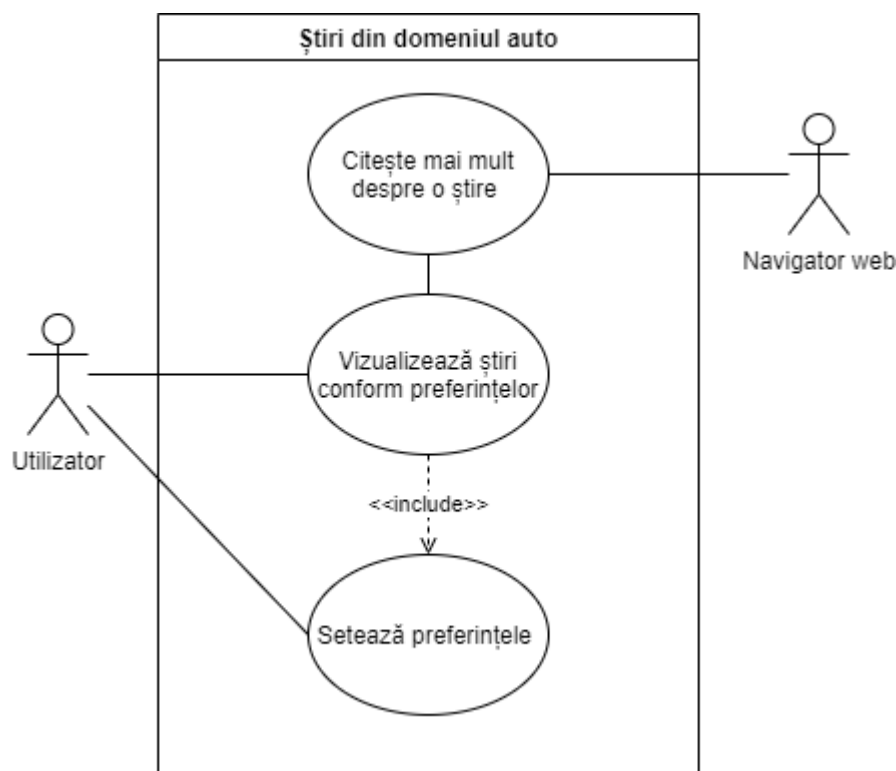


Figura 14: Scenariul de utilizare al activității “Știri din domeniul auto”

## Explorare cluburi

În Figura 15 de mai jos este prezentat scenariul de utilizare al activității “Explorare cluburi”. De îndată ce utilizatorul accesează această activitate el are la dispoziție două alegeri: fie caută o pagină de club specifică, ori optează pentru afișarea ultimelor postări conform urmărilor pe care le are. În cazul în care utilizatorul optează pentru a doua variantă, acesta va fi redirecționat către o nouă activitate unde are parte de funcționalitățile similare vizualizării ultimelor postări specifice unui club anume, doar că de această dată postările sunt colectate din mai multe pagini de cluburi. Cât despre prima opțiune, funcționalitățile sunt după cum urmează: utilizatorul are 3 meniuri principale prin care poate naviga “vizualizare postări”, “vizualizare anunțuri”, “vizualizare membri club”.

În cazul primului meniu, funcționalitățile sunt:

### Adaugă o nouă postare

După cum îi spune și numele, această funcționalitate surprinde adăugarea unei noi postări într-un club anume. Termenul de “postare” reprezintă nucleul aplicației, deoarece cu ajutorul acestora se poate realiza comunicarea între membrii aplicației la nivelul unui club. Conceptul de postare are la bază adresarea întregii comunități, făcând public un material care poate fi descris de un text, una sau mai multe poze sau videoclipuri.



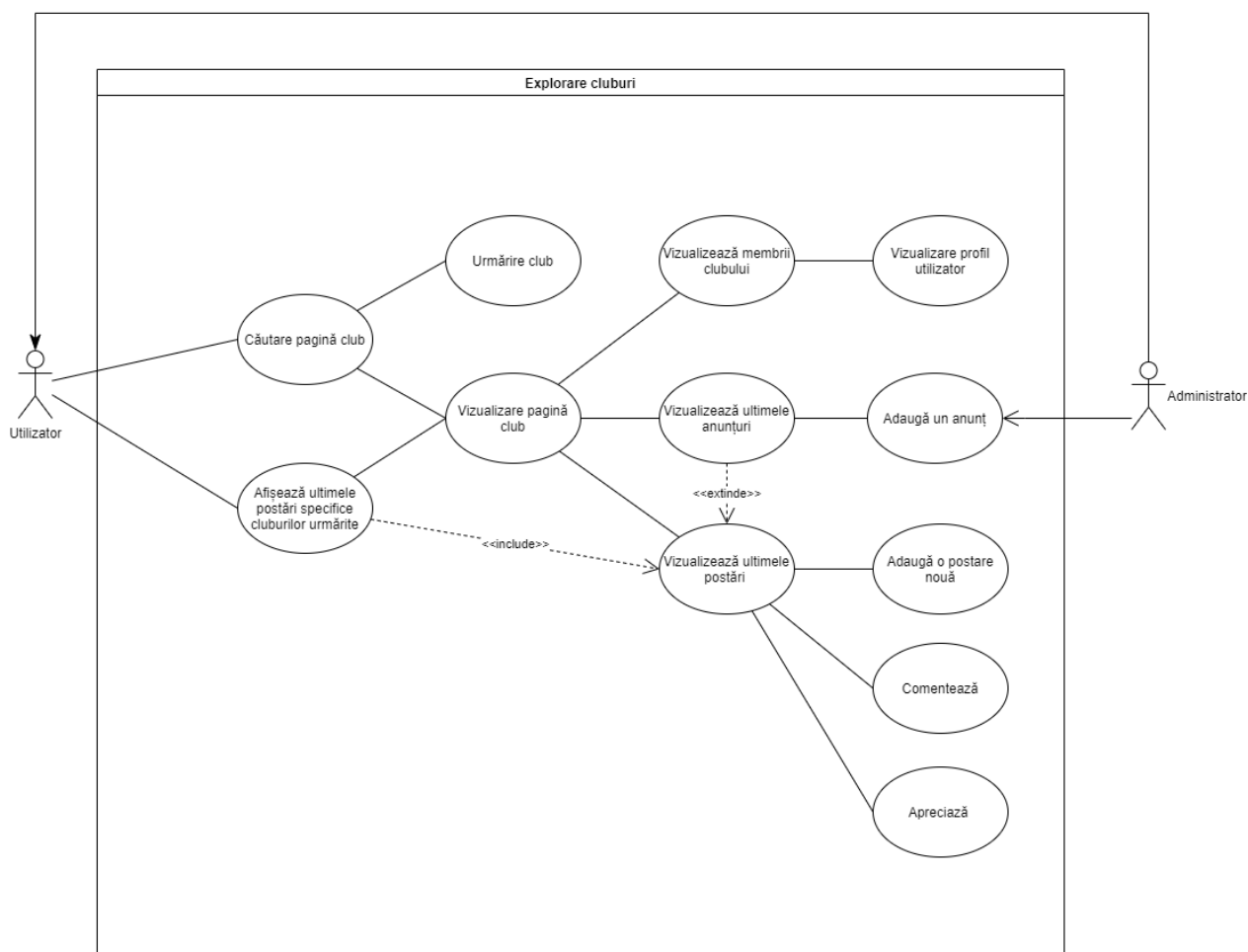


Figura 15: Scenariul de utilizare al activității “Explorare cluburi”

Acest material aduce dreptul la replică pentru ceilalți membri ai clubului, aceștia având la dispoziție posibilitatea de a aprecia și comenta la postare.

### Vizualizează ultimele postări

Corelând cu explicația de mai sus, această funcționalitate pune fiecare membru în situația exprimării părerii asupra postărilor dintr-un anumit club.

În cazul celui de-al doilea meniu, mai precis “**Vizualizare anunțuri**”, doar un administrator are permisiunea de a adăuga o postare, dreptul la replică fiind liber pentru orice utilizator.

De asemenea, pentru al treilea meniu, “**Vizualizare membri**”, după cum îi spune și numele, se pot vedea ce membri urmăresc clubul curent, având posibilitatea de a vedea profilul acestuia.

### Întâlniri auto

În Figura 16 de mai jos este prezentat scenariul de utilizare al activității “Întâlniri auto”.

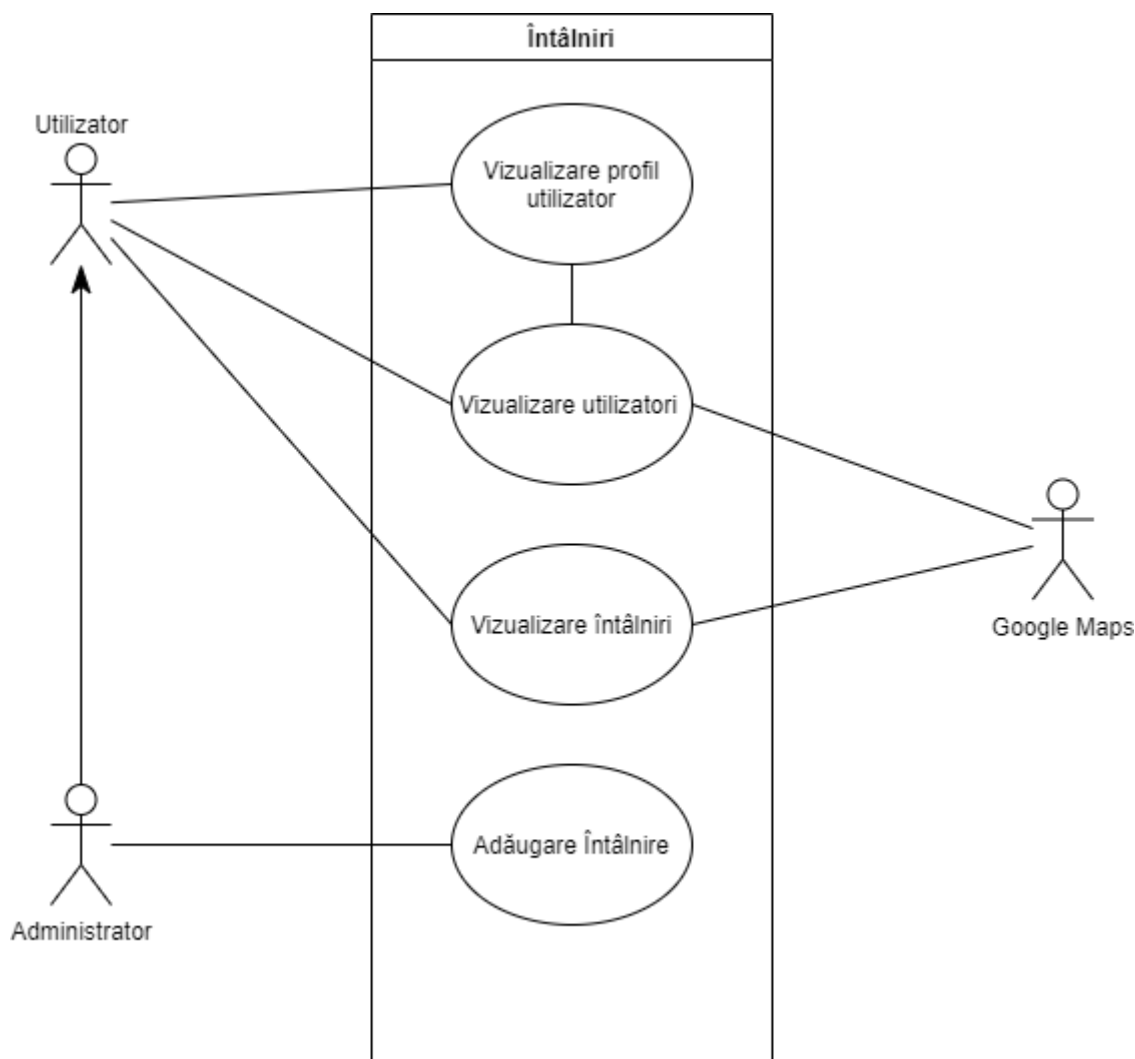


Figura 16: Scenariul de utilizare al activității “Întâlniri auto”

Un utilizator poate vedea locația celorlalți membri în timp real pe o hartă emisă de serviciul “Google Maps”. Pe lângă aceasta, se poate vizualiza profilul fiecărui utilizator vizibil pe hartă.

De asemenea, se pot vedea întâlnirile cu informațiile adiționale despre acestea. În plus, un administrator poate adăuga noi întâlniri specificând locația exactă, ora și data.

### Vizualizare profil utilizator

În Figura 17 de mai jos este reprezentat scenariul de utilizare al activității “Vizualizare profil utilizator”. Obiectivul pe care îl urmărește această activitate este acela de a facilita afișarea informațiilor personale și editarea acestora. Aici se pot vedea date precum: poză de profil, nume, prenume, biografie, autoturisme deținute, cluburi urmărite, atât de proprietar cât și de alți utilizatori. În egală măsură, aceste date pot fi editate, lucru care se face foarte intuitiv așa cum este descris în imagine.

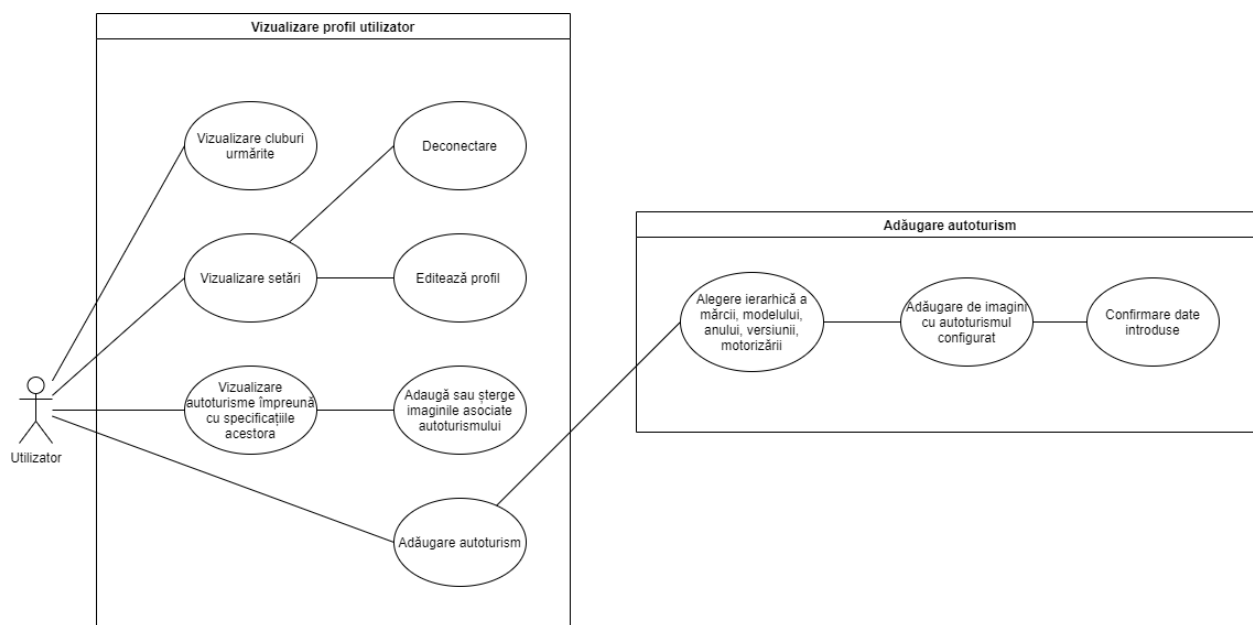


Figura 17: Scenariul de utilizare al activității “Vizualizare profil utilizator”

De asemenea, în această pagină se pot adăuga și autoturismele pe care utilizatorul le deține. Pentru aceasta se redirecționează către o nouă pagină unde se vor completa datele despre autoturism în ordine ierarhică privind marca, modelul, anul, versiunea și motorizarea, aceste date fiind prestabilite după cum am descris în capitolele anterioare. Mai apoi se pot adăuga fotografii personale cu autoturismul configurat. În caz că este omisă adăugarea de fotografii în momentul adăugării, acest lucru se poate face cu ușurință din pagina principală “Vizualizare profil utilizator”.

Tot așa, atât proprietarul cât și ceilalți utilizatori pot vedea ce cluburi auto sunt urmărite de acesta putând să fie redirecționați către pagina clubului respectivă.

## 2.4.2 Diagrame de clasă

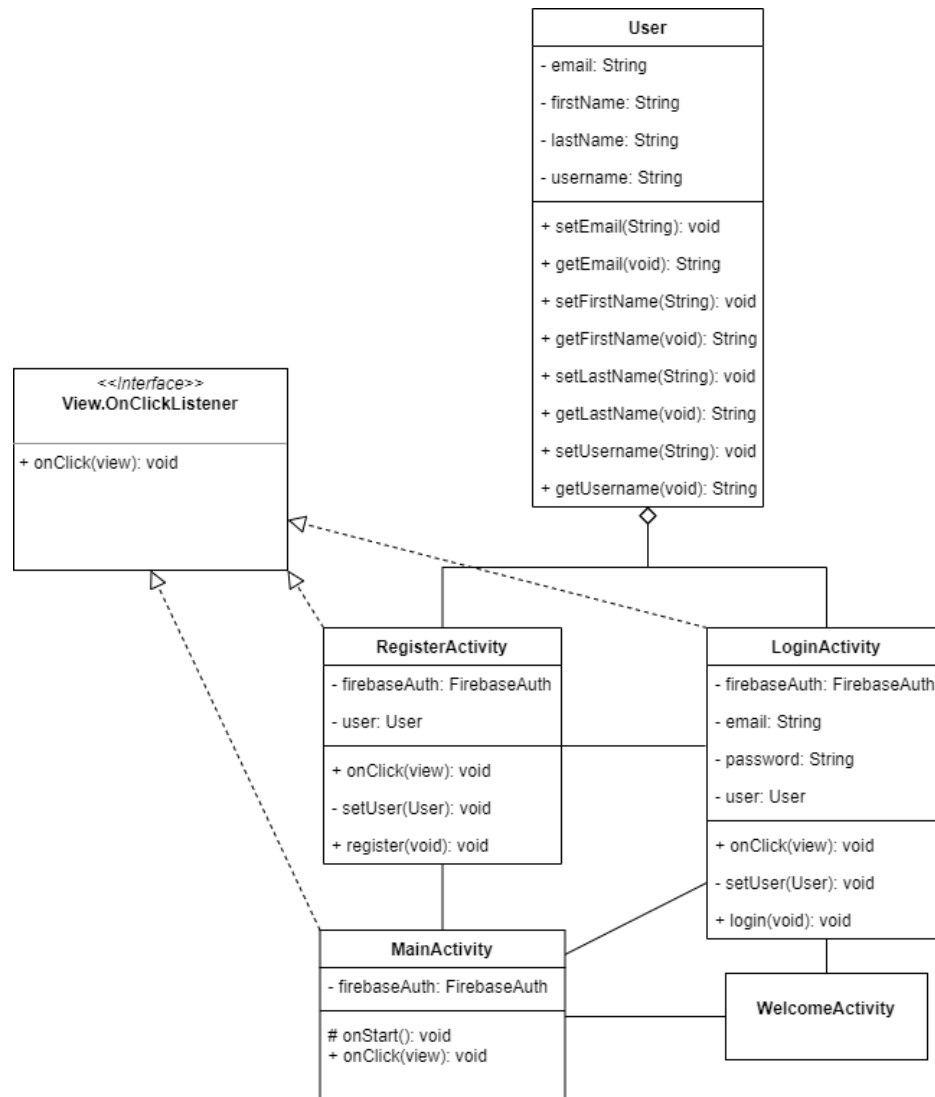


Figura 18: Diagrama de clase specifică autentificării

În continuare, voi prezenta diagramele de clase fundamentale ale aplicației, urmând mai apoi ca în capitolul următor “Implementare” să discut în amănunt despre acestea.

În Figura 18 de mai sus este descrisă diagrama de clase specifică autentificării. Activitatea principală cu care se deschide aplicația este “MainActivity”, făcându-se mai apoi o verificare dacă utilizatorul este deja autentificat, în caz afirmativ acesta este redirecționat către “WelcomeActivity” care este descris mai jos, iar în caz negativ, utilizatorul este redirecționat către “LoginActivity” sau “RegisterActivity” în funcție de alegerea acestuia.

În Figura 19 de mai jos este prezentată diagrama de clase care dictează schița care va fi folosită la implementarea structurii de bază. În linii mari, se va folosi o singură activitate

“WelcomeActivity” pentru a afișa conținutul altor fragmente, această arhitectură purtând numele de “MVP<sup>2</sup>”.

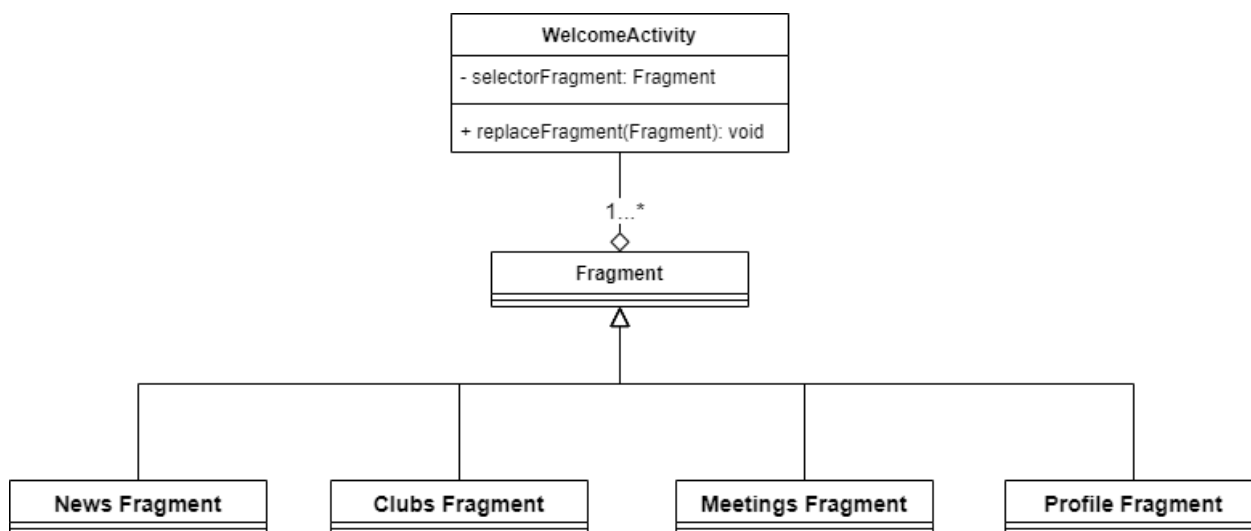


Figura 19: Diagrama de clase specifică activității de întâmpinare

“Prezentatorul” este un intermediar care face legătura între modele și vizualizarea acestora.

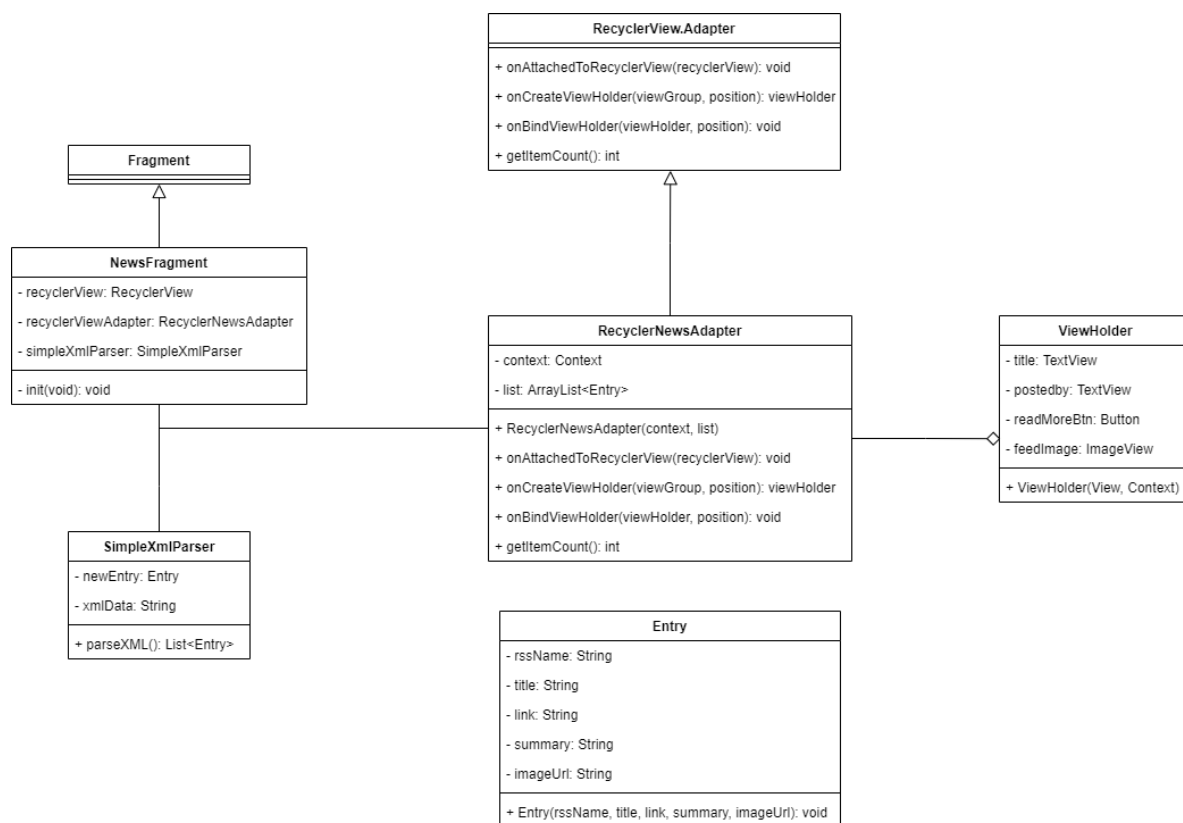


Figura 20: Diagrama de clase specifică fragmentului cu știri din domeniul auto

<sup>2</sup> Model-View-Presenter

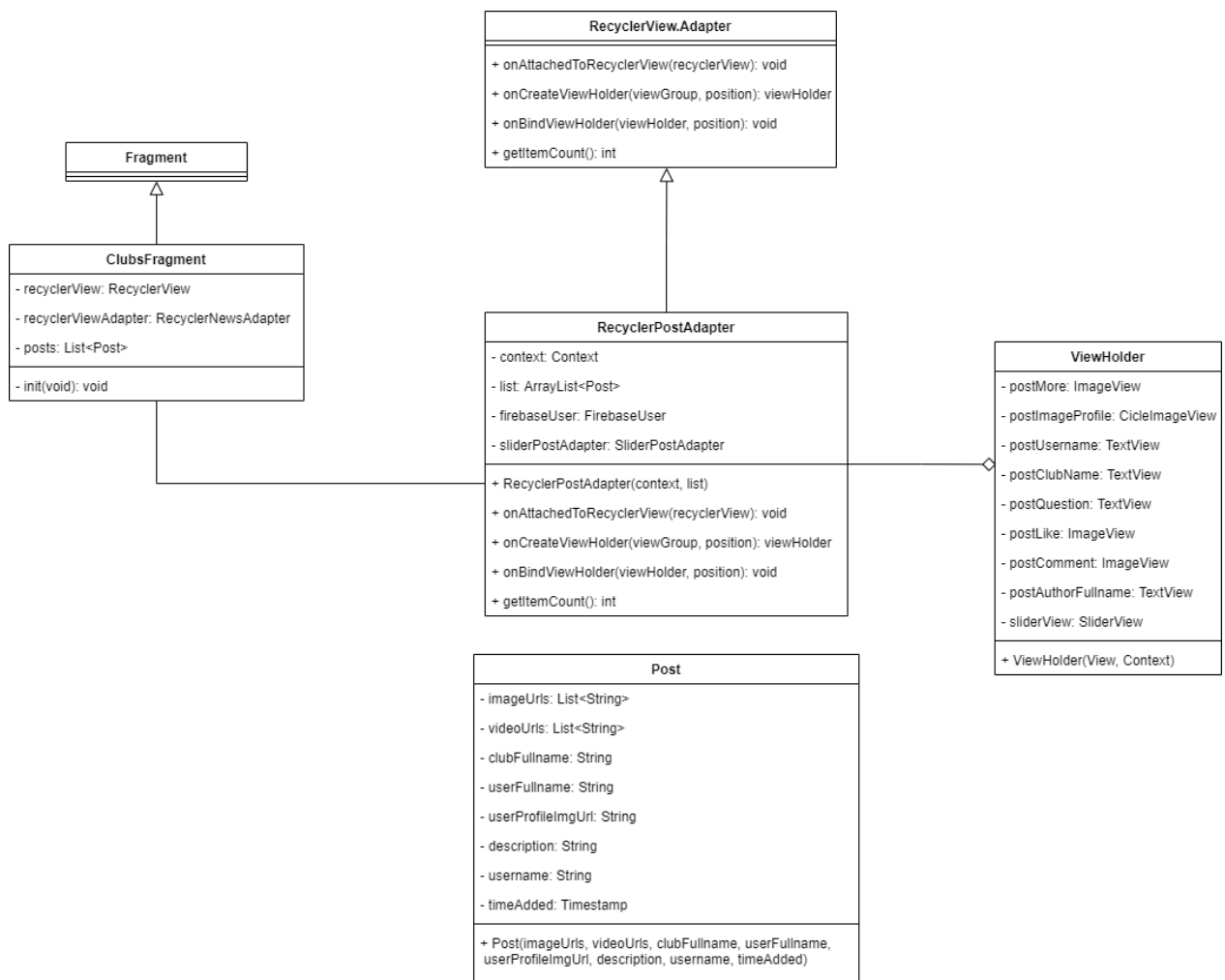


Figura 21: Diagrama de clase specifică fragmentului cu pagina de cluburi

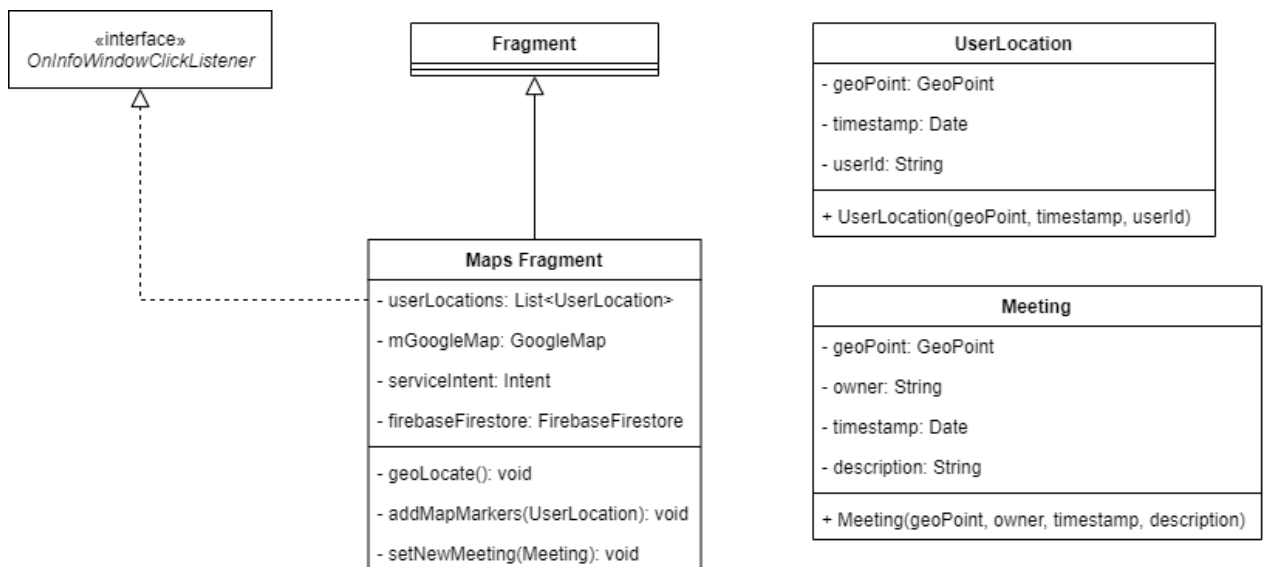


Figura 22: Diagrama de clase specifică fragmentului cu întâlniri auto

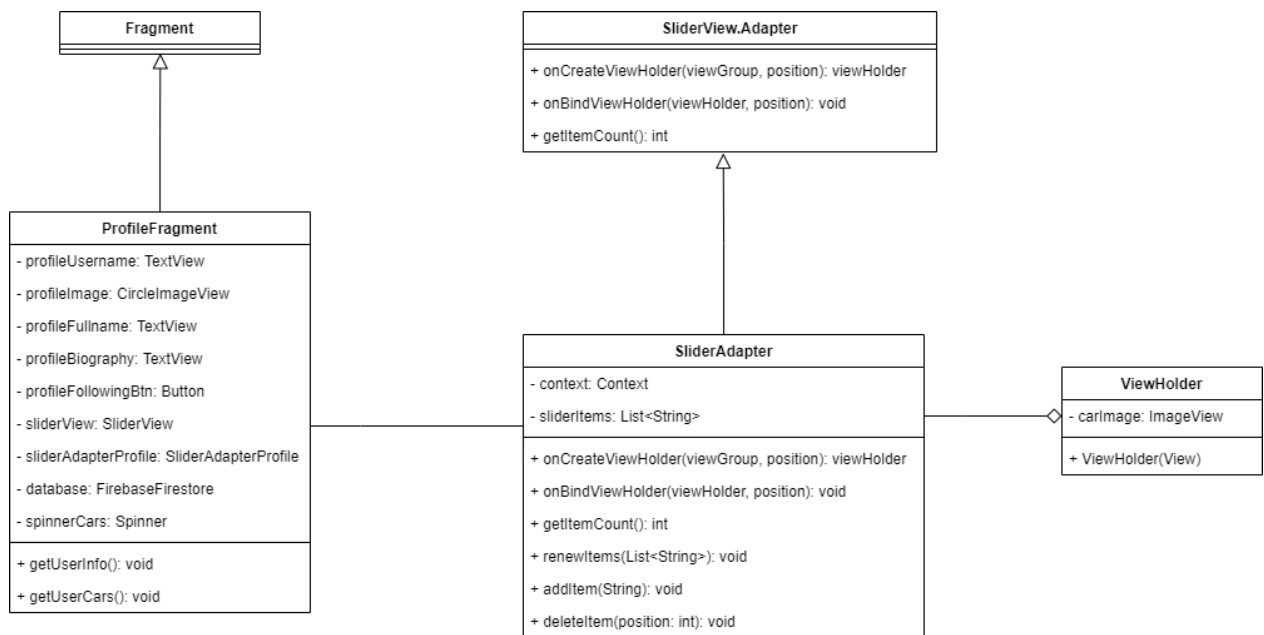


Figura 23: Diagrama de clase specifică fragmentului cu profilul utilizatorului

### 3 Implementare

Implementarea se va desfășura așa cum s-a descris în capitolul anterior “Proiectare arhitecturală”. Pentru limbajul de programare s-a optat pentru Java, deoarece am găsit diverse librării care au ajutat la ușurarea dezvoltării aplicației. Despre aceste librării o să vorbesc în capitolele următoare.

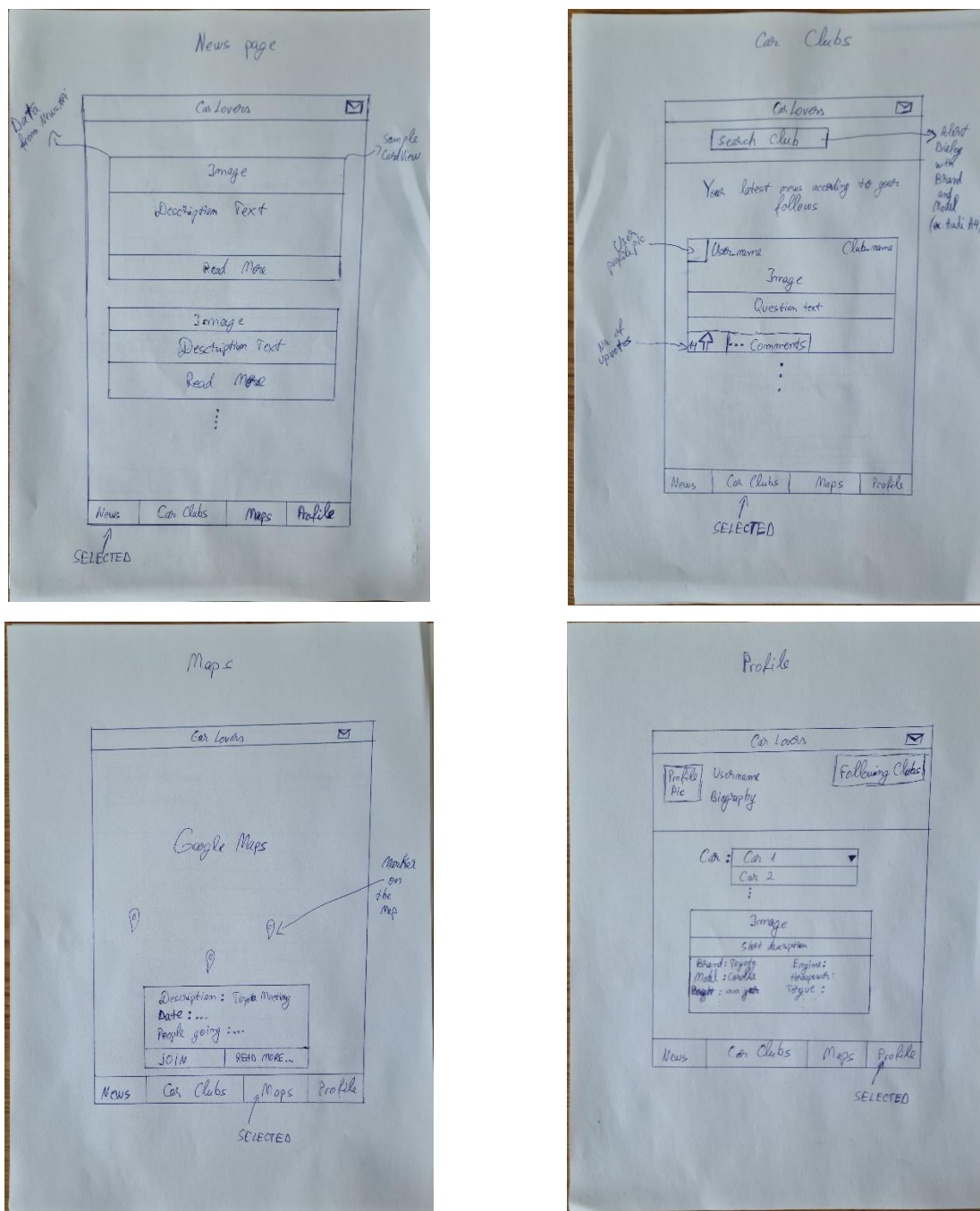


Figura 24: Schițele paginilor principale

În Figura 24 se pot observa schițele inițiale care au avut un impact mare asupra interfeței grafice, dar și asupra funcționalităților principale. După cum am precizat și în capitolul 2.4, aplicația se



poate împărți în patru module independente: Știri din domeniul auto, Cluburi auto, Întâlniri auto, Profil utilizator.

### 3.1 Activitatea principală

După cum este descris în 2.4.2, pentru implementare se va folosi o singură activitate care va prezenta conținutul celorlalte fragmente. Un Fragment reprezintă o porțiune reutilizabilă a interfeței grafice care nu poate exista de sine stătător, fiind necesar găzduirea acestuia de o activitate.

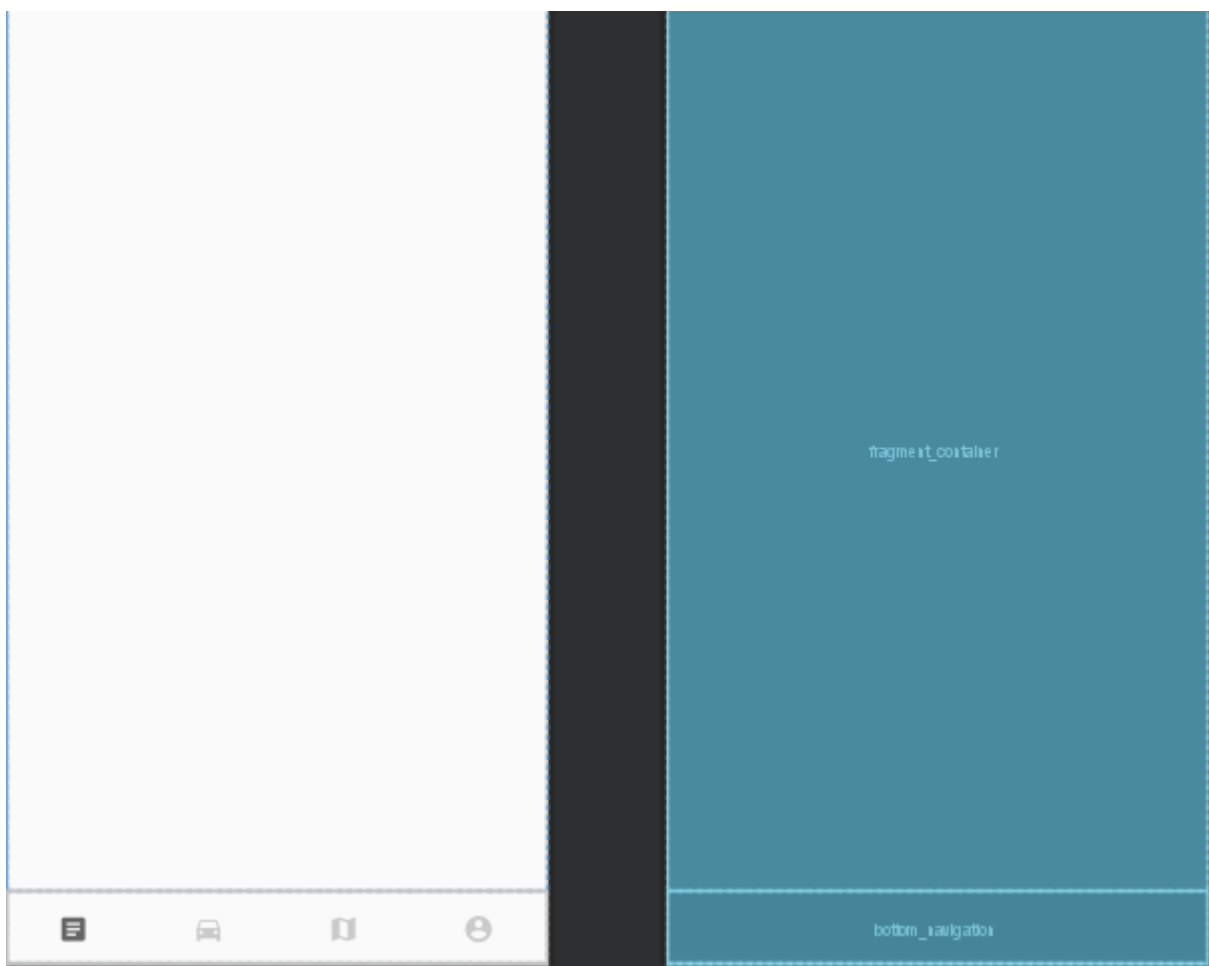


Figura 25: Interfața grafică și șablonul activității principale

În Figura 25 se poate observa că activitatea principală constă din două elemente grafice principale: un “*fragment container*” unde se va afișa conținutul celorlalte fragmente și un “*bottom navigation*” care are ca scop navigarea printre cele 4 fragmente principale.

### 3.2 Înregistrare și autentificare

The image displays two mobile application screens for 'CarLovers'. Both screens feature the app's logo at the top, which consists of the word 'CarLovers' in a bold, orange font, with the tagline 'The place you can help or get help' in a smaller, orange font below it.

The left screen is the registration page. It contains five input fields stacked vertically: 'Username', 'First Name', 'Last Name', 'Email', and 'Password'. Below these fields is a large, light gray button with the text 'REGISTER' in bold. At the bottom of the screen, there is a link that says 'Already a user? Log In'.

The right screen is the login page. It contains two input fields stacked vertically: 'Email' and 'Password'. Below these fields is a large, light gray button with the text 'LOGIN' in bold. At the bottom of the screen, there is a link that says 'Don't have an account? Register!'.

Figura 26: Interfața grafică pentru activitățile de înregistrare și autentificare

În Figura 26 de mai sus sunt reprezentate interfețele grafice specifice activităților de înregistrare și autentificare. Atât pentru înregistrare cât și pentru autentificare se folosește sistemul de autentificare Firebase. În momentul înregistrării unui nou utilizator, acesta va primi un mail de confirmare pentru a-și activa contul. În cazul în care contul nu este activat, autentificarea nu poate fi realizată. De îndată ce utilizatorul se autentifică cu succes, serviciul Firebase va stoca un “token” unic în memoria dispozitivului, acest lucru ajutând la accesările ulterioare ale aplicației, utilizatorul nefiind nevoit să se mai autentifice.

### 3.3 Știri din domeniul auto

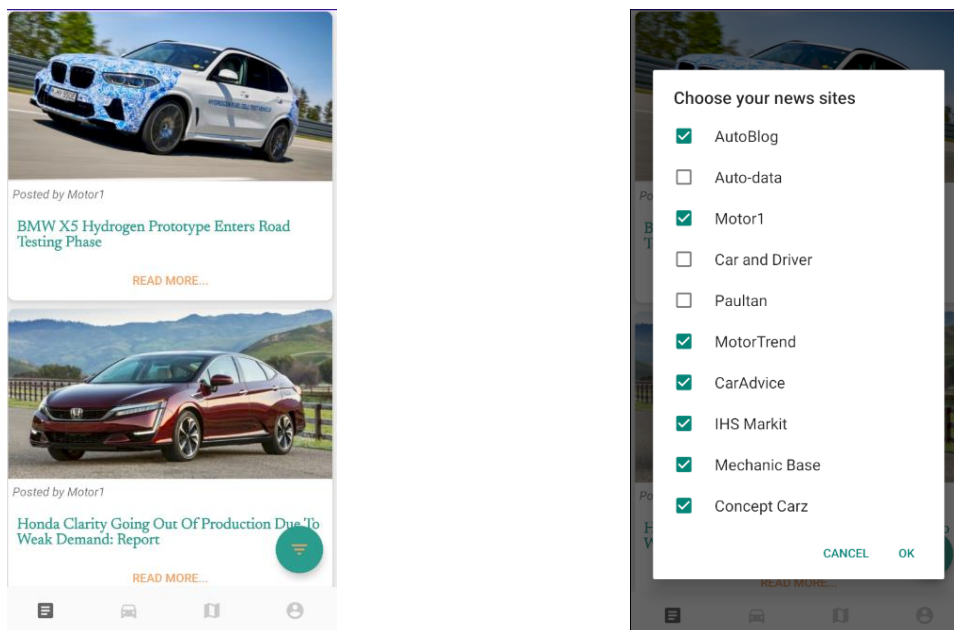


Figura 27: Interfața grafică a fragmentului “Știri din domeniul auto” și preferințele site-urilor de știri

Pentru a afișa informațiile despre știri, acestea trebuie descărcate și prelucrate în acest format. Așadar, pentru a obține acest flux de știri s-a făcut o analiză a site-urilor auto de încredere care pun la dispoziție fluxuri “RSS<sup>3</sup>”. Datele sunt descărcate în format „XML<sup>4</sup>”, fiecare producător având posibilitatea să-și personalizeze conținutul acestuia, făcând astfel dificilă prelucrarea tuturor elementelor. Astfel, s-a hotărât prelucrarea celor mai comune elemente: imagine, titlul paginii producătorului de știri, titlul noutății. De asemenea, s-a adăugat și posibilitatea redirectionării utilizatorului către pagina web de pe care este preluată știrea(butonul “Read More”). Un utilizator este liber să-și aleagă știrile a căror producători vor fi afișate, acest lucru fiind vizibil în Figura 27 în partea dreaptă. Un dezavantaj ar fi acela că producătorii sunt statici, lucru care nu permite adăugarea acestora de către utilizatori datorită faptului că fluxul “RSS” este dinamic, după cum s-a menționat și mai sus, fiind necesar prelucrarea explicită a acestuia. În plus, de la o sesiune la alta, preferințele utilizatorului sunt salvate local pentru a îmbunătăți experiența acestuia.

<sup>3</sup> Un sistem standardizat cu scopul distribuirii conținutului unei pagini web

<sup>4</sup> Extensible Markup Language – folosit pentru crearea RSS

### 3.4 Cluburi auto

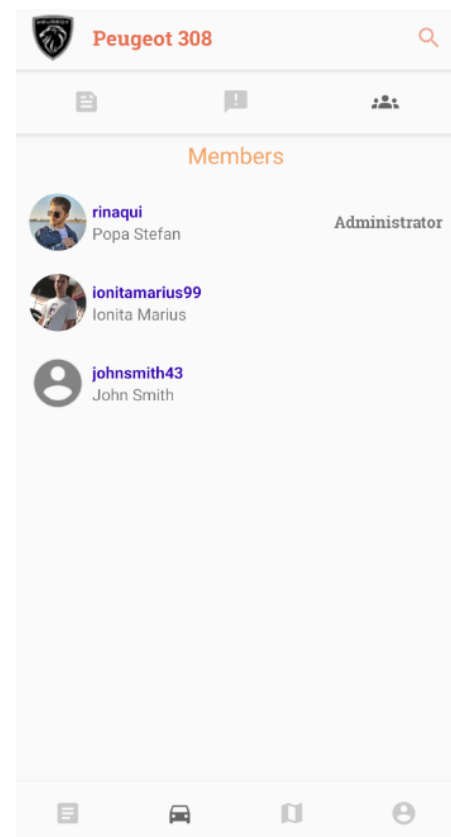
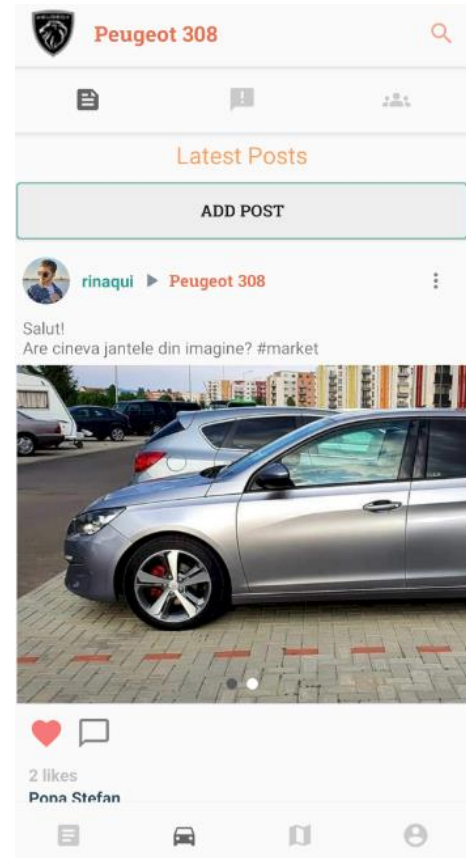
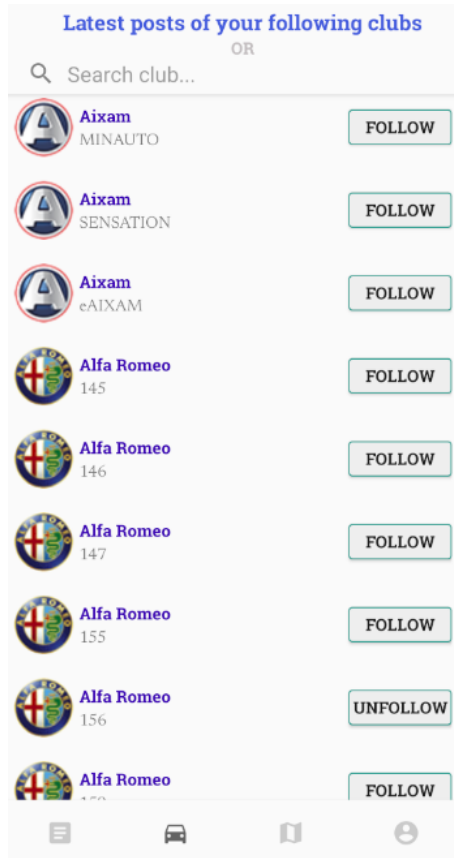




Figura 28: Interfața grafică a fragmentului “Cluburi auto” și activitățile conexe

După cum am precizat și în capitolele anterioare, cluburile auto reprezintă ideea principală a aplicației. Reamintesc faptul că există câte un club pentru fiecare model de autoturism, așa cum a fost precizat în 2.3.2.

În Figura 28 sunt surprinse următoarele funcționalități:

**Căutare și urmărire cluburi**, respectiv accesarea paginii unui club sau vizualizarea ultimelor postări specifice cluburilor urmărite de către utilizator. Se pot verifica cluburile urmărite în pagina profilului, după cum este descris mai târziu.

În cazul alegerii paginii unui club, utilizatorul va fi redirecționat către pagina specifică clubului (În Figura 28 este dat exemplul clubul auto “Peugeot 308”). Aici sunt disponibile 3 meniuri principale care se pot observa în partea de sus, ele fiind descrise prin imagini: Vizualizare ultimele postări, Anunțuri și Membri.

O **postare** poate conține text, imagini și videoclipuri. Dispunerea imaginilor și a videoclipurilor se fac în același loc, utilizatorul putând să gliseze printre ele. De asemenea, un videoclip poate fi văzut în ecran complet pentru o experiență îmbunătățită. Pentru a implementa aceste funcționalități au fost de folos următoarele biblioteci: “Android Image Slider” (smarteist,

Github, 2020) și “Exoplayer” (icbaker, 2021). În plus, o postare poate fi văzută de ceilalți membri ai clubului, aceștia având posibilitatea să comenteze sau să aprecieze.

În cazul alegerii “vizualizarea ultimelor postări specifice cluburilor”, este abstractizat conceptul de “latest posts” al unui club, afișând postări din toate cluburile urmărite de către utilizator. Diferențierea postărilor este făcută prin observarea etichetei din partea de sus a postării: nume de utilizator > club auto.

### 3.5 Întâlniri auto

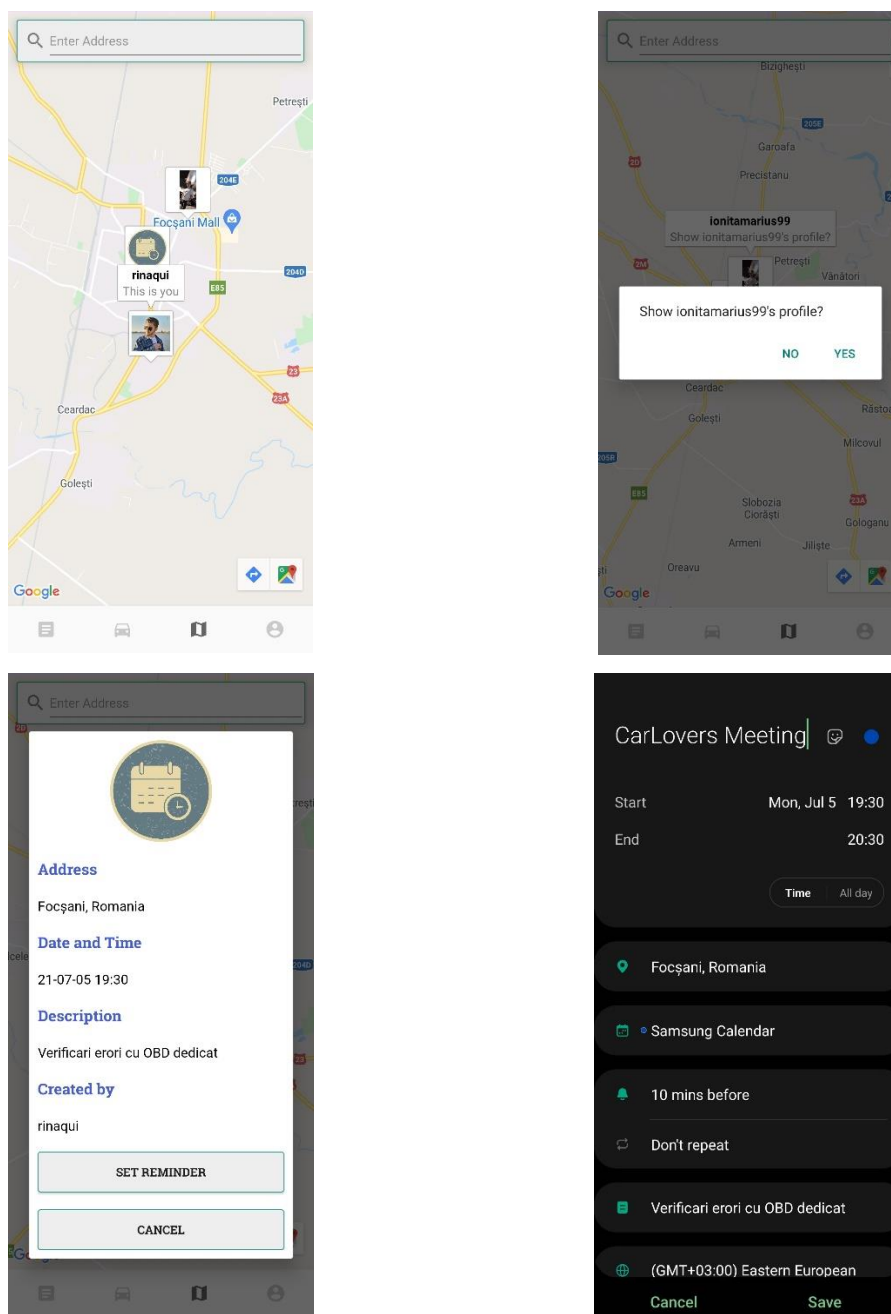
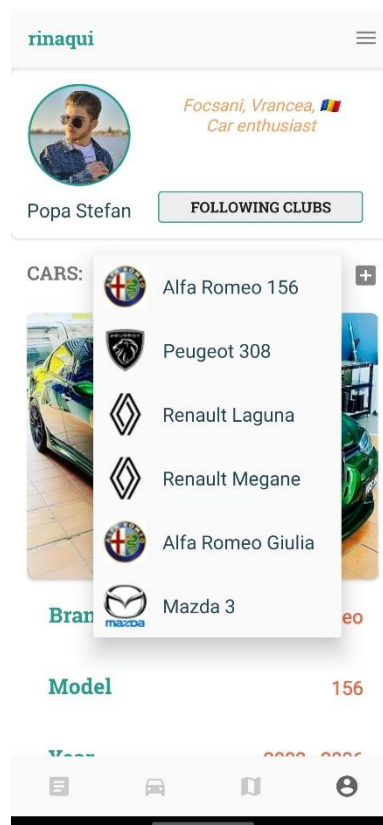


Figura 29: Interfața grafică a fragmentului “Întâlniri auto” și activitățile conexe

În Figura 29 de mai sus se pot vizualiza funcționalitățile principale ale fragmentului “Întâlniri auto” după cum urmează:

- Vizualizare și verificare utilizatori: Pe harta oferită de serviciul “Google Maps” se pot observa ceilalți utilizatori în timp real și în plus se poate verifica profilul fiecăruia în momentul selectării imaginii acestuia. Locația utilizatorului este salvată în timp real în momentul folosirii aplicației la intervale de 4 secunde.
- Vizualizare și setare memento cu întâlniri auto: Administratorii pot adăuga întâlniri auto cu diferite tematici, acestea fiind vizibile pe hartă tuturor utilizatorilor. De asemenea, un eveniment se poate salva în aplicația nativă de calendar a dispozitivului.

### 3.6 Profil utilizator



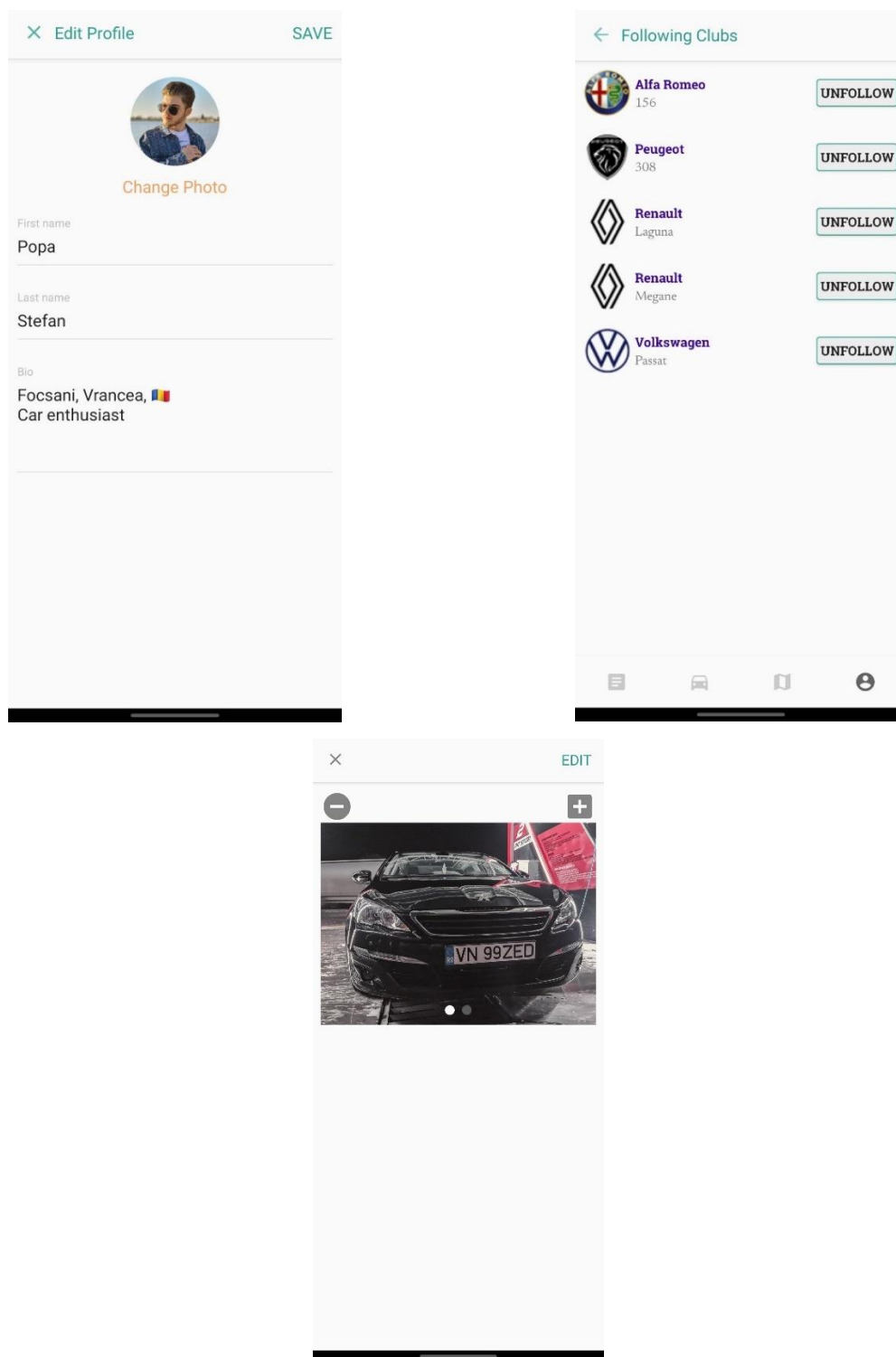


Figura 30: Interfața grafică a fragmentului “Profil utilizator” și activitățile conexe

În Figura 30 de mai sus sunt prezentate meniurile principale accesibile din fragmentul “Profil utilizator”. În pagină sunt afișate date personale despre utilizator: poză de profil, nume, prenume, biografie, autoturisme deținute. Pentru a naviga printre autoturismele acestuia se folosește un selector de tip “*spinner*”. Fiecare autoturism conține date despre marcă, model, an, versiune și



motorizare. De asemenea, se pot adăuga una sau mai multe imagini cu autoturismul, vizualizarea acestora făcându-se prin glisare, la fel ca și în cadrul postărilor. În plus, după cum am precizat și în capitolul 3.4, se pot vizualiza și paginile cluburilor auto urmărite, având posibilitatea redirectionării către acestea sau anularea urmăririi.

### 3.6.1 Adăugare de noi autoturisme

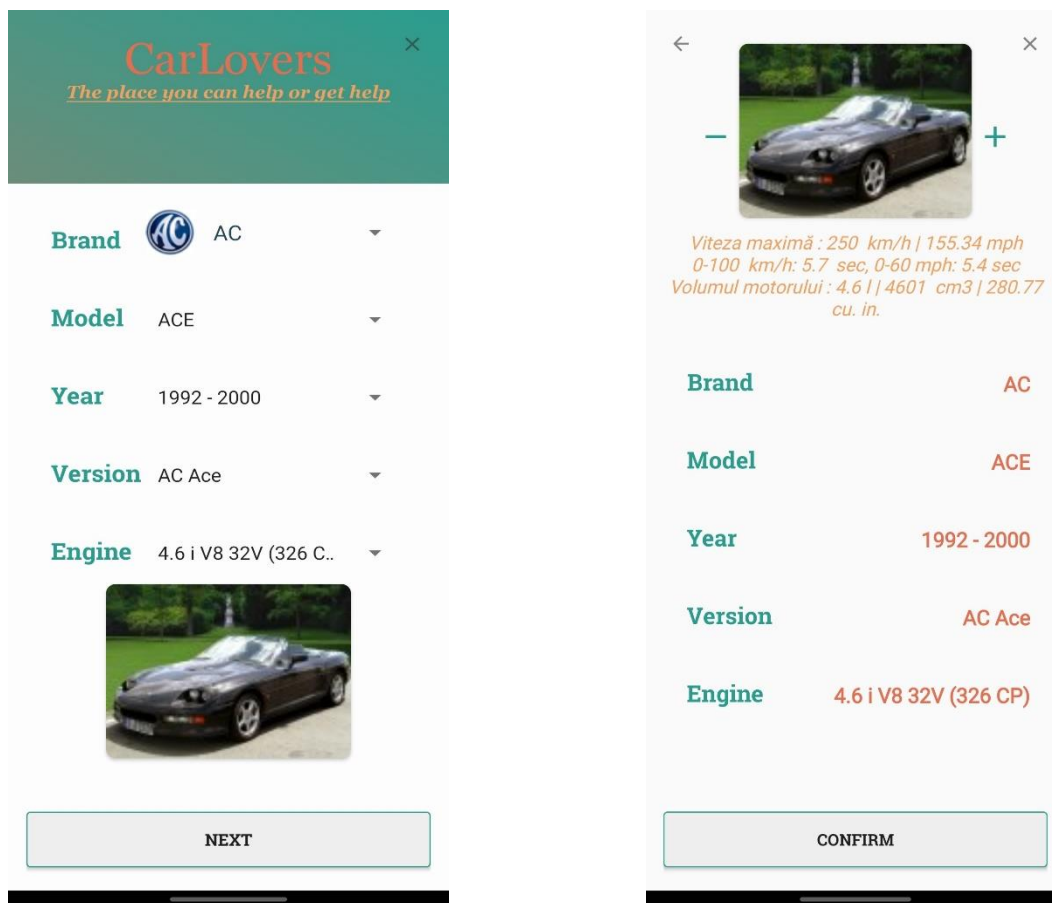


Figura 31: Interfața grafică specifică activităților de adăugare a unui autoturism

Pentru a adăuga noi autoturisme se face o selectare ierarhică așa cum se poate vedea în Figura 31 de mai sus. Se alege pe rând marca, modelul, anul, versiunea și motorizarea, iar în funcție de datele introduse se afișează o imagine de prezentare a autoturismului pentru a ajuta utilizatorul să facă o alegere corectă. În continuare, în partea de confirmare (imaginea din dreapta) sunt adăugate pe lângă informațiile anterioare și niște caracteristici generale, iar în această activitate se pot adăuga sau șterge imagini personale cu autoturismul. De asemenea, dacă utilizatorul omite etapa de adăugare de fotografii, acesta o poate face oricând și în pagina principală așa cum a fost descris mai sus.

## 4 Testare

Partea de testare este foarte importantă, deoarece există foarte multe dispozitive cu diferite rezoluții ale ecranului, lucru ce poate duce la erori atât în interfața grafică cât și în partea de funcționalitate. În continuare sunt descrise metodele de test folosite.

### 4.1 Interfața grafică

Pentru interfața grafică s-au folosit atât dispozitive reale cât și emulatoare virtuale. S-a realizat testarea tuturor elementelor grafice pe următoarele dispozitive:

- Dispozitive reale:
  - Samsung Galaxy S20 (dimensiune: 158.3mm pe diagonală, rezoluție: 3200x1440 px)
  - Huawei Mate 20 (dimensiune: 165.8mm pe diagonală, rezoluție: 1080x2244 px)
- Emulatoare virtuale
  - Pixel 3a (dimensiune: 142.2mm pe diagonală, rezoluție: 1080x2220 px)
  - Pixel XL (dimensiune: 139.7mm pe diagonală, rezoluție: 1440x2560 px)
  - Nexus 5 (dimensiune: 125.7mm pe diagonală, rezoluție: 1080x1920 px)

Interfața grafică a fost ajustată în urma testării pe dispozitivele enumerate mai sus. S-au ales aceste dispozitive pentru a putea testa funcționalitatea atât pe ecrane mici (Nexus 5), cât și ecrane mari (Samsung Galaxy S20). Cu siguranță că pentru o experiență și mai bună a utilizatorului trebuiesc făcute teste pe cât mai multe dispozitive, lucru care este costisitor atât din punct de vedere a timpului cât și a banilor. Pe partea de viitor se pot trimite formulare utilizatorilor prin care se poate cere feedback din acest punct de vedere, iar în cazul în care apar erori în acest sens sa se pot remedia pe dispozitivele raportate.

### 4.2 Funcționalitățile aplicației

Funcționalitățile aplicației s-au putut testa gradual de la o iterație la alta, datorită modului de implementare pe care l-am explicat anterior. Pentru testare s-au folosit mini scenarii prin care s-a observat cum reacționează aplicația la interacțiunea cu utilizatorul.

#### 4.2.1 Înregistrare și autentificare

Acest scenariu testează posibilitatea utilizatorului de a ajunge la ecranul principal. Pentru acesta, utilizatorul trebuie să se autentifice sistemului introducând emailul și parola, iar în cazul în care nu are aceste credențiale își poate crea unele folosind meniul de înregistrare.

1. Se afișează ecranul de încărcare:
  - a. Utilizatorul a fost deja conectat anterior: Mergi la pasul 4
  - b. Utilizatorul nu a fost deja conectat anterior: Mergi la pasul 2
2. Se afișează ecranul de înregistrare/autentificare:
  - a. Utilizatorul se autentifică cu credențialele introduse: Mergi la pasul 3
  - b. Utilizatorul își creează un cont nou: Mergi la pasul 2.a
3. Se verifică valabilitatea contului:
  - a. Utilizatorul are contul activat, mailul fiind verificat prin accesarea linkului trimis la adresa introdusă: Mergi la pasul 4
  - b. Utilizatorul nu are contul activat: Mergi la pasul 2
4. Se afișează meniul principal

#### **4.2.2 Serviciul de localizare**

Acest scenariu testează posibilitatea utilizatorului de a folosi modulul “Întâlniri auto”. Pentru acesta, utilizatorul trebuie să acorde toate permisiunile necesare aplicației, în caz contrar acesta va fi rejectat, iar aplicația se va închide.

1. Se accesează fragmentul “Întâlniri auto” din meniul principal:
  - a. Utilizatorul are toate permisiunile acordate aplicației: Mergi la pasul 4
  - b. Utilizatorul nu are permisiunile acordate aplicației: Mergi la pasul 2
2. Se afișează un dialog prin care utilizatorul trebuie să acorde permisiunile GPS:
  - a. Utilizatorul acceptă să acorde permisiunile GPS: Mergi la pasul 4
  - b. Utilizatorul refuză să acorde permisiunile GPS: Mergi la pasul 3
3. Aplicația se închide
4. Se verifică nivelul API al dispozitivului utilizatorului:
  - a. Utilizatorul deține un dispozitiv cu nivelul API mai mare decât 26: Mergi la pasul 5
  - b. Utilizatorul deține un dispozitiv cu nivelul API mai mic decât 26: Mergi la pasul 6
5. Se cer permisiunile de rulare în fundal a aplicației pentru a actualiza locația utilizatorului în timp real:
  - a. Utilizatorul acceptă să acorde permisiunile de rulare în fundal: Mergi la pasul 6
  - b. Utilizatorul nu acceptă să acorde permisiunile de rulare în fundal: Mergi la pasul 3
6. Se afișează conținutul fragmentului “Întâlniri auto”

## **Concluzii și direcții viitoare**

### **Opinia personală**

Nu este o noutate faptul că tehnologia a avansat din ce în ce mai mult în ultimii ani, iar simplele telefoane mobile a căror funcționalitate principală era de a iniția un apel telefonic sau de a trimite mesaje text au fost înlocuite de telefoanele mobile inteligente care sunt capabile să revoluționeze lumea prin caracteristicile sale (de exemplu: conexiunea la internet, GPS, realitate augmentată), păstrând de altfel o portabilitate sporită. Prin această evoluție, telefoanele mobile ne servesc din ce în ce mai mult la activitățile de zi cu zi reușind să automatizeze multe dintre acestea.

În această lucrare, am prezentat dezvoltarea unei aplicații pentru pasionații de mașini. Ideea de bază a fost să creez un mediu virtual care să fie benefic construirii și dezvoltării de comunități auto prin punerea la dispoziție a tuturor instrumentelor necesare în acest scop. În urma procesului de dezvoltare, aplicația a ajuns să aibă următoarele funcționalități principale: “Știri din domeniul auto”, “Cluburi auto”, „Întâlniri auto”. Pentru a ajunge la produsul finit a trebuit să trec printr-o serie de etape care au ajutat la construirea unui produs stabil într-un timp scurt. Aceste etape au fost planul de lucru al aplicației pornind de la proiectarea arhitecturală până la implementare și apoi testare.

Consider că am reușit să învăț lucruri noi atât pe partea de proiectare cât și implementare, dar în egală măsură am putut să-mi arăt abilitățile dobândite pe parcursul anilor de studiu.

### **Îmbunătățiri și extinderi ale aplicației pe viitor**

Această secțiune cuprinde potențiale direcții viitoare de cercetare și extindere legate de tema abordată. Pe parcursul dezvoltării aplicației, multe idei au fost luate în calcul, dar din cauza timpului a trebuit să le iau în considerare doar pe cele importante pentru a ajunge la un produs finit. În continuare, voi prezenta câteva idei de implementare pentru extinderea aplicației în viitor.

### **Publicarea aplicației**

În momentul de față aplicația este gata pentru a putea fi publicată. Datorită instrumentelor folosite pentru dezvoltarea aplicației, aceasta poate fi publicată doar pe „*Magazin Google Play*” putând fi

instalată de orice dispozitiv Android cu versiunea minimă de 7.0. De asemenea, pentru a publica produsul trebuie urmate o serie de pași și plătită o taxă de \$25.

### **Lansarea aplicației pe alte sisteme de operare**

Punctul anterior introduce problema publicării produsului publicului larg indiferent de sistemul de operare. De exemplu, pentru a publica aplicația și pe “App Store” specific sistemului de operare iOS este nevoie de crearea aplicației de la zero în Xcode, iar după cum am precizat și în capitolele introductive, acest lucru se poate realiza doar folosind un sistem de operare macOS.

### **Aplicație web**

O altă funcționalitate folositoare ar fi posibilitatea utilizatorului de a-și gestiona contul direct dintr-un navigator web, fie că acesta vrea să-și modifice date personale din cont sau pentru a accesa diferite funcționalități din aplicația mobilă. Acest lucru implică o complexitate sporită, deoarece trebuie sincronizată baza de date Firebase cât și sistemul de autentificare, iar unele funcționalități din aplicație necesită adaptarea tehnologiilor folosite în dezvoltarea mobilă la tehnologiile pentru dezvoltarea aplicației web, acest lucru fiind posibil doar prin investirea timpului în cercetarea și planificarea dezvoltării în acest sens.

### **Sistem de mesagerie și conversații**

Această funcționalitate ar trebui să fie prioritară implementării și cercetării în viitor, deoarece contribuie la îmbunătățirea experienței utilizatorilor prin faptul că oferă o comunicare mai ușoară între aceștia. De asemenea, sistemul de mesagerie ar trebui să aibă ca funcționalități și distribuirea de poze și videoclipuri, dar mai ales și de evenimente și în egală măsură de postări, acest lucru ajutând la creșterea popularității aplicației.

## Bibliografie

(2016). Bermingham, N., Prendergast, M., Boland, T., O’Rawe, M., Ryan, B. Developing Mobile Apps for Improving the Orientation Experience of First-Year Third-Level Students’. In Proceedings of 8th Annual International conference on Education and New Learning Technologies, Barcelona.

(2006). Anil Hemrajani, Agile Java Development with Spring, Hibernate and Eclipse.

(1994). Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides: Design Patterns: Elements of Reusable Object-Oriented Software (GangOfFour)

An Introduction to Agile Software Development:

[https://profs.info.uaic.ro/~adiftene/Licenta/Documentatie/Intro\\_to\\_Agile.pdf](https://profs.info.uaic.ro/~adiftene/Licenta/Documentatie/Intro_to_Agile.pdf)

(2018). Preluat de pe Wikipedia: <https://ro.wikipedia.org/wiki/SDK>

(2021). Preluat de pe StatCounter: <https://gs.statcounter.com/os-market-share/mobile/worldwide>  
icbaker.

(2021/6/14). *Github*. Preluat de pe <https://github.com/google/ExoPlayer>

(2020, 11 2). *Github*. Preluat de pe Android Image Slider: <https://github.com/smartest/Android-Image-Slider>

*Wikipedia*. Preluat de pe [https://ro.wikipedia.org/wiki/Android\\_\(sistem\\_de\\_operare\)](https://ro.wikipedia.org/wiki/Android_(sistem_de_operare))