

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

Specializarea Informatică Economică

Aplicație Android pentru recenzii și rating hotelier

LUCRARE DE LICENȚĂ

Coordonator: Mădălina ZURINI

Absolvent:

Andrei-Cosmin NEAGU

București   
2019

Cuprins

[1. Introducere 1](#_Toc9199536)

[1.1.Prezentare aplicație și motivația alegerii temei 1](#_Toc9199537)

[1.2. Sumar al tehnologiilor utilizate 2](#_Toc9199538)

[2. Analiza contextului si proiectarea aplicației Android 3](#_Toc9199539)

[2.1. Comparație cu alte aplicații software existente 3](#_Toc9199540)

[2.2 Proiectarea aplicației Android 6](#_Toc9199541)

[3. Tehnologizii utilizate 9](#_Toc9199542)

[3.1. Android 9](#_Toc9199543)

[3.2. Persistența datelor la distanță și la nivel local 14](#_Toc9199544)

[3.3. Servicii furnizate de către Google 16](#_Toc9199545)

[3.4. Librării utilizate în dezvoltarea aplicației 18](#_Toc9199546)

[4. Aplicație pentru recenzii și rating hotelier 21](#_Toc9199547)

[4.1. Cazuri de utilizare și modelare UML 21](#_Toc9199548)

[4.2. Proiectarea bazei de date 24](#_Toc9199549)

[4.3. Diagrame de procese şi colaborare în BPMN 25](#_Toc9199550)

[4.4. Manual de utilizare 26](#_Toc9199551)

[5. Concluzie 37](#_Toc9199552)

[6. Referințe 38](#_Toc9199553)

# 1. Introducere

## 1.1.Prezentare aplicație și motivația alegerii temei

Aplicațiile software create în zilele de astăzi vin deobicei ca un răspuns la cererea si nevoile utilizatorilor, iar succesul acestora pe piață este direct proporțional cu nivelul de eficiență și volumul de informații pe care aplicația respectivă o poate oferi utilizatorului.

Totodată, un factor extrem de important este ca interfața grafică a aplicației să fie concepută astfel încât aplicația să fie cât mai ușor de utilizat pentru potențialele persoane interesate de funcționalitățiile pe care le ofera aceasta.

Turismul este în continuare în plină expansiune, iar călătoritul a devenit foarte accesibil oamenilor, precum și accesul la aplicațiile mobile. Desi concurența este mare, dezvoltarea aplicațiilor mobile și actualizarea celor deja existente în acest domeniu este ofertant.

Aplicația HotelRater este un software pentru telefoanele care ruleaza pe sistemul de operare Android, iar aceasta acoperă un spectru destul de larg al necesitățiilor utilizatorilor care caută ajutor legat de rating hotelier, recenzii si nu numai, deoarece aplicația HotelRater ofera mult mai mult decât aceste două lucruri, HotelRater urmăreste să-i ofere utilizatorului un set de complet de utilități când vine vorba despre călătorit. De asemenea, interfața grafică pentru HotelRater este una intuitivă, cu elemente grafice sugestive, aceasta fiind totalmente accesibilă oricărui segment de populație.

Alegerea temei a fost într-un procent destul de mare datorată faptului că am observat această lacună în rândul aplicațiilor Android ce au o tematică hotelieră. Majoritatea aplicațiilor de acest gen includ doar rating și recenzii, sau doar una dintre acestea. Privind lucrurile din perspectiva unui om care își dorește să călătorească și să aibă la îndemâna cât mai multe informații legate de posibilele locuri unde se poate caza și alte obiective turistice din împrejurimi și nu numai, am ajuns la concluzia că o astfel de aplicație precum HotelRater și ceea ce îmi doresc să ofere lipsește din peisaj.

Cu aplicația HotelRater se ajunge la îmbinarea functionalitățiilor oferite de către majoritatea aplicațiilor de recenzii și rating hotelier deja existente pe piață, de unde rezultă o altă serie de functionalități care pot să îi ofere utilizatorului suport complet pe parcursul călătoritului, fără să mai fie necesară utilizarea altor aplicații în paralel.

Pe lângă recenzii și rating, în HotelRater a fost implementat un sistem cu ajutorul API-ului oferit de cei de la Google prin care utilizatorii pot vizualiza în interiorul aplicației hotelurile de pe o raza de zece kilometrii, într-un mod rapid, fiind de asemenea posibil să își salveze locațiile într-o listă de hoteluri favorite. În HotelRater pot fi vizualizate recenziile si ratingurile hoteluri din baza de date a aplicației, dar și din cea oferita public prin Google API.

Aplicația mai oferă si o metodă integrată de căutare a obiectivelor turistice, a restaurantelor din împrejurimi dar și din alte locații, lucru ce îi este mai la îndemână utilizatorului față de situația în care trebuie să apeleze la altă aplicație mobilă pentru a afla aceste informații. Utilizatorul poate să își salveze și notițe folositoare pe parcursul călătoritului.

HotelRater a fost conceput pentru a funcționa și când există conexiune la internet și când nu există. Utilizatorul este limitat în modul offline, deoarece nu poate accesa hărțiile, baza de date a hotelurilor și nici alte recenzii, însă poate să își vizualizeze hotelurile favorite, recenziile și ratingurile date de acesta în trecut, precum si profilul personal. Cu mici ajustări ale aplicației, aceasta poate fi utilizată în mod complet offline spre exemplu in holurile unor hoteluri pentru a oferii recenzii și rating hotelului respectiv sub forma de feedback, iar faptul că poate funcționa fără acces la internet duce la o eventuală utilizare în locuri mai izolate unde internetul nu este atât de accesibil sau conexiunea nu este stabilă.

Prin realizarea acestei aplicații se urmarește urmatorul obiectiv: utilizatorii sa aiba posibilitatea de a afla recenzii, ratinguri, statistici legate de hoteluri, pensiuni din doua surse diferite, din baza de date a aplicației, dar și cel de la Google. Totodată utilizatorii vor putea căuta detalii despre orice fel de locație de pe harta, vor putea să salveze hotelurile favorite si de asemenea vor putea vizualiza profilurile publice ale altor utilizatori care au facut recenzii.

Unicitatea acestei aplicații este datorată faptului că a urmărit să îmbine toate aceste funcționalități într-o singură aplicație. Într-adevăr, în momentul actual există un număr destul de mare de aplicații ce se ocupă cu recenziile și ratingul hotelier, însa consider ca mereu va exista loc pentru mici îmbunătătiri, precum cele pe care le-am adus în aplicația HotelRater, pentru a le oferi utilizatorilor o experiență cât mai bună. Caracteristicile aplicației vor fi prezentate mai detaliat în capitolul doi, in momentul comparației cu restul aplicațiilor.

## 1.2. Sumar al tehnologiilor utilizate

Aplicația a fost dezvoltată pentru platforma software Android prin intermediul mediului de dezvoltare integrat Android Studio SDK. În dezvoltarea aplicațiilor prin Android Studio se pot folosi limbajele de programare Java și Kotlin, pentru realizarea aplicației HotelRater a fost folosit Java. Pentru stocarea datelor au fost utilizate baze de date locale, dar și la distanță. Pentru stocarea de date locală a fost utilizat SQLite, dar și fișierele de preferință din Android, iar pentru stocarea datelor la distantă a fost utilizat Firebase. Din Firebase au fost utilizate trei module și anume: Authentication, Database si Storage.

Bineînteles, au fost folosite o serie de librării care au fost implementate pentru a realiza aplicația într-un mod cât mai modern și clar, librării utile și pentru partea de back-end, dar și cea de front-end. Câteva dintre cele utilizate pentru front-end sunt: RecyclerView, CardView, CircleImageView, Expandable-Layout, MaterialRatingBar, GraphView si MPAndroidChart care a stat la baza realizării elementelor statistice din aplicație. Cele pentru back-end sunt urmatoarele: Picasso, Retrofit, Retrofit-Gson și bineînțeles cele oferite de Google Firebase si Google Play-Services (places, maps, location). Tehnologiile folosite vor fi mult mai bine detaliate, punandu-se in evidenta si aplicabilitatea acestora in capitolul trei.

# 2. Analiza contextului si proiectarea aplicației Android

## 2.1. Comparație cu alte aplicații software existente

Voi oferi o scurtă descriere a celor mai recunoscute aplicații de pe Google Play care au tematica hotelieră și oferă servicii legate de recenzii, rating și rezervări, urmand să realizez o comparație concretă între acestea și aplicația HotelRater.

Booking.com este probabil cea mai bine cunoscută aplicație mobilă legată de rating si recenzii hoteliere, fiind cea mai apreciată aplicație de călătorie, pe care zeci de mii de oameni o utilizează să găsească oferte de cazare la hoteluri, pensiuni sau apartamente de închiriat. De asemenea, cei de la Booking.com ofera unul dintre cele mai mari API-uri pentru hoteluri. Potrivit referinței [16]această aplicație are aproximativ 29,094,365 de listări în 143,172 de locații diferite din 195 de tari la nivel global. În fiecare zi sunt înregistrate aproximativ un milion și jumătate de rezervari prin intermediul acesteia.Aplicația lor este într-adevar și cea mai bine pusă la punct la ora actuală, datorită faptului ca au o bază de date hotelieră vastă și o reputație imbatabil de bună pe piața.

Trivagoeste o aplicație mobilă ce constă într-un motor de căutare agregat, care se folosește de datele altui motor de căutare, turistic al carui principal scop sunt hotelurile. Potrivit referinței [17] această aplicație mobilă compară prețurile a peste șapte sute de mii de hoteluri, folosindu-se de API-uri precum cel al aplicației Booking.com. Această aplicație compară atât prețurile hoteliere cât și alte detalii legate de experiența de a fi cazat la un hotel. Utilizatorii incep ca și în restul aplicațiilor căutarea prin inserarea unui nume de oraș, țară sau regiune, fiind posibilă căutarea în funcție de anumite atracții turistice. Alte filtre de căutare sunt date de călatorie sau tipul preferat de cameră, existând și alte opțiuni precum accesul la internet, plajă, piscină sau wellness. Evaluările aplicației Trivago sunt deobicei bazate pe surse externe preluate prin diferite API-uri, dar și prin surse interne oferite de utilizatori, astfel ajungând la o evaluare completă.

TripAdvisoreste o aplicație software care deține și una dintre cele mai cunoscute aplicații mobile cu același nume, aceasta axându-se nu doar pe recenzii hoteliere, ci și pe recenzii ale restaurantelor, fiind oarecum diferită față de Trivago si Booking.com. TripAdvisor oferă recenzii hoteliere, pentru restaurante și la fel ca Trivago colaborează cu Booking.com și alte aplicații de nivel mai mic care totuși dispun de o bază de date relativ mare, având acces la API-ul lor pentru a oferi rezervări utilizatorilor. TripAdvisor conține și un forum interactiv în care călătorii publică diverse lucruri. Marea diferență între TripAdvisor și restul aplicațiilor este că aceasta este una din primele aplicații care a mers pe ideea conținutului generat de utilizatori.

Se poate observa, că aceste aplicații mobile și web care se afla în top în momentul de față, diferă între ele într-o anumită măsură. Un lucru este destul de evident, Booking.com fiind cea mai mare aplicație hotelieră care include și rezervări pe langa recenzii si ratinguri, deține oarecum controlul, deoarece restul aplicațiilor sunt obligate sa colaboreze cu Booking.com pentru putea efectua un serviciu complet, ceea ce implică și rezervările, nu doar recenzii și ratinguri.

Aplicația ce constituie subiectul acestei lucrări este într-o mică măsură similară cu restul aplicațiilor enumerate mai sus, implementând o serie întreaga de funcționalitați deja existente. Precum se poate vedea și în tabelul de comparație de mai jos, posibilitatea de a crea recenzii și a oferi rating hotelurilor este valabilă în cazul tuturor aplicațiilor. Însa, aplicația HotelRater, aduce ceva ce în restul aplicațiilor nu a fost adus până acum și anume informații statistice legate de ratingurile si recenziile aduse unor hoteluri specifice, dar si niște grafice cu o nuanță mai generală care nu au un impact atât de major, cele generale sunt doar pentru utilizatorii curioși.

Din păcate, la nivelul aplicației HotelRater a fost utilizată o bază de date de hoteluri mult mai mică decât cea a aplicației Booking.com, cauza fiind faptul că pentru a putea avea acces la o astfel de bază de date este necesară o colaborare contra cost. Se poate vedea că până și aplicațiile Trivago sau TripAdvisor colaborează cu Booking.com în această privință.

HotelRater în comparație cu aceste trei aplicații poate să ofere într-o anumită măsură niște funcționalități pentru utilizator in cazul în care nu există acces la internet, deoarece majoritatea datelor sunt stocate și la nivel local. Spre exemplu, spre deosebire de celelalte aplicații, în HotelRater utilizatorul poate să se autentifice dacă acesta a fost autentificat în trecut pe acelasi dispozitiv în prezența conexiunii la internet și poate ulterior să își vizualizeze hotelurile favorite, recenziile acestora, notițele salvate în aplicație, recenziile și ratingurile oferite de utilizator. În aplicațiile precum Trivago, TripAdvisor sau Booking.com nu poti face sau vedea absolut nimic fara prezența conexiunii la internet, iar în anumite situații ceea ce ofera HotelRater poate fi de ajutor.

O altă funcționalitate care nu este prezentă în celelalte aplicații este un astfel de motor de căutare integrat cu Google Maps precum cel din HotelRater. Prin intermediul acestei funcționalitați utilizatorul nu mai este obligat să se folosească de o altă aplicație pentru a afla informații legate de orice loc existent în Google Maps, indiferent de faptul că este un obiectiv turistic, un hotel sau un restaurant.

De asemenea, HotelRater oferă posibilitatea de a vedea toate hotelurile sau pensiunile de pe o rază de zece kilometrii printr-un simplu buton, iar dupa aceea utilizatorul poate salva locația împreuna cu niste detalii de bază și recenziile oferite de API-ul de la Google într-o listă pe care utilizatorul o poate accesa și în mod online și offline. Pe Booking.com sau TripAdvisor nu se întalnește așa ceva, însă pe Trivago există posibilitatea să salvezi anumite locații într-o listă de marcaje.

În HotelRater există posibilitatea de a vizualiza profilul altor persoane care au dat recenzii sau ratinguri, în profil se regăsesc anumite detalii personale pe care utilizatorul alege să le expună și numărul total de recenzii și ratinguri pe care le-a dat respectivul utilizator de-a lungul timpului. Când vine vorba de acest aspect, TripAdvisor este superior, deoarece la baza acestei aplicații avem conceptul de aplicație cu continut generat de utilizatori, TripAdvisor are caracteristicile unei aplicații de tip forum pe când alte aplicații nu au. În HotelRater, fiind o aplicație ce este fundamentată pe ideea de călătorie, turism și hoteluri, ca și aplicațiile Trivago si Booking.com, nu am implementat posibilitatea de a comunica direct cu ceilalți utilizatori, deoarece nu este necesar, spre deosebire de TripAdvisor care are un concept de bază diferit.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteriu de comparație | HotelRater | Booking.com | Trivago | TripAdvisor |
| Posibilitatea de a crea recenzii și rating | Da | Da | Da | Da |
| Vizualizarea unor statistici legate de recenzii si rating | Da | Nu | Nu | Nu |
| Bază de date vastă ce oferă posibilitatea de a realiza rezervări online pentru cazare | Nu, printr-un eventual acces la un API precum cel de la Booking.com, s-ar rectifica acest aspect | Da | Nu, aceasta foloseste API-ul de la Booking.com | Nu, aceasta foloseste API-ul de la Booking.com si alte aplicații mai mici |
| Acces offline la elemente din aplicație | Da | Nu | Nu | Nu |
| Acces la un motor de cautăre integrat cu Google Maps pentru a afla detalii legate de o locație | Da | Nu | Nu | Nu |
| Posibilitatea de a vizualiza hotelurile din apropiere pe hartă | Da | Nu | Da | Nu |
| Posibilitatea de a salva hotelurile favorite într-o lista, alături de recenziile acestora, pentru o accesare ulterioară | Da | Nu | Nu | Nu |
| Vizualizarea profilurilor altor utilizatori pentru statistici și detalii personale | Da | Nu | Nu | Da |
| Posibilitate de comunicare | Nu | Nu | Nu | Da |

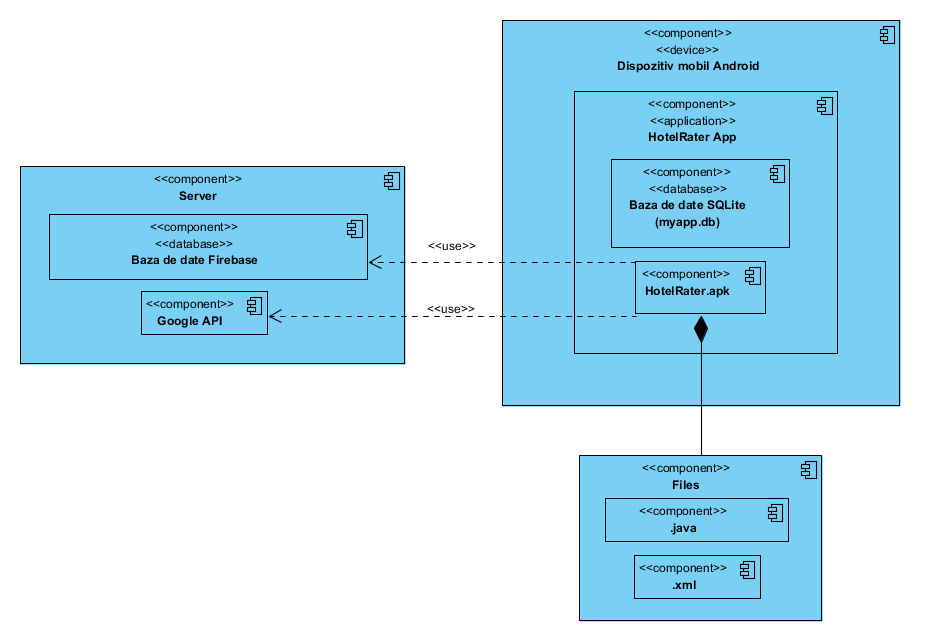
Table 1. *Comparație concisă a aplicațiilor Android deja existente*

## 2.2 Proiectarea aplicației Android

Întreaga aplicație a fost proiectată astfel încât să funcționeze doar pe dispozitivele mobile ce au sistemul de operare Android. A fost folosit limbajul Java SE, iar mediul de dezvoltare al aplicației este Android Studio.

Persistența datelor este realizată atât la nivel local cât și la distanță. Baza de date a aplicației este în principal la distanță, însă majoritatea datelor din cea la distanță sunt stocate și la nivel local in SQLite pentru a putea oferi utilizatorului acces la anumite funcționalități și în lipsa conexiunii la internet.

În figura de mai jos pot fi vizualizate componentele de bază de ale aplicației.

**

**Figura 19.** *Diagrama componentelor din aplicație*

Aplicația a fost structurată în trei pachete principale, unul pentru pentru interfață grafică, altul pentru clase ”models” și un pachet în care am introdus clasele utile, ”utils”.

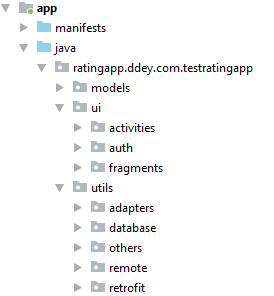
La nivelul părții de interfață grafică, am structurat pachetul în alte două subpachete în care am separat activitățiile de fragmente, fiind un proiect care momentan încă este la un nivel destul de mic, nu este necesară o structurare mai detaliată decât este.

Toate elementele grafice au fost create în XML, iar alte fișiere de tip png sau bitmap regăsite în proiect au fost create în Adobe Photoshop.

Bineînțeles, pachetul de clase ”models” este separat de întreaga aplicație. Însă mai există un pachet de clase la nivelul pachetului de retrofit din pachetul utilitar, deoarece acolo există niște modele care au fost generate prin intermediul unui convertor POJO (Plain Old Java Object) strict pentru lucrul HTTP realizat prin intermediul librăriei Retrofit.

În pachetul utilitar, avem partea de HTTP realizată prin Retrofit. Clientul Retrofit are o singură instanță în întreaga aplicație, fiind implementat ca o clasă de tipul singleton. Aceasta a fost folosită pentru lucrul cu API-ul de la Google.

Mai departe, avem un pachet remote în care este realizată o parsare a unui fișier JSON obținut într-un alt mediu de dezvoltare prin intermediul librăriei Jsoup pentru a putea extrage o serie de hoteluri de pe navigatorul web. La nivelul aplicației se face o verificare, iar în cazul în care baza de date din Firebase nu conține niciun hotel, atunci va fi realizată reintroducerea hotelurile care au fost extrase printr-un ”web scraper” Jsoup.



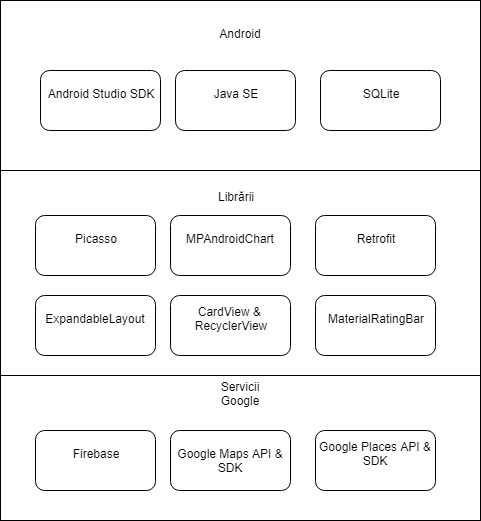
**Figura 19.** *Structura pachetelor din aplicație*

Există și un pachet dedicat adaptoarelor, care transpus informațiile din colecții de tipul ”List” în elemente de tip ”View”, precum RecyclerView sau ListView.

Pentru gestiunea operatiilor la nivelul bazei de date, avem doua clase care se ocupa de partea de operatii în SQLite, ”DatabaseHelper”, și ”FirebaseHelper” care se ocupă cu operațiile la distanță. De-a lungul dezvoltării aplicației, funcțiile celor doua clase de tip ”Helper” au fost îmbricate pentru a se obține o sincronizare între cele două baze de date. Toate constantele din întreaga aplicație sunt izolate într-o interfață cu nume specific.

Fiecărui obiect care are nevoie să fie stocat și local și la distanță i-a fost pus un atribut în care este salvat tokenul generat la nivel de Firebase pentru eventuale operații asupra acestora.

Mai jos este prezentată o diagramă în care se pot vedea principalele tehnologii și librării pe care le-am folosit în realizarea aplicației Android, elemente ce vor fi detaliate în următorul capitol.



**Figura 19.** *Diagramă cu tehnologiile utilizate în procesul de creare a aplicației HotelRater*

# 3. Tehnologizii utilizate

## 3.1. Android

Platforma de dezvoltare software a celor de la Google este probabil una dintre cele mai mari oportunități din prezent pentru dezvoltatorii de aplicații software. Google a anunțat în iarna lui 2007, aproximativ prin Noiembrie, așa zisa “Open Handset Alliance” și platforma Android, și bineînțeles prima versiune beta a fost anunțată cam tot pe atunci. În mai puțin de câteva luni, aproximativ un milion de oameni ar fi descărcat SDK-ul de pe platforma celor de la Google. [4]

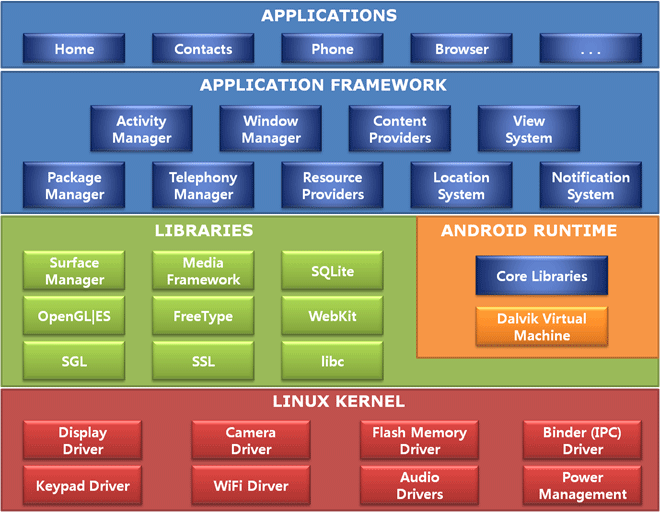
Dezvoltarea aplicatiilor Android reprezinta procesul prin care aplicațiile sunt create pentru mai multe tipuri de dispozitive ce suportă sistemul de operare Android. Conform celor spuse de bine cunoscuta companie Google, aplicațiile Android pot fi realizate în mai multe limbaje de programare, precum Java, C++ sau Kotlin, folosindu-se de kit-ul de dezvoltare software Android cunoscut sub denumirea standard Android SDK, de asemenea, este posibilă și utilizarea altor limbaje de programare însă nu sunt la fel de bine răspândite. [3]

Toate limbajele de programare care nu conțin mașina virtuală Java, JVM, spre exemplu JavaScript, C, C++ sau Assembly sunt dependente și au nevoie de ajutorul limbajelor bazate pe JVM, lucru ce poate fi cu ușurință adus în ecuație de către diferitele unelte valabile, însă acestea vor fi destul de restricționate pe planul API-ului. Majoritatea mediilor de dezvoltare Android și a uneltelor specifice continuă să evolueze încă de la primul kit de dezvoltare software lansat în anul 2008. [3]

Un JVM este o masină virtuală care îi permite calculatorului să ruleze programe din limbajul Java cât și alte programe care pot fi scrise în alte limbaje de programare dar care sunt compilate prin Java bytecode. JVM-ul este de fapt un calculator virtual sau abstract definit de o anumită specificație. Sistemul de colectare a gunoiului, “garbage-collector”, cel care se ocupă cu dezalocarea memoriei, și orice altă optimizare internă a mașinilor virtuale nu este specificată. Motivul principal pentru lipsa prezenței acestei informații este evitarea situațiilor în care pot exista constrangeri pentru dezvoltatori. Orice aplicație la baza căreia este Java poate fi rulată doar în cadrul implementării specificației abstracte ce ține de mașina virtuală Java. [5]

Kit-ul de dezvoltare software pentru Android, SDK-ul, conține un set vast de unelte utile printre care se enumeră următoarele: librării, depanator, documentație, mostre de coduri, ghiduri și nu în ultimul rând emulatorul bazat pe QEMU. În momentul de față platformele care pot rula acest kit de dezvoltare software sunt calculatoarele ce pot rula o versiune de Windows 7 sau mai nouă, Mac OS X 10.5.8 și Linux. Încă din anul 2015, undeva prin martie, SDK-ul nu mai este valabil pe platforma Android, ci poate fi regăsit în majoritatea aplicațiilor specializate pe dezvoltare. [3]

Potrivit figurii 1, se poate vedea că arhitectura sistemului Android este structurată pe niveluri, elementele depinzând de cele de pe un nivel anterior.

****

**Figura 1**. Arhitectura sistemului Android. Sursa: https://www.dev2qa.com/android-architecture-components-introduction/

Se poate vedea și în figura 1, de mai sus, faptul că în Android există patru straturi de bază. În concordanță cu referința [7] se enumeră următoarele straturi:

* Linux Kernel: Sistemul de operare Android a fost dezvoltat bazându-se pe Linux, dar a utilizat doar partea de kernel din sistemul de operare Linux, așadar tot codul scris în Java este transformat în cod C++ pentru a rula în stratul Linux kernel. Acest strat se va ocupa cu gestiunea driverelor din toate componentele fizice de nivel mic, lucru ce este descris în imagine.
* Librării Android: Acesta este al doilea strat din arhitectura sistemului Android, expunând librării bazate pe limbajul Java către dezvoltatorul de software. Toate librăriile sunt executate în Android Runtime Environment. Codul java de tip byte code este transformat în cod C++ de către mașina virtuală Dalvik iar după aceea este executat de Linux Kernel.
* Framework-ul aplicației: Acest strat se afla cu un nivel peste cel al librăriilor și furnizează aplicației niște servicii de bază pe care aplicația le poate utiliza. Fiecare serviciu în parte face referire la un concept din Android, fapt ce contribuie la ușurarea dezvoltării aplicației Android.
* Aplicația: Acesta este vârful straturilor, este rezultatul final al procesului de dezvoltare software.

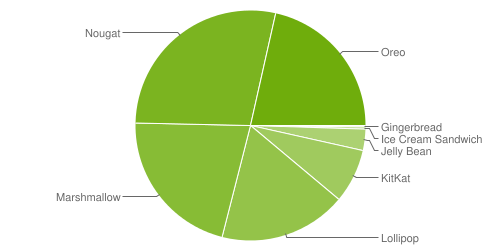
Modelul de securitate al aplicațiilor Android este specific Linux-ului, fiind un sistem de operare bazat pe kernel-ul Linux, sistemul fișierelor și regulile de securitate sunt specifice Linux. Utilizatorii ar putea să interacționeze cu dispozitivul Android folosind shell-ul Linux care este furnizat de către unealta adb. Deobicei dispozitivele menite dezvoltării software permit acces cu toate drepturile către „root”, însă la majoritatea nu este cazul. [2]

Problemele de securitate ale dispozitivele mobile există, deși sunt mai puțin vulnerabile la atacuri decât computerele. Câteva din tehnologiile mobile sunt folosite deobicei în astfel de atacuri pentru a primi sau trimite anumite date neautorizate sau fișiere ce pot cauza ulterioare probleme. Printre aceste tehnologii se enumeră telefonia (se pot realiza apeluri neautorizate ce duc la costuri suplimentare), bluetooth-ul, mesageria, rețelele wireless și NFC (abreviat de la Near Field Communication, folosit pentru eventuale plăți neautorizate). [2]

Conform referinței [6] acestea sunt câteva dintre cele mai importante caracteristici prezentate de Android la nivel software și hardware:

* La nivel de suport hardware, pot fi incluse – cameră video, GPS, accelerometru, giroscop, magnetometru, touchscreen, senzori de presiune, termometru și grafică 3D accelerată.
* Deși majoritatea aplicațiilor Android sunt scrise în Java, JVM-ul este inexistent în interiorul platformei și codul pe biți Java nu este executat. Clasele Java sunt compilate prin executabile de tipul Dalvik și rulează la runtime.
* Există o multitudine de dimensiuni de ecrane pentru display pentru această platformă și poate fi conectat la un ecran exterior prin intermediul unui cablu HDMI.
* Sunt utilizate librării pentru grafică 2D și librării pentru grafică 3D fundamentate pe OpenGL ES 2.0.
* La nivel de stocare locală este utilizat SQLite, o bază de date relațională, dar există și posibilitatea de a stoca în fisiere de preferință.
* Aplicațiile Android pot fi scrise și în HTML (Native).
* La nivel de conectivitate, Android oferă suport pentru următoarele tehnologii: EV-DO, LTE, Bluetooth și GSM/EDGE.
* Bluetooth dispune de trimiterea de contacte între telefoane, trimiterea de fișiere, accesarea agendelor telefonice de tipul (PBAP). Mai târziu, după versiunile de Android 3.1 a apărut suport pentru mouse, joystick și tastatură.
* Stocare externă – deobicei dispozitivele care rulează Android includ sloturi microSD și pot să citească carduri formatate Ext4, Ext3 sau FAT32. Stocarea formatată prin intermediul FAT32 este realizată cu ajutorul driverului furnizat de Linux Kernel.

De-a lungul timpului Android a cunoscut o serie de îmbunătățiri anuale prin care Android a ajuns la nivelul la care se află astăzi. Comparând cu rivalul din domeniu, Apple, care deține sistemul de operare iOS, Android mereu a oferit o libertate mult mai mare dezvoltatorilor software față de aceștia. Îmbunătățiriile majore sunt lansate odată pe an, alături de îmbunătățiri lunare la nivel de securitate. Alături de fiecare versiune de Android există un nume de cod pe care foarte mulți oameni îl folosesc pe postul numărului de versiune. Fiecare versiune este denumită după un anumit desert, spre exemplu prima versiune a fost denumită “Cupcake” în aprilie 2007, iar cea mai recent lansată se denumește “Pie” în august 2018. La început update-urile de sistem erau mai frecvente, urmând sa devină anuale tocmai de la versiunea “Ice Cream Sandwich” încolo.

****

**Figura 2**. Diagramă distribuție versiuni Android. Sursă: https://developer.android.com/about/dashboards

Datele ce susțin diagrama din figura 2 au fost colectate pe parcursul unei săptămâni, această perioada sfârșindu-se pe data de 26 octombrie 2018, versiunile care prezintă un procent mai mic de 1% nu au fost incluse în această diagramă.

**Tabel 2.** *Distribuția versiunilor de Android*

|  |  |  |
| --- | --- | --- |
| Nume de cod al versiunii | API | Distribuție |
| Gingerbread | 10 | 0.2% |
| Ice Cream Sandwich | 15 | 0.3% |
| Jelly Bean | 16,17,18 | 3% |
| KitKat | 19 | 7.6% |
| Lollipop | 21,22 | 17.9% |
| Marshmallow | 23 | 21.3% |
| Nougat | 24,25 | 28.2% |
| Oreo | 26,27 | 21.5% |

În urma consultării referinței [8] am realizat tabelul 2, unde se poate observa că în perioada respectivă, versiunea Nougat era mai răspândită decât restul, existând pe un procent de 28.2% de dispozitive la nivel global.

**Android Studio** este unul dintre mediile de dezvoltare Android cele mai des utilizat, existând posibilitatea de a dezvolta aplicații și în Eclipse IDE. Potrivit referinței [9] Android Studio este platforma integrate de dezvoltare oficială pentru dezvoltarea aplicațiilor Android, fiind fundamentat pe IntelliJ IDEA. Pe lângă editorul și uneltele de dezvoltare de mare excepție oferit de către cei de la IntelliJ, Android Studio conferă mai multe caracteristici care vin ca un ajutor destul de mare, accentuând productivitatea în timpul dezvoltării aplicațiilor de tip Android. Structura unui proiect realizat in Android Studio este destul de simplă, existând unul sau mai multe module cu cod sursă și fișiere de tip resursă. Aceste module pot fi:

* Modulul librăriilor
* Modulul pentru Google App Engine
* Modulul aplicației Android, format din următoarele fișiere:
  + Fișierul Manifest, conține un fișier manifest.xml de care sistemul are nevoie având informații esențiale precum componente și permisiuni.
  + Fișierul Java, conține toate fișierele ce conțin Java sau Kotlin
  + Fișierul Res, conține toate elementele de grafică de care are nevoie aplicația, spre exemplu imaginile din drawable, fonturi, culori, drawable, string-uri și layout-uri.

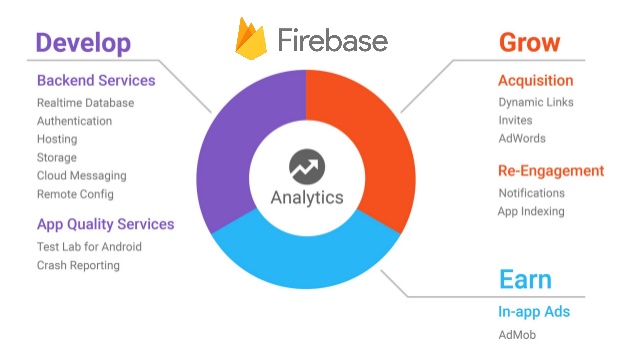
Caracteristicile cele mai importante ale mediului de dezvoltare Android Studio sunt următoarele conform referinței [9]:

* Suport integrat pentru Google Cloud Platform, ducând astfel la o integrare mai ușoară a modulelor Google Cloud Messaging și App Engine.
* Suport pentru NDK și C++
* Unelte de tipul “lint” ce se ocupă cu performanța, compatibilitatea versiuniilor, folosința anumitor module și alte tipuri de probleme
* Diverse framework-uri și unelte de testare
* Integrare cu GitHub și șabloane de coduri pentru a putea ajuta dezvoltatorul să dezvolte aplicațiile folosind fel si fel de caracteristici și pentru a îl ajuta să importe secvențe de cod din alte surse cu ușurință
* Conceptul de instant run pentru a pune în aplicare schimbările aplicației fără să mai fie necesar build-ul unui nou APK.
* Un mediu de dezvoltare unificat unde există posibilitatea dezvoltării software pentru toate dispozitivele Android
* Prezența unui emulator extrem de rapid care prezintă o mulțime de caracteristici noi
* Prezența unui sistem flexibil de tipul „build” care este bazat pe conceptul Gradle.

**Gradle Build System** este un sistem prezent în Android Studio ce constă într-un tool realizat special pentru a înființa fișierele APK, gestionând toate dependințele și furnizând o structură personalizată fișierelor. Fișierele APK, abreviat de la Android Application Package, este un format de fișier zip ce este constituit din bytecode și diferite resurse precum XML sau UI.

## 3.2. Persistența datelor la distanță și la nivel local

**Firebase** este o platformă de dezvoltare disponibilă și pentru aplicații mobile dar și pentru web, iar aceasta le furnizează dezvoltatorilor o serie întreagă de unelte și servicii care îi ajută să dezvolte aplicații de înaltă calitate, să își crească baza de date și să mărească profiturile. În anul 2011, înainte ca Firebase să devină Firebase-ul din ziua de astăzi, era un simplu startup denumit Envolve. La scurt timp după ce Firebase a fost creat ca o firmă separată drept un serviciu de back-end cu o funcționalitate real-time, în 2014 aceasta a fost cumpărată de către Google, urmând să evolueze rapid în platforma multifuncțională pe care o avem astăzi. Serviciile oferite de către Firebase sunt divizate în trei grupe, fapt văzut și în figura 3.

****

**Figura 3**. Serviciile furnizate de către Firebase. Sursă: https://hackernoon.com/introduction-to-firebase-218a23186cd7

Firebase Realtime Database este serviciul principal furnizat de către Firebase, acesta constă într-o bază de date non-SQL (nerelațională) care oferă stocare și sincronizare între utilizatori în timp real.

Această bază de date în timp real nu este nimic mai mult decât un obiect JSON de dimensiuni foarte mari asupra căruia dezvoltatorii pot face modificări în timp real. Cu un singur API, baza de date de la Firebase poate să furnizez unei aplicații valoarea curentă a unui obiect sau dată specifică și orice update asupra obiectului respectiv.

Sincronizarea în timp real ușurează accesarea datelor de către utilizatori indiferent de dispozitiv, fie că este dispozitiv mobil sau web. De asemenea, bazele de date în timp real ajută utilizatorii să comunice cu mai multă ușurință unii cu alții. Firebase furnizează pentru baza de date în timp real și un procedeu de autentificare foarte simplu și intuitiv pentru utilizator și anume Firebase Authentification.

Autentificarea prin acest modul furnizează servicii de backend, kit-uri de dezvoltare software foarte ușor de utilizat și librării de UI pentru partea de autentificare în aplicații. În mod normal, pentru a-ți crea un sistem propriu de autentificare ar dura poate chiar luni de zile, lăsând la o parte faptul ca este necesară o echipă specializată în menținerea sistemului funcțional. Cu Firebase Authentification, întregul sistem de autentificare poate fi creat în mai puțin de zece linii de cod. Firebase ofera autentificare prin mai multe metode: Google, Facebook, Twitter, numere de telefon și nu în ultimul rând metoda clasică prin e-mail și parolă. Folosirea serviciului de autentificare prin Firebase usurează întregul proces de dezvoltare al unui sistem de autentificare securizat, improvizând în același timp și experiența utilizatorului final cu aplicația.

Firebase a simplificat întregul proces de ”query” al bazei de date, realizând niște metode de interogare a bazei de date care pot fi înlănțuite. Funcțiile de ordonare din Firebase sunt următoarele: orderByKey(), orderByChild(‘child’), orderByPriority() și orderByValue().

Când vine vorba de imagini sau fișiere video ne folosim de o altă soluție oferită de către Firebase denumită Firebase Storage. Această metodă de stocare este o soluție unică pentru stocarea fișierelor, de tipul imagine sau video, generate de către user, indiferent că platforma este iOS, Android sau Web. Firebase Storage este construit în mod special pentru securitatea și scalarea aplicațiilor. Se folosește structura arborescentă specifică fișierelor din sistem pentru a structura datele stocate.

**SQLite** necesită foarte putină memorie, undeva la aproximativ 250kb, aceasta fiind valabilă pe orice dispozitiv Android. Fiecare dispozitiv ce suportă Android are de asemenea implementat din fabricație suportul pentru bazele de date SQLite, iar întregul proces este gestionat în mod automat încă de la creare, execuție până în momentul în care începe procesul de interogare a bazei de date.

Potrivit referinței [11] SQLite este o bază de date de tipul open-source, valabilă pe orice dispozitiv. Aceasta suportă caracteristicile standard oferite de către bazele de date relaționale, incluzând sintaxa SQL, tranzacțiile și alte comenzi SQL. Numele SQLite este sugestiv, deoarece această bază de date este într-adevăr o versiune mai simplificată a unei baze de date SQL, de aceea majoritatea comenzilor nici nu funcționează pe SQLite. Avantajele SQLite sunt:

* Bază de date de scală mai mică în comparație cu alte baze de date SQL
* Necesită foarte putină memorie
* Prezintă o bază de date gestionată în mod automat

Tipurile de date suportate de către SQLite sunt semnificativ mai puține decât cele din alte baze de date SQL.

* TEXT ( similar cu String) – pentru stocarea datelor de tip text
* INTEGER ( similar cu int) – pentru stocarea datelor de tip numeric
* REAL ( similar cu double) – pentru stocarea valorilor zecimale
* BLOB sau NULL – pentru stocare binară sau valori nule

**SharedPreferences** este una dintre metode prin care datele pot fi stocate în mod persistent la nivel local în Android. SharedPreferences îi oferă posibilitatea dezvoltatorului să stocheze și să extragă date cu ajutorul unei colecții oarecum mici de tipul cheie – valoare. Un obiect de tipul SharedPreferences indică locația fișierului ce conține perechiile de tipul cheie -valoare și furnizează metode simple pentru a le utiliza, fie prin citire sau scriere. Fiecare fișier de tipul SharedPreferences poate sa fie gestionat de către framework, iar acesta poate fi partajat sau ținut la nivel privat.

## 3.3. Servicii furnizate de către Google

**Google Maps** reprezintă un serviciu bazat pe web care furnizează informații geografice detaliate peste anumite regiuni și locații de pe întregul glob. Pe langă hărtiile obișnuite, Google Maps oferă și unghiuri de vizualizare aeriene și din satelit. Câteva dintre serviciile oferite de către Google Maps sunt următoarele:

* Un planificator de rute ce furnizează posibile direcții pentru șoferi, bicicliști, pietoni și transport public în cazul în care aceștia vor să se deplaseze dintr-o locație în alta.
* Google Maps pentru dispozitive mobile oferă un serviciu de tip GPS (Global Positioning System) dacă utilizatorii au acces la internet.
* Google Street View oferă utilizatorilor posibilitatea să vizualizeze și să navigheze vertical și orizontal asupra orașelor de pe glob într-un mod panoramic.
* API-ul Google face posibilă implementarea unei hărții pe dispozitive mobile și pe platforme web.

**Maps SDK** este un kit de dezvoltare software valabil pentru mai platforme printre care se regăsește și Android. Prin intermediul acestui SDK se pot adăuga hărți care sunt bazate pe date din Google Maps în interiorul aplicației. API-ul existent gestionează în mod automat tot accesul către serverele Google Maps, afișarea hărții, descărcarea datelor necesare și bineînțeles răspunsul din partea hărții în urma unor acțiuni ale utilizatorului. De asemenea, prin intermediul acestui API este posibil să fie adăugate poligoane, alte straturi grafice peste harta de bază, anumite marcatoare și să îi schimbe modul de vizualizare a hărtii pentru utilizator. Această serie de obiecte furnizează la rândul lor niște informații adiționale pentru diferitele locații de pe hartă, oferindu-i utilizatorul o interacțiune cât mai mare cu harta. Următoarele părți grafice pot fi inserate într-o hartă: [12]

* Seturi de imagini care să fie dispuse deasupra hărții de bază
* Elemente grafice de tipul bitmap dispuse în anumite locații de pe hartă
* Segmente închise de tipul poligon sau segmnete deschise de tipul unei linii
* Iconițe poziționate în anumite locații pe hartă

Accesibilitatea nu este o problemă, întregul suport pentru Android vine deja pregătit în Maps SDK.

**Google Places** este un serviciu bazat pe Google Maps, prin care sunt listate locațiile publice în urma unor căutări. În momentul în care un utilizator caută informații legate de o afacere locală va fi apelat acest API, furnizând astfel detalii despre locația respectivă.

**Places API** este serviciul care returneaza informații referitoare la locații folosindu-se request-uri de tip HTTP. Aceste locații sunt definite la nivelul acestui API drept locații geografice, puncte de interes sau unități locale. Există mai multe tipuri de request-uri valabile pentru acest API: [12]

* Place Search – returnează o listă de locații bazate pe locația utilizatorului sau pe un text de căutare.
* Place Details – returnează informații mult mai detaliate referitoare la o locație specifică, incluzând și recenziile acestora.
* Place Photos – furnizează acces la o multitudine de imagini cu locația respectivă stocate undeva în baza de date Google Places.
* Place Autocomplete – completează în mod automat adresa sau numele unei locații în timp ce utilizatorul scrie.
* Query Autocomplete – furnizează un serviciu ce ajută la anumite interogări pentru căutari geografice, returnând astfel interogări sugerate în timp ce utilizatorul scrie.
* Places IDs – fiecare locație este repezentată de către un cod de identificare unic pentru ca eventualele interogări să fie precise.
* Current Place – returnează o listă de locații în funcție de ultima locație a dispozi

Fiecare dintre serviciile Google Places de mai sus sunt accesate prin request-uri de tipul HTTP urmând să returneze un response de tipul XML sau JSON. Toate aceste request-uri trebuie să fie realizate printr-un protocol securizat “https://”, incluzând și cheie ce face referire la API. API-ul de la Google Places folosește un cod de identificare unic pentru fiecare locație în parte.

**Places SDK** reprezintă un kit de dezvoltare software ce permite dezvoltarea unor aplicații bazate pe anumite locații, care oferă un răspuns contextual locațiilor din jurul dispozitivului și nu numai. Astfel pot fi dezvoltate aplicații complexe în funcție de anumite locații care îl interesează pe utilizator, venind ca o completare asupra serviciilor geografice standard oferite de către Android, adică Android location services. Principalele puncte de intrare în SDK-ul pentru Android sunt următoarele interfețe:

* Places – oferă acces la nivel de cod către baza de date de la Google pentru o afacere locală sau orice altă locație, inclusiv locația curentă a dispozitivului.
* Autocomplete – oferă anumite widget-uri predefinite și concepute astfel încât să returneze sugestii predictive în urma anumitor căutări.

Printr-o locație se înțelege o anumită suprafață fizică care are o anumită denumire. Printre acestea se enumeră: afaceri locale, puncte de interes si orice altă locație geografică. La nivelul API-ului această locație este reprezentată de interfața „Place”. Obiectul returnat include mai multe informații, cum ar fi: locația geografica, adresa, număr de telefon, tipul locației, website, codul de indentificare al locației. [13]

## 3.4. Librării utilizate în dezvoltarea aplicației

**RecyclerView** este o variantă îmbunătățită și mai flexibilă a ListView-ului deși aceasta vine cu anumite complexități noi. Pentru a îmbunătății performanța unui ListView deobicei este folosit un model precum ViewHolder. Acest model este format dintr-o simplă clasă ce are anumite referințe către componentele vizuale în funcție de fiecare rând al ListView-ului. Ceea ce face RecyclerView față de ListView este că forțează folosirea modelului ViewHolder, care pentru ListView este opțional. Un RecyclerView necesită ca și un ListView o componentă de tip adaptor pentru a putea face legătura între date și partea grafică. RecyclerView poate fi introdus în aplicație prin accesarea suportului android pentru librării.

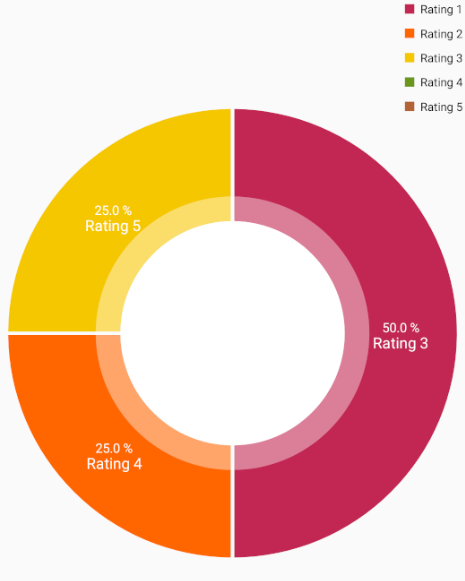
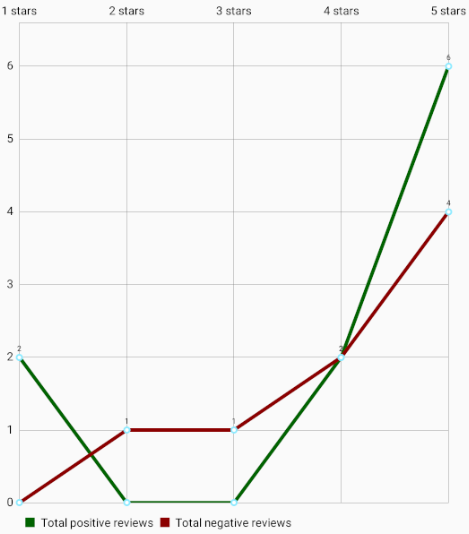
**CardView** este o componentă de UI ce conține informații în interiorul unui element grafic de forma unui patrulater. Este posibilă modificarea unghiului din colțuri, elevarea componentei și așa mai departe. Precum și RecyclerView, aceasta poate fi regăsită în suportul android pentru librării.

**ExpandableLayout** este o librărie ce îi oferă unui element grafic posibilitatea de a se extinde și de a reveni în starea inițială, existând și un set de animații predefinit de către librărie. Aceasta poate fi regăsită pe pagina GitHub a dezvoltatorului software AAkira. La nivelul aplicației a fost implementat obiectul de tipul ExpandableLinearLayout în care au fost introduse elemente grafice ce dețin informații pe care utilizatorul le poate vizualiza doar în cazul apăsării unui buton. După cum se vede în secvența de cod prezentată mai jos preluată din funcția onBindViewHolder a adaptorului pentru RecyclerView, în modelul ViewHolder al RecyclerView-ului unde sunt stocate hotelurile, am introdus un obiect de tipul ExpandableLinearLayout, acesta fiind atașat obiectului RecyclerView prin metoda setInRecyclerView(true). Prin intermediul obiectului de tipul SparseBooleanArray, expandState, am reușit să salvez starea fiecărui element/hotel din lista RecyclerView. În suprascrierea metodei onBindViewHolder, se atașează ExpandableLinearLayout de ViewHolderul din RecyclerView și este creat evenimentul de deschidere și inchidere, lucru ce poate fi vazut în următoarea secvență:

**final** Hotel model = **mList**.get(position);

viewHolder.**expandableLinearLayout**.setInRecyclerView(**true**);  
viewHolder.**expandableLinearLayout**.setExpanded(**expandState**.get(position));  
viewHolder.**expandableLinearLayout**.setListener(**new** ExpandableLayoutListenerAdapter() {  
 @Override  
 **public void** onPreOpen() {  
 changeRotate(viewHolder.**btnExpand**, 0f, 180f).start();  
 **expandState**.put(position, **true**);  
 }  
 @Override  
 **public void** onPreClose() {  
 changeRotate(viewHolder.**btnExpand**, 180f, 0f).start();  
 **expandState**.put(position, **false**);  
 }

**MPAndroidChart** reprezintă o librărie creată de către dezvoltatorul software PhilJay fiind disponibilă pe GitHub, iar aceasta funcționează pe Android API 8 sau mai mare, iar pentru includerea animațiilor este necesară folosirea unui API de peste 11. Această librărie oferă un set de instrumente predefinite complex prin care pot fi create reprezentări statistice ale datelor cu un aspect aparte. Una dintre cele mai interesante caracteristici ale acestei librării sunt animațiile care nu pot fi regăsite în alte librării și pe lângă acest lucru, ușurința cu care pot fi introduse datele. Aceste animații îmbunătățesc experiența pe care o are utilizatorul în momentul în care utilizează aplicația. Cu ajutorul acestei librării pot fi folosite mai mult de 25 de tipuri de animații și pot fi de asemenea create unele personalizate. Folosind MPAndroidChart se pot desena următoarele tiprui de grafice: grafic simplu sau grupat cu bare, grafic cu bare orizontal, grafic liniar simplu sau grupat și grafic de tipul “Pie-Chart.



**Figura 4.** LineChart și PieChart realizat cu MPAndroidChart în aplicație. Sursă: aplicație

**MaterialRatingBar** este o librărie ce îndreaptă anumite greșeli ale widget-ului RatingBar de bază, MaterialRatingBar extinde RatingBar și oferă un design mai bun care poate fi dus la fructificare doar în versiunile Android 3.0 sau mai mari. În cazul aplicației HotelRater, a fost necesar să redimensionez mai multe RatingBar-uri pentru a obține rezultatul din figura de mai jos, însă acest lucru n-ar fi fost posibil fără MaterialRatingBar. Librăria aceasta oferă scalabilitate widget-urilor, un aspect mai pretențios oferind dezvoltatorului libertate de modificare mult mai mare asupra elementului grafic RatingBar.

**Picasso** reprezintă o librărie pentru descărcarea, încărcarea și procesarea imaginilor fiind dezvoltată și actualizată de către compania Square Inc. Popularitatea acestei librării este datorată faptului că este foarte simplu să o utilizezi, fiind necesară scriere doar unei linii de cod pentru majoritatea caracteristicilor pe care le oferă. Pentru a se folosi această librărie, trebuie ca și în alte cazuri să fie introdusă dependința în fișierul build gradle. Picasso are mai multe funcții:[14]

* Scalarea și redimensionarea imaginilor
* Transformări complexe și rotiri ale imaginilor folosindu-se un minim de memorie în astfel de operații
* Posibilitatea de a pune o imagine pe post de eroare și pe post de înlocuitor când de exemplu nu există conexiune la internet și imaginea nu poate fi încărcată
* Caching la nivel de disk si memorie realizat în mod automat
* Oferă suport pentru descărcări realizate în paralel și oprirea unor request-uri

Picasso.*with*(**mActivity**)  
 .load(model.getImageUrl())  
 .placeholder(R.drawable.***ic\_image\_black\_24dp***)  
 .error(R.drawable.***ic\_error\_black\_24dp***)  
 .into(viewHolder.**ivHotel**);

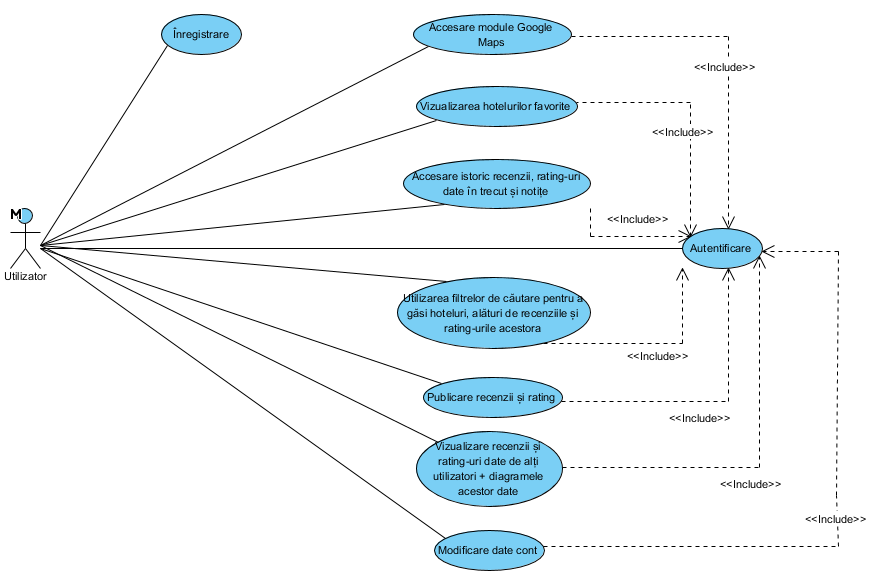
**Retrofit** este o librărie ce furnizează un client REST de tipul type-safe, adică compilatorul validează tipurile de date în timpul compilării, fiind valabil pentru Java și Android. Scopul acestei librării este de a ușura relația dintre o aplicație mobilă și serviciile web de tipul RESTful. Retrofit 2 are mult mai multe caracteristici decât versiunile anterioare dar are și API-ul intern modifica. A fost conceput cu OkHttp ca strat de bază pentru partea de rețelistică. Retrofit serializează într-un mod automat orice răspuns din partea serverului folosindu-se de POJO, abreviat de la Plain Old Java Object, ce trebuie să fie definit în cazul unor structuri de tipul JSON mai complexe.

Procesul de serializare al unui JSON necesită un „converter” pentru a îl aduce în primul rând la forma unui Gson. Dependința OkHttp vine la pachet cu cea de la Retrofit 2, nefiind necesară includerea sa separată.În următoarea secvență de cod se pot vedea adnotările HTTP și metodele GET pentru preluarea de date din Google API prin intermediul Retrofit 2, care urmează să producă un „Callback” ce va aduce prin metoda „onResponse” rezultatul metodei GET. [15]

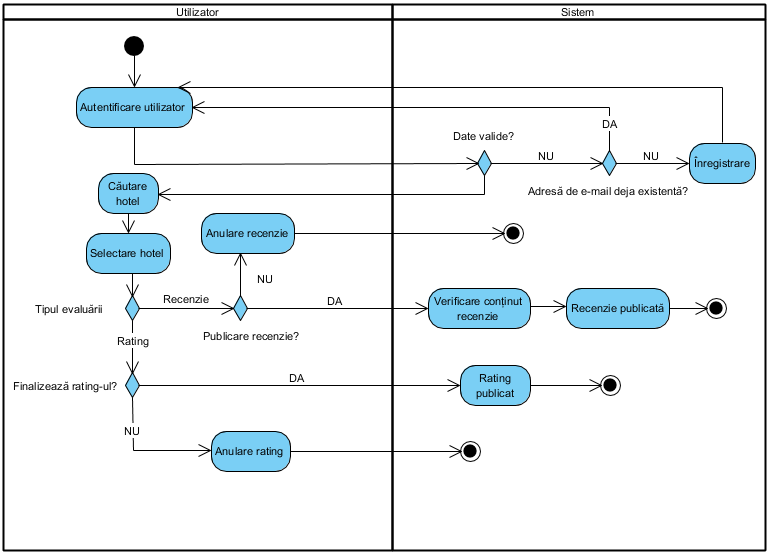
**public interface** API {  
 @GET  
 Call<MyPlaces> getNearbyPlaces(@Url String url);  
  
 @GET  
 Call<PlaceDetail> getDetailPlace(@Url String url);  
}

# 4. Aplicație pentru recenzii și rating hotelier

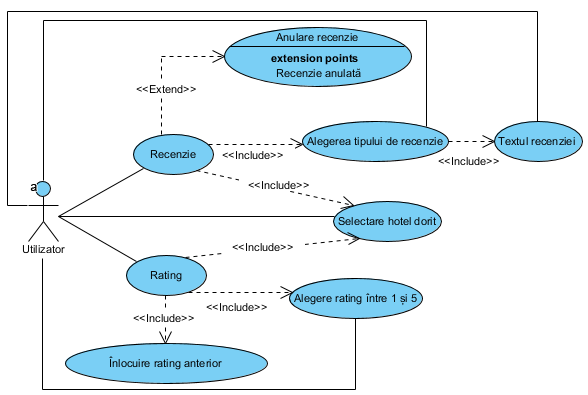
## 4.1. Cazuri de utilizare și modelare UML



**Figura 5.** *Diagramă pentru cazul de utilizare general*

****

**Figura 6.** *Diagramă de activitate pentru oferirea ratingului și publicarea recenziilor*

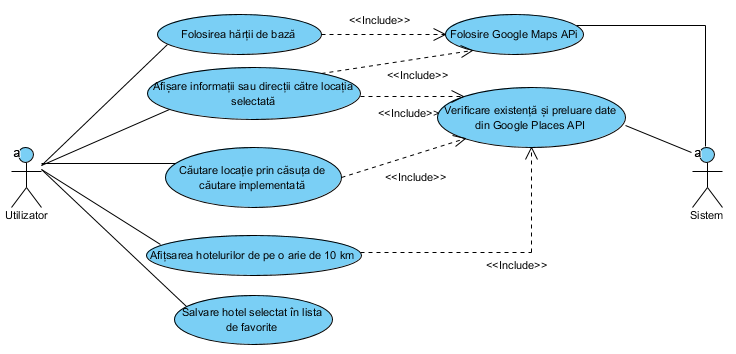
**

**Figura 7.** *Diagramă pentru cazul de utilizare „Publicarea recenziilor și a rating-urilor”*

|  |  |
| --- | --- |
| Element al cazului de utilizare | Descriere |
| Cod | CU01 |
| Stare | Schiță |
| Scop | Publicare unor recenzii sau oferirea de rating pentru un anumit hotel |
| Nume | Publicarea recenziilor și a rating-urilor |
| Actor principal | Utilizatorul aplicației |
| Descriere | În urma selecției unui hotel, utilizatorul poate să scrie o recenzie despre acesta sau să-i ofere un rating |
| Precondiții | Utilizatorul trebuie să fie autentificat și să existe conexiune la internet și baza de date |
| Postcondiții | Prezenț unei noi recenzii și a unui rating actualizat. |
| Declanșator | Dorința de a împărtăși experiența unui hotel cu alții. |
| Flux de bază | 1. Utilizatorul selectează hotelul dorit  2. Alege să ofere o recenzie sau rating  3. Scrierea unei recenzii complete plus tipul acesteia  4. Oferirea de rating |
| Fluxuri alternative | - |
| Relații | - |
| Frecvența utilizării | Des |
| Reguli ale afacerii | - |

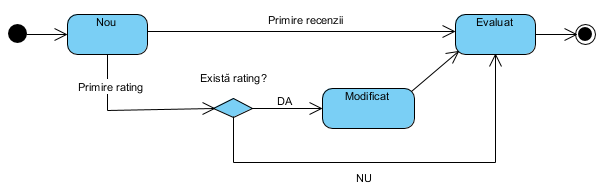
**Tabel 5**: *Descrierea sub formă de șablon a cazului de utilizare „Publicarea recenziilor și a rating-urilor”*

|  |  |
| --- | --- |
| Element al cazului de utilizare | Descriere |
| Cod | CU02 |
| Stare | Schiță |
| Scop | Aflarea unor informații cu ajutorul hărții |
| Nume | Accesare module Google Maps |
| Actor principal | Utilizatorul |
| Descriere | Utilizatorul se folosește de hartă pentru a afla informații despre diverse locații sau hoteluri |
| Precondiții | Utilizatorul trebuie să fie autentificat și să existe conexiune la internet. |
| Postcondiții | - |
| Declanșator | Utilizatorul dorește să utilizeze funcționalitășiile oferite de Google Maps |
| Flux de bază | 1. Utilizatorul accesează harta  2. Căutarea unei locații după nume și aflarea unor informații de bază legate de aceasta  3. Găsirea hotelurilor de pe o arie de 10 km  4. Salvarea anumitor hoteluri într-o listă de hoteluri favorite și posibilitatea de a afla direcții sau rezerva o camera prin intermediul altor aplicații |
| Fluxuri alternative | - |
| Relații | - |
| Frecvența utilizării | Des |
| Reguli ale afacerii | - |



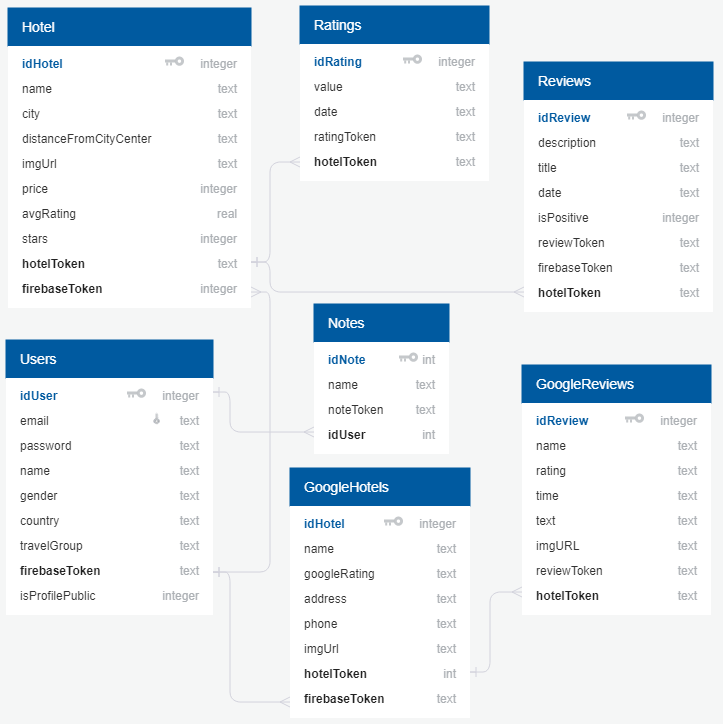
**Figura 8.** *Diagramă pentru cazul de utilizare „Accesare module Google Maps”*

**Tabel 5**: *Descrierea sub formă de șablon a cazului de utilizare „Accesare module Maps”*

****

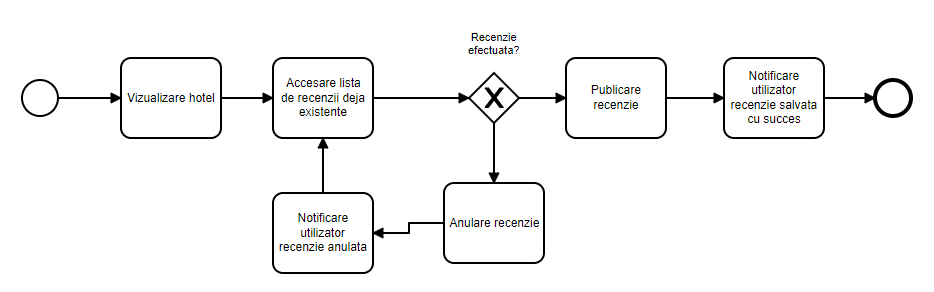
**Figura 9.** *Diagramă de stare pentru „HOTEL”*

## 4.2. Proiectarea bazei de date

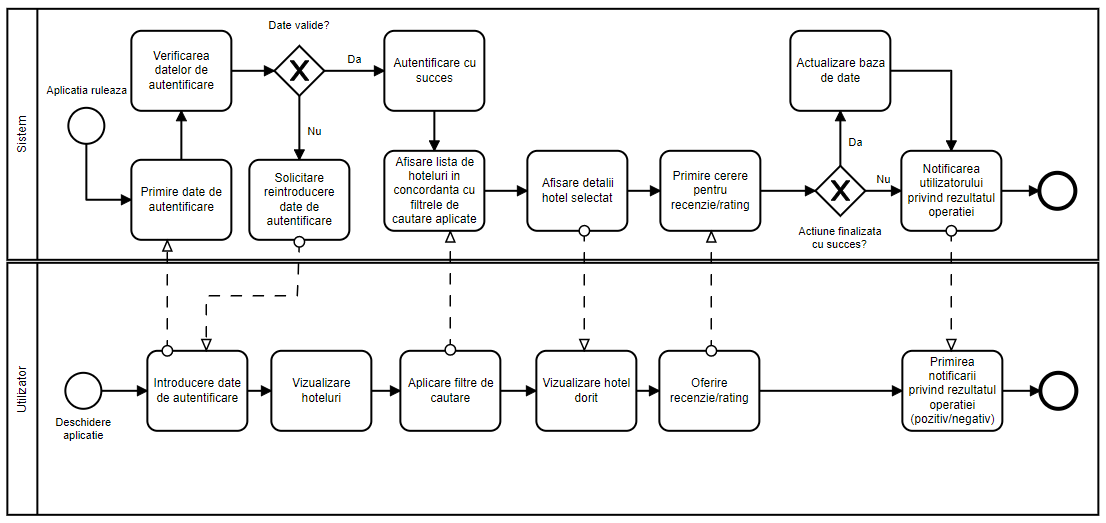
****

**Figura 10.** *Diagrama bazei de date SQLite a aplicației*

## 4.3. Diagrame de procese şi colaborare în BPMN



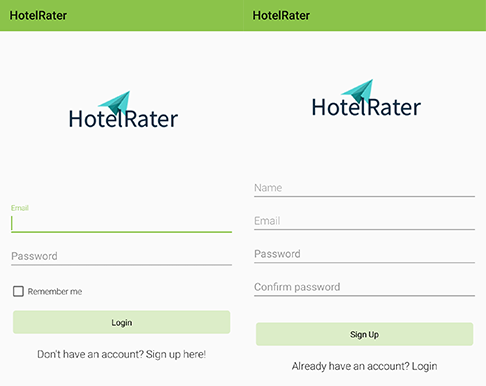
**Figura 15.** *Diagramă de proces BPMN pentru publicarea unei recenzii*

**

**Figura 16.** *Diagramă de colaborare BPMN pentru proces de desfasurare al aplicatiei, incluzand autentificarea si oferirea de recenzii si rating*

## 4.4. Manual de utilizare

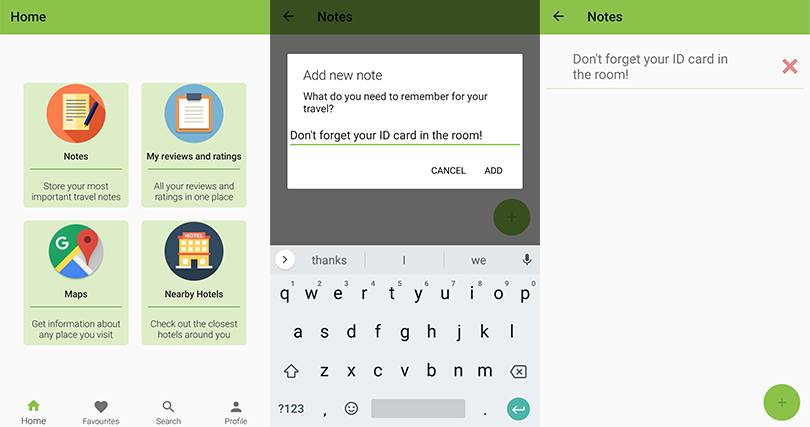
În acest subcapitol va fi prezentată aplicația într-un mod concis, prezentând fiecare funcționalitate a aplicației și orice element grafic cu care poate să interacționeze un eventual utilizator. HotelRater are o interfață foarte intuitivă fiind foarte accesibilă oricui chiar și în lipsa unui manual de utilizare.

****

**Figura 11.** Activitățiile de autentificare și înregistrare.

Autentificarea este primul ecran pe care îl poate vedea un utilizator în urma descărcării și instalării aplicației. În cazul în care utilizatorul folosește pentru prima oară aplicația este necesar să aibă conexiune la internet pentru a putea finaliza înregistrarea. Pentru a ajunge la ecranul de înregistrare trebuie apăsat textul sugestiv din partea inferioară a figurii 6. La nivelul activității de înregistrare, utilizatorul trebuie sa introducă toate datele corect, fiind prezente niște validări ale câmpurilor care trebuie să fie respectate. Se poate observa căsuța care poate fi marcată, de tipul „Checkbox”, care salvează utilizatorul autentificat, iar data viitoare când acesta va accesa aplicația nu va mai fi nevoit să introducă datele de autentificare, ci va fi redirecționat direct către ecranul principal.

Odată ce utilizator reușește să se înregistreze și se conectează pentru prima oară pe dispozitiv va putea ulterior să acceseze contul și fara prezența internetului, însă nu va putea experimenta absolut tot ce poate oferi HotelRater. Ca în majoritatea aplicațiilor, adresa de e-mail cu care se înregistrează utilizatorul trebuie să fie în mod obligatoriu unică.



**Figura 12.** *Fragmentul Home alături de desfășurarea activității Notes în care sunt salvate notițele utilizatorului*

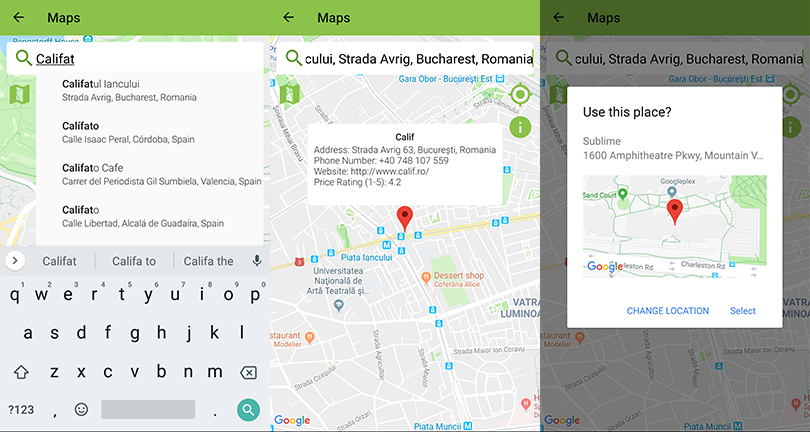
Odată ce utilizatorul este autentificat, va ajunge la ecranul ce afișează o bară de navigație care este ataștă de partea inferioară a ecranului, conform figurii 7. Această bară de navigație este un meniu prin care utilizatorul poate să navigheze între componentele principale ale aplicației.

În fragmentul „Home”, începând din stânga sus, putem să intrăm în notițele utilizatorului prin iconița sugestivă „Notes”. Aici utilizatorul poate să își vizualizeze notițele create în trecut chiar și în cazul în care nu are acces la internet, însă nu este posibilă ștergerea acestora fără accesul la internet. Crearea notițelor în schimb poate fi făcută și în cazul în care nu există conexiune la internet.

Pentru a se crea o notiță, utilizatorul trebuie să apese butonul sugestiv din dreapta jos a ecranului, urmând să apară o fereastră de tipul „Dialog”, în care trebuie inserat textul. Mai departe pentru salvare se apasă butonul „Add” și pentru oprirea operațiunii „Cancel”.

Tot în fragmentul „Home”, mai avem încă trei alegeri, „My reviews and ratings”, „Maps” și „Nearby Hotels”. „My reviews and ratings” ca și „Notes” ne oferă accesul la informație fără să fie necesară conexiunea la internet, în schimb cele legate de Google Maps API nu pot fi accesate.

Odată ce dăm click pe „My reviews and ratings” urmează un ecran în care este de fapt afișat un istoric al rating-urilor și al recenziilor pe care le-a dat utilizatorul de-a lungul timpului. Acestea sunt preluate dintr-o bază de date locală, însă de fiecare dată când internetul este activat aceasta este actualizată cu ceea ce este stocat în baza de date la distanță. Acest istoric nu poate fi modificat deoarece rating-urile și recenziile oferite de utilizatori nu pot fi șterse odată ce au fost făcute publice. Funcționând și fără internet, utilizatorul poate regăsi aici oricând istoricul.



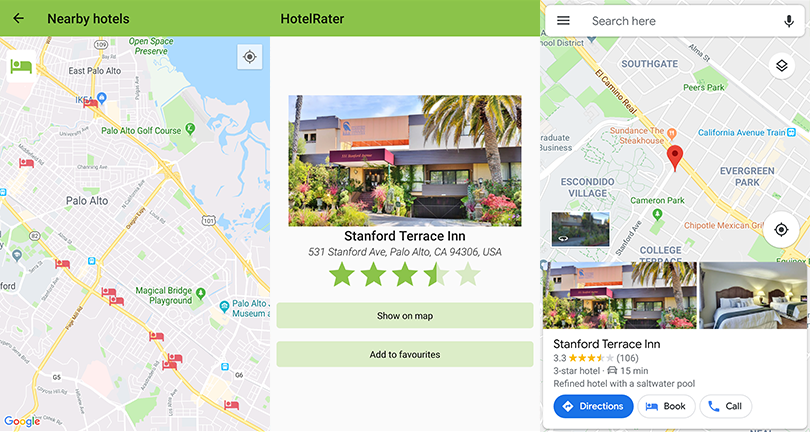
**Figura 13.** *Activitatea Maps prin care pot fi căutate informații legate de locații*

Mai departe utilizatorul poate opta pentru regăsirea unor informații de bază legate de orice fel de locație care există pe hartă și este înregistrată în API-ul celor de la Google. Accesând activitate „Maps”, utilizatorul va fi întâmpinat de o hartă ce conține câteva elemente pe care una obisnuită nu le-ar avea, ceea ce se vede în figura 7.

În partea cea mai superioară se regăsește un element grafic „AutoCompleteTextView”, prezența lupei sugerând faptul că este folosit pentru căutări, unde utilizatorul poate introduce ceea ce dorește să caute, primind sugestii bazate pe locațiile deja existente.

În urma selectării unei locații din lista celor care sunt sugerate, apare pe hartă un marcaj de culoare roșie ce indică locația căutată. Dacă utilizatorul doresțe să afle mai multe detalii despre aceasta este necesară apăsarea pe iconița sugestivă, ce conține litera „i” din dreapta sus, urmând să afișeze următoarele informații: adresă, număr de telefon, website și un rating.

Pentru recentrarea hărții asupra locației curente a dispozitivului trebuie să fie apăsată iconița din partea superioară a colțului drept. Se observă că în partea din stânga a hărții avem o iconiță asemănătoare unei hărti. Dacă se accesează aceasta, utilizatorul va fi trimis aplicația celor de la Google unde vor fi vizibile toate locațiile din baza lor de date și alte sugestii bazate pe locația curentă. După ce este selectată o locație în aplicația de la Google, aceasta va fi returnată în harta HotelRater prin apăsarea butonului „Select”.



**Figura 14.** *Activitatea Nearby Hotels prin care pot fi vizualizate rapid hotelurile din împrejurimi*

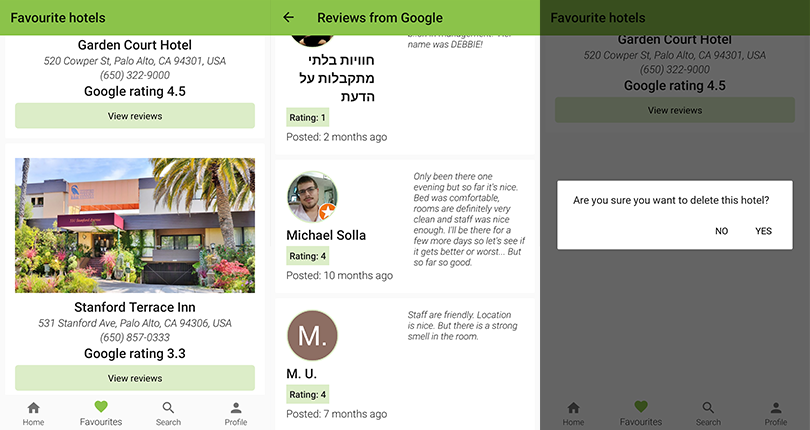
Activitatea ce se ocupă de afișarea hotelurilor din jurul dispozitivului a fost realizată separat de către restul modulelor legate de Google API. În fragmentul Home, apăsând pe butonul de tip card Nearby hotels, utilizatorul poate accesa conținutul.

Conform figurii 8, odată ce este încărcată harta, utilizatorul poate să apese pe butonul din sânga sus pentru a putea vizualiza hotelurile de pe o arie de zece kilometrii. Acestea apar sub forma unor iconițe rosii sugestive hotelurilor. Pentru a reveni la locația inițială acesta trebuie să apese pe singurul buton existent din partea dreaptă.

Dacă utilizatorul va apăsa pe unul dintre hoteluri va vedea un ecran în care sunt afișate anumite detalii legate de locația respectivă, numele, adresa și ratingul pe care această locație îl are pe Google.

În cazul în care utilizatorul alege varianta „Show on map”, acesta va fi redirecționat către aplicația celor de la Google unde va putea să își aleagă ori să sune, ori să facă o rezervare și bineînțeles să afle direcțiile către acea locație prin serviciul de tip GPS al aplicației Google Maps. De asemenea, în aplicația în care utilizatorul este redirecționat acesta poate vizualiza și mai multe fotografii ale locației, timpul pe care îl face până în locație și numărul de rating-uri pe care le-a primit locația în total. Informațiile de bază sunt afișate în HotelRater, iar dacă utilizatorul dorește să afle mai mult, poate face acest lucru prin butonul „Show on map”.

În cazul în care utilizatorul va alege să apese cel de al doilea buton, „Add to favourites”, acest ecran va dispărea iar hotelul împreună cu toate recenziile acestuia preluate din Google Places API va fi salvat într-o listă de hoteluri favorite pe care doar utilizatorul o poate accesa. Cu ajutorul acestei liste utilizatorul va putea oricând, indiferent că are conexiune la internet sau nu, să o vizualizeze.

****

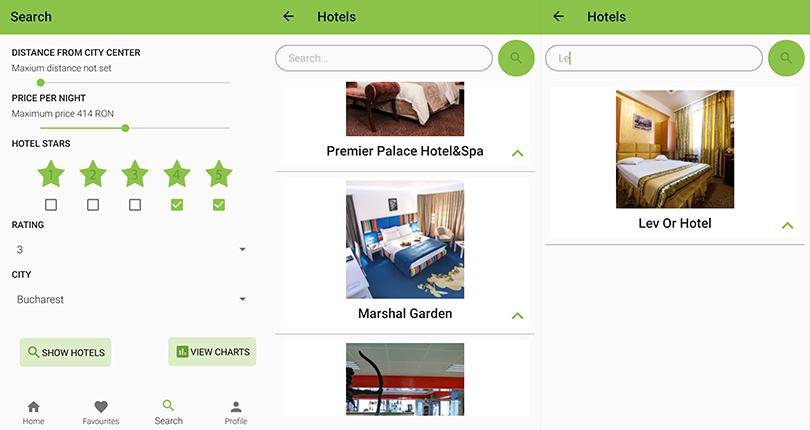
**Figura 15.** *Fragmentul „Favourites”, unde pot fi vizualizate toate hotelurile favorite împreună cu recenziile acestora*

Utilizatorul poate selecta din bara de navigație inferioară fragmentul „Favourites”, unde se regăsește lista personală cu toate hotelurile favorite culese cu ajutorul API-ului de la Google Maps și Google Places.

Această listă nu are limite de stocare, pot fi adăugate câte hoteluri dorește utilizatorul. În lista principală pot fi vizualizate doare anumite informații de bază legate de respectivele locații, printre acestea enumerându-se: numele hotelului, adresa exactă, un număr de telefon și rating-ul pe care hotelul respectiv îl are pe Google.

Apăsând pe butonul „View reviews”, utilizatorul este redirecționat către un alt ecran unde poate să vizualizeze câteva recenzii legate de locația respectivă. Acești utilizatori nu sunt neapărat existenți pe platforma HotelRater, însă au conturi Google. În această listă de recenzii a fiecărui hotel în parte, se pot vedea date precum numele utilizatorilor, ratingul pe care l-au oferit în momentul recenziei, textul recenziei date și momentul când a fost publicată recenzia. Prin vizualizarea acestor recenzii, utilizatorul poate să își facă o idee destul de clară legată de orice locație.

Acest fragment poate fi vizualizat și în lipsa conexiunii la internet, deoarece hoteluri și recenziile sunt salvate și la nivel local. În cazul supraaglomerării listei, utilizatorul are posibilitatea să șteargă un hotel prin a glisa degetul spre stânga sau dreapta asupra unui hotel din listă. După care apare un „Dialog” de confirmare a ștergerii. Pentru motive de persistență a datelor, acestea nu pot fi șterse decât în prezența internetului.

****

**Figura 16.** *Fragmentul „Search”, ecranul de unde pornește întregul proces de căutare a hotelurilor, recenziilor și a rating-urilor*

Al treilea element din bara de navigație de tip meniu, este „Search”. Alegând acest element din meniu, utilizatorul va fi întâmpinat de către un ecran în care sunt prezente mai multe tipuri de filtre de căutare, fapt ce poate fi văzut în ecranul stâng din figura 10,

Primul filtru de căutare este legat de distanța hotelului până în centrul orașului din care face parte. Dacă bara de progres nu este modificată, nealegându-se o anumită distanță, filtrul nu va fi aplicat căutării.

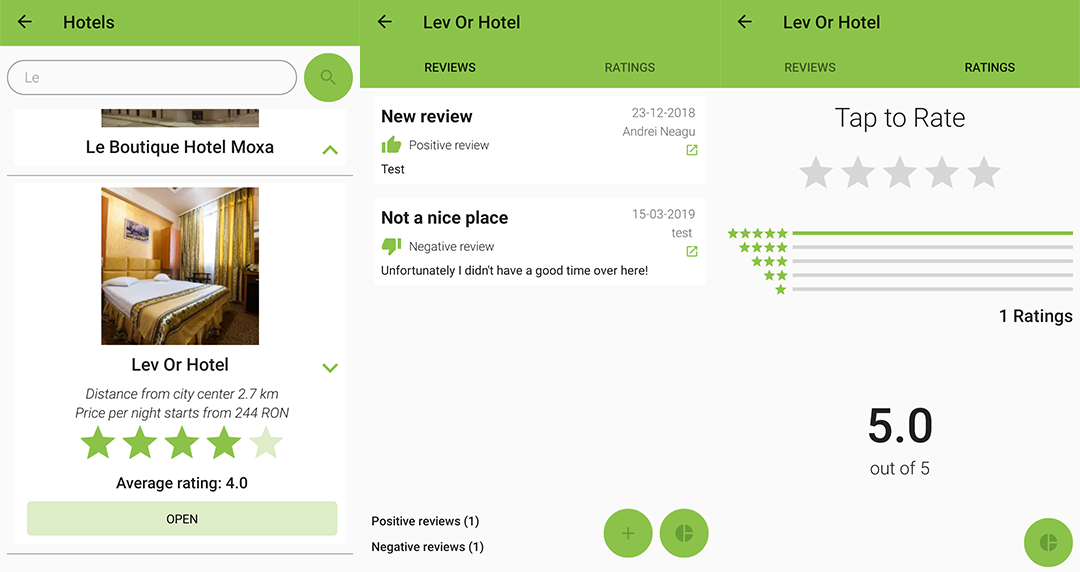
Al doilea filtru este legat de prețul maxim pe care îl are un hotel pe noapte. La fel ca și la filtrul distanței, dacă acesta nu este ales, va rămâne exclus în momentul căutării. După care avem numărul de stele, unde bineînțeles, vor fi afișate doar hotelurile cu numărul de stele ales. Dacă nu este selectat niciun număr de stele, vor fi afișate toate hotelurile.

Mai departe avem un filtru legat de rating, aici este vorba de rating-ul dat la nivelul aplicației HotelRater, nu cel preluat din Google API. În cazul în care utilizatorul alege unul din ratingurile de acolo, alegerea respectivă va reprezenta un minim de rating în filtrarea căutării de hoteluri. De asemenea, dacă nu este selectat, nu va exista filtru asupra rating-ului.

În final avem filtrarea dupa locație, iar aplicația fiind oarecum un prototip, aici avem o limită de selecții, fiind disponibile doar orașele Bucuresți, Brașov și Constanța. La fel ca și la celelalte filtre, dacă nu este selectat niciun oraș vor fi afișate toate hotelurile din cele trei orașe din baza de date HotelRater.

Dacă este apăsat butonul „SHOW HOTELS”, în cazul în care conexiunea la internet este disponibilă, va urma o listă de hoteluri ce sunt preluate din baza de date în funcție de filtrele alese de către utilizator, în caz contrar nu va putea fi realizată căutarea.

În ecranul de mijloc din figura 10 se observă lista hotelurilor. Aici prin intermediul locului de căutare din partea de sus, utilizatorul poate să caute hotelurile chiar și după primele litere, iar în urma apăsării iconiței de tip lupă, va fi încă odată lista filtrată în funcție de cuvintele sau literele introduse în căsuța de căutare.



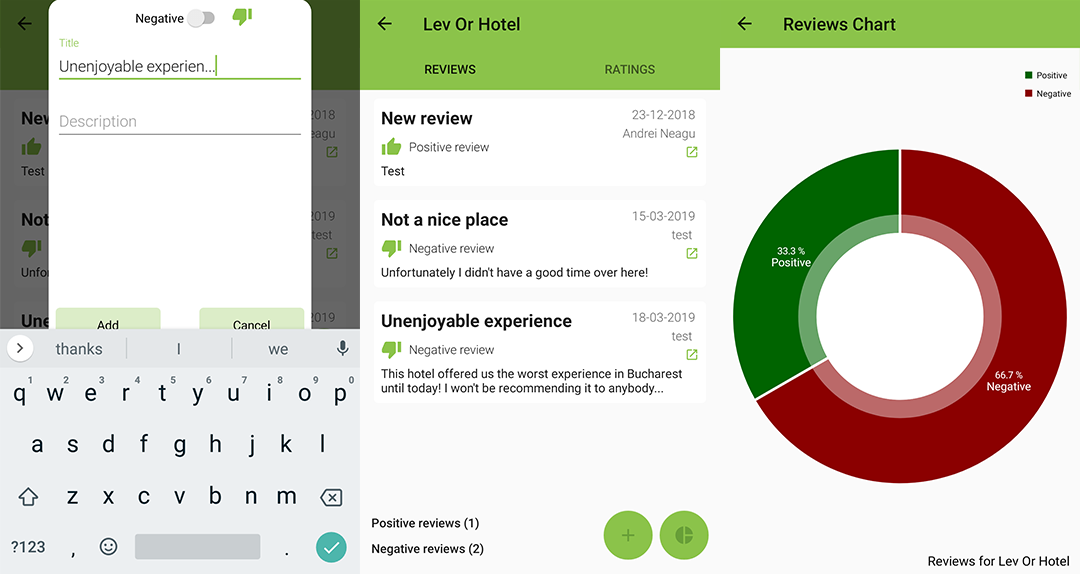
**Figura 17.** *Activitatea în care sunt listate hotelurile și o altă activitate ce conține o detaliere a recenziilor și rating-urilor*

Utilizatorul va apăsa pe iconița de tip săgeată din partea dreaptă a imaginii hotelului, urmând să apară un set de informații dupa apariția unei animații. Utilizatorul poate vedea distanța față de centru, prețul pe noapte minim, numărul de stele ale hotelului și media rating-urilor oferite de către utilizatorii HotelRater.

Prin apăsarea butonului „OPEN”, prezent în stânga figurii 11, se va deschide un alt ecran care este împărțit în doua fragmente. Acestea pot fi accesate prin a glisa ecranul spre stânga pentru recenzii sau spre dreapta pentru rating.

În partea recenziilor, ecranul din mijlocul figurii 11, putem să vedem întreaga listă de recenzii oferite de către utilizatori, fiind disponibil titlul recenziei, textul acesteia, data în care a fost publicată, numele utilizatorului și fie dacă această recenzie a fost una pozitivă sau negativă pentru a le fi mai ușor utilizatorilor să își facă o idee în cazul în care există un număr extrem de mare de recenzii. Numărul de recenzii și tipul acestora este contorizat în stânga jos.

De asemenea, dacă utilizatorul apasă pe iconița de sub numele unui utilizator din listă, poate fi vizualizat profilul respectivului utilizator în cazul în care are profilul public. Pe profilul unui utilizator pot fi găsite informații personale și date referitoare la activitatea acestuia în aplicația HotelRater. În partea rating-urilor, ecranul din dreapta figurii 11, se poate vedea numărul de rating-uri și media acestora.



**Figura 18.** Fragmentul în care se adaugă recenziile și un grafic al proporțiilor

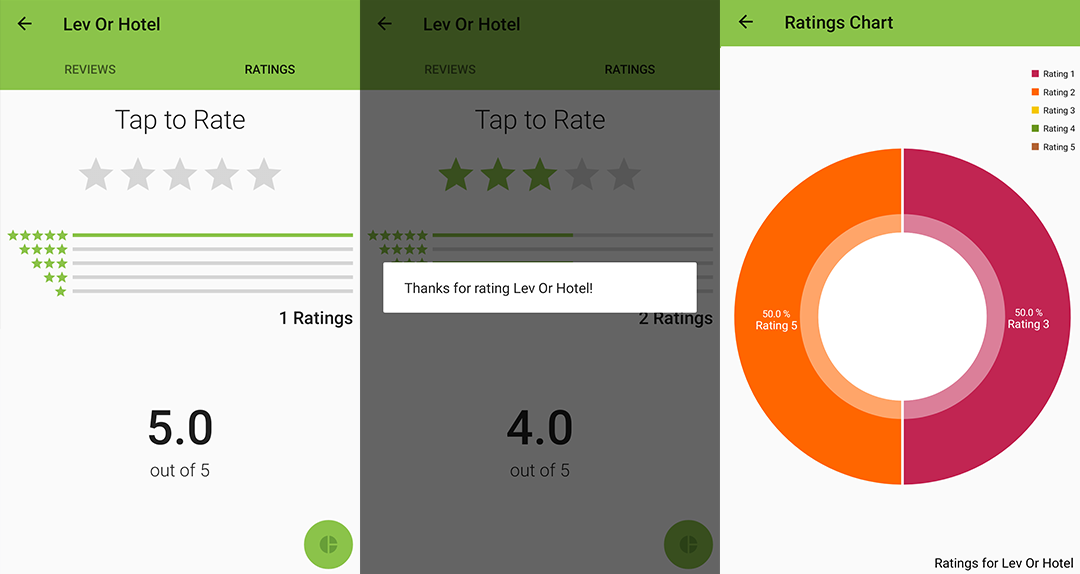
Utilizatorul trebuie să apese butonul de adăugare din dreapta jos vizibil pe mijlocul figurii 12, ulterior pe ecran va apărea un dialog în care acesta trebuie să introducă titlul recenziei și descrierea acesteia, descrierea are o anumită limita care nu îi permite utilizatorului să mai scrie odată ce este atinsă. Deasupra acestora prin intermediul elementul grafic de tipul „Switch”, utilizatorul alege tipul recenziei, pozitivă sau negativă. Aceste lucruri se pot observa in partea stângă a figurii 12. Dacă se apasă butonul „Add”, va fi publicată recenzia, iar dacă se apasă „Cancel”, operațiunea de adăugare a recenziei va fi anulată.

În cazul în care utilizatorii au profilul public acesta poate fi accesat prin intermediul iconițelor situate în dreapta recenziilor, iar in partea inferioară se poate vedea contorizarea tipurilor de recenzii.

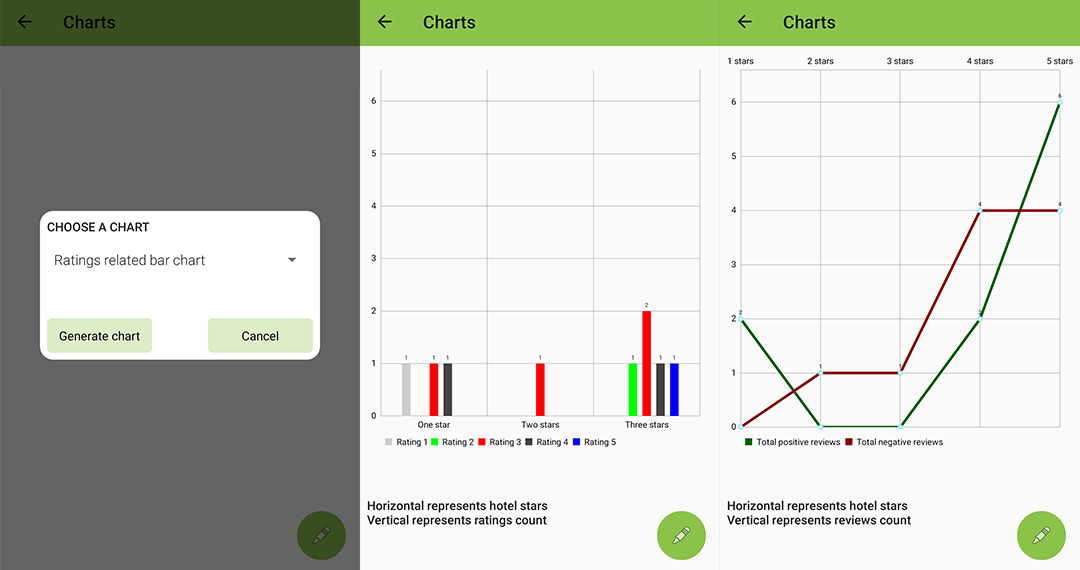
În momentul apăsării butonului ce sugestiv din dreapta jos, va fi deschisă o noua activitate în care utilizatorul poate vedea o diagramă, un grafic al proporțiilor tipurilor de recenzii pentru respectivul hotel.

După glisarea degetului pe ecran spre direcția dreaptă, se poate vedea ecranul unde se realizează rating-urile, acest ecran poate fi vizualizat în partea stângă a figurii 13. Pentru a se oferi un rating respectivului hotel tot ceea ce trebuie facut este să fie apăsat elementul grafic de tipul „RatingBar”. Acest rating poate fi dat de mai multe ori, cel anterior fiind reactualizat în baza de date de fiecare dată când este alt rating oferit. În același timp este actualizat în timp real și rating-ul mediu care poate fi văzut pe mijloc ecranului.

Pe mijloc se pot vedea într-un mod concis numărul de rating-uri total pe care le-a primit hotelul, cât și o oarecare proproție a acestora care poate fi văzută mai în detaliu la nivelul diagramei. Pentru a accesa diagrama este necesar ca utilizatorul să apese butonul sugestiv din dreapta jos de unde va fi pornită o activitate. În această activitate se vede graficul proporțiilor ce are în dreapta sus o mică legendă informativă.

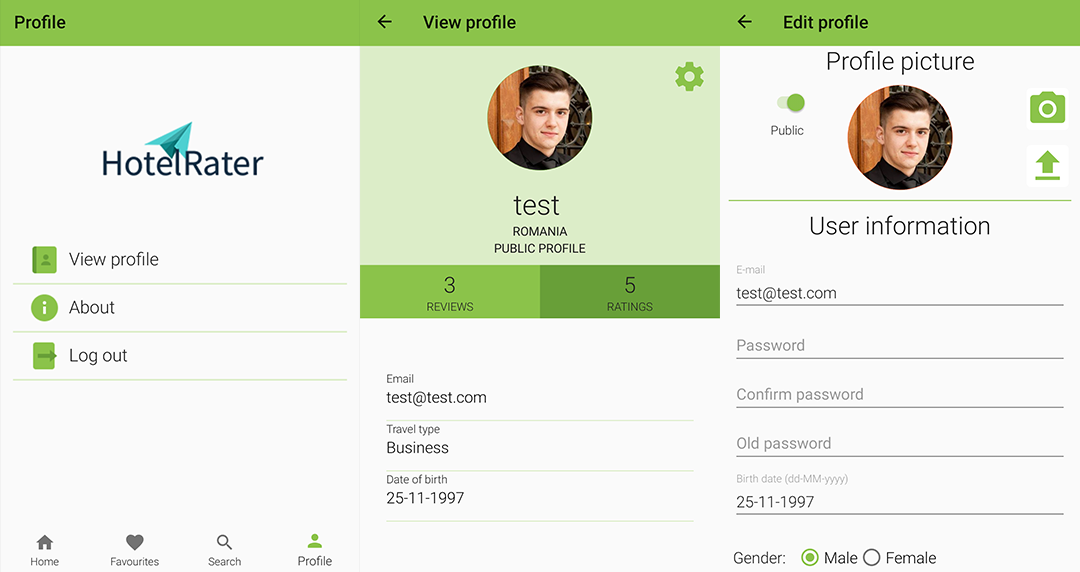
****

**Figura 19.** Fragmentul în care se adaugă rating-urile și un grafic al proporțiilor acestora



**Figura 20.** Activitatea ce conține statistici generale din aplicație

Pentru a fi accesată activitatea acestor grafice generale, se apasă butonul „Charts” din dreapta jos ce poate fi văzut în partea stângă a figurii 10. Odată ce aceasta este accesată, utilizatorul trebuie să aleagă unul dintre cele doua grafice, fie cel de tip bară fiecel liniar. Acest lucru se face prin selectarea opțiunii din „Dialog”, urmând să fie apăsat butonul „Generate chart”.

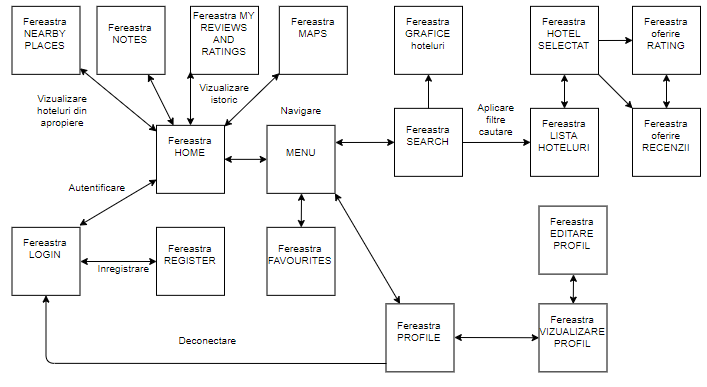
Aceste grafice sunt oarecum auxiliare, sunt mai mult informative în privința activității care există în aplicație, reprezentând numărul de rating-uri și recenzii oferite de-a lungul timpului în funcție de numărul de stele al hotelurilor. În primul grafic din figura 14, se observă numărul de rating-uri date pentru fiecare număr de stele al hotelurilor, iar în al doilea grafic se observă numărul de recenzii pozitive și negative în funcție de numărul de stele al hotelurilor. Utilizatorii curioși pot ajunge la astfel de date statistice publice legate de aplicația HotelRater, satisfăcându-și anumite posibile curiozități referitoare la aplicație. 

**Figura 21.** Fragmentul „Profile”, Activitatea pornită de către „View profile” și editarea informațiilor personale

Selectând elementul „Profile” din bara de navigație, ajungem la un alt meniu de dimensiuni mai mici cum se poate vedea în figura 15. Aici putem să vizualizăm informațiile personale apăsând „View profile”, putem afla informații despre aplicație apăsând „About” sau putem să ne deconectăm din aplicație prin apăsarea butonului „Log out”.

În cazul apăsării butonului „View profile”, se pot vizualiza datele utilizatorului din HotelRater. Aici putem vedea o poză de profil, țara de origine, daca profilul este public sau privat, numărul de recenzii și rating-uri date de-a lungul timpului, adresa de e-mail a utilizatorului, tipul general al călatoriilor și data nașterii. Este important de știut că aceste informații personale sunt opționale și trebuie completate doar în cazul în care utilizatorul își dorește acest lucru, obligatorii fiind doar adresa de e-mail, numele și parola care au fost introduse la înregistrarea utilizatorului în aplicație. Pentru a modifica aceste informații trebuie apăsat butonul de tip rotiță care este sugestiv pentru modificări, vizibil în partea dreaptă sus în mijlocul figurii 15. Pentru a putea fi accesată această activitate de modificare a datelor este necesară prezența internetului. Aici utilizatorul poate să se folosească fie de cameră, fie de galeria de fotografii pentru a își pune o fotografie de profil nouă. Toate datele cu excepția adresei de e-mail pot fi schimbate precum se vede în partea dreaptă a figurii 15. Câmpurile de date pot fi schimbate oricând atâta timp cât acestea respectă formele de validare prezente la nivelul câmpurilor. Pentru a fi salvate informațiile trebuie glisat ecranul în jos și apăsat butonul sugestiv „Save”, în caz contrar informațiile nu vor fi salvate.

În diagrama de mai jos se poate observa felul în care au fost proiectate interfețele utilizatorului.



**Figura 19.** *Diagrama realizata pentru a prezenta interfetele utilizatorului*

# 5. Concluzie

Am reușit prin intermediul aplicației HotelRater să creez un mediu foarte accesibil pentru orice tip de utilizator, în care să fie posibilă vizualizarea și oferirea de recenzii și rating către diverse hoteluri, alături de alte funcționalități secundare.

Ceea ce aduce HotelRater nou pe piața aplicațiilor hoteliere este faptul că reușește să-i ofere utilizatorului date din doua surse diferite, din HotelRater dar și dintr-o sursă exterioară, Google API, oferindu-i astfel utilizatorului un spectru mai larg de informații legate de diverse locații hoteliere.

Căutarea și afișarea hotelurilor din baza de date a aplicației este făcută cum a fost gandită tocmai de la început, într-un mod detaliat, permițându-i eventualului utilizator să își personalizeze filtrele de căutare în funcție de ceea ce are nevoie.

De asemenea, am reușit să implementez și un modul prin care utilizatorii să poată afla într-un mod sumar date statistice referitoare la rating-uri și recenzii afișate în niște grafice la nivelul aplicației.

Pe lângă modulele principale ale aplicației, am reușit să ofer prin HotelRater și niste funcționalități mai mici prin care utilizatorul poate să caute prin intermediul Google Maps detalii legate de diverse locații, să își salveze notițe la nivelul aplicației, să poată să își vadă istoricul de recenzii și rating-uri oferite de-a lungul activării în aplicație.

Momentan, aplicația este într-o formă bună, însă pentru a își atinge potențialul maxim este necesară o colaborare cu o firmă mare deja existentă pe piață pentru a putea fi obținut accesul la un API sau o bază de date de dimensiuni mari.

Am introdus la nivelul aplicației și un mic punct de plecare pentru o eventuală deviere a direcției curente pe care o are aplicația, un punct de plecare pentru o aplicație de tipul rețea de socializare. Momentan HotelRater nu permite comunicarea între utilizatori deoarece nu am propus să ating acest aspect în această lucrare. În momentul în care utilizatorii se uită prin recenzii și ratinguri aceștia își pot vizualiza anumite date personale în funcție de anumite setări de "privacy”. Mai departe acest aspect poate fi dezvoltat, fiind o direcție foarte solidă, deoarece se poate ajunge la un produs final bun în urma unei astfel de îmbricări de concepte.

În viitor, aș dori să duc această aplicație la următorul nivel prin îmbogățirea bazei de date cu hoteluri, printr-o eventuală colaborare cu Booking.com sau orice altă aplicație ce îmi poate oferi serviciile necesare, pentru a putea avea acces la eventuale rezervari prin API-ul oferit de acestia. Astfel, orice utilizator, indiferent de locație, va avea șansa să consume în totalitate ceea ce are de oferit HotelRater când vine vorba de călătorit, hoteluri, recenzii, rating și chiar rezervări hoteliere, pentru a acoperi toate necesitățiile utilizatorului.

În concluzie, se poate spune ca prin realizarea acestei lucrări am reușit să ating în totalitate planul de dezvoltare al aplicației Android, HotelRater, demonstrând ca se pot implementa mai multe module folositoare utilizatorului într-o aplicație cu tematică hotelieră ce are drept nucleu oferirea alături de vizualizarea ratingurilor și recenziilor aferente hotelurilor.

# 6. Referințe

[1] P. Pocatilu, F. Alecu, I. Ion, A. Zamfiroiu și B. Iancu, Programarea aplicatiilor Android, Editura ASE, 2015.

[2] P. Pocatilu, Informatica-Economica, Android Applications Security, vol. 15, nr. 3/2011

[3] Wikipedia, Android Software Development, <https://en.wikipedia.org/wiki/Android_software_development>. Extras 13/03/2019.

[4] R. Rogers, J. Lombardo, Z. Mednieks și B. Meike – Android Application Development – Programming with the Google SDK (pg 3)

[5] Wikipedia, Java Virtual Machine, <https://en.wikipedia.org/wiki/Java_virtual_machine>. Extras 13/03/2019.

[6] Wikipedia, List of features in Android, https://en.wikipedia.org/wiki/List\_of\_features\_in\_Android. Extras 13/03/2019.

[7] Android Arhitecture Components, <https://www.dev2qa.com/android-architecture-components-introduction/>. Extras la data de 13/03/2019.

[8] Dashboards Android, <https://developer.android.com/about/dashboards>. Extras 13/03/2019

[9] Android Studio Intro, <https://developer.android.com/studio/intro/>. Extras 14/03/2019.

[10] Introduction to Firebase, <https://hackernoon.com/introduction-to-firebase-218a23186cd7>. Extras 14/03/2019.

[11] Introduction to SQLite Database, <https://www.edureka.co/blog/introduction-on-android-sqlite-database/>. Extras 14/03/2019.

[12] Android SDK, <https://developers.google.com/maps/documentation/android-sdk>. Extras 14/03/2019.

[13] Android SDK Intro, <https://developers.google.com/places/android-sdk/intro>. Extras 14/03/2019.

[14] Android Picasso, <https://www.journaldev.com/13759/android-picasso-tutorial>. Extras 15/03/2019.

[15] Retrofit, <https://www.journaldev.com/13639/retrofit-android-example-tutorial>. Extras 15/03/2019.

[16] Wikipedia, Booking.com, <https://en.wikipedia.org/wiki/Booking.com>. Extras 12/03/2019.

[17] Wikipedia, Trivago, <https://ro.wikipedia.org/wiki/Trivago>. Extras 12/03/2019.