

**Facultatea de Automatică și Calculatoare**

****

**Programul de Licență: Automatică și Informatică**

**Aplicată**

****

**APLICATIE MOBILĂ ANDROID PENTRU UN RESTAURANT**

**Proiect de diplomă**

Alex-Ionuț Pacioagă

Coordonator:

Inginer Cristian Zimbru

2018

Cuprins

[Capitolul 1. Introducere](#page4) [1](#page4)

[1.1. Context](#page4) [1](#page4)

[1.2. Tema proiectului](#page5) [2](#page5)

[1.3.](#page6) Android - Prezentare Generală [3](#page6)

[Capitolul 2. Sisteme și tehnologii folosite](#page7) [4](#page7)

[2.1. Android](#page7) [4](#page7)

[2.1.1. Aplicații](#page9) [6](#page9)

[2.1.2. Dezvoltare](#page9) [6](#page9)

[2.1.3. Licenţa Android](#page11) [8](#page11)

[2.2. Firebase](#page11) [8](#page11)

[2.2.1. Design-ul și structurarea datelor în Firebase](#page12) [9](#page12)

[2.2.2. Inserția](#page14) Firebase în proiectul Android [11](#page14)

[2.2.3. Android, IOS și Firebase](#page14) [11](#page14)

[2.3. Limbajul](#page16) Java [13](#page16)

[2.3.1.](#page18) Informații generale [15](#page18)

[2.3.2.](#page22) Descrierea detaliată a limbajului [19](#page22)

[Capitolul 3. Prezentarea Aplicației](#page25) [22](#page25)

[3.1. Scopul aplicației](#page25) [22](#page25)

[3.2 Prezentarea interfețelor](#page26) [23](#page26)

[3.3. Implementarea aplicației](#page37)

[3.3.1. Implementarea generală](#page37)

[3.3.2. Implementarea în detaliu](#page39)

Capitolul 4. Exemple și simularea rezultatelor

4.1 Exemple de aplicații similare

4.2 Simularea aplicației

Capitolul 5. Concluzii

**Lista figurilor**

Fig 1.1 Apariția avioanelor

Fig. 1.2 Aeronavă cu destinație civilă

Fig. 2.1 Radar primar

Fig. 2.2 Radar secundar

Fig. 3.1 Tehnologia ADS-B

Fig. 4.1 ADS-B Funcționare

Fig. 4.2 Schema bloc a sistemului hardware

Fig. 5.1 Element antenă

Fig. 5.2 Conectarea elementelor antenei

**Capitolul 1. Introducere**

* 1. **Context**

În ziua de azi tehnologia se dezvoltă într-un ritm foarte alert, deoarece trăim în secolul vitezei, secol în care în fiecare zi exista posibilitatea ca cineva sa descopere ceva nou sau îmbunătăţeşte ceva existent. Aplicaţiile software cunosc și ele o creştere majoră iar multe din aceste aplicaţii au nevoie de conturi pentru a fi accesate. Odată cu creşterea puterii computaţionale creşte şi nevoia securităţii iar utilizatorii sunt nevoiți și obligați de multe din aplicații să își creeze un cont format dintr-un username, de obicei un e-mail sau un numar de telefon, precum si o parola cât mai puternică.

De vreme ce Android este un produs open-source, producătorii având posibilitatea de a-l modifica în mod gratuit, nu există configurații hardware sau software standard. Piața de telefoane inteligente este dominată de Android (care - potrivit estimărilor - își va menține supremația până în 2018), cu 82.8% (peste un miliard de dispozitive vândute) pe întreg anul 2015, în timp ce Apple deține doar 13.9%, la nivel mondial.

Așadar, dacă pentru utilizare personală nu se poate stabili un câștigător clar între Android și iPhone (deși produsul celor de la Apple pare să aibă un ușor avantaj printr-un număr mai mare de aplicații și prin loialitatea clienților), în cazul aplicațiilor dezvoltate de corporații situația este inversă, date fiind posibilitățile de instalare și limbajul de programare folosit.

Aplicațiile destinate dispozitivelor mobile reprezintă un segment extrem de productiv al economiei, doar vânzările din AppStore depășind încasările obținute din producția de filme de la Holywood.

* 1. **Tema proiectului**

“Imaginaţia noastră este singura limită a ceea ce putem spera să realizăm în viitor.”, constata inventatorul american Charles Franklin Kettering. În contextul societăţii

actuale, înţelesul acestei mărturii se poate extinde în sensul că, ceea ce conduce la progresul tehnologiei şi la dezvoltarea de software inovativ este finalmente, nimic altceva decât fantezia umană.

Aplicaţia este scrisă în limbajul de programare Java pentru sistemul de operare Android și utilizează Firebase ca bază de date.

Lucrarea de faţă are ca scop prezentarea unei aplicaţii care ilustrează posibilitățile gastronomice în privința alimentației zilnice de care un individ este responsabil, bazându-se îndeosebi pe construirea unui meniu particularizat:

* nevoia de a satisface necesitățile fizice;
* interfațarea cu o varietate de tehnologii ce stau la baza limbajului Java;
* se ia în vedere facilităţile puse la dispoziţie de sistemul de operare Android;

Aplicaţia ”Restaurant”, aşa cum am hotărât să o denumesc, pune în legătură utilizatorilor ei prin furnizarea unei largi varietăți de produse culinare persoanelor înregistrate care folosesc aceast program. Un utilizator are posibilitatea de a fi notificat în cazul în care se află în imposibilitatea de a se loga în baza de date returnând un mesaj generic, sau în cazul în care din motive diverse comanda nu se va putea realiza, iar clientului i se vor prezenta opțiunile alternative.

Pe lângă aceasta, un utilizator se poate înregistra în baza de date, creându-și un cont unic, pentru a-și desfăşura acțiunile de selecție respectiv comandă de produse culinare. Una dintre caracteristicile interesante puse la dispoziţie de acest program, este posibilitatea vizualizarii felurilor de mancare dorite într-un mod ușor accesibil și rapid.

* 1. **Android - Prezentare Generală**

Android este un Sistem de Operare mobil al carui baze s-au construit pe o versiune modificată de Linux (utilizat in gestiunea componentelor hardware dar si a proceselor și a memoriei) și biblioteci Java (pentru telefonie (audio/video), conectivitate, grafică, programarea interfețelor cu utilizatorul).

Acest sistem este un produs open-source, ce poate fi dezvoltat de producătorii de dispozitive mobile cu extensii proprietare pentru a-și particulariza platforma dupa propriile cerințe, dezvoltat în prezent de compania Google, conceput pe ideea transformării dispozitivelor mobile în adevărate mașini de calcul.

Google încearcă totuși să realizeze tranziția de la AOSP (Android Open Source Project) către GMS (Google Mobile Services), peste care sunt construite cele mai multe aplicații, în încercarea de a-și apăra acest proiect în fața concurenței. În acest sens, a fost dezvoltat proiectul Google One, prin care este oferit un set de specificații (ecran de 4.5 inchi - 845×480 pixeli, procesor quad-core, memorie 1GB RAM, spațiu de stocare 4GB, suport pentru dual sim) pe care producătorii de dispozitive mai ieftine trebuie să le respecte astfel încât acestea să fie compatibile cu un sistem Android, fără a întâmpina probleme de performanță. Acesta include toate aplicațiile și serviciile Google, la care se pot adăuga și altele, furnizate de producător sau operatorul de telefonie mobilă. În acest fel, se asigură calitatea (păstrând renumele Android) și controlul asupra veniturilor.

Comunitatea Android este în creștere, mai multe companii renunțând la propriul sistem de operare în favoarea acestuia, pentru a putea face față fenomenului iPhone.

În condițiile în care pe piața dispozitivelor mobile aplicațiile sunt cele care aduc avantajul competițional, beneficiul Android este reprezentat de abordarea unitară pentru dezvoltarea aplicațiilor. Cu alte cuvinte, o aplicație dezvoltată conform API-ului Android va putea rula pe mai multe dispozitive mobile pe care este instalat sistemul de operare respectiv. Pentru identificarea versiunilor se folosesc, de regulă, trei sisteme:

* un număr, ce respectă formatul major.minor[.build], desemnând dacă modificările aduse sunt substanțiale sau reprezintă ajustări ale unor probleme identificate anterior; versiunea curentă este 7.1.1, lansată la sfârșitul anului 2016;
* un nivel de API (același putând grupa un număr de mai multe versiuni), prin care se indică funcționalitățile expuse către programatori; versiunea curentă are nivelul de API 25;
* o denumire, având un nume de cod inspirat din lumea dulciurilor; termenii respectivi încep cu inițiale care respectă ordinea alfabetică; versiunea curentă este Nougat.

**Tabel 1.3 Versiuni Android**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versiune Android Nivel API Data Lansării Nume de Cod Cota de Piață | | | | |
| 8.1 - 8.1 | 26 | 21.08.2017 | Oreo | 0.?% |
| 7.1 - 7.1.1 | 25 | 04.10.2016 | Nougat | 0.3% |
| 7.0 | 24 | |  |  | | --- | --- | | 22.08.2016 |  | | Nougat | 0.9% |
| 6.0 - 6.0.1 | 23 | 05.10.2015 | Marshmellow | 30.7% |
| 5.1 - 5.1.1 | 22 | 09.03.2015 | Lollipop | 23.1% |
| 5.0 - 5.0.2 | 21 | 12.11.2014 | Lollipop | 9.8% |
| 4.4W –  4.4W.2 | 20 | 25.06.2014 | KitKat with wearable extensions |  |
| 4.4 - 4.4.4 | 19 | 31.10.2013 | KitKat | 21.9% |
| 4.3 | 18 | 24.07.2013 | Jelly Bean | 1.6% |
| 4.2.x | 17 | 13.11.2012 | Jelly Bean | 5.7% |
| 4.1.x | 16 | 09.07.2012 | Jelly Bean | 4.0% |
| 4.0.3 –  4.0.4 | 15 | 16.12.2011 | Ice Cream Sandwich | 1.0% |
| 4.0 - 4.0.2 | 14 | 19.10.2011 | Ice Cream Sandwich |  |
| 3.2 | 13 | 15.07.2011 | Honeycomb |  |
| 3.1 | 12 | 10.05.2011 | Honeycomb |  |
| 3.0 | 11 | 22.02.2011 | Honeycomb |  |
| 2.3.3-  2.3.7 | 10 | 09.02.2011 | Gingerbread | 1.0% |
| 2.3 – 2.3.2 | 9 | 06.12.2010 | Gingerbread |  |
| 2.2 – 2.2.3 | 8 | 20.05.2010 | Eclair |  |
| 2.1 | 7 | 12.01.2010 | Eclair |  |
| 2.0.1 | 6 | 03.12.2009 | Eclair |  |
| 2.0 | 5 | 26.10.2009 | Eclair |  |
| 1.6 | 4 | 15.09.2009 | Donut |  |
| 1.5 | 3 | 30.04.2009 | Cupcake |  |
| 1.1 | 2 | 09.02.2009 |  |  |
| 1.0 | 1 | 23.09.2008 |  |  |

În momentul în care se ia decizia cu privire la versiunea pentru care se dezvoltă o aplicație Android, trebuie avute în vedere și cotele de piață ale dispozitivelor mobile. Dezvoltarea unei aplicații Android pentru cea mai nouă versiune are avantajul de a se putea utiliza cele mai noi funcționalități expuse prin API.

Dezvoltarea unei aplicații Android pentru cea mai veche versiune are avantajul unei adresabilități pe scară largă. Un compromis în acest sens poate fi obținut prin intermediul bibliotecilor de suport, dezvoltate pentru fiecare versiune, prin intermediul cărora pot fi utilizate la niveluri de API mai mici funcționalități din niveluri de API mai mari (în limita capabilităților dispozitivului mobil respectiv).

Utilizarea acestora reprezintă o practică recomandată în dezvoltarea aplicațiilor Android.

**Capitolul 2. Sistem și tehnologii folosite**

**2.1. Android**

**2.1.1. Aplicații**

Platforma Android constituită dintr-un nucleu Linux, cu middelware, librării şi API-uri scrise în C, precum şi aplicaţii software ce rulează pe platforme ce includ librării compatibile cu Java, bazate pe Apache, o licenţă de tip free-software şi open source, aspect ce îl face atractiv în rândul dezvoltatorilor. [2]

Android a fost proiectat pentru a fi compatibil cu o gamă largă de componente hardware. Aceasta constituie o caracteristică importantă, întrucât permite producătorilor să conceapă şi să integreze elementele necesare dispozitivului Android, făcându-l atractiv atât în rândul programatorilor cât şi al consumatorilor.

Fiind o platformă destinată dezvoltării de aplicaţii pentru telefoane mobile, în realizarea ei s-a avut în vedere conservarea energiei acumulatorului precum şi gestionarea în mod eficient a resurselor de memorie existente. Aşa cum apare în , majoritatea dispozitivelor Android au utilizat încă de la început procesoare ARM, cunoscute a fi destul de puternice, raportat la consumul mic de energie. Cu toate acestea, unele corporaţiile care au portat Android pe alte

platforme au ales să folosească propriile lor CPU-uri (ex: Intel a utilizat propriul lor procesor Atom).

Inițial pentru dezvoltarea aplicațiilor Android s-a folosit mediul de dezvoltare Eclipse, ulterior în anul 2014 Google a lansat Android Studio, acesta devenind la scurt timp principalul mediu pentru dezvoltarea aplicațiilor mobile. Pentru realizarea acestei lucrări am folosit Android Studio versiunea 2.3.3. Există și alte kit-uri de dezvoltare incluzând aici Native Development Kit (NDK) pentru scrierea aplicațiilor în limbajele de programare C/C++ şi Google App Inventor, un mediu de programare vizuală.

Numărul aplicațiilor Android este într-o continuă creștere, aplicațiile ajung pe telefoanele utilizatorilor fie prin instalarea APK-urilor (Android Application package) fie prin descărcarea acestora de pe un magazin de aplicații. Principalul magazin de aplicații este Google Play, acesta conținea în iulie 2013 peste 1 milion de aplicații. În funcție de aplicație utilizatorul o poate descărca gratis sau este nevoit sa plătească pentru aceasta, aplicațiile gratuite conțințând de cele mai multe ori reclame, acestea fiind sursa de venit a aplicației.

[**2.1.2. Dezvoltare**](#page9)

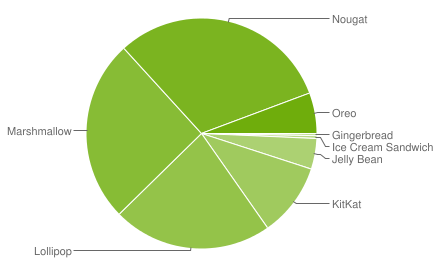
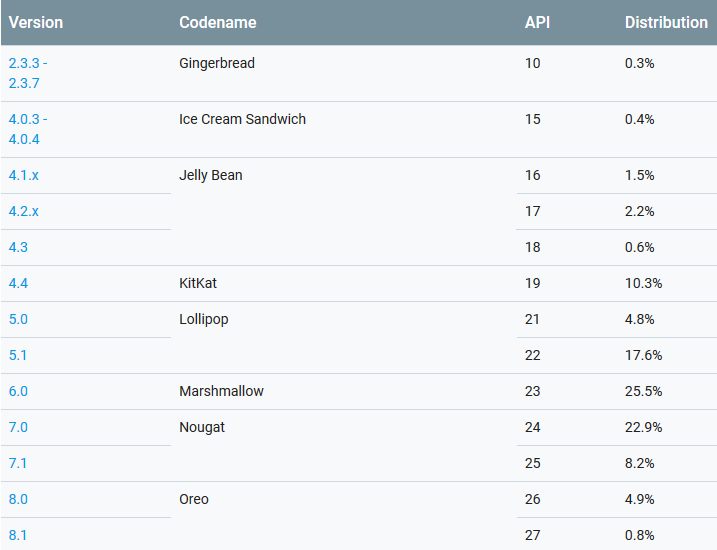
Android este dezvoltat în privat de către Google până cand ultimele modificări sunt testate şi sunt gata să fie lansate, atunci codul este facut public. Codul sursă va rula nemodificat doar pe puține dintre dispozitivele de pe piață, de obicei cele din familia Nexus. Codul sursă este modificat de cei mai mulți producători de telefoane mobile pentru a avea o interfață diferită de cea a concurenței, adăugând şi imbunătățiri. Google obișnuiește să lanseze o actualizare majoră o data la 6-9 luni, procesul de actualizare al dispozitivelor fiind mai lent decât în cazul sistemului de operare competitor dezvoltat de Apple IOS, principala cauză fiind modificarea codului sursă de către producătorii de telefoane.

Pentru dezvoltarea unei aplicații Android sunt necesare:

* kit-ul de dezvoltare pentru limbajul de programare Java
* SDK-ul de Android, pentru care se descarcă definițiile corespunzătoare unuia sau mai multor niveluri de API
* un mediu integrat de dezvoltare (IDE)
* Elipse, cu plugin-ul ADT (Android Developer Tools)
* Android Studio
* un dispozitiv pe care să se ruleze aplicațiile
* un emulator
  + Genymotion
  + Android Virtual Device (livrat împreună cu SDK-ul de Android)
* un telefon mobil cu sistemul de operare Android pentru care s-a dezvoltat aplicația

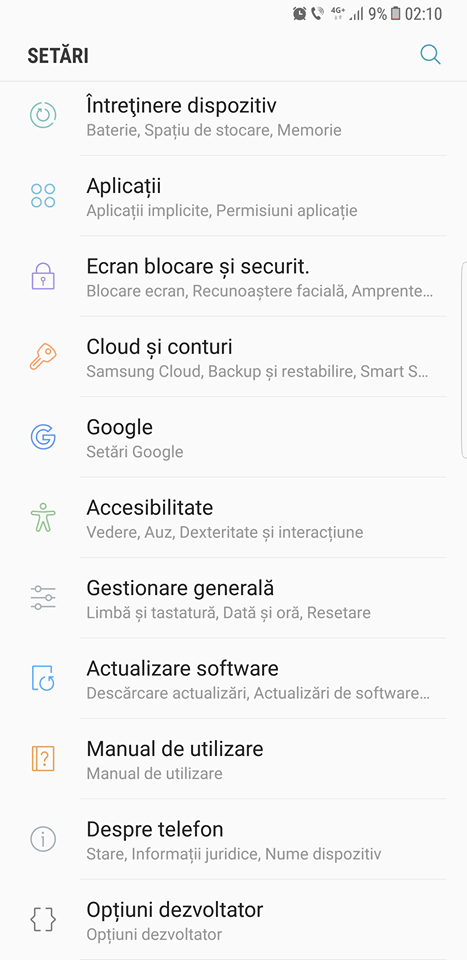
Procesul de actualizare necesită doar o conexiune la internet, întreg procesul fiind cuprins în sistemul de operare și fiind complet automat, utilizatorului nu îi mai rămâne decât să iși dea acordul că dorește ca procedura de actualizare să inceapă. Procesul lent de actualizare a fost criticat în numeroase rânduri de consumatori, acuzând firmele producătoare de dispozitive mobile că incearcă prin acest proces lent să îi oblige pe consumatori să iși cumpere dispozitive noi care vin din fabrică actualizate. Google a luat măsuri modularizând mai mult sistemul de operare astfel încât unele componente să poată fi actualizate prin intermediul magazinului Google Play. Pentru a face posibil acest lucru au introdus Google Play Services care a fost instalat pe aproape toate dispozitivele de la versiunea 2.2 până în prezent. Prin intermediul acestui serviciu, Google va putea actualiza sistemul folosind Google Play.

În figura 1.1 sunt prezentate principalele versiuni de Android împreuna cu un grafic care evidenţiază distribuția versiunilor printre dispozitivele mobile, ultima verisune de Android fiind Android 8.1 Oreo.

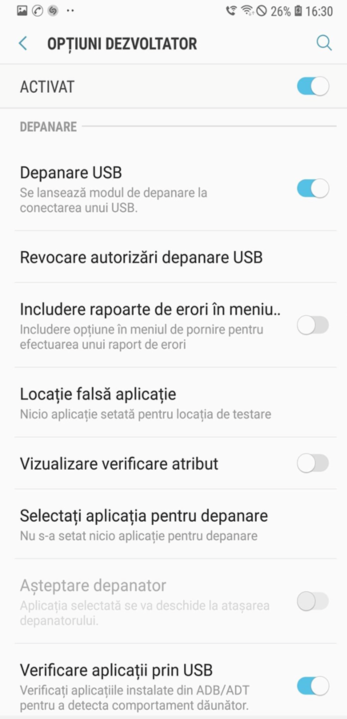


*Figura 1.1 – Sursa* [*http://developer.Android.com/about/dashboards/index.html*](http://developer.android.com/about/dashboards/index.html)

Pentru a se putea rula o aplicație pe un dispozitiv mobil fizic, trebuie să se activeze posibilitatea de depanare prin USB, din Settings → System → Developer Options. Această opțiune trebuie activată, ca de altfel și opțiunea Debugging → Android Debugging (pe unele sisteme poate apărea ca USB Debugging).

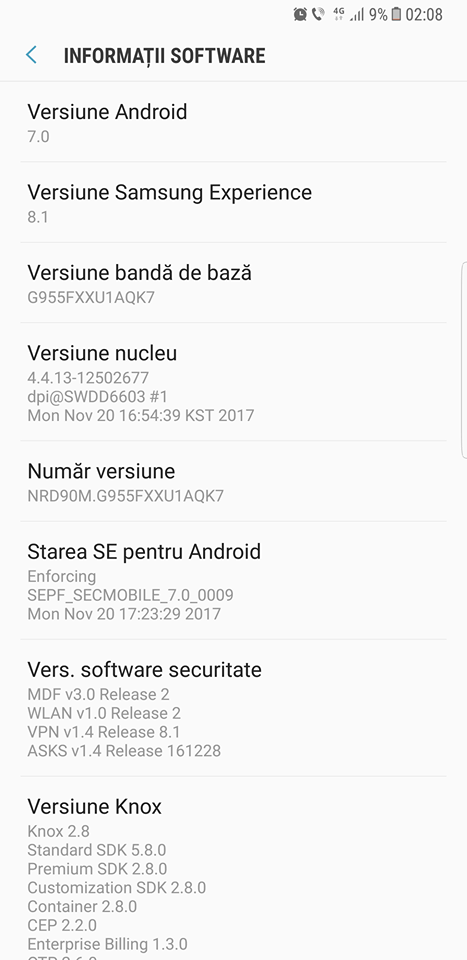


*Figura 1.1 – Optiuni dezvoltator 1*



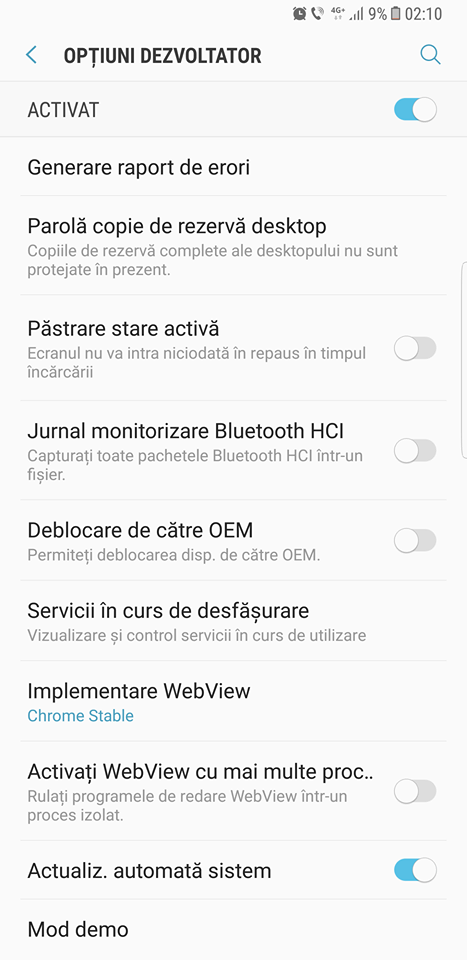
*Figura 1.2 – Activare Depanare USB*

În situația în care opțiunea Developer Options(Opțiuni dezvoltator) nu este disponibilă, aceasta poate fi vizualizată prin intermediul modului Developer, obțiunut prin apăsarea de mai multe ori asupra opțiunii Build Number(Număr versiune) din secțiunea Settings → System → About Phone.



*Figura 1.3 – Activare optiuni dezvoltator*

În momentul în care opțiunea este activată, accesând se va afișa meniul din figura următoare:



*Figura 1.4 – Optiuni dezvoltator 2*

Dacă telefonul nu este recunoscut la conectarea prin USB, trebuie instalate niște reguli pentru udev, conform instrucțiunilor de pe [stackexchange](http://unix.stackexchange.com/questions/119128/linux-mint-16-android-device-not-listed-with-lsusb). [1]

**2.1.3. Licenţa Android**

Codul sursă Android este open source şi este făcut public la fiecare lansare de versiune de către Google. API-ul Android, mediul de dezvoltare, accesul la documenție și multe alte facilități sunt oferite gratis de către Google. Astfel, pentru a dezvolta aplicații Android nu va trebui să plătești absolut nimic. Dezvoltatorul este cel care hotărăște prețul final al aplicației create de el. Această lucrare va fi făcută publică după prezentare iar codul sursă al aplicației va putea fi descărcat de pe platforma GitHub.

**2.2. Firebase**

[**2.2.1. Design Firebase**](#page12)

Firebase este o platformă de dezvoltare a aplicațiilor mobile și web dezvoltată de Firebase, Inc. în 2011, achiziționată apoi de Google în 2014.

Firebase este o baza de date în timp real care tinde spre viitor și nimic nu se poate compara cu o astfel de tehnologie dat fiind faptul ca majoritatea bazelor de date oblige utilizatorul sa faca apeluri HTTP pentru preluarea si sincronizarea datelor.

Aceste baze de date ofera informatiile doar atunci cand se cere.

Cand aplicatia se contecteaza la Firebase, conexiunea nu se face prin HTTP normal, ci printr-un WebSocket care este mult mai rapid ca un HTTP deoarece nu trebuie apelate WebSocket-uri individuale, un socket fiind sufficient.

Pentru a putea folosi Firebase este nevoie de un dispozitiv care ruleaza Android 4.0 (Ice Cream Sandwich) sau mai nou si servicii Google Play 15.0.0 sau o versiune ulterioara.

Toate informatiile din baza de date se sincronizeaza automat prin acel WebSocket, totul depinzand doar de viteza retelei de internet a utilizatorului.

Firebase trimite informatii noi de fiecare data cand se actualizeaza baza de date. Când clientul current salvează o modificare a datelor, toți clienții conectați primesc aproape instantaneu datele actualizate. [2]

Așadar, Firebase are o mulțime de avantaje precum:

• Autentificare prin Email & parolă. Google, Facebook si Github

• Date in timp real

• Ready-made api

• Securitatea datelor la nivel de nod

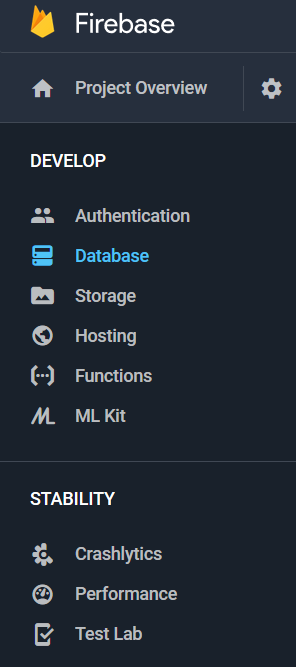
• Stocarea fișierelor susținută de Google Cloud Storage

• Găzduire fișiere statice

• Tratează datele ca fluxuri pentru a construi aplicații foarte scalabile

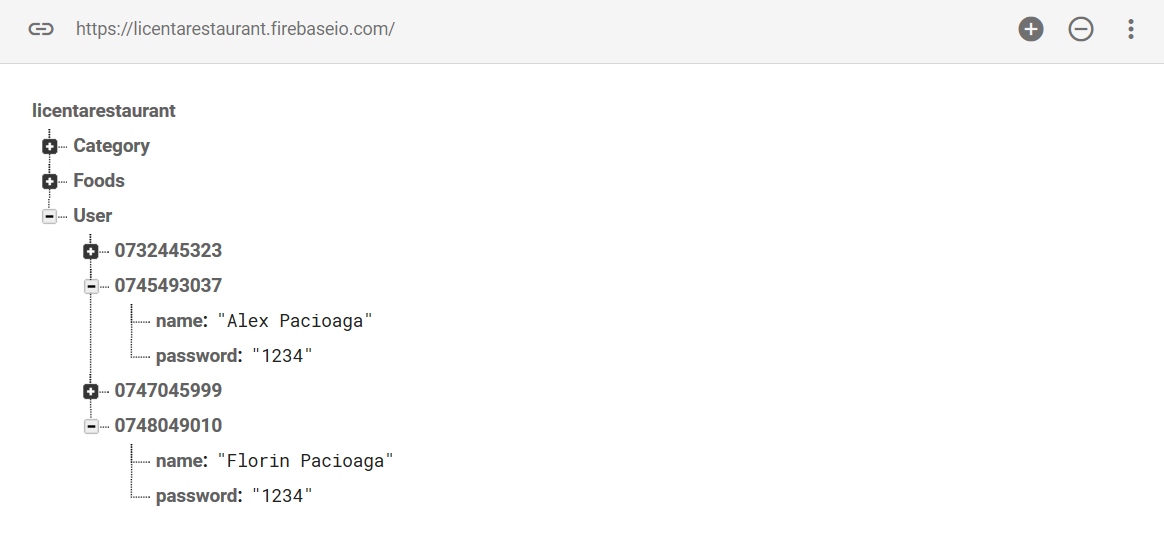
• Fără griji cu privire la infrastructură

Din meniul urmator se poate accesa rapid fiecare dintre caracteristicile bazei de date:



*Figura 2.1 – Bara de meniuri*

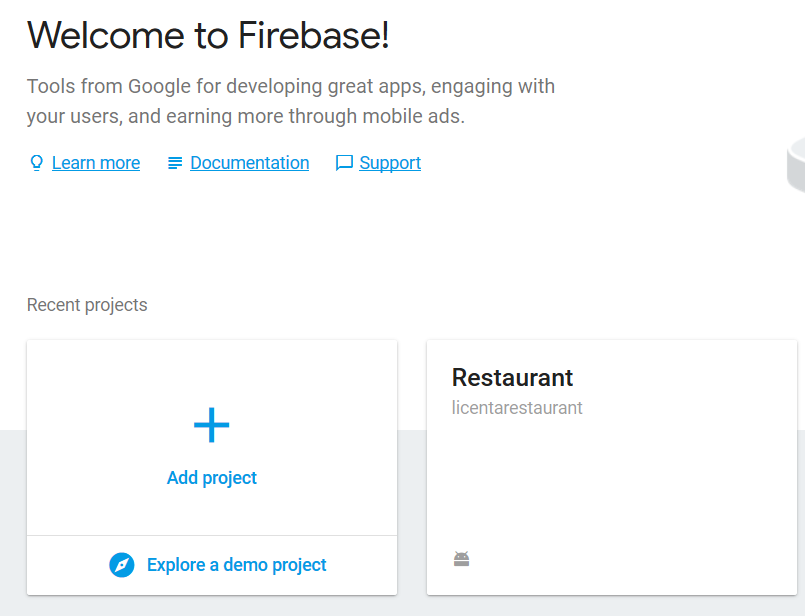
În secțiunea „Database” se poate monitoriza fiecare membru nou înregistrat împreuna cu informațiile legate de contul acestuia:



*Figura 2.2 – Date inserate în baza de date*

[**2.2.2. Adăugarea**](#page14) **Firebase în proiectul Android**

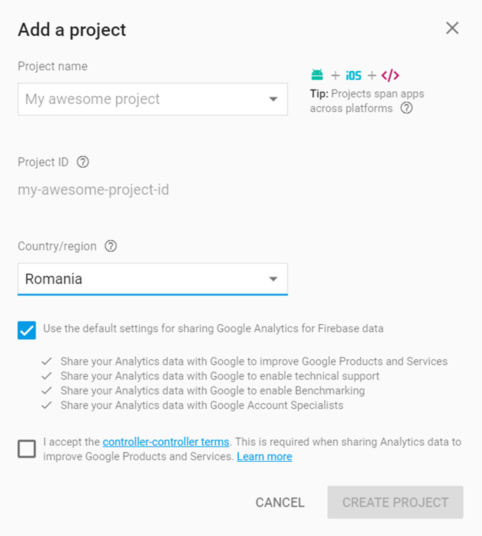
Pentru a adauga Firebase in proiectul nostru ne trebuie, in primul rand, un proiect Firebase nou creat si un fisier cu rol de a configure aplicatia mobile. Pentru a crea un proiect Firebase ne trebuie deschisa consola Firebase precum in urmatoarea figura: [2] [3]



*Figura 2.3 – Creare proiect Firebase 1*

Pentru a adăuga Firebase în aplicație, se va crea un proiect Firebase apăsând pe *„*Add project”.

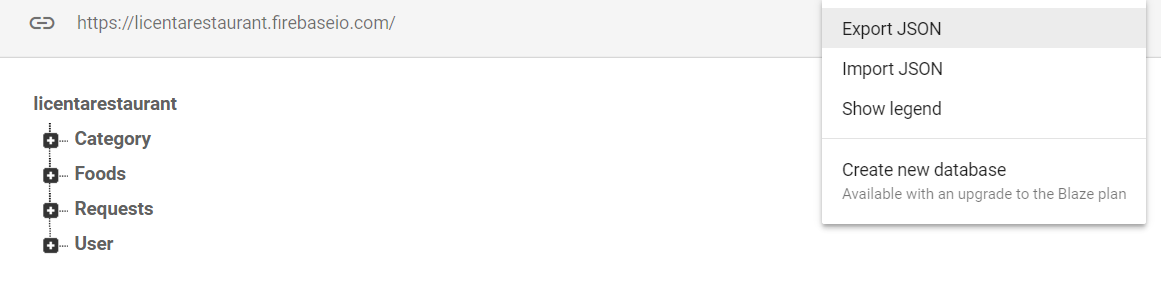
Când vi se solicită, introduceți numele pachetului și id-ul proiectului. Este important să introduceți numele id-ului pe care îl folosește aplicația, acest lucru putând fi setat doar atunci când se creează un nou proiect Firebase:



*Figura 2.4 – Creare proiect Firebase 2*

Dupa ce a fost creat, proiectul Firebase se găsește în consola, proiectul meu creat fiind numit *„*Restaurant*”*. Dacă există deja un proiect Google asociat aplicației mobile, se va selecta *„*Import Google Project”.

În final, după ce am terminat cu crearea proiectului, trebuie descărcat un fișier google-services.json. Acest fișier se poate descărca din nou, oricând este nevoie apăsând pe butonul „Export JSON”*:*



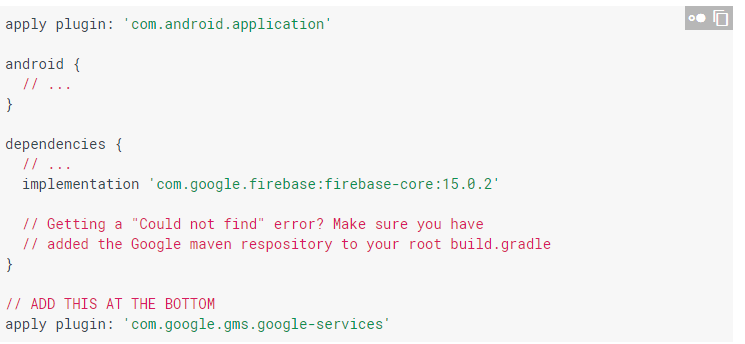
*Figura 2.5 – Descărcarea JSON a bazei de date*

Adaugarea SDK-ului reprezintă o parte importantă din inițializarea proiectului, așadar pentru pentru a integra bibliotecile Firebase într-unul dintre proiectele proprii, trebuie să se efectueze câteva sarcini de bază pentru pregătirea proiectului din Android Studio.

Mai întâi, trebuie adăugate regulile de bază în fișierul build.gradle care stă la baza proiectului din Android Studio, incluzând pluginul de servicii Google și depozitul Maven Google:

 *Figura 2.6 – Includerea plugin-ului de servicii Google 1*

Apoi, în modulul Gradle (de obicei, app/build.gradle), adăugați linia de plugin aplicată în partea de jos a fișierului pentru a activa pluginul Gradle:

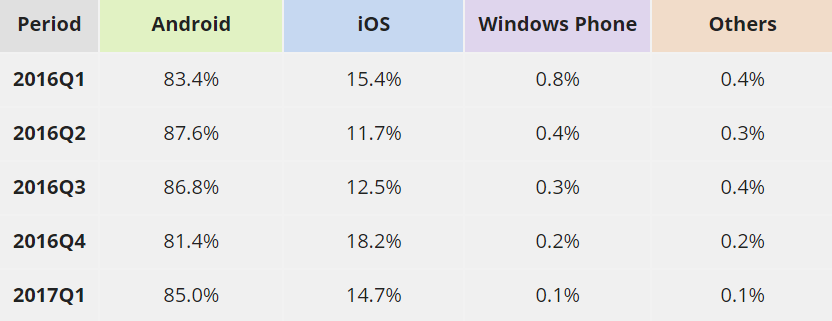


*Figura 2.7 – Includerea plugin-ului de servicii Google 2*

**2.2.3. Android si IOS**

În prezent, pe piața telefoanelor inteligente, 2 sisteme de operare iși fac simțită prezența și anume Android si IOS.

În tabelul următor sunt prezentate principalele sisteme de operare, adică Android, IOS, Windows Phone si cele care sunt în altă categorie, aflate pe piață si distribuția lor pe terminalele mobile:



*Figura 2.8 – Distribuția sistemelor de operare pentru telefoane mobile la nivel global.*

După cum se poate observa, platforma Android conduce detașat deținând 85 % din piața telefoanelor mobile în primul sfert din anul 2017. Cu toate acestea sistemul IOS nu trebuie neglijat el fiind majoritar pe piață în Statele Unite, Canada, Norvegia, Danemarca [4].

[**Capitolul 3. Prezentarea Aplicației**](#page25)

**3.1. Scopul aplicației**

Aplicația a fost creată cu scopul de a ușura munca oamenilor de a comanda de acasă mâncarea preferată din cadrul unui restaurant, nefiind obligați sa fie puși pe drumuri. Prin această aplicație, utilizatorul iși va crea un cont prin intermediul unui număr de telefon unic si a unei parole la alegere, neexistând constrângeri asupra acesteia.

În al doilea rând, în ziua de astăzi, a avea un **site funcţional** este util, dar nu este de ajuns. Acesta trebuie **optimizat** astfel încât să corespundă nevoilor de promovarea a restaurantului dar şi exigenţelor clienţilor. Scopul principal al unui site este atragerea unui număr mai mare de clienţi decât o face competiţia.

Fă în așa fel încât cititorii să găsească foarte ușor informațiile căutate pe site și pune accent pe **conținutul vizual**.

Se pare că majoritatea lanţurilor de restaurante au conştientizat importanţa unei mobile app, întrucât au început să dezvolte campanii de marketing mobile, targetate geografic, oferind totodată şi posibilitatea de a plasa comenzi de pe mobil.

**3.2 Prezentarea interfețelor**

Pentru dezvoltarea aplicației am folosit ca mediul de dezvoltare Android Studio, iar ca limbaj de programare Java. Aplicația este dezvoltată exclusiv pentru platforma Android. Telefonul folosit pentru programare este Samsung Galaxy S8+ insă puteam folosi orice alt telefon care rulează pe sistemul de operare Android și care îndeplinește condițiile menționate în Capitolul 1.

Atunci când utilizatorul va intra in aplicație, primul meniu pe care îl va intâlni este cel din figura 3.1:



*Figura 3.1 – Meniul de start al aplicației*

Pe această pagină avem un mesaj cu rolul de a influența utilizatorul spre a duce o viață cât mai sănătoasă, o imagine de fundal descriptivă si cele două butoane prin care un utilizator iși poate creea un cont nou prin intermediul butonului „SIGN UP” și de asemenea să se logheze mai apoi prin intermediul butonului „SIGN IN”.

În continuare, accesând buton „SIGN UP”, utilizatorul va fi nevoit sa completeze cele trei câmpuri cu date sugerate, adică numărul de telefon, numele și o parolă:

*Figura 3.3 – Meniul de inregistrare cont nou*

După completarea datelor, utilizator va putea crea contul prin apăsarea butonului din partea de jos a ecranului iar dacă înregistrarea a avut loc cu succes se va afișa mesajul „Sign up succesfully!”:

*Figura 3.3 – Rezultatul înregistrării cu succes*

În cazul in care numărul de telefon a fost folosit anterior în crearea unui cont, utilizatorul va fi atenționat prin mesajul „Phone number already registered”:

*Figura 3.3 – Rezultatul înregistrării cu succes*

Apoi, utilizatorul poate să revină în meniul de logare și să se logheze cu contul recent creat, având aici la dispoziție o opțiune suplimentară care poate fi folosită doar atunci cand este în meniul de logare „Remember me”. Bifând aceasta casuță utilizatorul nu va mai fi nevoit să se relogheze in cont dupa ce a închis aplicația din proces, datele acestuia fiind salvate si folosite la următoarea intrare în aplicație:

*Figura 3.3 – Rezultatul înregistrării cu succes*

Următoarea pagină a aplicației este reprezentată de meniul restaurantului unde putem vedea si alege din categoriile de mâncare care sunt disponibile clienților:

*Figura 3.3 – Rezultatul înregistrării cu succes*

De aici, clientul se poate orienta după categoriile sale preferate de mâncare și să aleagă una dintre cele afișate pe ecran:

*Figura 3.3 – Rezultatul înregistrării cu succes*

Dupa alegerea unei categorii, se va afișa urmatoarea pagină cu mâncarea specifică din categoria aleasă anterior de catre client și de unde acesta are la dispoziție două funcționalități, o bară de căutare de unde își poate căuta un preparat culinar preferat iar dacă acesta se gasește în meniul restaurantului va fi afișat ca sugestie iar clientul printr-un click își va selecta preparatul, iar o altă funcționalitate dispusa clientului este aceea de a putea adăuga mâncarea sa în lista de preferate iar când va accesa ulterior meniul restaurantului va putea să vadă mai ușor unde se găsește mâncarea sa preferată:

*Figura 3.3 – Poza meniu mancare*

*Figura 3.3 – Poza bara de search a*

*Figura 3.3 – Poza bara de search b*

*Figura 3.3 – Poza adaugare la preferate*

După ce clientul se va hotărî in privința mancării și merge mai departe, i se va deschide o nouă pagină în care

5. Bibliografie

[1] <http://unix.stackexchange.com/questions/119128/linux-mint-16-android-device-not-listed-with-lsusb> (Ultima accesare: 25.02.2018)

[2] Android Developers: http://developer.android.com/guide/basics/what-is-android.html, (Ultima accesare: 25.02.2018)

[3] <https://firebase.google.com/docs/android/setup> (Ultima accesare: 28.02.2018)

[4] Sursa Internation Data Corporation ([www.idc.com](http://www.idc.com/))