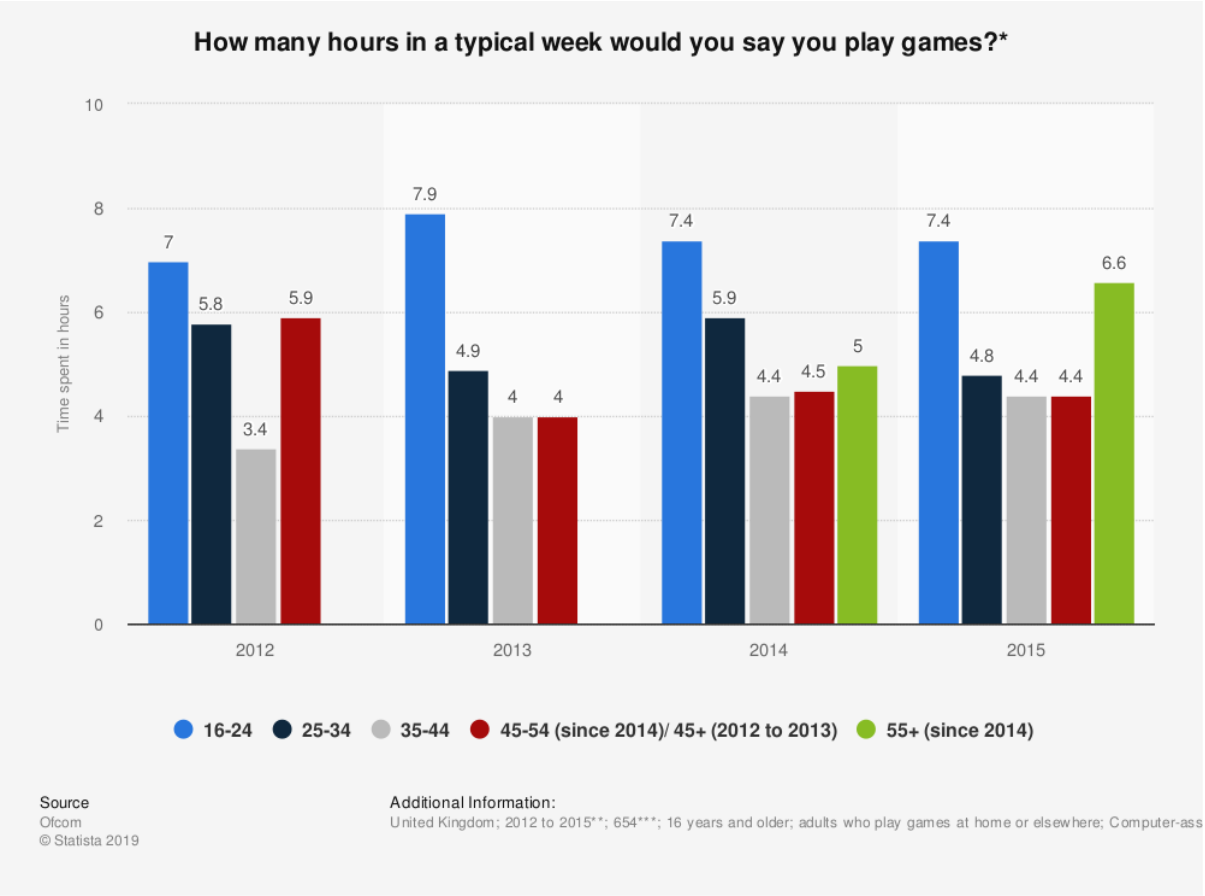
Dezvoltarea aplicatiilor mobile in android

Documentatie proiect

Cuprins

1. Introducere

Jocurile video au devenit o activitate preferata atat de copii, cat si de adulti in zilele noastre. Acestea au inlocuit in mare parte joaca in aer liber cu mingea, acum tinerii preferand sa stea in casa in fata calculatorului si sa joace un “Fifa” in loc sa iasa afara. Acest lucru este putin ingrijorator, dar lumea se schimba si trebuie sa ne adaptam la timpurile in care traim. Astfel pentru acest proiect am decis sa creez un joc video in Android Studio pentru a satisfice dorinta celor care vor sa isi petreaca timpul liber online.



Dupa cum se poate observa nu doar tinerii, ci si oamenii mai in varsta de 55 de ani (probabil pensionari cu mai mult timp liber) petrec mult timp jucand jocuri video.

Jocul creat de mine cauta in general o audienta mai tanara, acesta avang un concept destul de copilaresc. Am decis sa fac un joc in stil “catroon” cu poze desenate pentru a fi pe placut juniorilor dornici de joaca. Asta nu inseamna ca adolescentii sau adultii nu se pot bucura de acest joc, jocurile de genul avand in general o audienta diversificata. Jocul este “family friendly”, adica poate fi jucat de toate categoriile de varsta deoarece nu are continut neadecvat unei anumite categori.

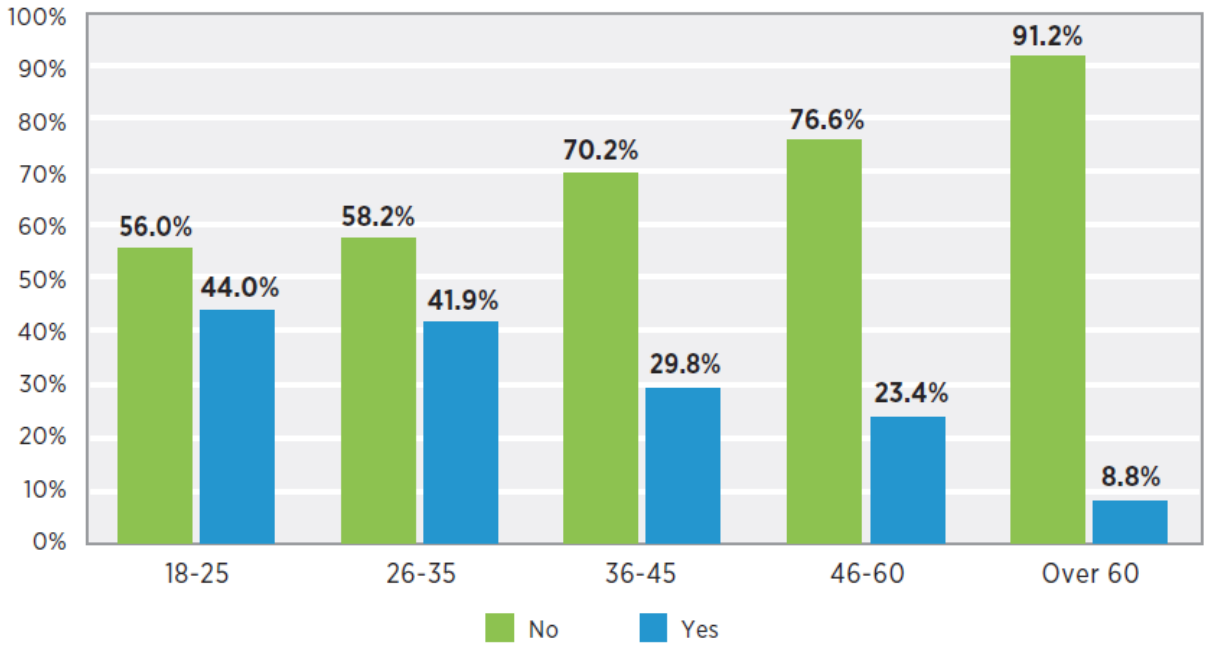
Am ales sa fac un joc deoarece si eu sunt o persoana careia ii place sa isi petreaca timpul liber jucandu-ma pe calculator si de mult timp am fost interesat de dezvoltarea de jocuri video. Jucatul jocurilor video a fost mereu un hobby pentru mine, inca de cand eram mic. Acum ca pot sa creez jocuri video eu insumi, am vrut sa o fac pentru a simti si eu cum e sa creez ceea ce mi-a placut de mult.

De asemenea multa lume se duce spre industria jocurilor video deoarece este una foarte larga cu mult potential. Unele persoane aleg sa creeze jocuri video ca si "game developer" sau sa testeze jocurile create de altii (game tester) pentru a gasi "bug-uri", astfel creatorii pot incerca sa le rezolve odata depistate pentru ca restul utilizatorilor sa aiba o experienta cat de placuta.

O alta ramura a acestei industrii este partea de streaming. Multe persoane au inceput canale de Youtube sau Twitch, aceste site-uri prezentand un software care ii ajuta pe utilizatori sa filmeze "gameplay-uri" si sa le incarce pe platforma. Odata incarcat restul utilizatorilor (consumatorii) pot viziona aceste videoclipuri pentru amuzament si eventual ajutor. Persoanele care incarca aceste videoclipuri pot lega parteneriate cu companii sau alte persoane de pe acea platforma, astfel pot primi bani pentru vizionarile pe care le strang pe aceste videoclipuri. Astfel un "content creator", cum sunt numite aceste persoane, poate practica aceasta profesie full time, avand un venit stabil si posibilitatea de a se dedica in totalitate acestei activitati.

Alta posibilitate la o cariera legata de jocurile video este optiunea de a deveni un jucator profesionist. Un jucator profesionist este o persoana care aloca foarte mult timp jocurilor video, astfel aspirand la o performanta deosebita. Dupa antrenament si multa dedicatie, unii jucatori care au talent spre asa ceva au oportunitatea sa fie cautati si contractati de echipe profesioniste. Acestia vor participa in turnamente numite E-sports si au sansa sa castige premii de pana la cateva milioane de dolari americani. Acest tip de "sport" este in general practicat de jucatorii mai tineri, fiind mai popular in aceasta categorie de varsta, dar asta nu inseamna ca nu este posibil ca un jucator mai in varsta sa participe in aceste turnamente.

Jucatori de jocuri video de diferite varste au fost intrebati daca ar vrea sa isi faca o cariera ca jucator profesionist:



Would you quit your job if you could support yourself as a professional video game player?

2. Despre joc

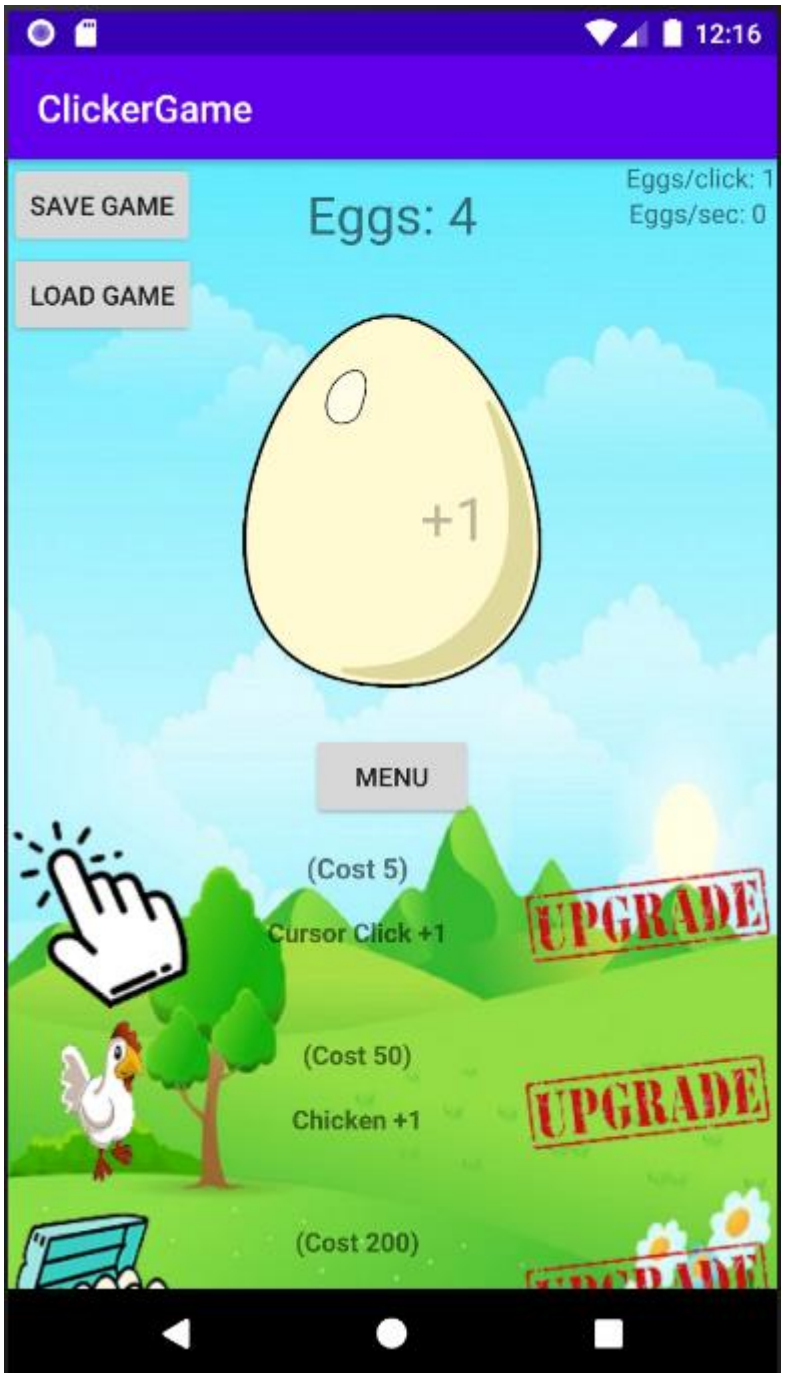
Tipul jocului se numeste “Clicker” si este un joc simplist. Scopul jocului este sa dai click pe un obiect pentru a face puncte sau bani. In acest joc obiectul este un ou si prin apasarea pe acest ou primesti mai multe oua. Ouale primite sunt apoi folosite pentru a cumpara upgrade-uri din meniul deschis din partea de jos a ecranului, upgrade-uri care te ajuta sa progresezi mai usor in joc. Aceste upgrade-uri constau in mai multe oua primite pe click astfel devenind mai usor sa faci mai multe puncte. Un alt tip de upgrade-uri care vor fi deblocate sunt cele pentru venit pasiv. Asta inseamna ca odata cumparate, acestea vor genera un venit pasiv pe secunda, primele unul mai mic, iar odata cu deblocarea celorlalte creste si venitul. Acest venit pasiv, initial fiind 0, creste odata cu cumpararea gainilor, cartoanelor cu oua si coteturilor.

Am folosit poze tip desene animate pentru a avea un aspect prietenos si placut pentru jucatorii de o varsta frageda. Aspectul care l-am cautat a fost unul colorat si cald, astfel copiii se pot bucura atat de interactivitatea jocului cat si de aspectul frumos si familiar, intalnit in desenele animate vizionate la televizor.

Pentru a putea continua jocul de unde a fost inchis am creat un sistem de save/load. Cand urmeaza sa se inchida jocul, butonul de save va deschide un nou meniu unde trebuie

introdus un nume de utilizator si o parola. In functie de acestea, utilizatorul poate apasa butonul de load, unde se introduc acele nume si parola pentru a relua jocul de unde a ramas cand a iesit.

Jocul nu are un final, acesta poate fi jucat la nesfarsit. Jucatorul poate sa continue sa cumpere upgrade-uri si sa creasca scorul fara o anumita limita. Pretul acestora creste odata cu cumpararea lor, iar odata cu acesta creste si bonusul de venit primit de la fiecare tip de upgrade.



3.Detalii tehnice

Jocul a fost creat folosind Android Studio in limbajul de programare JAVA. Am folosit Firebase pentru a creea o baza de date unde stochez datele utilizatorilor pentru a putea incarca jocul dupa ce acesta este inchis. Jocul este conectat la aceasta baza de date printr-un proiect server creat tot in Android Studio si scris in limbajul de programare Kotlin.

3.1 Jocul

Activitatea principala reprezinta meniul jocului, unde am pus un background negru, o imagine semnificativa si butonul de incepere al jocului care porneste activitatea noua (Game).

In aceasta activitate avem in fundal o imagine cu un camp deschis desenat in still "cartoon" si am introdus imaginea oului cu o animatie de rescale pentru a da un effect de "tap" cand este apasat pe ou. Imaginea are un OnClick care declanseaza actiunea odata apasat pe ea.

Xml-ul animatiei de rescale folosita la ou:

```
<?xml version="1.0" encoding="utf-8"?>
<scale xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="50"
    android:fromXScale="1"
    android:fromYScale="1"
    android:toXScale="0.8"
    android:toYScale="0.8"
    android:pivotX="50%"
    android:pivotY="50%"
    android:repeatMode="reverse"
    android:repeatCount="1"
/>
```

La apasarea oului apare pe ecran un TextBox cu o animatie de FadeIn si FadeOut care afiseaza numarul de puncte acumulat la fiecare click. Acesta apare si dispare dupa un scurt timp si este mutat la coordonate intamplatoare deasupra oului de fiecare data cand este facut un click pentru a da un efect interesant.

```
private void eggClick() {
    point+=click;
    points.setText("Eggs: "+Integer.toString(point));
    pointsanim.setX(rand.nextInt(300) + 500);
    pointsanim.setY(rand.nextInt(150) + 500);
    defaultButtonAnimation();
    System.out.println("Egg Clicked");
    pointsanim.setText("+"+click);
    pointsanim.setVisibility(View.VISIBLE);
    pointsanim.setAnimation(animation);
    if(animation.hasEnded())
        pointsanim.clearAnimation();
    pointsanim.setVisibility(View.INVISIBLE);
}
```

```
private void defaultButtonAnimation(){
    Animation fadeIn = new AlphaAnimation(0, 1);
    fadeIn.setInterpolator(new DecelerateInterpolator());
    fadeIn.setDuration(300);

    Animation fadeOut = new AlphaAnimation(1, 0);
    fadeOut.setInterpolator(new AccelerateInterpolator());
    fadeOut.setStartOffset(300);
    fadeOut.setDuration(300);

    animation = new AnimationSet(false);
    animation.addAnimation(fadeIn);
    animation.addAnimation(fadeOut);
}
```

Aici am adaugat ca exemplu codul pentru animatia de FadeIn si FadeOut mentionata mai sus. Prima funtie reprezinta functia apelata din OnClick-ul imaginii cu oul si este folosita pentru a gestiona TextBox-urile afectate si pentru a apela functia defaultButtonAnimation care la randul ei seteaza animatia.

Avem cateva TextBox-uri care afiseaza ouale (scorul curent), ouale primite pe fiecare click si ouale primite pe secunda de la venitul pasiv. Doua butoane in coltul din stanga sus al ecranului sunt folosite pentru a salva si incarca progresul. Odata apasate acestea deschid un layout care era inainte ascuns in care se cere introducerea unui nume de utilizator si a unei parole in doua EditText, iar apoi un buton care odata apasat verifica si trimite/cere informatia de la baza de date prin intermediul serverului. In coltul din dreapta sus al Layout-ului avem un ‘x’ care odata apasat il va inchide fara sa trimita nimic catre baza de date.

In partea de jos a ecranului avem un buton de meniu care odata apasat va deschide un ScrollView in care apar toate upgrade-urile posibile. In acesta avem in partea stanga o serie de imagini care reprezinta cate un upgrade fiecare. In mijloc avem niste TextBox-uri in care este afisat costul upgrade-ului si numele sau, sau mai exact beneficiul primit cu achizitionarea upgrade-ului. Textul acestora se va schimba in concordanta cu upgrade-urile cumparate pentru a arata utilizatorului noul pret si noul bonus odata cu deblocarea. In partea dreapta este cate o imagine pentru fiecare upgrade. Acestea au legat un OnClick care va verifica odata cu apasarea imaginilor daca utilizatorul are destule puncte pentru a putea debloca upgrade-ul urmator. In cazul in care punctele sunt suficiente upgrade-ul va fi deblocat, iar in caz contrar un TextBox cu aceeasi animatie de FadeIn si FadeOut folosita pentru TextBox-ul care arata punctele la apasarea oului va aparea pentru a anunta utilizatorul ca achizitia nu poate fi finalizata deoarece

jucatorul nu detine destule puncte. Acelasi tip de animatie de rescale folosita pentru ou a fost refolosita pentru imaginile de upgrade.

Odata cu deblocarea unui upgrade de venit pasiv, la inchiderea meniului se poate observa ca imaginea acelui upgrade apare in fundal, fapt care semnifica deblocarea acestuia. Initial aceste imagini sunt ascunse si devin vizibile doar in cazul in care primul nivel al unui upgrade a fost deblocat.

Conectarea la server din aplicatie este realizata intr-un Async Task, loc in care este initializat portul si IP-ul socket-ului si definite metodele de scriere si citire din server.

```
new AsyncTask<Void, Void, Void>() {
    @Override
    protected Void doInBackground(Void... voids) {
        try {
            socket = new Socket("192.168.2.100", 2557);
            System.out.println("Connected to Server");
            System.out.println(socket.toString());
            send = new PrintWriter(new BufferedWriter(new
OutputStreamWriter(socket.getOutputStream())), true);
            get = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
```

3.2 Serverul

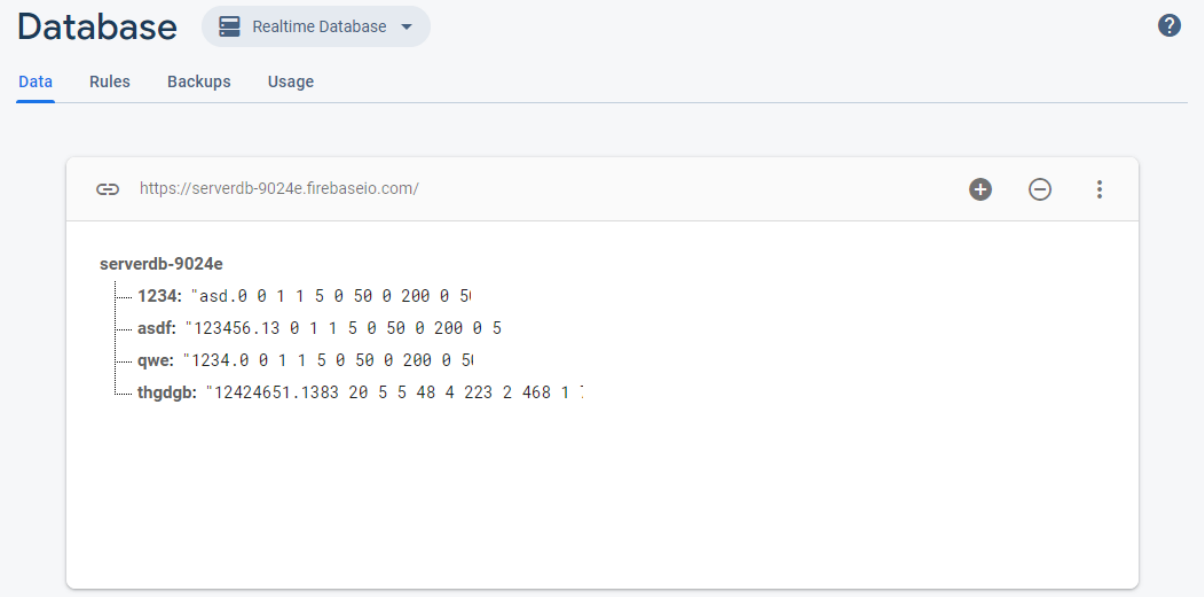
Proiectul pentru server consta intr-o singura activitate in care se initializeaza socket-ul cu acelasi port ca si cel din joc, apoi se stabileste referinta la baza de date din Firebase.

Dupa ce totul a fost initializat, se creeaza un thread in cadrul caruia serverul continua sa astepte informatii primite de la joc. Jocul va trimite un buffer cu o comanda “Save Game” si un string care va trebui prelucrat si introdus in baza de date in cazul in care vrem sa salvam jocul, iar in cazul in care vrem sa incarcam jocul comanda va fi “Load Game”, dupa care urmeaza numele de utilizator si parola introduse de jucator.

```
        val reader = BufferedReader(InputStreamReader(client.getInputStream()));
        while (true) {
            val read = reader.readLine();
            val line = read.toString();
            System.out.println(line);
            val endindexCommand = line.indexOf('*');
            val command = line.substring(0, endindexCommand);
            if (command.compareTo("Save Game") == 0) {
                val endindexName = line.indexOf(',');
                val username = line.substring(endindexCommand+1, endindexName);
                val score = line.substring(endindexName+1, line.length);
                databaseReference.child(username).child("").setValue(score);
            }
```

3.3 Firebase

In cadrul bazei de date avem doar o simpla ierarhie unde numele de utilizator este cheia principala. Fiecare utilizator are o parola folosita pentru inregistrare, dupa care urmeaza un sir de numere care semnifica scorul, upgrade-urile cumparate si costurile acestora.



Apare numele de utilizator primul ca si cheie principala, iar apoi avem parola, urmata de scoruri si upgrade-uri. Parola este separata de restul printr-un ‘.’(punct), iar restul de date sunt separate unul de altul printr-un ‘ ‘(spatiu).

4.Incheiere

In final am dedus ca creearea acestui proiect a fost o experienta interesanta si educativa. Am lucrat cu placere si sunt multumit de ceea ce am realizat. Nu este perfect, dar este suficient din punctul meu de vedere pentru timpul care l-am alocat. Proiectul a fost incarcat pe GitHub pentru vizionare ulterioara.

5.Bibliografie

https://firebase.google.com

https://github.com

https://stackoverflow.com

https://developer.android.com