APLICAȚIE ANDROID CA FACILITATOR DE COMUNICARE

PROIECT DE DIPLOMĂ

Autor: **Tatiana DOLJIKOV**

Conducător științific: **Enyedi SZILÁRD**

|  |  |
| --- | --- |
| DECAN  **Prof.dr.ing. Liviu MICLEA** | Vizat,  DIRECTOR DEPARTAMENT AUTOMATICĂ  **Prof.dr.ing. Honoriu VĂLEAN** |

Autor: **Tatiana DOLJIKOV**

Aplicație ANDROID ca facilitator de comunicare

1. **Enunţul temei:** *O scurtă descriere a temei proiectului de diplomă*

Piaţa aplicaţiilor este într-o continuă dezvoltare existând mai multe sisteme de operare ce ne stau la îndemână iar pentru deyvoltatori există o întrebare critică de început – ce platform ar trebui să fie utilizată pentru implementarea unei idei? Pentru companiile mai mari, cu mai multe resurse, dezvoltarea se poate face simultan pentru diferite platforme, în timp ce pentru magazinele mai mici, este o întrebare foarte critică, care ar putea determina succesul sau eșecul afacerii. Aplicaţiile de tip ANDROID sunt tot mai des folosite în îmbunătăţirea diferitor sectoare ale industriei. Fiind o platformă software şi un sistem de operare pentru dispozitive şi telefoane mobile, aplicaţiile Android sunt scrise în limbajul de programare Java, iar independența hardware-ului l-au clasat ca şi lider în acest domeniu ce este în creștere rapidă şi continuă..

1. **Conţinutul proiectului:** *(enumerarea părților componente) Pagina de prezentare, Declarație privind autenticitatea proiectului, Sinteza proiectului, Cuprins, Titlul capitolului 1, Titlul capitolului 2,… Titlul capitolului n, Bibliografie, Anexe.*
2. **Locul documentaţiei:** *Universitatea Tehnică din Cluj-Napoca*
3. **Consultanţi:**
4. **Data emiterii temei:**
5. **Data predării:**

Semnătura autorului

Semnătura conducătorului științific

**Declaraţie pe proprie răspundere privind**

**autenticitatea proiectului de diplomă**

Subsemnatul(a) **Tatiana DOLJIKOV** , legitimat(ă) cu \_\_BI seria CJ nr. 208975 , CNP 2900625807752,

autorul lucrării:

Tatiana Doljikov

elaborată în vederea susţinerii examenului de finalizare a studiilor de masterat

la **Facultatea de Automatică și Calculatoare**,

specializarea **Automatică și Informatică Aplicată,**

din cadrul Universităţii Tehnice din Cluj-Napoca,

sesiunea Iulie 2017 a anului universitar 2016-2017,

declar pe proprie răspundere, că această lucrare este rezultatul propriei activităţi intelectuale, pe baza cercetărilor mele şi pe baza informaţiilor obţinute din surse care au fost citate, în textul lucrării, şi în bibliografie.

Declar, că această lucrare nu conţine porţiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislaţiei române şi a convenţiilor internaţionale privind drepturile de autor.

Declar, de asemenea, că această lucrare nu a mai fost prezentată în faţa unei alte comisii de examen de licenţă.

In cazul constatării ulterioare a unor declaraţii false, voi suporta sancţiunile administrative, respectiv, *anularea examenului de licenţă*.

Data Prenume NUME

(semnătura)

**SINTEZA**

proiectului de diplomă cu titlul:

Aplicație ANDROID ca facilitator de comunicare

Autor: **Tatiana DOLJIKOV**

Conducător științific: **Titlu.ing. Enyedi SZILÁRD**

1. Cerinţele temei: Dezvoltarea unei aplicaţii traducator ce va ajuta utilizatorii în învăţarea şi cunoaşterea unor expresii tipizate în limba rusă cu o traducere din limba engleză.

2. Soluţii alese: Aplicaţia a fost dezvoltată cu ajutorul platformei software Android în limbajul de programare Java - în mediul de dezvoltare integrat Android Studio. Ca şi sistem de automatizare a surselor a fost utlizat Gradle iar pentru integrarea translatorului online s-a folosit introducerea de dependinţe, în caz particular dependinţa pentru Microsoft Translation.

3. Rezultate obţinute: Rezultatul îl reprezintă o aplicaţie Android denumită “Russian Translator” ce curpinde 5 categorii tipizate ce pot fi accesate din pagina principal la care se mai adaugă un “floating action bar” ce permite utilizatorului să introducă textul ce se doreşte a fi tradus, atât din engleză în rusă, cât şi din rusă în engleză. Deasemenea în cadrul acestei activităţi utlizatorul mai poate solicita şi opţiunea vocală a textului tradus.

4. Testări şi verificări: Testările şi verificările s-au făcut în modul Debug şi pe un telefon de tip Huawei P8 Lite.

5. Contribuţii personale:

6. Surse de documentare:

Semnătura autorului

Semnătura conducătorului științific

Cuprins

[1 Introducere 2](#_Toc477457095)

[1.1 Context general 2](#_Toc477457096)

[1.2 Obiective 2](#_Toc477457097)

[1.3 Specificații 2](#_Toc477457098)

[2 Studiu bibliografic 3](#_Toc477457099)

[3 Analiză, proiectare, implementare 4](#_Toc477457100)

[4 Concluzii 5](#_Toc477457101)

[4.1 Rezultate obținute 5](#_Toc477457102)

[4.2 Direcții de dezvoltare 5](#_Toc477457103)

[5 Reguli de formatare 6](#_Toc477457104)

[5.1 Formatarea paginii 6](#_Toc477457105)

[5.2 Titluri și stiluri 6](#_Toc477457106)

[5.3 Figuri, tabele și ecuații 7](#_Toc477457107)

[5.3.1 Figuri 7](#_Toc477457108)

[5.4 Tabele 7](#_Toc477457109)

[5.5 Ecuații 7](#_Toc477457110)

[5.6 Referințe bibliografice 8](#_Toc477457111)

[6 Bibliografie 9](#_Toc477457112)

# Introducere

## Context general

În cazul în care cineva cunoaşte o limbă mai exotică sau mai diferită de limba română ca şi alfabet şi vorbire, foarte des am auzit întrebarea « Spune ceva » sau « Zi-mi câteva cuvinte ». Anume această idee a stat la baza dezvoltării aplicaţiei Android prezentate în această lucrare, numit- « Russian Translator ».

Aplicaţia are la bază două limbi – limba engleză care este o limbă folosită intensiv ca [limbă secundară](https://ro.wikipedia.org/wiki/Limb%C4%83_secund%C4%83) sau ca [limbă oficială](https://ro.wikipedia.org/wiki/Limb%C4%83_oficial%C4%83) de-a lungul lumii, şi limba rusă care este scrisă în  [alfabetul chirilic](https://ro.wikipedia.org/wiki/Alfabetul_chirilic) şi care este vorbită de doar 7% din populaţia Uniunii Europene. Scopul acestei aplicaţii este de a ajuta utilizatorul să se familiarizeze cu nişte cuvinte cheie ale limbii ruse, să o poată recunoaşte atunci când este auzită şi bineînţeles să fie de ajutor în cazul unei călătorii într-una dintre ţările vorbitoare de limba rusă.

Aplicaţia Android dezvoltată are drept scop prezentarea unor cuvinte cheie care sa fie usor de memorat pentru cititor, acesta având posibilitatea să vadă partea scrisă a cuvântului sau expresiei de cuvinte precum şi să audă pronunţia acestuia.

Deşi exisă pe piaţă aplicaţii traducător pentru diferite limbi, ceea ce a stat la baza dezvoltării acesteia şi ceea ce ar putea să o diferenţieze de celelalte aplicaţii este simplitatea design-ului, uşurinţa utilizării şi navigării.

*In introducere familiarizați cititorul cu motivația lucrării, plasați lucrarea într-un context care să permită cititorului să înțeleagă obiectivele.*

*Descrieți importanța lucrării, de ce merita să o faceți, plasati ideile într-un context larg.*

*Sustineți studiul: de ce exact aceasta aplicație/implementare. Comentați asupra aspectelor teoretice sau practice care v-au facut să o alegeți.*

*In funcție de natura lucrării, ar putea fi necesar să prezentați informații de fond asupra domeniului în care se încadrează aplicația, mai ales dacă aveti o lucrare într-un domeniu multidisciplinar. In acest caz, puteți introduce terminologia pe care o utilizați în continuare.*

*Descrieti pe scurt lucrarea: ce conține fiecare capitol.*

## Obiective

Obiectivul principal ce a stat la baza dezvoltării aplicaţiei a fost faptul că pe piaţă există déjà aplicaţii ce ne ajută să traducem totul din diferite limbi, dar foarte puţine sunt specific unei singure limbi şi foarte puţine ne oferă şi nişte cunoştinţe de bază, în special pentru acele limbi în cae pronunţia sau scrierea diferă di limba engleză.

În acest sens am folosit un Translator oferit de Microsoft Azure care ne-a permis să integrăm partea de traducere personalizată după utilizator, atât la partea scrisă cât şi la partea de pronunţie, audio prin intermediul unui MediaPlazer pentru fiecare cuvânt sau expresie şi imagini pentru o memorare mai uşoară a acestor expresii.

*Enumerati și explicati obiectivele lucrării: ce v-ați propus să realizați în contextul prezentat anterior.*

*Obiectivele pot fi prezentate sub formă de listă care să evidențieze precis orientarea lucrării, să identifice conceptele fundamentale pe care le studiati, să stabilească scopul aplicației pe care o realizați, sau enuntați întrebarile la care intentionați să răspundeți în lucrare.*

## Specificații

La pornirea aplicaţiei ajungem pe pagina principal unde sunt afişate 5 categorii de cuvinte - Alfabet, Numere, Membri de familie, Culori şi Fraze, toate marcate cu diferite culori. Fiecare dintre aceste categorii poate fi accesată cu o singură atingere, aplicaţia redirecţionându-ne către pagina specific fiecărei categorii. Deasemenea, o parte important a acestei pagini este butonul “fab” care la atingere ne redirecţionează către traducătorul personalizat.

Pentru prima parte a plicaţiei, informaţia este citită prin intermediul unor fişiere XML ce conţin informaţii tag-uri specific fiecărei categorii necesare afişării celor 5 activităşi de pe pagina principal, şi anume – textul în engleză, textul în rusă, codul pentru media player şi codul imaginii pentru activităţile în care acestea sunt afişate.

Ca şi sistem de automatizare am folosit Gralde, fiind unul dintre cele mai renumite şi utilizate sisteme pentru limbajul de programare Java şi fiind folosit la build, packaging şi lansarea (deploy) codului. Pentru a putea integra în aplicaţia funcţionalitatea de traducător online, am adăugat o dependinţă şi drepturi de Internet în cadrul aplicaţiei.

*In specificațiile lucrării detaliați cerințele. Descrieți ce intentionați să obtineți. Vă puteți referi la funcțiile aplicației, interfață, nivele de performanță, structuri de date, elemente, securitate, fiabilitate, calitate, limitări, etc.*

# Studiu bibliografic

Conține o analiză a ceea ce s-a realizat/studiat anterior. Arătați că ați studiat materiale bibliografice și că ați înteles ceea ce ați citit.

Puteti include diferite puncte de vedere asupra problemei pe care o rezolvați în lucrare.

Nu uitați să citați corespunzător autorii oricărei idei extrase dintr-o sursă bibliografică.

# Analiză, proiectare, implementare

Pentru a putea dezvolta o aplicaţie de tip android, în primul rând ne trebuie mediul de dezvoltare integrat Android Studio ce are următoarele caracteristici:

* gradul de sprijin bazat pe construire ;
* refactorizarea codului specifică Android și remedierea rapidă a erorilor ;
* instrumente pentru îmbunătăţirea performanței ;
* un editor de layout bogat, care permite utilizatorilor să manipuleze componentele de UI (User Interface) mult mai uşor şi mai rapid ;
* opțiunea de a previzualiza layout-urile pe mai multe configurații de ecran.
* suport încorporat pentru platforma Google Cloud Platform, care permite integrarea cu Firebase Cloud Messaging ("Google Cloud Messaging") și Google App Engine ;
* dispozitiv virtual Android (Emulator) pentru a rula și depana aplicații ;

Aplicațiile sunt, în general, unele dintre cele mai importante componente a unei experiențe smartphone. Aceasta este deoarece include caracteristici de bază ale comunicări precum: mesajele text, apelurile și browserele de Internet, dar şi pentru cp o aplicație bine concepută poate fi cel mai important instrument pe care îl poate utiliza o persoană.

Există o mulțime de aplicații pentru a accesa Twitter și alte rețele sociale. Oricine poate accesa site-ul Twitter prin intermediul site-ului, dar aplicațiile permit oamenilor să adapteze experiența la propriile gusturi și dorințe, ceea ce o face o experiență mai plăcută și mai plină de satisfacții. Sau poate mai eficient și mai interesant - depinde de ce vrea să iasă din utilizator.

Aplicațiile educaționale pot avea un impact profund și sunt evident apropiate și îndrăgite de utilizatori. Educația video, ebook-urile și chiar jocurile pot ajuta oamenii din întreaga lume să învețe cum să citească sau să facă matematică, să investească sau să programeze.

Ecosistemul pentru aplicații Android este bogat și diversificat, iar utilizarea acestora schimbă viețile oamenilor. Unele servesc zeci de milioane de utilizatori, iar altele servesc doar zece utilizatori, dar în toate cazurile aplicaţiile au o putere transformatoare ca portaluri către informație. Numai aplicațiile nu vor face lumea mai bună, dar oamenii care le dezvoltă și le folosesc lucrează pentru a împărtăși informații, pentru a preda, pentru a colabora și pentru a crește din punct de vedere profesional, spriritual sau educaţional.

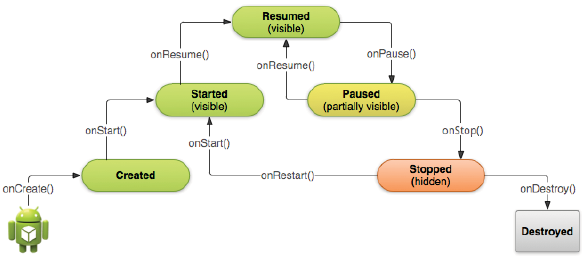
Aplicaţia de Traducător dezvoltată a pornit de la ideea că educaţia continuă este ceea ce ne motivează să cunoaştem alte culturi, să călătorim şi să încercăm să înţelegem oamenii ce vorbesc alte limbi.

Aplicaţia este construită pe activităţi tip android (Activity) ce reprezintă fiecare screen al aplicaţiei unde utilizatorul poate interacţiona cu acesta. Dacă luăm exemplul modelului MVC (Model-View-Controller), Activity este clasa Controller ce va primi toate informaţiile şi evenimente de la utilizator, iar layout-urile de XML şi UI vor fi partea de views. (**Raimon Rŕfols Montané, Laurence Dawson - Learning Android Application Development, 2016).**

O parte importantă a acestor activităţi este ciclul lor de viaţă, ele fiind plasate într-un coş şi de aici o activitatea poate avea una din următoarele statusuri :

* dacă o activitate este afişată 100% înseamnă că aceasta este prima în cozul execuţiei ;
* dacă o activitate este afişată dar parţial acoperită de alte părţi, de exemplu o fereastră de tp dialog sau orice altă activitate cu regiuni transparente, va fi într-un statut de pauză.
* Când o activitate nu mai este vizibilă şi o altă activitate este afişată, precedenta activitate este oprită şi nu mai este activă.
* Când o activitate este oprită, această poate fi ştearsă de tot pentru a elibera spaţiul din memorie, astfel în acest caz activitatea este finalizată. (**Raimon Rŕfols Montané, Laurence Dawson - Learning Android Application Development, 2016).**

Majoritatea aplicaţiior Android reprezintă o serie de ecrane conectate pe care utilizatorul le poate naviga, astfel startul pentru o experienţă unică o reprezintă conexiunile dintre activităţi. Acestea pot fi utilizate pentru a afişa informaţii, pentru a suna pe cineva sau chiar pentru a juca un joc şi reprezintă baza şi începutul unei aplicaţii de tip Android. Întregul ciclu de viaţă al unei activităţi Android începe de la metoda *onCreate()* şi până la metoda *onDestroy().* O Activitate este vizibilă de către utilizator din momentul chemării metodei *onStart()* şi până la metoda *onStop(),* şi în cele din urmă Activitatea este în prim plan între metodele *onResume()* şi *onPause().* Deşi nu este absolut necesar ca o Activitate să treacă prin fiecare etapă, cel mai des acesta este cazul.Diagrama de mai jos ne ilustrează ciclul de viaţă a unei activităţi de la creare şi până la distrugerea acesteia. (Learning Android Application Development).



În Android ne putem referi la o colecție sau la o serie de activități care sunt aranjate într-un stack, altfel cunoscut sub numele de « back stack ». Când un utilizator se mută dintr-o activitate în alta fiecare dintre acestea este plasată în stack în ordinea în careau fost vizitate. Atunci când o nouă activitate înlocuiește topul actual, activitatea curentă este împinsă în jos în stack și apoi oprită. Stackul de activități este un standard de tip LIFO (Last In First Out), dar acest stack se poate personaliza în funcţie de comportamentul necesar al aplicaţiei.

Fişierul AndroidManifest.xml este locul unde descriem întreaga aplicaţie, şi anume toată activităţile, serviciile, furnizorii de conţinut , permisiunile şli pictogramele folosite de aplicaţie, meniul şi multe alte configuraţii.

Fiecare activitate poate conţine o serie de componente pentru a putea prezenta utlizatorului informaţia dorită, iar scopul majorităţii componentelor este de a interacţiona cu acestea. Astfel, aplicaţia noastră în activitatea principală conţine 5 elemente de tip TetView si un element de tip FloatinActionButton, fiecare dintre acestea având ataşate evenimentul de onClick.

Butonul de tip FloatingActionButton are scopul de a promova o acţiune particulară din cadrul aplicaţiei. În cazul aplicaţiei Traducător acest button are rolul de a accesa partea de traducere online a unui text introdus de către utilizator.

Toate clasele îşi au locaţia în pachetul *com.example.tatiana.languagetranslator* din cadrul folderului *src/main/java*. Clasa principală care permite crearea primei experienţe a utilizatorului cu aplicaţia este MainActivity ce extinde AppCompatActivity (o clasă de bază pentru activităţi ce suportă caracteristicile librăriei action bar).

**public class** MainActivity **extends** AppCompatActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);

Cu ajutorul acestei extinderi am suprascris metoda onCreate şi am introdus funcţionalităţile şi evenimentele pe care dorim să le implementăm.

Aplicaţia include 5 categorii afişate pe pagina principală :

* *Alphabet ;*
* *Numbers ;*
* *Family members ;*
* *Colors ;*
* *Phrases.*

Pentru fiecare dintre această categorie avem un obiect de tip TextView căruia îî atribuim evenimentul corespunzător. În cazul nostru acest eveniment este realizat cu ajutorul metodei *onClick*.

TextView alphabet = (TextView) findViewById(R.id.***alphabet***);  
alphabet.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Intent alphabetIntent = **new** Intent(MainActivity.**this**, AlphabetActivity.**class**);  
 startActivity(alphabetIntent);  
 }  
});

Pentru fiecare dintre evenimente create am creat şi atribuit clasa specifică categoriei, şi anume AlphabetActivity, NumbersActivity, etc.

MainActivity mai conţine un eveniment diferit de celelalte cinci, şi anume un obiect de tip FloatingActionButton folosit în cadrul aplicaţiei pentru evidenţiaerea şi punerea în prim plan a funcţionalităţii de traducere online oferită de aplicaţie.

Toolbar toolbar = (Toolbar) findViewById(R.id.***toolbar***);  
setSupportActionBar(toolbar);  
  
FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.***fab***);  
fab.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Intent translationIntent = **new** Intent(MainActivity.**this**, Translation.**class**);  
 startActivity(translationIntent);  
 }  
});

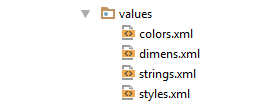
Toată partea de UI a fost configurată în documentele de tip xml amplasate în fiţierul specific *layout* şi ce cuprinde configurările ce ţin de culoare, dimensiune, afişare. Pentru a avea un cod cât mai compact şi mai lizibil, precum şi pentru a atinge rezultele dorite în ceea ce priveşte partea de interfaţă, pentru meniul principal am folosit două layout-uri: *content-main.xml* unde sunt incluse configurările pentru fiecare categorie de pagina principală. Acestea sunt incluse într-un *FramLayout* care sunt destinate pentru a bloca o secţiune de pe ecran pentru afişarea unui singur element, în cazul nostru denumirea activităţii.

*<!-- Alphabet category -->*<**FrameLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"  
 android:background="@color/category\_alphabet"**>  
 <**TextView  
 android:id="@+id/alphabet"  
 style="@style/CategoryStyle"  
 android:background="?android:attr/selectableItemBackground"  
 android:text="@string/category\_alphabet"** />  
</**FrameLayout**>

*activity\_main.xml* ce cuprinde configurările pentru Toolbar (bara de meniu) şi pentru floating button, incude *content\_main* şi este setat ca şi layout principal în cadrul metodei *onCreate*  din clasa MainActivity.

<**include layout="@layout/content\_main"** />

Elementele fixe cum sunt dimensiunea unue elemen, culoarea, textul ce trebuie afişat pe bara de meniu şi pe fiecare activitate în parte au fost definite în cadrul fişierului *values*:



Partea de resurse cuprinde multiple variante, în funcţie de rezoluţia ecranului sau orice altă configurare a telefonului, astfel că Android va alege automat varianta cea mai bună pentru dispozitivul utilizatorului li pentru ca acesta să aibă cea mai bună experienţă.

Aplicaţia dezvoltată conţine şi fişiere statice de tip XML care trebuie incluse în proiect fără nici o reformatare sau filtrare şi ce sunt incluse în fişierul denumit « assets » şi care vor fi accesate cu ajutorul clasei AssetManagemer.

Aceste documente au fost folosite în codul spefic fiecărei categorii pe care utilizatorul le poate accesa de pe pagina principală. Fiecare categorie are la bază un xml file care ne ajută să extragem datele necesare pentru afişarea listei de cuvinte/expresii.

În cadrul activităţilor de categorii avem următoarele elemente :

* *ImageView*
* *2 elemente de tip TextView*
* *Al doilea element de tip ImageView dar care va avea funcţionalitatea unui Media Player*

**Vederi comune (common views)**

Acestea ar putea fi cele mai utilizate componente Android sau widget-uri iar cu doar câteva elemente putem crea nucleul unei aplicaţii.

*TextView* este una dintre cele mai utilizate elemente şi ajută la afişarea unui text, care în mod normal nu poate fi editat. Acest text poate fi configurat, permite utilizatorului să copieze textul şi să deschidă adrese URL atunci când sunt detectate.

*EditText* este folosit în cadrul aplicaţiei noastre într-o altă activitate şi permite utilizatorului să editeze textul, fiind o subclasă a elementului *TextView,* deci menţine funcţionalităţile de bază, dar adaugă şi altele noi. Cu un singur click se deschide tastatura utilizatorului ce+i permite acestuia să modifice conţinutul din element.

*ImageView,* alături de celelalte două elemente menţionate mai sus este una dintre bazele şi fundamentale vederi în Android.

*Butonul* este un element simplu, iar conform descrierii din Google este un *push-button*  widget. Se ocupă de toate stările şi stilurile out-of-box.(Learning Android Application Development).

**ViewGroup comune (commons view group)**

Până acum am menţionat câteva widget-uri de bază care ne permit să afișăm un text, dau posibilitatea utilizatorilor să introducă date, să declanșeze o acțiune când butonul este apăsat și așa mai departe. Cu toate acestea, nu am menționat mecandismul care ne ajută să le plasăm corect pe ecran. Pentru a face acest lucru, vom utiliza nişte ViewGroups care ne va permite să definim modul în care sunt introduse elementele.

*FrameLayout* este una dintre cele mai simple clase layout. Este deobicei folosit pentru a include un singur view dar cu mai mulţi „copii”.

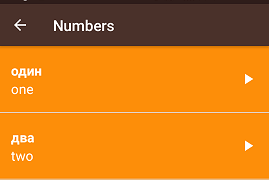
*LinearLayout* ne permite să afizăm mai multe elemnte fie vertical , fie orizontal şi este foarte des folosit dacă vrem să afişăm toate elementele la aceeaşi distanţă între ele.

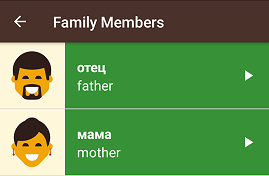
*RelativeLayout* este unul ce are o putere mare asupra poziţionării elementelor pe ecran, bazându-se pe fiecare dintre acestea.

*ListView* este o componentă puternică oentru afişări în Android, oferindu-ne posibilitatea afişării unui set de date într-un container derulabil. Fiecare element va fi afişat pe o singură linie şi prezentând utilizatorului doar un anumit număr de elemente, odată dispărute de pe ecran acestea se şterg şi se înlocuiesc cu cele ce urmează în listă. Această componentă este vitală pentru performanţa unei aplicaţii, în special când avem o listă mare de elemente ce trebuie afişate utilitarului.

Getterele şi setterele pentru aceste elemente au fost create în cadrul clasei Word în cadrul căreia avem două tipuri de Constructori.

**public** Word(String defaultTranslation, String russianTranslation, **int** mediaPlayer) {  
 **mDefaultTranslation** = defaultTranslation;  
 **mRussianTranslation** = russianTranslation;  
 **mMediaPlayer** = mediaPlayer;  
  
}  
  
**public** Word(String defaultTranslation, String russianTranslation, **int** imageResourceId, **int** mediaPlayer) {  
 **mDefaultTranslation** = defaultTranslation;  
 **mRussianTranslation** = russianTranslation;  
 **mImageResourceId** = imageResourceId;  
 **mMediaPlayer** = mediaPlayer;  
}

Doi constructori au fost definiţi deoarecere avem categorii pentru care nu afişăm decât cele două texte (engelză şi rusă) şi funcţionalitatea de audio, acestea fiin categoriile Alphabet, Numbers şi Phrases.

 Pentru celelalte două categorii rămase – Family şi Colors vom afişa şi o imagine corespunzătoate cunţntului afşat.

Codul aferent citirii a fost realizat odată cu încărcarea activităţii în metoda *onCreate(),*  pentru fiecare clasă având un document separat.

InputStream is = getAssets().open(**"numbers\_data.xml"**);

Odată citită informaţia, aceasta a fost atribuită elementului corespunzător din layout cu ajutorul unor gettere şi settere pentru fiecare element în parte şi prin intermediul unei clase WordAdapter. Anume în această clasă se decide dacă există un element de tip imageView în cadrul documentelor citite, şi în funcţie de rezultzat se decide afişarea acestuia.

ImageView imageView = (ImageView) listItemView.findViewById(R.id.***image***);  
**if**(currentWord.hasImage()){  
 imageView.setImageResource(currentWord.getImageResourceId());  
 imageView.setVisibility(View.***VISIBLE***);  
} **else**{  
 imageView.setVisibility(View.***GONE***);  
}

Fiecare dintre aceste activităţi conţin elementele menţionate mai sus pentru a putea afişa textul şi imaginea corespunzătoare. Ceea ce ţine de partea audio, fiecare clasa conţine câte un element din clasele MediaPlayer şi AudioManager. Android ne oferă clasa MediaPlayer pentru a putea accesa servicii de mediaplayer precum audio sau video. Astfel clasa MediaPalyer este utilizată pentru a controla redarea fișierelor audio și video și a fluxurilor acestora. Pentru aceasta se apelează metoda *create()* din această clasă ce ne returnează o instanţă.

**mediaPlayer** = MediaPlayer.*create*(NumbersActivity.**this**, word.getMediaPlayer());

Primul parametru reprezintă activitatea unde acest audio file trebuie să funcţioneze, iar cel de-al doilea parametru este file+ul care trebuie activat atunci când apelăm metoda. În cazul nostru, prin intermediul la Word class menţionată mai sus noi atribuim fişierul corespunzător ce este localizat în folder-ul *raw*. Odată atribuit obiectul, putem apela metodele ce ne vor ajuta la interpretarea fişierului, precum :

**mediaPlayer**.start();  
**mediaPlayer**.setOnCompletionListener(**mOnCompletionListener**);

O altă metodă recomandată atunci când folosim elemente de MediaPlayer este metoda *release()*. Această metodă se apelează atunci când obiectul nu mai este utilizat astfel încât resursele folosite de motorul de player intern asociat cu obiectul MediaPlayer să poată fi lansate imediat. Resursa poate include resurse singulare, cum ar fi componentele de accelerare hardware și eșecul de a apela metoda release() şi ce poate cauza ca instanțele ulterioare ale obiectelor MediaPlayer să nu funcţioneze conform aşteprărilor.

**private void** releaseMediaPlayer() {  
**if** (**mediaPlayer** != **null**) {  
 **mediaPlayer**.release();  
**mediaPlayer** = **null**;  
 **mAudioManager**.abandonAudioFocus(**mOnAudioFocusChangeListener**);  
 }  
}

O altă clasă importantă pentru un audio file este clasa AudioManager care oferă accesul la volumul şi control asupra soneriei. Pentru a putea folosi un astfel de obiect trebuie mai întâi să apelăm metoda getSystemService():

**mAudioManager** = (AudioManager) getSystemService(Context.***AUDIO\_SERVICE***);

O altă parte importantă a aplicaţiei este cea oferită de sistemul de build Gradle care include partea de management al dependinţelor şi care ajută nu numai la gestionarea modulelor interne, cât şi a librăriilor externe de la terţe părţi folosite şi de noi în cadrul dezvoltării aplicaţiei. Aplicațiile Android sunt construite pe acest sistem open source şi de ultimă generație care acceptă cu ușurință personalizările și este folosit pe scară largă în lumea Java. Pluginul Gradle pentru Android adaugă o gamă largă de caracteristici specifice aplicațiilor. În cazul aplicaţiei « Translator » pe lângă dependinţele de bază a mai fost adăugată şi dependinţa oferită de Microsoft Azure - Microsoft API Translator :

compile **'io.github.firemaples:microsoft-translator-java-api:0.8.3'**

*Aceasta parte a lucrării este flexibilă și depinde foarte mult de natura lucrării, poate fi organizată în mai multe capitole și conține contribuțiile personale ale autorului.*

*Includeți:*

* + *Detalii referitoare la analiză și proiectare:*
    - *descrierea metodelor pe care le-ați aplicat pentru rezolvarea problemei,*
    - *descrierea materialelor, procedurilor*
    - *calcule, tehnici, descrierea echipamentelor*
    - *metodologia de proiectare*
    - *informațiile necesare pentru ca cineva să poata reface lucrarea*
  + *Implementare :*
    - *Descrieti detaliile tehnice ale implementarii aplicatiei: mediul de implementare, modul de prezentare, modul de utilizare al aplicatiei, etc.*
  + *Testare si validare :*
    - *Descrieți metodologia de testare a aplicației și rezultatele*
    - *Includeți experimentele pe care le-ați realizat, analiza rezultatelor pe care le-ați obținut.*

# Concluzii

## Rezultate obținute

Evidentiați toate rezultatele pe care le-ați obtinut și trageți concluzii din ele. Puteți prezenta o analiză critică a ceea ce ați realizat comparativ cu alte lucrări/studii anterioare.

Includeți o listă a contribuțiilor pe care le-ați avut în domeniul temei abordate.

## Direcții de dezvoltare

Descrieți direcțiile posibile de dezvoltare.

# Reguli de formatare

## Formatarea paginii

* + Dimensiunea paginii: A4
  + Margini: 2.5 cm (sus, jos, stânga, dreapta)
  + Antet și subsol: 1.27 cm de la marginea paginii
  + În antetul paginii (header): titlul capitolului, centrat, stil: Header\_style
  + În subsolul paginii: numărul paginii, centrat

## Titluri și stiluri

Titlurile capitolelor și subcapitolelor se marchează cu stilurile Heading 1 – 4, conform documentului model anexat în format Word. Descrierea stilurilor utilizate în document este prezentată în Tabelul 5.1.

Tabelul .1. Stiluri utilizate în acest document

| Nr. | Stil | Utilizat pentru | Format |
| --- | --- | --- | --- |
| 1 | Normal | Text normal | Font: (Default) Cambria, 12 pt, Justified, Line spacing: Multiple 1.1 li, Space After: 6 pt |
| 2 | Titlu | Titlul proiectului, prima pagină | Font: 24 pt, Small caps, Centered Line spacing: single, Space Before: 126pt, After: 0 pt, |
| 3 | Titlu2 | Titlul proiectului, pagina de prezentare | Font:14pt, Bold, Centered |
| 4 | Heading 1 | Titlurile capitolelor (nivel 1) | Font: 24 pt, Indent: Left: 0 cm Hanging: 0.76 cm, Space Before: 24pt, After: 12pt |
| 5 | Heading 2 | Titlurile subcapitolelor (nivel 2) | Font: 14 pt, Bold, Indent: Left: 0 cm  Hanging: 1.02 cm, Space Before: 18pt, After: 12pt |
| 6 | Heading 3 | Titlurile secțiunilor (nivel 3) | Font: Bold, Indent: Left: 0 cm Hanging: 1.27 cm, Space Before: 6 pt, After: 6pt |
| 7 | Heading 4 | Titlurile secțiunilor (nivel 4) | Font: Italic, Indent: Left: 0 cm Hanging: 1.52 cm, Space Before: 2 pt, After: 0 pt |
| 8 | Caption | Legenda figurilor și tabelelor | Font: Italic, Font color: Text 1, Line spacing: single, Space After: 10 pt, |
| 9 | Header\_style | Antetul paginii | Font: 10 pt, Italic, Centered, Border: Bottom: (Single solid line, Background 1, 0.5 pt Line width) |

## Figuri, tabele și ecuații

### Figuri

Figurile se inserează în text centrate, cu etichetă de numerotare și legendă (Caption) în partea de jos a figurii. Numărul figurii include și numărul capitolului, după exemplul prezentat în Figura 5.1.



Figura .1. Figură exemplu, stil: Caption

## Tabele

Tabelele se inserează în text centrate, cu etichetă și legendă (Caption) în partea de sus a tabelului, aliniată la stânga. Numărul tabelului include și numărul capitolului, după cum este prezentat, de exemplu, în Tabelul 5.1.

## Ecuații

Ecuațiile se inserează în text centrate, cu numerotare în partea dreaptă. Numărul ecuației include și numărul capitolului, conform exemplului din relația (5.1).

|  |  |
| --- | --- |
|  | (5.1) |

## Referințe bibliografice

Se recomandă ca citarea referințelor bibliografice să fie făcută în formatul IEEE.

În secțiunea Bibliografie sunt prezentate exemple pentru: o citare a unui capitol dintr-o carte [1], un articol publicat într-o revistă [2] și un articol publicat la o conferință [3].

Detalii cu privire la formatul citării diverselor tipuri de referințe pot fi găsite în [4] sau [5].

Referințele bibliografice se pot insera în text utilizând facilitățile Word de a adăuga surse și bibliografie unui document (References -> Citations & Bibliography). Dacă formatul IEEE pentru bibliografie nu este instalat implicit în Word, se poate descărca gratuit de la:

<https://bibword.codeplex.com/wikipage?title=Styles&referringTitle=Home>

Instrucțiunile de instalare pentru diferite versiuni de Word se pot obține de la aceeași adresă.

# Bibliografie

|  |  |
| --- | --- |
| [1] | P. Nume, "Titlul capitolului," in *Titlul cartii*, Oras, Editura, 2016, pp. 1-24. |
| [2] | P. Nume, "Titlul articolului," *Titlul revistei,* vol. 1, no. 2, pp. 22-30, 2016. |
| [3] | P. Nume, "Titlul articolului," in *Numele conferintei*, Oras, 2015. |
| [4] | "IEEE Citation Reference," 2009. [Online]. Available: https://www.ieee.org/documents/ieeecitationref.pdf. |
| [5] | "IEEE Editorial Style Manual," 2016. [Online]. Available: https://www.ieee.org/documents/style\_manual.pdf. |