

UNIVERSITATEA “ALEXANDRU IOAN CUZA” IAȘI
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

PocketWorld – Realitate Augmentată

propusă de

Pantea Roxana-Florentina

Sesiunea: Februarie, 2018

Coordonator științific

Conf. Dr. Iftene Adrian

UNIVERSITATEA “ALEXANDRU IOAN CUZA” IAȘI
FACULTATEA DE INFORMATICĂ

PocketWorld – Realitate Augmentată

Pantea Roxana-Florentina

Sesiunea: Februarie, 2018

Coordonator științific

Conf. Dr. Iftene Adrian

DECLARAȚIE PRIVIND ORIGINALITATEA ȘI RESPECTAREA DREPTURILOR DE AUTOR

Prin prezenta declar că Lucrarea de licență cu titlul „*PocketWorld - Realitate Augmentată*” este scrisă de mine și nu a mai fost prezentată niciodată la o altă facultate sau instituție de învățământ superior din țară sau din străinătate. De asemenea, declar că toate sursele utilizate, inclusiv cele preluate de pe Internet, sunt indicate în lucrare, cu respectarea regulilor de evitare a plagiatului:

- toate fragmentele de text reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și dețin referința precisă a sursei;
- reformularea în cuvinte proprii a textelor scrise de către alți autori deține referința precisă;
- codul sursă, imaginile etc. preluate din proiecte *open-source* sau alte surse sunt utilizate cu respectarea drepturilor de autor și dețin referințe precise;
- rezumarea ideilor altor autori precizează referința precisă la textul original.

Iași, 01.02.2018

Absolvent *Pantea Roxana*

(semnătura în original)

DECLARAȚIE DE CONSIMȚĂMÂNT

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

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

Iași, 01.02.2018

Absolvent *Pantea Roxana*

(semnătura în original)

Cuprins

Motivație	7
Introducere	8
Capitolul I: Aplicații similare	9
I.1 Spyglass	9
I.2. Anatomy 4D	10
I.3. Star Walk 2	12
I.4. Concluzii	14
Capitolul II: Tehnologii utilizate	15
II.1. Android	15
II.1.1. Evoluție	16
II.1.2. Android SDK (Software Development Kit)	17
II.1.3 Android JDK (Java Development Kit)	18
II.1.4. Android Studio	18
II.2. Vuforia SDK	19
II.2.1. Înregistrare target	19
II.2.2. Integrare cu Android Studio	23
II.3. SQLite	25
II.3.1. Despre	25
II.3.1. SQLiteStudio	26
II.4. Concluzii	27
Capitolul III: Arhitectura aplicației și implementarea	28
III.1. Scanează și identifică obiectivul	31
III.2. Caută un obiectiv într-o listă dată	34
III.3. Găsește locurile din apropierea ta	36
III.4. Adaugă un nou element	39
III.5. Concluzii	42

Capitolul IV: Scenarii de utilizare	43
IV.1. Scenariul 1	43
IV.1. Scenariul 2	44
IV.1. Scenariul 3	44
IV.1. Scenariul 4	45
IV.5. Concluzii	46
V. Concluziile lucrării și posibilitatea extinderii aplicației în viitor	47
V. 1. Concluzii generale	47
V. 2. Muncă viitoare	47
Bibliografie	49

Motivație

În prezent, există numeroase dispozitive și tehnologii care evoluează odată cu trecerea timpului. Realitatea augmentată se situează în fruntea clasamentului cu cele mai utilizate tehnologii în domeniul jocurilor dar și al aplicațiilor care au la bază conceptul de software educațional. Acest domeniu stârnește curiozitatea, atât în rândul copiilor, cât și în rândul adulților prin inovațiile pe care le aduce sistemului clasic de învățare.

Am realizat această aplicație, fiind motivată de pasiunea mea pentru a călători și pentru a descoperi locuri noi. Fiecare țară are un istoric bogat și cel puțin un oraș în care se găsesc obiective turistice reprezentative, doar că poate nu suntem mereu informați corespunzător cu privire la acestea. De aceea, am creat PocketWorld, o aplicație cu ajutorul căreia poți recunoaște un obiectiv turistic și totodată să afli informații esențiale despre acesta, precum și obiectivele vecine.

Aplicația poate fi folosită atât ca o forma de orientare printre obiectivele unui oraș, cât și ca o metoda de învățare rapidă și distractivă pentru copiii captivați de imaginile din cărți.

Am apelat la realitate augmentată deoarece experiența utilizării aplicației devine mai interesantă și este un mod comod de a afla informații despre un obiectiv turistic aflat ori în realitate în fața utilizatorului, ori capturat într-o fotografie. PocketWorld oferă posibilitatea de a afla informații despre un obiectiv precum și punctele învecinate, prin scanarea sau căutarea lui într-o lista bogată. De asemenea, aplicația sugerează obiective din apropierea utilizatorului prin simpla activare a GPS-ului.

După selectarea sau identificarea obiectivului, aplicația deschide o pagina unde sunt expuse informațiile esențiale despre acesta, precum și o fotografie reprezentativă. Cele enumerate sunt extrase de pe pagina fiecăruia de Wikipedia.

Pe lângă obiectivele deja prezente în baza de date, utilizatorul mai poate adăuga altele pe care le consideră relevante. E posibil să-și dorească să înregistreze coordonatele unei cafenele cu specific sau a unui loc menționat de cineva. Astfel, punctele cheie vor fi mereu compactate într-o aplicație plăcută, ușor de utilizat și la îndemâna oricui.

Introducere

Lucrarea de față este structurată pe patru capitole care surprind elementele esențiale din crearea și utilizarea aplicației.

Capitolul I cuprinde o scurtă descriere a unor aplicații deja existente ce folosesc realitatea augmentată în scop educativ. Pentru fiecare ramură aleasă, realitatea augmentată reușește să atragă cât mai mulți utilizatori și să ofere oamenilor posibilitatea de a învăța cât mai multe lucruri noi printr-o manieră inovativă.

Capitolul II prezintă principalele tehnologii utilizate în construirea aplicației PocketWorld. Printre acestea se numără: Vuforia (pentru realitatea augmentată) și Android Studio (pentru restul aplicației). Voi prezenta cum am folosit fiecare tehnologie în parte și ce pași trebuie urmați pentru a le putea îmbina.

Capitolul III pune în evidență arhitectura și implementarea softului, resursele folosite și problemele pe care le-am întâmpinat pe parcursul integrării tuturor modulelor precum și rezolvările acestora.

Capitolul IV conține scenarii de utilizare a aplicației. Sunt descrise toate funcționalitățile aplicației în amănunt și pe exemplu concret însoțit de imagini.

Ultima parte a lucrării de față vizează concluziile și modurile prin care se poate îmbunătăți aplicația sau se poate dezvolta.

Capitolul I: Aplicații similare

Acest capitol descrie câteva aplicații ce folosesc realitatea augmentată cu scopul de a crea un soft educațional care să capteze cât mai mulți utilizatori.

I.1 Spyglass

Spyglass¹ este o aplicație care oferă o gamă largă de tehnologii pentru navigare în aer liber. Este dezvoltată pentru dispozitivele mobile cu Android și iOS. Folosește toți senzorii de care dispune un smartphone sau o tabletă pentru a afișa un număr vast de informații despre mediul înconjurător. Folosește camera pentru a augmenta elementele de navigare pe ecran. Utilizatorul poate salva puncte de interes pe harta aplicației, care îl îndrumă precis spre obiectivele alese (vezi Figura 1).

¹ <http://happymagenta.com/spyglass/index.html>



Figura 1: Imagine din aplicația Spyglass^[1]

De asemenea, aplicația pune la dispoziție un compas, sisteme de coordonate civile și militare, accelerometru pentru a măsura precis viteza deplasării, poate măsura mărimea obiectelor și oferă diferite tipuri de orientare în mediul înconjurător (vezi Figura 2). Printre acestea cele mai populare funcționalități sunt următoarele: posibilitatea marcării unor puncte de interes (locuri de pescuit sau vânătoare), locația exactă pe hartă și direcția de orientare, verificarea altitudinii în călătoriile în zone muntoase, estimarea înălțimii obiectelor, calcularea distanței și unghiurilor, conversia coordonatelor și orientarea în locuri necunoscute de utilizator.



Figura 2: Orientarea în spațiul cosmic (în stânga) și terestru (în dreapta)^[1]

I.2. Anatomy 4D

Prin intermediul aplicației *Anatomy 4D*² și al imaginilor tipărite, studenții, profesorii, medicii și toți cei care vor să învețe despre anatomie, sunt captați de experiența interactivă 4D a corpului uman. *Anatomy 4D* folosește realitatea augmentată și alte tehnologii de vârf pentru a crea un soft perfect pentru educația din secolul 21.

Aplicația duce spectatorii într-o călătorie în interiorul corpului și inimii umane, dezvăluind relațiile spațiale ale organelor, scheletului, mușchilor și sistemelor corporale. Acest mediu de învățare tridimensional este ușor de utilizat și poate fi folosit și în sălile de clasă (vezi Figura 3).

² <http://anatomy4d.dagri.com>



Figura 3: Vizualizarea corpului uman [2]

Anatomy 4D se remarcă prin:

- Modul în care poți explora corpul uman și inima în detaliu. (vezi Figura 3 și Figura 4).
- Expunerea unor organe specifice unui anumit sistem. De exemplu, te poți axa doar pe sistemul osos, muscular sau respirator.
- Schimbarea vizualizării organelor de la corp masculin la cel feminin și invers.
- Faptul că poți mări sau micșora organele pe care vrei să le vezi pentru a obține informațiile dorite.

În Figura 4 este evidențiată ilustrarea 3D a inimii. Aplicația poate reda chiar și modul în care inima pompează sângele, iar perspectivele de vizualizare se pot schimba în funcție de preferința utilizatorului.

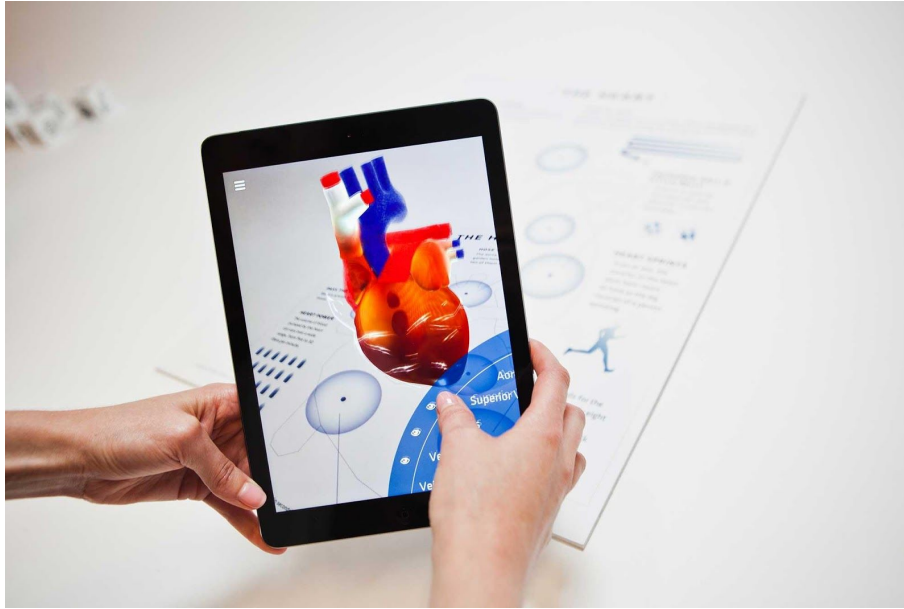


Figura 4: Vizualizare inimii [2]

Anatomy 4D este disponibilă pe iPhone, iPad, dar și pe platforma Android.

I.3. Star Walk 2

Star Walk ³ este o aplicație dezvoltată pe platforma android și iOS. Folosește locația și senzorii dispozitivului pentru a genera o perspectivă augmentată a corpurilor cerești în timp real (vezi Figura 5). Aplicația identifică aproximativ 250.000 de obiecte cerești (stele și constelații, sisteme solare, galaxii, sateliți și ploi de meteoriți).

Cu Star Walk 2⁴ nu descoperim doar acestea, ci putem să ajungem la perspectiva fiecărei planete în parte. În acest caz, putem găsi informații specifice, cum ar fi: distanța până la soare, temperatura, dimensiunea vizuală și altele. Dacă oferim permisiunea pentru a accesa camera telefonului, putem vedea cu ajutorul realității augmentate unde sunt poziționate stelele în raport cu telefonul, după cum se observă în Figura 6.

³ <https://play.google.com/store/apps/details?id=com.vitotechnology.StarWalk&hl=ro>

⁴ <https://play.google.com/store/apps/details?id=com.vitotechnology.StarWalk2Free&hl=ro>

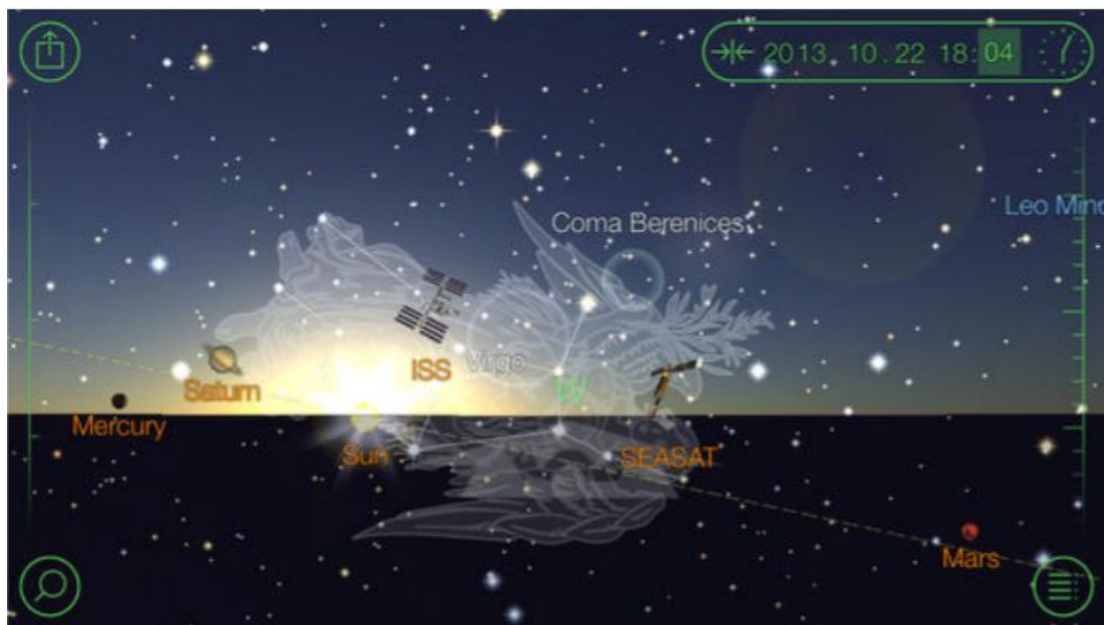


Figura 5: Poziția sateliților folosind funcționalitatea ”Călătorie în timp” cu ajutorul aplicației Star Walk ^[5]



Figura 6: Vizualizarea constelației cu ajutorul aplicației Star Walk 2 ^[6]

Aplicația este disponibilă pentru iPhone și iPad, dar există și câteva variante de aplicații asemănătoare pentru platforma Android.

I.4. Concluzii

Domeniile aplicațiilor de realitate augmentată cu scop educativ sunt diversificate, după cum se poate observa din prezentarea de mai sus a celor trei aplicații. În domeniul geografiei, se găsesc diferite aplicații care se bazează pe realitatea augmentată văzută prin intermediul camerei de telefon, dar nu am găsit niciuna care să semene cât mai mult cu ideea pe care am avut-o eu în realizarea acestui proiect. Consider că aplicația mea are spirit inovativ și ar ajuta mulți adulți și copii atât în dezvoltarea cunoștințelor, a asimilării într-un mod plăcut a informațiilor despre punctele principale de interes dintr-un oraș, cât și în orientarea ușoară într-un oraș necunoscut.

Aceste exemple subliniază ideea că învățământul și deschiderea oamenilor spre cunoașterea altor locuri s-ar dezvolta mult mai mult dacă s-ar utiliza în timpul orelor de la școală cât și în timpul liber aplicații de acest tip. La clasele mici, copiii își pierd imediat interesul pentru lecție, se plictisesc și își găsesc altă ocupație. De aceea, atenția lor poate fi sporită prin activități ce includ noile tehnologii. Dacă nu sunt atenți pentru informația prezentată, vor deveni atenți pentru plusul de noutate și vor reține fără să-și dea seama materia școlară. În cazul adulților, este mereu util un ajutor atunci când nu știu la ce să se aștepte de la un oraș nou. Prin intermediul aplicației, pot asimila o cantitate mare de cunoștințe vis-a-vis de posibilele puncte de interes chiar pe parcursul călătoriei.

Capitolul II: Tehnologii utilizate

În procesul de creare al aplicației, am folosit programul Android Studio pentru a realiza trei dintre funcționalitățile principale, iar pentru cea de-a patra funcționalitate am folosit platforma Vuforia.

II.1. Android

Android⁵ este o platformă software dezvoltat de grupul Google. Android este open source și este conceput pentru utilizarea pe dispozitive mobile, smartphone-uri, tablete, GPS, TV. Oferă un framework bogat cu ajutorul căruia se pot crea diferite aplicații sau jocuri, în limbajul de programare Java. Framework-ul oferă numeroase librării, documentație, tutoriale, un emulator care simulează un device mobil pe platforma de dezvoltare proprie, dar și un IDE (Android Studio⁶).

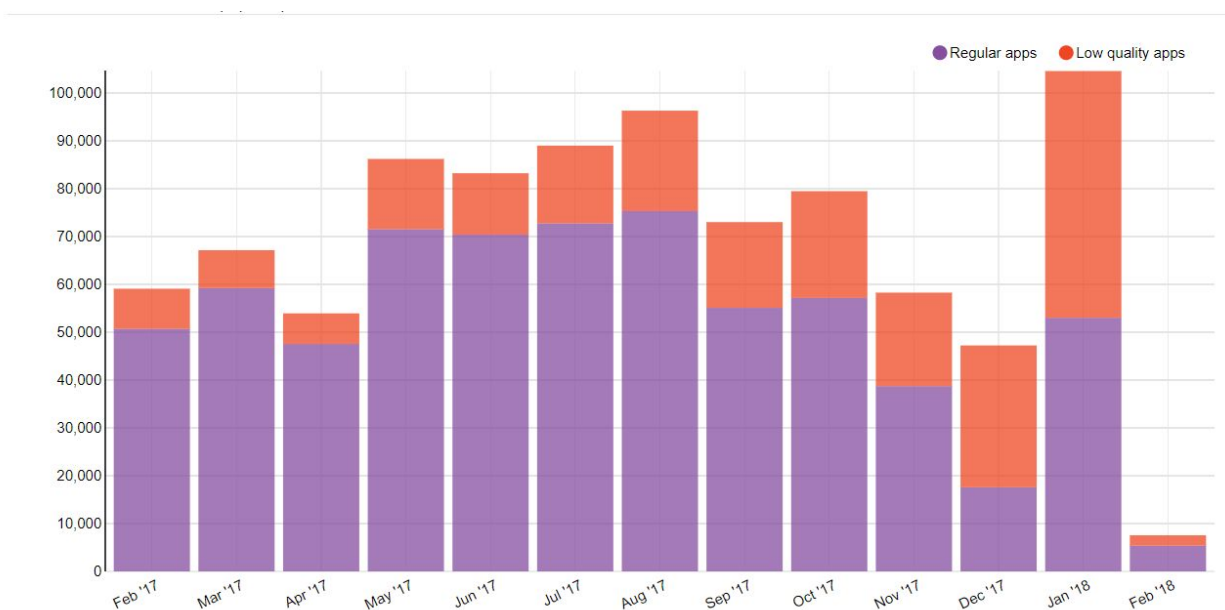


Figura 7: Număr aplicații Android noi pentru fiecare luna din Februarie 2017 pana în Februarie 2018 [8]

⁵ <https://www.android.com/>

⁶ <https://developer.android.com/studio/index.html>

Dezvoltatorii de software și aplicații pot utiliza Android pentru a dezvolta aplicații mobile, ce pot fi vândute în magazinul Google Play⁷. În prezent peste un milion de aplicații sunt puse la dispoziție pe acest magazin virtual. Numărul aplicațiilor apărute în ultimul an poate fi vizualizat în mod detaliat, pe fiecare lună în Figura 7.

Deoarece sunt dezvoltate ca produse Google, utilizatorii au posibilitatea de a-și conecta dispozitivul mobil la alte produse Google, cum ar fi serviciile de stocare în cloud, e-mail și servicii video.

II.1.1. Evoluție

Din momentul lansării și până în prezent au apărut numeroase versiuni de Android. Evoluția în timp a versiunilor este următoarea (Wikipedia, 2018):

- *Android 1.0* (23 Septembrie 2008);
- *Android 1.5 – Cupcake* (27 Aprilie 2009);
- *Android 1.6 – Donut* (15 Septembrie 2009);
- *Android 2.0/2.1 – Eclair* (26 Octombrie 2009);
- *Android 2.2 – Froyo* (20 Mai 2010);
- *Android 2.3 – Gingerbread* (6 Decembrie 2010);
- *Android 3.0/3.1/3.2 – Honeycomb* (22 Februarie 2011);
- *Android 4.0 – Ice Cream Sandwich* (18 Octombrie 2011);
- *Android 4.1/4.2 – Jelly Bean* (9 Iulie 2012);
- *Android 4.4 – KitKat* (31 Octombrie 2013);
- *Android 5.0/5.1 – Lollipop* (12 Noiembrie 2014);
- *Android 6.0 – Marshmallow* (5 Octombrie 2015);
- *Android 7.0/7.1.2 – Nougat* (22 August 2016);
- *Android 8.0/8.1 – Oreo* (21 August 2017).

⁷ <https://play.google.com/store>



Figura 8: Evoluția versiunilor de Android ^[9]

Prima versiune comercială, Android 1.0 a apărut în Septembrie 2008. Versiunile 1.0 și 1.1 nu au un nume specific, însă începând din aprilie 2009 cu Android 1.5 – Cupcake, versiunile de Android preiau numele diverselor tipuri de dulciuri în ordine alfabetică, după cum este evidențiat în Figura 8. Aplicația pe care am dezvoltat-o rulează pe Android 5.1 Lollipop⁸.

II.1.2. Android SDK (Software Development Kit)

Android SDK⁹ este un set de unelte de dezvoltare utilizate în crearea aplicațiilor pe platforma Android. Setul include următoarele componente: biblioteci necesare, debugger, emulator, exemple de cod sursă, tutoriale și documentație ca ajutor în dezvoltare.

Un emulator simulează un dispozitiv și îl afișează pe calculator. El oferă posibilitatea de a dezvolta și testa aplicații Android fără a avea componenta hardware necesară (adică un telefon sau o tabletă).

⁸ <https://developer.android.com/about/versions/android-5.1.html>

⁹ <https://developer.android.com/studio/index.html>

De fiecare dată când Google introduce o nouă versiune de Android, este introdus și un nou SDK. De aceea, pentru a putea scrie programe cu ultimele caracteristici, dezvoltatorii trebuie să descarce și să instaleze fiecare versiune de SDK actuală pentru un anumit tip de telefon.

Programatorii pot utiliza SDK-ul și pentru a scrie programe la linia de comandă, însă cea mai utilizată metodă este prin intermediul unui IDE (Integrated Development Environment). Majoritatea acestora conțin și o interfață grafică ce are ca scop să mărească performanța și viteza de lucru. Cum aplicațiile Android sunt scrise în Java, utilizatorul ar trebui să aibă instalat și un JDK (Java Development Kit).

II.1.3 Android JDK (Java Development Kit)

Android JDK¹⁰ este un mediu de dezvoltare al aplicațiilor Java. Acesta include un JRE (Java Runtime Environment), un interpretor Java, un compilator (javac), un arhivator (jar), un generator de documentație (javadoc) și alte instrumente necesare dezvoltării Java.

Dezvoltatorii de aplicații Java care sunt la început de drum pot face confuzie între JRE și JDK. Pentru a rula aplicații Java și applet-uri, putem descărca doar JRE-ul, însă pentru a le dezvolta avem nevoie și de JDK.

Cele mai cunoscute unelte din JDK sunt java și javadoc. Ambele se pot rula de la linia de comandă (command prompt). Fișierele Java sunt fișiere text simple, dar care au extensia “.java”. După scrierea și salvarea codului sursă, compilatorul “javac” este invocat pentru a crea fișierele “.class”, iar apoi comanda “java” poate fi apelată pentru a rula programul.

II.1.4. Android Studio

Android Studio¹¹ este un IDE de la Google care oferă utilizatorilor instrumentele necesare dezvoltării aplicațiilor pentru sistemul de operare Android. Fundația sau punctul de plecare al IDE-ului este IntelliJ IDEA¹². Mediul de lucru oferă posibilitatea de a crea aplicații nu doar

¹⁰ <https://developer.android.com/studio/index.html>

¹¹ <https://developer.android.com/studio/index.html>

¹² <https://www.jetbrains.com/idea/>

pentru telefon și tabletă, ci și pentru televizoare Android, ochelari și ceasuri Smart și alte dispozitive care au la bază acest sistem de operare.

Android Studio este construit pe ideea unei continue integrări, ceea ce face posibilă găsirea rapidă a greșelilor și repararea lor la momentul scrierii codului. Acest lucru optimizează timpul de dezvoltare și oferă un avantaj utilizatorului.

II.2. Vuforia SDK

Vuforia SDK¹³ este un kit de dezvoltare software pentru realitatea augmentată care utilizează tehnologia Computer Vision¹⁴ pentru recunoașterea imaginilor și obiectelor 3D simple.

Această capacitate de înregistrare a imaginilor permite dezvoltatorilor să identifice cu o foarte mare precizie elemente din lumea reală, în momentul vizualizării lor prin camera unui dispozitiv mobil.

II.2.1. Înregistrare target

În continuare voi descrie cum am folosit Vuforia în aplicația mea. Pentru început mi-am făcut cont pe <https://developer.vuforia.com> pentru a îmi înscrie aplicația și a obține cheia de licență pe care am folosit-o ulterior în integrarea cu Android Studio.

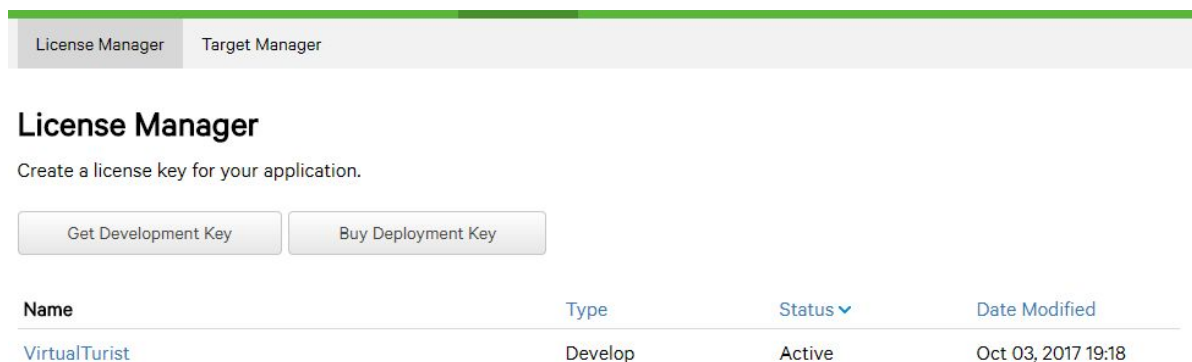


Figura 9: Înscrierea aplicației pe portalul Vuforia

¹³ <https://developer.vuforia.com/downloads/sdk>

¹⁴ https://en.wikipedia.org/wiki/Computer_vision

License Manager Target Manager

[Back To License Manager](#)

Add a free Development License Key

App Name

You can change this later

License Key

Develop

Price: No Charge

Reco Usage: 1,000 per month

Cloud Targets: 1,000

VuMark Templates: 1 active

VuMarks: 100

Figura 10: Obținerea cheii de licență

Am creat apoi baza de date pentru înregistrarea imaginilor de tip marker pe care le-am folosit pentru realitatea augmentată. După cum se observă în Figura 10, numele bazei de date este `touristic_objectives`. Imaginile, precum și nivelul de augmentare al fiecăreia, este ilustrat în Figura 11. Se observă că hărțile pe care le-am înregistrat sunt augmentate și pot fi folosite cu succes în aplicație.

License Manager Target Manager

Target Manager

Use the Target Manager to create and manage databases and targets.

[Add Database](#)

Database	Type	Targets	Date Modified
touristic_objectives	Device	5	Oct 25, 2017 12:08

Figura 11: Crearea bazei de date pentru Vuforia

touristic_objectives
[Edit Name](#)

Type: Device

Targets (5)

Add Target

Download Database (All)






<input type="checkbox"/> Target Name	Type	Rating	Status	Date Modified
<input type="checkbox"/>  colosseum_4	Single Image	★★★★★	Active	Oct 25, 2017 12:08
<input type="checkbox"/>  colosseum_2	Single Image	★★★★★	Active	Oct 25, 2017 12:03
<input type="checkbox"/>  colosseum_5	Single Image	★★★★★	Active	Oct 25, 2017 12:02
<input type="checkbox"/>  colosseum_1	Single Image	★★★★★	Active	Oct 25, 2017 11:49
<input type="checkbox"/>  colosseum_0	Single Image	★★★★★	Active	Oct 25, 2017 11:46

Figura 12: Baza de date din Vuforia a aplicației PocketWorld

Portalul oferă posibilitatea de a vedea caracteristicile ce corespund criteriilor de realitate augmentată pentru fiecare imagine din baza de date. În Figura 12 se pot vedea caracteristicile uneia din imaginile corespondente Colosseumului. După cum sugerează ratingul, imaginea este bine augmentată, punctele de tip marker există și pot constitui un întreg care să se poată fi recunoscut de orice dispozitiv.

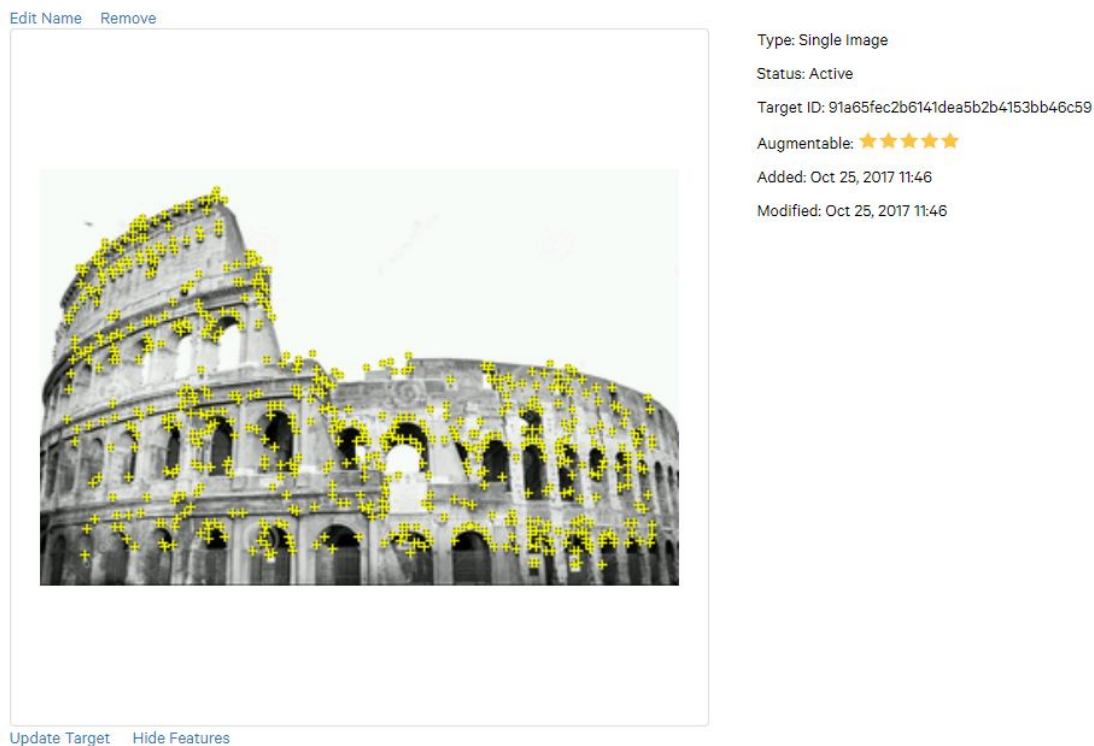


Figura 13: Imaginea Colosseumului cu punctele de tip marker

După ce am urmat toți pașii de mai sus, am mai avut de urmat unul singur: să descarc dataset-ul selectând platforma de dezvoltare Android Studio, Xcode or Visual Studio (ca în Figura 14).

Download Database

5 of 5 active targets will be downloaded

Name:

touristic_objectives

Select a development platform:

☒ Android Studio, Xcode or Visual Studio

☐ Unity Editor

Cancel

Download

Figura 14: Selectarea platformei pentru descărcarea bazei de date

II.2.2. Integrare cu Android Studio

Pentru început am creat proiectul pe platforma Android Studio. Apoi am importat librăria Vuforia.jar pentru Android descărcată de pe site-ul oficial.

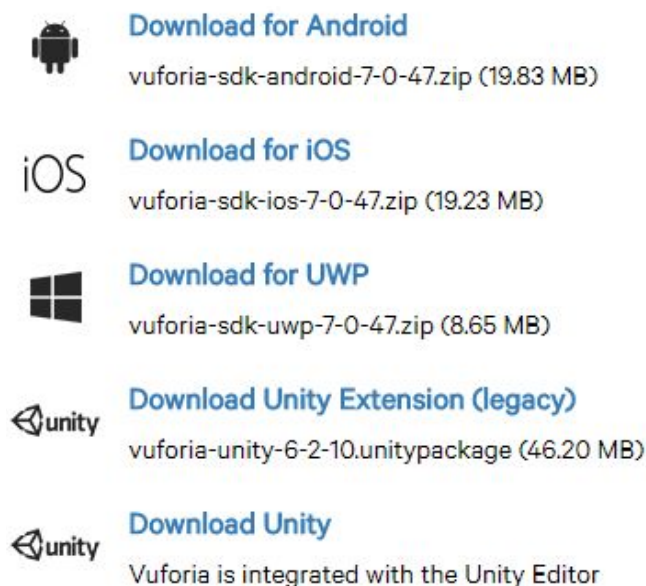


Figura 15: Librăriile dezvoltate de Vuforia pentru diferite platforme

A urmat importarea celor două fișiere de la Vuforia pentru a putea integra partea de realitate augmentată. În Figura 16, se pot observa cele două fișiere importate: baza de date (despre care am vorbit în descrierea kitului de dezvoltare Vuforia) și fișierul xml descărcat de pe site-ul oficial cu datele propriu-zise despre target-uri.

În Vuforia, conceptul de *target* este reprezentat prin definiția a doua clase: *Trackable* și *TrackableResult*. *Trackable* este clasa de bază utilizată pentru identificarea tuturor obiectelor pe care Vuforia SDK le poate detecta și urmări în spațiul 3D. Tot despre aceasta clasă spunem că definește proprietățile de bază pentru un obiect țintă dat (target), precum numele său, id-ul și datele specifice definite de user la înregistrarea în baza de date.

În prezent, Vuforia definește următoarele tipuri de tracker-e:

- ObjectTarget - un obiect Trackable ce constă într-un singur obiect;

- ImageTarget - un obiect Trackable ce constă într-o imagine (Pentru dezvoltarea aplicației PockerWorld am folosit acest tip de obiect);
- CylinderTarget - un obiect Trackable ce constă în imagini aplicate unei suprafețe cilindrice sau conice;
- MultiTarget - un obiect Trackable ce constă în mai multe imagini combinate într-un spațiu dat;
- Marker - un obiect Trackable ce constă într-un marker;
- Word - un obiect Trackable ce reprezintă elemente de text.

Vuforia analizează frame-urile capturate de cameră și încearcă să potrivească elementele ce sunt vizibile în spațiul video înregistrat cu unul sau mai multe target-uri dintr-un set predefinit. Când se găsește o potrivire, este creat un obiect *TrackableResult* și trimis aplicației.



 touristic_objectives.dat	360,112	359,722	DAT File
 touristic_objectives.xml	515	244	XML Document

Figura 16: Fișierele necesare pentru folosirea realității augmentate în Android Studio

```
<?xml version="1.0" encoding="UTF-8"?>
<QCARConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="qcar_config.xsd">
  <Tracking>
    <ImageTarget name="colosseum_4" size="0.500000 0.333203" />
    <ImageTarget name="colosseum_2" size="0.500000 0.375234" />
    <ImageTarget name="colosseum_5" size="0.500000 0.295204" />
    <ImageTarget name="colosseum_1" size="0.500000 0.281169" />
    <ImageTarget name="colosseum_0" size="0.500000 0.327560" />
  </Tracking>
</QCARConfig>
```

Figura 17: Fișierul xml cu informații despre target-uri

Pentru a putea folosi uneltele pe care le oferă Vuforia (în special ARCamera și Target Image de care am avut nevoie în realizarea proiectului), a trebuit să introduc cheia de licență obținută de pe portalul de la Vuforia și să activez încărcarea bazei de date.

Cheia de licență este dată ca parametru funcției `setInitParameters` (Figura 18) ce face inițializarea parametrilor pentru Vuforia, iar fișierul xml este încărcat în sesiune pentru ca obiectele țintă să poată fi identificate după nume (Figura 19) prin intermediul unor tracker-e.

```
Vuforia.setInitParameters(mActivity, mVuforiaFlags, "AZOLmp////////AAAAGfNVhNnYh0KlnZqwOrUpXDpazXkg/" +
    "Oji7E8+OHbmV5ZhyN3L2kmEtRkBW/bJFHqwl0E29e4BNM/" +
    "Hmlz7PAmhrQQJvWlr/8q+gnyXQ/WOsh5pURpalT0LJoGWN" +
    "YRFadpRmdYRyCdIfTC3w8nU99A46b3KSRP0jGM6vuual01" +
    "q2fSqcYbPcl0nAu0em/OU/fI68QEhac07ZaK/gdOR9sYdD" +
    "VWSsKmF5d1CG8bzkRpFf1EYbTEklizd6e7U3Ifee7EgAjdU" +
    "mhKVSwbGtxEX6dMPX59Ajt0VDxbj0geTkN16WRJ7UuxmZ3" +
    "v9VEwQ0khtLVel6FCpjSics9rohYaSQ4NnrIa8D0yoAHXFp" +
    "esyQjJDuDh");
```

Figura 18: Setarea key-ului pentru accesarea resurselor Vuforia

```
mDataSet="touristic_objectives.xml";
if (!mCurrentDataset.load(mDataSet,
    STORAGE_TYPE.STORAGE_APPRESOURCE))
    return false;
```

Figura 19: Încărcarea xml-ului asociat aplicației PocketWorld

II.3. SQLite

SQLite¹⁵ este o mică bibliotecă C care implementează un motor de baze de date SQL încapsulat, oferă posibilitatea de a-l introduce în diverse sisteme și necesită zero-configurare.

II.3.1. Despre

SQLite este diferit de majoritatea altor motoare de baze de date SQL prin faptul că a fost proiectat pentru a fi simplu:

- Simplu de administrat.
- Simplu de folosit.
- Simplu de a fi încapsulat într-un program mai mare.
- Simplu de întreținut și configurat.

¹⁵ <https://www.sqlite.org>

SQLite este foarte sigur. Siguranța este o consecință a simplismului. Cu mai puțină complicitate, este mai puțin loc de greșeli. Simplitatea într-un motor de baze de date poate fi fie un punct tare, fie unul slab, depinde de motivul pentru care este necesar. Simplitatea lui se datorează sacrificării altor caracteristici, precum controlul la accesul detaliat, un set bogat de funcții implementate, proceduri stocate.

II.3.1. SQLiteStudio¹⁶

SQLiteStudio este un program creat pentru gestionarea ușoară și plăcută a bazelor de date de tip SQLite. Interfața grafică este bazată pe framework-urile QT pentru a putea fi compatibilă cu majoritatea sistemelor de operare.

Am ales acest tool pentru a accesa și modifica ușor informațiile din baza de date destinată aplicației PocketWorld (Figura 20 și Figura 21).

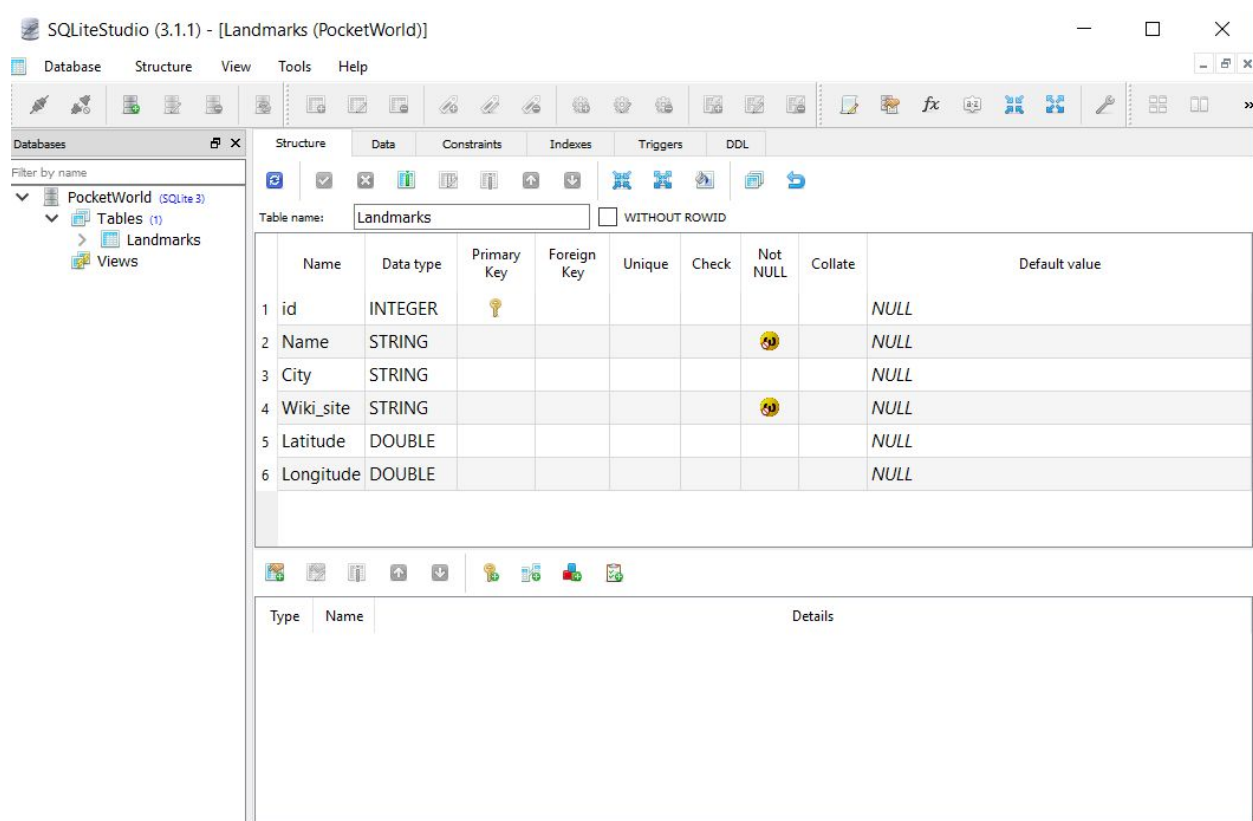


Figura 20: Structura tabelului Landmarks din baza de date utilizată de aplicația PocketWorld

¹⁶ <https://sqlitestudio.pl/index.rvt>

Capitolul III: Arhitectura aplicației și implementarea

În acest capitol voi prezenta în ce constă fiecare modul în parte și cum l-am implementat. În Figura 22, se pot observa cele patru funcționalități principale: *Scan landmark*, *Search landmark*, *Lookup for closest landmarks* și *Add new landmark*. Pentru fiecare am inclus și tehnologia pe care am utilizat-o pe parcursul realizării proiectului.

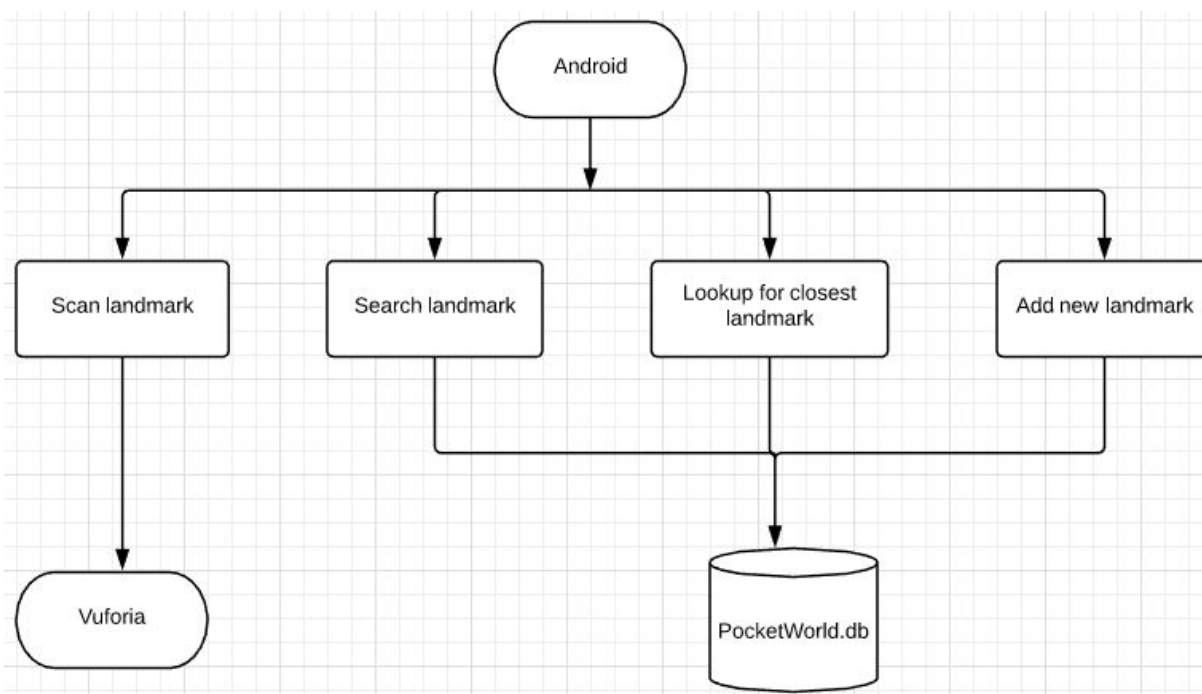


Figura 22: Arhitectura generală a aplicației

În Figura 23 voi ilustra împărțirea pe clase a funcționalităților principale. Aici se pot observa și clasele ajutătoare utilizate, cât și cele scrise de mine. Toate activitățile trebuie introduse în fișierul `AndroidManifest.xml` pentru a putea funcționa. În cazul în care este folosită o activitate, însă nu este trecută în acest fișier, aplicația se închide în momentul accesării ei. De asemenea, trebuie menționate și permisiunile necesare aplicației. Permisiunile folosite în proiectul meu sunt:

```

<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

```

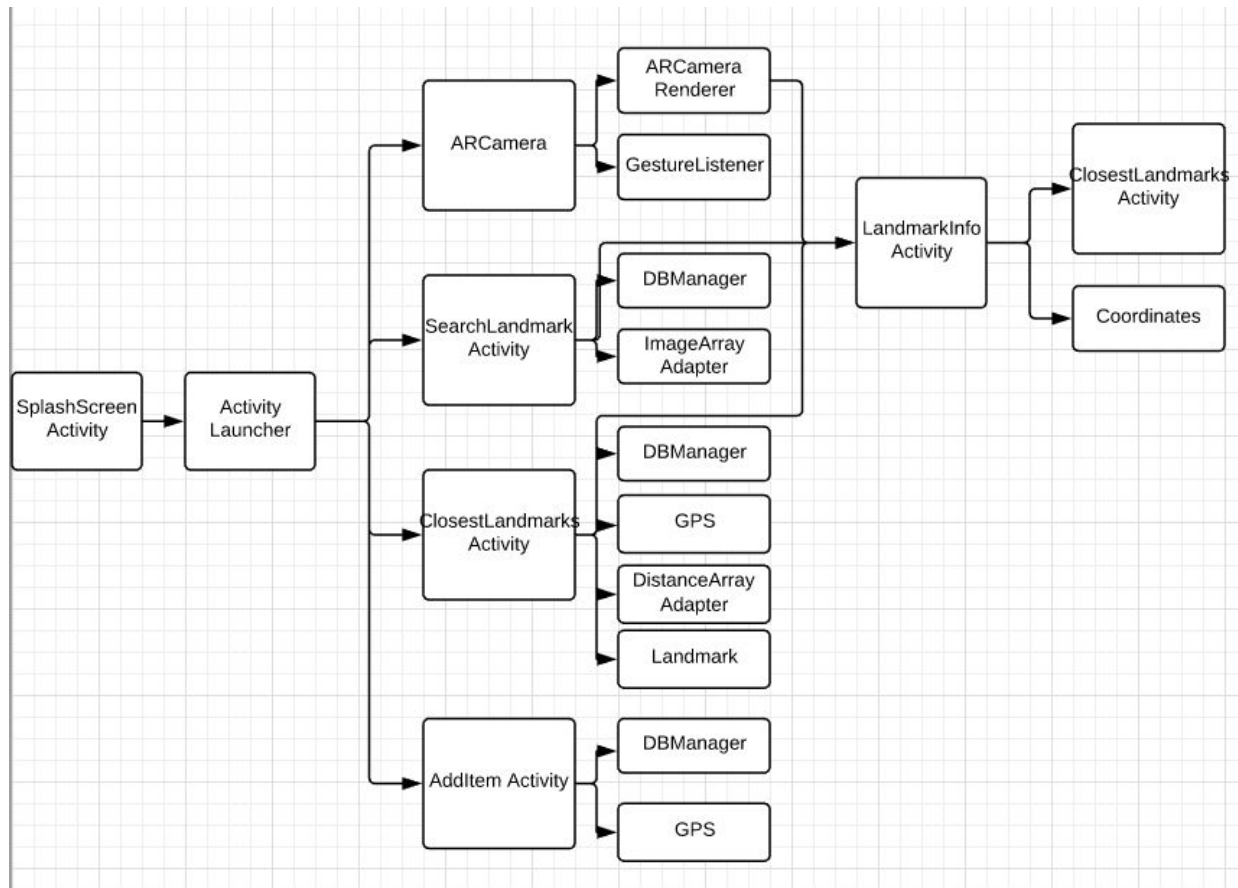


Figura 23: Împărțirea claselor pe fiecare modul în parte

Uneori, aplicațiilor le ia ceva timp să se deschidă, în mod deosebit când rulează pentru prima dată pe un dispozitiv. În loc de a lăsa clasicul ecran negru în aceasta etapa de deschidere, am ales utilizarea unui splash screen pentru o comoditate vizuală și o evidențiere de logo (Figura 24). După câteva milisecunde sau secunde (depinde de dispozitiv) este afișată pagina principală (Figura 25).



Figura 24: Splash screen-ul aplicației Pocket World

Clasa *ActivityLauncher* este responsabilă cu reunirea activităților principale. După cum se poate observa în Figura 27, layout-ul acesteia conține un *ImageView* utilizat pentru afișarea logoului aplicației și 4 butoane, fiecare declanșând funcționarea unui modul.

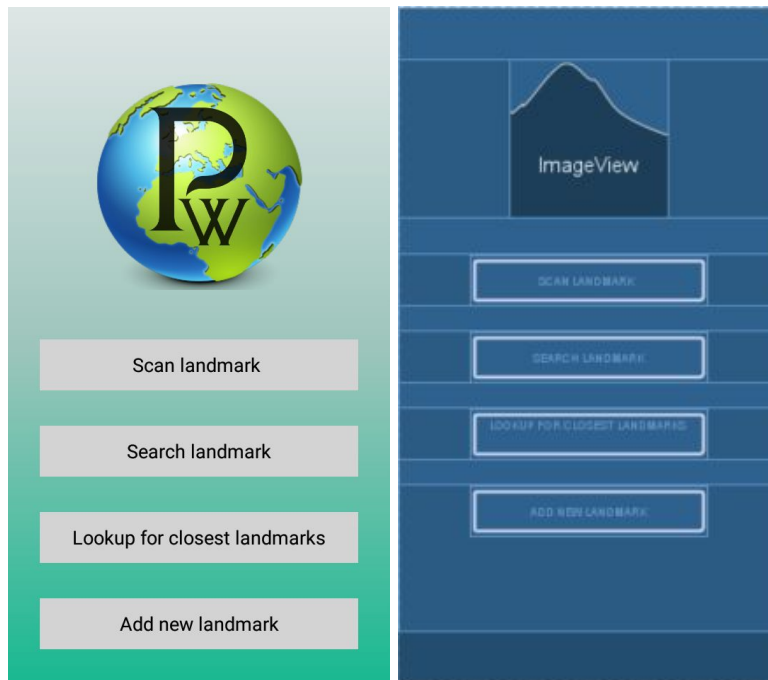


Figura 25: Pagina principală a aplicației Pocket World și layout-ul acesteia

În continuare voi detalia caracteristicile de implementare pentru fiecare modul în parte.

III.1. Scanează și identifică obiectivul

Acest modul are la bază, realitatea augmentă. El corespunde clasei ARCamera din Figura 23. După realizarea pașilor descriși în capitolul II pentru Vuforia, aceasta parte din proiectul realizat în Android Studio folosește în principal clasele ARCamera și ARCameraRenderer. Acestea, împreună cu celelalte clase apelate (AppRenderer, AppRendererControl, ApplicationSession, ApplicationControl, ApplicationException) sunt preluate de pe site-ul oficial ca o completare a librăriei Vuforia.jar. Ele sunt preluate separat și incluse în proiect, deoarece sunt destinate editării. Cu ajutorul lor, se pot proiecta obiecte 3D (create cu OpenGL) sau doar se pot gestiona imaginile identificate într-un mod simplu. În cazul aplicației PocketWorld, clasa ARCamera a fost utilizată pentru afișarea unui progress bar (LoadingDialogHandler, Figura 26), pentru declanșarea modului *focus* când se detectează o atingere a ecranului (GestureListener) și pentru apelarea clasei ARCameraRenderer, unde a fost introdusă cheia pentru accesul la Vuforia (Figura 18) și unde s-au gestionat imaginile.

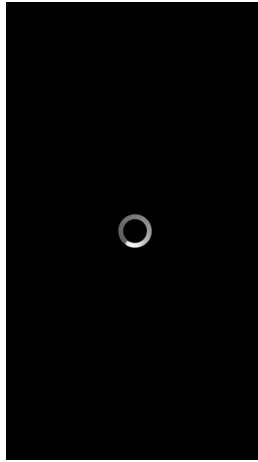


Figura 26: Progress bar afișat pe durata efectuării setărilor pentru Vuforia

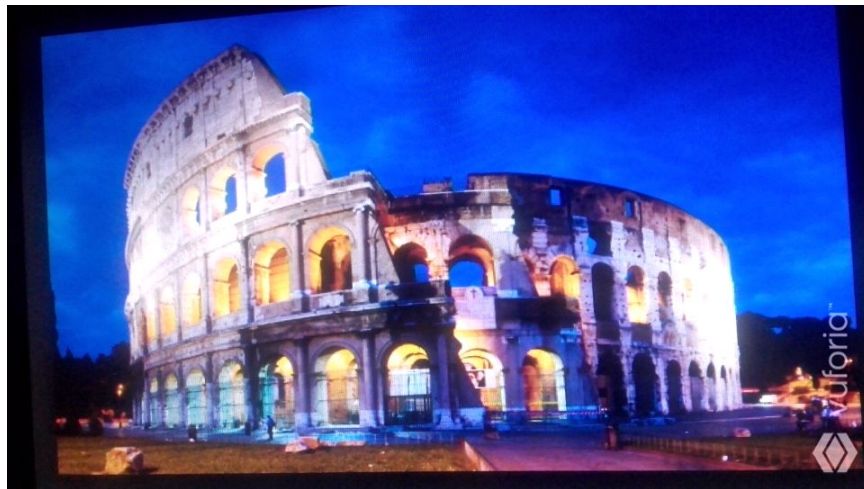


Figura 27: Camera inițializată cu setările pentru Vuforia

Deoarece numărul de obiecte *TrackableResult* este foarte mare, acestea fiind transmise aplicației cu o frecvență ridicată (câteva zeci pe secundă), am utilizat o metodă prin care se asigură existența și persistența obiectului identificat. Metoda constă în contorizarea obiectelor identificate până la o limită dată, dacă acestea apar sub același nume. Se permite, de asemenea și un număr mic de abateri. Spre exemplu, dacă se identifică continuu trackere cu numele *colosseum_x* și apare un tracker regăsit sub alt nume, acesta va fi ignorat, deoarece, la nivel de

milisecunda pot apărea și astfel de erori. Numărătoarea se reia în momentul în care numărul tracker-elor diferite este mai mare decât o limită dată.

În momentul în care numărul identificărilor aceluiși obiect ajunge la un număr predefinit, aplicația accesează activitatea *LandmarkInfoActivity*, o pagină ce extrage informații de pe Wikipedia și le afișează în timp real (Figura 28).

Asocierea între numele tracker-ului, al instanței din baza de date și al imaginii se realizează pe baza unor reguli de denumire. Spre exemplu, obiectivul cu numele *Statue of Liberty* din baza de date, va avea asociate imaginile înregistrate în Vuforia precum *statue_of_liberty_x* (numele doar cu litere mici, cuvintele delimitate de “_”, iar la final “_x”, unde “x” reprezintă un număr întreg), iar imaginea asociată din ListView (Figurile 29, 30) se va regăsi cu numele *statueofliberty* (utilizarea literelor mici și eliminarea delimitatorilor).

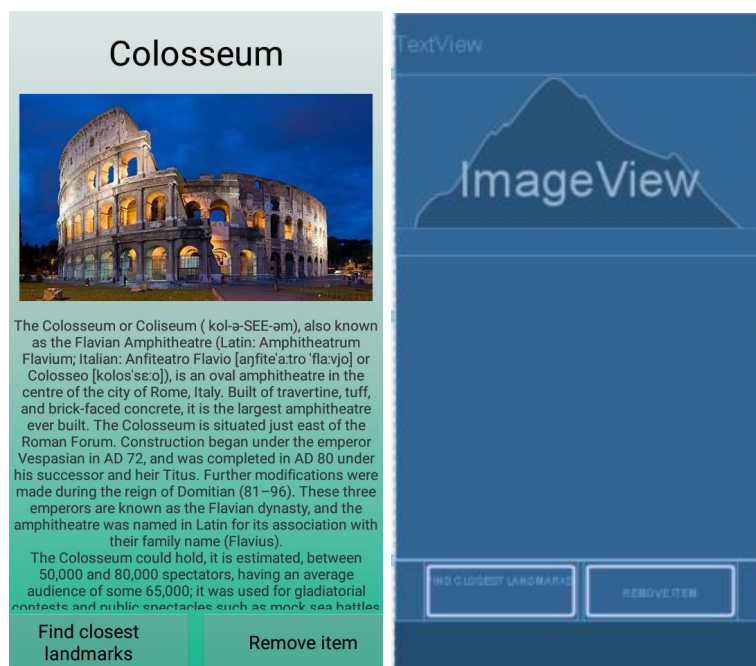


Figura 28: Pagina cu informațiile despre obiectul identificat Colosseumului (în stânga) și layout-ul asociat (în dreapta)

Din acest punct, utilizatorul are posibilitatea de a identifica obiectivele ce se află în același oraș cu obiectivul înregistrat în instanța activității *LandmarkInfoActivity* (apăsarea

butonului cu numele *Find closest landmarks*) sau de a-l șterge definitiv din baza de date (apăsarea butonului cu numele *Remove item*).

III.2. Caută un obiectiv într-o listă dată

Deoarece nu mereu avem la dispoziție o imagine a obiectivului dorit sau nu suntem în fața acestuia pentru a-l putea scana și identifica, utilizatorul are posibilitatea de a-l selecta dintr-o listă dată. Această listă poate reuni și puncte de interes care nu sunt neapărat reprezentate de niște obiecte precum statui sau clădiri, ci sunt niște locuri precum parcuri, ele neputând fi identificate doar pe baza unor tracker-e.

În acest scop am creat un modul al cărei funcționalitate este foarte simplă, ea oferind o listă cu toate obiectivele înregistrate în baza de date și o bară de căutare foarte eficientă pentru filtrarea lor după nume.

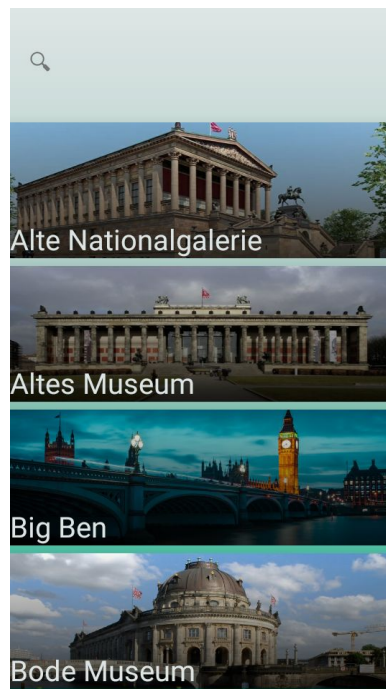


Figura 29: Căutarea și selectarea obiectivelor

Pentru implementare am utilizat următoarele clase:

- SearchLandmarkActivity:
 - Reprezintă activitatea principală, cea care gestionează și afișează lista cu obiectivele înregistrate în baza de date, ordonate alfabetic;
 - Este constituită dintr-un SearchView și un ListView. În funcție de textul introdus în bara de căutare, lista de elemente se modifică în timp real, fără a fi necesară apăsarea unui buton de tipul “Submit”. Căutarea nu este case sensitive (Figura 30).
 - Selectarea unui element determină deschiderea paginii cu informații despre obiectivul respectiv (Figura 31).

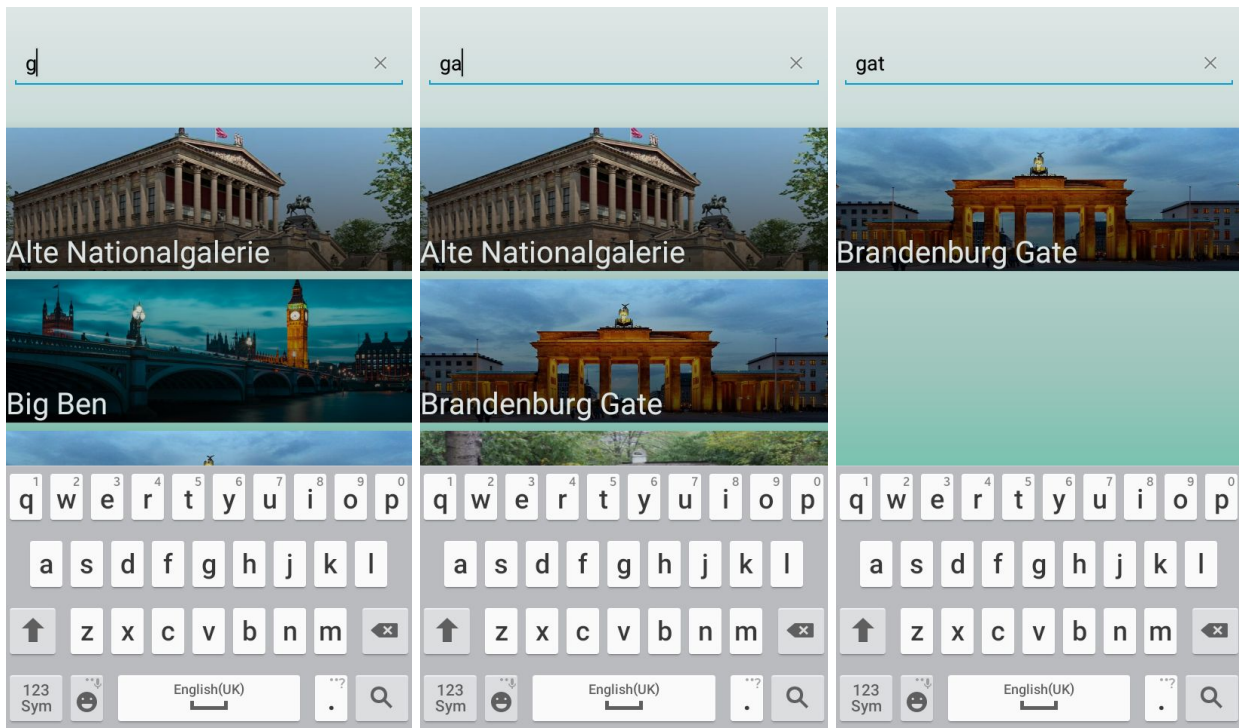


Figura 30: Căutare în timp real a obiectivelor din baza de date

- ImageArrayAdapter:
 - Clasa este realizată prin moștenirea clasei *ArrayAdapter<Object>*
 - Prin intermediul ei, în locul unei liste clasice de nume, utilizatorului i se arată o listă care pe lângă nume, conține și o imagine a obiectivului. Imaginile se obțin din folderul *res/drawable* în cazul cele predefinite de aplicație, sau din folderul

PocketWorld_images din memoria telefonului în cazul celor memorate în urma inserării unui nou element (III.4)



Figura 31: Element din listă, în urma atașării ImageArrayAdapter

- DBManager:
 - Reprezintă un adaptor pentru baza de date SQLite. Am realizat această clasă în vederea gestionării bazei de date cu numele *PocketWorld.db*, baza de date ce conține o singură tabelă în care sunt înregistrate informații despre obiective (nume, wiki id, latitudine, longitudine, oraș). Structura poate fi observată în Figura 20.
 - Caracteristicile cele mai importante ale acestei clase sunt: popularea bazei de date, returnarea rezultatelor unor interogări și ștergerea unor elemente.
 - Printre funcționalitățile cele mai importante ale acestei clase, se numără preluarea numelor obiectivelor, inserarea informațiilor despre un nou obiectiv, ștergerea unui obiectiv, preluarea coordonatelor, a orașului sau a wiki id-ului în funcție de nume. De menționat este faptul că wiki id-ul reprezintă numele distinctiv al unei căutări Wikipedia. Spre exemplu, pentru https://en.wikipedia.org/wiki/Statue_of_Liberty, wiki id-ul este *Statue_of_Liberty*.

III.3. Găsește locurile din apropierea ta

De cele mai multe ori, când ne aflăm într-un oraș necunoscut, nu suntem mereu conștienți de faptul că putem avea obiective turistice în jur deoarece suntem axați doar pe cele cunoscute.

De aceea, este foarte util când ai o aplicație care, prin apăsarea unui singur buton îți spune obiectivele din apropiere.

În acest sens, am creat acest modul reprezentat de următoarele clase:

- **ClosestLandmarksActivity:**
 - Activitatea principală care îmbină toate celelalte clase în vederea obținerii unor informații utile și plăcut afișate (Figura 32).
 - Are aproximativ aceeași structură precum *SearchLandmarkActivity* (Figura 25), cu diferența că îi lipsește *SearchView*-ul și are adaptor pentru *ListView* puțin diferit.
 - În momentul accesării activității din meniul principal, este pornită funcționalitatea de localizare. În urma pornirii acesteia, se extrag coordonatele, se identifică orașul, iar apoi se afișează obiectivele din orașul respectiv sortate crescător după distanța din punctul unde se afla telefonul/tableta până la locația obiectivului. În cazul în care aceasta activitate este accesată din *LandmarkInfoActivity*, coordonatele se extrag din baza de date pe baza numelui transmis.

```
private static final String LOCAL_CODE="LOCAL";
private static final String BROAD_CODE="BROAD";

Bundle extras=getIntent().getExtras();
if (extras!=null){
    code=extras.getString("CODE");
    if (code.equals(BROAD_CODE)){
        landmarkName=extras.getString("LANDMARK");
    }
    else{
        landmarkName="you";
    }
}
```

- Dacă clasa este accesată din meniul principal, se transmite prin intent codul **LOCAL_CODE**, în caz contrar, codul **BROAD_CODE**.



Figura 32: Afișarea obiectivelor din apropierea utilizatorului aplicației

- DBManager: descrisă mai sus, în secțiunea III.2
- GPS:
 - Această clasă este utilizată pentru preluarea coordonatelor locale, aflarea orașului în funcție de acestea, verificarea stării funcției GPS.
- DistanceArrayAdapter:
 - Diferența dintre acest adaptor și *ImageArrayAdapter*, prezentat anterior este că, pentru fiecare element în parte, se afișează și distanța până la obiectivul ilustrat.
 - Selectarea oricărui element, deschide pagina cu informațiile aferente lui extrase de pe Wikipedia (Figura 30).
- Landmark:
 - În această clasă am memorat informațiile relevante pentru un obiectiv (Figura 33). Cel mai important este faptul că calculează și reține distanța până într-un punct dat.

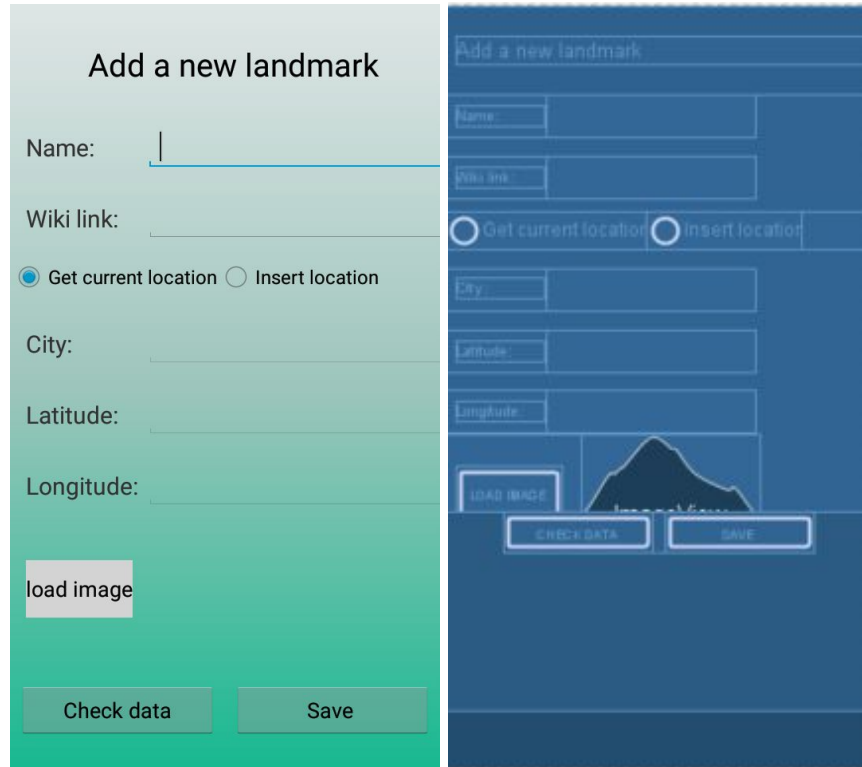
Landmark
<pre>private Double latitude; private Double longitude; private String name; private Double distanceThroughPoint;</pre>
<pre>public Double getLatitude(); public void setLatitude(Double latitude); public Double getLongitude(); public void setLongitude(Double longitude); public String getName(); public void setName(String name); public Double getDistanceThroughPoint(); public void setDistanceThroughPoint(Double distance); private double getDistanceBetween2Points(double lat1, double long1, double lat2, double long2);</pre>

Figura 33: Structura clasei Landmark

III.4. Adaugă un nou element

Este foarte posibil ca atunci când vizităm un oraș nou, să ni se sugereze sau să vrem să vizităm un anumit loc precum un restaurant cu specific sau un muzeu abia deschis sau poate să fim impresionați de un anumit punct și să vrem să-l înregistrăm pentru a ține cont pe viitor.

Acest caz a fost motivul realizării modulului de inserare a unui nou element în lista predefinită de obiective. El este reprezentat de clasa *AddItemActivity*, clasă ce utilizează instanțe ale DBManager și GPS, descrie la punctele anterioare.



Figurile 34: Adăugarea unui nou element

După cum se poate observa în Figura 34, utilizatorului îi va fi prezentat un formular unde, pentru inserarea și utilizarea mai apoi a unor informații cât mai precise, va trebui să completeze câmpurile *Nume*, *Wiki link*. Din acest punct, utilizatorul trebuie să opteze pentru preluarea coordonatelor locale (*Get current location*) sau pentru inserarea manuală a acestora (*Insert location*). Este oferită posibilitatea încărcării unei imagini din galeria telefonului pentru a o asocia listView-urilor.

În urma apăsării butonului *Check data*, sunt afișate datele introduse pentru a fi verificate de către utilizator (Figura 35).

Add a new landmark

City:

Latitude:

Longitude:

load image

Name: Home
City: Iași
Latitude: 47.1696276
Longitude: 27.5639425
WikiTail: random_wiki

Check data Save


Add a new landmark

City:

Latitude:

Longitude:

load image



Name: Pantheon
City: Rome
Latitude: 41.8910494
Longitude: 12.4443387
WikiTail: Pantheon_Rome

Check data Save

Figura 35: Verificarea datelor după inserare

Dacă datele sunt corecte, se salvează în baza de date prin simpla apăsare a butonului *Save*. După cum ilustrează Figura 36, datele au fost salvate.

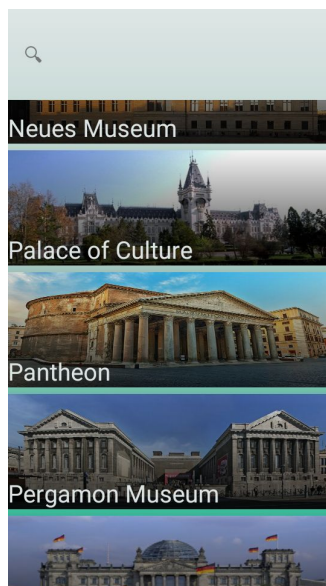


Figura 36: Existenta unei noi înregistrări în baza de date după adăugare

III.5. Concluzii

Capitolul III are rolul de a crea o imagine de ansamblu asupra arhitecturii aplicației și asupra implementării. În această secțiune am prezentat idei generale, porțiuni de cod, diagrame și schițe. Prezentarea fiecărui modul a fost conturată după diagramele de la începutul capitolului.

Capitolul IV: Scenarii de utilizare

Fiecare aplicație stârnește interesul oamenilor în momentul în care aceștia testează toate funcționalitățile în parte. Dacă din funcționalitățile testate, măcar 2-3 îi sunt pe plac unei persoane, ea va continua să folosească aplicația. În caz contrar, softul va fi deinstalat la scurt timp. În acest capitol voi prezenta câteva scenarii de utilizare însoțite de diagrame UseCase.

IV.1. Scenariul 1

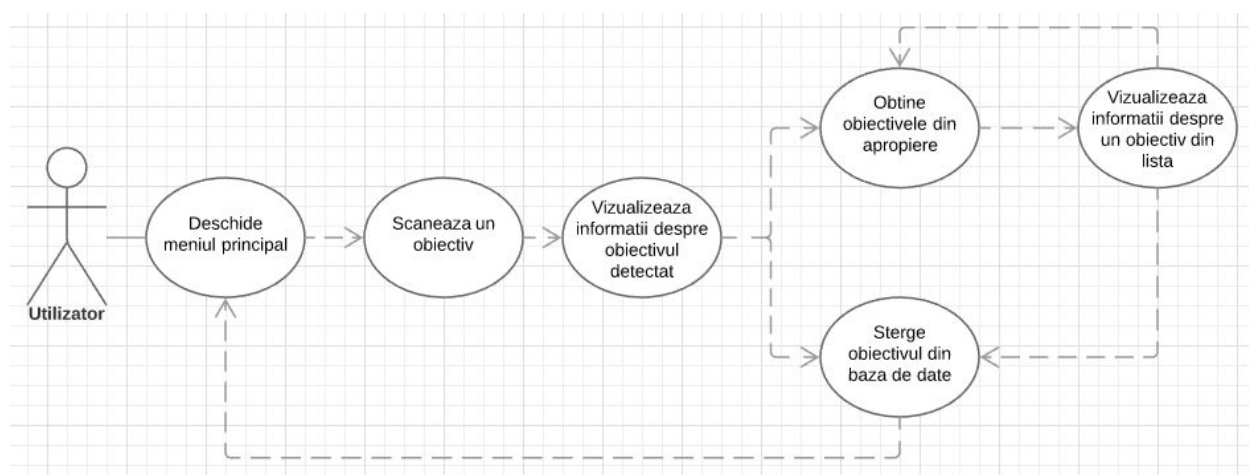


Figura 37: Scenariu de utilizare pentru realitatea augmentată

Primul scenariu ține de realitatea augmentată. Pentru a vedea cum funcționează trebuie respectați următorii pași. În primul rând trebuie deschisă aplicația la modulul *Scanează și identifica obiectul*. Cu ajutorul camerei de la telefon, vor putea fi identificate imaginile și obiectele plasate cu ajutorul realității augmentate. Pentru buna recunoaștere a unui obiectiv, este indicată înregistrarea a aproximativ 10 imagini din diferite unghiuri fără fundal. În spate, partea de recunoaștere a imaginii văzute prin intermediul telefonului este realizată cu ajutorul kitului de dezvoltare Vuforia. Acesta are un algoritm care caută în baza de date să vadă dacă poate găsi imaginea vizualizată. În caz afirmativ, se deschide o pagina cu informații de pe site-ul Wikipedia. Atenție însă, pentru ca trebuie să existe conexiunea la internet.

Din acel punct utilizatorul are posibilitatea să aleagă între a reveni la scanare, a obține o listă cu obiectivele învecinate sau a șterge obiectivul din baza de date.

IV.1. Scenariul 2

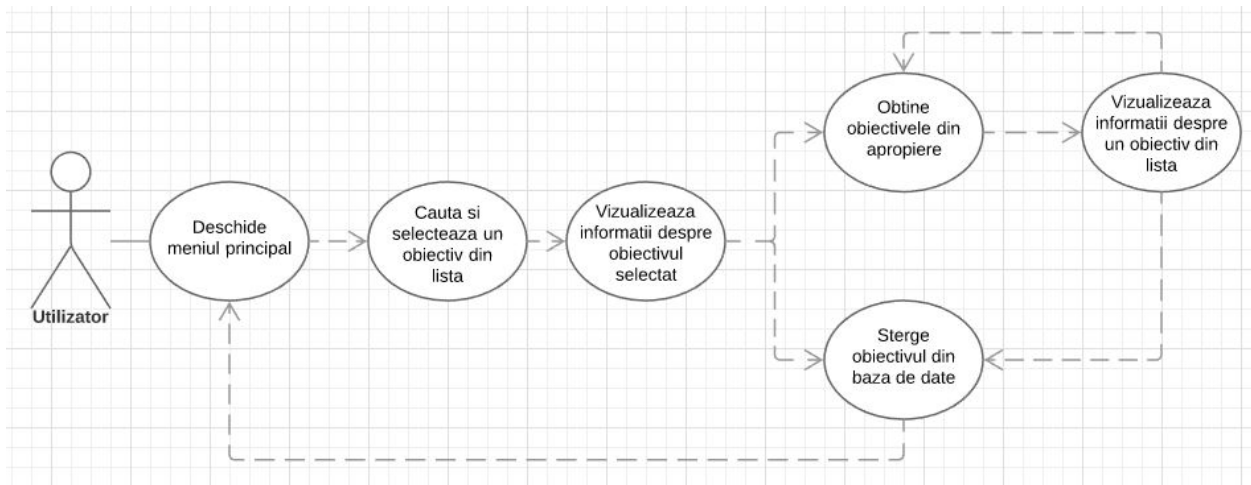


Figura 38: Scenariu de utilizare pentru selectarea manuala a obiectivului din lista

Să presupunem că utilizatorul nu se află în fața unui obiectiv, dar vrea să afle mai multe despre acesta. În acest caz, este suficient să acceseze modulul *Caută un obiectiv într-o lista dată*, să caute cu ajutorul barei de căutare și să selecteze un obiectiv din cele afișate. În continuare, se deschide pagina cu informații.

IV.1. Scenariul 3

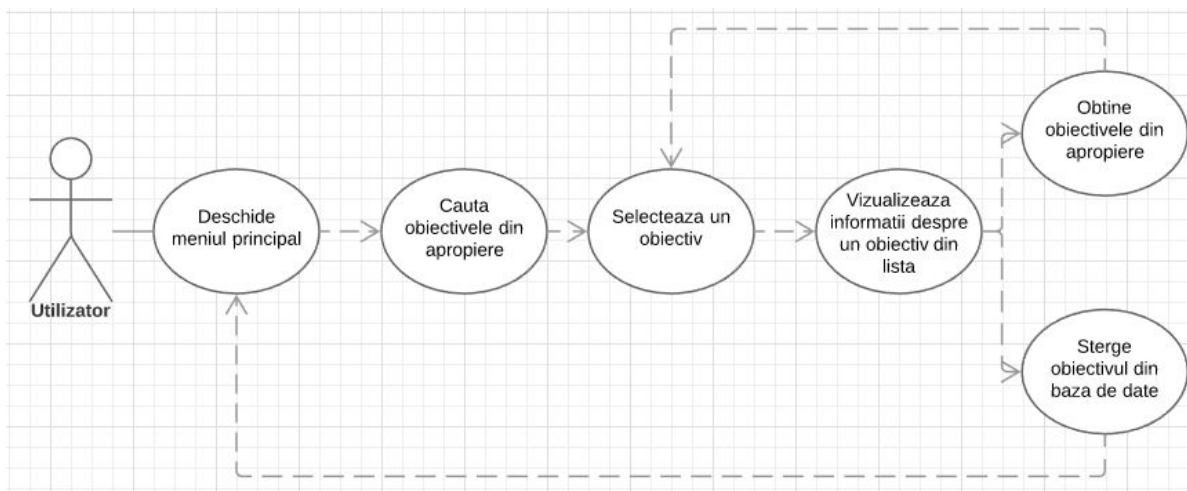


Figura 39: Scenariu de utilizare pentru determinarea obiectivelor din apropiere

De multe ori s-a întâmplat să fim într-un loc și să nu știm ce avem în apropiere. Modulul *Găsește locurile din apropierea ta* vine în ajutor. Acesta accesează GPS-ul și îți oferă o listă cu obiectivele din apropiere.

IV.1. Scenariul 4

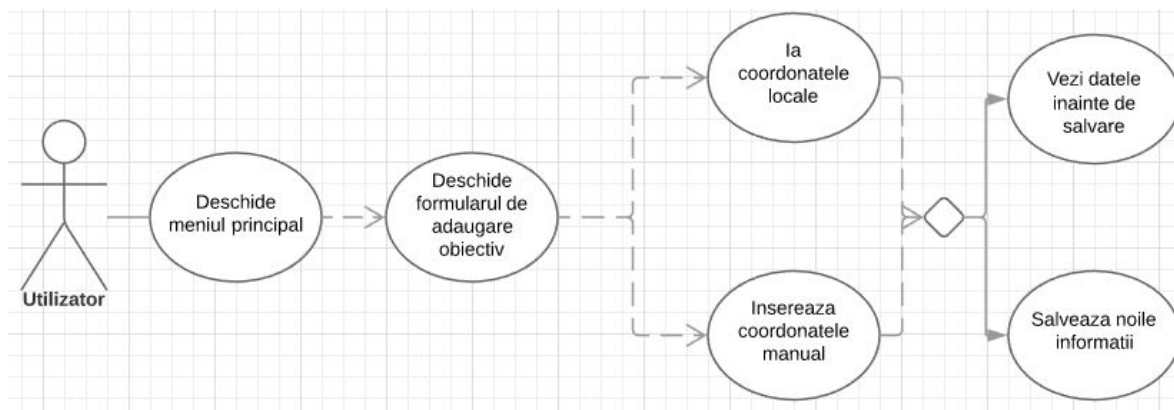


Figura 40: Scenariu de utilizare pentru adăugarea unui nou obiectiv în baza de date

În cazul în care printre obiectivele înregistrate în baza de date nu se regăsește unul dorit de utilizator, el are ocazia de a-l memora, prin simpla accesare a modulului *Adaugă un nou element*. Este afișat un formular de înregistrare ce conține câmpuri pentru *Nume*, *Wiki link*, *City*, *Latitude*, *Longitude*, un buton ce face posibilă selectarea unei imagini din galeria personală și 2 butoane radio pentru a alege între preluarea automată a coordonatelor locale sau inserarea manuală a acestora.

Deoarece regula de denumire a imaginilor ține de numele obiectivului, nu va fi permisă selectarea unei imagini dacă câmpul *Nume* este gol.

În cazul în care nu este introdus un link Wikipedia valid în câmpul *Wiki link*, nu vor putea fi afișate informații deoarece nu vor fi găsite (Figura 41).

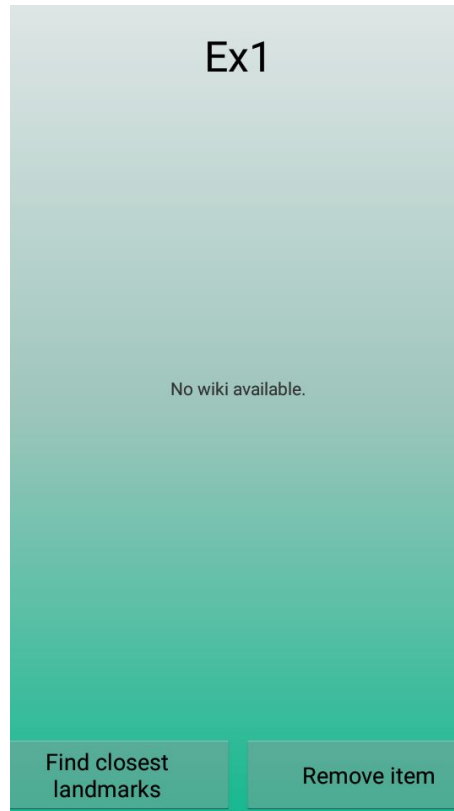


Figura 41: Link wiki invalid

Dacă este selectată opțiunea de preluare automată a coordonatelor, câmpurile *City*, *Latitude*, *Longitude* devin inactive, în caz contrar, e necesară completarea lor cu niște date valide.

În continuare, opțiunile sunt de verificare a datelor și de salvare a acestora.

IV.5. Concluzii

Rolul acestui capitol este de a crea o viziune de ansamblu asupra funcționalităților principale ale aplicației. Fiecare modul în parte are un grad mai mic sau mai mare de utilizare. Dacă primele 3 module (*Scanează și identifică obiectivul*, *Caută un obiectiv într-o lista dată*, *Găsește locurile din apropierea ta*) vor fi utilizate mai des, cel de-al 4-lea (*Adaugă un nou element*) se va utiliza doar în anumite momente.

V. Concluziile lucrării și posibilitatea extinderii aplicației în viitor

V. 1. Concluzii generale

Tehnologiile din ziua de astăzi sunt într-o continuă schimbare, iar acest lucru vine cu numeroase beneficii. Pe lângă ușurința transmiterii informațiilor, atenția utilizatorilor de software educațional este captată de noile idei de prezentare. Cum am menționat și în capitolele anterioare, atât copiii cât și adulții învață mai multe în această manieră și le este mai ușor să asocieze locuri și fapte față de cea clasică. În realizarea proiectului meu, am luat în considerare acest aspect. Raportat doar la mici ramuri ale geografiei, istoriei turismului, folosirea softului asigură reținerea noțiunilor generale ale țărilor din Europa.

Aplicația are rolul și de a dezvolta simțul de memorare al utilizatorilor cu ajutorul realității augmentate. Datorită faptului că prin simpla scanare a unui obiectiv sunt oferite informații generale despre acesta și posibilitatea asocierii cu alte obiective apropiate printr-o simplă apăsare de buton, se conturează o legătura mult mai strânsă între imagini, informații, evenimente descrise în pagina wiki.

În opinia mea, aplicațiile de acest tip aduc un beneficiu major în dezvoltarea cunoștințelor generale. Ideea de a învăța lucruri noi prin intermediul asocierilor este una destul de bună deoarece elevii vor memora unele lucruri fără a fi conștienți de acest lucru. Același lucru se aplica și în cazul adulților, la ei adăugându-se și noțiunea de comoditate în orientare. O călătorie spre un oraș nou poate deveni un răgaz perfect pentru a afla niște informații despre obiectivele ce îl marchează, fără prea mult efort sau prea multe căutări.

V. 2. Muncă viitoare

Consider că această aplicație poate deveni mult mai complexă și își poate lărgi aria de răspândire. Ideea mea generală, înainte de a mă apuca de proiectul în sine, a fost să creez o platformă care să îmbine realitatea augmentată cu partea de turism. Un fel de manual al obiectivelor turistice cu acces ușor și legături rapide între ele.

Această aplicație poate migra de la nivel local, la stadiul de aplicație în rețea, unde orice obiectiv adăugat este înregistrat pe un server, verificat și în urma unui accept să fie integrat în instanțele aplicației de pe toate telefoanele sau tabletele.

De asemenea, se pot realiza legături strânse cu GoogleMaps pentru a oferi, pe lângă lista cu obiective alăturate, o harta cu acestea și moduri de a ajunge la ele.

Aplicația poate face apel la site-urile oficiale ale obiectivelor prezente în baza de date pentru a oferi utilizatorilor informații precum program de acces, preț pentru intrare, densitatea vizitatorilor în funcție de oră, etc.

Și de ce să nu putem uni și cu site-urile de turism? Nu ar fi ideal să avem un asistent care să urmărească prețurile și să ne plănuiască un city break spre locațiile preferate cu niște obiective pe care dorim să le vedem?

În opinia mea, această aplicație poate fi baza unui asistent de călătorii, care să ne țină la curent cu obiectivele din jurul nostru, cu istoria lor, cu modalități de a ajunge la ele fără a căuta zilnic informații despre orașe, prețuri pentru zboruri sau cazări.

Bibliografie

Bibliografie

- [1] Happymagenta (2018) *Spyglass* <http://happymagenta.com/spyglass/index.html> (accesat ultima dată în Februarie 2018)
- [2] Anatomy 4D (2018) *Anatomy 4D. An extraordinary journey inside the human body and heart.* DAQRI <http://anatomy4d.daqri.com> (accesat ultima dată în Februarie 2018)
- [3] GooglePlay (2018) *Star Walk - Sky View: Explore the Stars.* Google. <https://play.google.com/store/apps/details?id=com.vitotechnology.StarWalk&hl=ro> (accesat ultima dată în Februarie 2018)
- [4] GooglePlay (2018) *Star Walk 2 Free - Identify Stars in the Sky Map.* Google. <https://play.google.com/store/apps/details?id=com.vitotechnology.StarWalk2Free&hl=ro> (accesat ultima dată în Februarie 2018)
- [5] VitoTechnology (2018) *Star Walk. Stargazing guide.* Vito Technology, Inc. <http://vitotechnology.com/star-walk.html> (accesat ultima dată în Februarie 2018)
- [6] iTunes (2018) *Star Walk 2 - Night Sky Map 4+ Constellation Star Finder AR.* Vito Technology Inc. <https://itunes.apple.com/us/app/star-walk-2-night-sky-map/id892279069?mt=8> (accesat ultima dată în Februarie 2018)
- [7] Android (2014) *The world's most popular mobile OS. From phones and watches to cars and TVs, customize your digital life with Android.* Astrium, DigitalGlobe. <https://www.android.com/> (accesat ultima dată în Februarie 2018)
- [8] AppBrain (2018) *Number of Android applications.* AppBrain. <http://www.appbrain.com/stats/number-of-android-apps> (accesat ultima dată în Februarie 2018)
- [9] Google+ (2018) *Android Customizers* <https://plus.google.com/communities/108472674452767095777/stream/688f802a-64f2-48eb-915f-f7a7a1e4f145> (accesat ultima dată în Februarie 2018)

- [10] Android (2018) *Android Studio. The Official IDE for Android*. <https://developer.android.com/studio/index.html> (accesat ultima dată în Februarie 2018)
- [11] JetBrains (2018) *IntelliJ IDEA: The Java IDE for Professional Developers by JetBrains*. <https://www.jetbrains.com/idea/> (accesat ultima dată în Februarie 2018)
- [12] Vuforia (2018) *Vuforia SDK to build Android, iOS, and UWP applications for mobile devices and digital eyewear*. <https://developer.vuforia.com/downloads/sdk> (accesat ultima dată în Februarie 2018)
- [13] Wikipedia (2018) Computer Vision. https://en.wikipedia.org/wiki/Computer_vision (accesat ultima dată în Februarie 2018)
- [14] Wikipedia (2018) Android. [https://ro.wikipedia.org/wiki/Android_\(sistem_de_operare\)](https://ro.wikipedia.org/wiki/Android_(sistem_de_operare)) (accesat ultima dată în Februarie 2018)
- [15] Vuforia (2018) *How To Use the Trackable Base Class*. Developer Portal. <https://library.vuforia.com/articles/Solution/How-To-Use-the-Trackable-Base-Class.html> (accesat ultima dată în Februarie 2018)