



UNIVERZITET U BEOGRADU  
FAKULTET ORGANIZACIONIH NAUKA

**ZAVRŠNI (MASTER) RAD**

**RAZVOJ MOBILNE APLIKACIJE ZA UČENJE JAPANSKOG  
JEZIKA ZASNOVANE NA OBJEKTIMA UČENJA**

Mentor:

Prof. dr Marijana Despotović-Zrakić

Student:

Miloš Milutinović 3502/2011

Beograd  
Jul, 2012

Komisija koja je pregledala rad  
kandidata MILUTINoviĆ (ZORAN) MILOŠA  
pod naslovom RAZVOJ MOBILNE APLIKACIJE ZA UČENJE JAPANSKOG JEZIKA ZASNOVANE  
NA OBJEKTIMA UČENJA i odobrila odbranu:

Mentor: dr Marijana Despotović-Zrakić, vanredni profesor

---

Član: dr Božidar Radenković, redovni profesor

---

Član: dr Dejan Simić, redovni profesor

---

## APSTRAKT

U ovom radu je predstavljena mobilna aplikacija za učenje japanskog jezika razvijena na Android platformi za mobilne uređaje, u kojoj su primjenjeni koncepti objekata učenja i učenja kroz igru, i njen razvoj.

Koncept objekata učenja predstavlja osnovu osmišljenog procesa učenja – kompletan edukativni sadržaj je distribuiran u niz objekata učenja, malih, kompaktnih jedinica sastavljenih iz nekoliko vrsta sadržaja koji se mogu menjati nezavisno od aplikacije i agregirati u veće celine u skladu sa zahtevima korisnika; korišćeni model objekata učenja je razvijen kao aplikacioni profil IEEE LOM-a prilagođen potrebama aplikacije, prvenstveno uprošćavanjem i izbacivanjem nepotrebnih elemenata.

U radu su opisani arhitektura Android aplikacije, njene osnovne komponente i sve funkcionalnosti dostupne korisniku, korišćeni model objekata učenja koji opisuje i podatke i metapodatke, kao i komplementarna PHP aplikacija koja pruža podršku Android aplikaciji u vidu perzistentnog skladištenja edukativnih sadržaja i korisničkih podataka.

U funkcionalnosti aplikacije spadaju: učenje reči uz odgovarajuće audio zapise, učenje pisama japanskog jezika, testiranje znanja, vežbanje pisanja znakova korišćenjem tač-skrin funkcionalnosti, poređenje napretka sa drugim korisnicima aplikacije i nekoliko aktivnosti motivacionog karaktera.

Na polju motivacije korisnika, korisnicima je ponuđena aktivnost igre pamćenja znakova, sistem nagrađivanja za napredak prezentacijom zanimljivosti o Japanu i jednostavan pristup pregledu opštih statistika i poređenju istih sa drugim korisnicima.

Ključne reči: mobilno učenje, MALL, objekti učenja, učenje kroz igru

## CURRICULUM VITAE

Miloš Milutinović

E – mail: knezmilosXIII@gmail.com

### LIČNI PODACI:

- Rođen 25. marta 1988.
- Mesto stanovanja: Kedrova 13/13, Beograd

### OBRAZOVANJE:

- Diplomirao na fakultetu organizacionih nauka 02. septembra 2011.

### DODATNE KVALIFIKACIJE:

- Objavljeni radovi:

Miloš Milutinović, Dušan Barać, Marijana Despotović-Zrakić, Aleksandar Marković, Božidar Radenković, „Razvoj mobilne aplikacije za učenje japanskog jezika – FONJAPGO“, Management, vol. 60, pp 27-33, septembar 2011.

Miloš Milutinović, „Dostavljanje objekata mobilnog učenja za proučavanje japanskog jezika“, Infoteh, Jahorina, Hotel "Bistrica", 21 – 23. mart 2012.

# SADRŽAJ

Sadržaj.....	v
Lista slika .....	vi
Lista tabela.....	vii
1. Uvod .....	1
1.1. Mobilno obrazovanje i njegovo mesto u modernom društvu.....	2
1.2. Personalna motivacija i ciljna grupa .....	2
1.3. Kraći pregled i opis strukture samog rada .....	4
2. Mobilno obrazovanje .....	6
2.1. Definicije.....	7
2.2. Perspektive .....	7
2.3. Specifičnosti .....	8
2.4. Mobilne platforme i tehnologije .....	10
2.5. Android platforma .....	12
2.5.1. Verzije Android operativnog sistema.....	13
2.5.2. Arhitektura Android sistema.....	17
2.5.3. Životni ciklus aktivnosti .....	21
2.5.4. Resursi u Android aplikacijama.....	25
2.5.5. Korisnički interfejs .....	28
3. Učenje jezika potpomognuto mobilnim tehnologijama.....	30
3.1. Tehnologije obuhvaćene MALL-om .....	31
3.1.1. Telefonski pozivi & SMS poruke.....	32
3.1.2. Snimanje/reprodukacija audio zapisa .....	33
3.1.3. Internet i mogućnosti obrade i interakcije sa korisnikom .....	34
3.2. Praktični radovi na temu učenja jezika uz pomoć mobilnih tehnologija.....	35
3.3. Objekti učenja .....	36
3.3.1. Definicije i karakteristike .....	36
3.3.2. Primena objekata učenja u konkretnim aplikacijama .....	38
3.4. Japanski jezik .....	40
3.5. Postojeće komercijalne aplikacije za učenje japanskog jezika .....	41
3.5.1. Desktop rešenja .....	42
3.5.2. Online rešenja .....	43
3.5.3. Mobilna (Android) rešenja .....	44
4. Razvoj JapGO aplikacije .....	45
4.1. Projektovanje aplikacije.....	45
4.1.1. Korisnički zahtevi .....	45
4.1.2. Dijagram slučajeva korišćenja .....	46
4.1.3. Slučajevi korišćenja.....	47
4.1.4. Model objekata učenja .....	55
4.1.5. Model podataka – PMOV, relacioni model.....	57
4.2. Arhitektura aplikacije.....	65
4.2.1. Domenske klase .....	67
4.2.2. Korisnički interfejs i tok kretanja korisnika.....	69
4.2.3. Veb aplikacija i komponente za komunikaciju sa njom.....	71

4.2.4.	Lokalna baza i komponente za komunikaciju sa njom.....	74
4.2.5.	Komponente za komunikaciju sa Android sistemom .....	76
4.2.6.	Mehanizam za generisanje testova .....	78
4.2.7.	Aktivnost – vežbanje pisanja.....	79
4.2.8.	Taksativno navedene komponente: .....	82
4.3.	Funkcionalnosti aplikacije .....	83
4.3.1.	Prvo pokretanje aplikacije.....	84
4.3.2.	Učenje reči.....	85
4.3.3.	Sekcija za učenje pisama.....	87
4.3.4.	Učenje znakova .....	87
4.3.5.	Igra pamćenja .....	90
4.3.6.	Vežbanje pisanja .....	90
4.3.7.	Testiranje naučenog .....	94
4.3.8.	Pregled i poređenje statistika korisnika .....	95
4.3.9.	Sistem za nagrađivanje korisnika zanimljivostima o Japanu .....	97
4.3.10.	Centralni proces učenja .....	97
4.3.11.	Podešavanja aplikacije & help sistem.....	98
5.	Zaključak .....	99
	Literatura .....	101

## **LISTA SLIKA**

Slika 1.	Arhitektura Android platforme.....	17
Slika 2.	Životni ciklus aktivnosti .....	23
Slika 3.	Različit prikaz za različitu rezoluciju uređaja .....	26
Slika 4.	Hijerarhijski prikaz elemenata korisničkog interfejsa .....	29
Slika 5.	Dijagram slučajeva korišćenja .....	46
Slika 6.	Model edukativnog sadržaja objekata učenja.....	56
Slika 7.	Model metapodataka objekta učenja .....	57
Slika 8.	PMOV model podataka korišćen u web aplikaciji .....	58
Slika 9.	PMOV model podataka korišćen u Android aplikaciji.....	61
Slika 10.	Arhitektura JapGO aplikacije .....	66
Slika 11.	Dijagram domenskih klasa .....	68
Slika 12.	Klase paketa "interfejs" i tok kretanja korisnika .....	70
Slika 13.	Struktura paketa „web“ i web aplikacije .....	73
Slika 14.	Struktura paketa „baza“ .....	75
Slika 15.	Klase paketa "util" i njihove funkcije u aplikaciji .....	77
Slika 16.	Mehanizam za učitavanje i prikaz pitanja na testu.....	79
Slika 17.	Dijagram klasa koje učestvuju u generisanju pitanja .....	79
Slika 18.	Aktivnost za vežbanje pisanja .....	81
Slika 19.	Prikaz osnovnih funkcionalnosti FONJAPGO aplikacije.....	84
Slika 20.	Početna podešavanja aplikacije .....	85
Slika 21.	Početni ekran aplikacije .....	85
Slika 22.	Prikaz i pretraga reči iz baze podataka .....	86
Slika 23.	Detaljan prikaz reči .....	86
Slika 24.	Prikaz sekcije "Writing" pri: a) vertikalnoj orijentaciji b) horizontalnoj orijentaciji ..	87

Slika 25. Neke od stranica sa kana znacima u okviru "Learn Kana" aktivnosti.....	88
Slika 26. Detaljan prikaz kana znakova.....	88
Slika 27. Stranice sa kandi znacima.....	89
Slika 28. Detaljan prikaz kandi znakova.....	89
Slika 29. Podešavanje igre pamćenja i dve različito podešene instance.....	90
Slika 30. Aktivnost za vežbanje pisanja pri: a,b) vertikalnoj orijentaciji b) horizontalnoj orijentaciji .....	91
Slika 31. a) Početni prikaz b) Prikazan odeljak za kontrolu vodilja c) Uključene vodilje d) Sakriven odeljak za kontrolu vodilja .....	91
Slika 32. Aktivnost za crtanje sa ugašenim i uključenim fade efektom .....	92
Slika 33. Izgled minimiziranog interfejsa zavisno od drugih podešavanja aktivnosti .....	92
Slika 34. Podešavanja "papira", interfejsa i vodilja.....	93
Slika 35. Podešavanja "papira" i odgovarajući izgled aktivnosti .....	93
Slika 36. Transparentnost maksimiziranog i minimiziranog interfejsa.....	94
Slika 37. Pozicioniranje vodilja i izgleda aktivnosti po pozicioniranju.....	94
Slika 38. a) Odeljak "Tests" b) Podešavanja testa c,d) Instance testa sa različitim podešavanjima.....	95
Slika 39. Pregled statistika .....	95
Slika 40. Izmena sličice profila.....	96
Slika 41. a,b) Registracija korisnika c,d) Dodavanje prijatelja .....	96
Slika 42. a) Otključavanje zanimljivosti tokom rada b) odeljak sa zanimljivostima c) prikaz jedne od kategorija d) prikaz jedne od zanimljivosti .....	97
Slika 43. Spisak raspoloživih lekcija i prikaz jedne od lekcija (u procesu/pređene) .....	98
Slika 44. a) Globalni meni, b) Podešavanja, c) Help kategorije, d) Pregled izabrane help teme .....	98

## **LISTA TABELA**

Tabela 1. Metode za upravljanje životnim ciklusom aktivnosti.....	24
Tabela 2. Kako organizovati resurse u Android projektu.....	27
Tabela 3. Slučajevi korišćenja.....	47
Tabela 4. Komponente aplikacije .....	82

## **1. UVOD**

Jezik je, počevši od svog nastanka negde u davnoj ljudskoj istoriji, služio kao sredstvo komunikacije među ljudima. Ipak, različita plemena i narodi su govorili različitim jezicima, što je predstavljalo nepremostivu barijeru za međusobno razumevanje i saradnju. Da je poznavanje stranih jezika važno, govori nam i stara latinska poslovica: „koliko jezika govorиш, toliko ljudi vrediš“. Ovo je pogotovo istina u našem informacionom dobu, kada su ljudi sa svih strana planete, različitih rasa, jezika i vera, povezani modernim komunikacionim tehnologijama i praktično samo na jedan „klik“ udaljeni jedni od drugih. Informaciono-komunikacione tehnologije su nastale kako bi pomogle povezivanju i razmeni informacija, misli i ideja, i upravo ih ta sličnost i povezanost koju imaju sa jezikom čini idealnim kandidatom za pomoć u učenju jezika.

Čak i ako ne unesemo nikakvu promenu u sam proces učenja stranog jezika, nove tehnologije nam neizmerno pomažu da pristupimo obilju dostupnih sadržaja. Knjige za učenje stranih jezika, rečnici, literatura na drugim jezicima se može naći u riznicama znanja širom interneta; razni testovi, kvizovi, kartice za propitivanje se mogu naći po sajtovima i blogovima posvećenim jeziku kojeg želimo da naučimo; dostupna muzika i video materijali (filmovi, serije itd) omogućavaju da proces učenja od napornog postane zabavan. Ukoliko ne uspemo da nađemo odgovarajuće resurse, možemo se lako povezati sa drugima koji uče isti jezik ili čak sa maternjim govornicima čije je znanje živog jezika vrednije od običnih reči na papiru.

Ako odemo još i korak dalje, otkriva nam se more mogućnosti za uključenje modernih tehnologija u proces učenja jezika. Prenosivi inteligentni uređaji nam omogućavaju da se na svakom mestu u svako doba dana, kada nama to najviše odgovara, možemo prihvati učenja, time olakšavajući taj, obično spor i dugotrajan proces. Ovo otvara vrata i za čitav nov soj studenata jezika – onih koji jezike uče iz hobija, iz zabave, kao način da razgibaju sopstvene intelektualne sposobnosti. Tu je i princip učenja kroz igru, implementiran kroz razne igre kucanja, prepoznavanja reči, kvizove, čitave avanturističke kompjuterske igre zasnovane na učenju jezika i razne aplikacije edukativno-zabavnog tipa.

Na kraju, ključne reči koje karakterišu ulogu modernih tehnologija u oblasti učenja su: lak pristup informacijama, mobilnost/prenosivost, povezanost, inovativni načini predstavljanja

informacija; pridevi koji opisuju srž modernih tehnologija uopšte i pokazuju da ima dosta mesta za njihovu primenu u procesu učenja stranih jezika.

## **1.1. Mobilno obrazovanje i njegovo mesto u modernom društvu**

Danšnje, moderno društvo karakteriše visok stepen mobilnosti u svim pogledima. Pojedinci nisu više geografski ograničeni; postoji ceo niz zanimanja čije se osnovne delatnosti obavljaju u pokretu, a takođe se javlja i specifična vrsta poslova, tipično visoko-tehnološki orijentisanih, koji napuštaju klasičan pojam radnog vremena. Čak i izvan posla, nepresušni izvori zabave i načini da se utroši vreme vode hroničnom nedostatku slobodnog vremena i manjoj spremnosti pojedinaca da posvete duži, kontinuirani vremenski period pojedinim aktivnostima.

U takvim uslovima, mnoge aktivnosti, pa i obrazovanje, su prisiljene da se menjaju i prilagode novonastalim uslovima. Veće ulaganje vremena u jednu aktivnost kao što je učenje danas predstavlja luksuz koji većina ne može sebi da priušti. Sa druge strane, vrednost informacija i znanja nikada nije bila veća. Kao odgovor na zahteve i potrebe modernog društva, nameće se razvoj novih paradigmi u oblasti obrazovanja, pogotovo onih koje se oslanjaju na mobilne tehnologije. Zahvaljujući razvoju mobilnih platformi javlja se mogućnost i realna potreba implementacije dela ili čitavog procesa učenja na mobilnim uređajima čime bi se omogućilo učenje u pokretu i kratkim vremenskim intervalima, odnosno bilo gde, bilo kad.

Najveći napredak i potencijal u tom pogledu postoji u oblasti učenja jezika, koja je univerzalno privlačna, rasprostranjena i gde je gradivo pogodno za finiju granulaciju podređenu uslovima mobilnog učenja. Pored zadovoljavanja potreba mobilnosti, mobilne aplikacije za učenje treba da unaprede proces učenja i da na sve moguće načine pokušaju da dodatno motivišu učenike. Takođe, čini se da se pojam objekata učenja prirodno uklapa u jedno ovakvo okruženje i pruža potencijal za obogaćivanje procesa učenja.

## **1.2. Personalna motivacija i ciljna grupa**

Ovaj rad predstavlja aktualizaciju nekoliko mojih interesovanja i mišljenja koja su se oformila tokom studija. Pre svega, baveći se informaciono komunikacionim tehnologijama koje karakterišu česte promene našao sam se u poziciji gde sam se većinu vremena oslanjao u

znanja dobavljeni preko mreže svih mreža – Interneta. Ovo je polako počelo da menja moju perspektivu na obrazovanje uopšte. I pored ogromnog uticaja modernih tehnologija na klasično, formalno obrazovanje, pomeranje granica se nije odvijalo u učionicama i amfiteatrima već napolju, na ulicama i u domovima i na kraju, u sajber-prostoru koji je predstavljao jedno čvrsto vezivno gradivo između učenika i učitelja, knjige i čitaoca i učionice i doma. Takođe, velika raznovrsnost i maštovitost edukativnih sadržaja ih čini privlačnim čak i za hobističko učenje u slobodno vreme. Na kraju, počeo sam da se interesujem za proces sticanja znanja uopšte i formirao sam stav – sve je moguće naučiti, dokle god su dostupni odgovarajući materijali, a samo učenje će biti daleko efikasnije ako je sadržaj prezentovan na zanimljiv način i, pre svega, ako postoji dovoljno motivacije, koja se pak može podstići prezentacijom.

Na izbor teme su značajno uticali lični razolzi, u ovom slučaju dugogodišnje bavljenje japanskim jezikom. Kao neko ko uči japanski jezik u slobodno vreme a opet ima odgovarajuću tehničku podlogu, istovremeno sam mogao da se smestim i u ulogu kreatora jedne aplikacije za učenje japanskog jezika ali i u ulogu korisnika te aplikacije. Moj proces učenja se zasnivao na učenju malih jedinica gradiva, prosečno 15 minuta dnevno, povremeno i u pokretu iz pomoć jednostavne aplikacije za listanje znakova japanskog pisma. Uočio sam da, iako postoje odgovarajuće aplikacije za mobilne uređaje, tipično su bile manjeg obima i bavile se samo nekim od aspekata japanskog jezika (npr. samo ponavljanjem znaka, samo rečnik itd.), a svakako nije postojala ni jedna aplikacija koja je dozvoljavala rad na srpskom jeziku.

Samim tim, pokušao sam da razvijem aplikaciju koja bi bila nešto opširnija, koja bi mogla pre svega i meni da pomogne u učenju i koja bi bila fleksibilnija u pogledu edukativnih sadržaja koje može da prezentuje. Postojeće aplikacije su se tipično ograničavale na jedan jezik, u najboljem slučaju nudeći nekoliko verzija aplikacije za nekoliko jezika. JapGO aplikacija treba da demonstrira jednu efektivnu kombinaciju raznovrsnih aktivnosti i vidova prezentacije uz korišćenje edukativnih sadržaja koji su fleksibilni, potencijalno na raznim jezicima i usmereni na različite teme.

Rezultat ovog rada bi trebalo da bude razvijena aplikacija za učenje japanskog jezika koja bi zadovoljila niz postavljenih zahteva koji proizilaze iz početnog problema – mobilnost i učenje u pokretu, poboljšanje kvaliteta celokupnog procesa učenja, primenu motivacionih tehnika koje bi mogle da privuku učenike i pozitivno deluju na njihovo zadovoljstvo i, sa tehničke strane, adekvatnu primenu objekata učenja u aplikaciji kako bi se omogućila višestruka

fleksibilnost edukativnih sadržaja – u pogledu dodavanja novih i ažuriranja postojećih sadržaja bez izmena u aplikaciji, u pogledu procesa učenja i u pogledu strukture i trajanja lekcija koje se prezentuju korisnicima.

U skladu sa postavljenim zahtevima, gotova aplikacija će biti prilagođena učenju u pokretu i idealna za one koji nemaju previše slobodnog vremena ili se često kreću. Ponađeni edukativni sadržaj će biti raznovrsan i fleksibilan pa u pogledu nivoa predznanja neće postojati ograničenja. Što se tiče samog procesa učenja, opet zahvaljujući fleksibilnosti objekata učenja a i same aplikacije, biće omogućeno sprovođenje različitih procesa učenja prema potrebama korisnika, pa će aplikacija moći da zadovolji i one koji je koriste kao osnovu svog procesa učenja i one koji je koriste samo kao ispomoć, za ponavljanje i utvrđivanje znanja. Na kraju, primenom multijezičnosti, aplikaciju će moći da koriste korisnici sa bilo kog govornog područja za koje su izrađeni odgovarajući objekti učenja.

### **1.3. Kraći pregled i opis strukture samog rada**

Predmet ovog rada je razvoj Android aplikacije koja bi pomagala globalnom procesu učenja prvenstveno omogućavajući učenje jezika u pokretu, bilo gde, bilo kada.

Ovo, uvodno poglavlje uopšteno govori o razlozima koji vode nastanku aplikacije obrađene u ovom radu. Govori o potrebama modernog društva i tome kako se moderne tehnologije razvijaju da zadovolje ove potrebe i, konkretno, motivaciji za razvoj JapGO aplikacije i ciljnoj grupi kojoj je namenjena, odnosno čije potrebe treba da zadovolji.

Drugo poglavlje predstavlja pregled literature u polju mobilnog obrazovanja uopšte. U ovom poglavlju date su definicije mobilnog obrazovanja, specifičnosti mobilnog okruženja i dalje perspektive za razvoj kao i pregled postojećih mobilnih platformi i trenutni nivo njihovih sposobnosti i tehnologija.

Treće poglavlje se bavi konkretnijom temom, primenom mobilnih tehnologija u oblasti učenja jezika (Mobile Assisted Language Learning, MALL) i daje pregled radova u oblasti, definiše pojam MALL-a, i daje pregled različitih tehnologija mobilnih uređaja i načine na koje se mogu primeniti i kako su primenjene u drugim radovima. Dat je i pregled radova u oblasti koji se bave razvojem konkretnih aplikacija i koriste jednu ili više nabrojanih tehnologija.

Četvrto poglavlje se bavi konceptom objekata učenja, pregledom literature u dotičnoj oblasti i radovima koji primenjuju objekte učenja u klasičnom i mobilnom okruženju.

Peto poglavlje opisuje Android platformu koja je korišćena pri razvoju JapGO aplikacije, daje pregled njene istorije i razvoja, arhitekture sistema i njegovih komponenata.

Sledeće, šesto poglavlje, daje kratak pregled nekih karakteristika japanskog jezika koje su figurisale u kasnijem razvoju aplikacije i njenih funkcionalnosti.

Sedmo poglavlje daje pregled postojećih komercijalnih aplikacija za učenje japanskog jezika, svrstanim u tri kategorije – desktop aplikacije, onlajn aplikacije i mobilne aplikacije.

Osmo poglavlje se bavi projektovanjem i razvojem aplikacije. Opisan je i projektovan model objekata učenja i model podataka koji će biti korišćen u aplikaciji.

U devetom poglavlju su opisani arhitektura Android i komplementarne veb aplikacije, elementi, resursi i način funkcionisanja pojedinih mehanizama.

Deseto poglavlje daje pregled i detaljan opis pojedinačnih funkcionalnosti aplikacije.

U jedanaestom poglavlju je dat zaključak koji uključuje kratak pregled onoga što je urađeno i eventualnih mogućnosti za budući razvoj.

## **2. MOBILNO OBRAZOVANJE**

Sa razvojem informaciono-komunikacionih tehnologija javlja se mnoštvo mogućnosti za njihovu primenu u oblasti obrazovanja. Sam proces učenja se zasniva na pravovremenom dostavljanju i prezentovanju informacija onima koji uče. Za dostavu informacija mogu se upotrebiti telekomunikacione tehnologije, prvenstveno računarske mreže i mreža svih mreža, Internet; za prezentaciju informacija, u ovom slučaju nekakvih edukativnih sadržaja, već postoji ceo niz multimedijalnih i interaktivnih rešenja.

Pored ovoga, u procesu učenja je važna i evaulacija napretka učenika koja se obavlja kombinacijom interakcije sa predavačem (ukoliko učestvuje u procesu), kontinualnim vođenjem evidencije savladavanja jedinica sadržaja i direktnim testiranjem i proverom znanja. Moderne tehnologije lako zadovoljavaju i ove zahteve – dostupni su razni oblici dvosmerne i višesmerne komunikacije gde udaljenost obično ne igra ulogu, dok logika u uređajima i programima za učenje može da preuzme deo odgovornosti ili kompletну ulogu predavača i daleko detaljnije da prati i reaguje na njegov napredak.

U skladu sa time, očigledno je da praktično bilo koji proces učenja, bilo formalni ili neformalni, može da ima koristi od primene informaciono-komunikacionih tehnologija. Dok se formalni proces učenja često (ali ne uvek) odvija na jednom, predoređenom mestu, neformalno učenje se odvija na mestima koja nisu predviđena za tu svrhu ili čak u pokretu. Primer su studenti koji ponavljaju materiju za ispit na putu autobusom do fakulteta, lekari koji osvežavaju svoje medicinsko znanje tokom vizita, studenti jezika koji vežbaju jezičke veštine dok putuju u inostranstvo. Svi ovi primeri formalnog i neformalnog učenja se odvijaju dok su ljudi u pokretu.

Oblast koja se bavi primenom modernih tehnologija kako bi se olakšalo/omogućilo učenje u pokretu se naziva M-obrazovanje (M-učenje, mobilno učenje). M-obrazovanje se može posmatrati kao jedan aspekt E-obrazovanja koji je fokusiran na učenje u različitim kontekstima uz pomoć mobilnih uređaja.

## **2.1. Definicije**

MOBIlearn projekat je usvojio sledeću definiciju M-obrazovanja: „Bilo kakav vid učenja koji se odvija kada učenik nije na fiksnoj, unapred definisanoj lokaciji, ili učenje koje se odvija kada učenik koristi prilike za učenje stvorene prisustvom mobilnih tehnologija“ [1].

Definicija koju je usvojio Kaleidoscope obuhvata i obrazovno okruženje: „Mobilno učenje je učenje kroz niz konteksta, pri čemu je fokus na mobilnosti učenika u interakciji sa prenosivim tehnologijama“ [2].

Projekat WELCOME na univerzitetu u Regensburgu definiše mobilno obrazovanje kao: „Bilo koji servis ili sredstvo koji omogućavaju učeniku opšte elektronske informacije i obrazovni sadržaj koji pomažu u sticanju znanja bez obzira na lokaciju i vreme“ [3].

Ovom definicijom je pokriven niz aspekata. Prvo, bilo koji servis koji se uklapa u ovu definiciju može biti deo mobilnog obrazovanja. Uključene su i dopunske usluge kao što je obezbeđivanje uređaja za studente i izgradnja bežične infrastrukture. Dalje, definicija se fokusira na informaciju ili sadržaj u elektronskom obliku. Ovime se isključuju situacije poput učenja u kafeu iz štampanih materijala, gde je mobilnost očigledno prisutna, ali nije reč o mobilnom obrazovanju. Nije neophodno da servis obezbeđuje obrazovni sadržaj – usluge koje redukuju studentima potrebu za sekundarnim informacijama (poput rokova za predaju radova, termina dodatnih predavanja, rasporedi po studentskim domovima, čak i raspored javnog prevoza) pomažu u obrazovanju redukujući vreme i trud koji se obično troši da bi se stekle potrebne informacije.

## **2.2. Perspektive**

Osnovni pogledi na mobilno učenje polaze iz četiri perspektive:

- Tehnocentrične – mobilno učenje se sagledava kao učenje putem mobilnih uređaja.
- U relaciji sa e-obrazovanjem [4] – mobilno obrazovanje se sagledava kao proširenje e-obrazovanja. Ove definicije su često sveobuhvatne i ne pomažu u opisu jedinstvene prirode mobilnog učenja. Prema [5] tehnocentrične i definicije zasnovane na e-učenju samo pokušavaju da smeste „mobilno učenje negde u spektru prenosivosti u okviru e-učenja“.

- Proširivanje formalnog obrazovanja - U literaturi o mobilnom učenju, formalno obrazovanje se često karakteriše kao nastava licem-u-lice, ili stereotipno kao predavanja. Ova perspektiva se ne može usvojiti kao sasvim tačna. Oblici obrazovanja na daljinu postoje preko 100 godina. Postavlja se pitanje mesta mobilnog obrazovanja u odnosu na sve oblike tradicionalnog obrazovanja, ne samo na ono koje se odvija u učionicama.
- Fokusirane na učenika – U radu [1] autori predlažu fokus na mobilnost učenika i razvoj teorije učenja u mobilnoj eri zasnovanoj na teoriji aktivnosti. Njihov fokus je na mobilnom učenju kao komunikaciji u kontekstu.

Prema Dezmondu Kiganu postoji kontinuum u funkcionalnosti uređaja korišćenih za e-obrazovanje i m-obrazovanje i u njemu nema jasne granice između e- obrazovanja i m-obrazovanja. On smatra da je za pravo određenje mobilnog učenja odlučujuća mobilnost, a ne funkcionalnost. Mobilno učenje kao pojam treba ograničiti na uređaje koji „staju u damsку tašnicu ili u džentlmenski džep“ [6].

Savill-Smith i Kent su u svom radu [7] identifikovali pet osnovnih razloga za jačanje popularnosti mobilnih uređaja (palmtop računara) u učenju:

- relativno jeftini uređaji
- mogućnost rada bilo gde, bilo kada
- poboljšavaju informacionu pismenost
- pomažu pri kolaborativnom učenju
- podržavaju nezavisno učenje

Dodatno navedeni razlozi su: pomažu u motivisanju učenika, poboljšavaju organizacione sposobnosti, podstiču osećaj odgovornosti, služe kao referentni materijali, prate napredak učenika i mogu da se koriste za procenu njihovog znanja. Iako je u radu vršena studija sa palmtop računarima, svi izvedeni zaključci su više nego primenjivi na moderne pametne telefone koji po karakteristikama lako nadmašuju ranije palmtop uređaje.

### **2.3. Specifičnosti**

Postoje određene specifičnosti mobilnog obrazovanja u odnosu na druge vidove učenja koje proizilaze iz karakteristika mobilnog okruženja. Šarples, Tejlor i Vavula u radu "Towards a

Theory of Mobile Learning” [8] navode koje je koncepte neophodno uvažiti pri formiranju definicije mobilnog učenja.

Osnovna specifičnost je pretpostavka kontinuiranog kretanja. Učimo kroz prostor usvajajući ideje i izvore saznanja na jednom mestu i primenjujemo ih ili razvijamo na drugom. Učimo kroz vreme revidirajući znanje koje je ranije stečeno u drugačijem kontekstu i, šire kroz ideje i strategije koje smo usvojili u ranim godinama i koje pružaju trajni okvir za učenje. Krećemo se sa teme na temu upravljujući nizom ličnih projekata učenja, umesto da se držimo jednog, jedinstvenog kursa. Dolazimo u kontakt sa tehnologijom i iz njega izlazimo, na pr. kada otklopimo i zaklopimo poklopac mobilnog telefona.

Mobilno učenje se ne izdvaja od postojećih oblika učenja, već se oni sagledavaju pod novim svetлом. Čak i u školi đaci se kreću iz učionice u učionicu i prelaze sa teme na temu.

Naglašavajući mobilnost učenja kao objekat analize možemo bolje da razumemo kako se znanje i veštine prenose kroz različite kontekste i kako da projektujemo tehnologije koje će podržati društvo u kom se učenje sve češće odvija u kratkim intervalima između svakodnevnih aktivnosti.

Teorija mobilnog učenja mora da obuhvati značajan obim učenja koje se odvija izvan klasičnih lokacija formalnog obrazovanja učenje koje se dešava izvan učionica i amfiteatara dok ljudi započinju i strukturiraju svoje aktivnosti kako bi omogućili obrazovne procese i ishode. Ključno je razumevanje načina na koje ljudi stupaju u interakciju sa svojim okruženjem kako bi stvorili privremeno okruženje za učenje.

Mora se zasnivati na savremenim shvatanjima o aktivnostima koje omogućavaju uspešno učenje. Ono je:

- usmereno na učenika;
- izgrađuje znanja i veštine učenika omogućavajući im da zaključuju na osnovu sopstvenog iskustva
- usmereno prema znanju;
- kurikulum se gradi iz na postojećoj osnovi proverenog znanja, i efikasno prenosi uz inventivno korišćenje koncepata i metoda
- usmereno prema veštinama procenjivanja sposobnosti učenika
- usmereno prema zajednici, gde uspešniji učenici pomažu sporijim i rade na međusobnom proširenju znanja.

Konačno, teorija mobilnog učenja mora da uzme u obzir sveprisutnost ličnih i deljenih tehnologija.

Mobilnost se posmatra:

- u fizičkom prostoru – kretanja ljudi
- tehnologije – prenosivi uređaji i izvori koji se mogu nositi sa sobom, kao i promene aktivnog uređaja
- u konceptualnom prostoru – neprestana smena obrazovnih tema i skokovi pažnje mešu njima
- mobilnost u društvenom prostoru – studenti čine različite društvene grupe, uključujući porodicu, radno mesto, učionicu.

Mobilno učenje je nemoguće precizno definisati [8]. Umesto toga opisane su osnovne karakteristike mobilnog učenja. Ono:

- omogućava građenje znanja u različitim kontekstima
- omogućava učenicima da konstruišu značenja
- mobilne tehnologije često menjaju obrasce aktivnosti učenja i rada
- kontekst mobilnog učenja je više od samo vremena i prostora.

Džounsova navodi šest razloga zašto mobilno učenje može motivisati [9]:

- kontrola (nad ciljevima)
- posedovanje
- zabavnost
- komunikacija
- učenje u kontekstu
- kontinuitet između konteksta.

## 2.4. Mobilne platforme i tehnologije

Tehnologije koje se primenjuju za razvoj mobilnih aplikacija treba posmatrati sa više aspekata: servisi, uređaji, platforme, jezici za razvoj aplikacija, način bežične komunikacije, tipovi prikaza, tipovi multimedijalnih sadržaja. Više detalja o ovim tehnologijama se može pronaći u [10][11].

Istraživanja u pogledu upotrebljivosti mobilnih uređaja u procesu učenja jezika započinju sa javljanjem prvih prenosnih uređaja kao što su prenosni kasetofoni, kasnije MP3 plejeri, PDA uređaji i mobilni telefoni. Upotrebljivost mobilnih telefona i PDA uređaja je u prošlosti bila ograničena njihovim tehničkim karakteristikama, prvenstveno ekranima manje rezolucije koji su predstavljali problem za prikaz veće količine teksta, slike i video materijala ali i nezgrapnim metodama unosa, ograničenim memorijskim i obradnim kapacitetima i sporadičnom konekcijom sa internetom [12]. Veliki korak u rešavanju navedenih problema je načinjen javljanjem iPhone, Android i drugih modernih platformi koje karakterišu uređaji većih ekrana, inovativne metode interakcije kao što su virtualne tastature, poboljšane karakteristike i dodatne hardverske komponente. Na polju konektivnosti 3G, 4G i Wi-Fi za vezu sa internetom i USB i Bluetooth za vezu sa drugim uređajima postaju standard, a ugrađeni GPS, kamere, senzori, akcelerometri i kompasi postaju vredni izvori kontekstnih informacija [13].

Ovoliko usložnjavanje mobilnih uređaja je za posledicu imalo postojanje nekoliko nekompatibilnih operativnih sistema na tržištu. Razvoj aplikacija sada zahteva ograničavanje na jednu od platformi ili dodatan skup veština i dodatno ulaganje rada u razvoj varijanti za druge platforme. Pored poznavanja programskog jezika koji se koristi za pisanje aplikacija na nekoj platformi, potrebno je poznavati i specifičan API sistema, odnosno interfejse ka ugrađenim funkcionalnostima uređaja, kao što su pristup kameri, mehanizmu za slanje poruka, imeniku itd. Konkretno, dva vodeća operativna sistema na tržištu, Android i iOS koriste redom Javu i Objective-C. Alternativa ovom pristupu je korišćenje standardnih, široko rasprostranjenih web tehnologija - HTML-a, CSS-a i JavaScript-a za razvoj web aplikacija koje će se izvršavati u okviru web pretraživača na mobilnim uređajima. Ovi pretraživači na savremenim uređajima ne zahtevaju posebne, uprošćene varijante web stranica i, uz korišćenje AJAX tehnologije, kvalitet interakcije sa korisnikom se diže na nivo klasičnih aplikacija. Značajna podrška za ovaj pristup postoji kod iOS-a u vidu alata za izradu web aplikacija za iPhone (Dashcode, iWebKit).

Čak i za klasične aplikacije, korišćenje interneta je nezaobilazno. Moderne platforme omogućavaju korišćenje HTTP protokola, web servisa ili čak i direktno ostvarivanje TCP konekcija. Sa time na raspolaganju, aplikacije za učenje mogu da se projektuju da koriste onlajn resurse radi štednje memorijskog prostora koji može biti nedovoljan za smeštanje veće količine edukativnih audio/video materijala. Internet nudi i niz servisa koji se mogu upotrebiti čak i kada nisu direktno namenjeni učenju; u kontekstu učenja jezika primeri bi bili prevodilački servisi kao Google Translate, RSS feed-ovi na jeziku koji se uči, jezički alati (npr.

web aplikacija za konverziju čirilice u latinicu) i slično. Na kraju, proces učenja se može znatno obogatiti uključivanjem interakcija sa učiteljima i sa drugim učenicima na razne načine - preko wiki web aplikacija i specifičnih foruma [14][15] pa do kompletnih rešenja za interakciju licem-u-lice u sajber prostoru [16].

## **2.5. Android platforma**

Android je mobilni operativni sistem prvo bitno razvijen od strane Android kompanije, koju je kupio Gugl 2005. godine. Nakon kupovine, Gugl je zatražio stvaranje Open Handset Alliance (OHA) udruženja, koja je sada angažovano u pružanju podrške i daljem razvoju sistema. Android je zasnovan na izmenjenoj verziji Linux jezgra. Po broju prodanih pametnih telefona, zasnovanih na Android OS u SAD, u drugom kvartalu 2010. godine, Android se nalazi na prvom mestu sa udelom od 33 procenata.

Android ima veliku zajednicu programera za pisanje aplikativnih programa koji proširuju funkcionalnost uređaja. Trenutno postoji preko 450.000 aplikacija dostupnih za Android, a po nekim procenama i više, što ga čini drugim najpopularnijim mobilnim razvojnim okruženjem. Programeri pišu kontrolisani kod u Java jeziku, kontrolišući uređaj preko Gugl razvojne Java biblioteke.

Početak Android distribucije najavljen je 5. novembra 2007. godine zajedno sa osnivanjem Open Handset Alliance, udruženja 78 kompanija posvećene unapređenju otvorenih standarda za mobilne uređaje. Gugl je veći deo Android koda objavio pod Apache licencom, licencom slobodnog softvera i licencom otvorenog koda.

Android OS se sastoji od 12 miliona linija koda, uključujući 3 miliona linija XML koda, 2,8 miliona linija C koda, 2,1 miliona linija Java koda i 1,75 miliona linija C++ koda..

Android predstavlja softverski stek namenjen mobilnim telefonima koji u sebi uključuje operativni sistem, posrednički softver kao i ključne aplikacije za normalno funkcionisanje. Android SDK obezbeđuje alatke i API koji su neophodni za dalji razvoj aplikacija na Android platformi koristeći JAVA programski jezik.

Shodno svojoj politici OHA ističe sledeće karakteristike Android programskog okruženja:

- **otvorenost** - programeru omogućava potpunu slobodu u razvoju novih i već postojećih aplikacija, a proizvođaču uređaja slobodno korišćenje i prilagođavanje platforme bez plaćanja autorskih prava;
- **sve aplikacije su ravnopravne** - što znači da ne postoji razlika između osnovnih jezgarnih aplikacija uređaja i dodatnih aplikacija. Svim aplikacijama omogućen je ravnopravni pristup resursima pokretnog uređaja što daje mogućnost prilagođavanja uređaja specifičnim potrebama individualnog korisnika;
- **automatsko upravljanje životnim ciklusom aplikacije** - omogućava nadzor pokretanja i izvršavanja aplikacija na sistemskom nivou optimizovanim korišćenjem memorije i snage uređaja. Krajnji korisnik više ne brine o gašenju određenih aplikacija pre pokretanja drugih;
- **uklanjanje granica "klasičnih" aplikacija** - mogućnost razvoja novih i inovativnih aplikacija zasnovanih na međusobnoj kolaboraciji tehnologija;
- **brz i jednostavan razvoj aplikacija** - omogućen je bogatom bazom korisnih programske biblioteke (eng. libraries) i alata za izradu aplikacija;
- **visokokvalitetni grafički prikaz i zvuk** - podržana 2D vektorska i 3D OpenGL (engl. Open Graphics Library) grafika, te ugrađeni kodeci svih često korišćenih audio i video formata;
- **kompatibilnost sa većinom sadašnjeg i budućeg hardvera** - uključuje prenosivost Androidovih aplikacija na ARM, k86 i ostale arhitekture, te prilagodljivost sistema ulaznim i izlaznim komponentama.

### 2.5.1. Verzije Android operativnog sistema



#### Android 1.0/1.1

Prva verzija operativnog sistema Android (1.0) predstavljena u Septembru 2008. godine, premijeru doživela na telefonu kompanije HTC (HTC Dream), razvijenom u saradnji sa Guglom. Pošto je Android 1.0 patio od mnogih dečjih bolesti, Gugl je šest meseci kasnije izbacio novu verziju (1.1) sistema koja je ispravila par "kritičnih problema", podršku za glasovnu pretragu, mogućnost plaćenih aplikacija u onlajn prodavnici ("Market"), ispravljene probleme u vezi sa servisom "Gmail" i još nekoliko sitnih prepravki.



#### Android 1.5 ("Cupcake")

Izlazi u Maju 2009. Nova verzija donela je značajna poboljšanja sistema u vidu novih mogućnosti kao što su: snimanje video sadržaja, podrška za stereo blutut slušalice, podesiva tastatura, prepoznavanje glasa, mogućnost

kreiranja sopstvenih aplikacija za početni ekran (Widgets). U ovom periodu se javlja veliki broj novih modela telefona koji su bili pokretani Android operativnim sistemom.



### **Android 1.6 ("Donut")**

Nova verzija, 1.6, je imala zadatak da isprati napredak konkurencije. U 1.6 su uvedena brojna poboljšanja već široko prihvaćenog operativnog sistema.

Najznačajnija poboljšanja u verziji 1.6 su: novi izgled onlajn prodavnice koji je omogućio korisnicima lakši pristup i bržu pretragu željenih aplikacija; kamera i video kamera integrисани u Galeriju, koja je sada omogućavala izbor više datoteka za istovremeno brisanje; podrška za nove tehnologije kao što su: CDMA/EVDO, 802.1x i "text-to-speech"; podrška za navigacione aplikacije; razna poboljšanja po pitanju brzine, pretrage grafičkog interfejsa, itd. Sa druge strane, sa ovom verzijom do izražaja dolazi problem koji će pratiti Android operativni sistem sve do današnjih verzija – pošto je za razvoj sistema korišćen HTC Dream, ažuriranje je bilo otežano drugim proizvođačima pa su se javljali novi modeli telefona sa starim verzijama sistema.



### **Android 2.0/2.1 ("Eclair")**

Vremenom HTC Dream polako zaostaje za konkurencijom po pitanju tehničkih mogućnosti, te Gugl u saradnji sa Motorolom razvija novi telefon (Motorola Droid) koji će služiti za razvoj operativnog sistema i aplikacija i kojeg je pokretala nova verzija Androida, 2.0. Android 2.0 je smatrana za prvu veliku nadogradnju sistema, koja je donela razne značajne promene, od kojih su najbitnije: podrška za "Microsoft Exchange"; podrška za "Facebook"; poboljšanja veb pretraživača; aplikacija za pretraživanje mapa ("Google Maps") i pregled satelitskih snimaka; novi Jutjub vidžet za brzo postavljanje svojih snimaka na servere Jutjuba; novi, lakši način upravljanja itd.



I pored saradnje sa Motorolom na njihovom telefonu, Gugl paralelno razvija i svoj telefon, Google Nexus One, koristeći najmodernije tehnologije i novu, 2.1 verziju Android sistema. Ova verzija sistema je masovno prihvaćena od svih proizvođača telefona koji su bili uključeni u Android projekat i predstavlja prvu sledeću "masovnu" verziju nakon verzije 1.5. Poboljšanja u ovoj verziji su: poboljšana ukupna brzina sistema; podrška za nove veličine ekrana i rezolucije istih; novi grafički interfejs web pretraživača i podrška za HTML5; podrška za blic za kameru; digitalni zoom; podrška za registrovanje pokreta više prstiju od jednom, tzv. "Multi-touch"; poboljšana tastatura; podrška za blutut verzije 2.1 (blutut transfer datoteka); "Live Wallpapers"...



### **Android 2.2 ("Froyo")**

Posle dužeg perioda, u maju 2010. godine Gugl predstavlja verziju 2.2 koja uvodi najveću listu poboljšanja do tada, sa najznačajnijim unapređenjem - podrškom za fleš animacije implementiranom u saradnji sa kompanijom Adobe. Druga uvedena poboljšanja su: veliko povećanje ukupne brzine sistema, optimizacija korišćenja memorije; dodatno povećanje procesiranja aplikacija koja je ostvarena primenom "JIT" ("Just in Time") skripte; integracija "Chrome" V8 JavaScript u veb pretraživač; mogućnost deljenja internet konekcije sa drugim uređajima putem USB ili WiFi konekcije ("Tethering"); opcija za brzo isključivanje mobilne mreže; nova onlajn prodavnica sa mogućnošću automatske nadogradnje aplikacija; podrška za "upload" preko veb pretraživača; podrška za instaliranje aplikacija na proširivu memoriju (microSD); podrška za Adobe Flash 10.1; podrška za snimanje i prikazivanje videa u visokoj rezoluciji 720p...



### **Android 2.3 ("Gingerbread")**

6.Decembra 2010. godine izbacuje novu verziju (2.3) i novi telefon, razvijen u saradnji sa Samsungom (Google Nexus S). Brojna poboljšanja uključuju: redizajniran GUI; podršku za ekrane rezolucije veće od 800x480; VOIP podršku; podršku za WebM/VP8 video; podršku za NFC; unapređena "copy/paste" funkcija; nova "multitouch" tastatura; audio, video i grafička unapređenja namenjena razvoju video igara; podršku za nove senzore poput žiroskopa; poboljšanu kontrolu energije i aplikacija; podršku za više kamere; zamenu "YAFFS" "EXT4" sistemom datoteka...



### **Android 3.0 ("Honeycomb")**

Promene koje donosi novi operativni sistem se odnose pre svega na prilagođenost većim ekranima i sanžnijim konfiguracijama tablet uređaja: novi grafički interfejs namenjen tablet uređajima; radna površina u tri dimenzije (3D) sa novim, redizajniranim vidžetima; novi način simultanog izvršavanja aplikacija ("Multitasking"); veb pretraživač napravljen posebno za potrebe tablet uređaja; podrška za video čakanje ("Chat") koristeći Gugl aplikaciju "Google Chat"; interfejs koji se izvršava na nivou grafičkog procesora ("Hardware Acceleration"); podrška za proceore sa više jezgara... Prvi uređaj koji koristi novu verziju operativnog sistema Android, 3.0 "Honeycomb" je Motorola Xoom. Ovaj uređaj poseduje procesor sa dva jezgra, "nVidia Tegra2" proceor koji podržava puštanje videa u pravoj HD rezoluciji 1080p.



### **Android 4.0.\* ("Ice Cream Sandwich")**

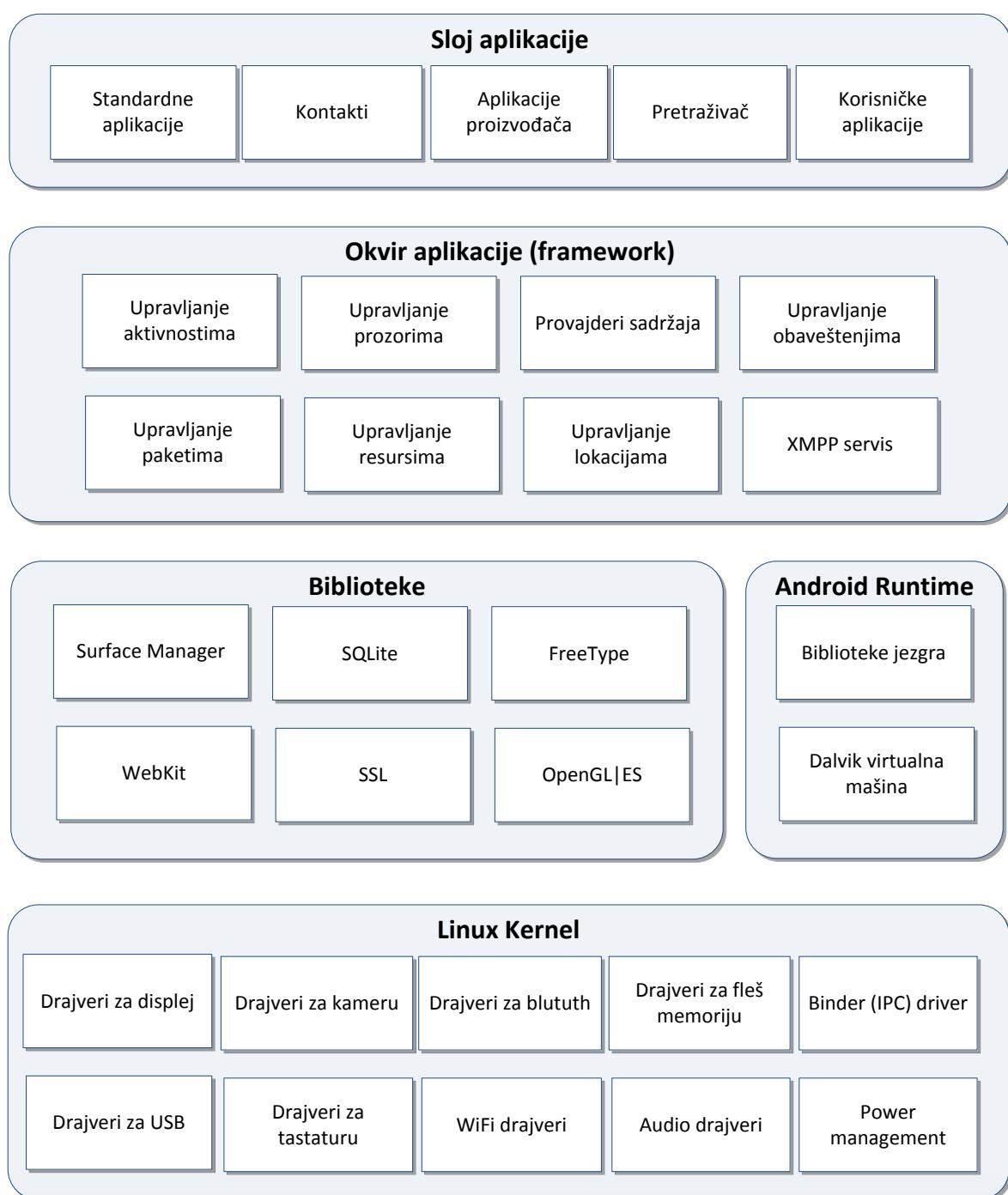
SDK za ovu verziju androida je javno objavljen 19. oktobra 2011. Cilj ove verzije je bio objedinjavanje verzija za telefone i za tablete. Pored toga, u ovoj verziji su se našla mnoga poboljšanja i optimizacije postojećih mehanizama (kamera, interfejs, stabilnost, ispravke bagova) i nove funkcionalnosti, između ostalog: virtualni dugmići koji menjaju sve fizičke dugmiće, integrisano pravljenje slike trenutnog ekrana, bolja korekcija grešaka pri unosu, startovanje aplikacija preko zaključanog ekrana, otključavanje telefona prepoznavanjem lica korisnika, Android Beam – sistem za NFC, hardverska akceleracija interfejsa, snimanje u 1080p rezoluciji itd.

Neke osnovne funkcionalnosti Android OS-a su:

- Prikaz na uređajima – OS prilagođava izgled različitim tipovima uređaja i kompatibilan je sa 2D, 3D grafičkim bibliotekama
- Skladištenje podataka – SQL lite relaciona baza podataka za upravljanje podacima
- Konektivnost – Android podržava konektivnost sa svim standardnim tehnologijama: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE i WiMAX.
- Razmena poruka – SMS i MMS
- Web browser – WebKit browser, zasnovan na Chrome-u
- Java support – Android aplikacije se pišu u Java programskom jeziku, Dalvik virtuelna mašina
- Podrška za multimedijalne podatke - WebM, H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, MP3, MIDI, OGG, WAV, JPEG, PNG, GIF, BMP.
- Streaming media support - RTP/RTSP streaming (3GPP PSS, ISMA), HTML download (HTML5 <video> tag), Adobe Flash Streaming (RTMP), HTTP Dynamic Streaming
- Dodatna hardverska podrška - kamera, touchscreen, GPS, brzinometar, žiroskop, magnetometar, senzori udaljenosti i pritiska, termometri
- Okruženje za razvoj – Android Development Plugin
- Tržiste – Google play (bivši Android Market) je online katalog aplikacija koje se mogu preuzeti i instalirati na Android uređaje, bez potrebe za računarom.
- Multi-touch – mogućnost uređaja da prepozna više istovremenih dodira
- Bluetooth
- Video poziv
- Multitasking – multitasking aplikacija
- Funkcionalnosti zasnovane na glasu – Pozivanje, pretraga, navigacija
- Tethering – Android telefon može da se iskoristi kao wireless hotspot

## 2.5.2. Arhitektura Android sistema

Arhitektura Android sistema bazira se na Linux-u 2,6 jezgru (eng. kernel) koja se koristi kao sloj apstrakcije hardvera (HAL, eng. Hardvare Abstraction Laier). Razlog za korišćenje jezgra operativnog sistema Linux je dokazana pogonska podrška (eng. driver model), mogućnost upravljanja memorijom i procesima, sigurnosni model, mrežni sistem, te dokazana robusnost, konstantni razvoj i unapređivanje sistema. Slika 1.1 prikazuje arhitekturu Android sistema.



Slika 1. Arhitektura Android platforme

### **2.5.2.1. Linux jezgro**

Najvažniji elementi jezgra su:

- pogonski program za međuprocesnu komunikaciju između aplikacija i usluga (eng. service) (eng. Binder (IPC) Driver)
- upravljanje napajanjem (eng. Power Management) - komponenta Androida temeljena na standardnom Linuxovom sistemu upravljanja napajanjem.

### **2.5.2.2. Izvorne programske biblioteke**

Izvorne programske biblioteke (eng. native libraries) pisane su u programskim jezicima C i C++ i čine sledeći sloj u arhitekturi sistema. Neke od značajnijih su:

- UI - program za upravljanje grafičkim interfejsom (eng. Surface Manager) - biblioteka odgovorna za pravilno iscrtavanje različitih aplikacionih komponenti u vremenu i prostoru;
- OpenGL ES (eng. OpenGL for Embedded Systems) - biblioteke koje se koriste za hardversku 3D akceleraciju (ukoliko je podržana) ili za 3D rasterizaciju;
- SGL (eng. Scalable Graphics Library) - predstavlja 2D biblioteke na kojima je zasnovana većina aplikacija. Spomenimo još da se 2D i 3D elementi mogu kombinovano prikazivati u jednom korisničkom interfejsu;
- Media Framework - grupa kodeka za snimanje i reprodukciju audio formata, video formata i nepomičnih slika. Omogućena je od strane PacketVideo;
- FreeType - biblioteka koja služi za vektorsku rasterizaciju oblika pisma (eng. font);
- SSL (eng. Secure Socket Layer) - omogućava rezervnu komunikaciju preko Interneta;
- SQLite - programska biblioteka koja implementira bazu podataka (eng. database engine);
- WebKit - Jezgro pretraživača koji podržava JavaScript i ostale standarde na malom uređaju;
- System C library - implementacija standardne C-ove sistemske biblioteke (libc) izvedene iz operativnog sistema BSD.

### **2.5.2.3. Radno okruženje - Android Runtime**

Idući sloj u arhitekturi Androida je radno okruženje (eng. Android runtime) kojeg čine virtuelni mašina Dalvik (DVM, eng. Dalvik Virtual Machine) i jezgrene biblioteke (eng. core librari).

DVM je registrski bazirana virtuelna mašina, dok je klasična Javina virtuelna mašina (JVM, eng. Java Virtual Machine) baziran na steku. Bazne biblioteke pisane su u programskom jeziku Java i predstavljaju sve esencijalne klase kao što su klase za manipulaciju kolekcijama, klase za komunikaciju sa okolinom i slično. Bitna novost je i to što se Androidom jezgrene biblioteke razlikuju od biblioteka u Java Standard Edition (J2SE) i Java 2 Micro Edition (J2ME).

Umesto upotrebe standardnog Java 2 Micro Edition (J2ME) kao mehanizma za pokretanje javinih aplikacija na mobilnim uređajima, Google je razvio sopstvenu virtuelnu mašinu za Android. DVM je najverovatnije razvijen kako bi se zaobišla problematika s dozvolama oko korišćenja Sunovog J2ME. Svaki mobilni uređaj koji u sebi sadrži J2ME mora kod Suna licencirati bilo kakvu promenu izvornog koda J2ME-a.

Virtuelna mašina Dalvik dobila je ime po istoimenom Islandskom gradu u kojem su živeli preci glavnog developera DVM-a Dan Bornstein.

Osnovna razlika između Sun Java virtuelnih mašina i DVM-a je u tome što su prvo spomenuti virtualni uredjaju bazirani na steku, dok je DVM registrski baziran virtuelni mašina. Međukod (eng. bytecode) Dalvik virtuelne mašine transformiše se pomoću alata dk (koji je sastavni deo Android SDK-a) iz Javinih klasnih datoteka (eng. Java class file) prevedenih Javnim revodiocem u novu klasu \*.dex (eng. Dalvik Executable) formata. Međukod koji izvršava DVM nije Javin međukod, nego upravo spomenuti .dex oblik. Transformacija formata omogućava bolju prilagođenost za rad na manjim procesorima boljim iskorišćavanjem raspoložive memorije i procesorske snage. Rezultat svega je mogućnost višestrukog instanciranja same virtuelne mašine što znači da se svaka Android aplikacija pokreće kao zasebni proces, s vlastitom instancom virtuelne mašine Dalvik.

#### **2.5.2.4. Application Framework**

Sloj aplikativnih okvira (eng. Application Framework) napisan je u programskom jeziku Java i sadrži proširiv skup programskih komponenti kojeg koriste sve aplikacije uređaja. Framework podržava mnogobrojne open source jezike kao što su openssl, sqlite i libc. Takođe podržava i jezik Android jezgra. Sa gledišta sigurnosti, framework se bazira na UNIX file system ovlašćenjima koja osiguravaju da aplikacije poseduju samo one mogućnosti koje im je vlasnik telefona dao pri instalaciji aplikacije.

Neki od važnijih elemenata su:

- upravljanje aktivnostima (eng. Activiti Manager) - upravljanje životnim ciklusom aplikacije,
- upravljanje programskim paketima (eng. Package Manager) - sadrži informaciju o aplikacijama instaliranim na sistemu
- upravljanje prozorima (eng. Window Manager) - upravljanje aplikacionim prozorima,
- upravljanje pozivima (eng. Telephone Manager) - sadrži API-je koji se koriste pri izradi aplikacija za upravljanje pozivima,
- pružaoci sadržaja (eng. Content Providers) - omogućavaju zajedničko korišćenje podataka od strane više aplikacija,
- upravljanje resursima (eng. Resource Manager) - služi za skladištenje delova aplikacije koji nisu kod (npr. slike),
- sistem grafičkog prikaza (eng. View Sistem) - sadrži bazu gotovih grafičkih prikaza i alata (eng. vidget),
- upravljanje lokacijski zasnovanim uslugama (eng. Location Manager) i
- upravljanje notifikacijama (eng. Notification Manager) - upravljanje notifikacijama i događajima (npr. dospeće poruke, nadolazeći sastanak).

Aplikativni sloj je poslednji sloj u arhitekturi sistema Android i čine ga korisničke aplikacije uređaja. Predstavlja sloj vidljiv krajnjem korisniku. Uključuju neke od standardnih sistemskih aplikacija kao što su Veb pretraživač, lista kontakata, telefon, itd.

### **2.5.2.5. SQLite**

SQLite je veoma mali (oko 500kb) sistem za upravljanje relacionim bazama podataka, koji je integriran u Android. Zasniva se na pozivima funkcija i jedinstvenih fajlova, gde se čuvaju sve definicije, tabele i podaci. Ovaj jednostavan dizajn je više nego pogodan za platformu kao što je Android. Postoji veliki broj karakteristika koje zavise od hardvera, kao što su velika količina multimedijalnih fajlova, podrška konekcija, GPS, unapređena podrška za kameru i jednostavna GSM telefonija.

### **2.5.2.6. Fajl AndroidManifest.xml**

Svaka aplikacija mora da poseduje svoj AndroidManifest.xml fajl u korenom folderu projekta. Naziv manifest fajla mora da bude AndroidManifest.xml. Manifest sadrži neophodne

informacije o aplikaciji i o Android sistemu. Ove informacije sistem mora da poseduje pre početka izvršavanja aplikacije.

Neke od funkcija manifest fajla su:

- Sadrži naziv Java paketa u okviru kojeg je napisan izvorni kod aplikacije. Naziv paketa predstavlja jedinstveni identifikator aplikacije.
- Opisuje komponente aplikacije – aktivnosti, servise, primaocce poruka i provajdere sadržaja. Dakle, manifest fajl sadrži nazive klasa u kojima su opisane komponente aplikacije. Na primer, aplikacija se neće startovati ukoliko u manifest fajlu nije navedena njena početna aktivnost, čak ni u slučaju da fajl sa Java izvornim kodom te aktivnosti postoji.
- Određuje koji procesi će biti domaćini komponentama aplikacije.
- Definiše koje privilegije (permissions) aplikacija mora da poseduje da bi mogla nesmetano da funkcioniše.
- Takođe definiše koje privilegije moraju druge aplikacije da poseduju da bi mogle da koriste komponente aplikacije.
- Definiše minimalnu verziju Android API-a koji zahteva aplikacija.
- Definiše listu biblioteka koje aplikacija koristi

### **2.5.3. Životni ciklus aktivnosti**

Aktivnost (Activity) predstavlja komponentu aplikacije koja se uglavnom može poistovetiti s jednim konkretnim prozorom aplikacije u kojem je korisnik u mogućnosti da izvrši određenu radnju. Na primer, aktivnost služi da bi korisniku mobilnog telefona omogućila pozivanje određenog broja, slikanje fotografije korišćenjem ugrađene kamere, slanje e-mail poruke, pregled mape i dr. Aplikacija može da sadrži jednu ili više definisanih aktivnosti, pri čemu je jedna od aktivnosti uvek definisana kao primarna aktivnost. Prelaz između aktivnosti odvija se tako što aktuelna aktivnost poziva novu. Iako više aktivnosti čini jedan kompaktan korisnički interfejs, treba imati na umu da su one međusobno nezavisne. Svaka aktivnost implementira se kao zasebna klasa koja nasleđuje klasu Activity, pa je sama odgovorna za čuvanje svog stanja u životnom ciklusu aplikacije.

Klasa aktivnosti je glavni deo svake Android aplikacije. Programer veći deo vremena provodi na definisanju i implementaciji aktivnosti za svaki ekran u aplikaciji.

Svakoj Android aplikaciji mora biti dodeljena aktivnost unutar Android manifest fajla. Druge klase aktivnosti mogu biti označene u manifest fajlu da se pokrenu pri određenim

okolnostima. Ovim sekundarnim ulaznim tačkama se upravlja pomoću Android manifest fajla sa određenim filterima.

Aplikacije mogu biti prekinute zbog raznih događaja višeg prioriteta, kao što je telefonski poziv. U jednom trenutku može postojati samo jedna aktivna aplikacija. Android aplikacije su odgovorne za upravljanje njihovim stanjem, kao i njihovom memorijom, resursima i podacima. Android operativni sistem može isključiti aktivnost koja je na pauzi, stopirana ili uništena kada je memorija slaba. Ovo znači da svaka aktivnost koja nije u prvom planu mora biti isključena. Drugim rečima, Android aplikacija mora održavati stanje i biti spremna da bude prekinuta ili čak isključena u svakom trenutku.

Životnim ciklusom aktivnosti jedne Android aplikacije upravlja se implementacijom odgovarajućih metoda. Svaka aktivnost ima tri osnovna stanja:

- *Resumed (Running)*
  - Aktivnost je pokrenuta i fokusirana.
- *Paused*
  - Druga aktivnost je pokrenuta i fokusirana (u prvom planu), ali je tekuća aktivnost i dalje pokrenuta i vidljiva. Na primer, druga aktivnost je u prvom planu, dok je tekuća aktivnost vidljiva, ali ne zauzima ceo ekran. Aktivnost čije je stanje *paused* je potpuno „živa“ (objekat klase Activity se nalazi u memoriji, zadržava sva stanja i ostaje u window manager-u), međutim ova aktivnost može da bude „ubijena“ u slučaju niske slobodne memorije.
- *Stopped*
  - Aktivnost čije je stanje *stopped* je i dalje „živa“, ali radi u pozadini. Objekat klase Activity i dalje se nalazi u memoriji, čuva sva stanja, ali se više ne nalazi u okviru window manager-a. Ovu aktivnost više ne vidi korisnik i u slučaju potrebe za memorijom sistem može da je „ubije“.

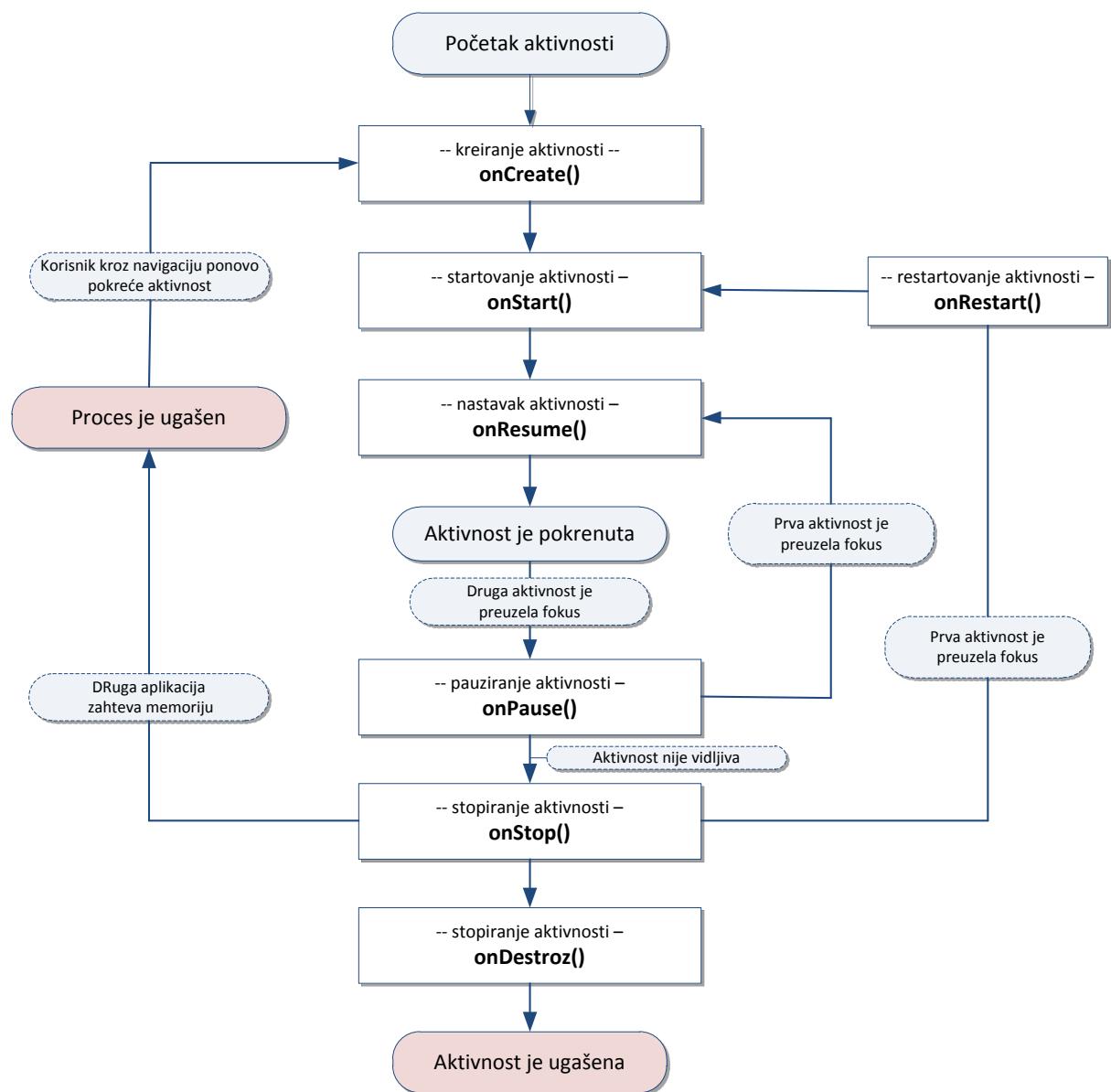
Životni ciklus jedne aktivnosti prikazan je na slici 4.1. Dakle, kao što je napomenuto na početku ovog poglavlja, fazama životnog ciklusa aktivnosti upravljaju metode `onCreate()`, `onRestart()`, `onStart()`, `onResume()`, `onPause()`, `onStop()` i `onDestroy()`.

Životni ciklus aktivnosti započinje metodom `onCreate()`, a završava se zajedno sa metodom `onDestroy()`. U okviru metode `onCreate()`, programer bi trebalo da definiše izgled i globalno stanje aktivnosti, kao što je raspored elemenata korisničkog interfejsa. Implementacijom metode `onDestroy()` trebalo bi oslobođiti korišćene resurse. Na primer, ukoliko vaša aktivnost

koristi pozadinsku nit koja preuzima podatke sa Interneta, nit bi trebalo kreirati u okviru metode `onCreate()`, a zaustaviti je u okviru metode `onDestroy()`.

Aktivnost je vidljiva (*visible*) između poziva metoda `onStart()` i `onStop()`. Dok je aktivnost vidljiva, korisnik je vidi na ekranu i može da vrši interakciju sa njom (npr. da unosi podatke). Metoda `onStop()` se poziva kada se startuje nova aktivnost, a tekuća aktivnost više nije vidljiva.

Aktivnost je fokusirana (u prvom planu) između poziva metoda `onResume()` i `onPause()`. Tada se ta aktivnost nalazi ispred svih ostalih aktivnosti. Aktivnost može često da prelazi iz prvog plana u pozadinu i obrnuto. Na primer, metoda `onPause()` se poziva kada uređaj pređe u stanje mirovanja (*sleep*).



Slika 2. Životni ciklus aktivnosti

U tabeli 1. su prikazane fundamentalne metode za upravljanje životnog ciklusa i njihov opis.

Tabela 1. Metode za upravljanje životnim ciklusom aktivnosti

Metoda	Opis	Može da se „ubije“ posle?	Sledeća metoda
onCreate()	Poziva se kada se aktivnost prvi put kreira. Ovde bi trebalo kreirati poglede (views), povezati (bind) podatke sa listama i sl. Ovoj metodi prosleđuje se objekat tipa Bundle koji sadrži prethodno stanje aktivnosti.	Ne	onStart()
onRestart()	Poziva se posle stopiranja aktivnosti, ali neposredno pre njenog ponovnog startovanja.	Ne	onStart()
onStart()	Poziva se neposredno pre nego što aktivnost postane vidljiva korisniku. Metoda koja se poziva posle ove može da bude onResume() u slučaju da aktivnost prelazi u prvi plan, ili onStop() u slučaju da postaje sakrivena.	Ne	onResume() ili onStop()
onResume()	Poziva se neposredno pre nego što aktivnost započne interakciju sa korisnikom.	Ne	onPause()
onPause()	Poziva se neposredno pre nego što sistem nastavi izvršavanje druge aktivnosti. Ova metoda se obično koristi da sačuva podatke, zaustavi animacije i prekine sa izvršavanjem procesa koji koriste procesorsko vreme. Ova metoda se mora izvršiti što je brže moguće, zato što sledeća aktivnost neće biti nastavljena dok se ova metoda završi. Metoda koja se poziva posle ove može da bude onResume() u slučaju da aktivnost prelazi u prvi plan, ili onStop() u slučaju da postaje sakrivena.	Da	onResume() ili onStop()
onStop()	Poziva se kada korisnik više ne vidi aktivnost. Ovo se može dogoditi kada je u toku uništavanje aktivnosti ili zbog nastavljanja izvršavanja neke druge aktivnosti. Metoda koja se poziva posle ove može da bude onRestart() ukoliko je potrebna ponovna interakcija aktivnosti i korisnika ili onDestroy() u slučaju da aktivnost više nije potrebna.	Da	onRestart() ili onDestroy()
onDestroy()	Poziva se neposredno pre uništavanja aktivnosti. Ovo je poslednji poziv koji će aktivnost da primi. Metoda onDestroy() može biti pozvana kada se završava sa izvršavanjem aktivnosti (neko je pozvao metodu finish()) ili zato što sistem privremeno uništava instancu aktivnosti da bi oslobodio prostor. Pomoću metode isFinishing() može se napraviti razlika između ova dva scenarija.	Da	nema

Kolona „Može da se „ubije” posle?“ označava da li sistem sme da „ubije“ proces koji omogućava pokretanje aktivnosti bilo kada nakon završetka izvršavanja date metode.

#### **2.5.4. Resursi u Android aplikacijama**

Sve android aplikacije sastoje se od dve stvari:

- funkcionalnost – predstavlja kod kako se aplikacija ponaša na neki zahtev. Ovde spadaju svi algoritmi koji pokreću aplikaciju
- resursi ili podaci – obuhvataju tekstove, slike, audio i video fajlove, datoteke, ikone i druge podatke koje koristi aplikacija.

Resursi obuhvataju tekst u obliku stringa, slike I ikone, audio datoteke, video zapise i druge podatke koje koristi aplikacija.

Resursi se dele na dva tipa: resurse aplikacije i resurse sistema. Resurse aplikacije definiše developer unutar fajlova Android projekta i tačno su određeni za neku aplikaciju. Resursi sistema su standardni resursi koje definiše Android platforma i dostupni su svim aplikacijama kroz Android SDK.

Resursi aplikacije se prave i čuvaju unutar Android projekta u /res direktorijumu. Koristeći već definisanu ali fleksibilnu strukturu direktorijuma, resursi su organizovani, definisani i upakovani sa paketom aplikacije. Resursi aplikacije se ne dele sa ostatkom Anroid sistema.

Svi resursi aplikacije se čuvaju u strukturi direktorijuma /res i sastavljeni su u projekat u trenutku pravljenja aplikacije. Resursi aplikacije se mogu koristiti programerski. Takođe na njih se mogu odnositi resursi drugih resursa aplikacija.

Svaki put kad se sačuva fajl sa resursima u Eclipsu, R.java klasa fajl se ponovo sastavlja kako bi se ubacile nove izmene. Ako nije korišćen pravi direktorijum ili pravila o nazivu fajla, pojaviće se greška u kompjuleru u prozoru Eclipse Problems.

Resursima aplikacije se može pristupiti programerski koristeći generisnu klasu koja se zove R.java. Kako bi se dodala referenca na resurs, mora se doći do objekta resursa aplikacije koristeći klasu getResources() i tako napraviti odgovarajući poziv na komandu, koji je zasnovan na tipu resursa koji želi da se doda.

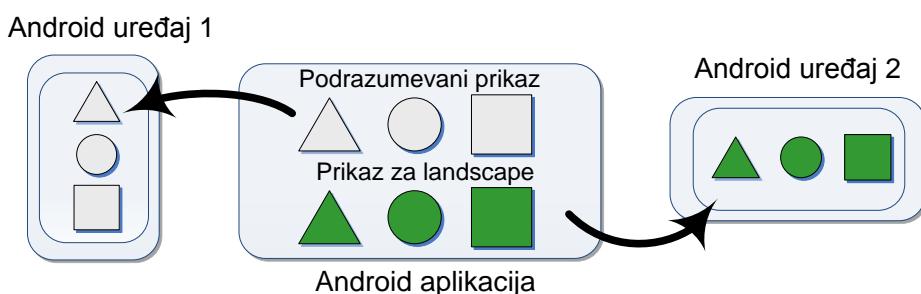
Developeri mogu da pristupe resursima Android sistema (kao dodatak resursima aplikacije). Mogu se deliti mnogi resursi sistema kroz više aplikacija zbog običnog izgleda i osećaja. Na primer, klasa resursa Android sistema (android.R.string) sadrži stringove za reči kao što su OK, Cancel, Yes, No, Cut, Copy i Paste.

Sistemski resursi se čuvaju u android paketu. Postoje klase za svaki od većih tipova resursa. Na primer, klasa android.R.string sadrži string sistemskog resursa. Kako bi se dobio string sistemskog resursa za OK prvo se mora koristiti statička komanda klase Resursa getSystem() kako bi se doveo globalni sistemski objekat Resource. Onda se poziva komanda getString() sa odgovarajućim string imenom resursa.

#### 2.5.4.1. Upravljanje i pristup resursima

U ovom poglavlju biće prikazano kako android aplikacije skladište i pristupaju važnim resursima kao što su stringovi, grafika i ostali podaci. Takođe će biti prikazano kako da organizujete android resurse u okviru projekta kako bi se na najlakši način mogla izvršiti lokalizacija i prilagođavanje aplikacija za različite konfiguracione uređaje. Kako bi se omogućila kompatibilnost sa različitim uređajima potrebno je resurse organizovati u folderima(direktorijumima) i podfolderima kako bi se svaki deo android aplikacije mogao održavati nezavisno od ostalih delova aplikacije.

Podrazumevani stil prikazivanja aplikacije čuva se na putanji **res/layout/** međutim možete podešiti u konfiguracionom fajlu da kada se aplikacija pokrene na neki uređaj koji ima drugačiju rezoluciju ekrana, npr koristi Landscape orientaciju, da tada poziva funkcije za prikaz is foldera **res/layout-land/** gde će na ovaj način sama aplikacija zavisno od rezolucije uređaja prikazati adekvatan sadržaj.



Slika 3. Različit prikaz za različitu rezoluciju uređaja

Korišćenje resursa koristeći Eclipse razvojno okruženje sa dodatkom Android Asset Packaging Tool je veoma jednostavno, nije potrebno kompajlirati resurse, to će Eclipse sam uraditi,

međutim ako koristite neko drugo razvojno okuženje potrebno je sve resurse komplajlirati iz komandne linije ili pak možete napisati skriptu koja će to uraditi, potom tu skriptu možete koristiti za sve vaše aplikacije.

Definisanje podataka aplikacije kao resursa je dobra vežba programiranja. Grupisanje resursa aplikacije i njihovo spajanje u paket aplikacije ima sledeće koristi:

- Kod je jasniji i lakši za čitanje, što dovodi do manje grešaka.
- Resursi su organizovani po tipovima i sigurno su jedinstveni
- Resursi su dobro smešteni za prilagođavanje telefonima

#### **2.5.4.2. Tipovi resursa**

Android aplikacije koriste veliki broj različitih tipova resursa kao što je tekst, grafika, šeme u boji za dizajn korisničkog interfejsa itd. Ovi resursi se nalaze u rez/ direktorijumu vaše android aplikacije.

Fajlovi resursa sačuvani u /res direktorijumu moraju da poštuju sledeća pravila:

- Imena fajlova resursa moraju biti napisani malim slovima
- Imena fajlova resursa mogu sadržati samo slova, brojeve, donju crtu, i tačke
- Imena fajlova resursa (i XML atributi imena) moraju biti jedinstveni.

U narednoj tabeli prikazano je kako se čuvaju resursi koje podržava Andoird SDK.

Tabela 2. Kako organizovati resurse u Android projektu

Tip resursa	Potreban direktorijum	Naziv	XML tag
Strings	/res/values/	strings.xml	<string>
Niz stringova	/res/values/	strings.xml	<string-array>, <item>
Boolens	/res/values/	bools.xml	<bool>
Colors	/res/values/	colors.xml	<color>
Dimensions	/res/values/	dimens.xml	<dimen>
Integers	/res/values/	integers.xml	<integer>
Mešovit niz	/res/values/	arrays.xml	<array>, <item>
Jednostavnii elementi (Drawables)	/res/values/	drawables.xml	<drawable>
Graphics (slike)	/res/drawable/	icon.png, logo.jpg itd.	
Meni	/res/menu/	Primer:	<menu>

		mainmenu.xml helpmenu.xml	
XML fajlovi	/res/xml/		
Izgled (Layouts)	/res/layout/		
Stilovi i teme	/res/values/	styles.xml themes.xml	<style>
Animacija	/res/drawable/ /res/anim/	Primer: sekvenca1.xml sekvenca2.xml	<animation-list> <item>

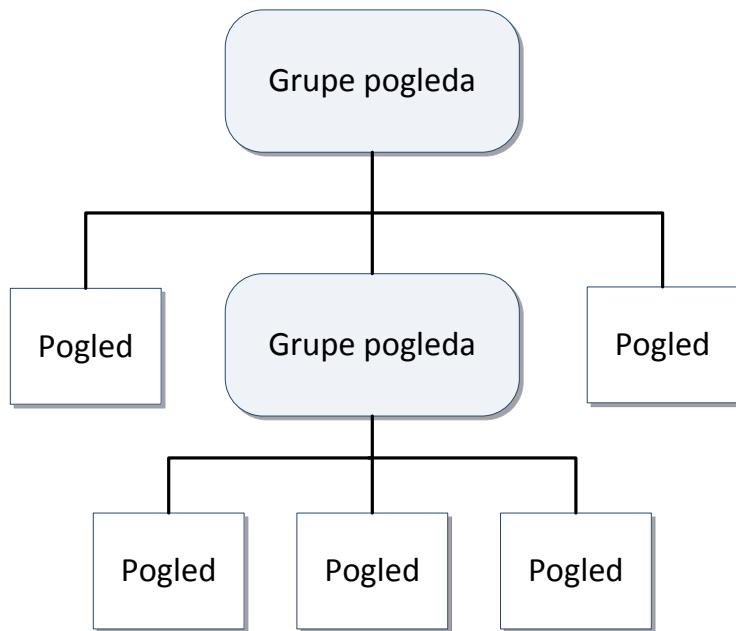
### 2.5.5. Korisnički interfejs

Postoje dva načina dizajniranja korisničkog interfejsa: proceduralno i deklarativno.

Proceduralni dizajn odnosi se na pisanje Java koda, a deklarativni na pisanje XML (eng. Extensible Markup Language) koda. U praksi se za kreiranje grafičkog korisničkog interfejsa uglavnom koristi XML.

Kreiranjem interfejsa aktivnosti dobijaju svoju funkcionalnost, tj. vidljivost na ekranu uređaja i na taj se način omogućava interakciju sa korisnikom. Osnovne jedinice korisničkog interfejsa su objekti pogled (*View*) i grupe pogleda (*ViewGroup*):

- pogled – objekat čija struktura u sebi nosi zapis izgleda i sadržaja određenog pravougaonog područja na ekranu, upravlja iscrtavanjem elemenata, pomeranjem sadržaja na ekranu (*scrolling*) i ostalim faktorima koji utiču na izgled definisanog dela ekrana. U hijerarhijskom stablu objekti pogled su listovi stabla. Android raspolaže sa već gotovim skupovima objekata ove vrste kao što su dugmad, kvadratići za odabir (eng. checkbok) i slično. Ovi objekti nasleđuju klasu View;
- grupe pogleda – posebna vrsta objekta pogled koja sadrži i upravlja grupom zavisnih objekata pogleda i grupe pogleda čime je omogućena kompleksnost prikaza korisničkog interfejsa. Objekti ove vrste su instance klase ViewGroup.



Slika 4. Hijerarhijski prikaz elemenata korisničkog interfejsa

Iscrтavanje elemenata stabla započињe od korena stabla tako што aktivnost prvo poziva svoju `setContentView()` metodu i Android operativnom sistemu predaje referencu na koren objekat. Svaki podčvor iscrtava se sam pozivanjem `draw()` metode i to pod uslovom da ћvor sam postavlja zahtev za lokacijom i veličinom, ali roditeljski ћvor (grupa pogleda) donosi konačnu odluku o veličini prostora za iscrtavanje podčvora i njegovoј lokaciji na ekranu. Svaka grupa pogleda je odgovorna za renderovanje prikaza svojih podčvorova.

### **3. UČENJE JEZIKA POTPOMOZNUTO MOBILNIM TEHNOLOGIJAMA**

Učenje jezika je pogodna oblast za primenu modernih tehnologija u vidu e-obrazovanja i m-obrazovanja. Učenje jezika predstavlja tematiku koja je potencijalno zanimljiva pojedincima svih starosnih doba, stepena obrazovanosti i zanimanja, a ne zahteva nikakvo predznanje. Granulacija gradiva omogućava lako grupisanje i strukturiranje manjih jedinica u veće i prilagođavanje potrebama korisnika. Samim tim i motivacije učenika i konteksti u kojima se odvija učenje mogu značajno da variraju što stvara veliki prostor za primenu modernih tehnologija.

Učenje potpomognuto mobilnim tehnologijama (Mobile Assisted Language Learning, MALL) označava pristup učenju jezika koji se ili u potpunosti oslanja na korišćenje mobilnih uređaja, ili je delom potpomognut i poboljšan njihovom upotrebljom.

Pogodnost modernih tehnologija za implementaciju i proverenih, ustaljenih i modernih, eksperimentalnih procesa učenja je potkrepljena brojem istraživanja u ovoj oblasti kao i brojem postojećih, komercijalnih rešenja. Rešenja za učenje stranih jezika koja primenjuju moderne tehnologije možemo podeliti u tri grupe, prema vrsti platforme za koju su namenjena: desktop aplikacije, online aplikacije, mobilne aplikacije.

Desktop rešenja se mogu posmatrati kroz manji broj obimnih i kvalitetnih profesionalnih jezičkih aplikacija sa jedne strane i veliki broj manjih aplikacija osmišljenih od strane entuzijasta. Ono što ih karakteriše je to što zahtevaju potpunu posvećenost korisnika i dosta vremena za savladavanje jezičkih lekcija. Online rešenja su izgrađena na principima umrežavanja i povezivanja korisnika. Jedan od interesantnih pristupa je kombinovanje sajtova za društveno umrežavanje sa učenjem jezika gde se korisnici usmeravaju da pomažu jedni drugima u procesu učenja. Tendencija je i da se iskoriste i proizvodi samih korisnika, tj. da se oni uključe u proces stvaranja i zatim razmene raznovrsnih jezičkih resursa. Prednost mobilnih rešenja je upravo u mobilnosti jer u svakom trenutku omogućavaju lagano učenje ili podršku ozbiljnijem učenju koje se odvija korišćenjem nekog drugog alata (u jezičkoj školi, korišćenjem literature, desktop aplikacija i slično).

Danas postoji trend konvergencije svih navedenih rešenja. Desktop aplikacije se sve češće oslanjaju i na internet za dobavljanje jezičkih resursa i povezivanje korisnika; onlajn

aplikacije, shodno trendu smeštanja aplikacija u oblak, postaju sposobne da zamene desktop aplikacije i dostupne su na svim platformama; na kraju, mobilna rešenja se razvijaju uporedno sa platformama za koje su namenjena i postaju sve složenija, pri tome se oslanjajući na internet konektivnost koja postaje standard za mobilne uređaje. Ipak, i pored ovog trenda, mobilni uređaji se i dalje ističu kao najpogodnija tehnologija za primenu novih ideja. Iz osobine mobilnosti proizilazi fleksibilnost konteksta u kom se vrši učenje nedostupna desktop aplikacijama. Sa druge strane, onlajn aplikacije koje karakteriše univerzalnost pristupa ne mogu u potpunosti da iskoriste mogućnosti uređaja sa kog se vrši pristup i osteljive su na probleme uzrokovane ograničenom konektivnošću.

Istraživanja su pokazala da dobro projektovane mobilne aplikacije imaju motivišući uticaj na učenike jezika i da generalno nude dovoljno prilika za učenje kako bi stvarale pozitivan uticaj na proces učenja. Dokazana je njihova vrednost u obogaćivanju formalnog učenja jezika u školi. Takođe je pokazano da se obezbeđivanjem prilike za učenje u neformalnom okruženju, povećava ulaganje vremena koje učenici ulažu u učenje [17].

Zahvaljujući svojoj raznovrsnosti i rasprostranjenosti, mobilni uređaji postaju alati za sverprisutno učenje. Pogotovo su od značaja moderni mobilni telefoni koji u sebi obuhvataju niz funkcionalnosti - vođenje razgovora (sa predavačem, sa automatskim sistemom), slanje i prijem SMS poruka (za dostavljanje kratkih lekcija), snimanje i puštanje audio zapisa (za snimanje i pregled lekcija), internet konektivnost (za pristup resursima i kolaboraciju), itd.

### **3.1. Tehnologije obuhvaćene MALL-om**

Moderni mobilni uređaji, pre svega mobilni telefoni, poseduju niz funkcionalnosti koje predstavljaju primenu raznovrsnih tehnologija. Ove funkcionalnosti praktično predstavljaju jedan bogat skup alata koji se mogu upotrebiti u učenju jezika bilo pojedinačno, bilo međusobnim kombinovanjem. U nastavku će biti obrađene sledeće funkcionalnosti mobilnih uređaja:

- Mogućnost obavljanja telefonskih poziva i slanja SMS poruka, koja predstavlja osnovnu funkcionalnost mobilnih uređaja i koja je, kao takva, lako dostupna.
- Mogućnost snimanja i reprodukcije audio zapisa, što je funkcionalnost koja je duže vremena prisutna, počevši još od prenosnih kasetofona, pa do modernih mp3 plejera i opet mobilnih telefona.

- Mogućnost upotrebe interneta i procesiranja i interakcije sa korisnikom, funkcionalnost koja konačno izjednačava mobilne telefone i slične uređaje (PDA, tablet) sa klasičnim računarima.

### **3.1.1. Telefonski pozivi & SMS poruke**

Budući da je mogućnost obavljanja poziva osnovna funkcionalnost mobilnog telefona, može se iskoristiti u procesu m-learninga, pogotovo u oblasti učenja jezika za preslušavanje audio materijala ili čak dvosmernu komunikaciju na jeziku koji se uči. Twarog i Pereszlenyi-Pinter tvrde u svom radu [18] da sistem za dostavljanje lekcija, audio zapisa i drugih materijala koji se koriste na regularnim časovima preko telefona može efektivno da se koristi za učenje stranog jezika.

Druga funkcionalnost, praktično nerazdvojiva od mobilnih uređaja, je mogućnost slanja i prijema kratkih tekstualnih prouka (SMS). Ove poruke se mogu iskoristiti za dostavljanje edukativnih sadržaja korisnicima. Naprednije varijante kao MMS mogu da sadrže i bogatije, multimedijalne sadržaje. Studija izvedena u Japanu [19] sa lekcijama engleskog jezika dostavljenim studentima tri puta dnevno preko SMS-a je pokazala da su studenti koji su primali lekcije preko SMS-a ostvarili bolje rezultate (po broju naučenih reči i poboljšanju rezultata na testovima) od onih koji su učili preko Web-a i u klasičnoj papirnoj formi. Prepostavka autora je da je pozitivan uticaj delom ostvaren zahvaljujući „push-media“ osobini SMS poruka koje su podsticale ponavljanje i učenje sa razmakom više puta u toku dana.

Brown [20] analizira tri moda interakcije sa korisnikom, na primeru aplikacije za učenje španskog jezika, i definiše njihove prednosti i mane:

- Tekstualni kviz – pitanja predstavljena u kvizu predstavljaju zgodne kompaktne celine koje mogu da se upotrebe u kratkim intervalima slobodnog vremena; sa druge strane, koncentrisanje na ekran može da bude naporno zavisno od osvetljenja i situacije a, ukoliko se koristi samo tekst, iskustvo učenja nije dovoljno sveobuhvatno.
- Komunikacija sa instruktorem korišćenjem mobilnog telefona – komunikacija sa živom osobom daje idealne uslove za učenje stranog jezika; problem može biti teže razumevanje preko telefona i obezbeđivanje dovoljno vremena za sve učenike.

- Interaktivna audio komunikacija sa pregledom rečnika i testiranjem znanja preko mobilnog telefona – prednost ovog pristupa je što može da se vrši u paraleli sa drugim, svakodnevnim aktivnostima, uz mogućnost opsluživanja velikog broja korisnika i vođenja detaljne evidencije o njima na osnovu koje će se vršiti personalizacija aktivnosti. Nedostaci su tehničke prirode – potrebna je tehnologija za prepoznavanje govora i razvoj kvalitetnog audio-interfejsa.

Istraživanje učenja jezika u uslovima gde je proces značajno fragmentiran i gde učenik ne može da posveti kontinualnu pažnju niti sa sigurnošću odvoji neki manji interval vremena za učenje (npr. tokom vožnje) daje sledeće koncepte koje treba poštovati pri razvoju sistema i aplikacija namenjenih takvim uslovima:

- Ponavljanje nasuprot uvođenju novih koncepata – zahteva manje pažnje u uslovima podeljene koncentracije.
- Oslanjanje na poznat, utvrđen mod interaktivnosti – koristiti standardan model interaktivnosti i izbegavati aktivnosti koje zahtevaju složene instrukcije.
- Minimiziranje interaktivnosti – dozvoliti korisniku da može da napusti aktivnost u bilo kom trenutku i vrati joj se kasnije.
- Modularne jedinice učenja – podeliti gradivo u diskrete, zaokružene jedinice koje se mogu savladati bez obzira na vremenska ograničenja.
- Personalizovana interakcija – dozvoliti korisniku da sam izabere aktivnost i prilagodi iskustvo svom stilu učenja, stanju uma, nivou na kom se nalazi i nivou koncentracije.

### **3.1.2. Snimanje/reprodukciјa audio zapisa**

U studiji na univerzitetu u Viskonsinu [21] mobilni uređaji (PDA) su upotrebljeni na nekoliko jezičkih kurseva, kao ispomoć formalnom obrazovanju. Na časovima norveškog učenici su mogli, po nalogu predavača, da pristupe kratkim aktivnostima postavljenim na web serveru koje su im bile dostupne i od kuće. Na taj načim, čas je obogaćen tehnološkim sadržajima bez potrebe da se časovi premeštaju u klasičnu učionicu opremljenu računarima što bi zahtevalo i da se nalgasak celog časa prebaci na tehnološku stranu. Na časovima francuskog primenjeno je nekoliko oblika čet-aktivnosti pod nadzorom predavača koji je posećivao čat-sobe učenika; najefektivnijim se pokazalo korišćenje malih grupa koje bi kasnije učestvovali u zajedničkoj oralnoj raspravi tema pomenutih u čet prostoriji. Na časovima latinske književnosti, učenicima su obezbeđeni interaktivni latinski tekstovi kojima su učenici mogli da dodaju akcente i tako akcentovane ih predaju predavaču.

Prenosni uređaji koji imaju sposobnost snimanja i puštanja audio zapisa su odavno dostupni, u početku u obliku prenosnih kasetofona, a kasnije u obliku daleko komfornijih MP3 plejera. Današnji mobilni telefoni takođe imaju ove mogućnosti, dok uređaji specifično namenjeni puštanju/snimanju zvuka opstaju zahvaljujući većoj kompaktnosti i nižoj ceni.

Analizom raznih oblika upotrebe iPod muzičkih plejera u nastavi na Duke univerzitetu, Belanger formuliše sledeće zaključke [22]:

- Distribucija sadržaja (lekcija) – studenti su ocenili mogućnost naknadnog pristupa audio-sadržajima sa časova značajnom, pogotovo u slučajevima gde je razumevanje izgovorenog sadržaja na stranom jeziku bilo od važnosti
- Snimanje predavanja u učionici – iPod se pokazao kako koristan alat za snimanje predavanja i interakcija u učionici; najbolji rezultati su ostvareni u snimanju interakcija u malim grupama.
- Snimanje na terenu – i ovde je iPod ocenjen kao koristan alat od strane studenata i nastavnika; ocenjeno je da se njegovim korišćenjem na terenu poboljšava iskustvo i da su sami časovi zanimljiviji.
- Alat za podršku nastavi – Dok su neki predmeti, prvenstveno jezički, koristili iPod deo nastave (za razliku od jednostavnog obezbeđivanja pristupa lekcijama), student su često samostalno nalazili načine upotrebe iPod-a koji nisu bili predviđeni nastavom.
- Alat za skladištenje i prenos materijala – zahvaljujući velikom dostupnom prostoru na iPod-u (20GB), studenti su ga koristili za prenos materijala sa nastavnih računara kući i obrnuto.

### **3.1.3. Internet i mogućnosti obrade i interakcije sa korisnikom**

Pored mobilnih telefona i njihovih osnovnih funkcionalnosti – telefonskih razgovora i slanja SMS-ova, značajan element M-learninga su predstavljali PDA uređaji i, kasnije, pametni telefoni. Današnji pametni telefoni su preuzeли sve napredne osobine nekadašnjih PDA uređaja, a čak i obični telefoni iz nižeg cenovnog ranga podrazumevano poseduju internet

konektivnost i solidnu sposobnost reprodukcije multimedijalnih sadržaja. Ovo je izuzetno bitan aspekt mobilnih telefona sa strane mobilnog obrazovanja.

Ono što omogućava upotrebu interneta korišćenjem mobilnih telefona su, u osnovi, njihove hardverske sposobnosti koje su značajno porasle u proteklih nekoliko godina. Zahvaljujući većim ekranima, većom obradnom moći i memorijom, boljim grafičkim sposobnostima i raznolikim vidovima povezivanja (Bluetooth, Wi-Fi, 3G itd.) moguće je razviti mnogo složenije aplikacije koje bi, uz kvalitetniju prezentaciju i bogatstvo multimedije, mogle da stanu rame uz rame sa klasičnim računarima.

### **3.2. Praktični radovi na temu učenja jezika uz pomoć mobilnih tehnologija**

Postoji čitav niz radova koji se bave upotrebom pojedinih mogućnosti mobilnih uređaja u učenju jezika, i njihovim povezivanjem sa drugim modernim (desktop, onlajn) tehnologijama. Studija "Move, idioms!" se bavi proučavanjem strategija i pristupa učenju u okruženju gde je kombinovano formalno učenje u učionici i neformalno izvan nje, pri čemu su učenici bili usmereni na kreiranje i međusobno komentarisanje sadržaja. Ključni element studije je bila aplikacija za mobilne uređaje koja je predstavljala neophodan alat za stvaranje sadržaja i za obavljanje interakcije sa drugim studentima [14].

Dalje, jedna studija predlaže korišćenje sličnog sistema (StudentPartner) koji kombinuje mobilne uređaje sa GPS-om i internet za tagovanje i opisivanje lokacija u okruženju na jeziku koji se uči. Ovo sa jedne strane uključuje studente u kreiranje sadržaja, a sa druge strane olakšava učenje povezivanjem edukativnog sadržaja sa poznatim lokacijama [15]. U obe navedene studije značajnu ulogu igra kontekst u kom se obavlja učenje; studija [23] detaljnije razmatra granulaciju konteksta na nivou lokacije i na nivou pojedinačnih objekata, u kombinaciji sa četiri pristupa prikazu informacija sa različitim stepenima automatizacije. Uticaj multimedijalnih elemenata na mobilnim uređajima na proces učenja je detaljno razmotren u [24] kao i tehnički problemi koji se javljaju i preporuke za maksimizaciju njihovog efekta.

Još neki od primera uspešnih aplikacija za učenje jezika su: MEL-application - učenje Engleskog jezika kroz niz aktivnosti preko mobilnog uređaja (lekcije, video i audio materijali, kvizovi, igra pamćenja) [17]; Mobilna flash aplikacija FML4ESL - materijal podeljen u mini-lekcije, aplikacija razvijana za rezoluciju 240\*320, neprilagođena za veće ekrane, lekcije

propraćene audio zapisima i slikama [25]; PIMS (Personalized intelligent mobile learning system) - u kombinaciji sa personalizovanim sistemom za građenje rečnika, aplikacija za pomoć učenju, korišćena u kombinaciji sa klasičnim časovima [26]; PALLAS - sistem korišćenje sistema sa mobilnih uređaja moguće putem klijentske aplikacije, mobilnog brauzera ili putem SMS-a, jezičke aktivnosti obuhvataju testove, vežbe, rečnik koji se pokreću manuelno i aktivnosti koje se okidaju zavisno od kontekstnih parametara [27].

Danas, Android OS predstavlja mobilni operativni sistem sa najvećim brojem korisnika u svetu. Veći broj istraživanja se odnosi na razvoj i primenu Android aplikacija u različitim sferama ljudskog delovanja [28]. Android OS obezbeđuje mnoštvo različitih tehnologija i resursa nalik desktop operativnim sistemima, pruža mogućnost za razvoj kvalitetnog korisničkog interfejsa, uz pomoć „touch-screen“ ekrana.

### **3.3. Objekti učenja**

U oblasti elektronskog učenja javlja se koncept "objekata učenja", malih, kompaktnih edukativnih celina skoncentrisanih na jedan cilj. Iako različiti autori daju različite definicije za objekte učenja, naglasak se kod većine stavlja na njihove osobine: manji obim, međusobna nezavisnost, ponovna upotrebljivost, mogućnost agregiranja. Ove osobine se pogodno mogu iskoristiti i u mobilnom okruženju.

Najznačajnije osobine objekata učenja su:

- Dostupnost – objekti učenja bi trebalo da budu označeni pomoću metapodataka kako bi mogli da se uskladište i referenciraju u bazi podataka.
- Ponovna upotrebljivost – jednom napravljeni objekti učenja treba da budu sposobni da funkcionišu u različitim kontekstima.
- Interoperabilnost – objekti učenja bi trebalo da budu nezavisni od načina koji se koristi za njihovo dostavljanje i sistema za upravljanje znanjem u kojem se koriste.

#### **3.3.1. Definicije i karakteristike**

Postoji čitav niz definicija objekata učenja. Neke od definicija su dosta uopštene i mogu da se odnose na praktično bilo kakve materijale koji mogu da se koriste u učenju dok se druge definicije koncentrišu na određene aspekte objekata učenja.

IEEE definiše objekat učenja kao bilo koji entitet, digitalan ili ne-digitalan koji može da se koristi za učenje, edukaciju ili obuku [29].

L'Allier definiše objekat učenja kao najmanje nezavisno strukturno iskustvo koje sadrži cilj, aktivnost učenja i procenu znanja. U ovom kontekstu, cilj je element objekta učenja koji predstavlja izjavu u vezi sa rezultatom aktivnosti učenja zasnovanom na nekim kriterijumima, aktivnost učenja je komponenta koja podučava korisnika kako bi ostvario zadati cilj, a provera znanja treba da utvrди da li je cilj postignut [30].

Pithamber R. Polsani u svom radu [31] tvrdi da je IEEE-ova definicija previše opšta i nepraktična pošto dodeljuje isti konceptualni status i ne-digitalnim objektima kao što je kompjuterski hardver i digitalnim objektima kao što su slike, čime onemogućava korišćenje termina „objekat učenja“ na koristan način. Za L'Allier-ovu definiciju tvrdi da, iako jasno artikulisana, ograničava ponovnu upotrebljivost objekata učenja time što zahteva definisanje cilja, metoda i mehanizma provere znanja unapred. Ovi elementi bi trebalo da su definisani situacijom podučavanja (kontekstom) a ne samim objektom učenja.

Po Poslaniju, neophodno je prvo definisati dva principa koji predstavljaju kamen temeljac za definiciju objekata učenja: učenje i ponovnu upotrebljivost. Da bi nekakav digitalni objekat ili resurs zadobio status objekta učenja, mora da bude umotan u nameru učenja (Learning Intention), koja ima dva aspekta – formu i relaciju. Digitalni objekat mora da bude inkorporiran u formu i da obezbedi relaciju sa sobom kao objektom učenja kako bi omogućio razumevanje tog objekta. Forma i relacija obezbeđuju mehanizam za interno funkcionisanje objekta učenja, a ponovna upotrebljivost mu daje određenu vrednost. Samo kada je objekat učenja projektovan tako da bude upotrebljiv od strane više developera u različitim kontekstima, može da iskoristi osobine fleksibilnosti, skalabilnosti i prilagodljivosti zasnovane na informacionim tehnologijama. Iz svega navedenog, proizilazi sledeća definicija: objekat učenja je nezavisna, samostalna jedinica edukativnog sadržaja koji je namenjen ponovnoj upotrebi u različitim kontekstima podučavanja.

Rory McGreal u svom radu „Learning Objects: A Practical Definition“ daje pregled postojećih definicija iz otvorene literature i svrstava ih u nekoliko kategorija zavosno od korišćene terminologije [32]:

- Bilo šta (*Asset, Component, Learning Resource*) – Ideja je da bilo koji objekat može da posluži u učenju i samim tim je objekat učenja; nema potrebe unapred

ograničavati šta može a šta ne može da bude objekat učenja. Da li će se neki objekat računati kao objekat učenja zavisi samo od načina njegove upotrebe, a ne i od njegove prirode.

- Bilo šta digitalno (*Content Object, Information Object, Knowledge Object, Media Object, Raw Media Element, Reusable Information Object (RIO)*) – Ova definicija se odnosi samo na digitalne objekte; reč „objekat“ u ovim definicijama svoje poreklo vuče iz termina „objektno orijentisano programiranje“
- Bilo šta namenjeno učenju (*Educational Object, Learning Object*) – Ne ograničava se samo na digitalne objekte već obuhvata pojmove kao što su web stranice, aplikacije, knjige, kalkulatori, mikroskopi.
- Specifično edukativno okruženje (*Reusable Learning Object (RLO), Unit of Learning, Unit of Study*) – Neki zagovornici objekata učenja ne prihvataju generičke definicije koje se odnose na digitalne objekte učenja i dodaju određena ograničenja. Ova ograničenja se bave ili specifičnim problemima za određenu oblast upotrebe ili čisto uokviruju standardni pristup kod nekih preciznije definisanih operacija sa objektima učenja.

### **3.3.2. Primena objekata učenja u konkretnim aplikacijama**

Longmire W. daje određene preporuke dizajnerima objekata učenja u pogledu njihovog dizajna i strukture [33]:

- Jezik i terminologija u određenoj tematskoj oblasti treba da budu konzistentni, što će omogućiti lako rastavljanje i ponovno sastavljanje objekata učenja.
- Informacije treba da budu prezentovane u lako razumljivom formatu korišćenjem tabela, lista stavki i kolona po potrebi.
- Informacije treba da budu organizovane sa prikazom na ekranu na umu, uz praćenje standardnih tehnika dizajna web sadržaja
- Informacije distribuirane u više objekata treba da budu nesekvencijalne. Informacije treba da budu samostalne i da lako mogu da se prilagode različitim kontekstima. Drugim rečima, treba izbeći međusobno referenciranje među objektima.
- Uniformno predstavljanje sadržaja u svim objektima. Treba izbeći „uvodne“ i „zaključne“ objekte.

- Koristiti ključne reči u elementima koji se mogu pretraživati.
- Koristiti jezik i sadržaj razumljiv široj publici. Na ovaj način se izbegava potreba za manuelnim prilagođivanjem objekata pri ponovnoj upotrebi.

Integracija repozitorijuma objekata učenja i jezičkih programa stvara određene prilike ali i izazove; Caws, Friesen i Beaudoi ih analiziraju na primeru svoje web aplikacije za učenje francuskog jezika i odgovarajućeg FLORE repozitorijuma objekata učenja [34]. Ovaj repozitorijum je zasnovan na aplikacionom profilu IEEE LOM-a dobijenim uprošćavanjem i izbacivanjem nepotrebnih elemenata i prilagođavanjem određenih elemenata osmišljanjem taksonomija za specifičnu namenu podučavanja francuskog jezika u zemljama frankofonije.

U uslovima mobilnog okruženja, potrebno je u određenoj meri prilagoditi objekte učenja, pogotovo u domenu prezentacije. Churchill, D. daje niz preporuka vezanih za dizajn mobilnih objekata učenja namenjenih prikazu na malim ekranima prenosnih uređaja [35]. Među datim preporukama su:

- Vizualna prezentacija informacija (fotografije, dijagrami, grafici, ikone, simboli), uz ponavljanje istih informacija korišćenjem više vidova prezentacije ukoliko je potrebno.
- Ugrađena interaktivnost.
- Svi elementi da budu deo sveobuhvatne prezentacije.
- Projektovati za prikaz na jednom ekranu i minimalnom prostoru.
- Minimizirati upotrebu audio/video klipova, ne preterivati sa bojama i dekorativnim elementima, držati se standardnog fonta.
- Po potrebi podeliti prostor korišćenjem frejmova.
- Projektovati za prikaz u full-screen modu i landscape orijentaciji koja može da pruži veću fleksibilnost; minimizirati skrolovanje.
- Eliminisati složene interakcije i projektovati kratke zadatke.
- Omogućiti zumiranje i koristiti pokretljive/smanjive/preklopljene/poluprovidne elemente interfejsa
- Projektovati sadržaj tako da omogući učenicima da povezuju koncepte iz stvarnog sveta sa onima prezentovanim kroz objekte učenja
- Projektovati za analitičku upotrebu tako da korisnici mogu da unose vrednosti iz okruženja i dobiju rezultate
- Projektovati za eksperimentisanje – omogućiti manipulaciju sadržajem i posmatranje rezultujućih promena

- Projektovati za razmišljanje, privući pažnju korisnicima postavljenim tragovima i podstaknuti im radoznalost.
- Projektovati za ponovnu upotrebu – uzeti u obzir mogućnost prezentacije objekata na različitim uređajima, računaru, projektoru, interaktivnoj tabli itd.

### **3.4. Japanski jezik**

Za razliku od učenja nekog od evropskih jezika koji se služi latiničnim (ili ciriličnim) pismom, učenje japanskog je dosta otežano za ljude sa zapada zbog složenog pisma u upotrebi. Učenje japanskog zahteva posvećivanje podjednako vremena učenju pisanja/čitanja kao i učenju reči/gramatike, za razliku od nekog evropskog jezika gde bi se odmah mogli baciti na učenje novih reči.

Japanski jezik koristi tri pisma:

- hiraganu
- katakanu
- kandži

Kandži, ili u bukvalnom prevodu sa japanskog „kineski znaci“ je kinesko znakovno pismo koje su japanci usvojili negde oko 5. veka nove ere, dok pre toga njihov jezik nije imao pisanu formu. Danas se u svakodnevnoj upotrebi oko 2000 zakonom propisanih kandži znaka, i sve novine i moderna literatura se štampaju koristeći taj fond znakova. Ovi znakovi se uče tokom obavezognog obrazovanja, tokom 6 godina osnovne i 3 godine niže srednje škole i očekuje se da ih svako zna. Pored ovih znakova, postoje još i znakovi koji se izučavaju na fakultetima i arhaični znakovi, koji se vrlo retko mogu sresti u modernoj literaturi. Sa druge strane, broj kineskih znakova potrebnih za život u kini se kreće od najosnovnijih 2000 do 4000-5000 koliko se očekuje od bolje obrazovanih pojedinaca.

Svaki kandži nosi neko značenje i može da predstavlja celu reč ili deo neke reči. Kandžiji mogu da imaju više čitanja/značenja, i ona se grupišu u dve kategorije: **onyomi** i **kunyomi**. Onyomi su čitanja preuzeta od jednog od kineskih jezika u vreme kada su preuzeti i kineski znaci; ova čitanja se danas više ne poklapaju preterano sa modernim kineskim čitanjima. Onyomi čitanja se obično koriste kada se reč slaže iz više kandži znakova. Kunyomi čitanje je čisto japansko čitanje i obično se koristi kada znak stoji sam i predstavlja neku reč.



Značenje: cvet

Onyomi (on-čitanje): **ka** (か)

(koristi se u složenicama, npr. „kafun“ – polen)

Kunyomi (kun-čitanje): **hana** (はな)

(kompletna reč, znači „cvet“)

Pošto se japanski i kineski jezik jako razlikuju, japanci su imali problema u upotrebi kineskog pisma. Da bi prilagodili ove znake svom jeziku, vremenom su izmislili dva pomoćna pisma, hiraganu i katakanu. Ova pisma imaju po 48 znakova i lako se mogu posmatrati kao japanski ekvivalent naše cirilice i latinice. Hiragana se koristi za pisanje japanskih reči, a katakana za pisanje stranih reči, onomatopeja, imena biljaka, životinja, minerala, naučne termine, imena kompanija itd. Znaci ovih pisama su u potpunosti ekvivalentni jedni drugima (kao slova cirilice i latinice) i, kao i zapadna pisma, nemaju nikakva značenja osim glasova na koje se odnose. Razlika u odnosu na zapadna pisma je ta što, izuzev slova a, i, o, e, u, n, ostali znakovi predstavljaju slogove (ka,ki,ko,ke,ku,ma,mi,mo...). Hiragana i katakana se često pominju zajedničkim imenom „kana“.

Pored ova tri pisma, japanske reči napisane korišćenjem latinice se nazivaju „romađi“, u bukvalnom prevodu sa japanskog „roman letters“. Početnici koji uče japanski jezik kreću od reči ispisanih romađijem, tj. latinicom, a onda vremenom mogu da pređu na reči ispisane samo kanom (hiraganom/katakanom, zavisno od same reči) i na kraju na reči ispisane kandžima. Reči ispisane kandžijima mogu biti sastavljene samo od kandžija (npr. kod imenica koje se nikada ne menjaju kroz rod/množinu/padeže) ali mogu imati i kana znakove (pogotovo kod glagola pri promenama vremena i sl.). Takođe, neke prostije reči i gramatičke konstrukcije uopšte ne koriste kandži znake.

### **3.5. Postojeće komercijalne aplikacije za učenje japanskog jezika**

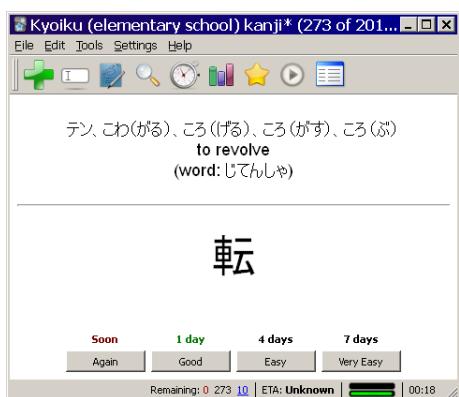
Učenje stranih jezika korišćenjem računara, a od skorije i raznih prenosnih uređaja je toliko rasprostranjeno da bi se lako mogao napisati čitav rad o postojećim rešenjima. U ovom odeljku će biti navedena neka od tih rešenja sa kratkim opisima primjenjenog pristupa/tehnologija i implementiranih funkcionalnosti.

### 3.5.1. Desktop rešenja



#### Rosetta stone

Rosetta stone je program za učenje jezika zasnovan na pristupu imerzije učenika, gde se učenje vrši po principu učenja maternjeg jezika. Program se instalira nezavisno od jezičkih paketa koji su dostupni za veliki broj svetskih jezika i ima sveden interfejs sa minimumom teksta. Pri učenju se ne daju značenja reči već se radi sa slikama i audio zapisima, uz primenu slikovitih kvizova i prepoznavanja izgovorenih reči, na taj način omogućavajući učenje nezavisno od toga koji jezik korisnik govori. Poslednja verzija ovog programa uspešno uključuje i internet u dobitnu jednačinu, obezbeđujući onlajn tutore, govornike jezika koji se uče. Ova verzija takođe postoji i u varijanti za mobilne uređaje i predstavlja sveobuhvatan primer primene modernih tehnologija u procesu učenja jezika.



#### Anki

Daleko jednostavniji program od gorepomenutog, Anki je interesantan jer može da se koristi za učenje bilo koje oblasti, ne samo jezika. Slično Rozeti, i Anki dolazi prazan i omogućava skidanje paketa „kartica“ sa kojih se uči. Učenje se vrši ponavljanjem kartica i samoocenom znanja i vrti se oko vremenski ograničenih sesija za koje se vode statistike i prema kojima se prilagođava tempo učenja. Internet se ovde koristi na jednostavniji, ali efektan način, jer osim skidanja novih paketa reči, korisnici mogu da prave svoje pakete i dele ih sa drugim korisnicima programa.

### 3.5.2. Online rešenja

The screenshot shows the Lang8 website interface. At the top, there are links for 'write', 'Lang8 Contents', and 'My Contents'. Below this is a sidebar with various user statistics and links like 'Write a new entry', 'My Friends', 'Groups', 'Language Matches', and 'Avatar'. The main content area displays a feed of user posts. One post from 'lang8 staff' asks if notifications should be shown in Chinese. Another post discusses an international exchange party in Tokyo. There are also posts about earthquakes in Japan and premium features. The interface includes social sharing options and a 'Read more' link.

### Lang8

Lang8 je sajt koji kombinuje moderne ideje društvenog umrežavanja sa učenjem jezika. Na ovom sajtu korisnici prave profil pri čemu biraju svoj maternji jezik i jezik koji žele da nauče. Onda mogu da pišu „dnevnik“ (ekvivalent fejsbukovom „zidu“) na željenom jeziku, gde će ih maternji govornici tog jezika slobodno ispravljati, a oni sami će ispravljati rečenice koje drugi pišu u svojim dnevnicima. Ukratko, sajt koji u potpunosti iskorišćava svoj medij – internet, povezujući ljude sa različitim krajeva sveta i predstavlja odličnu lokaciju za učenje iako ne sadrži nikakve jezičke resurse – za stvaranje istih su zaduženi korisnici.

The screenshot shows the Google Translate homepage. It features a search bar with 'From: English' and 'To: Serbian'. Below the search bar is a large input field with the placeholder 'Type text or a website address or translate a document.' At the bottom of the page, there are links for 'Google Translate for my: Searches, Videos, Email, Phone, Chat, Business' and buttons for 'About Google Translate', 'Turn off instant translation', 'Privacy', and 'Help'.

### Google translate

Revolucionaran onlajn servis za prevodenje sa velikim brojem dostupnih jezika – iako nije prvenstveno namenjen učenju jezika, višestruko je koristan kao pomoć u tom procesu. Velika baza reči koja se zasniva na statističkom prevodenju i mogućnost slobodnog korišćenja servisa iz drugih aplikacija omogućavaju „zaranjanje u jezik“ surfovanjem po internet stranicama na stranom jeziku korišćenjem neke ekstenzije za veb pregledač radi prevodenja nerazumljivih delova stranice (npr. „gTranslate“).

The screenshot shows the Livemocha.com website. At the top, there are links for 'Practice Activities', 'Vocabulary Courses', 'Video & Audio Clips', 'Flashcards', 'Writing Practice', 'Speaking Practice', and 'Games'. The main content area displays various language learning activities. A sidebar on the right shows a progress bar for a vocabulary course and a link to 'Get Started'.

### Livemocha.com

Lang8 je verovatno prvi sajt u svojoj kategoriji, ali nije i jedini. Livemocha prati istu logiku – međusobno pomaganje onih koji uče jezike, ali je i upotpunjuje sa bogatim jezičkim resursima i aktivnostima zasnovanim na web 2.0 platformi. Tu su još i top liste i rangiranje onih koji uče kao dodatna motivacija. U pitanju je obimna mešavina raznih, već pomenutih ideja za unapređenje učenja jezika modernim tehnologijama.

### 3.5.3. Mobilna (Android) rešenja



komponenata kao prethodna aplikacija. Nedostatak je nekorišćenje interneta.



### Japanese phrases with audio

Aplikacija za učenje oko hiljadu gotovih fraza, u potpunosti pokrivenih audio zapisima. Učenje se obavlja pomoću „fleš kartica“ sa dodatnim zanimljivostima za svaku frazu koje pomažu pamćenju. Obnavljanje se vrši jednostavnim kvizom. Fraze su podjeljene u logičke kategorije i moguće je pregledanje baze naučenih reči.

### Human Japanese Lite

Obimna aplikacija koja obuhvata reči sa audio zapisima, „lekcije“ koje se postepeno prelaze, kvizove za utvrđivanje znanja i deo za učenje kana znakova i njihovo utvrđivanjem putem igre „memorije“. Sadrži i rečnik svih reči u aplikaciji i animacije redosleda povlačenja kana znakova. GUI je prijatan za oko i ne oslanja se na klasični izgled Android grafičkih

komponenata kao prethodna aplikacija. Nedostatak je nekorišćenje interneta.

### Learn Japanese Pro Phrasebook

Jednostavna aplikacija namenjena onima koji žele da se upoznaju sa jezikom a najviše putnicima u Japan. Sadrži 700 reči i fraza sa audio zapisima, korisne zanimljivosti za putnike i pretragu baze reči. Na kraju, iako ne koristi internet. Iako relativno siromašna opcijama, iskorišćava mobilnost platforme da se pozicionira kao korisna alatka za putnike.

### JA Sensei

Aplikacija sa snažnim fokusom na japanska pisma, sa detaljnim prikazom redosleda povlačenja japanskih znakova, velikom bazom kandi znakova podjeljenoj po JLPT nivoima i razredima. Iskorišćena tehnologija „tačskrin“ ekrana omogućava vežbanje pisanja znakova crtanjem po ekranu.



## **4. RAZVOJ JAPGO APLIKACIJE**

Android sistem je idealan sistem za razvoj aplikacije za učenje/pomoć učenju u pokretu. Činjenica da je u pitanju operativni sistem za mobilne uređaje, a da opet obezbeđuje širok spetkar tehnologija i resursa nalik regularnim desktop operativnim sistemima pruža mogućnost za razvoj kvalitetnog korisničkog interfejsa, uz pomoć „touch-screen“ ekrana, dok razne opcije za povezivanje na internet višestruko mogu da povećaju upotrebljivost svake aplikacije koja upotrebi tu sposobnost. „Mobilna aplikacija za učenje japanskog jezika“ je projekat sa idejom da se napravi potpuno funkcionalna aplikacija za učenje japanskog jezika i da se usput prikaže životni ciklus razvoja jedne takve aplikacije koja bi upotrebila neke od dostupnih tehnologija Android operativnog sistema. Aplikacija je nazvana „JapGO“, što treba da asocira na podstrek („GO!“) za učenje japanskog jezika, a takođe je i višejezična igra reči, pošto se jezik na japanskom kaže „go“ (語).

### **4.1. Projektovanje aplikacije**

#### **4.1.1. Korisnički zahtevi**

JapGO je aplikacija koja treba da ispuni niz zahteva kako bi bila efikasna i efektivna u mobilnom okruženju. Pre svega treba da posluži bilo kao centralni deo procesa učenja japanskog jezika, bilo kao ispomoć u određenim oblastima. Edukativni sadržaj treba da bude prezentovan u određenim celinama, „mini-lekcijama“ pogodne veličine za uslove mobilnog okruženja, ali i da se svim sadržajima može pristupiti nezavisno od edukativnog konteksta.

Potrebno je obezbediti način za prezentaciju reči, uz poželjnu mogućnost puštanja audio klipova sa izgovorom reči. Dalje, treba na odgovarajući način prezentovati znake japanskih pisama – prikaz svih dostupnih znakova i detalje o pojedinačnim znakovima – i omogućiti korisniku način uvežbavanja pisanja znakova.

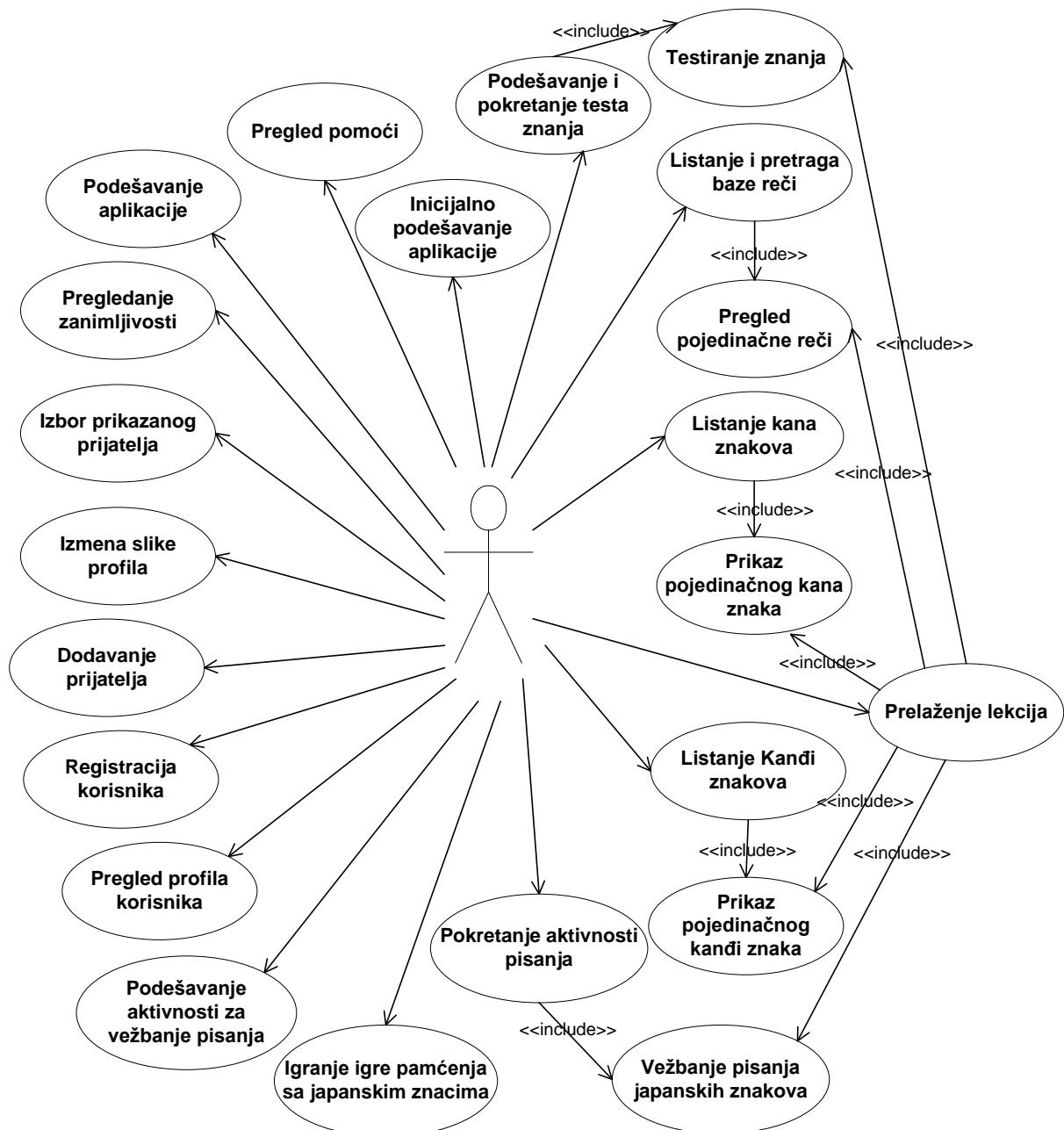
Za proveru znanja potrebno je obezbediti test znanja za svaku od lekcija koji će obuhvatati pitanja vezana za sadržaj lekcije; takođe omogućiti da kviz bude dovoljno fleksibilan u pogledu broja pitanja na testu, broja odgovora po pitanju, težine i varijante, i dozvoliti korisniku slobodan pristup svim podešavanjima nezavisno od pojedinačnih lekcija.

Aplikacija treba da bude multijezička – korisniku treba dati izbor jezika sa kog se vrši učenje i zatim prilagoditi sve prezentovane podatke i korisnički interfejs tom jeziku. Potrebno je da se

vodi statistika o korisnikovom napretku i omogućiti poređenje ove statistike sa drugim korisnicima.

Kao vid motivacije obezbediti sistem za nagrađivanje korisnika skupljanjem poena i otključavanjem zanimljivosti o japanu i njegovoj kulturi. Takođe, obezbediti dodatnu podršku učenju složenih japanskih pisama u vidu jednostavne igre pamćenja gde će „kartice“ predstavljati japanski znaci i njihovi latinični zapisi koje onda treba povezati.

#### 4.1.2. Dijagram slučajeva korišćenja



Slika 5. Dijagram slučajeva korišćenja

Tabela 3. Slučajevi korišćenja

<b>Slučajevi korišćenja:</b>	
1. Inicijalno podešavanje aplikacije	12. Testiranje znanja
2. Pregled pojedinačne reči	13. Podešavanje i pokretanje testa znanja
3. Listanje i pretraga baze reči	14. Pregled profila korisnika
4. Prikaz pojedinačnog kana znaka	15. Registracija korisnika
5. Listanje kana znakova	16. Dodavanje prijatelja
6. Prikaz pojedinačnog kandi znaka	17. Izmena slike profila
7. Listanje kandi znakova	18. Izbor prikazanog prijatelja
8. Igranje igre pamćenja sa japanskim znacima	19. Podešavanje aplikacije
9. Vežbanje pisanja japanskih znakova	20. Pregled pomoći
10. Podešavanje aktivnosti za vežbanje pisanja	21. Pregledanje zanimljivosti
11. Pokretanje aktivnosti pisanja	22. Prelaženje lekcija

#### 4.1.3. Slučajevi korišćenja

<p><b>SK1: Slučaj korišćenja -</b></p> <p><b>Inicijalno podešavanje aplikacije</b></p> <p>Naziv: Inicijalno podešavanje aplikacije</p> <p>Svrha: Definisanje osnovnih parametara aplikacije</p> <p>Akteri: Korisnik</p> <p>Učesnici: Korisnik i sistem</p> <p>Preduslov: Aplikacija nije pokretana ranije</p> <p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>Korisnik pokreće aplikaciju po prvi put.</li> <li>Sistem prikazuje ekran za inicijalno podešavanje aplikacije i dobavlja/prikazuje listu dostupnih jezika preko interneta.</li> <li>Korisnik unosi svoje ime, bira željeni jezik od ponuđenih, željenu težinu i obim lekcija koje će mu biti dostupne kroz aplikaciju.</li> <li>Korisnik stiska dugme "OK".</li> <li>Sistem sa web servera skida odgovarajući komplet objekata učenja i instalira ih u lokalnu bazu podataka.</li> <li>Sistem prikazuje glavni ekran aplikacije.</li> </ol>	<p><b>SK2: Slučaj korišćenja -</b></p> <p><b>Pregled pojedinačne reči</b></p> <p>Naziv: Pregled pojedinačne reči</p> <p>Svrha: Pregled reči japanskog jezika</p> <p>Akteri: Korisnik</p> <p>Učesnici: Korisnik i sistem</p> <p>Preduslov: Startovana aktivnost za prikaz reči</p> <p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>Korisnik koristi dugmiće za kretanje napred/nazad po spisku dostupnih reči da prikaže reč čije detalje želi da vidi.</li> <li>Sistem prikazuje izabrano reč napisanu različitim japanskim pismima, njen prevod i dugme za puštanje audio zapisa reči.</li> <li>Korisnik klikće na dugme „Play audio“ (pusti audio zapis).</li> <li>Sistem pušta audio zapis izgovaranja date reči.</li> </ol> <p><b>Alternativni scenario:</b></p> <ol style="list-style-type: none"> <li>Ukoliko je iz nekog razloga nemoguće</li> </ol>
--	--

<p><b>Alternativni scenario:</b></p> <p>2.1. Ukoliko postoji problem sa konekcijom, sistem prikazuje poruku o grešci i korisniku nudi ponavljanje koraka br. 2 ili odustajanje.</p> <p>6.1. Ukoliko postoji problem sa konekcijom, sistem prikazuje poruku o grešci i korisniku nudi ponavljanje koraka br. 6 ili odustajanje.</p>	<p>uspostaviti konekciju sa internetom, sistem prikazuje podatke o reči, a skriva dugme za puštanje audio zapisa.</p>
<p><b>SK3: Slučaj korišćenja - Listanje i pretraga baze reči</b></p> <p>Naziv: Listanje i pretraga baze reči</p> <p>Svrha: Nalaženje željenih reči radi prikaza</p> <p>Akteri: Korisnik</p> <p>Učesnici: Korisnik i sistem</p> <p>Preduslov: /</p> <p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>Korisnik bira tab „Words“ (reči).</li> <li>Sistem prikazuje listu svih reči u bazi podataka.</li> <li>Korisnik unosi traženi tekst u predviđeno polje.</li> <li>Sistem sužava listu da prikazuje samo reči koje sadrže uneti tekst u sebi ili u svom prevodu na japanski.</li> <li>Korisnik bira željenu reč.</li> <li>Sistem obezbeđuje pregled pojedinačne reči – <b>include</b> „Pregled pojedinačne reči“.</li> </ol> <p><b>Alternativni scenario:</b></p> <ol style="list-style-type: none"> <li>Ukoliko nijedna reč ne odgovara traženom tekstu, umesto rezultata se ispisuje poruka da nije nađena nijedna reč.</li> </ol>	<p><b>SK4: Slučaj korišćenja - Prikaz pojedinačnog kana znaka</b></p> <p>Naziv: Prikaz pojedinačnog kana znaka</p> <p>Svrha: Pregled kana znakova</p> <p>Akteri: Korisnik</p> <p>Učesnici: Korisnik i sistem</p> <p>Preduslov: Startovana aktivnost za prikaz kana znakova</p> <p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>Korisnik koristi dugmiće za kretanje napred/nazad po spisku dostupnih znakova da prikaže znak čije detalje želi da vidi.</li> <li>Sistem prikazuje izabran znak, redosled povlačenja linija, njegov latinični zapis i reč – primer iz baze podataka.</li> <li>Korisnik klikće na dugme „Another example“ (još jedan primer)</li> <li>Sistem prikazuje dodatne reči – primere iz baze podataka</li> </ol> <p><b>Alternativni scenario:</b></p> <ol style="list-style-type: none"> <li>Ukoliko ne postoji reč – primer u bazi podataka, neće se prikazati korisniku, kao ni dugme „Another example“.</li> </ol>

<p><b>SK5: Slučaj korišćenja -</b>  <b>Listanje kana znakova</b></p> <p>Naziv: Listanje kana znakova  Svrha: Nalaženje željenih kana znakova radi prikaza  Akteri: Korisnik  Učesnici: Korisnik i sistem  Preduslov: /</p> <p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>1. Korisnik bira tab „Writing“ (pisanje).</li> <li>2. Sistem prikazuje aktivnosti vezane za vežbanje pisama.</li> <li>3. Korisnik bira aktivnost "Browse Kana" (pretraživanje kana znakova).</li> <li>4. Sistem prikazuje aktivnost za pretraživanje kana znakova.</li> <li>5. Korisnik koristi dugmiće za kretanje napred/nazad da prikaže željenu tabelu.</li> <li>6. Sistem prikazuje odgovarajuću tabelu sa kana znacima.</li> <li>7. Korisnik bira željeni znak iz tabele.</li> <li>8. Sistem obezbeđuje pregled izabranog znaka – <b>include</b> „Prikaz pojedinačnog kana znaka“.</li> </ol>	<p><b>SK6: Slučaj korišćenja -</b>  <b>Prikaz pojedinačnog kandži znaka</b></p> <p>Naziv: Prikaz pojedinačnog kandži znaka  Svrha: Pregled kandži znakova  Akteri: Korisnik  Učesnici: Korisnik i sistem  Preduslov: Startovana aktivnost za prikaz kandži znakova</p> <p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>1. Korisnik koristi dugmiće za kretanje napred/nazad po spisku dostupnih znakova da prikaže znak čije detalje želi da vidi.</li> <li>2. Sistem prikazuje izabran znak, redosled povlačenja linija, njegov latinični zapis, značenje, čitanja ispisana različitim pismima i reč – primer iz baze podataka.</li> <li>3. Korisnik klikće na dugme „Another example“ (još jedan primer).</li> <li>4. Sistem prikazuje dodatne reči – primere iz baze podataka.</li> </ol> <p><b>Alternativni scenario:</b></p> <ol style="list-style-type: none"> <li>2.1. Ukoliko ne postoji reč – primer u bazi podataka, neće se prikazati korisniku, kao ni dugme „Another example“.</li> </ol>
<p><b>SK7: Slučaj korišćenja -</b>  <b>Listanje kandži znakova</b></p> <p>Naziv: Listanje kana znakova  Svrha: Nalaženje željenih kandži znakova radi prikaza  Akteri: Korisnik  Učesnici: Korisnik i sistem  Preduslov: /</p>	<p><b>SK8: Slučaj korišćenja -</b>  <b>Igranje igre pamćenja sa japanskim znacima</b></p> <p>Naziv: Igranje igre pamćenja sa japanskim znacima  Svrha: Učenje znakova kroz igru memorije  Akteri: Korisnik  Učesnici: Korisnik i sistem  Preduslov: /</p>

<p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>1. Korisnik bira tab „Writing“ (pisanje).</li> <li>2. Sistem prikazuje aktivnosti vezane za vežbanje pisama.</li> <li>3. Korisnik bira aktivnost “Browse Kanji” (pretraživanje kandži znakova).</li> <li>4. Sistem prikazuje aktivnost za pretraživanje kandži znakova.</li> <li>5. Korisnik koristi dugmiće za kretanje napred/nazad da prikaže željenu tabelu.</li> <li>6. Sistem prikazuje odgovarajuću tabelu sa kana znacima.</li> <li>7. Korisnik bira željeni znak iz tabele.</li> <li>8. Sistem obezbeđuje pregled izabranog znaka – <b>include</b> „Prikaz pojedinačnog kandži znaka“.</li> </ol>	<p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>1. Korisnik bira tab „ Writing“ (pisanje).</li> <li>2. Sistem prikazuje aktivnosti vezane za vežbanje pisama.</li> <li>3. Korisnik bira aktivnost “Memory Game” (igra pamćenja).</li> <li>4. Sistem prikazuje ekran sa podešavanjima nove igre</li> <li>5. Korisnik bira veličinu table i skup znakova koji će biti korišćeni u igri i stiska dugme „Start“.</li> <li>6. Sistem pokreće igru pamćenja sa unetim podešavanjima i prikazuje tablu.</li> <li>7. Korisnik klikće na parove kartica dok ne poveže sve parove na tabli.</li> <li>8. Sistem prikazuje koliko je korisnik osvojio poena.</li> <li>9. Korisnik pritiska dugme „OK“.</li> <li>10. Sistem prikazuje korisniku tab „Writing“.</li> </ol>
<p><b>SK9: Slučaj korišćenja - Vežbanje pisanja japanskih znakova</b></p> <p>Naziv: Vežbanje pisanja japanskih znakova</p> <p>Svrha: Vežbanje pisanja japanskih znakova</p> <p>Akteri: Korisnik</p> <p>Učesnici: Korisnik i sistem</p> <p>Preduslov: Pokrenuta aktivnost za vežbanje pisanja japanskih znakova</p> <p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>1. Sistem prikazuje ekran za vežbanje pisanja znakova.</li> <li>2. Korisnik podešava prikaz vodilja uz pomoć obezbeđenih kontrola.</li> <li>3. Sistem prikazuje vodilje za izabrani znak</li> <li>4. Korisnik po želji uključuje „fade“ efekat i</li> </ol>	<p><b>SK10: Slučaj korišćenja - Podešavanje aktivnosti za vežbanje pisanja</b></p> <p>Naziv: Podešavanje aktivnosti za vežbanje pisanja</p> <p>Svrha: Podešavanje aktivnosti</p> <p>Akteri: Korisnik</p> <p>Učesnici: Korisnik i sistem</p> <p>Preduslov: Pokrenuta aktivnost za vežbanje pisanja japanskih znakova</p> <p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>1. Korisnik klikće na dugme sa podešavanjima aktivnosti za vežbanje pisanja</li> <li>2. Sistem prikazuje tri dugmeta sa različitim kategorijama podešavanja.</li> </ol>

<p>minimizira interfejs.</p> <p>5. Sistem primjenjuje podešavanja.</p> <p>6. Korisnik crta prstom po ekranu.</p>	<p>3. Korisnik bira željenu kategoriju podešavanja (pisanje, interfejs, vodilje).</p> <p>4. Sistem prikazuje odgovarajući dijalog sa podešavanjima.</p> <p>5. Korisnik menja podešavanja i pritiska dugme „OK”.</p> <p>6. Sistem primjenjuje podešavanja i prikazuje aktivnost za vežbanje pisanja.</p>
<p><b>SK11: Slučaj korišćenja - Pokretanje aktivnosti pisanja</b></p> <p>Naziv: Pokretanje aktivnosti pisanja</p> <p>Svrha: Pokretanje aktivnosti pisanja</p> <p>Akteri: Korisnik</p> <p>Učesnici: Korisnik i sistem</p> <p>Preduslov: /</p> <p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>Korisnik bira tab „Writing“ (pisanje).</li> <li>Sistem prikazuje aktivnosti vezane za vežbanje pisama.</li> <li>Korisnik bira aktivnost "Writing pad".</li> <li>Sistem pokreće aktivnost za vežbanje pisanja – <b>include</b> „Vežbanje pisanja japanskih znakova“</li> </ol>	<p><b>SK12: Slučaj korišćenja – Testiranje znanja</b></p> <p>Naziv: Testiranje znanja</p> <p>Svrha: Testiranje znanja</p> <p>Akteri: Korisnik</p> <p>Učesnici: Korisnik i sistem</p> <p>Preduslov: Startovana aktivnost za testiranje znanja</p> <p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>Sistem prikazuje pitanje, ponuđene odgovore, broj poena, broj tačnih i netačnih odgovora korisniku.</li> <li>Korisnik daje odgovor na pitanje</li> <li>Koraci 1 i 2 se ponavljaju dokle god ima pitanja.</li> <li>Sistem prikazuje statistike korisniku.</li> <li>Korisnik pristiska „OK“.</li> <li>Sistem vraća korisnika na prethodnu aktivnost.</li> </ol>
<p><b>SK13: Slučaj korišćenja - Podešavanje i pokretanje testa znanja</b></p> <p>Naziv: Podešavanje i pokretanje testa znanja</p> <p>Svrha: Testiranje znanja na slobodno izabranom setu pitanja</p> <p>Akteri: Korisnik</p>	<p><b>SK14: Slučaj korišćenja - Pregled profila korisnika</b></p> <p>Naziv: Pregled profila korisnika</p> <p>Svrha: Pregled sopstvenog napretka i poređenje sa tuđim</p> <p>Akteri: Korisnik</p>

<p>Učesnici: Korisnik i sistem Preduslov: /</p> <p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>1. Korisnik bira tab „Tests“ (testovi).</li> <li>2. Sistem prikazuje dostupne vrste testova (random word, random kana, random kanji, worst word, worst kana, worst kanji).</li> <li>3. Korisnik bira jednu od vrsta testova.</li> <li>4. Sistem prikazuje ekran sa podešavanjima testa (broj pitanja, broj odgovora po pitanju, skup iz kog će se vući pitanja).</li> <li>5. Korisnik po želji menja podešavanja i pritiska dugme „Start“.</li> <li>6. Sistem pokreće testiranje znanja sa unetim podešavanjima – <b>include</b> „Testiranje znanja“.</li> </ol>	<p>Učesnici: Korisnik i sistem Preduslov: /</p> <p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>1. Korisnik bira tab „Profile“.</li> <li>2. Sistem prikazuje statistike korisnika i njegovog izabranog prijatelja.</li> </ol>
<p><b>SK15: Slučaj korišćenja - Registracija korisnika</b></p> <p>Naziv: Registracija korisnika Svrha: Registracija korisnika Akteri: Korisnik Učesnici: Korisnik i sistem Preduslov: Prikazan profil korisnika, dozvoljeno korišćenje interneta u podešavanjima</p> <p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>1. Korisnik pritiska dugme "Menu" na telefonu.</li> <li>2. Sistem prikazuje meni.</li> <li>3. Korisnik bira stavku "Register".</li> <li>4. Sistem prikazuje dijalog za registraciju.</li> <li>5. Korisnik unosi željeno korisničko ime i pritiska dugme "Register".</li> </ol>	<p><b>SK16: Slučaj korišćenja - Dodavanje prijatelja</b></p> <p>Naziv: Dodavanje prijatelja Svrha: Dodavanje prijatelja sa kojima je moguće poređiti statistike Akteri: Korisnik Učesnici: Korisnik i sistem Preduslov: Prikazan profil korisnika, dozvoljeno korišćenje interneta u podešavanjima</p> <p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>1. Korisnik pritiska dugme "Menu" na telefonu.</li> <li>2. Sistem prikazuje meni.</li> <li>3. Korisnik bira stavku "Add Friends" (dodaj prijatelje).</li> <li>4. Sistem prikazuje ekran za dodavanje</li> </ol>

<p>6. Sistem kontaktira server i validira uneto korisničko ime.</p> <p>7. Sistem zatvara dijalog, ispisuje poruku o uspešnoj registraciji.</p> <p>8. Korisnik pritiska dugme "OK".</p> <p>9. Sistem vraća korisnika na pregled profila i vrši update korisničkih podataka sa serverom.</p> <p><b>Alternativni scenario:</b></p> <p>7.1. Ukoliko postoji problem sa konekcijom, ispisuje se poruka o tome i zaustavlja se dalje izvršenje scenarija.</p> <p>7.2. Ukoliko je uneto korisničko ime zauzeto, ispisuje se poruka o tome i od korisnika zahteva ponavljanje koraka 5.</p>	<p>prijatelja.</p> <p>5. Korisnik unosi tekst po kom će se vršiti pretraga korisnika aplikacije i pritiska dugme "Search".</p> <p>6. Sistem kontaktira server, dobavlja listu svih korisnika čija se imena poklapaju sa unetim tekstrom i prikazuje je korisniku.</p> <p>7. Korisnik bira jednog od ponuđenih korisnika.</p> <p>8. Sistem zatvara ekran za dodavanje prijatelja, dodaje prijatelja u bazu i korisniku prikazuje pregled profila sa uporedno prikazanim statistikama izabranog prijatelja</p> <p><b>Alternativni scenario:</b></p> <p>6.1. Ukoliko postoji problem sa konekcijom, ispisuje se poruka o tome i zaustavlja se dalje izvršenje scenarija.</p> <p>6.2. Ukoliko nema korisnika čije se ime poklapa sa unetim tekstrom, ispisuje se poruka o tome.</p>
<p><b>SK17: Slučaj korišćenja - Izmena slike profila</b></p> <p>Naziv: Izmena slike profila</p> <p>Svrha: Izmena slike profila koja je vidljiva drugim korisnicima</p> <p>Akteri: Korisnik</p> <p>Učesnici: Korisnik i sistem</p> <p>Preduslov: Prikazan profil korisnika</p> <p><b>Osnovni scenario:</b></p> <p>1. Korisnik pritiska dugme "Menu" na telefonu.</p>	<p><b>SK18: Slučaj korišćenja - Izbor prikazanog prijatelja</b></p> <p>Naziv: Izbor prikazanog prijatelja</p> <p>Svrha: Izbor prijatelja čije će statistike biti prikazane uporedno sa korisnikovim.</p> <p>Akteri: Korisnik</p> <p>Učesnici: Korisnik i sistem</p> <p>Preduslov: Prikazan profil korisnika, korisnik ima bar jednog prijatelja</p> <p><b>Osnovni scenario:</b></p> <p>1. Korisnik klikće na ime prijatelja čije su</p>

<p>2. Sistem prikazuje meni.</p> <p>3. Korisnik bira stavku "Change profile image" (izmeni sliku profila).</p> <p>4. Sistem prikazuje dijalog za izmenu slike profila.</p> <p>5. Korisnik klikće na dugme "Load".</p> <p>6. Sistem otvara prikaz galerije svih slika na telefonu</p> <p>7. Korisnik bira jednu od slika.</p> <p>8. Sistem korisniku prikazuje prethodni dijalog.</p> <p>9. Korisnik klikće na sliku koju želi da mu bude slika profila (staru ili novu).</p> <p>10. Sistem zatvara dijalog i namešta izabranu sliku profila.</p> <p><b>Alternativni scenario:</b></p> <p>5.1. Korisnik klikće na dugme "Clear" čime samo uklanja postojeću sliku i vraća nazad default sliku za profil.</p>	<p>statistike prikazane pored njegovih</p> <p>2. Sistem prikazuje dijalog za izbor prijatelja.</p> <p>3. Korisnik bira jednog od prijatelja sa spiska.</p> <p>4. Sistem zatvara dijalog i vraća korisnika na pregled profila, gde će sada biti prikazane statistike izabranog prijatelja.</p>
<p><b>SK19: Slučaj korišćenja - Podešavanje aplikacije</b></p> <p>Naziv: Podešavanje aplikacije</p> <p>Svrha: Podešavanje aplikacije</p> <p>Akteri: Korisnik</p> <p>Učesnici: Korisnik i sistem</p> <p>Preduslov: Prikazan jedan od osnovnih tabova aplikacije (Learn, Words, Writing, Tests, Profile, Trivia)</p> <p><b>Osnovni scenario:</b></p> <p>1. Korisnik pritiska dugme "Menu" na telefonu.</p> <p>2. Sistem prikazuje meni.</p> <p>3. Korisnik bira stavku "Settings" (podešavanja).</p>	<p><b>SK20: Slučaj korišćenja - Pregled pomoći</b></p> <p>Naziv: Pregled pomoći</p> <p>Svrha: Pregled pomoći</p> <p>Akteri: Korisnik</p> <p>Učesnici: Korisnik i sistem</p> <p>Preduslov: Prikazan jedan od osnovnih tabova aplikacije (Learn, Words, Writing, Tests, Profile, Trivia)</p> <p><b>Osnovni scenario:</b></p> <p>1. Korisnik pritiska dugme "Menu" na telefonu.</p> <p>2. Sistem prikazuje meni.</p> <p>3. Korisnik bira stavku "Help" (pomoć).</p> <p>4. Sistem prikazuje sve kategorije help-a.</p>

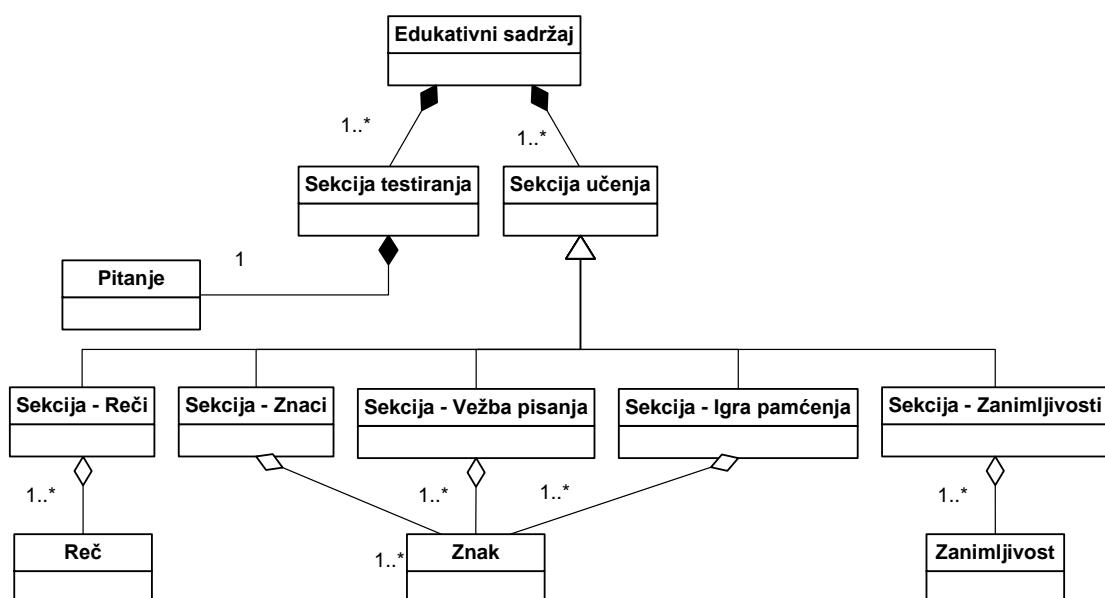
<p>4. Sistem prikazuje sva podešavanja aplikacije.</p> <p>5. Korisnik menja podešavanja po želji.</p>	<p>5. Korisnik bira jednu od kategorija.</p> <p>6. Sistem prikazuje dijalog sa tekstom izabrane help teme.</p> <p>7. Korisnik lista stranice teme korišćenjem dugmića za kretanje napred/nazad.</p>
<p><b>SK21: Slučaj korišćenja – Pregledanje zanimljivosti</b></p> <p>Naziv: Pregledanje zanimljivosti</p> <p>Svrha: Pregledanje zanimljivosti</p> <p>Akteri: Korisnik</p> <p>Učesnici: Korisnik i sistem</p> <p>Preduslov: korisniku ima otključanu bar jednu zanimljivost</p> <p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>Korisnik bira tab „Trivia“ (zanimljivosti).</li> <li>Sistem prikazuje aktivnosti za pregled zanimljivosti</li> <li>Korisnik bira jednu od kategorija zanimljivosti</li> <li>Sistem prikazuje sve zanimljivosti u izabranoj kategoriji</li> <li>Korisnik bira jednu od zanimljivosti.</li> <li>Sistem prikazuje sadržaj zanimljivosti.</li> </ol>	<p><b>SK22: Slučaj korišćenja - Testiranje znanja</b></p> <p>Naziv: Testiranje znanja</p> <p>Svrha: Testiranje znanja</p> <p>Akteri: Korisnik</p> <p>Učesnici: Korisnik i sistem</p> <p>Preduslov: /</p> <p><b>Osnovni scenario:</b></p> <ol style="list-style-type: none"> <li>Korisnik bira tab „Learn“ (učenje).</li> <li>Sistem prikazuje sve dostupne lekcije.</li> <li>Korisnik bira jednu od lekcija.</li> <li>Sistem prikazuje odeljke sadržane u lekciji.</li> <li>Korisnik bira odeljak</li> <li>Sistem prikazuje odgovarajuću aktivnost – <b>include</b> „Pregled pojedinačne reči“, „Prikaz pojedinačnog kana znaka“, „Prikaz pojedinačnog kandi znaka“, Vežbanje pisanja japanskih znakova“, „Testiranje znanja“.</li> <li>Kada korisnik pređe sve odeljke, sistem prikazuje dugme “Test your knowledge”.</li> <li>Korisnik klikće na dugme.</li> <li>Sistem prikazuje test korisniku – <b>include</b> „Testiranje znanja“.</li> </ol>

#### 4.1.4. Model objekata učenja

Edukativni sadržaji koji se čuvaju u bazi podataka su gradivni delovi objekata učenja. Centralni proces učenja se zasniva na prezentovanju sadržaja objekata učenja, a sve pojedinačne aktivnosti učenja odgovaraju pojedinačnim vrstama sadržaja koji mogu biti

delovi objekata učenja. Objekti učenja se sastoje iz konkretnog edukativnog sadržaja (Sl. 6) koji po obimu i vrsti varira od objekta do objekta i metapodataka koji opisuju taj sadržaj (Sl. 7).

Edukativni sadržaj, u skladu sa definicijama objekata učenja, obuhvata sekcije učenja i sekcije provera znanja. Sekcije učenja dalje mogu biti sekcije za učenje reči, znakova, za vežbanje pisanja, igra pamćenja ili prezentacija zanimljivosti. Svaka od ovih sekcijskih ima drugaćiju prezentaciju u samoj aplikaciji. Jedan objekat učenja može da se sastoji iz različitog broja/kombinacija sekcijskih - od jednostavnih objekata koji sadrže samo jednu sekciju do složenih objekata sa više ili čak svim sekcijskim u jednom objektu.



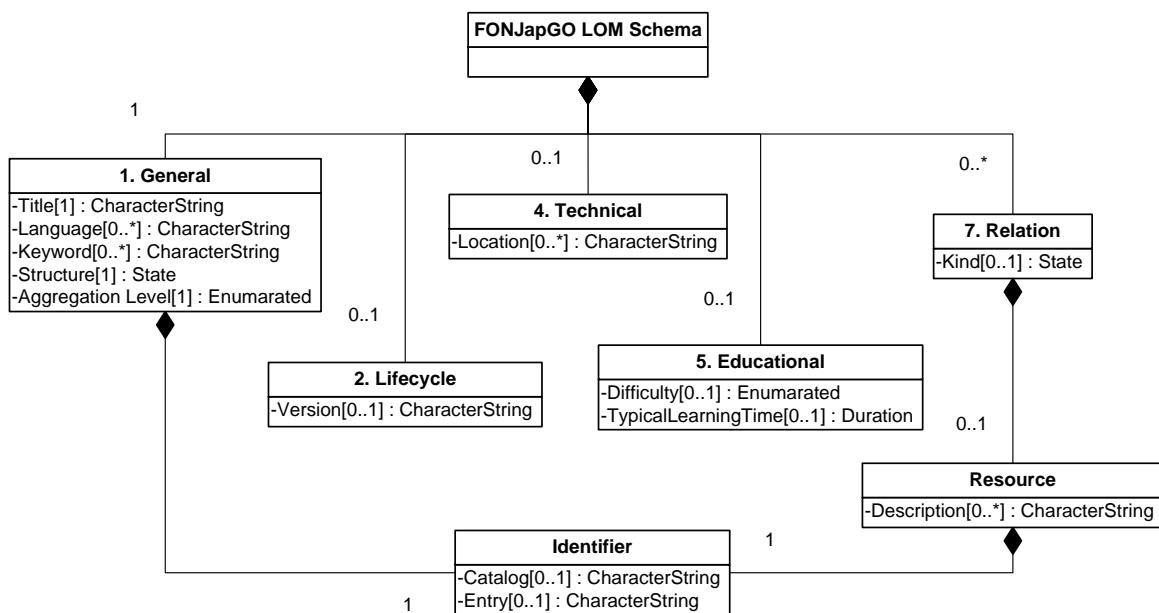
Slika 6. Model edukativnog sadržaja objekata učenja

Metapodaci su bitan deo objekata učenja jer omogućavaju skladištenje i pretraživanje objekata iz nekog repozitorijuma po nekom kriterijumu, agregiranje manjih objekata u veće, ponovnu upotrebu itd. Model metapodataka je dobijen kreiranjem aplikacionog profila postojećeg IEEE LOM-a. Osnovni model je dosta uprošćen za potrebe JapGO aplikacije i iz njega su izbačene komponente 3.Meta-Metadata, 6.Rights, 8.Annotation i 9.Classification.

Elementi koji su uključeni su 1.General koji predstavlja opšte podatke - ime i identifikaciju objekta, jezik na kom je njegov sadržaj, ključne reči koje mogu koristiti u eventualnoj pretrazi, ulogu objekta u agregaciji i nivo agregacije na kom se nalazi. Dalje, 2.Lifecycle evidentira verziju objekta, a 4.Technical njegovu adresu. Tehnički uslovi su odbačeni kao nepotrebni zbog relativne homogenosti sadržaja koji su predstavljeni u tekstualnom formatu

ili kao linkovi ka dodatnim sadržajima. 5.Educational element je takođe dosta uprošćen; zadržane su samo informacije o težini objekta i tipičnom vremenu potrebnom za savladavanje. Na kraju, 7.Relation je zadržan zbog uloge koju ima u određivanju odnosa agregacije objekata time što dozvoljava referenciranje sastavnih objekata i/ili objekata čiji je tekući objekat deo; takođe, može se koristiti za naznačavanje objekata sa edukativnim sadržajem koji je preduslov za upotrebu tekućeg objekta. Pored svega navedenog, načinjena je još jedna izmena u skladu sa kontekstom korišćenja objekata.

Pošto će objekti sadržati strogo jezički sadržaj (npr. engleski), svi Langstring atributi, koji se sastoje iz kombinacije teksta i oznake jezika na kom je taj tekst, su zamenjeni običnim CharacterString-ovima. Podrazumeva se da se i u podacima i u metapodacima objekata koristi jezik naznačen u Language atributu General elementa i da objekat na jednom jeziku neće biti zanimljiv učenicima koji govore drugi jezik; za njih će potencijalno postojati drugi komplet objekata na njihovom jeziku.



Slika 7. Model metapodataka objekta učenja

#### 4.1.5. Model podataka – PMOV, relacioni model

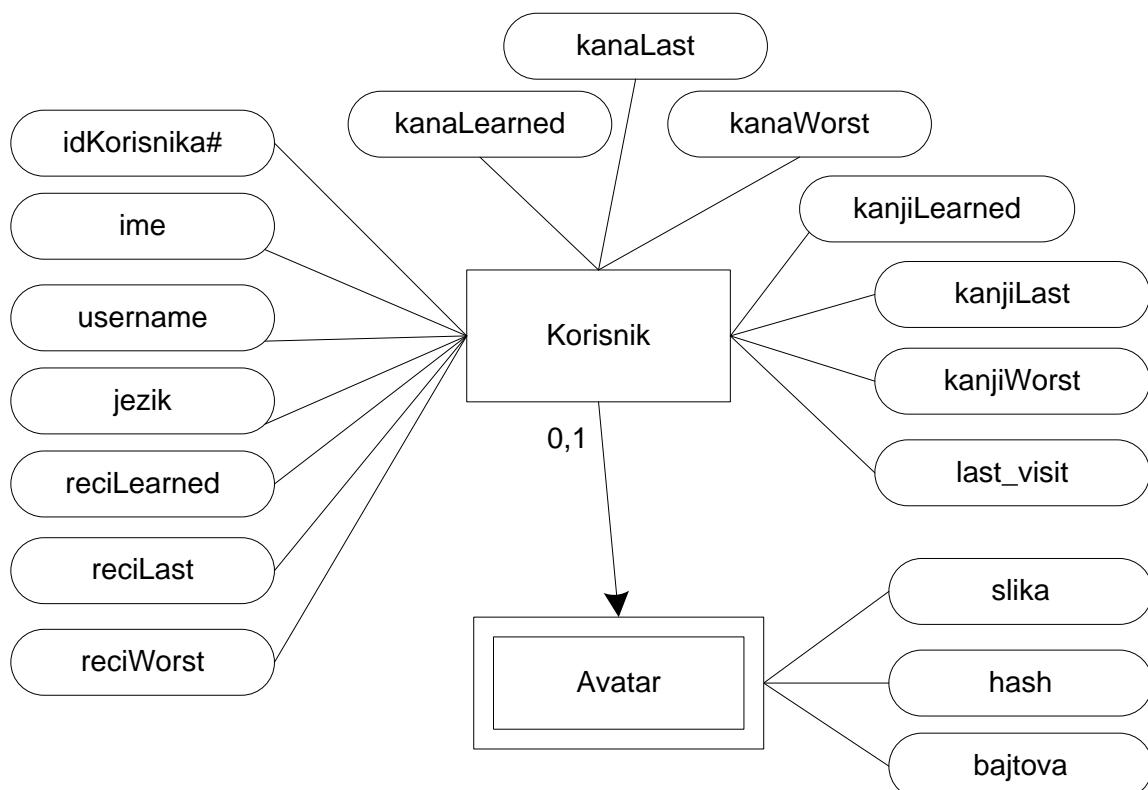
##### 4.1.5.1. Model podataka – web aplikacija

Podaci koji se čuvaju u onlajn skladištu kojem se pristupa uz pomoć veb aplikacije obuhvataju podatke o korisničkim statistikama i njihovim avatarima. Iako su u ovom radu i objekti učenja uskladišteni i indeksirani na istom serveru, čuvaju se u odvojenoj bazi

podataka. Podaci predstavljeni na slici 8 su podaci konkretnе (JapGO) aplikacije dok su objekti učenja zamišljeni tako da budu univerzalno dostupni bilo kojoj aplikaciji za učenje japanskog jezika i samim time nisu ni na koji način zavisni od aplikacije.

Podaci o korisnicima obuhvataju njihov id, ime i korisničko ime (username), jezik na kojem obavljaju učenje (srpski, engleski, itd...), statistike dobavljene iz aplikacije i vreme zadnjeg apdejta ovih podataka. Ovi podaci treba da budu dostupni drugim korisnicima radi poređenja svojih rezultata sa drugima.

Korisnici mogu da postave željenu sliku u svoj profil (avatara). Ova slika će biti prikazivana prilikom pretrage i izbora korisnika. Podaci koji se čuvaju o sliki su putanja do slike (koja će takođe biti uskladištena negde na serveru, samo u obliku klasičnog fajla), hash i veličina slike (koji će se koristiti za efikasno poređenje slika kako bi se utvrdilo da li je došlo do promene i da li je potrebno da korisnikova aplikacija uploaduje novu sliku na server).



Slika 8. PMOV model podataka korišćen u web aplikaciji

#### 4.1.5.2. Relacioni model – web aplikacija

**Korisnik** (idKorisnika, ime, username, jezik, reciLearned, reciLast, reciWorst, kanaLearned, kanaLast, kanaWorst, kanjiLearned, kanjiLast, kanjiWorst, last\_visit)

**Avatar** (idKorisnika, slika, hash, bajtova)

Tabela Korisnik		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	
Atributi	<u>idKorisnika</u>	Integer	not null			INSERT / UPDATE CASCADES Avatar  DELETE CASCADES Avatar
	ime	String	not null			
	username	String	not null			
	jezik	String	not null			
	reciLearned	Integer	(default: 0)			
	reciLast	String				
	reciWorst	String				
	kanaLearned	Integer	(default: 0)			
	kanaLast	String				
	kanaWorst	String				
	kanjiLearned	Integer	(default: 0)			
	kanjiLast	String				
	kanjiWorst	String				
	last_visit	Integer	(default: 0)			

Tabela Avatar		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	
Atributi	<u>idKorisnika</u>	Integer	not null			INSERT RESTRICTED korisnik  UPDATE RESTRICTED korisnik  DELETE /
	slika	String				
	hash	String				
	bajtova	String	(default: 0)			

#### 4.1.5.3. Model podataka – Android aplikacija

Centralni objekat modela je „learn\_obj“ koji predstavlja pojedinačne objekte učenja; u atributima ovih objekata čuvaće se svi njihovi metapodaci. Edukativni sadržaji koje objekti učenja nose u sebi obuhvataju (ako posmatramo najmanje jedinice) reči japanskog, kana znake i kandži znake. Takođe, svaki objekat učenja će imati po jedan kviz kojim će se testirati poznavanje svih sadržanih elemenata. Samim tim, u modelu su projektovani objekti Japanski, Kana i Kanji i koji će sadržati odgovarajuće reči i znake i agregirani objekti learn\_obj\_reci, learn\_obj\_kana i learn\_obj\_kanji koji povezuju objekat učenja sa njegovim sadržajem i

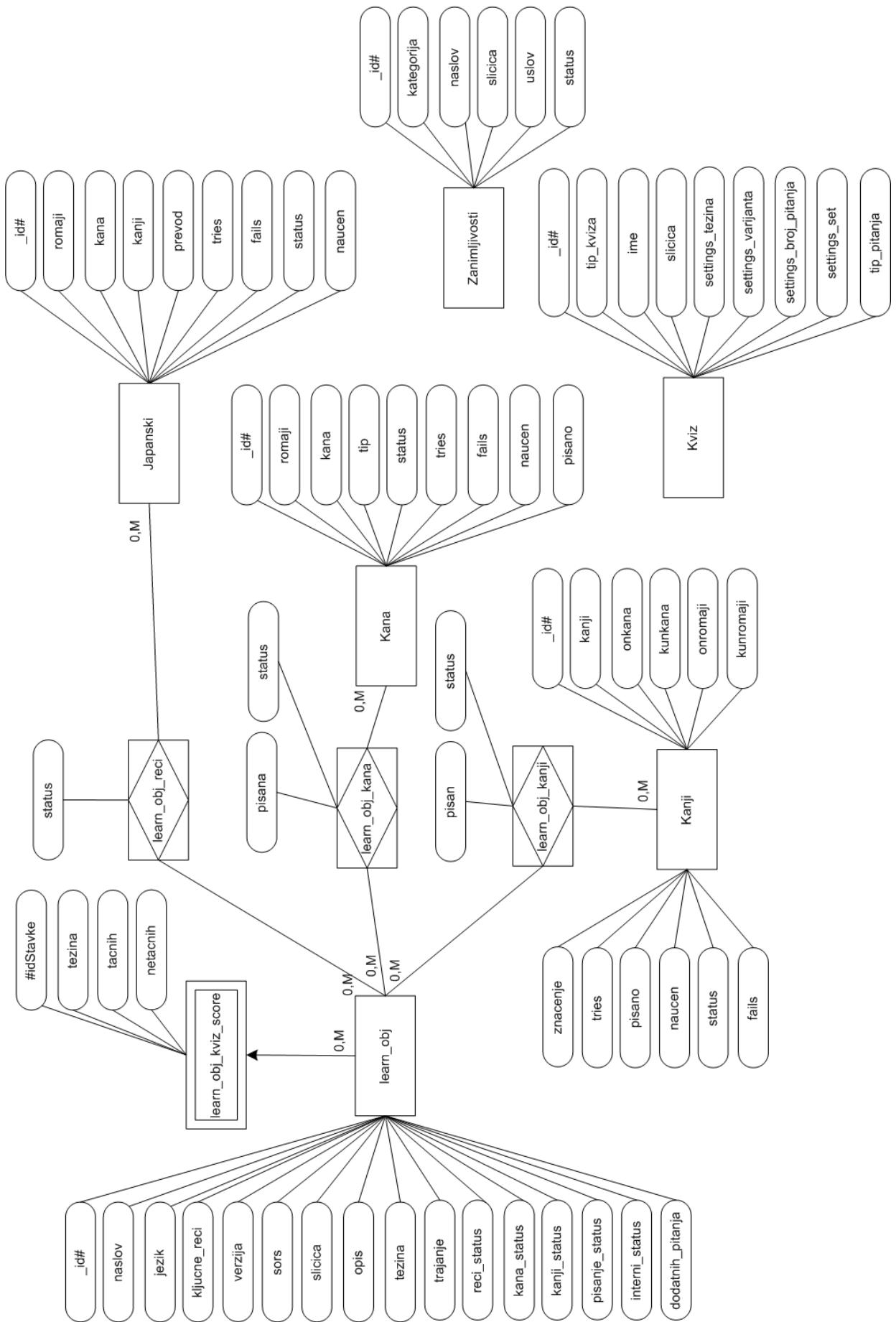
definišu status svakog od elemenata. Na ovaj način je moguće da se svaki element (reč, znak) nađe u više objekata.

Objekat „Japanski“ je namenjen čuvanju informacija o rečima japanskog jezika i sadži atributе kao što su latinični zapis reči (romaji), kana zapis reči (kana), kandži zapis reči, (kanji) i prevod. Takođe, određene informacije se evidentiraju za svaku reč ponaosob – koliko puta je neka reč prikazana korisniku u kvizovima (tries), koliko puta je pogrešno odgovorio na pitanje (fails) i status i vreme kada je reč naučena (status, naucen).

Objekti „Kana“ i „Kanji“ su dosta slični objektu „Japanski“ i još sličniji međusobno. Ovi objekti čuvaju kana i kandži znakove; nalik na objekte u kojima se čuvaju reči, i ovde se vodi evidencija za svaki znak (tries, fails, status, naucen). Osim toga, evidentiraju se odgovarajući elementi – latinični i japanski zapis znakova i tip (hiragana/katakana) kod „Kana“ objekata, on/kun latinični i on/kun kana zapisi, prevod i sam kandži zapis kod „Kanji“ objekata.

Informacije o kvizovima koji se mogu samostalno pokrenuti se čuvaju u „Kviz“ objektu i obuhvataju osnovne karakteristike kviza (ime, sliku) i njegova podešavanja – težinu, veličinu, skup pitanja. Objekti učenja imaju svoje kvizove i za njih postoji poseban objekat, „learn\_obj\_kviz\_score“ koji služi samo za evidenciju uspeha na njima. Druge informacije kao što su težina, veličina i skup pitanja su predodređene na osnovu samog objekta učenja (njegove težine i njegovog sadržaja) pa se te informacije ne čuvaju nigde specifično.

Na kraju, nezavisno od objekata učenja, u objektu „Zanimljivosti“ se čuvaju metapodaci koji opisuju pojedinačne zanimljivosti; konkretni sadržaj zanimljivosti će, zbog veličine, biti čuvan u obliku datoteka u okviru fajl sistema. Podaci obuhvataju kategoriju zanimljivosti, njen naslov, sličicu, uslov koji treba da je ispunjen da bi mogla da se otključa i status (zaključana/otključana/pogledana).



Slika 9. PMOV model podataka korišćen u Android aplikaciji

#### 4.1.5.4. Relacioni model – Android aplikacija

**Japanski** (idReci, romaji, kana, kanji, prevod, tries, fails, stauts, naucen)

**Kana**(idKane, romaji, kana, tip, tries, fails, stauts, naucen, pisano)

**Kanji**(idKanjija, kanji, onKana, kunKana, onRomaji, kunRomaji, znacenje, tries, fails, stauts, naucen, pisano)

**Kviz**(idKviza, tip\_kviza, ime, slicica, settings\_tezina, settings\_varijanta, settings\_broj\_pitanja, settings\_set, tip\_pitanja)

**Zanimljivost**(idZanimljivosti, kategorija, naslov, uslov, slicica, status)

**learn\_obj** (idObjekta, naslov, jezik, kljucne\_reci, verzija, sors, slicica, opis, tezina, trajanje, reci\_status, kana\_status, kanji\_status, pisanje\_status, interni\_status, dodatnih\_pitanja)

**learn\_obj\_kviz\_score**(idObjekta, idStavke, tezina, tacnih, netacnih)

**learn\_obj\_reci**(idObjekta, idReci, status)

**learn\_obj\_kana**(idObjekta, idKane, status, pisana)

**learn\_obj\_kanji**(idObjekta, idKanjija, status, pisan)

Tabela Japanski		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	
Atributi	<u>idReci</u>	Integer	not null			INSERT / UPDATE CASCADES learn_obj_reci DELETE RESTRICTED learn_obj_reci
	<u>romaji</u>	String	not null			
	<u>kana</u>	String	not null			
	<u>kanji</u>	String				
	<u>prevod</u>	String	not null			
	<u>tries</u>	Integer	(default: 0)			
	<u>fails</u>	Integer	(default: 0)			
	<u>status</u>	Integer	(default: 0)			
	<u>naucen</u>	Integer	(default: 0)			

Tabela Kana		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	
Atributi	<u>idKane</u>	Integer	not null			INSERT / UPDATE CASCADES learn_obj_kana DELETE RESTRICTED learn_obj_kana
	<u>romaji</u>	String	not null			
	<u>kana</u>	String	not null			
	<u>tip</u>	Integer	(default: 0)			
	<u>pisano</u>	Integer	(default: 0)			
	<u>tries</u>	Integer	(default: 0)			
	<u>fails</u>	Integer	(default: 0)			
	<u>status</u>	Integer	(default: 0)			
	<u>naucen</u>	Integer	(default: 0)			

Tabela Kanji		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Struktурно ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabele	
Atributi	idKanjija	Integer	not null			INSERT / UPDATE CASCADES learn_obj_kanji DELETE RESTRICTED learn_obj_kanji
	kanji	String	not null			
	onKana	String				
	kunKana	String				
	onRomaji	String				
	kunRomaji	String				
	znacenje	String	not null			
	pisano	Integer	(default: 0)			
	tries	Integer	(default: 0)			
	fails	Integer	(default: 0)			

Tabela Kviz		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Struktурноogrаничење
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabele	
Atributi	idKviza	Integer	not null			INSERT / UPDATE / DELETE /
	tip_kviza	Integer	(default: 0)			
	ime	String	not null			
	slicica	String				
	settings_tezina	Integer	(default: 0)			
	settings_varijanta	Integer	(default: 0)			
	settings_broj_pitanja	Integer	(default: 0)			
	settings_set	Integer	(default: 0)			
	tip_pitanja	Integer	(default: 0)			

Tabela Zanimljivost		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Struktурно ogrаничење
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabele	
Atributi	idZanimljivosti	Integer	not null			INSERT / UPDATE / DELETE /
	kategorija	Integer	(default: 0)			
	naslov	String	not null			
	uslov	Integer	(default: -1)			
	slicica	String				
	status	Integer	(default: 0)			

Tabela learn_obj		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Struktурноogrаничење
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabele	
Atributi	idObjekta	Integer	not null			INSERT / UPDATE CASCADES learn_obj_kana, learn_obj_kanji, learn_obj_reci, learn_obj_kvizi_score
	naslov	String	not null			
	jezik	String	not null			
	kljucne_reci	String				

verzija	Integer	(default: 1)			DELETE CASCADES learn_obj_kana, learn_obj_kanji, learn_obj_reci, learn_obj_kvizi_score
sors	String				
slicica	String				
opis	String				
tezina	Integer				
trajanje	Integer				
reci_status	Integer	(default: 0)			
kana_status	Integer	(default: 0)			
pisanje_status	Integer	(default: 0)			
interni_status	Integer	(default: 0)			
dodatnih_pitanja	Integer	(default: 0)			

Tabela learn_obj_reci		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT RESTRICTED learn_obj, Japanski
	idObjekta	Integer	not null			UPDATE RESTRICTED learn_obj, Japanski
	idReci	Integer	not null			
	status	Integer	(default: 0)			DELETE /

Tabela learn_obj_kana		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT RESTRICTED learn_obj, Kana
	idObjekta	Integer	not null			UPDATE RESTRICTED learn_obj, Kana
	idKane	Integer	not null			
	pisana	Integer	(default: 0)			DELETE /
	status	Integer	(default: 0)			

Tabela learn_obj_kanji		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT RESTRICTED learn_obj, kanji
	idObjekta	Integer	not null			UPDATE RESTRICTED learn_obj, kanji
	idKanjija	Integer	not null			
	pisana	Integer	(default: 0)			DELETE /
	status	Integer	(default: 0)			

Tabela learn_obj_kviz_score		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Struktorno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	
	idObjekta	Integer	not null			INSERT RESTRICTED learn_obj
	idStavke	Integer	not null			UPDATE RESTRICTED learn_obj
	tezina	Integer	(default: 0)			
	tacnih	Integer	(default: 0)			
	netacnih	Integer	(default: 0)			DELETE /

## 4.2. Arhitektura aplikacije

FONJAPGO aplikacija se sastoji iz dva dela: aplikacije u užem smislu (Android aplikacije) i veb aplikacije koja služi za evidentiranje statistika korisnika i poređenje sa rezultatima drugih korisnika. Troslojna arhitektura aplikacije je prikazana na Sl. 10.

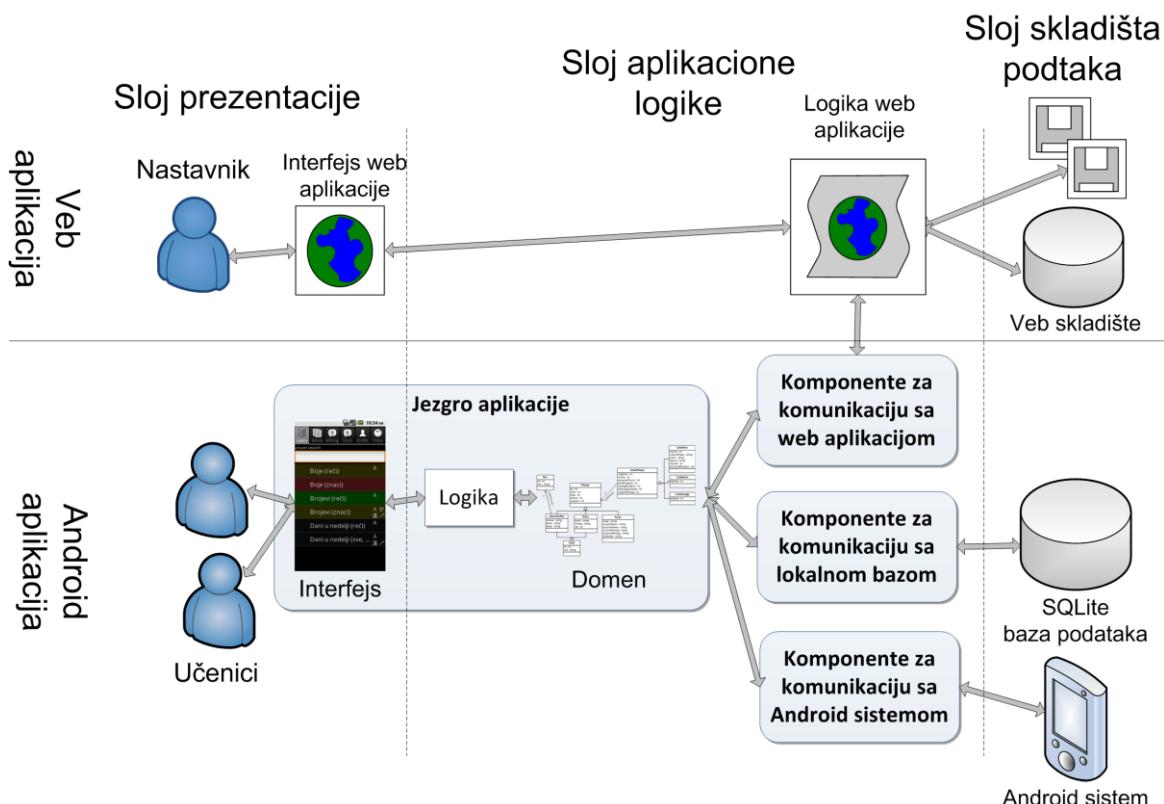
Prezentacioni sloj se zasniva na osnovnim elementima korisničkog interfejsa u okviru Android OS-a. Dinamički interfejs omogućava jednostavno korišćenje svih funkcija u okviru aplikacije. Budući da sam proces učenja ne zahteva nikakvu specifičnu obradu u domenu poslovne logike već se zasniva na kvalitetnoj prezentaciji jezičkih resursa i interaktivnosti, većina aplikacione logike je u tesnoj vezi sa prezentacijom. Domenski objekti koji se koriste u aplikaciji predstavljaju objekte učenja, japanske reči sa odgovarajućim prevodima, znakove japanskih pisama, pitanja, kvizove i zanimljivosti; svi ovi objekti se persistiraju u lokalnoj bazi podataka. Jezgro aplikacije se dalje oslanja na niz komponenata koje ga povezuju sa izvorima/skladištima podataka koja sadrže aplikacione podatke, jezičke resurse i korisničke podatke.

Prva od ovih komponenata je zadužena za komunikaciju sa bazom podataka. Broker baze podataka koji predstavlja interfejs ka bazi je dostupan iz svih delova aplikacije i stara se o čitanju jezičkih resursa (reči/znakova/zanimljivosti/kvizova) iz baze i vođenju evidencije o istim. Baza je implementirana korišćenjem SQLite sistema koji je dostupan na Android platformi, ali se po potrebi može izmeniti zarad prilagođavanja nekoj drugoj platformi, pri čemu izmene u samom jezgru aplikacije neće biti potrebne budući da ova navedena komponenta predstavlja interfejs koji u potpunosti apstrahuje način implementacije skladišta podataka, koje u kranjem slučaju ne mora da bude ni baza podataka već može biti nekakav drugačiji sistem (fajl sistem, onlajn skladište itd.).

Druga komponenta za komunikaciju sa Android sistemom dodatno apstrahuje karakteristike Android platforme od ostatka aplikacije i služi za:

- skladištenje jednostavnih aplikacionih podataka koji se čuvaju iz sesije u sesiju (koristi se Androidov sistem deljenih preferenci)
- učitavanje/parsiranje datoteka (iz privatnog skladišnog prostora dodeljenog svakoj Android aplikaciji)
- dinamičko učitavanje slika
- prikaz kratkih poruka korisniku
- dobavljanje, privremeno čuvanje i puštanje audio klipova reči (koje se oslanja na komponentu za komunikaciju sa web aplikacijom)

Treća komponenta ima ulogu da komunicira sa veb aplikacijom i pristupa onlajn resursima (audio datotekama). Ovaj proces se obavlja u pozadini, nezavisno od trenutnog stanja aplikacije i bez blokiranja interakcije sa korisnikom. Veb aplikacija čuva i isporučuje korisniku objekte učenja na odgovarajućem jeziku i takođe evidentira njegove uspehe radi poređenja sa drugim korisnicima.



Slika 10. Arhitektura JapGO aplikacije

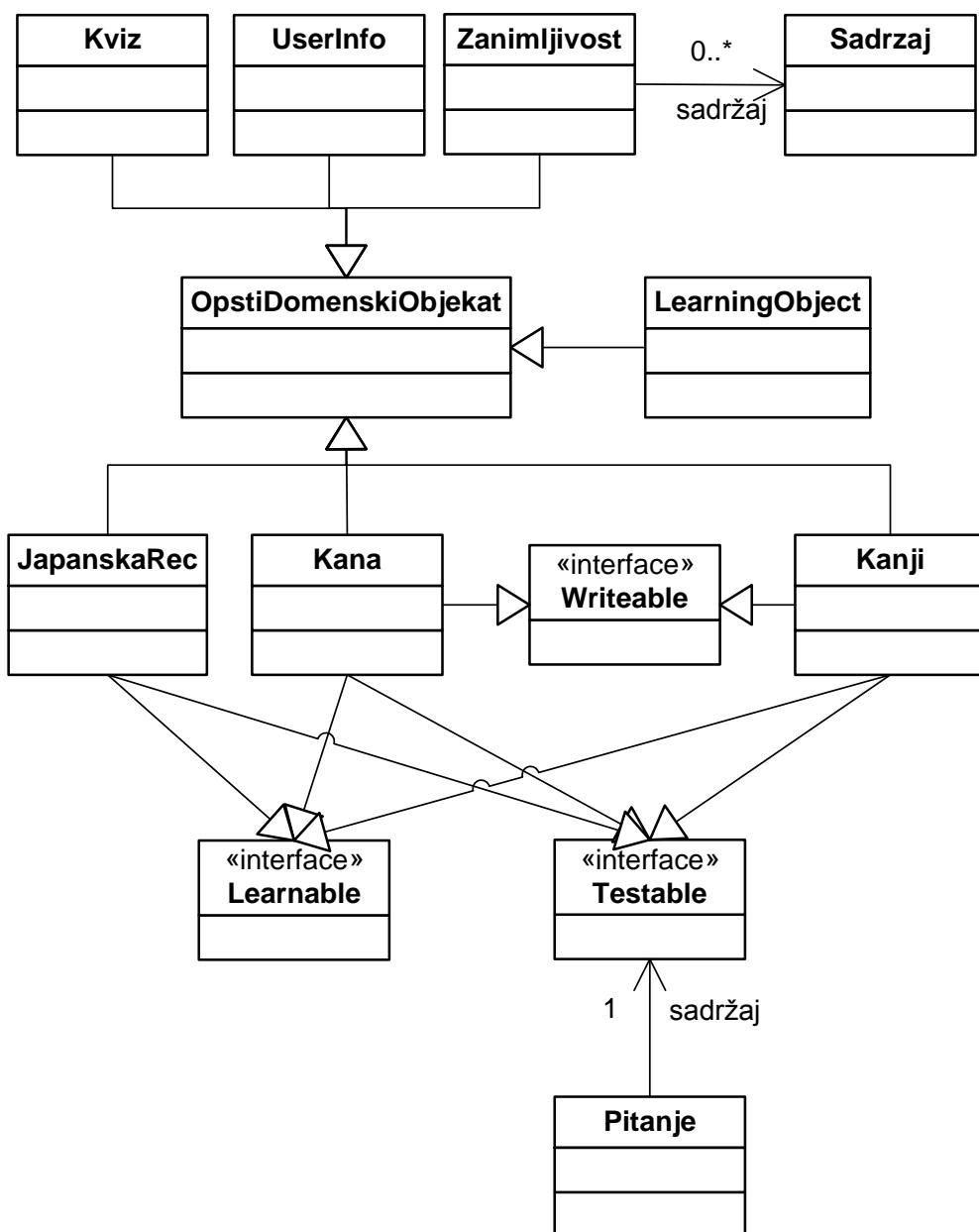
#### 4.2.1. Domenske klase

Iz velikog broja klasa korišćenih u projektu, domenske klase nose poseban značaj (Slika 11). Ove klase se koriste da kroz aplikaciju prenose podatke persistirane u bazi podataka, pa predstavljaju takozvane „objekte modela“ (model objects). Tri najznačajnije klase su **JapanskaRec**, **Kana** i **Kanji** koje predstavljaju tri osnovne vrste edukativnih elemenata od kojih su sastavljene lekcije, tj. objekti učenja. Ove veze sa **LearningObject** klasom nisu predstavljene, kao ni dodatne klase koje nose informacije značajne u kontekstu tekućeg objekta učenja, sa ciljem da se očuva jednostavnost prikazanog dijagrama.

JapanskaRec, Kana i Kanji se koriste u raznim delovima aplikacije, često u međusobno istom ili sličnom kontekstu. Iz tog razloga su kreirana tri interfejsa koja predstavljaju određene funkcionalnosti, usluge, koje ovi objekti treba da pruže aplikaciji u pojedinim njenim delovima. **Learnable** interfejs predstavlja „nešto što se može naučiti“ i sadrže metode koje se tiču statusa i vremena naučenosti objekta. Sve tri navedene klase implementiraju ovaj interfejs. **Testable** interfejs predstavlja „nešto čije se poznavanje može proveriti“, i takođe je implementiran od strane sve tri klase. Ovaj interfejs sadrži metode za evidenciju broja koliko puta su određene reči/kane/kandžiji figurisali kao pitanje i koliko je puta korisnik odgovorio tačno/pogrešno; takođe, tu su i metode za dobavljanje „pitanja“ i „odgovora“ za neki element. Korišćenje interfejsa omogućava značajnu fleksibilnost – ukoliko bismo poželeli sada da dodamo neku novu vrstu edukativnih sadržaja u aplikaciju, npr. „gramatičko pravilo“ ili „japansku rečenicu“, dovoljno bi bilo da implementiraju ovaj interfejs i mogli bi se prezentovati kao deo testova aplikacije. Na kraju, interfejs **Writeable** je namenjen za „nešto što može da se ispiše“ i implementiran je od strane Kana i Kanji klase. Na ovaj način razne aktivnosti aplikacije rade sa generičkim Writeable-ima umesto sa konkretnim Kana i Kanji objektima i mogu da preuzmu sve potrebne podatke za njihov prikaz bez potrebe da znaju sa kojom konkretnom implementacijom rade. Primer su aktivnosti za vežbanje pisanja, igra pamćenja i neki od internih mehanizama aplikacije.

Pored ove tri klase, tu su još klase **Kviz**, čiji objekti nose karakteristike i podešavanja nezavisnih testova znanja, **UserInfo** koja obuhvata podatke o korisnicima (ili tekućem korisniku ili nekom od njegovih prijatelja) i koja se koristi pri razmeni podataka sa serverom, i na kraju, **Zanimljivost** koja sadrži metapodatke koji opisuju zanimljive informacije o Japanu prezentovane u obliku članaka.

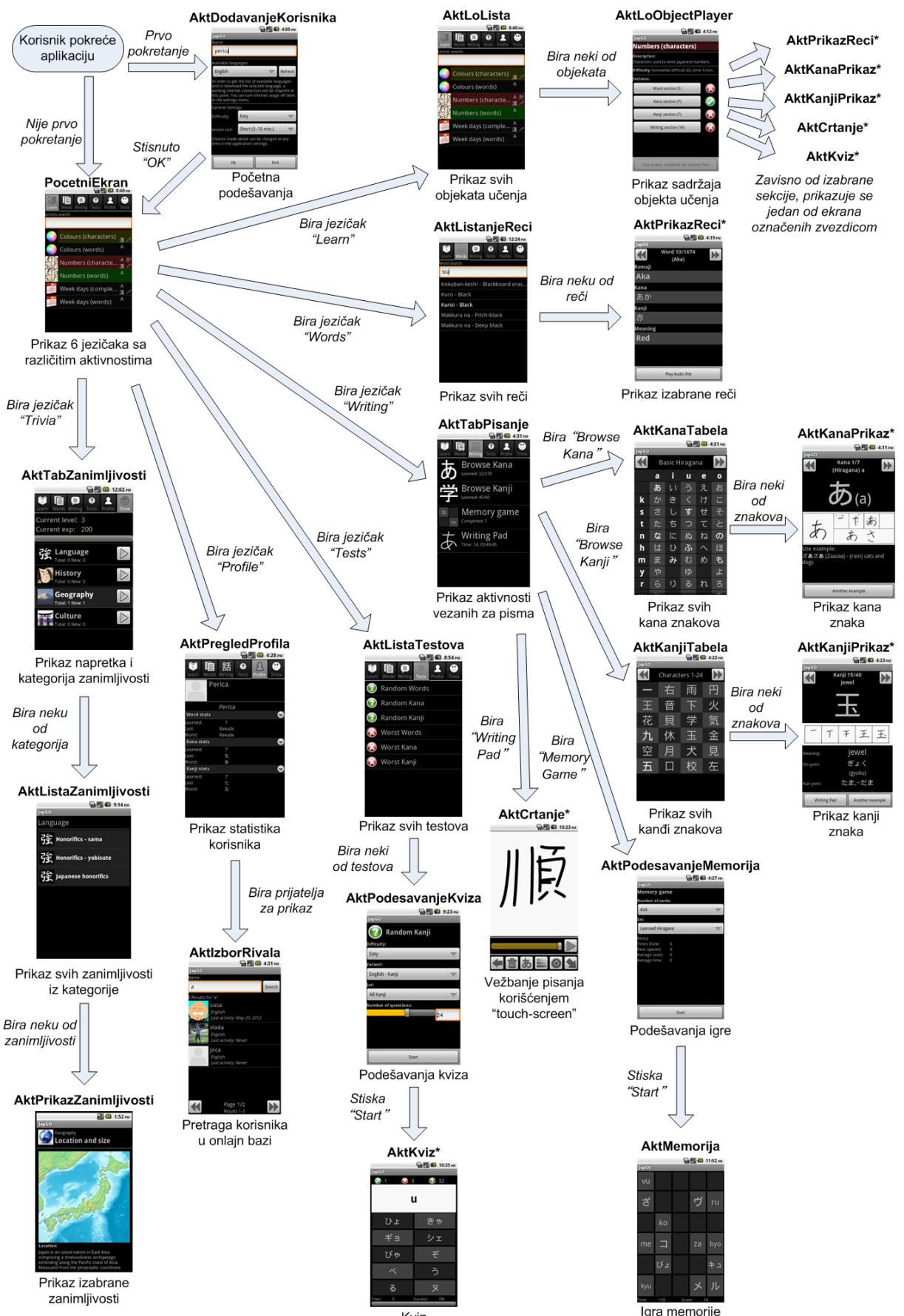
Iz perspektive implementacije, najvažniji element domenskih klasa je klasa **OpstiDomenSKIObjekat**. Ova klasa, kao što njeni ime govori, predstavlja generičke domenske objekte koji imaju mogućnost persistiranja/čitanja iz baze podataka. Klase ove metode koje sve klase domenskih objekata moraju da naslede obuhvataju metode koje daju informacije o njihovim tabelama, ključevima, kolonama, identifikacionim uslovima i uslovima za pretragu. BrokerBazePodataka koji komunicira sa bazom je dizajniran tako da koristi ove metode i radi sa generičkim objektima klase OpstiDomenSKIObjekat. Na ovaj način je dovoljno definisati nekoliko metoda – za čitanje, ubacivanje, izmenu i brisanje opštih domenskih objekata – i te metode će prihvavati i uspešno raditi sa bilo kojom od implementacija te klase.



Slika 11. Dijagram domenskih klasa

#### **4.2.2. Korisnički interfejs i tok kretanja korisnika**

Na slici 12 je dat jedan opšti pregled ekrana aplikacije i načina kretanja korisnika kroz interfejs i aktivnosti. Za svaki od ekrana je dato ime odgovarajuće klase, slika ekrana i njegov kratak opis. Kretanje sa ekrana na ekran je prikazano putem strelica nad kojima su napisani uslovi za prelazak. Na ovoj slici nisu prikazane sve putanje niti svi ekran, jer bi njihovo prikazivanje značajno umrsilo dijagram. Šest osnovnih „tokova“ koji su prikazani u rasporedu približno odgovarajućem smeru kretanja kazaljke na satu su: rad sa objektima učenja (lekcijama), rad sa rečima, rad sa znacima japanskih pisama, utvrđivanje znanja na testovima, pregled profila i pregled zanimljivosti o Japanu.



Slika 12. Klase paketa "interfejs" i tok kretanja korisnika

#### 4.2.3. Veb aplikacija i komponente za komunikaciju sa njom

Strukturu veb aplikacije čine tri PHP skripte:

- jezici.php
- zanimljivosti.php
- korisnik.php

**Jezici.php** se stara o dobavljanju liste raspoloživih jezika (što je određeno setom objekata učenja dostupnim na serveru) i preuzimanju odgovarajućeg kompleta objekata učenja. Ova skripta se kontaktira pri prvom pokretanju aplikacije, kada se preuzimaju edukativni sadržaji na odgovarajućem (izabranom) jeziku.

**Zanimljivosti.php** se stara o dostavljanju zanimljivosti o Japanu. Ova skripta evidentira zanimljivosti pohranjene na serveru, predaje njihov spisak korisničkoj aplikaciji po zahtevu i po potrebi dozvoljava njihovo preuzimanje. Zanimljivosti se čuvaju u datotekama u XML formatu koje su grupisane po folderima namenjenim pojedinačnim jezicima. Ispod je dat primer jedne datoteke (sa skraćenim tekstualnim delovima):

##### \english\13.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<zanimljivost kategorija="1">
    <naslov>Hatsune Miku</naslov>
    <slicica>13-1.jpg</slicica>
    <sadrzaj>
        <slika>13-2.jpg</slika>
        <tekst>Hatsune Miku is a singing synthesizer...</tekst>
        <slika>13-3.jpg</slika>
        <tekst>In late November 2009, ...</tekst>
        <podnaslov>Hatsune Miku concert</podnaslov>
        <slika>13-4.jpg</slika>
        <tekst>As a virtual idol...</tekst>
    </sadrzaj>
</zanimljivost>
```

U pitanju je jednostavan XML dokument; koren element `<zanimljivost>` sadrži atribut „kategorija“ koji se odnosi na broj kategorije kojoj pripada (jezik, geografija, istorija, kultura). Deca ovog elementa su `<naslov>` koji sadrži naslov zanimljivosti, `<slicica>` sa

putanjom do sličice koja predstavlja tekuću zanimljivost i <sadrzaj> koji u sebi sadrži kompletan sadržaj zanimljivosti. U okviru sadržaj taga mogu da se nađu <slika> (sa putanjom do slike na serveru), <tekst> i <podnaslov> koji se kasnije formatiraju na odgovarajući način pri prikazu u Android aplikaciji.

**Korisnik.php** skripta preuzima podatke o korisniku od korisničke aplikacije i skladišti ih u MySQL bazi podataka. Korisnici mogu da pretražuju ovu bazu i koristeći Android aplikaciju izaberu nekog od korisnika (ili više njih) kao svog prijatelja. Sličice koje korisnici mogu da izaberu za svoj profil (i koje se kasnije prikazuju pri pretrazi korisnika/prijatelja) se čuvaju u fajl sistemu.

Komunikacija između veb i Android aplikacije je organizovana na sledeći način – Android aplikacija šalje HTTP POST zahtev za neku od stranica, sa odgovarajućim nizom parametara, a kao odgovor dobija JSON String koji se onda na jednostavan način korišćenjem org.json.JSONObject i org.json.JSONArray klase parsira. U nastavku su dati primeri zahteva/odgovora u slučaju kada korisnička aplikacija zahteva spisak korisnika čije ime sadrži neki string („s“ u primeru).

### Sadržaj HTTP POST zahteva za /korisnik.php

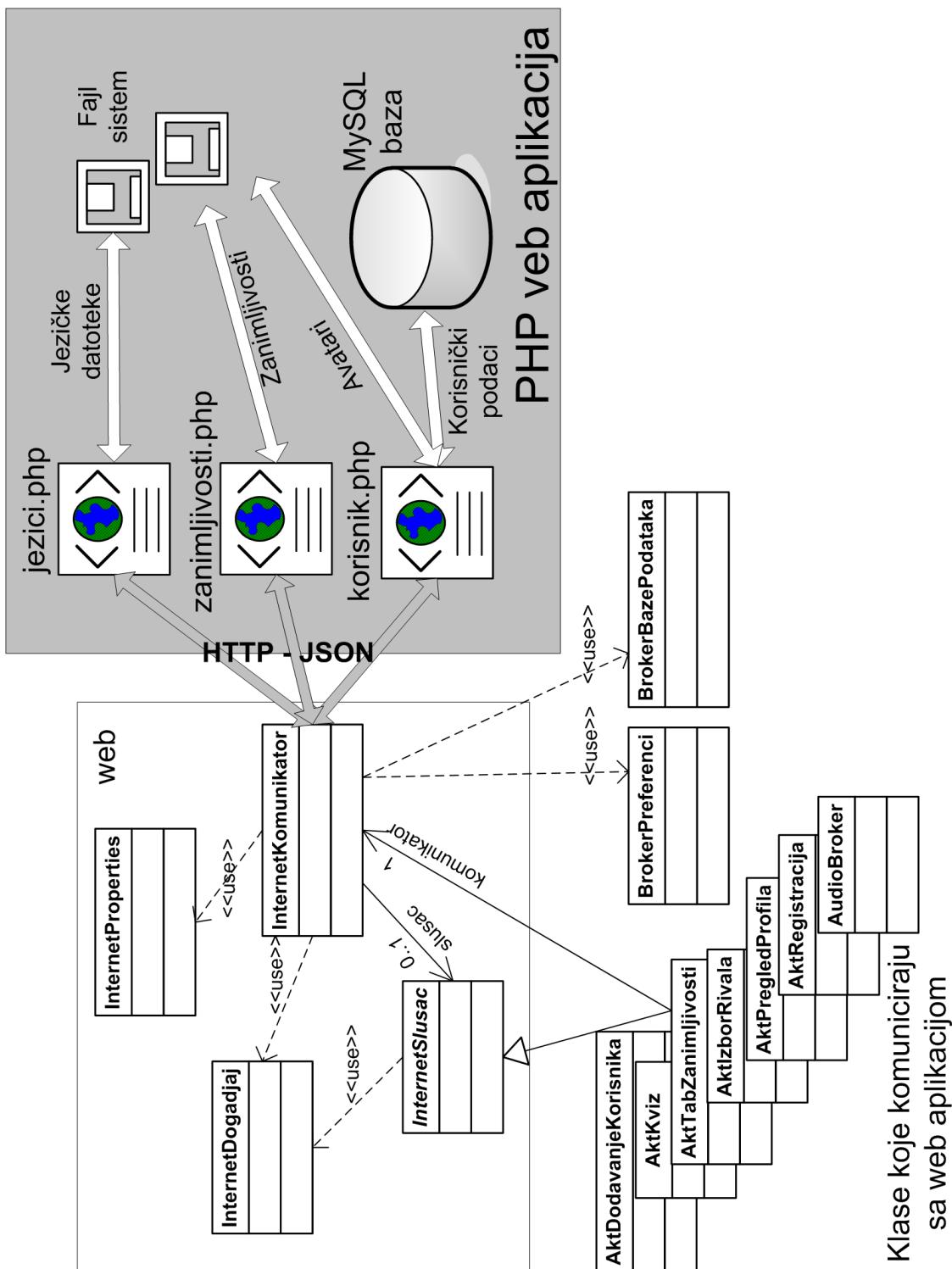
```
action=getRivali&globalID=18&trazeno=s&stranica=1&izbaci_idjeve=15
```

### Sadržaj HTTP odgovora (JSON)

```
{"ljudi": [
    {"id": "2", "ime": "miske", "jezik": "English", "last_visit": "0",
     "slika": "avatari\\avatar.2.jpg"},
    {"id": "3", "ime": "susa", "jezik": "English",
     "last_visit": "1337489622", "slika": "avatari\\avatar.3.jpg"},
    {"id": "5", "ime": "Sumi", "jezik": "English",
     "last_visit": "1338291020", "slika": "avatari\\avatar.5.jpg"}],
    "status": true,
    "rezultata": "5",
    "rezPoStranici": 3,
    "brojStranica": 2,
    "stranica": "1"}
```

Komponente za komunikaciju sa veb aplikacijom u okviru Android aplikacije obuhvataju klasu **InternetKomunikator** koja je implementirana kao singleton i koja sadrži niz metoda za

komunikaciju. Ova klasa se oslanja na InternetProperties klasu koja sadrži URL-ove PHP skripti i konstante koje sadrže imena/vrednosti POST parametara. Klase koje žele da komuniciraju sa veb aplikacijom implementiraju **InternetSlusac** interfejs, šalju svoj zahtev i zatim bivaju obaveštene u trenutku pristizanja odgovora preko metode InternetSlusac interfejsa koja prima InternetDogadjaj objekat kao parametar. **InternetDogadjaj** sadrži informacije o uspehu/neuspehu komunikacije i telo odgovora dobijenog u HTTP odgovoru.



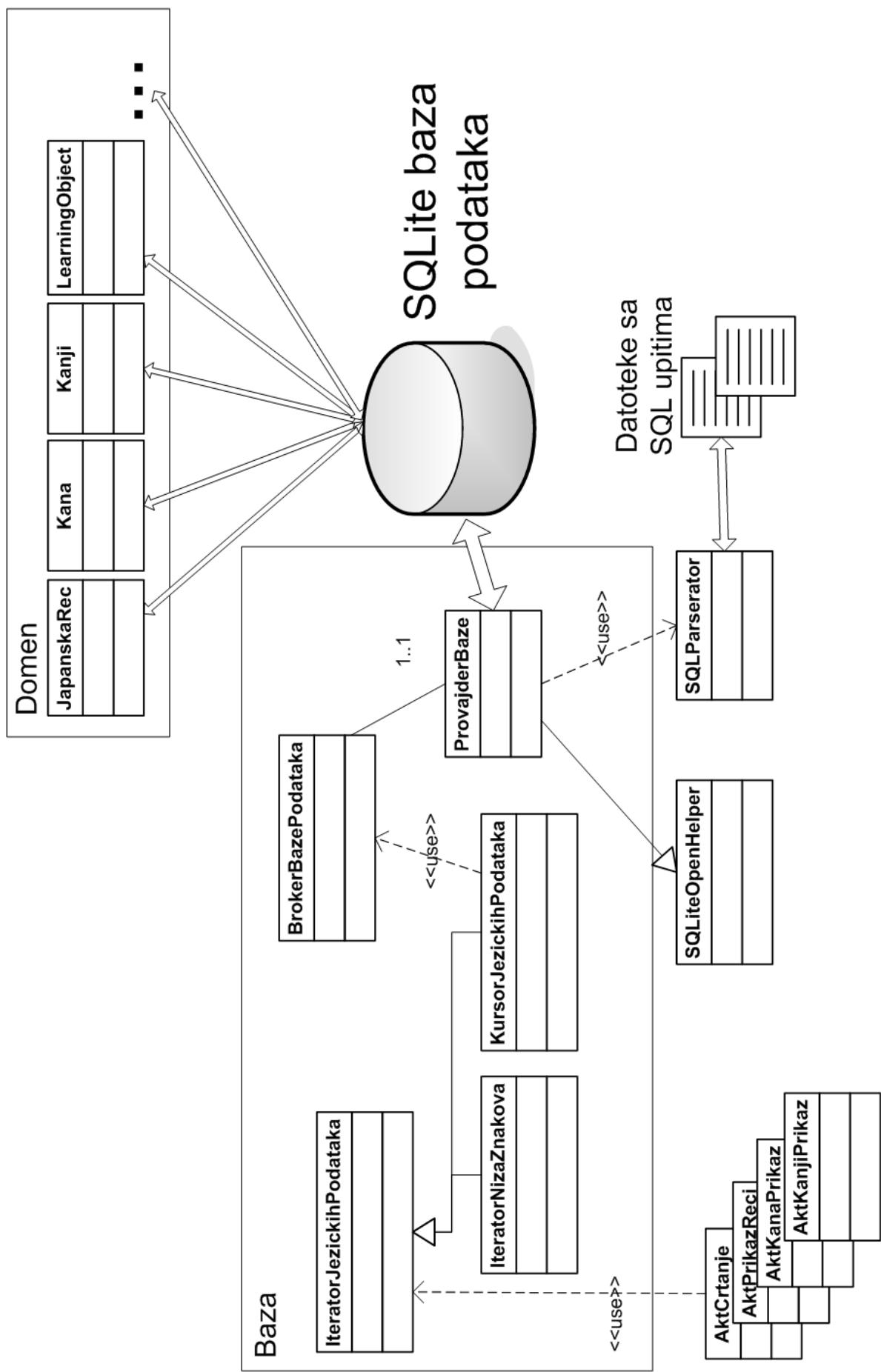
Slika 13. Struktura paketa „web“ i web aplikacije

#### **4.2.4. Lokalna baza i komponente za komunikaciju sa njom**

Za skladištenje većine podataka u okviru same aplikacije koristi se SQLite baza podataka. Za komunikaciju sa ovom bazom se koristi Androidova klasa **SQLiteOpenHelper** koja je nasleđena u klasi ProvajderBaze. U okviru klase **ProvajderBaze** se reimplementira metoda koja kreira bazu podataka pri prvom pokušaju da joj se pristupi. ProvajderBaze u sebi sadrži niz referenci na sve CREATE TABLE upite svih domenskih objekata i izvršava ih sve ukoliko treba kreirati novu bazu podataka. Osim toga, koristi klasu **SQLParserator** da izvrši sve SQL upite definisane u assets/sql folderu projekta (ukoliko ih ima). Ovi upiti generalno služe da popune novu bazu podataka nekim inicijalnim podacima.

Za konkretan rad sa domenskim objektima, za njihovo učitavanje, ažuriranje, dodavanje i brisanje iz baze podataka se koristi klasa **BrokerBazePodataka**. Kao što je već pomenuto u delu koji se bavio domenskim klasama, ova klasa radi sa generičnim objektima tipa OpstiDomenskiObjekat i može da izvrši sve operacije nad objektima bilo koje klase koja je nasleđuje. BrokerBazePodataka je projektovan kao singleton i može mu pristupiti bilo koja od klasa aplikacije.

Neke od aktivnosti aplikacije prezentuju niz edukativnih jedinica korisnicima i dozvoljavaju korisniku da se prebacuje sa jedne na drugu korišćenjem dugmića levo/desno. Problem nastaje kada se uzme u obzir da su takve aktivnosti korisniku dostupne na dva načina – kao sastavni deo lekcija (objekata učenja) ili nezavisno. Kada se vrši nezavistan pristup, prezentuju se svi podaci iz baze podataka; kada se tim aktivnostima pristupa kroz neku od lekcija, prezentuju se samo oni elementi koji su deo lekcije. Kako bi se omogućilo da takve aktivnosti funkcionišu na unifikovan način, bez obzira da li se prezentuju svi elementi ili samo određen niz, projektovan je interfejs **IteratorJezickihPodataka<T>** koji deklariše metode za kretanje levo/desno po nekom skupu edukativnih jedinica i za preuzimanje određenih elemenata tekuće jedinice. Ovu klasu dalje nasleđuju **IteratorNizaZnakova** koji je namenjen radu sa znakovima sadržanim u nizu (odnosno listi) i **KursorJezickihPodataka** koji radi sa Cursorom i vrši kretanje po bazi podataka. Koristeći ovaj sistem, aktivnost treba samo da zna da radi sa IteratoromJezickihPodataka dok je konkretna implementacija poznata samo onome ko startuje aktivnost i ima znanje o širem kontekstu njenog startovanja.



Slika 14. Struktura paketa „baza“

#### **4.2.5. Komponente za komunikaciju sa Android sistemom**

Komponente za komunikaciju sa Android sistemom predstavljaju niz raznovrsnih, nepovezanih klasa koje obezbeđuju jednostavan interfejs za komunikaciju sa raznim aspektima Android sistema i tipično su realizovane kao klase sa statičnim metodama ili singletoni.

**BrokerPreferenci** je klasa koja treba da omogući persistiranje malih količina primitivnih tipova podataka. Na Android platformi je to posao SharedPreferences klase koja obavlja snimanje/čitanje iz specijalnih XML fajlova; rad sa ovom klasom se nalazi u osnovi BrokerPreferenci.

**InfoDisplej** klasa sadrži nekoliko metoda za brzo kreiranje tzv. Toast-ova i najjednostavnijih dijaloga koji sadrže samo poruku i dugme OK. Složeniji dijalozi se kreiraju tamo gde su potrebni ili imaju sopstvene klase.

**AudioBroker** se stara o dobavljanju audio zapisa reči japanskog sa interneta i njihovom privremenom čuvanju u okviru fajl sistema. Ovim keširanjem audio fajlova se poboljšava efikasnost aplikacije i smanjuje upotreba interneta.

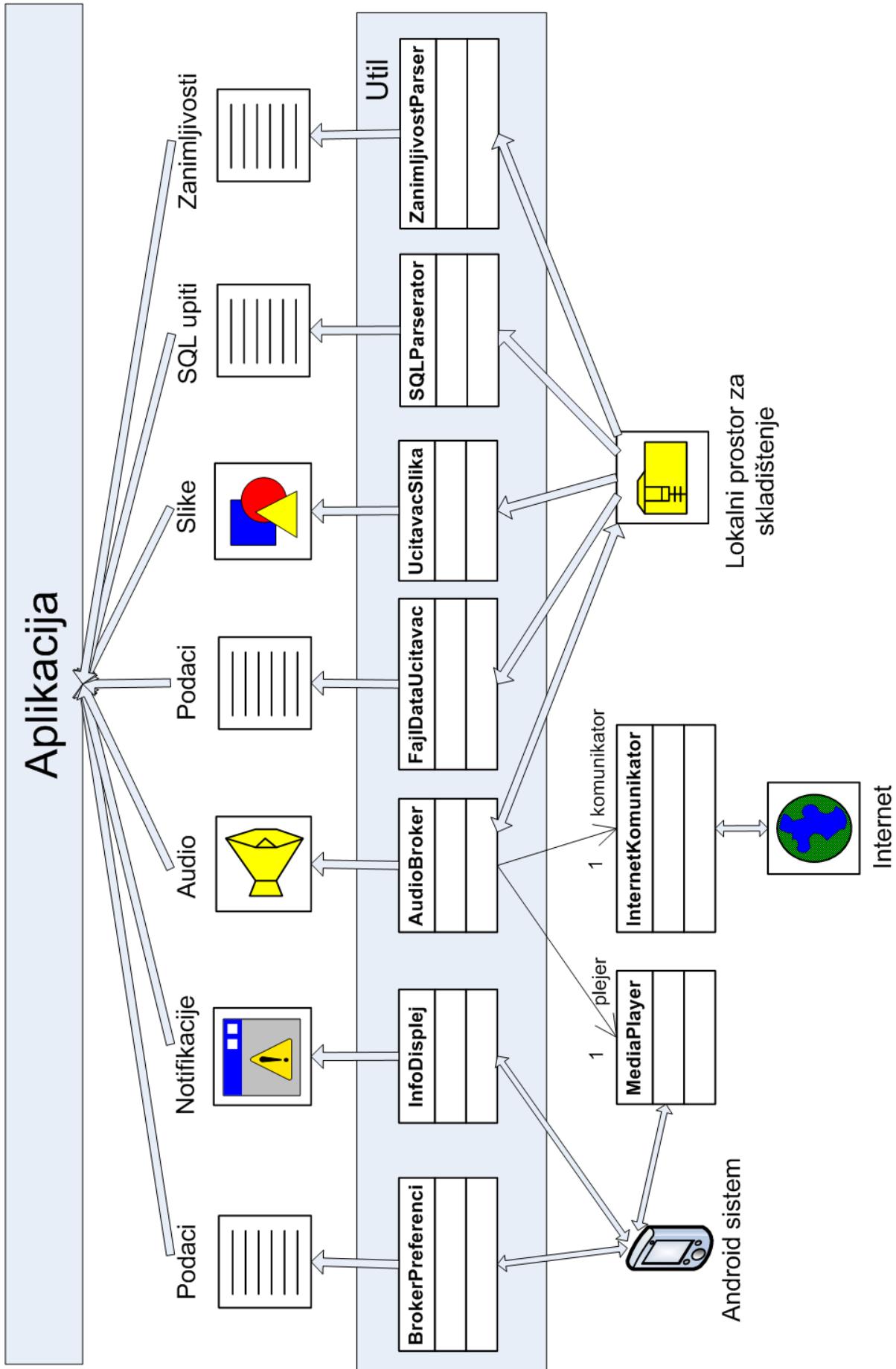
**FajlDataUcitavac** ima ulogu parsiranja određenih vrsta tekstualnih fajlova i vraćanja podataka u vidu liste Stringova.

**UcitavacSlika** učitava slike sa zadatim imenom, pri čemu obavlja proveru na dva mesta – u okviru assets-a aplikacije i u okviru privatnih fajlova; ukoliko ne nađe traženu sliku ni na jednom od ta dva mesta, vraća default „no image“ sliku. Slika se vraća u obliku objekta Bitmap klase.

**SQLParserator** se koristi od strane SQLiteOpenHelpera pri kreiranju baze podataka; ova klasa vraća niz SQL upita iz zadatog SQL fajla.

**ZanimljivostParser** obavlja parsiranje XML fajlova koji sadrže zanimljivosti. Parsiranje se obavlja uz pomoć SAX parsera, a rezultat parsiranja su objekti klase Zanimljivost.

Korišćenjem niza ovakvih klasa, od glavnog dela aplikacije se apstrahuju svi implementacioni detalji, što omogućava laku buduću izmenu, na primer – premeštanje lokacije gde se čuvaju fajlovi, promenu formata nekih fajlova, promenu izvorišta audio zapisa itd.



Slika 15. Klase paketa "util" i njihove funkcije u aplikaciji

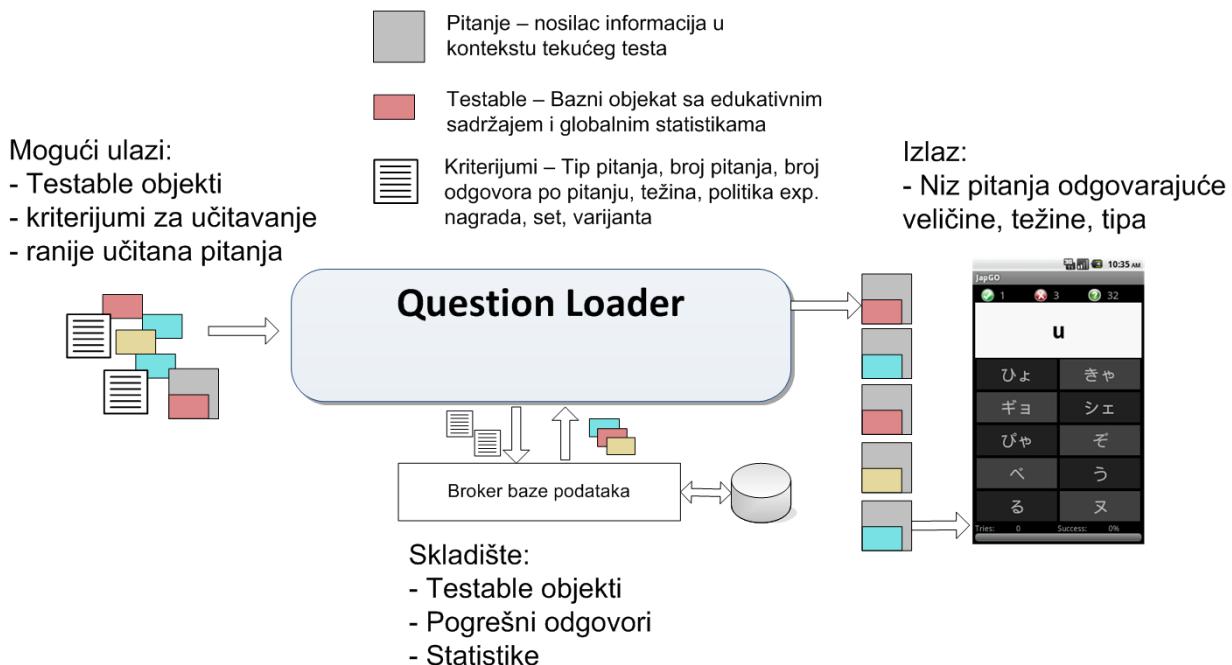
#### 4.2.6. Mehanizam za generisanje testova

Mehanizam za generisanje testova je jedan od zanimljivijih mehanizama aplikacije pošto mora da radi sa nizom raznovrsnih parametara i bazom podataka i da svoj posao obavlja asihrono kako ne bi došlo do zastoja u prezentaciji pitanja korisniku.

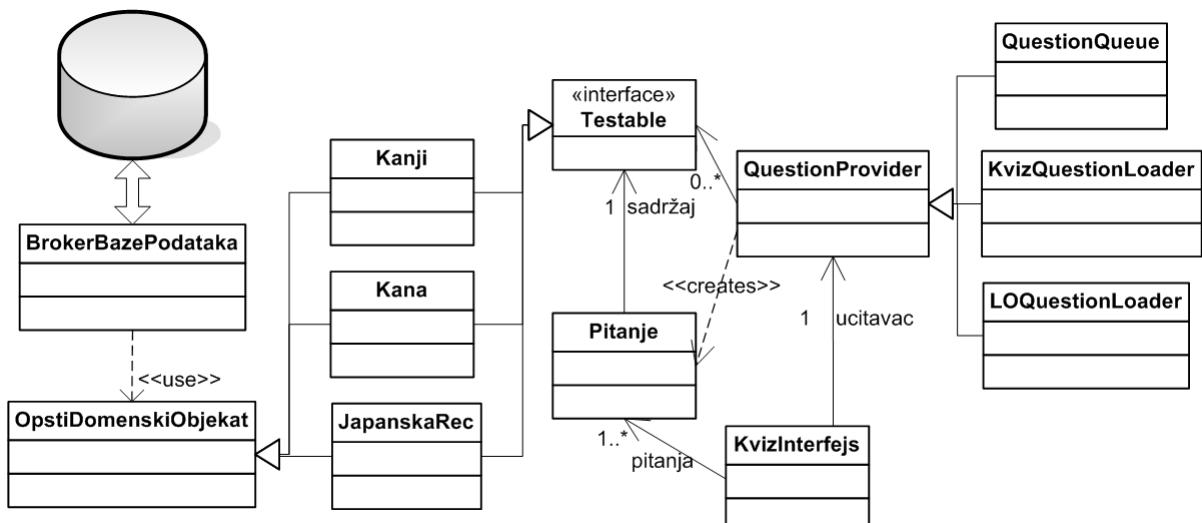
Osnovni delovi sistema za testiranje su komponenta za učitavanje pitanja koja ih dodaje na kraj jednog reda (queue) i interfejs za interakciju sa korisnikom koji uzima pitanja sa njegovog početka i prikazuje ih (Sl. 16). Ova dva procesa (učitavanje i prikazivanje) se obavljaju konkurentno, time iskorišćavajući vreme koje je korisniku potrebno da odgovori na pitanje i izbegavajući blokiranje interfejsa dok se sva pitanja učitaju.

Dijagram klasa koje učestvuju u ovom procesu je dat na slici 17. Komponente za učitavanje implementiraju **QuestionProvider** interfejs i postoji ih nekoliko vrsta zavisno od namene. Ulaz komponente za učitavanje predstavljaju objekti-osnovne jedinice sadržaja koji implementiraju interfejs Testable. Preko ovog interfejsa se generički pristupa sadržajima objekata bez obzira da li su u pitanju reči, kana ili kanji znaci. Pored njih ulaz predstavlja i niz kriterijuma na osnovu kojih će biti kreirani objekti-pitanja ispunjeni kontekstnim informacijama važnim za tekući kviz. Ovi kriterijumi obuhvataju: broj pitanja na testu, broj netačnih odgovora po pitanju, varijantu pitanja, nagradno „iskustvo“ koje tačani odgovori daju i eventualno ograničenje seta pitanja koja dolaze u obzir.

U aplikaciji trenutno postoje tri implementacije QuestionProvider interfejsa. Prva je jednostavan **QuestionQueue** koji samo obmotava listu gotovih pitanja. Pretpostavka je da u nekim situacijama može postojati već učitana lista (npr. ponavljanje testa) koja se samo može proslediti objektu ove klase. Druga implementacija je **QuizQuestionLoader** koja je namenjena učitavanju pitanja za nezavisno pokrenute/konfigurisane kvizove i čiji su parametri uglavnom sadržani u Kviz objektu. Ova klasa na osnovu primljenih parametara tipično priprema u nekoj meri slučajno izabrana pitanja. Na kraju, treća implementacija je **LOQuestionLoader**. Ova implementacija generiše pitanja za kvizove koji su deo lekcija, odnosno objekata učenja i parametre prima u obliku LearningObject-a za koji se generiše kviz. Pitanja se ovde učitavaju prema edukativnom sadržaju LearningObjecta.



Slika 16. Mehanizam za učitavanje i prikaz pitanja na testu



Slika 17. Dijagram klasa koje učestvuju u generisanju pitanja

#### 4.2.7. Aktivnost – vežbanje pisanja

Aktivnost za vežbanje pisanja se oslanja na ceo niz klasa za svoje funkcionisanje, pa će u ovom delu biti malo detaljnije opisana. Globalni prikaz ove aktivnosti je dat na slici 18.

Prikaz aktivnosti za vežbanje pisanja se dobija kombinovanjem nekoliko slojeva koji su smešteni jedan iznad drugog. Ovaj mehanizam je implementiran tako što svaki sloj predstavlja jedan *custom view* i u sebi sadrži i prezentacionu logiku i logiku koja se odnosi na njegove dužnosti u aktivnosti za vežbanje pisanja.

Počevši od dna, prvi sloj je sloj za crtanje, **PapirView**, koji implementira interfejs **Papir**. Ovaj sloj je providan i sadrži logiku za reagovanje na dodire prstima po ekranu i iscrtavanje odgovarajućih linija, kao i za opciono bleđenje linija sa vremenom (uz pomoć ugnježdenog Threada).

Iznad ovog sloja nalazi se **VodiljeView** koji iscrtava linije vodilje, odnosno neki od znakova. Ovaj View implementira interfejs **VodiljeKontroler** i stara se za prikazivanje/skrivanje vodilja, za izmenu njihove boje, pozicije i veličine. Pozicije ova dva sloja se mogu izmeniti u podešavanjima; po defaultu se sloj sa vodiljama nalazi iznad papira, ali je zato delom providan; moguće je smestiti ga na dno u kom slučaju će linije koje korisnik vuče u potpunosti prekrivati linije-vodilje.

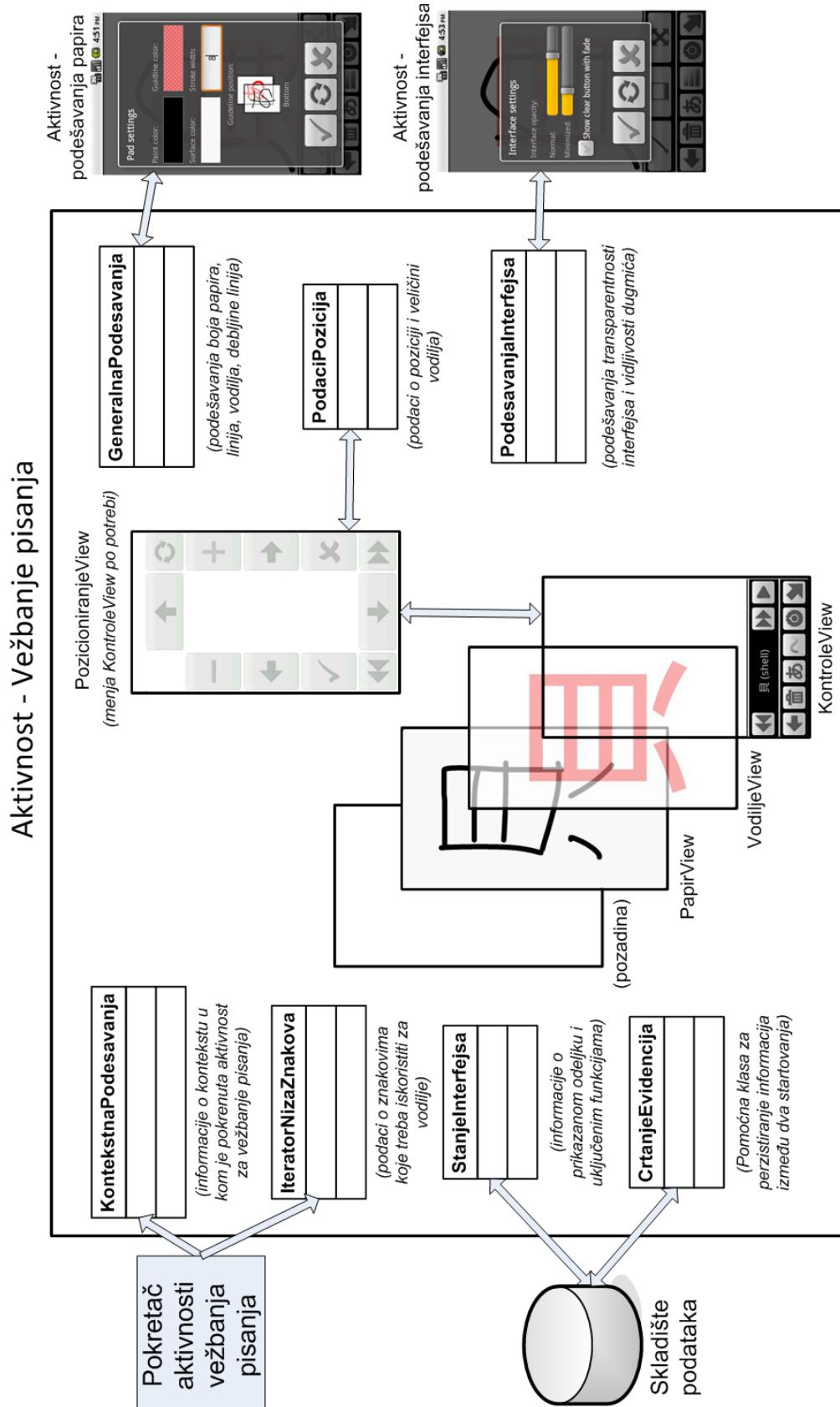
Na vrhu se nalazi najznačajniji sloj – **KontroleView** koji predstavlja kontroler svih funkcija aktivnosti i koji iscrtava korisnički interfejs – dugmiće i sekcije koje se mogu prikazati/skloniti. Ukoliko korisnik aktivira pozicioniranje vodilja, ovaj sloj privremeno nestaje i biva zamjenjen slojem koji sadrži **PozicioniranjeView** sa kontrolama za pozicioniranje i promenu veličina vodilja.

Aktivnost za vežbanje pisanja može biti startovana u nekoliko različitih situacija – kroz neku od lekcija (objekata učenja), prilikom nezavisnog pregleda nekog od znakova i, konačno, kompletno nezavisno od bilo koje druge aktivnosti u aplikaciji. Zavisno od konteksta u kom je aktivnost startovana, postoje određena ograničenja koja se moraju primeniti na interfejs i uopšte funkcionisanje ove aktivnosti. (startovanje iz lekcije povlači obavezan prikaz vodilja). Ove informacije i ograničenja se aktivnosti dostavljaju kroz objekat klase **KontekstnaPodešavanja**. Za pristup znakovima koje treba prikazati vodiljama aktivnosti se dostavlja objekat koji implementira InteratorNizaZnakova, što omogućava isti način rada i ako se podaci vade direktno iz baze podataka i ako se radi o jednostavnom nizu od nekoliko elemenata .

Pored ovih ulaznih informacija, sama aktivnost učitava i perzistira određene informacije pri svakom startovanju. Ove informacije se u memoriji nalaze u objektu klase **StanjeInterfejsa**, gde se pamte uključene kontrole i prikazane sekcije u interfejsu, dok se za druge, globalne informacije koristi pomoćna klasa **CrtanjeEvidencija** i njene statične metode za perzistiranje jednostavnih, nestruktuiranih podataka. Osim toga, aktivnost koristi primenjuje podešavanja koja se dobijaju od aktivnosti za podešavanje papira, interfejsa i

sloja za pozicioniraje; ove informacije se čuvaju u objektima klasa

### GeneralnaPodesavanja, PodaciPozicija i PodesavanjeInterfejsa.



Slika 18. Aktivnost za vežbanje pisanja

#### 4.2.8. Taksativno navedene komponente:

Tabela 4. Komponente aplikacije

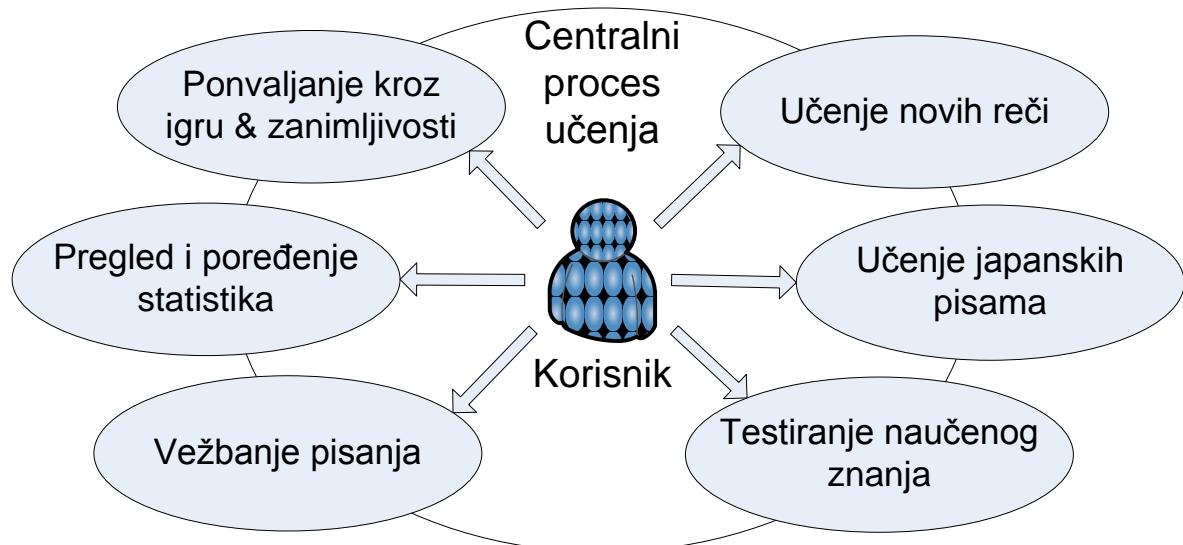
Paketi	Klase	Paketi	Klase
crtanje	AktCrtanje AktCrtanjeGeneralnaPodesavanja AktCrtanjeInterfejsPodesavanja CrtanjeEvidencija CrtanjeKontroler  GeneralnaPodesavanja KontekstnaPodesavanja KontroleView Papir PapirView PodaciPozicija  PodesavanjaInterfejsa PozicioniranjeView StanjeInterfejsa VodiljeKontroler VodiljeView	memorija	AktMemorija AktPodesavanjeMemorija Kartica LogikaMemorija MemorijaEvidencija
data	BrokerBazePodatka IteratorJezickihPodatka IteratorNizaZnakova KursorJezickihPodatka ProvajderBaze	pismo	AktKanaPrikaz AktKanaTabela AktKanjiPrikaz AktKanjiTabela AktTabPisanje CharacterDisplayAbstract
domen	JapanskaRec Kana Kanjii Learnable OpstiDomenskiObjekat Pitanje Testable Writeable	profil	AktIzborRivala AktIzmenaAvatara AktPregledProfila AktRegistracija AvatarDAO
gui	AutoRepeatImageButton Fadeable FadeableLinearLayout FadeableView TransparentImageButton	reci	KorisnikUserInfoDAO PrijateljUserInfoDAO UserInfo ZadatakDobaviListu ZadatakDobaviSlicice  ZadatakRefreshRivala ZadatakRegistrujSe ZadatakUpdateUserInfo ZapisListeKorisnika  AktListanjeReci AktPrikazReci
help	AktHelp LogikaHelp	util	Aplikacija ArgValidator  AudioBroker BrokerPreferenci FajlDataUcitavac GrafikaUtils InfoDisplej  MeniKonstante PaketPodesavanja SqlParserator  StaticAsyncTask StringOperacije TekstoFarbac  UcitavacSlika VrednostPaginacije VrednostPaginacijeString
instalacija	AktDodavanjeKorisnika		
interfejs	PocetniProzor Redirektor AktPodesavanjeAplikacije		
kviz	AktKviz AktListTestova AktPodesavanjeKviza		

	Kviz	ZanimljivostParser
	KvizQuestionLoader	InternetDogadjaj
	LogikaKviz	InternetKomunikator
	LOQuestionLoader	InternetProperties
	QuestionProvider	InternetSlusac
	QuestionQueue	AktListaZanimljivosti
	SnimacPitanja	AktPrikazZanimljivosti
<b>lobjekts</b>	AktLObjectPlayer	AktTabZanimljivosti
	AktLOLista	LevelUpPopUpCreator
	LearningObject	LogikaZanimljivosti
	LearningObjectElement	Sadrzaj
	LearningObjectKana	Zanimljivost
	LearningObjectKanji	AlphaPatternDrawable
	LearningObjectRec	ColorPanelView
	LogikaLO	ColorPickerDialog
	LOKvizScore	ColorPickerView
	EvidencijaGlobal	
<b>logika</b>	EvidencijaKorisnik	
	KonstanteLokacije	
	LogikaIskustvo	
	LogikaKana	
	LogikaKanji	
	LogikaRec	
	LogikaWriteable	
	PodesavanjaGlobal	

### 4.3. Funkcionalnosti aplikacije

Osnovna uloga JapGO aplikacije je da omogući učenje japanskog jezika na zabavan, interaktivn način, putem mobilnih uređaja, kako bi se studenti motivisali da uče. Aplikacija je u potpunosti prilagođena specifičnostima japanskog jezika, održava bazu japanskih reči i odgovarajuću bazu reči-prevoda koje mogu biti na nekom od dostupnih jezika. Osnovni proces učenja se odvija kroz sesije u kojima korisnik dobija zadatak da nauči neki broj novih reči ili znakova, mogućnost da utvrdi naučeno putem igre pamćenja, vežbe pisanja ili prezentacije relevantnih zanimljivosti, da bi se na kraju njegovo znanje testiralo pomoću kviza. Sve aktivnosti koje su dostupne kao deo centralizovanog procesa učenja su dostupne i nezavisno čime je obezbeđena upotrebljivost aplikacije u raznim kontekstima učenja koji zavise od potreba i karakteristika korisnika i procesa učenja. Korisnik može da testira svoje znanje na listama reči kategorisanim po nekom kriterijumu, da vrši pregled svih znakova hiragana/katakana pisama i nekog broja kanđija i da pristupa njihovim pojedinačnim detaljnijim prikazima; može da pristupa kvizovima za kana i kanđi znakove i menja im

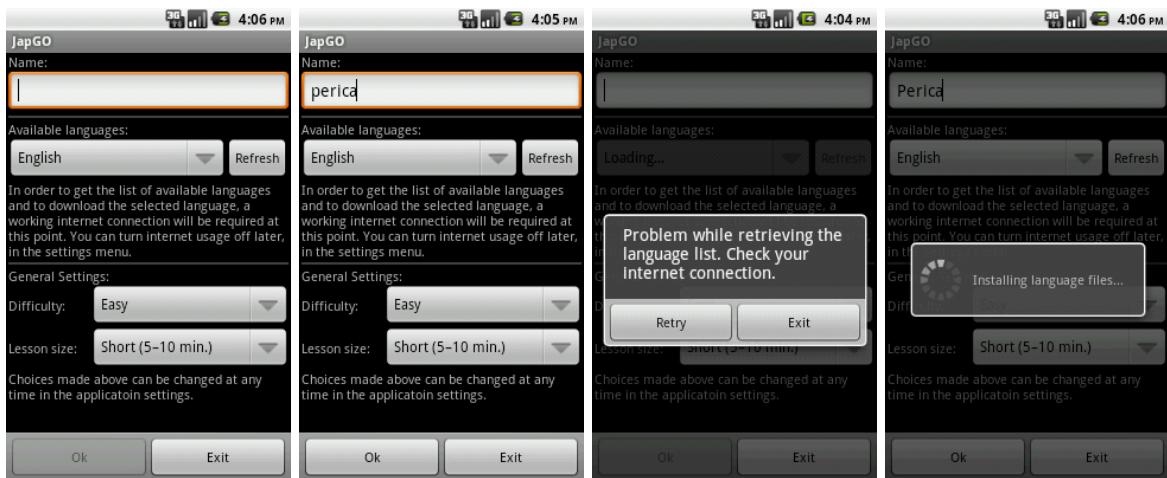
parametre po želji; može da vežba pisanje željenih znakova korišćenjem „touuh-screen“ mogućnosti. Na slici 19 prikazane su osnovne funkcionalnosti JapGO aplikacije.



Slika 19. Prikaz osnovnih funkcionalnosti FONJAPGO aplikacije

#### 4.3.1. Prvo pokretanje aplikacije

Pri prvom pokretanju aplikacije, korisniku se prikazuje ekran sa početnim podešavanjima aplikacije (slika 20). Ovde mu se traži unos imena, izbor jednog jezika od dostupnih koji će se dalje koristiti za interfejs i edukativne sadržaje i izbor težine i trajanja lekcija. Lista dostupnih jezika se dobavlja sa servera i zavisi od raspoloživih objekata učenja i jezika na kojem je njihov sadržaj (trenutno su obezbeđeni srpski i engleski). Izbor težine utiče na podešavanja aplikacije i utiče na stvari kao što su nagradni poeni, potrebni procenti za prelazak lekcije, broj odgovora po pitanju u kvizovima itd. Veličina lekcije određuje da li će se objekti učenja dostupni na serveru agregirati u veće objekte po osnovu svojih metapodataka u kojima su naglašene sve međusobne veze (ukoliko postoje). Ako dođe do nekog problema (pucanja veze i sl.), aplikacija će o tome obavestiti korisnika. Kada korisnik unese ime i pritisne dugme OK, aplikacija će se povezati sa serverom, dobaviti odgovarajući skup objekata učenja i persistirati ga u lokalnoj bazi podataka.



Slika 20. Početna podešavanja aplikacije

Po završetku instalacije, korisniku će biti prikazan početni ekran aplikacije (slika 21). Pri sledećim pokretanjima aplikacije, ovaj ekran će uvek biti prvi prikazan korisniku. Na ovom ekrani su izlistane sve lekcije dostupne korisniku. Pošto lekcije obuhvataju razne aktivnosti (učenje reči, znakova, vežbanje pisanja, testiranje znanja...) u nastavku će prvo biti prezentovane pojedinačne aktivnosti kojima je moguće pristupiti i nezavisno od lekcija, pa će tek onda biti prikazane kao deo centralnog procesa učenja.

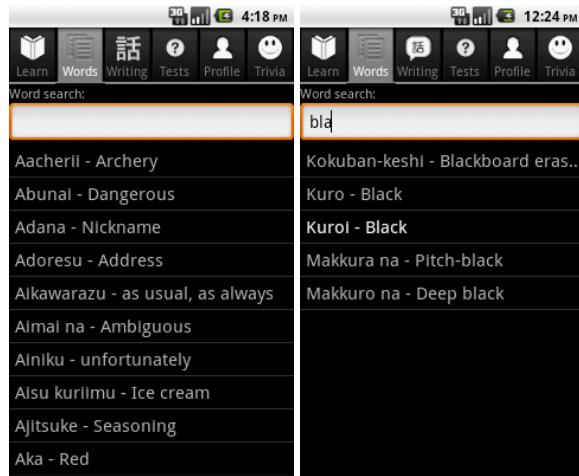


Slika 21. Početni ekran aplikacije

### 4.3.2. Učenje reči

Klikom na tab „Words“ (reči) korisniku se prikazuje odeljak za pregledanje reči (Slika 22). Centralni deo ovog odeljka predstavlja lista svih reči u bazi podataka koja je formata: „reč na japanskom – prevod“. Ova lista se može skrolovati, ili se može smanjiti broj reči na njoj unošenjem teksta u tekstualno polje za pretragu. Po unosu teksta, prikazaće se samo

rezultati koji u reči/prevodu sadrže uneti tekst. Reči koje su već predene radom na lekcijama će biti osvetljene. Klikom na neku od reči dobija se njen detaljniji prikaz (Slika 23).



Slika 22. Prikaz i pretraga reči iz baze podataka

Detaljni prikaz reči obuhvata prikaz reči ispisane latinicom, kana i kanđi pismima i njenih prevoda (može ih biti više). Ukoliko je korišćenje interneta dozvoljeno u podešavanjima aplikacije i ukoliko konekcija sa internetom postoji, aplikacija preuzima odgovarajući klip u kojem se izgovara zadata reč. Ako je audio klip za neku reč već preuzet skoro, ostaće sačuvan u privatnom prostoru aplikacije i upotrebljen bez ponovnog skidanja. Kompletan proces se obavlja u pozadini i ne ometa korisnika pri listanju reči. Ukoliko u podešavanjima aplikacije nije dozvoljeno korišćenje interneta, dugme za puštanje audio klipova će biti sakriveno.



Slika 23. Detaljan prikaz reči

#### 4.3.3. Sekcija za učenje pisama

Značajna pažnja u aplikaciji je posvećena japanskim pismima, hiragani, katakani i kandiju. Hiragana i Katakana su potpuno ekvivalentna jedno drugom pa se često uzimaju pod zajedničkim imenom „kana”, i tako su kategorisana i u aplikaciji.

Sekcija „Writing“ sadrži odeljke za prikaz i učenje kana i kanji znakova, za podešavanje i pokretanje igre pamćenja i za aktiviranje aktivnosti za vežbanje pisanja pomoću touch-screen funkcionalnosti. Na slici 24 je prikazan sadržaj sekcije „Writing“. Za svaki od odeljaka je data relevantna statistika – za učenje kana i kandji znakova je to broj naučenih znakova, za igru pamćenja broj odigranih igara, a za aktivnost vežbanja pisanja je to ukupno vreme provedeno u vežbanju.



Slika 24. Prikaz sekcije "Writing" pri: a) vertikalnoj orientaciji b) horizontalnoj orientaciji

#### 4.3.4. Učenje znakova

U okviru JapGO aplikacije, znaci japanskih pisama su podeljeni u dve kategorije – kanu (što obuhvata hiraganu i katakanu) i kandji. U skladu sa time obezbeđene su aktivnosti za učenje kana znakova i za učenje kandji znakova.

Izborom opcije „Learn Kana“ (učenje kane), korisniku će biti prikazan ekran na slici 25. Ova aktivnost sadrži tabelarni prikaz svih kana znakova (po 46 osnovnih za hiraganu i katakanu, oko 200 sa kombinacijama i tzv. „kvačicama“). Znaci su raspoređeni preko nekoliko stranica koje se mogu menjati i uređeni su u tabelarnom obliku. Kao kod pregleda reči, znaci koji su već predjeni u lekcijama su osvetljeni.

The image contains three separate screenshots of the JapGO application interface. Each screenshot shows a grid of Japanese characters (Kana) with their corresponding Romanized pronunciations and some additional information.

Basic Hiragana					Extended Hiragana					Basic Katakana						
a	i	u	e	o	ya	yu	yo	ki	きや	きゅ	きょ	a	i	u	e	o
あ	い	う	え	お	しゃ	しゅ	しょ	し	しゃ	ちゅ	ちょ	ア	イ	ウ	エ	オ
か	き	く	け	こ	ちや	ちゅ	ちょ	ch	ちや	ちゅ	ちょ	カ	キ	ク	ケ	コ
さ	し	す	せ	そ	にや	にゅ	によ	n	にや	にゅ	によ	サ	シ	ス	セ	ソ
た	ち	つ	て	と	ひや	ひゅ	ひょ	hi	ひや	ひゅ	ひょ	タ	チ	ツ	テ	ト
な	に	ぬ	ね	の	みや	みゅ	みょ	mi	みや	みゅ	みょ	ナ	ニ	ヌ	ネ	ノ
は	ひ	ふ	へ	ほ	りや	りゅ	りょ	ri	りや	りゅ	りょ	ハ	ヒ	フ	ヘ	ホ
ま	み	む	め	も	ぎや	ぎゅ	ぎょ	gi	ぎや	ぎゅ	ぎょ	マ	ミ	ム	メ	モ
や	り	ゆ	よ		じや	じゅ	じょ	ji	じや	じゅ	じょ	ヤ	ユ		ヨ	
ら	り	る	れ	ろ								ラ	リ	ル	レ	ロ

Slika 25. Neke od stranica sa kana znacima u okviru "Learn Kana" aktivnosti

Klikom na neki od znakova dobija se njegov detaljniji prikaz (slika 26). Ovaj prikaz uključuje: znak, njegovo čitanje, redosled povlačenja linija i neku reč iz baze reči koja sadrži prikazani znak. Ukoliko u bazi postoji još reči koje sadrže prikazani znak, mogu se prikazati klikom na „Another example“ dugme. Ako korisnik želi da promeni prikazan znak, može da koristi strelice za navigaciju date u vrhu ekrana.



Slika 26. Detaljan prikaz kana znakova

Ekvivalentno odeljku za kana znake, obezbeđen je „Learn Kanji“ odeljak koji sadrži određeni broj kandži znakova (Slika 27). Trenutno je u aplikaciji obezbeđeno samo 40 demonstrativnih znakova, dok dalji unos zavisi od kreiranih objekata učenja.



Slika 27. Stranice sa kandži znacima

Klikom na neki od prikazanih znakova, dobija se njegov detaljni prikaz (Slika 28). Detaljni prikaz kandžija se pomalo razlikuje od prikaza kana znakova zbog njihove prirode. Prikazani su: znak, redosled povlačenja linija, značenje te „on“ (izvorna kineska) i „kun“ (domaća japanska) čitanja ispisana i latinicom i hiraganom. Kao i u ranijim primerima, kod reči i kana znakova, može se prebaciti na sledeći/prethodni znak u bazi korišćenjem strelica za navigaciju u vrhu ekrana. Isto tako se može dobiti i prikaz još neke od reči iz baze podataka koja sadrži prikazani znak klikom na „Another Example“. Ako korisnik želi da provežba pisanje prikazanog znaka može da klikne na „Writing Pad“ i biće prebačen u aktivnost za vežbanje pisanja sa trenutno prikazanim znakom iscrtanim u vidu vodilja za vežbu.



Slika 28. Detaljan prikaz kandži znakova

#### 4.3.5. Igra pamćenja

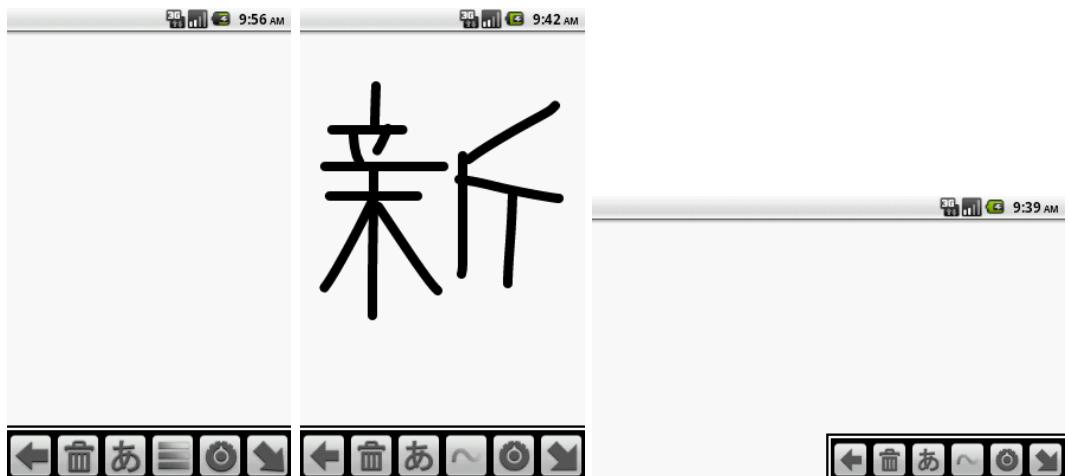
Kao dodatno sredstvo motivacije za učenje, korisniku je dostupna igra pamćenja u kojoj je neophodno povezati parove japanskih znakova i njihovih latiničnih zapisa. Potrebno je da korisnik izabere opciju „Memory Game“ čime se korisniku prikazuje aktivnost za podešavanje igre. Veličina table se može podešiti od 4\*4 do 10\*6 polja, a moguće je izabrati i set znakova koji će se koristiti za igru. Kada korisnik završi sa podešavanjem, klikom na „Start“ započinje igru (Slika 29.). Sama igra funkcioniše kao klasična igra memorije gde korisnik otvara po dve kartice za redom; ukoliko su različite zatvaraju se, a ukoliko su iste ostaju otvorene i korisnik dobija poene za njih.



Slika 29. Podešavanje igre pamćenja i dve različito podešene instance

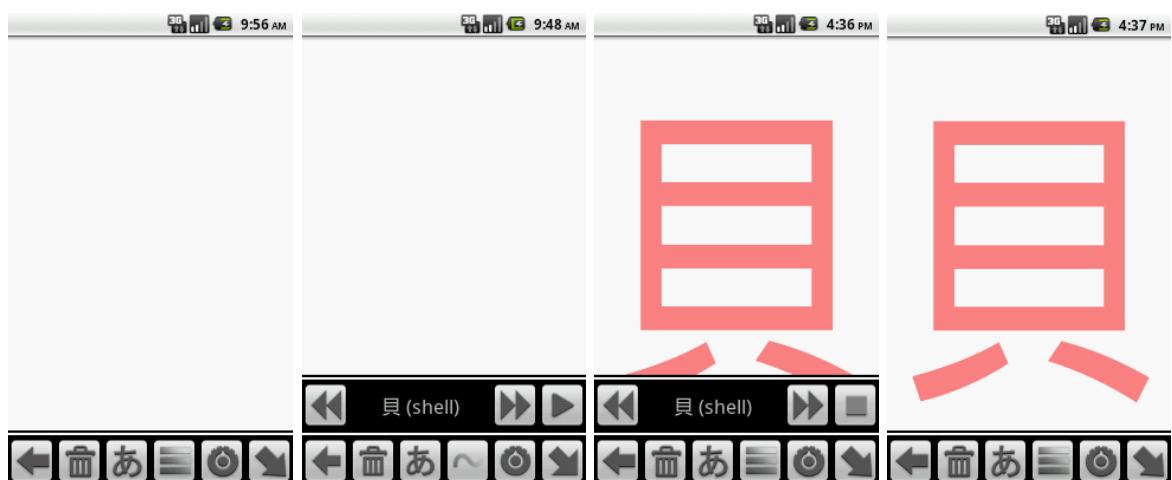
#### 4.3.6. Vežbanje pisanja

Zbog specifičnosti japanskih pisama i brojnosti njihovih znakova (oko 200 kana znakova i 2000+ kandija), za njihovo učenje je potrebna uvežbanost prilikom pisanja. JapGO aplikacija koristi „touch-screen“ mogućnost Android telefona i poseduje odeljak za pisanje znakova po ekranu (Slika 30). Kontrola ove aktivnosti se obavlja putem 6 dugmića u dnu ekrana. Prvo od dugmića (strelica koja pokazuje uлево) predstavlja „undo“ funkcionalnost koja briše poslednju povučenu liniju. Drugo dugme (kanta za otpatke) čisti čitav ekran.



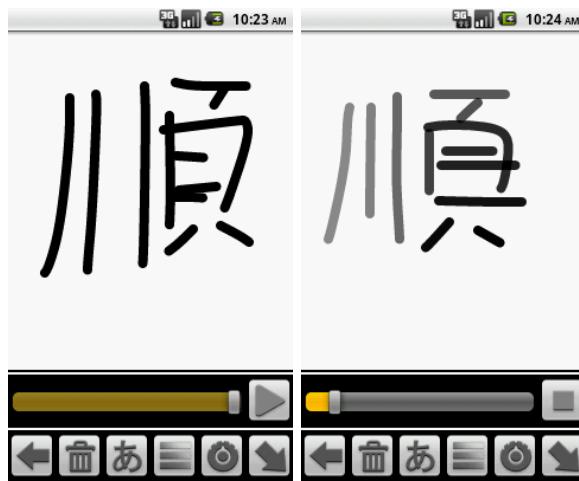
Slika 30. Aktivnost za vežbanje pisanja pri: a,b) vertikalnoj orijentaciji b) horizontalnoj orijentaciji

Korišćenjem sledećeg, trećeg dugmeta, korisnik može da prikaže odeljak za kontrolu linija vodilja i zatim uključi prikaz linija vodilja za neki znak po želji (Slika 31). Ako je aktivnost vežbanja pisanja startovana kroz neku od lekcija ili za neki od znakova, vodilje će biti automatski uključene; u varijanti sa lekcijama njihovo gašenje neće biti dozvoljeno. Trenutno prikazan znak se može menjati korišćenjem strelica levo/desno.



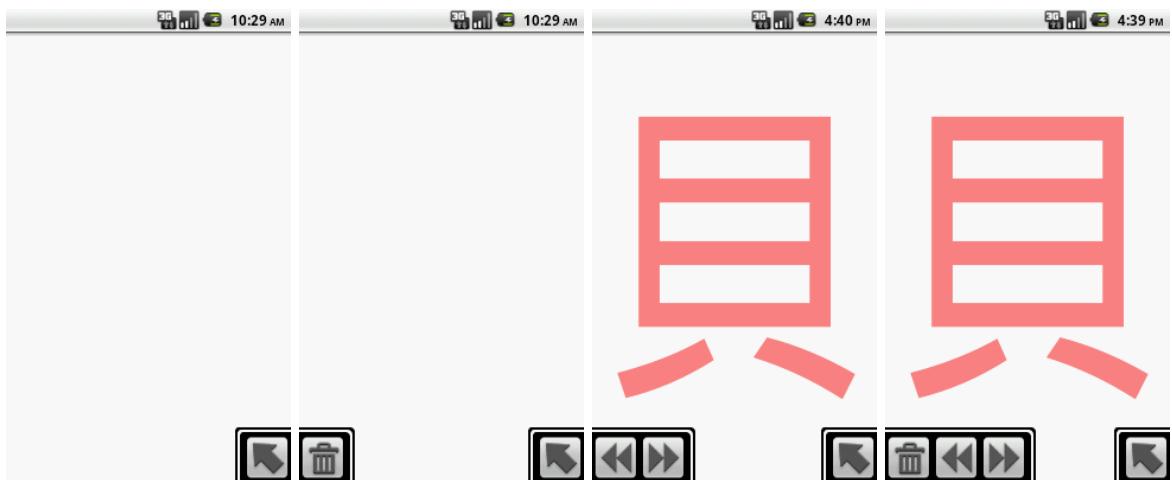
Slika 31. a) Početni prikaz b) Prikazan odeljak za kontrolu vodilja c) Uključene vodilje d) Sakriven odeljak za kontrolu vodilja

U ovoj aplikaciji „touch-screen“ funkcionalnost je potpuno prilagođena procesu učenja japanskih znakova tako što je omogućeno postepeno nestajanje povučenih linija brzinom koju korisnik odredi pomoću slajdera. Na ovaj način korisnik može mnogo brže da ispisuje znakove (ili jedan isti znak) za redom, bez zastajanja da obriše ekran. Za kontrolu „fade“ efekta korisnik pomoći četvrtog dugmeta prikazuje odgovarajući odeljak (Slika 32). Korišćenjem ponuđenih kontrola korisnik može da uključi/ugasi fade efekat ili da promeni brzinu nestajanja linija.



Slika 32. Aktivnost za crtanje sa ugašenim i uključenim fade efektom

Kako bi korisnik mogao da iskoristi najveći mogući prostor za vežbanje pisanja, obezbeđena je mogućnost minimizacije interfejsa korišćenjem poslednjeg, šestog dugmeta (strelica koja pokazuje dole desno). Klikom na ovu strelicu, interfejs se minimizira. U minimiziranom interfejsu ostaje prikazano samo nekoliko kontrola. Koje kontrole će biti prikazane zavisi od podešavanja interfejsa. Kontrola koja je uvek prikazana je strelica za maksimizaciju interfejsa; osim nje, mogu biti prikazani dugme za brisanje ekrana onda kada fade efekat nije uključen (pa je neophodan metod manualnog čišćenja ekrana) i strelice za menjenjanje prikazanih vodilja (kada su uključene vodilje) (Slika 33).



Slika 33. Izgled minimiziranog interfejsa zavisno od drugih podešavanja aktivnosti

Peto dugme prikazuje dugmiće za startovanje tri vrste različitih podešavanja – podešavanja „papira“, podešavanja interfejsa i podešavanja vodilja (Slika 34).



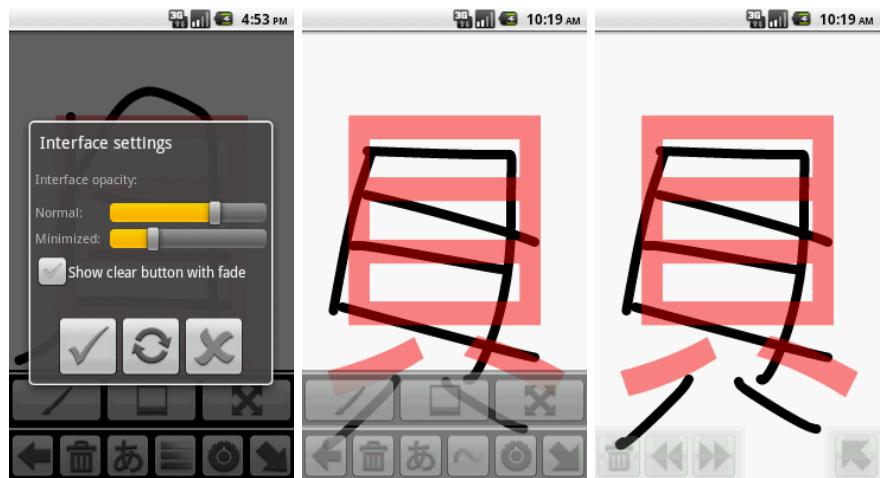
Slika 34. Podešavanja "papira", interfejsa i vodilja

U podešavanja papira (Slika 35) spada izbor boje farbe, vodilja, površine papira, debljine linija i Z-pozicije vodilja. Sva podešavanja se mogu jednim klikom vratiti na standardna. Na slici ispod su prikazana standardna i korisnička podešavanja i izgled aktivnosti koji odgovara tim podešavanjima.



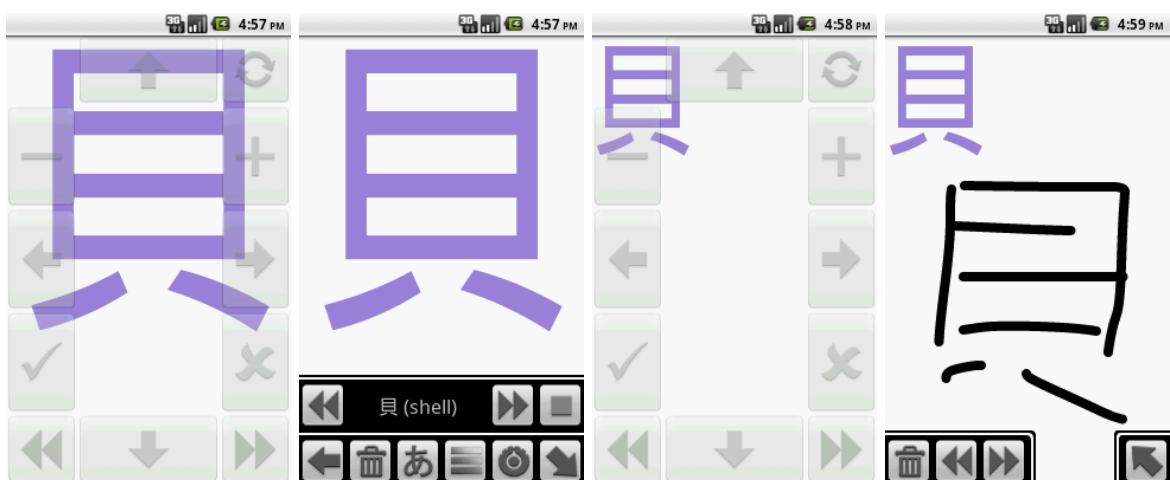
Slika 35. Podešavanja "papira" i odgovarajući izgled aktivnosti

Podešavanja interfejsa dozvoljavaju određivanje stepena njegove vidljivosti, što je još jedan način uvećavanja površine za crtanje (Slika 36).



Slika 36. Transparentnost maksimiziranog i minimiziranog interfejsa

Treća kategorija podešavanja omogućava promenu pozicije/veličine vodilja (Slika 37).



Slika 37. Pozicioniranje vodilja i izgleda aktivnosti po pozicioniranju

#### 4.3.7. Testiranje naučenog

U odeljku „Tests“ se može vršiti provera znanja naučenih reči, kana i kanji znakova pomoću testova koje je moguće detaljno podešiti. Podešavanja obuhvata izbor težine koja određuje broj odgovora na svako pitanje (od 4 do 10), izbor varijante pitanja (šta je pitanje, a šta je odgovor), izbor seta (naučeno/nenaučno) i broja reči/pitanja koja će biti postavljena korisniku (Slika 38 a,b).

Izgled testa će zavisiti od nameštenih podešavanja (Slika 38 c,d). Test se sastoji iz jednog pitanja i određenog broja odgovora. Za svako pitanje se vodi posebna evidencija koliko puta je prikazano korisniku i koliki je procenat tačnosti, što je i prikazano tokom izvođenja testa. Težina koju je korisnik izabrao na početku će uticati na broj ponuđenih odgovora kao i na

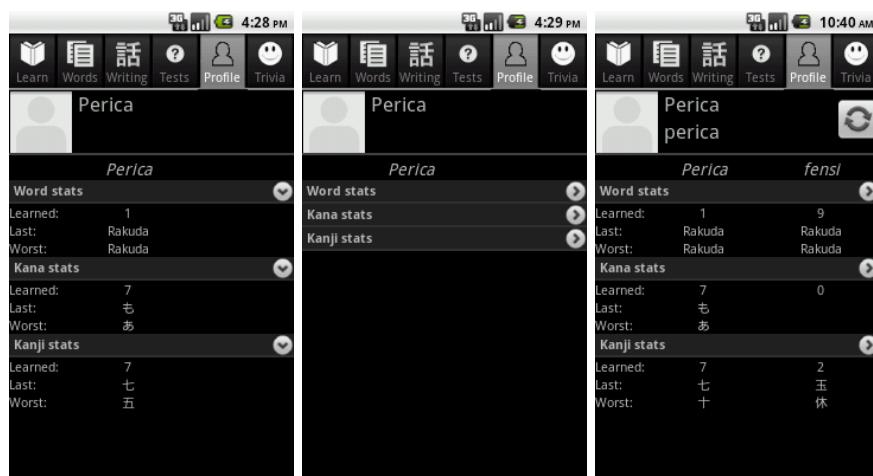
varijantu pitanja, koja mogu biti na japanskom (ispisana različitim pisimima) ili na odgovarajućem jeziku. Na kraju testa se prikazuje i ukupna statistika za izvedeno testiranje, koja obuhvata broj tačnih/netačnih odgovora i prosek rezultat ostvarenih na svim dotadašnjim pokušajima testa.



Slika 38. a) Odeljak "Tests" b) Podešavanja testa c,d) Instance testa sa različitim podešavanjima

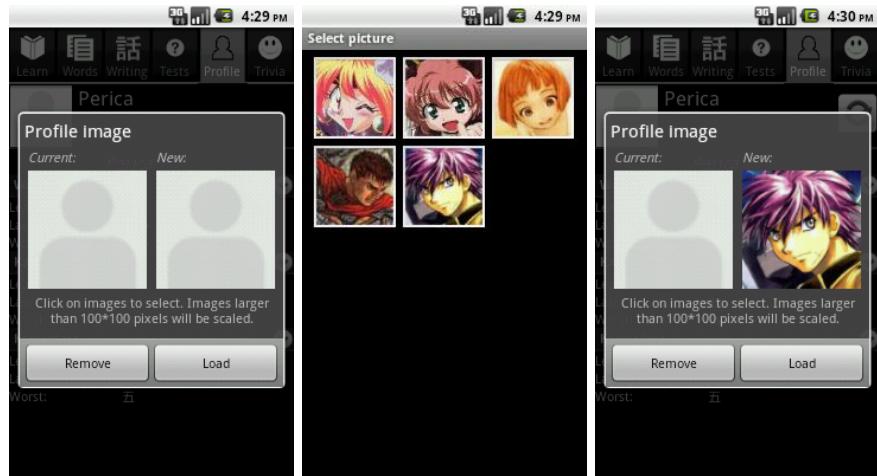
#### 4.3.8. Pregled i poređenje statistika korisnika

Jezičak „Profile“ daje pregled statistika korisnika i njegovih prijatelja upredo (ukoliko je registrovan i izabrao bar jednog prijatelja). Statistike su grupisane u kategorije koje se mogu sakriti/prikazati (Slika 39).



Slika 39. Pregled statistika

Klikom na sličicu profila (ili preko menija) korisnik može da prikaže dijalog za izmenu sličice. Može se postaviti bilo koja sličica manja od 100\*100 piksela koja se u tom trenutku nalazi na telefonu (Slika 40).



Slika 40. Izmena sličice profila

Da bi mogao da prikaže statistike drugih korisnika uporedno, korisnik mora prvo da se registruje. Dijaligu za registraciju se pristupa iz menija. Registracija je jednostavna i dobijeni nalog je vezan za mobilni uređaj korisnika pa nema potrebe za smišljanjem lozinke. Ukoliko je korisničko ime zauzeto, tražiće se unos novog od strane korisnika (Slika 41 a,b). Kada završi sa registracijom, korisnik može iz menija da izabere opciju „Add friends“ što mu prikazuje aktivnost za pretragu korisnika. Korisnik unosi tekst u dato polje i stiska „Search“; aplikacija će uspostaviti vezu sa veb serverom i dobaviti i prikazati paginiran spisak svih korisnika čije se ime poklapa sa unetim tekstrom (Slika 41 c,d).



Slika 41. a,b) Registracija korisnika c,d) Dodavanje prijatelja

#### 4.3.9. Sistem za nagrađivanje korisnika zanimljivostima o Japanu

Odeljak "Trivia" (Slika 42 b,c,d) omogućava korisniku pregled otključanih zanimljivosti o Japanu i japanskoj kulturi. Ove zanimljivosti su grupisane u nekoliko kategorija i otključavaju se u skladu sa korisnikovim napretkom (Slika 43 a) kao dodatni vid motivacije.

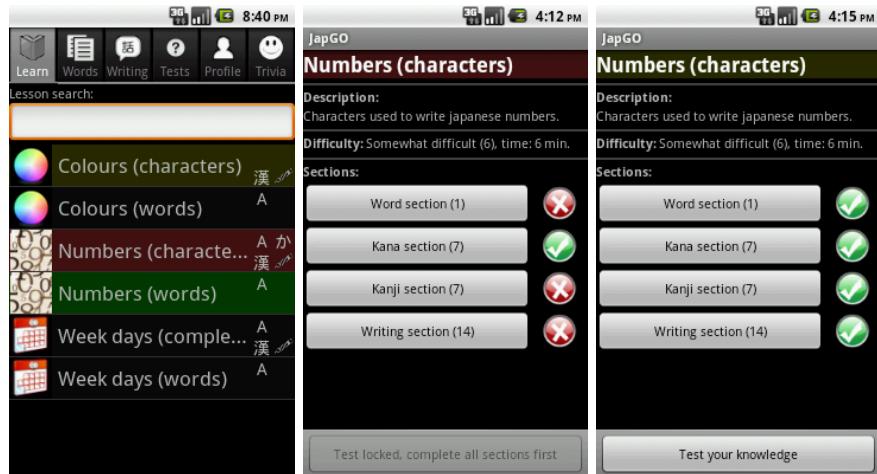


Slika 42. a) Otključavanje zanimljivosti tokom rada b) odeljak sa zanimljivostima c) prikaz jedne od kategorija d) prikaz jedne od zanimljivosti

#### 4.3.10. Centralni proces učenja

Prvi odeljak, "Learn" (Slika 43) prikazuje listu pristunih lekcija (objekata učenja). Prikazani su ime objekta, sličica (ukoliko postoji) i znaci sa desne strane koji opisuju sadržaj objekta. Zavisno od stanja u kom se objekat nalazi (neotvoren, otvoren, pređen edukativni sadržaj, kompletiran test), njegova stavka će biti obojena drugačijom bojom. Klikom na neki od objekata korisniku se prikazuju elementi sesije koje treba da savlada kako bi mogao da otključa test znanja koji zaokružuje sesiju. Elementi koji se ovde javljaju su dostupni i nezavisno u drugim delovima aplikacije, i objašnjeni su ranije.

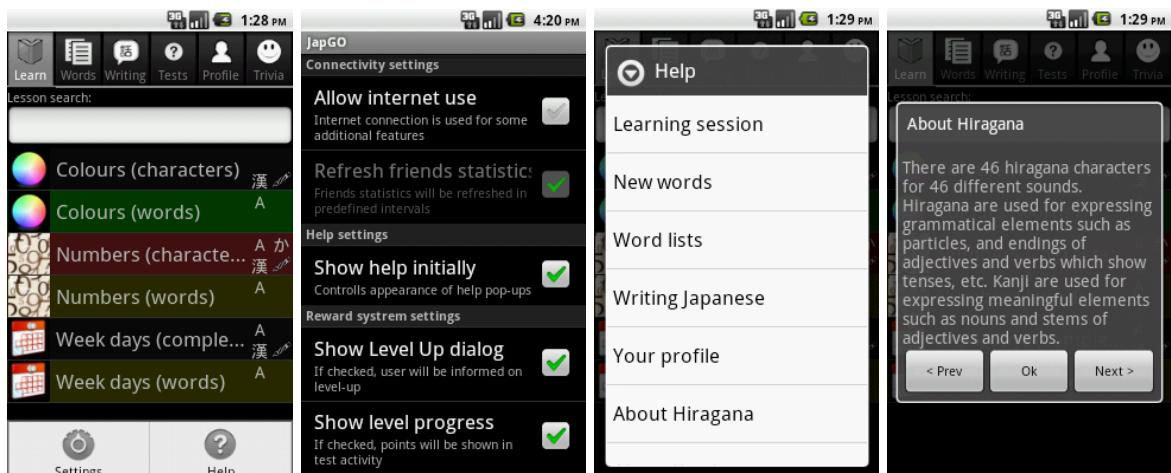
Odeljak "Learn" vodi korisnika kroz sve navedene aktivnosti, sa malim razlikama u načinu njihovog funkcionisanja. Pri nezavisnom pristupanju aktivnostima, dozvoljen je pristup bilo kom resursu, kao i svim podešavanjima aktivnosti. Pristupom kroz "Learn" odeljak, korisniku se prikazuju tačno određeni resursi po unapred definisanom redosledu i podešavanjima prilagođenim nivou težine objekta učenja. Kada korisnik pređe sve sekcije, otključava mu se kviz koji obuhvata pitanja vezana za novonaučene reči/znake. Ukoliko korisnik da pogrešan odgovor tokom sesije učenja, aplikacija prikazuje dijalog sa tačnim odgovorom.



Slika 43. Spisak raspoloživih lekcija i prikaz jedne od lekcija (u procesu/pređene)

#### 4.3.11. Podešavanja aplikacije & help sistem

Globalnim podešavanjima aplikacije i help sistemu se može pristupiti iz bilo koje od 6 glavnih sekcija aplikacije korišćenjem hardverskog „Menu“ dugmeta (Slika 44 a). U globalnim podešavanjima (Slika 44 b) je moguće onemogućiti korišćenje interneta aplikaciji, što će prekinuti update-ove korisničkih informacija i osvežavanje zanimljivosti, odrediti da li će help sekcije prikazivati same od sebe pri prvom ulasku u određeni deo aplikacije i da li će napredak korisnika biti prikazivan tokom testiranja znanja. Help sistem nudi određeni broj kategorija (Slika 44 c) koje se mogu pregledati pojedinačno (Slika 44 d).



Slika 44. a) Globalni meni, b) Podešavanja, c) Help kategorije, d) Pregled izabrane help teme

## 5. ZAKLJUČAK

Iako je trenutno engleski i dalje globalni jezik, razvoj ekonomije drugih zemalja kao što su Kina i Indija svakako vodi ka višejezičnom svetu i sigurno je da će u budućnosti biti sve više onih koji će govoriti po par stranih jezika. Zato je jasno da je polje primene modernih tehnologija u učenju jezika perspektivno i da se mogu očekivati razna unapređenja sa jedne strane i konvergencija postojećih tehnologija sa druge strane, pogotovo na prenosnim uređajima koji su do skoro predstavljali samo pomoćno sredstvo učenju a sada, razvojem platformi kao što je Android, imaju potencijala da direktno učestvuju u bilo kom obliku učenja.

Aplikacija JapGO efektivno demonstrira sposobnost Android platforme da podrži solidnu aplikaciju za učenje jezika, uz neke specifične pogodnosti kao što je „touch-screen“ tehnologija. Ova aplikacija se može pohvaliti solidnom bazom reči koja se korisniku prezentuje u odgovarajućim dozama prema logici aplikacije, kompletном pokrivenošću japanskih kana pisama i potpuno fleksibilnom kvizu za ponavljanje naučenih pojmoveva. Osim tih osnova, korišćen je „touch-screen“ za vežbanje ispisivanja znakova, uz neke inovativne pristupe kao što je postepeno nestajanje ranije povučenih linija radi olakšavanja ponavljanja i iscrtavanje linija vodilja. Takođe, iako aplikacija nije izgrađena oko upotrebe interneta, on je u njoj efikasno iskorišćen. Zahvaljujući tome što je edukativan sadržaj raspoređen u objekte učenja, postiže se fleksibilnost čitavog sistema po više parametara:

- Fleksibilnost jezika – dovoljno je samo dodati proizvoljan broj objekata učenja u onlajn skladište na željenom jeziku i aplikacije će automatski moći da ih koristi.
- Fleksibilnost dužine lekcija – korisnici mogu da prelaze kraće lekcije (objekte učenja) ili duže (agregirane)
- Fleksibilno gradivo – gradivo se može lako izmeniti, dopuniti i upotrebiti u drugom kontekstu.
- Fleksibilan proces učenja – redosled učenja, redosled prelaženja aktivnosti zavise samo od korisnika.

Takođe, audio fajlovi za pojedine reči se nalaze na internetu, time štedeći prostor na uređajima sa manje memorije; pošto aplikacija vrši tzv. „pretragu“ dostupnih audio fajlova, dovoljno je proširiti bazu audio zapisa na internetu da bi aplikacija bez ikakve promene mogla da ih koristi. Slično audio fajlovima i zanimljivosti o Japanu (koje mogu u sebi da nose multimedijalne sadržaje) se čuvaju na istom serveru i skidaju po potrebi. Na kraju, internet

se koristi i za jednostavan oblik povezivanja sa drugim korisnicima, pregledanjem njihovih rezultata, što se koristi kao oblik motivacije pri učenju.

Pored svega što je implementirano, postoje i brojne mogućnosti za proširenje JapGO aplikacije. Prvo, uvođenje jače komponente društvenog umrežavanja među korisnicima, npr. omogućavanjem razmene poruka među rivalima ili kreiranjem multiplejer igara/takmičenja u znanju; unapređenje statistika korisnika primenom raznih grafičkih skala, praćenjem napretka i nivoa; integracija veb aplikacije sa statistikama u sajt koji bi ih prikazivao grafički i formirao razne rang-liste; dodavanje specijalnih periodnih lista reči koje bi se skidale sa interneta i funkcionalne kao posebna takmičenja među svim korisnicima nekog nivoa; omogućavanje korisnicima da doprinesu dodavanjem reči i spravljanjem specifičnih lista koje bi se slobodno skidale i integrisale u aplikaciju; napredno prepoznavanje redosleda vučenja linija i oblika iscrtanih znakova i ocenjivanje korisnika na osnovu toga... Mogućnosti su brojne, i iako je tržište ovakvih aplikacija već solidno popunjeno, i dalje postoji prostor za inovativne ideje koje bi mogle da privuku korisnike.

## LITERATURA

- [1.] O'Malley C., Vavoula G., Glew J. P., Taylor J., Sharples M., Lefrere P.: „WP4 – Guidelines for learning / teaching / tutoring in a mobile environment", MOBILearn/UoN, UoB, OU/D4.1/1.0, 2003
- [2.] Seta L., Gentile M., Taibi D., Arrigo M., Fulantelli G., Novara G., Giuseppe O. Di: „Multimodality in a Mobile Learning Environment", 19th International Conference on Database and Expert Systems Application, IEEE, DOI 10.1109/DEXA.2008.66
- [3.] Lavín-Mera P., Moreno-Ger P., Fernández-Manjón B.: „Development of educational videogames in m-Learning contexts" ,Second IEEE International Conference on Digital Games and Intelligent Toys Based Education, DOI 10.1109/DIGITEL.2008.21, IEEE, 2008
- [4.] Keegan D., „Mobile Learning: The Next Generation of Learning", Distance Education International, 2005
- [5.] Traxler J., Kukulska-Hulme A., „Evaluating Mobile Learning: Reflections on Current Practice", Proceedings of mLearn 2005, Cape Town, S. Africa, 25-28 October 2005
- [6.] Keegan D.: „The Incorporation of Mobile Learning into Mainstream Education and Training", mLearn 2005, Cape Town, S. Africa, 25-28 October 2005
- [7.] Savill-Smith, C. and Kent, P., "The Use of Palmtop Computers for Learning", 2003
- [8.] Sharples M., Taylor J., Vavoula G., "Towards a Theory of Mobile Learning", Proceedings of mLearn 2005, Cape Town, S. Africa, 25-28 October
- [9.] Jones, A., Issroff, K., Scanlon, E., Clough, G., and McAndrew, P., "Using mobile devices for learning in informal settings: Is it motivating? ", Paper presented at the IADIS International conference mobile learning., Dublin, Ireland, 14-16 July, 2006.
- [10.] H.-R. Chen and H.-L. Huang, "User Acceptance of Mobile Knowledge Management Learning System: Design and Analysis", Educational Technology & Society. vol. 13, No.3, pp 70-77, 2010.
- [11.] M. Wang, R. Shen, D. Novak, and X. Pan, "The impact of mobile learning on students' learning behaviours and performance: Report from a large blended classroom", British Journal of Educational Technology, vol. 40, No.4, pp 673-695, 2009.
- [12.] G.M. Chinnery, "Going to the MALL: Mobile assisted language learning", Language Learning & Technology, Vol. 10, No. 1, pp 9-16, 2006.
- [13.] R. Godwin-Jones, "Mobile apps for language learning", Language Learning & Technology, Vol. 15, No. 2, pp 1-11, 2011.

- [14.] L.-H. Wong, C.-K. Chin, C.-L. Tan, and M. Liu, "Students' Personal and Social Meaning Making in a Chinese Idiom Mobile Learning Environment", *Educational Technology & Society*, Vol. 13, No. 4, pp 15-26, 2010.
- [15.] S.-C. Cheng, W.-Y. Hwang, S.-Y. Wu, R. Shadiev, and C.-H. Xie, "A Mobile Device and Online System with Contextual Familiarity and its Effects on English Learning on Campus", *Educational Technology & Society*, Vol. 13, No. 3, pp 93-109, 2010.
- [16.] N.-S. Chen and Y. Wang, "Testing Principles of Language Learning in a Cyber Face-to-Face Environment", *Educational Technology & Society*, Vol. 11, No. 3, pp 97-113, 2008.
- [17.] Sandberg J., Maris M., and de Geus K., "Mobile English learning: An evidence-based study with fifth graders", *Computers & Education*, vol. 57, No.1, pp 177–193, 2011
- [18.] Twarog, L., & Pereszlenyi-Pinter, M. (1988). Telephone-assisted language study and Ohio University: A report. *The Modern Language Journal*, 72, 426-434.
- [19.] Thornton, P., & Houser, C. (2005). Using mobile phones in English Education in Japan. *Journal of Computer Assisted Learning*, 21, 217-228.
- [20.] Brown, E. (Ed.) (2001, January 8). Mobile learning explorations at the Stanford Learning Lab. Speaking of Computers, 55. Stanford, CA: Board of Trustees of the Leland Stanford Junior University.
- [21.] Samuels, J. (2003, August 15). Wireless and handheld devices for language learning. *Proceedings of the 19th Annual Conference on Distance Teaching and Learning*, Madison, WI.
- [22.] Belanger, Y. (2005, June). "Duke University iPod first year experience final evaluation report".
- [23.] T. de Jong, M. Specht, and R. Koper, "A Study of Contextualised Mobile Information Delivery for Language Learning", *Educational Technology & Society*, Vol. 13, No. 3, pp 110-125, 2010.
- [24.] S.H.R. Joseph and M. Uther, "Mobile devices for language learning: multimedia approaches", *Research and Practice in Technology Enhanced Learning*, Vol. 4, No. 1, pp 7-32, 2009.
- [25.] F.B. Anaraki, "A Flash-Based Mobile Learning System for Learning English as a Second Language", *Proceedings of the 2009 International Conference on Computer Engineering and Technology*, IEEE Computer Society, Washington, DC, USA, 2009. [
- [26.] C.-M. Chen and S.-H. Hsu, "Personalized intelligent mobile learning system for supporting effective English learning", *Educational Technology and Society*, vol. 11, No.3, pp 153-180, 2008.

- [27.] S.A. Petersen and J.K. Markiewicz, "PALLAS: Personalised language learning on mobile devices", Proceedings of the Fifth IEEE International Conference on Wireless, Mobile, and Ubiquitous Technology in Education, IEEE Computer Society, Washington, DC, USA, 2008.
- [28.] T. Vidas, C. Zhang, and N. Christin, "Toward a general collection methodology for Android devices", Digital Investigation, Vol. 8, Supp.1, pp 14-24, 2011.
- [29.] IEEE Learning Technology Standards Committee (LTSC), Draft Standard for Learning Object Metadata, IEEE 1484.12.1-2002, 15. July 2002,  
[http://ltsc.ieee.org/wg12/files/LOM\\_1484\\_12\\_1\\_v1\\_Final\\_Draft.pdf](http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf)
- [30.] L'Allier, James J.: "Frame of Reference: NETg's Map to the Products, Their Structure and Core Beliefs.", 1997.
- [31.] Polsani, P.: "Use and abuse of reusable learning objects", Journal of Digital Information, 3 (4), Article 16, 2003.
- [32.] Rory McGreal: "Learning Objects: A Practical Definition", The International Journal of Instruction Technology & Distance Learning. Duquesne University, Pittsburgh PA, September 2004.
- [33.] Longmire, W., A primer on learning objects, ASTD Learning Circuits, 2000.
- [34.] Caws, C., Friesen, N., & Beaudoin, M.: „A new learning object repository for language learning: Methods & possible outcome”, Interdisciplinary Journal of Knowledge and Learning Objects, 2, pp 111-124, 2006
- [35.] Churchill, D.: "Conceptual Model Learning Objects and Design Recommendations for Small Screens", Educational Technology & Society, 14 (1), pp 203–216, 2011.