SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Seminarski rad iz otvorenog računarstva

Lua

Stjepan Antivo Ivica

Voditelji: Mr. sc. Siniša Tomić, dipl. ing. Vlatka Paunović

Mentor: prof.dr.sc. Mario Žagar

Zagreb, kolovoz 2012.

Sadržaj

1. Uvod

Mobilna aplikacije jest programska podrška namijenjena pametnim mobitelima i ostalim mobilnim uređajima. Mobilne aplikacije najčešće pomažu svojim korisnicima da olakšano koriste razne internet usluge kao i obavljanje ostalih smislenih radnji koje imaju stvarnu korist za korisnika uređaja na kojem se pokreću.

Cilj ovog seminara jest izgraditi mobilnu aplikaciju koja će svome korisniku omogućiti pristup dvojezičnom (hrvatski i engleski jezik) informacijskom paketu. Namjena aplikacije jest da trenutni, ali i budući studenti mogu pomoću nje pregledavati predmete, studijske programe te pomoći im pri odluci koje predmete će upisati. Izrađena aplikacija mora biti pod nekim oblikom licencije otvorenog koda.

Podatci informacijskog paketa pohranjeni su SQLite bazi podataka. Baza podataka je prethodno pripremljena te se rad aplikacije temelji na sadržaju i prikazu podataka koji se nalaze u bazi podataka. Baza podataka se sastoji od šesnaest tablica u kojima su raspodijeljeni razni korisni podatci o predmetima te dodatnim podatcima koje se vežu na postojeće predmete, nastavnom osoblju koje sudjeluje u odgovarajućim predmetima i podatci o smjerovima na kojima se predmeti nalaze. Pregledavanje predmeta u informacijskom paketu treba biti omogućeno prema nekom skupu parametara.

Zahtjevi na napisanu aplikaciju su da mora biti napisana u programskom jeziku Lua te da mora raditi na barem jednoj od traženih platformi. Tražene platforme su iPhone i Android. Dodatni poželjni zahtjevi koji su prepušteni slobodnom izboru su mogućnost izrade aplikacije za više platformi te nadogradivost aplikacije na aplikaciju koja bi pokrivala više funkcionalnosti vezanih uz sadržaje i potrebe studenata Fakulteta elektrotehnike i računarstva.

2. Lua

Lua je višeparadigmatski skriptni jezik, visoke apstraktne razine dizajniran da ima maleni memorijski otisak s ciljem proširive semantike. Jezik nije vezan uz pojedinu platformu budući da je napisan u jeziku ISO C.

Dijelove koda koje Lua izvodi nazivamo komadima (eng. *chunks*). To su građevne jedinice koda od kojih gradimo Lua program. Komadi mogu biti jedna linija koda ili čitava datoteka. Takvi komadi mogu biti jednostavni ili složeni, memorijski veliki ili mali, a o tomu odlučuje programer. Pojednostavljeno govoreći komad je samo organiziran redoslijed naredbi.

Jezik je slabo tipizirana tj. prevodioc dopušta implicitne prijelaze između tipova, a sami tipovi varijabli se ne navode (Lua prevodioc ih zna prepoznaje prema kontekstu tj. vrijednosti varijable). Tipovi se određuju dinamički.

Pri stvaranju globalne varijable nužno ju je definirati (nemoguće je deklarirati globalnu varijablu). Lokalne varijable mogu se deklarirati ključnom riječju *local*.

Čitanje neinicijalizirane varijable je dopušteno, no pročitana vrijednost će biti *nil*. To je posebna vrijednost osmišljena kako bi bila različita od svake druge vrijednosti. Svrha joj je predstavljati stanje nevrijednosti tj. odsutnosti korisne informacije. Sve varijable prije inicijalizacije su postavljene na vrijednost *nil*.

Nije moguće eksplicitno brisati varijable. Ako su kratkog vijeka trajanja tada ih treba označiti kao lokalne varijable. Globalne varijable možemo predodrediti za brisanje iz memorije dodjeljujući im vrijednost *nil* nakon čega će u nekom trenutku sustav za upravljanje memorijom (sakupljač smeća) osloboditi zauzet prostor.

Jezik ima vrlo korisno osmišljen način komentiranja koda. Redoslijed znakova „--“ označava jednolinijske komentare, dok višelinijski komentari trebaju omeđiti redoslijedima znakova „--[[“ te „--]]“. Vrlo je korisno to što se učinak višelinijskih komentara može ukloniti dodavanjem još jednog znaka „-“ u redoslijed znakova koji označava početak višelinijskog komentara (tada je kod omeđen s „---[[“ i „--]]“

2. 1 Tipovi

Boolean tip ima dvije vrijednosti: istina (*true*) i laž (*false*). Taj tip predstavlja (dobro poznati) logički tip podataka i kao takav najčešće se koristi u izrazima koji sadrže uvjete. Valja pripaziti da u Lui svaka vrijednost može predstavljati uvjet. Tako se *false* i *nil* smatraju pod neispunjenim uvjetom dok se svaka druga vrijednost smatra ispunjenim uvjetom (kao nula i prazni string).

Jedini brojevni tip u ovom jeziku su realni brojevi dvostruke preciznosti. Lua nema cjelobrojne vrijednosti. Motivacija iza toga leži u činjenici da ako se koristi dvostruka preciznost neće nastati greška zaokruživanja (osim ako je broj čak mnogo veći od bilijardu) te zbog toga što mnogi današnji procesori rade aritmetičke operacije brže s brojevima s pomičnim zarezom nego što to čine s cjelobrojnim brojevima. Valja napomenuti da je vrlo lagan postupak prevođenja programa napisanog u Lui tako da koristi neki drugi tip za brojeve kao što su cjelobrojni ili realni brojevi jednostruke preciznosti (što je vrlo korisno kada se koristi platforma na kojoj nema podrške za realne brojeve s dvostrukom preciznosti). Brojevni tip se implicitno ovisno o kontekstu može implicitno pretvoriti u niz znakova.

Niz znakova (eng. *string*) je nepromjenjivi tip podataka. Nizovi znakova u Lua programskom jeziku mogu sadržavati sve znakove uključujući i null znak (što znači da u taj tip podataka možemo spremati bilo kakve binarne podatke poput čitavih datoteka). Valja napomenuti da prevodioc ovisno o kontekstu niz znakova se može implicitno pretvoriti u broj (ako se u potpunosti sastoji od znamenaka).

Tablice su tip koje implementiraju asocijativno polje. Mogu se indeksirati bilo kojim znakom ili nizom znakova ili bilo kojom drugom vrijednosti koja postoji u jeziku osim vrijednošću *nil*. Nemaju unaprijed određenu vrijednost, a dodavanje elemenata se obavlja dinamički. Tablice su jedini tip koji koristimo kako bi modelirali podatkovne strukture. Koristimo ih za predstavljanje skupove, nizove, strukture, redove pa čak i programske module, no i mnoge druge podatkovne strukture. Tablice nisu ni vrijednosti ni varijable, one su objekti. Tablice je nemoguće deklarirati, mogu se samo definirati preko svog konstruktora. Pri usporedbi varijabli koje sadrže tablice, ako koristimo operator „==“ Lua će usporediti reference tih varijabli, a ne vrijednosti tablica.

Funkcija je građanin prvog reda što znači da je ravnopravna ostalim tipovima podataka (može se spremati u varijable) te može biti povratna vrijednost ili argument druge funkcije. Funkcije se mogu gnijezditi s pristupom leksičkom djelokrugu tj. svaka ugnježđena funkcija ima pristup varijablama prema funkciji koja ju ugnježđuje (takve varijable nazivamo eksterne lokalne varijable). Lua može pozivati funkcije iz standardne C biblioteke (budući da je napisana u tome jeziku). Sve funkcije u jeziku Lua su anonimne. Pri usporedbi varijabli koje sadrže funkcije, ako koristimo operator „==“ Lua će usporediti reference tih varijabli.

Korisniku je omogućeno spremati proizvoljan tip podataka iz C jezika u varijable, no takvi podatci nemaju definirane operacije osim jednakosti i dodjeljivanja. Također jezik podržava višedretvenost što nije pokriveno u ovom radu.

2.2 Integrirano razvojno okružje (eng. *IDE*)

Kako bi se sam program napisan u jeziku Lua preveo u strojni kod dovoljan je ANSI C kompajler (npr. *gcc*). Isprobano je više okružja za pisanje Lua programa. Odabrano razvojno okružje jest Lua Develompent Tools (LTD). LTD je zasnovan na višejezičnom okružju Eclipse. Okružje je dostupno na operacijskim sustavima Windows, Mac OS te Linux.

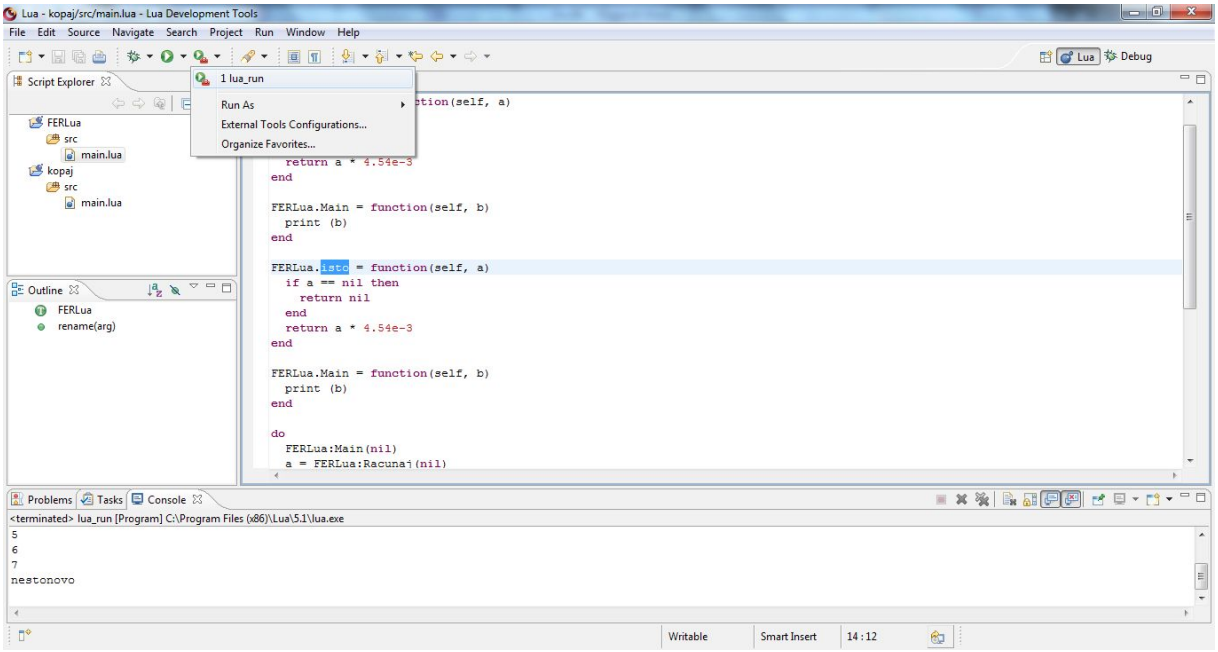
2.2.1 Postupak Instalacije

Na stranici [11] dovoljno je skinuti distribuciju za odabran operacijski sustav. Distribucija je arkiva veličine oko šezdeset megabajta. Nakon što korisnik preuzme arhivu treba ju otpakirati i pokrenuti izvršnu datoteku LuaDevelopmentTools.

Samo okružje je još uvijek u razvoju. Prema riječima razvojnog tima [3] rad koji se obavlja nad projektom je fokusiran na korisnički doživljaj te trenutno ne postoji namještena konfiguracija za prevođenje/izvođenje programa. (Dakle okružje će upozoriti na greške, organizirati kod, ali se zasad ništa ne može prevesti/pokrenuti.)

Korisnik može sam namjestiti prevodioc koristeći External Tools. Na stranici [12] je opisan općenit postupak dodavanja vanjskih alata. Ono što korisnik treba napraviti je dodati Lua virtualnu mašinu kao prevodioc nakon čega će moći

prevesti/pokrenuti program. Razvojno okružje je prikazano na Slika 1. Integrirano razvojno okružje Lua Develomplent Tools.



Slika 1. Integrirano razvojno okružje Lua Develomplent Tools

3. Integrirano razvojno okružje za izradu mobilne aplikacije

Pri odabiru integriranog razvojnog okružja za izradu mobilnih aplikacija postoje određeni zahtjevi. Najvažniji zahtjev je da treba biti pod nekom open source licencom. Okružje je treba biti dostupno na operacijskom sustavu Windows (alternativno Linux). Treba imati prevodioc koji će projekt prevesti u aplikaciju za ciljanu platformu. Također treba imati emulator mobilnog uređaja na kojemu se može vrtjeti zadani projekt tj. aplikacija. Pri pretrazi prostora mogućih rješenja prihvatljiva su se smatrala i ona rješenja u kojima različite komponente (prevodioc, emulator ..) nisu integrirane u jednom programu. Sekundarni zahtjevi koje je bilo poželjno ispuniti jest da oružje podržava višeplatformsku izradu. Spomenimo neka rješenja koja nisu zadovoljila navedene zahtjeve: Corona, Lua Development Tools + virtualna mašina Lua + Corona, Moai, PhoneWax, Android Scripting Environment itd.

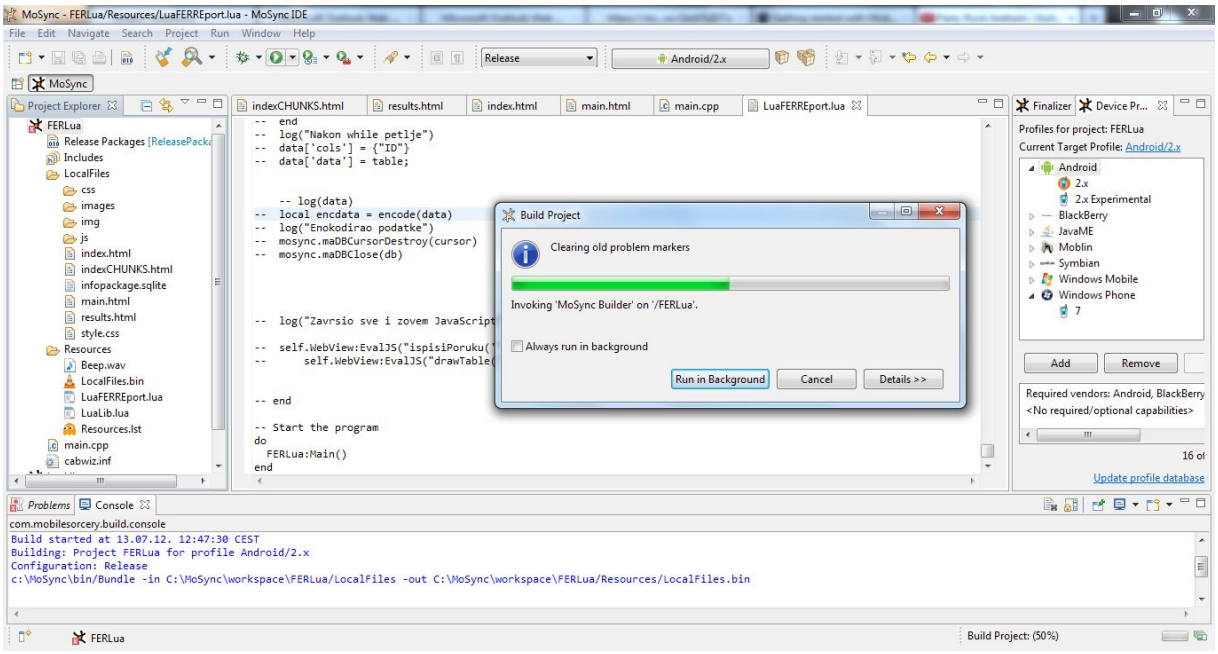
Pronađeno rješenje koje zadovoljava sve zahtjeve jest integrirano razvojno okružje MoSync uz korištenje projekta MobileLua.

3.1 MoSync

MoSync [4] je višeplatformski alat za razvoj mobilnih aplikacija. Dostupan je na operacijskim sustavima Windows te Mac OS. Aplikacije se pišu u jezicima C/C++ te HTML5/JavaScript. MoSync posjeduje dvije licence. GPL 2 licencu (open source licenca) te komercijalnu licencu. Sve funkcionalnosti potrebne za ovaj seminar su pokrivene u MoSync SDK pod GPL 2 licencom. Izgled alata MoSync se može vidjeti na Slika 2. Integrirano razvojno okružje MoSync.

Aplikacije napisane u MoSyncu i prevedene za ciljanu platformu se ne razlikuju od aplikacija napisanih u razvojnim alatima te ciljane platforme. MoSync ima svoj emulator mobilnog uređaja, ali na tom emulatoru nije podržana Native UI biblioteka (grafička biblioteka samog operacijskog sustava za iscrtavanje grafičkih elemenata). Jedini uređaji za koje trenutno radi pristup Native UI bibliotekama su Android, iPhone te Windows Phone.

Za MoSync je razvijen Wormhole Library. To je tehnologija koja omogućuje prebacivanje toka izvođenja programa između koda napisanog u C/C++ i koda napisanog u HTML5/JavaScript. Trenutno je tehnologija podržana na platformama Android, iPhone te Windows Phone.



Slika 2. Integrirano razvojno okružje MoSync

3.2 MobileLua

MobileLua je most između programskog jezika Lua i okružja za razvoj mobilnih aplikacija MoSync. Trenutno MobileLua podržava samo uređaje Android i iOS.

MobileLua API se sastoji od tri dijela. *Lua API* je sučelje visoke apstraktne razine koja radi sa Lua objektima za upravljanje događajima, interakcijom s korisnikom, NativeUI i mrežne komunikacije. Ova biblioteka se proširuje u svakoj novoj verziji MobileLue. Drugi dio Lua System API je biblioteka na razini sustava koja nudi pomoćne funkcije za komunikaciju sa strukturama podataka na razine jezika C. Funkcije unutar ove biblioteke su predznačene sa riječju *Sys* i većina pripadajućih funkcija je napisana u jeziku C/C++. Posljednji dio MoSync API koji se sastoji od

funkcija koje pristuapaju uslugama samog mobilnog uređaja kao pozivi, tekstualne poruke, korisničko sučelje, OpenGL, povezivanje s okolinom, datotečni sustav, senzori itd. Većina funkcija unutar ove biblioteke je predznačena riječju *ma*. MoSyncove C++ biblioteke nisu dostuopne Lui. Umjesto toga odgovarajuće funkcije su napisane koristeći programski jezik Lua s odgovarajućim funkcionalnostima.

Budući da Lua podržava Native UI te WebView dodatke moguće je koristiti Luu uz HTML5/JavaScript. Lua tada postaje jezik domaćin za WebView sastavljen od dokumenata napisanih HTML-u i JavaScriptu. Lua poput JavaScripta je dinamični jezik što znači da se napisani kod prevodi tijekom izvođenja. Poruke JavaScripta možemo slati direktno prema razini jezika C, što znači da možemo te iste poruke proslijediti i na razinu jezika Lua. Jednako tako Lua može pozvati izvođenje zadane JavaScript funkcije.

3.3 Postupak instalacije integriranog razvojnog okružja

Preuzeti Mosync s web stranice [9] i instalirati ga. Odabrana verzija MoSynca mora biti verzija 3.0 ili novija zbog kompatibilnosti s trenutnom verzijom MobileLue. Datoteke izvornih kodova projekta MobileLua treba preuzeti na sljedećoj stranici [10]. Potom treba pokrenuti MoSync IDE nakon čega se korisnik može učlaniti u zajednicu i sudjelovati u njoj (preporučljivo je registrirati se).

MobileLua se uvodi u okružje tako da se uvede barem projekt LuaLib. Projekt se uvode odabirom File/Import/Existing MoSync project into workspace nakon čega se mogu odabrati projekti unutar preuzetih datoteka projekta MobileLua i uvesti u radno okružje.

Lua interpreter se nalazi u projektu pod nazivom LuaLib. To je biblioteka koju koriste svi MobileLua projekti. Projekt mora biti pokrenut barem jednom kako bi načinio potrebne datoteke za povezivanje programskoj jezika Lua i okružja MoSync.

Za dodavanje uređaja osim ugrađenog MoSync emulatora MoRE (MoSync Runtime Environment) postupak se može naći na [8].

Kako bi se koristio emulator uređaja iPhone dovoljno je instalirati MoSync distribuciju namijenjenu OS X operacijskim Apple Mac računalu. Ako postojeći OS X operacijski sustav nema sadržan iPhone emulator tada treba dodatno instalirati iPhone SDK uključujući Xcode 4.

Android emulator se može ugraditi u obje postojeće distribucije MoSynca (za Windows i OS X). Nakon instalacije bilo koje od distribucija prvi korak je skinuti Android SDK i instalirati ga prateći upute na [13]. Zatim koristeći AVD manager iz Android SDK alata instalirati barem jednu platformu za emulaciju pri tom valja paziti da se željena platforma nalazi na popisu [14]. Nakon instalacije platforme unutar MoSync okružja koristeći MoSync alate treba dodati stazu gdje se nalazi Android SDK.

3.4 Korištenje integriranog razvojnog okružja

MoblileLua projekti zasigurno se sastoje od barem tri dijela. Prvi dio je datoteka main.cpp. To je C++ program koji treba inicijalizirati Lua interpreter i pokrenuti Lua aplikaciju. Drugi dio je sama Lua aplikacija koja može biti sadržana unutar više modula i biblioteka napisanih u Lua programskom jeziku. Treći dio je datoteka resource.lst koja iako je opcionalna, preporučeno ju je imati u projektu. To je datoteka koja sadrži popis svih resursa koje program koristi. Lua kod je upakiran kao resurs koji se otpakirava prilikom pokretanja aplikacije.

Svaki se projekt može prevesti za ciljanu platformu mobilnog uređaja. Dovoljno je označiti željeni projekt i uređaj u popisu dostupnih uređaja i pokrenuti ga.

4. Mobilna aplikacija FEatheR

Izrađena mobilna aplikacija nazvana FEatheR zato što je ideja vodilja pri izradi ove aplikacije bila napraviti aplikaciju za minimalnim memorijskim otiskom i pri tom ostvariti učinkovite i efikasne mehanizme. Uz to istaknuta slova trebaju podsjećati na skraćenicu FER koja predstavlja Fakultet elektrotehnike i računarstva. Aplikacija je dostupna za mobilne platforme Android i iPhone.

4.1 Korištenje mobilne aplikacije

Odmah nakon pokretanja mobilne aplikacije FEatheR dočekuje nas pozdravni ekran sa sadržajem prikazanim na Slika 3. Zaštitni znak aplikacije Feathe. Zaštitni znak sadrži motiv pera, ime aplikacije te zaštitni znak Fakulteta elektrotehnike i računarstva.

Slika 3. Zaštitni znak aplikacije FeatheR

4.1.1 Obrazac za pretragu predmeta

Nakon što korisnik pritisne proizvoljno mjesto na ekranu aplikacija će iscrtati obrazac kao na Slika 4. Obrazac za pretragu predmeta. preko kojega korisnik može pretraživati predmete zadajući parametre pretrage. Na priloženoj slici neka od polja za unos parametara su već popunjena s proizvoljnim podatcima.

Slika 4. Obrazac za pretragu predmeta.

U obrascu se nalazi ukupno šesnaest različitih parametara prema kojima se mogu pretraživati predmeti. Neki od parametara su međusobno ovisni. Ako korisnik ne želi da neki parametar utječe na pretragu tada to polje treba ostaviti praznim ili Ako je prisiljen napraviti barem jedan odabir tada treba odabrati opciju SVI za taj parametar koji želi isključiti iz pretrage.

Parametar ID predmeta odgovara matičnom broju predmeta pod kojim je taj predmet spremljen u ISVU bazi podataka. Taj broj je jedinstven za svaki predmet.

Ako korisnik pokuša predati obrazac, a u ovome polju se nalaze simboli koji nisu brojčane znamenke, aplikacija će upozoriti korisnika i neće predati obrazac. Uneseni broj mora točno odgovarati željenom predmetu.

U polju za unos naziv predmeta korisnik može unijeti znakove koji se nalaze u skupu hrvatske abecede ili neku od brojčanih znamenki. Pretražuju se predmeti za koje vrijedi da je uneseni podatak dio ili potpun naziv predmeta.

U polja za unos ime nastavnika i prezime nastavnika korisnik smije unijeti samo znakove koji se nalaze u skupu hrvatske abecede. Ako korisnik odluci unijeti ili ime ili prezime uneseni podatci mogu odgovarati dijelu ili cijelom odgovarajućem podatku.

Kao razinu na kojem se predmet predaje korisnik može odabrati vrijednosti SVI, Preddiplomski i Diplomski. Vrijednost SVI se odnosi na predmete koji se predaju na preddiplomskom i diplomskom dok se vrijednosti Preddiplomski i Diplomski odnose na razinu preddiplomskog i diplomskog zasebno.

Korisnik može filtrirati predmete prema količini ECTS bodova koliko vrijede. Moguće je zadati vrijednost te pretraživati predmete koji su jednaki, strogo manji ili strogo veći toj vrijednosti.

Za vrijednosti semestra, ako je odabrana vrijednost razine SVI ili Preddiplomski tada korisnik može odabrati vrijednosti od prvog do šestog semestra. Ukoliko je odabrana razina strogo Diplomski tada korisnik može odabrati samo razine od prvog do četvrtog semestra.

Odabrani jezik može biti en što predstavlja engleski jezik, hr što predstavlja hrvatski jezik te SVI što predstavlja kombinaciju svih ponuđenih jezika (trenutno samo navedena dva).

U padajućim izbornicima za smjer i tip na kojem se predmet polaže korisnik može odabrati jedan od postojećih smjerova odnosno tipova koji su pohranjeni u bazi podataka, ali također može i odabrati vrijednost SVI čime se pretražuju predmeti na svim postojećim smjerovima odnosno tipovima. Oba popisa ponuđenih izbora se dinamički određuju ovisno o odabranoj vrijednosti jezika i odabranoj vrijednosti razine predmeta. Isped imena svakog smjera odnosno tipa predmeta nalazi se

odgovarajuće slovo koje predstavlja razinu (simbol također ovisno o jeziku na kojem je ponuđen izbor). „P“ predstavlja razinu preddiplomskog na hrvatskom jeziku, „D“ razinu diplomskog na hrvatskom jeziku, „U“ razinu preddiplomskog na engleskom jeziku te „G“ razinu diplomskog na engleskom jeziku.

Unutar područja za unos opisa predmeta korisnik može unijeti dio (ili čak čitav) opis predmeta prema kojemu će aplikacija pretražiti predmete sa zadanim sadržajem unutar opisa predmeta.

Sati predavanja i sati laboratorijskih vježbi su parametri koje korisnik može koristiti da bi daljnje suzio svoju pretragu. Pretraživati će se samo oni predmeti koji imaju strogo manji, strogo veći ili jednak broj sati predavanja ili laboratorija prema unesenim podatcima.

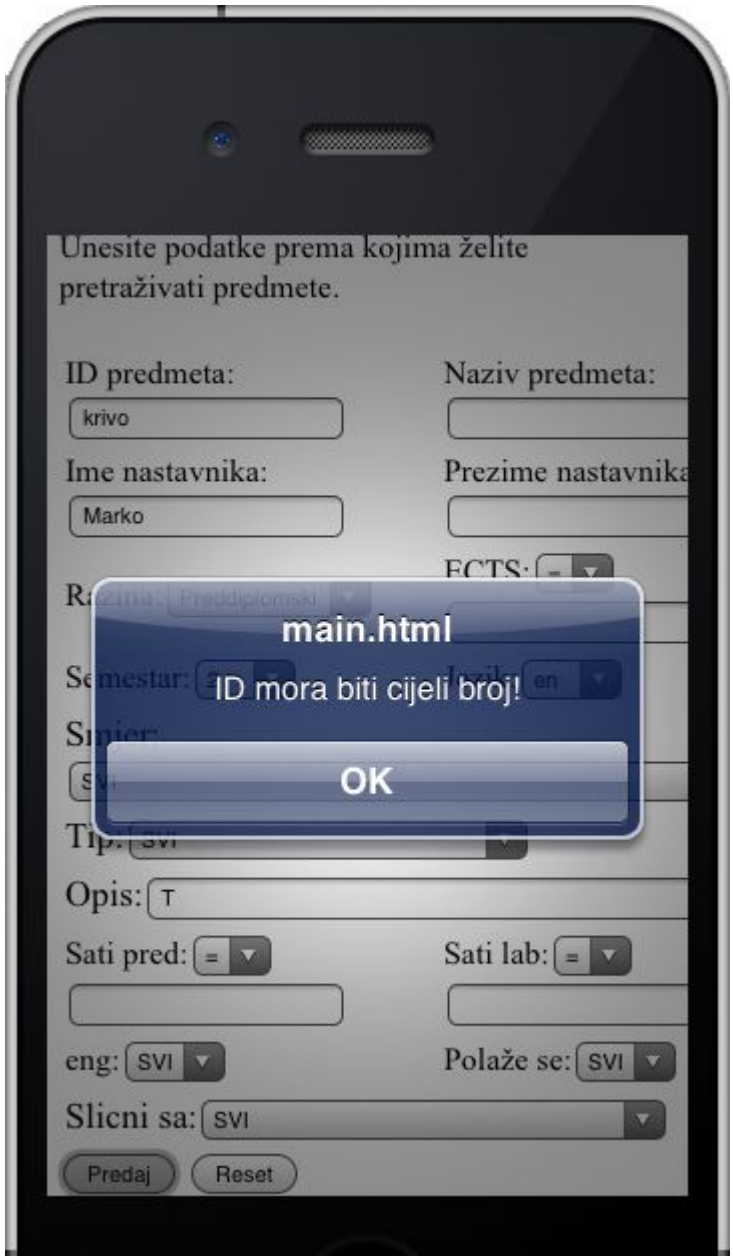
Razine predmeta na kojima su predmeti dostupni su redom od razine nula do razine tri. Ako korisnik želi pretraživati predmete s bilo kojom od četiri postojeće razine treba odabrati opciju SVI.

Korisnik može ograničiti pretragu na predmete koji se polažu odnosno na predmete koji se ne polažu odabirom odgovarajuće vrijednosti u izborniku koji određuje da li se predmet polaže. Kao i kod ostalih lista s ponuđenim opcijama korisnik može odabrati opciju SVI.

Posljednji parametar omogućuje korisniku da u pretragu uključi uvjet da su predmeti koji se pretražuju srodni na nekom od odabranih sveučilišta. Ponuda sveučilišta odgovara popisu svih sveučilišta s kojima je poznat barem jedan srodan predmet.

Ako korisnik u neko od polja unese neki od nedopuštenih znakova za taj parametar aplikacija će ga upozoriti što je prikazano na Slika 5. Aplikacija upozorava korisnika na neispravan unos parametara. Aplikacija će upozoriti korisnika tako što će dojaviti sve znakove koji nisu dopušteni te polja u kojima treba načiniti ispravke.

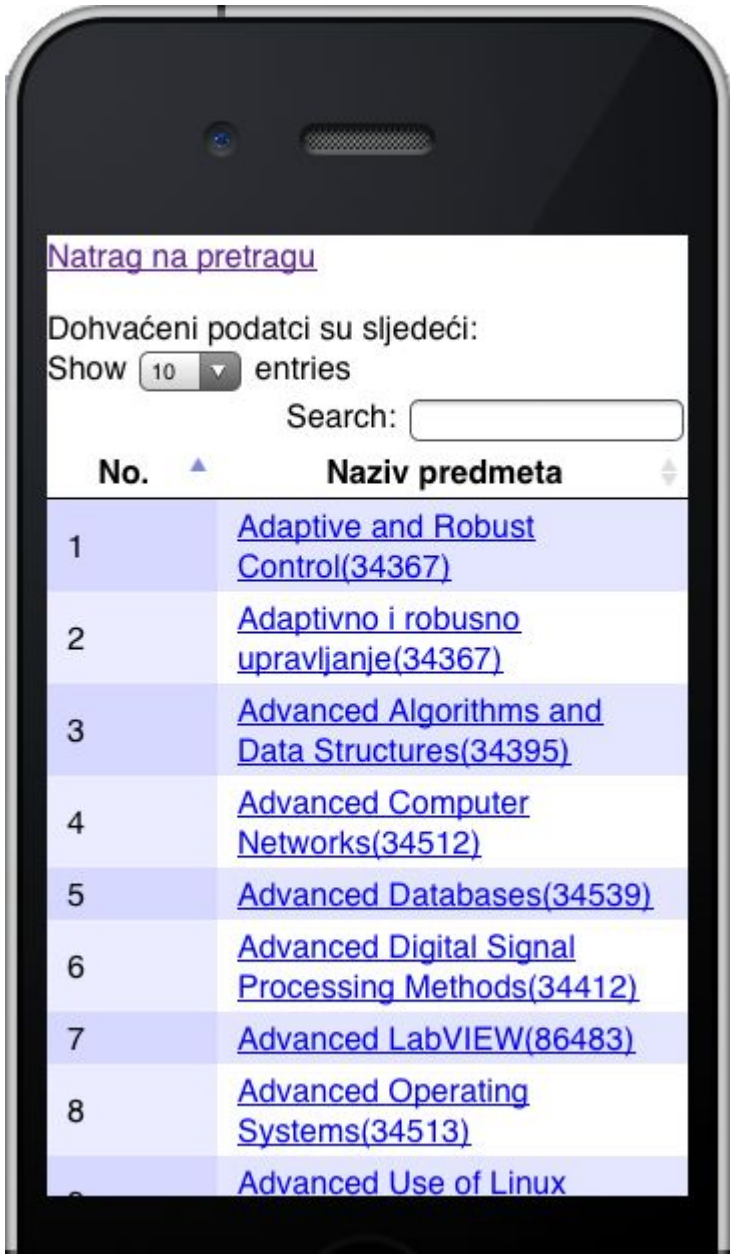
Ako su svi podatci uneseni unutar svoje domene aplikacija će prihvatiti obrazac, obraditi podatke i pronaći popis predmeta koji odgovaraju zadanim parametrima.



Slika 5. Aplikacija upozorava korisnika na neispravan unos parametara

4.1.2 Rezultati pretrage

Na samom vrhu rezultata pretrage korisniku je ponuđen povratak na obrazac za pretragu.



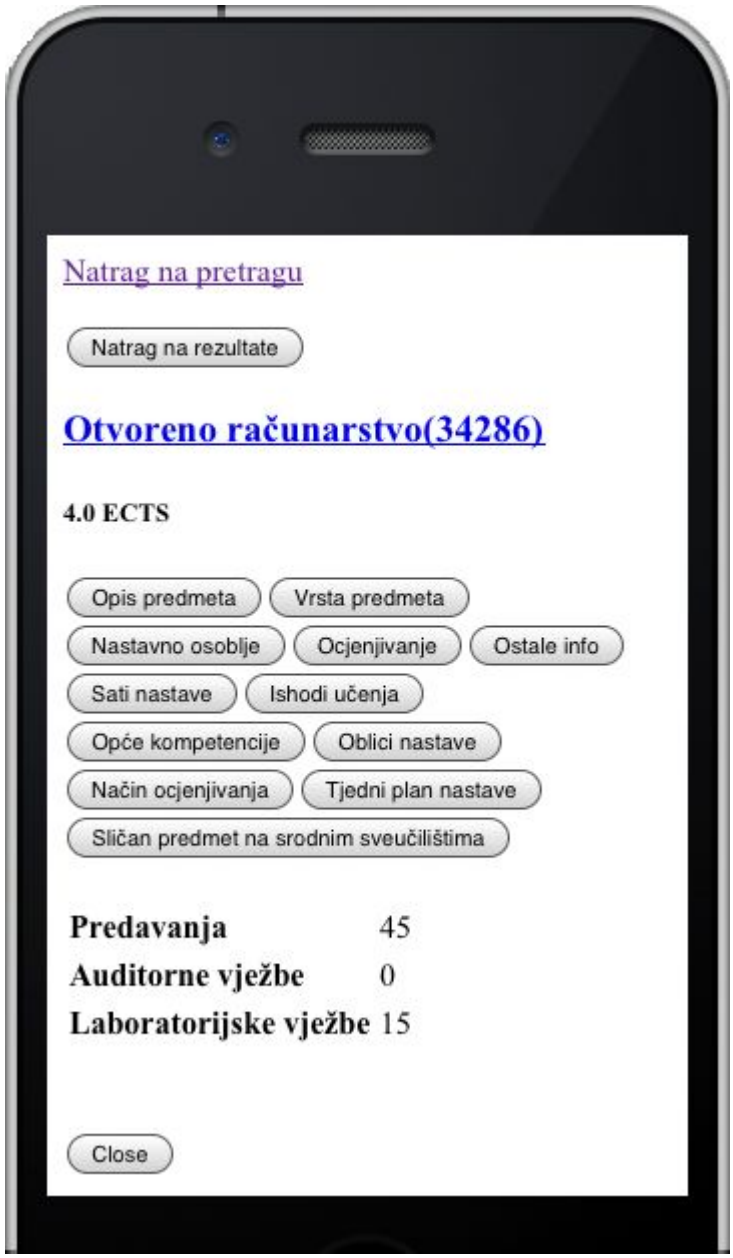
Slika 6. Prikaz pronađenih rezultata

Ako predani podatci sadrže nedopuštene znakove aplikacija upozorava korisnika na krivu upotrebu te ispisuje odgovarajuću poruku. Ako nakon obrade primljenih valjanih podataka nije pronađen niti jedan predmet koji odgovara zahtjevima aplikacija izvješćuje korisnika o tome ispisom poruke. Pronađe li se barem jedan zadovoljavajući predmet aplikacija će sve predmete ispisati unutar tablice kao što je prikazano na . Svaki rezultat prikazan sadrži podatak o rednom broju rezultata (koji nema nikakve druge važnosti osim pobrojavanja) te hiperveze s

imenom i jedinstvenom šifrom predmeta. Klikom na hipervezu aplikacija će prijeći na prikaz zadanog predmeta. U navedenoj tablici moguće je daljnje pretraživati rezultate unosom uzorka za pretragu u području *search*. Korisnik također može postaviti broj prikazanih rezultata po stranici te prelaziti na sljedeću ili prethodnu stranicu. Dodatno moguće je promijeniti poredak rezultata klikom na atribut tablice. Ispod tablice ispisan je broj pronađenih rezultata

4.1.3. Prikaz predmeta

Na samom vrhu stranice korisniku je ponuđena hiperveza na stranicu sa obrascom za pretragu. Ispod hiperveze nalazi se gumb koji nudi korisniku povratak na popis rezultata. Ukoliko su na predani pogrešni podatci o predmetu koji ne postoji u bazi podataka aplikacija će upozoriti korisnika o krivoj upotrebi. Pod pretpostavkom da je korisnik ispravno slijedio sve korake aplikacije na zaslonu će se ispisati podatci o predmetu. U samom zaglavlju podataka nalazi se hiperveza na matičnu stranicu predmeta, a sama hiperveza predstavljena je nazivom predmeta i njegovim jedinstvenom šifrom. Ispod hiperveze nalazi se podataka o broju etcs bodova koji odgovaraju težini predmeta. Uz hipervezu na korištenje korisniku je ponuđen i skup gumbi što je prikazano na . Gumbi predstavljaju organizaciju u grupe svih dostupne podatke za zadani predmet. Korisnik može pritiskom na gumb pristupiti željenoj grupi podataka kao što je opisano na slici. Nakon što odabere jednu skupinu podataka može otvoriti novu skupinu podataka klikom na neki od preostalih gumbi ili zatvoriti navedenu grupu pritiskom na gumb za zatvaranje otvorenih podataka koji se nalazi odmah ispod podataka kojima je korisnik pristupio.



Slika 7. Organizacija podataka o traženom predmetu

4.2. detalji implementacije mobilne aplikacije

Bitne datoteke koje ostvaruju mehanizme aplikacije te su razvijene u sklopu ovog seminara možemo imenovati kao FERLua.lua, index.html, main.html, results.html, subject.html, infopackage.sqlite. Rad same aplikacije započinje pokretanjem koda koji se nalazi u datoteci main.cpp. Otpakiravaju se datoteke potrebne za rad aplikacije te se učitava biblioteka LuaLib.lua i modul FERLua.lua.

Biblioteka LuaLib.lua je potrebna za ostvarivanje mosta između jezika Lua ss ostalim mosync komponentama (biblioteka potječe iz projekta MobileLua).

4.2.1 Modul FERLua.lua

Modul možemo podijeliti na tri značajna dijela. To su funkcije za rad s bazom podataka., funkcije za predznačivanje niza znakova (eng. *escape string functions*) te FERLua funkcije.

Samo je jedna funckija u skupini za rad s bazom podataka, to je funkcija *DBCursorGetColumnText.* Ona dohvaća vrijednost atributa za zadani pokazivač na redak rezultata upita te poziciju željenog atributa.

Od funkcija za predznačivanje niza znakova najznačajnija je *quotestring*. Zadatak te funkcije je uzeti bilo kakav niz znakova, enkodirati ga u format utf – 8 te ograditi ga navodnicima.

FERLua funckije se nazivaju tako zato što su pohranjene u FERLua objektu. U FERLua objektu pohranjen je niz varijabli koje su zadane kao konstante koje omogućuju programeru prilagodljivost na promjene u bazi podataka te zatim niz odgovarajućih funkcija koje provjeravaju da li njihov argument pripada zadanom skupu (služe kao provjera da li su podatci koji su stigli unutar poznatih domenskih ograničenja). U FERLua objektu pohranjene su tri funkcije koje se pozivaju iz JavaScript sloja to su: *initializeMainHtml*, *resultsCompute* i checkExistSubject. Funkcija *initializeMainHtml* postavlja odgovarajuće varijable koje sadržavaju podatke koje trebaju biti ponuđeni u obrascu, a ne mogu se smatrati stalnima kroz verzije baze podataka (npr. broj semestara preddiplomskog se može smatrati nepromjenjivom vrijednosti kroz verzije baze podataka dok se imena smjerova ne mogu). U funkciji *resultsCompute* se pretražuju predmeti prema zadanim parametrima uz provjeru parametara te se u potrebne strukture podataka pohranjuju podatci. Jednom dohvaćeni podatci se pohranjuju te se pri promjenama obrasca ne trebaju dohvaćati ponovo. *checkExistSubject* je funkcija koja će provjeriti da li se nalazi zadani predmet te ovisno o njegovom postojanju, odnosno ne postojanju će poduzeti niz potrebnih akcija ovisno o ishodu prethodne provjere.

Nakon što se modul učita poziva se samo jedna funkcija koja se nalazi u članu objekta FERLua, a to je FERLua.Main. U navedenoj funkciji registrira se mogućnost pritiska gumba na mobilnom uređaju za povratak iz aplikacije. Ako se navedeni događaj uistinu dogodi aplikacija treba stati s izvođenjem rada.

4.2.2 Pozivanja između slojeva JavaScript i Lua

Kada se u jednom od navedenih slojeva programskog koda pozove funkcija iz drugog sloja rad pokreće se neovisni slijed izvođenja instrukcija u tom sloju. U tom slučaju dakle sloj koji poziva drugi sloj nastavlja sa svojim izvođenjem, a pozvana funkcija u sloju pozvanika započinje sa svojim radom dok niz instrukcija koje slijede nakon poziva ne bude dovršen.

Zbog ovakvog načina rada javlja se problem sinkronizacije slojeva i problem pristupa istim memorijskim lokacijama.

4.2.3 Stranica index.html

Postavljen je logo i njegova veličina uz ispis pozdravnog teksta. Klikom bilo gdje na tijelo stranice poziva se funkcija koja će preusmjeriti na stranicu main.html.

4.2.4 Stranica main.html

Pri učitavanju stranice učitava se obrazac i polja kojima su unaprijed poznate vrijednosti se ispunjavaju. Možemo odijeliti tri skupine funkcija u JavaScript kodu. To su funkcije pohranu i dohvat podataka koji unaprijed nisu poznati obrascu (to su podatci u čije vrijednosti ne možemo biti sigurni da se neće mijenjati kroz verzije baze podataka), funkcije za promjenu obrasca te funkcije za provjeru obrasca. Jednom dohvaćeni podatci se pohranjuju te se podatci međusobno kombiniraju spajanjem. Budući da su podatci pohranjeni u poredanom rastućem poretku pri spajanju podataka ovo je uzeto u obzir kako bi se očuvao poredak i u izbornicima.

Dohvat i pohrana podataka se odvijaju odmah nakon učitavanja tijela obrasca. Nakon što su podatci pohranjeni obrazac se osvježava namjernim pozivanjem funkcije za promjenu obrasca.

Funkcije za promjenu obrasca mijenjanju ponuđene podatke u obrascu ovisno o odabranim vrijednostima parametra jezik i parametra razina te se automatski pozivaju pri promjeni neke od tih dvaju vrijednosti.

Funkcije za provjeru obrasca se pozivaju nakon što korisnik zatraži predaju obrasca. Funkcije provjeravaju da li su svi parametri unutar domene u kojoj se smiju nalaziti. Ako neki od parametara nije unutar svoje domene obrazac se neće predati te će aplikacija upozoriti korisnika s informacijama koje ukazuju na prvo polje i prvi uzročnik koji je uzrokovao grešku. Obrazac kada je ispunjen ispravnim podatcima prenosi podatke metodom GET.

4.2.5 Stranica results.html

Tijelo stranice sadrži samo oznake područja koja će kasnije biti ispunjena s podatcima. Područja su prostor za hipervezu na početni obrazac, područje za sadržaj te područje za ispis tablice. Nakon što se učita tijelo stranice dohvaćaju se svi podatci primljeni metode GET. Provjerava se postojanje parametara i njihova prihvatljivost (nalaze se unutar svoje domene). Ukoliko podatci nisu prihvatljivi u području za sadržaj će se ispisati poruka o pogrešno zadanim parametrima. Nasuprot tomu ako su podatci valjani pozvati će se već opisana funkcija u Lua sloju *resultsCompute.* Ta će funkcija ukoliko ne pronađe niti jedan rezultat uzrokovati ispis poruke u području sadržaj koja obavještava korisnika o nepostojanju predmeta s traženim parametrima. U suprotnom ukoliko nađe barem jedan predmet uzrokovati će ispis poruke o uspjehu u području sadržaj te uzrokovati ispis podataka u području za ispis tablice.

Za ispis tablice se koristi plug-in za jQuery pod imenom DataTables. Procedura za ispis tablice se odvija u funkciji *drawTable*. To je jedina funkcija koja koristi jQuery i navedeni plug-in. U sljedećim verzijama aplikacije plan je izmijeniti funkciju na neko manje memorijski zahtjevno rješenje. Jer na trenutno rješenje

zauzima dodatnih 0.5 MB prostora. Rješenje je trenutno prihvatljivo dok se ne dobije povratna reakcija korisnika nakon čega će se moći odlučiti u kojem smjeru će se nastaviti razvijati aplikacija. Korisnici će trebati odlučiti o načinu na koji žele da im se ponude rezultati i koliko su im bitna neka dodatna svojstva u prikazu.

4.2.6 Stranica subject.html

Stranica na kojoj se prikazuju predmeti podijeljena je na polja u kojima se mogu postavljati odgovarajući podatci. Valja istaknuti tri polja. To su polje za gumbe, polja označeno kao skladište za podatke te polje za gumb koji zatvara skladište za podatke. Nakon učitavanja tijela stranice pozvati će se funkcija iz Lua sloja koja provjerava da li zaprimljeni podatci odgovaraju nekom postojećem predmetu. Ukoliko predmet ne postoji Lua će pozvati ispis odgovarajuće poruke u JavaScript sloju. Postoji li predmet u Lua sloju će se sakupiti svi dostupni podatci o predmetu te će se organizirati ispis podataka na zaslon. Na vrhu podataka se ispisuje hiperveza koja preusmjerava na url dohvaćen iz baze podataka. Tekst hiperveze ujedno služi i kao naslov, a sastoji se od naziva predmeta i njegove šifre. Potom se grupiraju dohvaćeni podatci i spremaju unutar asocijativnog polja. Spremljeni podatci su spremljeni u obliku spremnom za umetanje u polje označeno kao skladište podataka. Generiraju se gumbi koji pune polje gumba na temelju ključeva asocijativnog polja koje sadrži sve podatke koji su korisni. Pritiskom na specifičan gumb posebnoj funkciji se predaje argument koja ovisno o primljenom argumentu područje za prikaz podataka puni odgovarajućom grupom iz asocijativnog polja. Zatim se to polje postavlja vidljivim na zaslonu te se također postavlja vidljivim i gumb u području za gumb koji zatvara skladište podataka. Pritiskom na taj gumb gasi se vidljivost polja označeno kao skladište za podatke i vidljivost samog gumba za zatvaranje skladišta podataka(jer više nije potreban)..

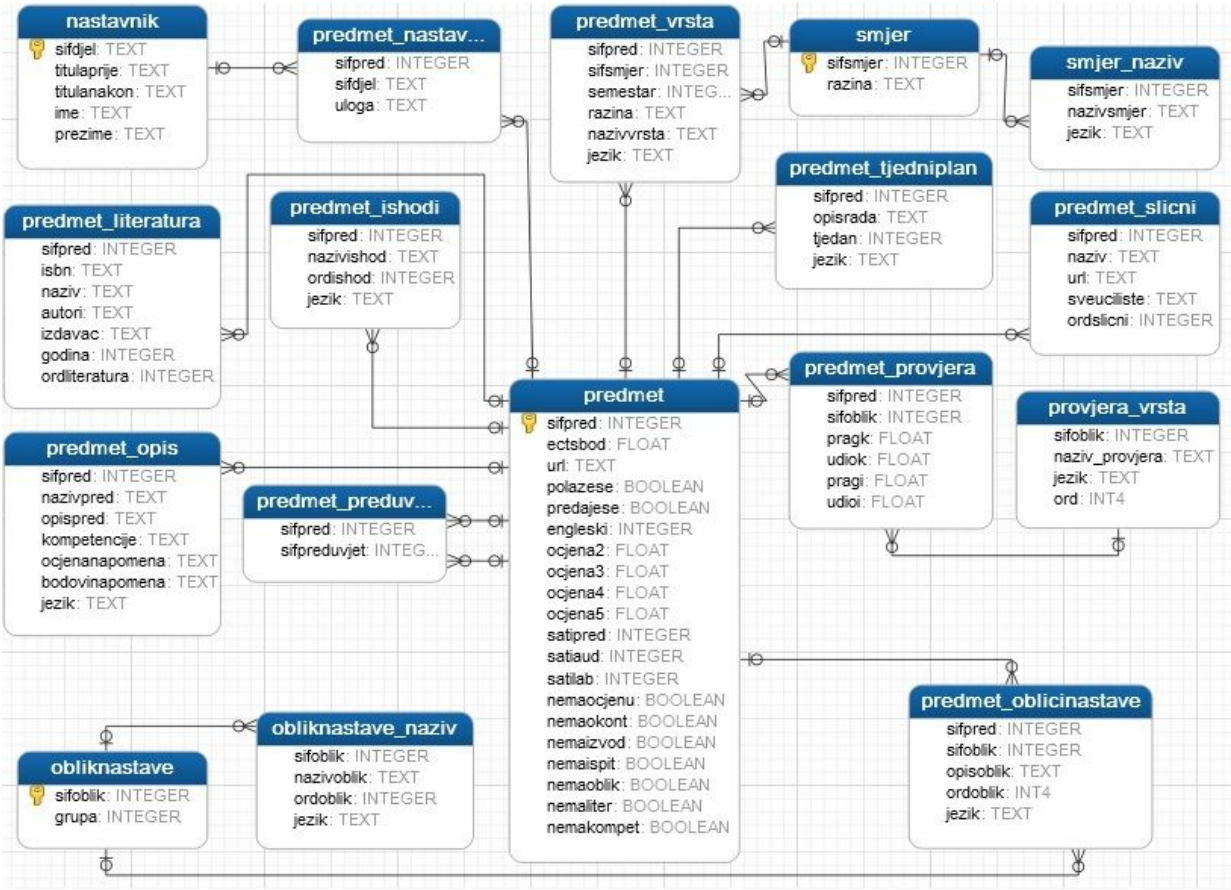
4.2.7 Baza podataka infopackage.sqlite

Baza podatka sadrži pohranjene podatke iz informacijskog paketa Fakulteta elektrotehnike i računarstva. E – R model baze podataka korištene u trenutnoj verziji

mobilne aplikacije FEatheR prikazan je na Slika 8. E-R model baze podataka. U daljnjem nastavku teksta opisano je nužno sučelje baze podataka koje aplikacija očekuje. Sučelje je opisano pobrojanim imenima tablice i atributima koje mora sadržavati. Uz svaki atribut je naveden očekivani tip odvojen dvotočjem.

1. predmet (sifpred : INTEGER, ectsbod : FLOAT, url : TEXT, polazese : BOOLEAN, predajese : BOOLEAN, engleski : INTEGER, ocjena2 : FLOAT, ocjena3 : FLOAT, ocjena4 : FLOAT, ocjena5 : FLOAT, satipred : INTEGER, satiaud : INTEGER, satilab : INTEGER, nemaocjenu : BOOLEAN, nemaokont : BOOLEAN, nemaizvod : BOOLEAN, nemaispit : BOOLEAN, nemaoblik : BOOLEAN, nemaliter : BOOLEAN, nemakompet : BOOLEAN)
2. predmet_opis (sifpred : INTEGER, nazivpred : TEXT, opispred : TEXT, kompetencije : TEXT, ocjenanapomena : TEXT, bodovinapomena : TEXT, jezik : TEXT)
3. predmet_nastavnik (sifpred : INTEGER, sifdjel : TEXT, uloga : TEXT)
4. Nastavnik (sifdjel : TEXT PRIMARY KEY, titulanakon : TEXT, ime : TEXT, prezime : TEXT)
5. predmet_literatura (sifpred : INTEGER, naziv : TEXT, autori : TEXT, izdavac : TEXT, godina : INTEGER, ordliteratura : INTEGER)
6. predmet_preduvjet (sifpred : INTEGER, sifpreduvjet INTEGER)
7. predmet_ishodi (sifpred : INTEGER, nazivishod : TEXT, ordishod : INTEGER, jezik : TEXT)
8. predmet_tjedniplan (sifpred : INTEGER, opisrada : TEXT, tjedan : INTEGER, jezik : TEXT)
9. predmet_vrsta (sifpred : INTEGER, sifsmjer : INTEGER, semestar : INTEGER, razina : TEXT, nazivvrsta : TEXT, jezik : TEXT)
10. smjer_naziv (sifsmjer : INTEGER, nazivsmjer : TEXT, jezik : TEXT)
11. obliknastave_naziv (sifoblik : INTEGER, nazivoblik : TEXT, ordoblik : INTEGER, jezik : TEXT)
12. predmet_oblicinastave (sifpred : INTEGER, sifoblik : INTEGER, opisoblik : TEXT, ordoblik : INT4, jezik : TEXT)
13. provjera_vrsta (sifoblik : INTEGER, naziv_provjera : TEXT, jezik : TEXT, ord : INT4)
14. predmet_provjera (sifpred : INTEGER, sifoblik : INTEGER, pragk : FLOAT, udiok : FLOAT, pragi : FLOAT, udioi : FLOAT)

15. predmet_slicni (sifpred : INTEGER, naziv : TEXT, url : TEXT, sveuciliste TEXT, ordslicni : INTEGER)



Slika 8. E-R model baze podataka

5. Zaključak

Lua je vrlo jednostavan i moćan skriptni jezik. Integrirana razvojna okružja su još u razvoju, no jedno takvo Lua Development Tools se pokazalo korisnim za izradu Lua programa.

Mogućnost izrade više-platformske mobilne aplikacije stavlja programera i njegovu aplikaciju u zavidnu prednost pred aplikacijama namijenjenim samo za jednu platformu. Okružje MoSync omogućuje svojim korisnicima – programerima u takvoj namjeri. Te uz dodatak projekta MobileLua moguće je ugraditi Lua programe unutar aplikacije.

Aplikacija FEatheR svome korisniku omogućuje samo pretraživanje i prikaz podataka iz informacijskog paketa Fakulteta elektrotehnike i računarstva.

Ako se promjeni sadržaj informacijski paket aplikacija ostaje korisna ukoliko se definirano sučelje između baze podataka i aplikacije očuva. Aplikacija je vrlo fleksibilna na promjene i dodavanje novih parametara za pretragu.

Ova aplikacija se može proširiti tako da skup trenutnih funkcionalnosti bude samo dio nove aplikacije koja će podržavati velik raspon funkcionalnosti i sadržaja vezanih uz Fakultet elektrotehnike i računarstva što bi bio logičan plan za budućnost aplikacije FEatheR.

6. Literatura

[1] *Programming in Lua*

http://www.lua.org/pil/index.html

[2] Stranice kolegija otvoreno računarstvo

http://www.fer.unizg.hr/predmet/or

[3] *Lua launch configuration*

http://www.eclipse.org/forums/index.php/t/273959/

[4] *MoSync – wikipedia*

http://en.wikipedia.org/wiki/MoSync

[5] *Getting started with MobileLua*

https://github.com/divineprog/mobilelua/wiki/Getting-started

[6] *MobileLua API*

https://github.com/divineprog/mobilelua/wiki/MobileLua-API

[7] *The MoSync Emulator and MoRE*

http://www.mosync.com/documentation/manualpages/more-mosync-runtime-environment

[8] *Emulating a Device*

http://www.mosync.com/documentation/manualpage/emulating-device

[9] *MoSync*

www.mosync.com

[10] *MobileLua*

https://github.com/divineprog/mobilelua

[11] Postupak instalacije *Lua Development Tools*

http://www.eclipse.org/koneki/ldt/#installation

[12] *Running external tools*

http://help.eclipse.org/helios/index.jsp?topic=%2Forg.eclipse.platform.doc.user%2Ftasks%2Ftasks-exttools-running.htm

[13] *Android SDK*

http://developer.android.com/sdk/index.html

[14] *Feature/Platform Support*

http://www.mosync.com/widepage/feature-platform-support

[15] *Build an app with JavaScript and Lua - Dynamic language interplay*

http://www.mosync.com/content/mixing-javascript-and-lua-dynamic-language-interplay

7. Sažetak

Lua

Lua je imperativni više – paradigmatski skriptni jezik sa malenim memorijskim otiskom, lako ugradljiv u druge sustave. Lua programske moduli u ovom seminaru su razvijani u okružju Lua Development Tools. Sama mobilna aplikacija je izgrađena u okružju MoSync uz dodatak projekta MobileLua. Naziv mobilne aplikacije jest FEatheR i njena namjena je ponuditi pregled informacijskog paketa Fakulteta elektronike i računarsva svome korisniku. Dijelove aplikacije možemo podijeliti na pozdravni zaslon, obrazac za pretragu, rezultate pretrage te prikaz predmeta.

Ključne riječi: Lua, Lua Development Tools, mobilne aplikacije, MoSync, MobileLua, FEatheR

Lua

Lua is a multi – paradigm scripting language with a small memory footprint, easy embeddable. Lua programming modules in this course are developed in an integrated developer environment of Lua Development Tools. The mobile application is built in the integrated developer environment of MoSync with addition of MobileLua project. Name of the application is FEather and its purpose is to provide an overview of the information package of Faculty of Electronics and Computing to its user. Application can be divided into the welcome screen, search form, search results and subject display.

Keywords: Lua, Lua Development Tools, mobile aplications, MoSync, MobileLua, FEatheR