



Univerza v Mariboru

*Fakulteta za elektrotehniko,  
računalništvo in informatiko*

Alen Sotlar

# **MOBILNI PRIROČNIK O VADBI IN PREHRANI ZA OPERACIJSKI SISTEM ANDROID**

Diplomsko delo

Maribor, september 2012

# **Mobilni priročnik o vadbi in prehrani za operacijski sistem Android**

## **Diplomsko delo univerzitetnega študijskega programa**

Študent: Alen Sotlar  
Študijski program: univerzitetni študijski program  
Računalništvo in informacijske tehnologije  
Mentor: red. prof. dr. Milan Zorman  
Somentor: asist. Bojan Butolen, univ. dipl. inž. rač. in inf.  
Lektorica: Ines Kolar, univ. dipl. pedagoginja in prof. slovenščine

Maribor, september 2012



Univerza v Mariboru

Fakulteta za elektrotehniko,  
računalništvo in informatiko

Številka: E1000047


Datum in kraj: 17. 04. 2012, Maribor

Na osnovi 330. člena Statuta Univerze v Mariboru (Ur. L. RS, št. 01/2010)  
izdajam

### SKLEP O DIPLOMSKEM DELU

1. Alenu Sotlarju, študentu univerzitetnega študijskega programa RAČUNALNIŠTVO IN INFORMACIJSKE TEHNOLOGIJE, se dovoljuje izdelati diplomsko delo pri predmetu Razvoj programske opreme.
2. MENTOR: red. prof. dr. Milan Zorman  
SOMENTOR: asist. Bojan Butolen
3. Naslov diplomskega dela:  
MOBILNI PRIROČNIK O VADBI IN PREHRANI ZA OPERACIJSKI SISTEM ANDROID
4. Naslov diplomskega dela v angleškem jeziku:  
MOBILE PERSONAL TRAINER AND NUTRITION ADVISER FOR ANDROID OS
5. Diplomsko delo je potrebno izdelati skladno z "Navodili za izdelavo diplomskega dela" in ga oddati v treh izvodih (dva trdo vezana izvoda in en v spiralo vezan izvod) ter en izvod elektronske verzije do 30. 09. 2012 v referatu za študentske zadeve.

Pravni pouk: Zoper ta sklep je možna pritožba na senat članice v roku 3 delovnih dni.

Dekan:  
  
\_\_\_\_\_

Obvestiti:

- kandidata,
- mentorja,
- somentorja,
- odložiti v arhiv.

II

II

## ZAHVALA

Zahvaljujem se mentorju red. prof. dr. Milanu Zormanu in somentorju asist. Bojanu Butolenu, za pomoč in vodenje pri opravljanju diplomskega dela.

Posebna zahvala gre tudi Saši za pomoč in podporo pri pisanju diplomske naloge, staršem za pomoč in omogočanje študija ter Špeli za podporo med študijem.

# **Mobilni priročnik o vadbi in prehrani za operacijski sistem Android**

**Ključne besede:** mobilni priročnik, operacijski sistem Android, vadba, prehrana

**UDK:** 004.45:621.395.721.5(043.2)

## **Povzetek**

*Cilj diplomske naloge je raziskati mobilno platformo Android in razviti mobilni priporočnik o hrani in vadbi za operacijski sistem Android. V prvem delu diplomske naloge predstavimo platformo Android, razvojno okolje Eclipse, v katerem smo razvijali aplikacijo in SQLite ter SQLite Data Browser, ki smo ga uporabili za razvoj podatkovnih baz. V nadaljevanju opišemo potek razvoja mobilne aplikacije, njen načrt, gradnike ... Na koncu predstavimo še njeno testiranje in rezultat.*

# **Mobile personal trainer and nutrition advisor for Android OS**

**Key words:** mobile guide, Android OS, exercise, nutrition

**UDK:** 004.45:621.395.721.5(043.2)

## **Abstract**

*The aim of the thesis is to explore the Android mobile platform and developed a mobile guide for Android OS. In the first part we present the Android platform, the Eclipse development environment in which we have developed an application, SQLite and Sql Data Browser, which was used to develop databases. In the second part we describe the course of development of the mobile applications, its design, widgets ... Finally presented is testing and result.*

## Kazalo vsebine

1	UVOD .....	1
2	MOBILNA PLATFORMA ANDROID .....	3
2.1	Zgodovina.....	4
2.2	Android SDK.....	5
2.3	Android Market – Google Play .....	6
2.4	Glavne lastnosti platforme.....	7
3	RAZVOJNO OKOLJE ECLIPSE IN ECLIPSE FOUNDATION .....	8
3.1	SQLite.....	8
3.2	SQLite Database Browser .....	9
4	IMPLEMENTACIJA MOBILNE APLIKACIJE .....	10
4.1	Delovno okolje .....	10
4.2	Načrt aplikacije.....	12
4.3	Izdelava logotipa in gumbov .....	14
4.4	Shranjevanje in pridobivanje podatkov .....	15
4.4.1	Pridobivanje podatkov iz SQLite podatkovne baze .....	15
4.4.2	Ustvarjanje podatkovne baze znotraj programa .....	17
4.5	Implementacija nekaterih funkcionalnosti.....	18
4.5.1	Implementacija aktivnosti (ang. activity).....	18
4.5.2	Implementacija opcije/tipke menu .....	18
4.5.3	Implementacija spustnega menu-ja (ang. spinner oz. drop down list).....	19
4.5.4	Implementacija gumbov (ang. button) .....	20
4.5.5	Implementacija sporočil (ang. toast message).....	20
4.5.6	Implementacija opozoril (ang. alert box) .....	21
4.6	Hrana, kalorije in vadba.....	21
4.6.1	Glikemični indeks .....	21
4.6.2	Kako se meri glikemični indeks? .....	22

4.6.3	Pomen glikemičnega indeksa .....	22
4.6.4	Kalorije .....	23
4.6.5	Kako izračunamo kalorično vrednost jedi? .....	23
4.6.6	Računanje porabljenih kalorij .....	23
4.6.7	Indeks telesne mase .....	24
5	TEST APLIKACIJE .....	25
5.1	Testni primeri .....	25
5.2	Testni mobilni telefon.....	26
5.3	Testiranje aplikacije.....	28
6	REZULTATI.....	28
7	SKLEP .....	32
8	VIRI, LITERATURA .....	33
9	PRILOGE.....	37
9.1	Kratek življenjepis.....	37



## Seznam slik

Slika 1: Android (arhitektura) [29].....	3
Slika 2: Google Play.....	7
Slika 3: SQLite Database Browser.....	10
Slika 4: delovno okolje.....	11
Slika 5: načrt aplikacije.....	13
Slika 6: logotip aplikacije.....	14
Slika 7: gumb uredi profil.....	14
Slika 8: merjenje glikemičnega indeksa.....	22
Slika 9: Samsung Galaxy SII.....	27
Slika 10: začetni zaslon.....	29
Slika 11: opravljena aktivnost.....	30
Slika 12: glikemični indeksi, indeks telesne mase in nasveti.....	31

## Seznam tabel

Tabela 1: zgodovina Androida.....	5
Tabela 2: število prenosov aplikacij.....	7
Tabela 3: koeficienti za računanje kalorij.....	24
Tabela 4: indeks telesne mase.....	25
Tabela 5: specifikacije Samsung Galaxy SII.....	27

## Seznam primerov kode

Primer kode 1: razred DataBaseHelper.....	17
Primer kode 2: spremenljivke za ustvarjanje baze.....	17
Primer kode 3: metoda <i>onCreate()</i> .....	18
Primer kode 4: metoda <i>ustvariVnos()</i> .....	18
Primer kode 5: metoda <i>getPorabljeneKcal()</i> .....	18
Primer kode 6: implementacija menu-ja.....	19
Primer kode 7: implementacija <i>spinner</i> -ja.....	20
Primer kode 8: implementacija gumba.....	20

Primer kode 9: implementacija obvestila .....	20
Primer kode 10: implementacija opozorila .....	21

## SEZNAM UPORABLJENIH KRATIC

- SQL standardni poizvedbeni jezik za delo s podatkovnimi bazami  
(ang. Structured Query Language)
- SDK paket za razvoj programske opreme  
(ang. Software Development Kit)
- GNU Unixu podoben operacijski sistem  
(ang. Gnu's Not Linux)
- OS operacijski sistem  
(ang. Operation System)
- IDE integrirano razvojno okolje  
(ang. Integrated Development Environment)
- IT informacijske tehnologije  
(ang. Information Technology)
- IP intelektualna lastnina  
(ang. Intellectual Property)
- CSV ang. comma-separated values
- JDK komplet priborov za razvijanje Jave  
(ang. Java Development Kit)
- ADT razvojna orodja Android  
(ang. Android Development Tools)
- KCAL kilokalorije  
(ang. Kilocalories)
- ITM indeks telesne mase  
(ang. Body Mass Index)

# 1 UVOD

Živimo v času, ko je mobilni telefon nuja in v času, ko ima ogromno ljudi v lasti pametne mobilne telefone. Delež teh telefonov se vsak dan bliskovito povečuje in samo v lanskem letu je bilo prodanih (po podatkih IDC) skoraj 500 milijonov takšnih naprav. V nekaterih državah pa delež teh naprav dosega že skoraj 50 % (Velika Britanija, ZDA, Kanada ...) [14]. Med temi podatki velja izpostaviti veliko rast nakupov, saj se ob vsakem, novem letu prodaja poveča za 30 % in več (odvisno od države).

Posledično s prodajo naprav se povečujeta tudi izdelovanje in prenos aplikacij. Samo Android aplikacij je nastalo od marca 2009 do junija 2012 (objavljenih na Google Play) več kot pol milijona, število prenosov pa se je povzpelo na 20 milijonov [13].

Prav zaradi teh podatkov, zaradi ogromnega zanimanja za pametne telefone in zaradi ogromnega števila prenosov aplikacij smo se odločili, da v diplomskem delu razvijemo aplikacijo za mobilne telefone.

Zakaj ravno Android? Prva in zagotovo najpomembnejša stvar je v njegovem odprtokodnem operacijskem sistemu, ki je po mojih izkušnjah preprost za uporabo in nam dovoljuje veliko opcij in rešitev. Prav tako nam programska oprema (razvija jo podjetje Google z Open Hand Set Alliance) omogoča razvoj aplikacij za mobilne naprave. Naslednja, po mojem mnenju prav tako zelo pomembna zadeva, je brez dvoma delež Android naprav, med pametnimi telefoni. Ta je po podatkih IDC in tudi Garthner (odstopanja so majhna, raziskava pa je bila narejena v prvem četrtletju leta 2012) [14] blizu 60 %, po navedenih raziskavah mu najbolj sledi iOS z nekaj več kot 20 %, deleži ostalih operacijskih sistemov pa so dosti nižji (Symbian okoli 7–8 %, RIM okoli 6 %, Windows pa le okoli 2 %).

Tako odločitev ni bila težka in smo v okviru diplomske naloge razvili aplikacijo ActivityGuide, ki naj bi delovala kot nekakšen priročnik (ang. guide). Z aplikacijo bi uporabnik pridobil nekatere informacije o prehrani, kalorijah in vadbi, predstavila pa bi mu tudi indeks telesne teže. Ker je aplikacija napisana za platformo Android, smo se z njeno izdelavo poglobljeje seznanili z omenjeno platformo, spoznali pa smo tudi razvojno okolje Eclipse in SQLite za delo s podatkovni bazami.

V prvem delu diplomske naloge bomo povedali nekaj podatkov o platformi Android, razvojnem okolju Eclipse in pa o SQLite (SQLite Database Browser), ki nam je služil za upravljanje in kreiranje podatkovnih baz.

V drugem poglavju predstavimo Android, predvsem njegove glavne lastnosti, razvoj platforme do danes, SDK, ki je pomemben za razvoj aplikacij in seveda Google Play kot skladišče aplikacij in ostalih multimedijskih vsebin. V tretjem poglavju je na kratko predstavljeno razvojno okolje Eclipse, opišemo pa tudi SQLite in SQLite Browser, ki smo ga uporabili za kreiranje podatkovnih baz, ki jih aplikacija uporablja.

Drugi del diplomske naloge je namenjen aplikaciji ActivityGuide. V poglavjih, ki se nanašajo na ta del, je predstavljeno delovno okolje, v katerem je bila aplikacija razvita, načrt aplikacije in shema. Za aplikacijo smo skonstruirali tudi logotip in ostale grafične gradnike. Predstavljeno je pridobivanje podatkov iz podatkovne baze, prav tako so predstavljene določene komponente aplikacije, za konec pa sledi opis funkcionalnosti. V zadnjem poglavju tega dela opišemo še testiranje aplikacije, kar vključuje testne primere in opis testnega mobilnega telefona.

V zadnjem delu sledijo še rezultati aplikacije, sklep in literatura.

## 2 MOBILNA PLATFORMA ANDROID

Android je skupek programske opreme za mobilne naprave, ki vključuje operacijski sistem s programsko opremo (mobilne aplikacije), ključno za delovanje. Android OS pa je operacijski sistem, razvit na Linux jedru in GNU programski opremi ter je namenjen mobilnim napravam, kot so pametni telefoni, dlančniki, tablični računalniki ...

Arhitektura Androida sestoji iz štirih slojev, in sicer aplikacije, aplikacijskega ogrodja, knjižnice in Linux jedra. Samo jedro je napisano v programskem jeziku C, knjižnice v programskem jeziku C++, uporabniški vmesnik pa v jeziku Java. Arhitektura je vidna na sliki 1.

Android je v tem trenutku najbolj popularna mobilna platforma, nameščena je na več milijonov naprav v več kot 190-ih državah na svetu. Vsak dan pa se tej številki pridruži še okoli milijon novih uporabnikov (Android naprav) [15].

Platforma nam omogoča enostavno uporabo in veliko naprednih rešitev, ki delujejo skupaj z ostalimi gradniki, senzorji (pospeškometer, gps, kamera ...).

Arhitektura je prikazana na sliki 1:



Slika 1: Android (arhitektura) [29]

## 2.1 Zgodovina

Android, Inc. [16] je bil ustanovljen oktobra 2003 za razvoj pametnih mobilnih naprav, ki bi se lahko čim bolj prilagodile potrebam in željam posameznika. Avgusta 2005 je Google prevzel lastništvo nad podjetjem in s tem napovedal vstop v svet mobilnih naprav.

Novembra 2005 je bila ustanovljena Open Handset Alliance, to je združenje številnih podjetji, kot so HTC, Intel, LG, Broadcom, Google, Motorola, Nvidia, Samsung itd., z namenom ustvariti odprto mobilno platformo, ki bi spodbujala hitrejša inovacije in napredek. Hkrati pa so predstavili svoj prvi skupen projekt Android – platformo za mobilne naprave, ki bi bila zgrajena na Linux jedru.

Prva beta verzija platforme je bila izdana 5. novembra 2007, SDK teden dni kasneje. Na prvo komercialno verzijo (1.0) pa je bilo potrebno počakati do 23. Septembra 2008. Prva Android naprava je bila HTC Dream. Od leta 2009 se za vsako verzijo uporablja tudi tako imenovano opisno ime (ang. codename), ki izvira iz imena slaščic oziroma sladice. Prva takšna verzija je bila verzija 1.5 – Cupcake, za njo pa so bile izdane še številne verzije. Glavne lastnosti verzij in njihova imena predstavlja tabela 1:

<b>VERZIJA</b>	<b>DATUM IZDAJE</b>	<b>IME</b>	<b>LASTNOSTI (nove)</b>	<b>LINUX VERZIJA</b>
Android 1.0	5. 9. 2008	/	internetni brskalnik Google storitve Wi-Fi in Bluetooth	/
Android 1.1	9. 2. 2009	/	shranjevanje prilog v sporočilih	/
Android 1.5	30. 4. 2009	Cupcake	tipkovnica s predlogom vnosa pripomočki – widgets video snemalnik (MPEG-4, 3GP) kopiraj in prilepi za brskalnik nalaganje na YouTube	2.6.27
Android 1.6	15. 9. 2009	Donut	podpora za WVGA resolucijo zaslona podpora za 802.1x, VPNs glasovni in tekstovni vnos za iskanje kontaktov in za iskanje po internetu nov Android Market	2.6.29
Android 2.0	26. 10. 2009	Eclair	(Nova različica)	2.6.29

			Bluetooth 2.1 možnost iskanja po sporočilih (SMS, MMS) optimizacija strojne opreme – občutna pohitritev odzivnega časa OS nov uporabniški vmesnik animacije za ozadje	
Android 2.2	20. 5. 2010	Froyo	optimizacija (hitrost, spomin) nalaganje aplikacij na spominsko kartico spremenjen Android Market, omogoča samodejno posodabljanje aplikacij podpora za Adobe Flash Wi-Fi dostopna točka	2.6.32
Android 2.3	6. 12. 2010	Gingerbread	posodobitev uporabniškega vmesnika dva nova senzorja – girooskop, barometer orodje za kopiranje in lepljenje datotek podpora za SIP VoIP izboljšana (virtualna) tipkovnica	2.6.35
Android 3.0	22. 2. 2011	Honeycomb	optimizacija za tablične računalnike video klepet v Google Talk	2.6.36
Android 4.0	19. 10. 2011	Ice Cream Sandwich	primi in spusti (ang. drag-and- drop) za ustvarjanje map detekcija obraza za odklepanje aplikacij Wi-Fi Direct nov spletni brskalnik	3.0.1
Android 4.1	predvideno v ¾ 2012	Jelly Bean	optimizacija za tablične računalnike dvojni OS (ang. dual-boot) z MS Windows	/
Android 4.2	predvideno 2013	Key Lime Pie	/	/

Tabela 1: zgodovina Androida

## 2.2 Android SDK

Razvijanje programske opreme za Android je proces nastajanja novih aplikacij. Aplikacije so praviloma zapisane v programske jeziku Java, za njihov razvoj pa potrebujemo SDK. SDK je potreben za začetek razvoja v okolju Eclipse, saj nam zagotovi vse potrebno [18].



SDK priskrbi programski vmesnik in vključuje celovit nabor orodji za razvoj. Nudi nam:

- razhroščevalnik (ang. debugger);
- knjižnice;
- upravljalnik virtualne naprave in emulator, to nam omogoča test aplikacije brez fizične naprave, saj emulator posnema delovanje strojne opreme;
- dokumentacijo;
- primere kode;
- vodiče.

Uradno podprto razvojno okolje (IDE) je Eclipse, ki vključuje vtičnik za ADT.

### 2.3 Android Market – Google Play

Google Play [13] (včasih poznan in ustanovljen kot Android Market) je spletna distribucija multimedijskih vsebin, ki jo je ustanovilo podjetje Google in služi kot spletna trgovina za glasbo, filme, knjige in Android aplikacije ter igre.

Aplikacija (Google Play) je že nameščena na vsaki napravi, ki ustreza pogojem o združljivosti z Googlom, z OS Android. To nam omogoča, da lahko iščemo in nalagamo aplikacije, ki so jih razvili drugi razvijalci. Po podatkih Googla je bilo marca 2012 na voljo več kot 450 000 aplikacij.

Google Play omogoča tako avtomatsko kot ročno posodabljanje aplikacij. Prav tako pa tudi filtrira aplikacije in nam prikaže samo tiste, ki so združljive (ang. compatible) z našo napravo.

Prikaz aplikacij, ki so bile na voljo in število prenosov (od marca 2009 do junija 2012):

LETO	MESEC	ŠT. APLIKACIJ	ŠT. PRENOSOV
2009	marec	2 300	
	december	16 000	
2010	april	38 000	
	avgust	80 000	1 milijarda
	oktober	100 000	
2011	maj	200 000	3 milijarde

	julij	250 000	6 milijard
	december	380 297	10 milijard
2012	februar	450 000	
	maj	500 000	15 milijard
	junij	600 000	20 milijard

Tabela 2: število prenosov aplikacij

Aplikacija Google Play ni odprtokodna, do nje lahko dostopajo samo naprave z OS Android in so hkrati združljive z zahtevami Googla. Najdemo jo lahko na [play.google.com](http://play.google.com).

Na sliki 2 je prikazan izgled:



Slika 2: Google Play

## 2.4 Glavne lastnosti platforme

Glavne funkcionalnosti, ki jih podpira platforma Android:

- naprava Dalvik (virtualna naprava) je program, ki izvaja aplikacije. Je integrirana v sistem Android in se zadnje čase uporablja tudi na pametnih televizijah in ne samo na pametnih telefonih oz. tabličnih računalnikih;
- vgrajen internetni brskalnik (osnova je WebKit);
- podpora za kodeke (audio, video);

- baza SQLite za shranjevanje podatkov;
- tehnologija, kot npr. Bluetooth, wifi, 3G;
- vgrajene naprave – gps, pospeškometer, kamera;
- pri mobilnih telefonih podpira mobilno telefonijo;
- grafični vmesnik za aplikacije z OpenGL grafiko;
- več jezičnost.

### **3 RAZVOJNO OKOLJE ECLIPSE IN ECLIPSE FOUNDATION**

Eclipse je razvojno okolje, implementirano v programskem jeziku Java. Je prosto dostopen in namenjen pisanju aplikacij v različnih programskih jezikih [28]. Omogoča razvoj preprostih in tudi kompleksnejših programov. Vsakemu programu ustreza projekt s primernim imenom, projekte pa hranimo v delovnem prostoru (ang. workspace).

V največji meri je okolje poznano po svoji odprtosti, za razvoj aplikacij, ki so napisane v programskem jeziku Java (ostali programski jeziki so podprti z vtičniki za IDE).

Projekt Eclipse je bil prvotno ustanovljen v novembru 2001 s strani podjetja IBM. Eclipse Foundation je bila ustanovljena leta 2004 kot neprofitna in neodvisna organizacija, ki naj bi služila kot pomočnik za Eclipse skupnost. Danes Eclipse združuje tako družbe kot posameznike v programski industriji.

Fundacija se financira s pomočjo članarin in nudi štiri storitve za Eclipse, in sicer IT infrastrukturo, IP upravljanje, razvoj in razvoj celotnega okolja (ang. Ecosystem Development).

#### **3.1 SQLite**

SQLite [19] je knjižnica (sistem za upravljanje s podatkovno bazo), ki temelji na jeziku in ogrodju (ang. database engine) SQL. Koda je prosto dostopna vsem, tako za privatno kot tudi komercialno uporabo, in je v javni domeni (ang. public domain).

## 3.2 SQLite Database Browser

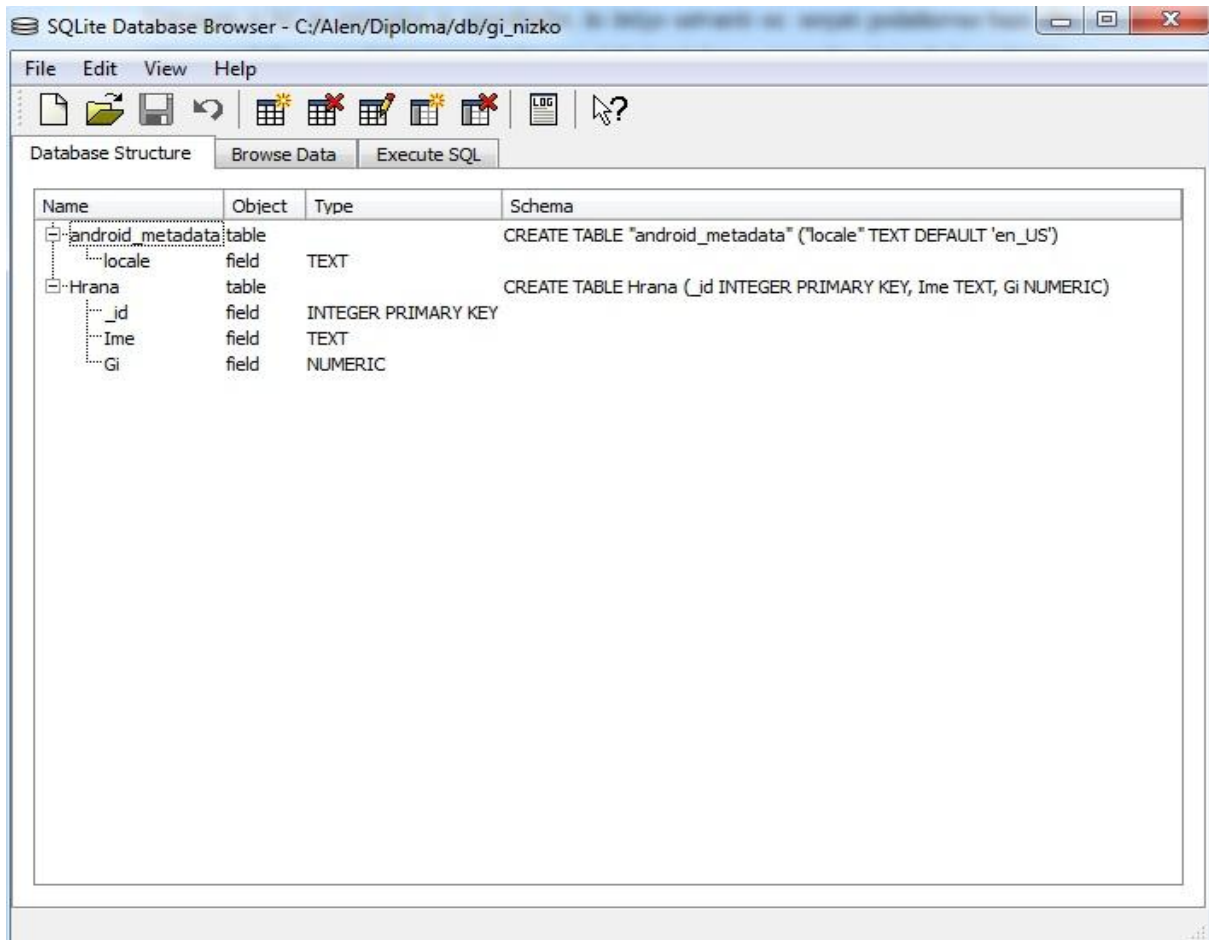
Gre za brezplačen, odprtokoden program, ki je namenjen upravljanju (ustvarjanje nove, dopolnjevanje obstoječe in urejanje) SQLite podatkovne baze [20].

Ustvarjen je bil za razvijalce in uporabnike, ki želijo ustvariti oz. urejati podatkovno bazo za iskanje podatkov s podobnimi preglednicami (tabelami) kar v vmesniku, brez da bi se bilo potrebno naučiti SQL ukaze.

Nudi nam:

- ustvarjanje zbirk;
- ustvarjanje, definiranje, spreminjanje in brisanje tabel;
- ustvarjanje, definiranje in brisanje indeksov (ang. index);
- brskanje, urejanje, dodajanje in brisanje vnosov;
- iskanje po podatkih;
- uvoz in izvoz (ang. import and export) teksta;
- uvoz in izvoz CSV datotek;
- uvoz in izvoz iz SQL datotek;
- SQL ukaze in poizvedbo podatkov.

Torej SQLite Browser teži k temu, da ostaja čim bolj preprost, da bi še naprej lahko nudil podporo tako razvijalcu kot končnemu uporabniku. Slika 3 prikazuje SQLite Database Browser:



Slika 3: SQLite Database Browser

## 4 IMPLEMENTACIJA MOBILNE APLIKACIJE

### 4.1 Delovno okolje

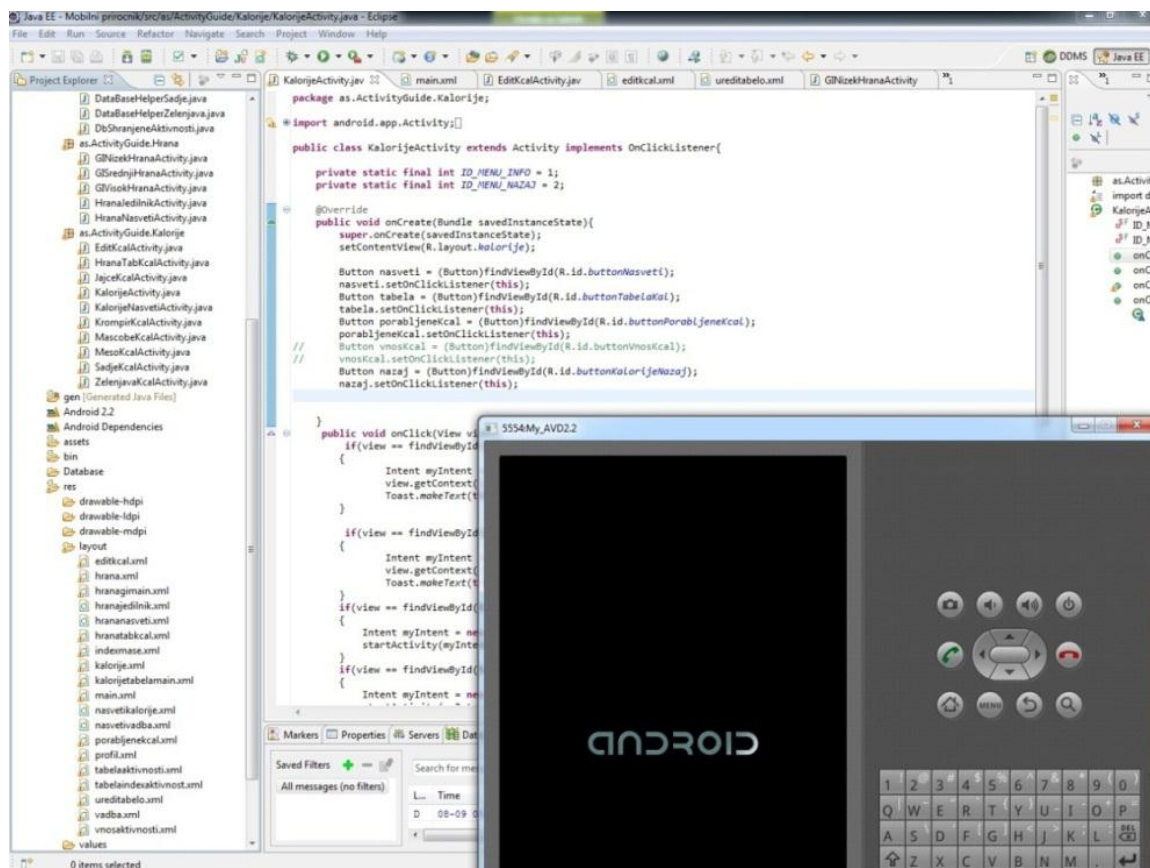
Za razvoj aplikacije smo izbrali orodje Android SDK, ki ga brezplačno dobimo na uradni strani Androida, kjer prav tako najdemo podrobna navodila za samo namestitev programskega okolja. Najprej smo prenesli JDK in ga namestili, nato smo namestili razvojno okolje Eclipse (okolje smo opisali v tretjem poglavju). Prenesli smo še paket SDK, ki ga je bilo potrebno razpakirati na trdi disk. V razvojnem okolju Eclipse smo nato namestili vtičnik ADT in vnesli pot do paketa SDK. V zadnjem koraku smo preko SDK/AVD upravljalnika prenesli in namestili potrebne pakete za razvoj aplikacije. Za potrebe podatkovne baze pa smo prenesli še SQLite in pa SQLite Database Browser, ki je prav tako brezplačno dostopen na spletu. Sam

SQLite Database Browser ne potrebuje nameščanja, ampak se lahko začne uporabljati takoj, ko se razpakira na disk.

Vse skupaj je bilo nameščeno na operacijskem sistemu Windows 7 Ultimate x64.

Aplikacijo smo razvili za platformo Android 2.2 in se lahko izvaja na vseh napravah z nameščenim sistemom Android 2.2 (najmanj) ali novejšim sistemom. Z AVD upravljalnikom smo ustvarili tudi emulator, ki smo ga uporabljali za preizkušanje aplikacije.

Če povzamemo, smo uporabili naslednje: JDK verzija 7, razvojno orodje Eclipse verzija 3.7.0, Android verzija 2.2, SQLite verzija 1.0.81.0 in pa SQLite Data Browser verzija 2.0.



Slika 4: delovno okolje

## 4.2 Načrt aplikacije

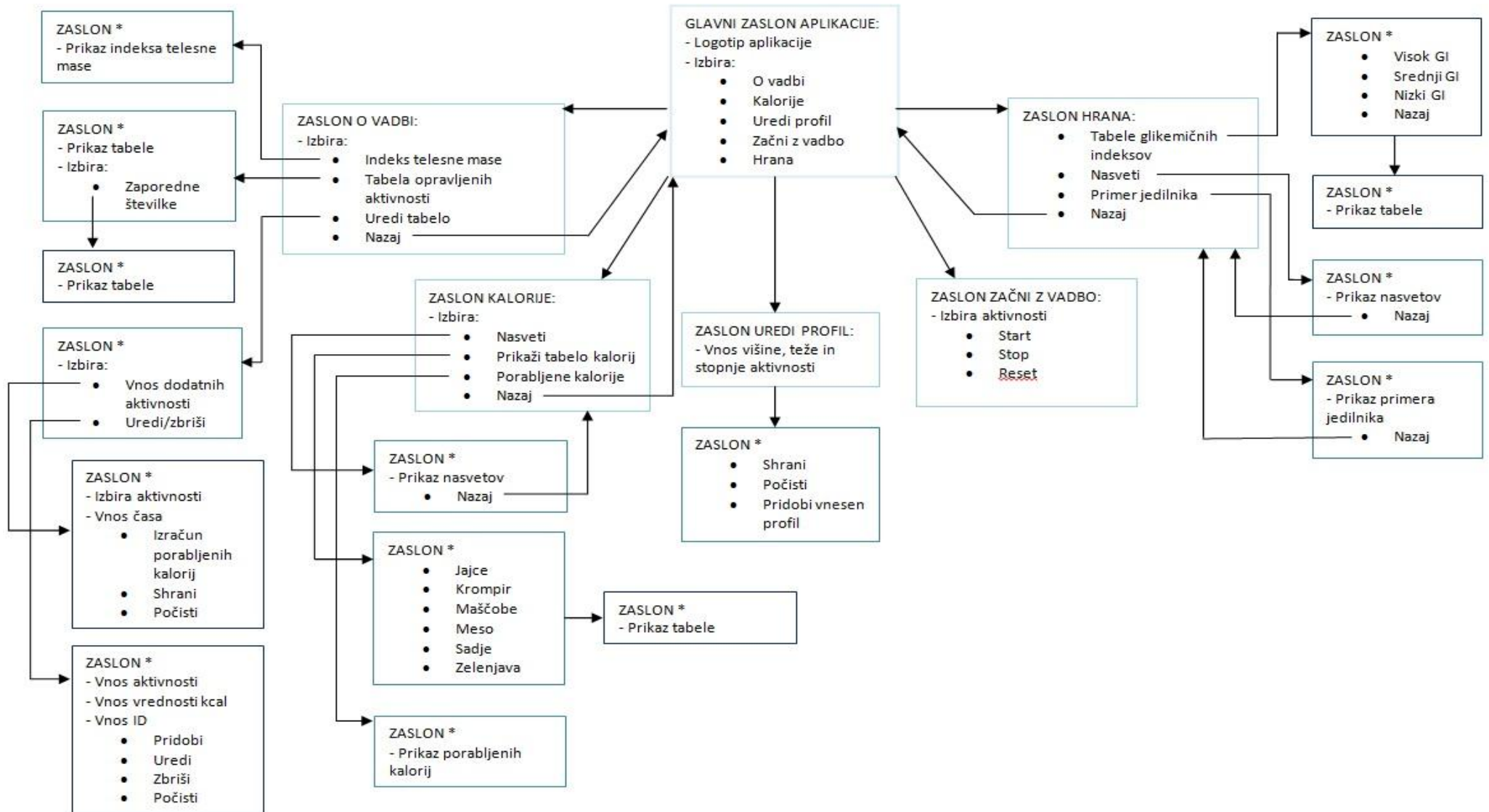
Zamisel in načrt končne podobe aplikacije sta bila prva koraka pri nastajanju le-te. Ker gre za aplikacijo, ki pokriva športno aktivnost, prehrano in kalorije, je osnovna funkcionalnost aplikacije računanje kalorij pri posamezni športni aktivnosti. Tako so tudi vse ostale stvari povezane s pridobivanjem oz. izgubljanjem kalorij. Ob takšni predpostavki smo aplikacijo razdelili na štiri dele. Prvi in najpomembnejši del je računanje kalorij, potem je tu še del o hrani, v katerem so predstavljena živila in njihov glikemični indeks, tretji del govori o kalorijah, v katerem so energijske vrednosti živil in pa zadnji del, del o vadbi, v katerem beležimo rezultate opravljenega dela. Povsem neodvisno od tega pa je bilo potrebno za pravilno delovanje aplikacije ustvariti tudi profil uporabnika, ki mora vsebovati podatke o višini, teži in stopnji aktivnosti uporabnika.

Za realizacijo takšnega scenarija smo morali ločeno razviti še podatkovno bazo oz. tabele s podatki, ki smo jih kasneje vključili v samo aplikacijo. Eno podatkovno bazo smo morali razviti tudi znotraj aplikacije, ta je bila namenjena shranjevanju opravljenih aktivnosti.

Aplikacijo smo razvijali tako, da ima še veliko možnosti za samo nadgradnjo. Predvsem v grafičnem pogledu ima še kar nekaj rezerve.

Narisali smo tudi shemo aplikacije, ki vsebuje vse glavne gradnike in prikazuje prehod med nivoji in povezavo med zasloni. Aplikacija naj bi tako vsebovala več zaslonov (19) in veliko možnosti izbire (gumbi, padajoči menu za izbiro ...). Ob zagonu aplikacije se najprej pokaže glavni zaslon, ki nam takoj ponudi izbiro za prehod na štiri glavne dele aplikacije in izbiro za urejanje profila. Glavni zaslon vsebuje še logo aplikacije, ki je bil ustvarjen zgolj za ta namen. Ob kliku na določeno izbiro se uporabnik premakne na nižji (izbran) nivo, ki ga lahko kadarkoli zapusti z izbiro nazaj ali pa nadaljuje na globlji nivo s poljubno izbiro na tem področju. Uporabnik lahko kadarkoli zapusti aplikacijo s pritiskom na tipko menu/izhod.

Shema funkcionalnega delovanja mobilne aplikacije ActivityGuide je predstavljena na sliki 5.



Slika 5: načrt aplikacije



### 4.3 Izdelava logotipa in gumbov

Logotip za aplikacijo smo izdelali s programom Adobe Photoshop, v katerem smo prav tako izdelali gumbe oz. njihovo grafično podobo. Izdelave smo se lotili v več korakih. Najprej smo na list papirja narisali skico logotipa in gumbov. Logotip smo sestavili kot Android robotka, ki se ukvarja s športno aktivnostjo. Ker je za Android »rezervirana« zelena barva, smo se tudi sami odločili, da bo le-ta osnovna barva našega logotipa. Logotip smo prenesli v program, kjer smo dodelali linije, napravili senčenje in ga pobarvali. Izdelali smo dve velikosti, in sicer eno, ki bo primerna za logotip, in drugo, ki bo služila kot ikona aplikacije.

Na sliki 6 je prikazan logotip aplikacije:



Slika 6: logotip aplikacije

Izdelali smo še grafično obliko gumbov, ki smo jih potrebovali za aplikacijo. Kreirali smo gumbe, ki vsebujejo tako napis kot sliko, in gumbe, ki imajo zgolj napis ali pa zgolj sliko. Tudi tukaj smo se odločili za kombiniranje z zeleno barvo. Uporabljali smo različna orodja (oblike po meri), dodali efekte senčenja in svetleče učinke (ang. glow effect). Oblika gumba je prikazana na sliki 7:



Slika 7: gumb uredi profil

## 4.4 Shranjevanje in pridobivanje podatkov

### 4.4.1 Pridobivanje podatkov iz SQLite podatkovne baze

Za potrebe aplikacije smo ločeno razvili podatkovno bazo (tabele s podatki), za izdelavo pa smo uporabili SQLiteDataBrowser. Naše SQLite baze vsebujejo dve tabeli, in sicer tabelo *android\_metadata*, ki je potrebna za pravilno delovanje in prepoznavanje baze znotraj Android aplikacije. Ta tabela vsebuje zgolj en element z imenom *locale*, ki je tipa text. Vsebuje pa tudi našo glavno tabelo z imenom *Hrana*, ki vsebuje tri elemente, in sicer *\_id*, ki je primarni ključ tipa integer, *Ime*, tipa text in pa *Kcal*, tipa numeric. Za potrebe tabel z glikemičnimi indeksi smo tabelo *Hrana* malo preuredili in tako vsebuje namesto stolpca *Kcal*, stolpec *Gi* ki je tipa numeric.

Vse baze so shranjene v mapi assets, da se lahko skupaj s programom prenesejo na mobilno napravo. Da smo lahko podatkovne baze uporabili (pridobili podatke iz njih), je bilo potrebno zgraditi primeren razred, tako imenovan pomočnik (ang. DataBase Helper), znotraj katerega smo implementirali vse potrebno za dostop in delo z bazami.

Tak pomožni razred za delo s podatkovnimi bazami smo zgradili za vsako bazo posebej, vključuje pa naslednje elemente:

- pot (ang. path) in ime baze;
- metodo *createDataBase()*, ki preveri, če baza že obstaja in jo v primeru, da ne obstaja, ustvari. To naredimo tako, da v mapi assets poišče ustrezno datoteko (datoteko, ki ustreza imenu baze) in jo kopiramo v novo ustvarjeno bazo;
- metodo *copyDataBase()*, ki poskrbi, da se baza prepíše iz enega mesta na drugo;
- metodo *openDataBase()*, ki nam odpre bazo in nam omogoči, da lahko delamo z njo;
- metodo *close()*, ki poskrbi za zapiranje baze;
- metodo *myQuery()*, ki kot argument dobi stavek za poizvedbo in nato poizvedbo tudi izvede in vrne rezultate.

Primer kode razreda sledi spodaj (gre za razred *DataBaseHelperMeso.java*):

```
public class DataBaseHelperMeso extends SQLiteOpenHelper{  
  
    private static String DB_PATH = "/data/data/as.ActivityGuide/databases/";  
    private static String DB_NAME = "meso";  
    private SQLiteDatabase myDataBase;  
    private final Context myContext;
```

```

public DataBaseHelperMeso(Context context) {
    super(context, DB_NAME, null, 1);
    this.myContext = context;
}

public void createDataBase() throws IOException{

    boolean dbExist = checkDataBase();

    if(dbExist){
    }
    else{
        this.getReadableDatabase();

        try {

            copyDataBase();

        } catch (IOException e) {

            throw new Error("Error copying database");

        }

    }

}

private boolean checkDataBase(){

    SQLiteDatabase checkDB = null;

    try{

        String myPath = DB_PATH + DB_NAME;
        checkDB = SQLiteDatabase.openDatabase(myPath, null, SQLiteDatabase.OPEN_READONLY);

    }catch(SQLiteException e){
    }
    if(checkDB != null){

        checkDB.close();

    }
    return checkDB != null ? true : false;
}

private void copyDataBase() throws IOException{

    InputStream myInput = myContext.getAssets().open(DB_NAME);

    String outFileName = DB_PATH + DB_NAME;

    OutputStream myOutput = new FileOutputStream(outFileName);

    byte[] buffer = new byte[1024];
    int length;
    while ((length = myInput.read(buffer))>0){
        myOutput.write(buffer, 0, length);
    }

    myOutput.flush();
    myOutput.close();
    myInput.close();

}

public void openDataBase() throws SQLException{

    String myPath = DB_PATH + DB_NAME;
    myDataBase = SQLiteDatabase.openDatabase(myPath, null, SQLiteDatabase.OPEN_READONLY);

}

@Override
public synchronized void close() {

    if(myDataBase != null)
        myDataBase.close();

    super.close();

}

@Override
public void onCreate(SQLiteDatabase db) {
    // TODO Auto-generated method stub

```

```

    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // TODO Auto-generated method stub
    }

    public Cursor myQuery(String myquery)
    {
        try{
            Cursor c = myDataBase.rawQuery(myquery,null);
            return c;
        }
        catch(Exception x)
        {
            return null;
        }
    }
}

```

Primer kode 1: razred DataBaseHelper

#### 4.4.2 Ustvarjanje podatkovne baze znotraj programa

Za shranjevanje opravljenih aktivnosti smo znotraj programa razvili podatkovno bazo *tabelaAktivnosti*, ki vsebuje tabelo vadba in shranjuje naslednje vrednosti: *\_id*, ki je primarni ključ tipa integer, *aktivnosti*, tipa string, in pa *cas* ter *vrednost\_kcal*, tipa double.

Razred *DbShranjeneAktivnosti*, ki ustvari podatkovno bazo, vsebuje naslednje elemente:

- metodo *onCreate()*, ki ustvari podatkovno bazo z vrednostmi, ki so podane v programu.
- metodo *onUpgrade()*, ki posodobi obstoječo bazo;
- metodo *ustvariVnos()*, ki v bazo shrani podatke;
- metodo *getData()*, ki nam vrne podatke;
- metodo *getPorabljeneKcal()*, ki nam vrne seštevek vseh porabljenih kalorij;
- metodo *getCas()* in *getKcal()*, ki nam vrmeta čas in kalorije, ki so shranjene v bazi;
- metodo *posodobiVnos()*, ki posodobi že obstoječi vnos;
- metodo *izbrišiVnos()*, ki izbriše željen vnos.

```

public static final String KEY_ROWID = "_id";
public static final String KEY_NAZIV = "aktivnost";
public static final String KEY_CAS = "cas";
public static final String KEY_VREDNOST = "vrednost_kcal";
private static final String DATABASE_NAME = "tabelaAktivnosti";
private static final String DATABASE_TABLE = "vadba";

```

Primer kode 2: spremenljivke za ustvarjanje baze

```

public void onCreate(SQLiteDatabase db) {

```

```

db.execSQL("CREATE TABLE " + DATABASE_TABLE + " (" +
    KEY_ROWID + " INTEGER PRIMARY KEY AUTOINCREMENT," +
    KEY_NAZIV + " STRING NOT NULL, " +
    KEY_CAS + " DOUBLE NOT NULL, " +
    KEY_VREDNOST + " DOUBLE NOT NULL);"
);
}

```

Primer kode 3: metoda *onCreate()*

```

public long ustvariVnos(String aktivnost, double cas, double vrednostKcal) {
    ContentValues cv = new ContentValues();
    cv.put(KEY_NAZIV, aktivnost);
    cv.put(KEY_CAS, cas);
    cv.put(KEY_VREDNOST, vrednostKcal);
    return myDatabase.insert(DATABASE_TABLE, null, cv);
}

```

Primer kode 4: metoda *ustvariVnos()*

```

public double getPorabljeneKcal()
{
    String[] stolpci = new String[]{ KEY_ROWID, KEY_NAZIV, KEY_CAS, KEY_VREDNOST};
    Cursor c = myDatabase.query(DATABASE_TABLE, stolpci, null, null, null, null, null);
    double rezultat = 0;

    int iVrednost = c.getColumnIndex(KEY_VREDNOST);

    for(c.moveToFirst(); !c.isAfterLast(); c.moveToNext()){
        rezultat = rezultat + Double.parseDouble(c.getString(iVrednost));
    }
    return rezultat;
}

```

Primer kode 5: metoda *getPorabljeneKcal()*

## 4.5 Implementacija nekaterih funkcionalnosti

### 4.5.1 Implementacija aktivnosti (ang. *activity*)

Zaradi razvejanosti aplikacije smo razvili več aktivnosti. Osnovna aktivnost se imenuje *ActivityGuideApp*, ki vodi glavni tok programa. Aktivnost se kliče ob inicializaciji programa in poskrbi za gradnjo glavnega zaslona (*main.xml*).

### 4.5.2 Implementacija opcije/tipke menu

Zaradi boljše preglednosti in lažje uporabe aplikacije smo se odločili implementirati dodatno možnost, in sicer dodatne izbire ob uporabi tipke menu. Gre za fizično tipko, ki jo načeloma vsebujejo Android naprave (od Android 3.0 pa ta tipka ni več obvezna [21]). Menu je bil prvotno razvit oz. namenjen izbiri »akcij« oz. novih aktivnosti, kot so iskanje, nastavitve in podobno. Vendar pa smo se v našem primeru odločili, da ga uporabimo bolj za pomoč in lažje prehajanje po nivojih. Tako nam (odvisno od nivoja) največkrat ponudi opcijo nazaj (vrnemo

se na prejšnjo aktivnost), opcijo izhod, ki zapre aplikacijo, in pa pomoč, ki nam na danem nivoju razloži pomen in pravilno uporabo aplikacije.

Gumb menu je v android aplikacijah že implementiran, vendar brez izbir (prazen), zato je potrebno metodo ob implementaciji prepisati. Sledi primer implementacije menu-ja z izbiro nazaj in izhod.

```
private static final int ID_MENU_IZHOD = 1;
private static final int ID_MENU_NAZAJ = 2;

@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    MenuItem item1 = menu.add(Menu.NONE, ID_MENU_IZHOD, Menu.NONE,
        R.string.opcijaIzhod);
    MenuItem item2 = menu.add(Menu.NONE, ID_MENU_NAZAJ, Menu.NONE,
        R.string.opcijaNazaj);

    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    if(item.getItemId() == ID_MENU_NAZAJ)
    {
        Intent myIntent = new Intent(KalorijeActivity.this, ActivityGuideApp.class);
        startActivity(myIntent);
    }

    if(item.getItemId() == ID_MENU_IZHOD)
    {
        this.finish();
        return true;
    }

    return false;
}
```

Primer kode 6: implementacija menu-ja

#### 4.5.3 Implementacija spustnega menu-ja (ang. spinner oz. drop down list)

Gradnik spustni menu (ang. spinner) smo v aplikaciji uporabili za izbiro želene športne panoge. Gradnik nam ponudbi izbiro, ki jo potrdimo s klikom. Potrebno je ustvariti polje z naborom vrednosti, ki jih gradnik vsebuje. V našem primeru so le-te tipa String. Nato pa je potrebno gradnik še implementirati in mu določiti poslušalca (ang. listener).

Sledi primer implementacije gradnika *spinner*:

```
String[] moznosti= {"Aerobika", "Badminton"};
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item,
    moznosti);

Spinner spinner = (Spinner)findViewById(R.id.spinner1);
spinner.setAdapter(adapter);
```

```
spinner.setOnItemSelectedListener(this);

if(spinner.getSelectedItemPosition() == 0)
{
    //če je izbrana aerobika
}

```

Primer kode 7: implementacija *spinner*-ja

#### 4.5.4 Implementacija gumbov (ang. *button*)

V naši aplikaciji so gumbi osnovni gradniki, zato se tudi največkrat ponovijo. Omogočijo nam možnost izbire, in sicer največkrat za izbiro nove aktivnosti, lahko pa tudi za shranjevanje, brisanje in pridobivanje podatkov. Za pravilno uporabo gumba je le-tega potrebno najprej implementirati, za odzivanje na pritisk pa mu je potrebno določiti poslušalca.

Sledi primer implementacije gumba, ki zažene štoparico:

```
Button start= (Button)findViewById(R.id.start);
start.setOnClickListener(this);

public void onClick(View view) {
    if (view == findViewById(R.id.start)){
        stoparica.start();
    }
}

```

Primer kode 8: implementacija gumba

#### 4.5.5 Implementacija sporočil (ang. *toast message*)

Za boljšo preglednost in lažje sledenje aplikaciji (in našim akcijam), smo se odločili implementirati sporočila/obvestila. Obvestila se največkrat pokažejo ob spremembi aktivnosti, to je ob pritisku na gumb. Delujejo pa izključno zato, da obveščajo. Za uporabo gradnika Toast je potrebno vključiti knjižnico `android.widget.Toast`.

Sledi primer obvestila, ki nam pove da smo izbrali opcijo začetek vadbe:

```
import android.widget.Toast;

Toast.makeText(this, "Izbrali ste Začetek vadbe!", Toast.LENGTH_SHORT).show();

```

Primer kode 9: implementacija obvestila

#### 4.5.6 Implementacija opozoril (ang. alert box)

Opozorilna polja smo v aplikacijo vključili zaradi boljše preglednosti shranjevanja aktivnosti. Tako nam ponudi rezultat kot sporočilo polja in izbiro za zapustitev opozorila. V našem primeru lahko zapustimo opozorilo tako, da shranimo rezultat ali pa brez shranjevanja.

Sledi primer implementacije opozorilnega polja, ki nam ponudi dve opciji za izhod:

```
AlertDialog alertDialog = new AlertDialog.Builder(VadbaActivity.this).create();

alertDialog.setTitle("Bravo!");
alertDialog.setMessage("Aerobika je trajala: " + cas + " minut.");

alertDialog.setButton(Dialog.BUTTON_NEUTRAL, "Nazaj (brez shranjevanja)", new
DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog, int which) {
    }
});

alertDialog.setButton(Dialog.BUTTON_POSITIVE, "Shrani aktivnost!", new
DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog, int which) {
    }
});
```

Primer kode 10: implementacija opozorila

## 4.6 Hrana, kalorije in vadba

V aplikaciji smo pod poglavjem Hrana in Kalorije predstavili nekatere vrednosti živil. Tako smo pri hrani prikazali tabelo glikemičnih indeksov živil, pri kalorijah pa kalorijske vrednosti živil. Prav tako pa smo se na kalorije oz. računanje le-teh osredotočili pri vadbi sami. Zato bomo v naslednjih podpoglavjih predstavili nekatere pojme in prikazali model računanja porabe kalorij.

### 4.6.1 Glikemični indeks

Glikemični indeks [11] je število, ki je običajno omejeno med vrednostma 0 in 100, vendar je lahko tudi višje. Gre za indeks, ki živila razvršča glede na to, kako zaužitje določenega živila vpliva na dvig ravni sladkorja v krvi v primerjavi z referenčnim živilom. Referenčno živilo ima indeks 100 in običajno predstavlja raztopino fruktoze. Večji kot je indeks, večji je dvig sladkorja v krvi pri zaužitju tega živila.

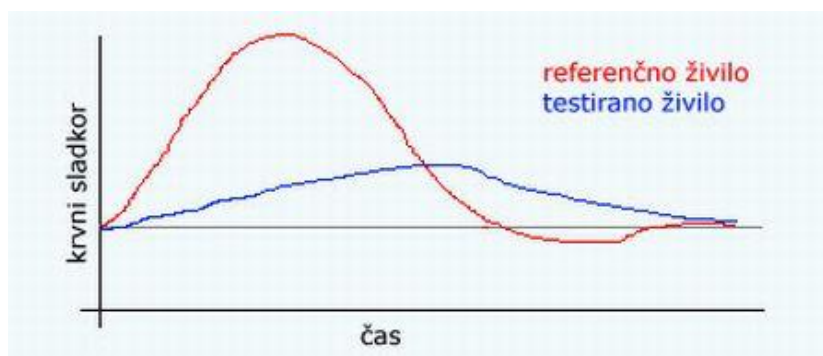


#### 4.6.2 Kako se meri glikemični indeks?

Za vsako živilo je potrebno dobiti najmanj deset zdravih prostovoljcev. Ti nato zjutraj na tešče zaužijejo živilo, za katerega se določa glikemični indeks. To živilo mora vsebovati natanko 50 g prebavljivih ogljikovih hidratov. Prostovoljcem se nato 2 uri, vsakih 15–30 minut, meri krvni sladkor. Rezultate meritev se zapisuje na graf, na katerem se na koncu izriše krivulja naraščanja sladkorja v krvi. Za konec se izračuna še površino pod dobljeno krivuljo (pŽivila).

Enak postopek se na isti skupini testirancev ponovi še enkrat (ne na isti dan), le da tokrat testiranci zaužijejo 50 g glukoze. Tudi pri tem testu se na koncu izračuna površino pod krivuljo (pGlukoze).

Glikemični indeks prestavlja razmerje med pŽivila in pGlukoze, ki ga pomnožimo s 100. Končni glikemični indeks je povprečje indeksov vseh testirancev. Grafični prikaz merjenja, je viden na sliki 8:



Slika 8: merjenje glikemičnega indeksa

#### 4.6.3 Pomen glikemičnega indeksa

Ogljikovi hidrati so primarni vir energije za delovanje telesa. Po zaužitju se ogljikovi hidrati iz hrane najprej prebavijo do enostavnih sladkorjev in nato absorbirajo v krvni obtok, zaradi česar pride do porasta krvnega sladkorja. Delovanje telesa je najboljše, če je nivo krvnega sladkorja relativno konstanten, če torej ni večjega nihanja. V primeru, da krvni sladkor močno pade, se počutimo brez energije in postanemo lačni. Če se nivo krvnega sladkorja nenadoma poveča pa se iz trebušne slinavke poveča izločanje hormona inzulina. Inzulin znižuje količino sladkorja v krvi, poleg tega pa odvečni sladkor spreminja v trigliceride, ki se nato shranjujejo v maščobne celice, zaradi česar se redimo.

Večji in hitrejši kot je dvig sladkorja v krvi, več inzulina se sprosti, temu pa sledi večja pretvorba v trigliceride in hiter padec krvnega sladkorja. Najhitrejši dvig sladkorja povzročajo živila z visokim glikemičnim indeksom.

Uživanje živil z nizkim glikemičnim indeksom torej onemogoča nihanje krvnega sladkorja in s tem inzulina, posledično pa tudi nihanje v počutju ter je pomemben mehanizem za kopičenje odvečne telesne maščobe.

#### 4.6.4 Kalorije

S kalorijami merimo energijsko vrednost hrane. Največkrat govorimo o kcal, torej o kilokalorijah. Gre za mersko enoto, ki določa energijo, ki jo vsebuje hrana [12].

#### 4.6.5 Kako izračunamo kalorično vrednost jedi?

Pri izračunavanju kaloričnosti jedi gre za seštevanje energijske vrednosti posameznih sestavin po makro hranilih, in sicer:

- 1 gram ogljikovih hidratov ima 4 kcal, prav tako ima 4 kcal tudi 1 gram beljakovin;
- 1 gram maščob pa ima 9 kcal.

#### 4.6.6 Računanje porabljenih kalorij

Merjenje energijske porabe pri različnih aktivnostih je zelo zahtevno in zelo težko je povsem natančno opraviti take meritve. Zato smo se odločili, da si bomo pomagali z Bannisterjevo tabelo, ki kaže energijsko porabo pri različnih telesnih aktivnostih, le-ta je izražena v kcal/min po kilogramu telesne teže. Te podatke smo obdelali in priredili našim potrebam, tako smo dobili koeficiente, ki v našem primeru predstavljajo porabo kcal vsako minuto za 10 kg telesne mase. Ti koeficienti so tako osnova za celotno računanje porabljenih kalorij. Gre za približne izračune.

V tabeli 3 so koeficienti, ki se uporabljajo v aplikaciji:

ŠPORTNA AKTIVNOST	KOEFICIENT [kcal /10kg min]
aerobika	0.8

<b>badminton</b>	0.7
<b>biljard</b>	0.4
<b>borilne veščine</b>	2
<b>dvigovanje uteži</b>	0.5
<b>hoja – počasi</b>	0.4
<b>hoja – zmerna</b>	0.6
<b>hoja – hitra</b>	1.3
<b>kolesarjenje – rekreativno</b>	1.3
<b>košarka, nogomet, odbojka (igra)</b>	1.3
<b>planinarjenje</b>	1
<b>plavanje</b>	1
<b>pomivanje tal</b>	0.094
<b>sedenje s pogovorom</b>	0.04
<b>sobno kolo</b>	1.2
<b>spanje</b>	0.03
<b>tenis – rekreativno</b>	1.3
<b>tek 9.5 km/h</b>	1.5
<b>tek 13.5 km/h</b>	1.9
<b>tek 16 km/h</b>	2.6

Tabela 3: koeficienti za računanje kalorij

Poraba kalorij pri različnih športnih aktivnostih predstavlja bistven element pri ustvarjanju energijskega dnevnega primanjkljaja, ki se mora zgoditi, če želimo telesno težo znižati. Primerna dnevna poraba kalorij na račun telesno-gibalnih aktivnosti naj bi znašala od 500 do 1000 kcal.

#### 4.6.7 Indeks telesne mase

Indeks telesne mase [7] je številka, s katero lahko relativno dobro ocenimo podhranjenost ali debelost odrasle osebe. Izračun je priporočljiv za osebe starejše od 19 let, mlajšim se odsvetuje, saj telo še ni dokončno razvito. Indeks predstavlja količnik med težo (v kilogramih) in telesno višino (v metrih) na kvadrat.

V aplikaciji uporabljamo spodnjo tabelo 4 za preverjanje rezultatov:

INDEKS TELESNE MASE	RAZLAGA INDEKSA
manj kot 18.5	podhranjenost
18.5 – 24.9	normalna hranjenost
25.0 – 29.9	prekomerna telesna teža
več kot 30	debelost

Tabela 4: indeks telesne mase

ITM dobro ocenjuje delež telesne maščobe, vendar ga ne smemo uporabljati kot natančen izračun. Obstajajo korelacije med ITM in deležem telesne maščobe, ki so odvisne od spola in starosti. Ženske imajo v povprečju večji delež maščobe kot moški. Prav tako ima starejša oseba z istim indeksom kot mlajša večji delež maščobe. Ocena ITM ni primerna za trenirane športnike, ki imajo veliko mišične mase in majhen delež maščobnega tkiva. Kot vsaka podobna metoda je tudi klasifikacija hranjenosti pri ITM izračunana na podlagi povprečja – ker je ITM odvisen le od teže in višine in ne upošteva deleža maščobe, imajo trenirani športniki velik ITM, vendar zaradi tega nikakor niso debeli.

## 5 TEST APLIKACIJE

Aplikacijo smo testirali na dveh napravah. Prva naprava je bil emulator za Android 2.2 znotraj razvojnega okolja Eclipse. Druga pa je bila fizična naprava Samsung I9100 Galaxy S II. Testiranje je potekalo preko testnih primerov in naključnega testiranja.

### 5.1 Testni primeri

S testnimi primeri smo preizkusili celotno funkcionalnost:

- zagon aplikacije, izbira Uredi profil. Testiranje možnosti vnosa, testiranje delovanja gumba Shrani in Počisti, pregled delovanja tipke menu in izbira izhod;

- zagon aplikacije, izbira Uredi profil. Testiranje pravilnega delovanja gumba Pridobi profil in pregled pravilnosti podatkov, pregled delovanja tipke menu in opcije nazaj;
- zagon aplikacije, izbira Hrana, izbira vseh opcij pod izbiro hrana (tabela glikemičnih indeksov, nasveti, primer jedilnika) do zadnjega nivoja. Testiranje pravilnosti delovanja gumbov, pravilnosti prikaza podatkov in delovanje tipke menu;
- zagon aplikacije, izbira O Vadbi, pregled delovanja gumbov Indeks telesne mase in Tabela opravljenih aktivnosti. Pregled pravilnega prikazovanja ter izračuna indeksa telesne mase;
- zagon aplikacije, izbira O vadbi, izbira Uredi tabelo, pregled možnosti dodajanja in urejanja podatkov;
- zagon aplikacije, izbira Kalorije, testiranje vseh gumbov in pregled pravilnega prikaza iz podatkovne baze;
- zagon aplikacije, izbira Začni z vadbo, pregled opcij vadbe, pregled delovanja gumbov Start, Stop in Reset. Pregled pravilnega shranjevanja v podatkovno bazo oz. izhod brez shranjevanja;
- zagon aplikacije, sprehod po vseh nivojih in testiranje gumba menu;
- v zadnji fazi je testiranje potekalo naključno, izbira poljubne opcije, premikanje po nivojih (naprej in nazaj) in opazovanje pričakovanih rezultatov.

## 5.2 Testni mobilni telefon

Za testni mobilni telefon se je uporabljal Samsung I9100 Galaxy S II [30]. Nameščen ima operacijski sistem Android 2.3, ima kapacitivni Super AMOLED Plus zaslon resolucije 480 x 800 in Dual-core 1.2 GHz Cortex-A9 procesor.

Slika telefona je predstavljena na sliki 9, specifikacije pa so zapisane v tabeli 5:



Slika 9: Samsung Galaxy SII

	<b>OPIS</b>
<b>splošno</b>	izdan aprila 2011
<b>dimenzije</b>	125.3 x 66.1 x 8.5
<b>zaslon</b>	Super amoled plus, 16M barv, resolucija 480 x 800, 4.3 palčni
<b>spomin</b>	16Gb, 1Gb RAM
<b>kamera</b>	8MP, resolucija 3264 x 2448 pik
<b>povezovanje</b>	Bluetooth, USB, Internet HTML brskalnik, SyncML, WiFi, TV-izhod, PC Sync
<b>operacijski sistem</b>	Android 2.3 Gingerbread
<b>baterija in napajanje</b>	kapaciteta 1650mAh, do 1100min pogovora (2G) in 520min (3G)
<b>senzorji</b>	pospeškometer, RGB senzor svetlobe, barometer, senzor bližine, kompas

Tabela 5: specifikacije Samsung Galaxy SII

### 5.3 Testiranje aplikacije

Testiranje aplikacije je najprej potekalo na emulatorju Android 2.2. Uporabili smo testne primere in jih po vrsti preizkusili. Pri testiranju ni bilo večjih težav, aplikacija je delovala po pričakovanjih, prehod med aktivnostmi je bil tekoč. Nekaj težav se je pokazalo le pri podatkovni bazi, vendar smo le-te odpravili, kajti ugotovili smo, da je bilo potrebno le počistiti spomin in aplikacijo ter bazo ponovno naložiti. V zadnji fazi testiranja smo uporabili zadnji testni primer in aplikacijo še naključno testirali, tako smo vsako izbiro večkrat izbrali, se sprehodili skozi celotno aplikacijo in popravili še zadnje nepravilnosti (neustrezen napis, trajanje »toast« sporočila ...).

Nato smo postopek ponovili še na fizični napravi. Aplikacija se je izvajala nekoliko bolj tekoče, delovala je stabilno in je ob pravilni uporabi vračala pravilne (realne) podatke.

## 6 REZULTATI

Izdelali smo aplikacijo, ki nam služi kot priročnik o vadbi in prehrani, hkrati pa nam omogoča, da pri aktivnosti preverimo svojo uspešnost. Aplikacija nam takoj po zagonu omogoča urejanje (nastavitev) profila, kar je ključnega pomena za pravilno delovanje aplikacije in izračunov znotraj aplikacije. Hkrati pa nam s pritiskom na logotip aplikacije ponudi štiri, glavne funkcionalnosti.

Na sliki 10 je prikazan začetni zaslon ob zagonu aplikacije.



Slika 10: začetni zaslon

S pritiskom na logotip pridobimo glavni zaslon, znotraj aplikacije. Ta nam ponudi možnost izbire med področji Začni z vadbo, Hrana, Kalorije in O vadbi. Z izbiro posameznega področja so nam na voljo nove izbire. Pri področju Začni z vadbo najprej izberemo zeleno športno panogo in nato pričnemo z vadbo. Ob koncu vadbe nam aplikacija poda rezultat uspešnosti vadbe.

Primer rezultata pri aktivnosti je prikazan na sliki 11:





Slika 11: opravljena aktivnost

Pri ostalih področjih pa izvemo stvari bolj informativne narave. Ob izbiri področja o hrani ali o kalorijah izvemo koristne informacije o tem področju. Podane so tabele s kaloričnimi vrednosti živil, da lahko lažje oblikujemo svoj jedilnik, možnost pa imamo pogledati tudi glikemične indekse živil, kar nam daje še bolj popolno sliko pri prehranjevanju. Program nam omogoča tudi preverjanje našega indeksa telesne mase in poda rezultate oz. obvestilo glede na njegovo vrednost in stopnjo aktivnosti. Preverimo lahko še skupno uspešnost vadbe (porabo vseh kalorij med vadbo).

Slika 11 predstavlja prikaz različnih aktivnosti (ang. *Activity*), primer prikaza glikemičnih indeksov živil, podatke o indeksu telesne mase in nasvete za vadbo.

Vrednost glikemičnega indeksa!		Indeks telesne mase!	Nasveti o hranili!
Vrednost	Ime	Indeks telesne mase je:	*** **
37	Paradižnikova juha	22.3	*** **
35	Jabolka in hruške		Hrana zagotavlja energijo za vzdrževanje telesne temperature in človeške aktivnosti. Omogoča normalno rast in obnovo organizma.
34	Graham testenine		*** **
34	Energijska tablica		Ogljikovi hidrati so količinsko najpomembnejši vir energije v prehrani in so odločilni za nemoteno delovanje centralnega živčnega sistema (glukoza). Zagotavljajo polnjenje energijskih rezerv.
33	Čokoladno mleko		*** **
32	Posneto mleko		ce je količina OH v prehrani prevelika, se ti lahko spremenijo v masco. Energijska vrednost OH je: 1g OH = 3.5 kcal.
31	Sojino mleko		*** **
31	Suho sadje		Glavni viri OH so: izdelki iz zita, krompir, riz in sladkorji.
30	Stročji fižol	<i>manj kot 18.5 - Podhranjenost</i>	*** **
29	Banana (polzrela)	<i>18.5 - 24.9 - Normalna hranjenost</i>	Povprečna dnevna poraba OH znasa okoli 500 gramov, pri sportnikih do 1000 gramov.
29	Leča	<i>25.0 - 29.9 - Prekomerna telesna masa</i>	*** **
27	Mleko (polnomastno)	<i>več kot 30 - Debelost</i>	Glikemični indeks nam pove, kako hitro OH dvignejo krvni sladkor.
25	Grenivka		*** **
24	Breskve		zivila z nizjim GI omogocajo počasnejše prehajanje sladkorja v kri, s tem se izognemo visokom sladkorja, ki bi se pretvoril v masco.
23	Ječmen	<b>Tabela indeksov:</b>	
23	Slive	<b>Vse je super! Bravo!</b>	
22	Fruktoza		
22	Češnjice		
18	Soja		
16	Arašidi		
15	Nemasten jogurt 0,1%		

Slika 12: glikemični indeksi, indeks telesne mase in nasveti

## **7 SKLEP**

V diplomskem delu smo spoznali mobilno platformo Android in potek razvoja mobilne aplikacije. Razvili smo informativno, mobilno aplikacijo (priročnik) za Android OS – ActivityGuide. Razvoj in testiranje je potekalo v razvojnem okolju Eclipse. Omogoča nam spremljanje porabe kalorij pri posameznih aktivnostih in nudi informacije s področja prehrane in vadbe. V diplomskem delu je predstavljen potek izdelovanja aplikacije, predstavljeni pa so tudi ključni gradniki. Na koncu je opisan še postopek testiranja.

V prihodnosti bomo mobilno aplikacijo še nadgradili, povečali bomo bazo podatkov in nabor aktivnosti, dopolnili grafično podobo in dodali nove funkcionalnosti ter jo mogoče ponudili ostalim uporabnikom preko spletne trgovine Google Play.

Za konec naj dodam še osebno mnenje glede platforme Android. Menim, da gre za hitro razvijajočo platformo, ki ponuja skoraj vse, kar si uporabnik zaželi, hkrati pa razvijalcem omogoča dobro podporo in veliko skupnost ljudi, kar razvijanje še nekoliko olajša.

## 8 VIRI, LITERATURA

- [1] Juan-Manuel Fluxà, Using your own SQLite database in Android applications, dostopno na:  
<http://www.reigndesign.com/blog/using-your-own-sqlite-database-in-android-applications/>  
[zadnji obisk: 25. 7. 2012]
- [2] Bon&Žur, Poraba in vnos kalorij, dostopno na:  
<http://www.bonzur.si/rekreacija-sport/poraba-in-vnos-kalorij/>  
[zadnji obisk: 25. 7. 2012]
- [3] Hujšanje in sprememba načina življenja, Poraba kalorij pri športu, dostopno na:  
<http://www.hujsanje.net/osebnitrener/poraba-kalorij-pri-sportu/>  
[zadnji obisk: 25. 7. 2012]
- [4] Stackoverflow, Get time of chronometer widget, dostopno na  
<http://stackoverflow.com/questions/526524/android-get-time-of-chronometer-widget>  
[zadnji obisk: 26. 7. 2012]
- [5] Hujšaj zdravo, Poraba kalorij in telesna aktivnost, dostopno na:  
[http://hujšaj-zdravo.si/sport\\_in\\_vadba/oblikovanje\\_telesa/poraba\\_kalorij\\_pri\\_sportu/](http://hujšaj-zdravo.si/sport_in_vadba/oblikovanje_telesa/poraba_kalorij_pri_sportu/)  
[zadnji obisk: 26. 7. 2012]
- [6] Primusweb, Fitness Partner, dostopno na  
<http://www.primusweb.com/fitnesspartner/weight.htm>  
[zadnji obisk: 30. 7. 2012]
- [7] Osebni trener, Indeks telesne mase, dostopno na:  
[http://www.osebnitrener.com/index.php?option=com\\_chronocontact&Itemid=94](http://www.osebnitrener.com/index.php?option=com_chronocontact&Itemid=94)  
[zadnji obisk: 2. 8. 2012]
- [8] Hujšaj zdravo, Kalorije tabela, dostopno na:

- [http://www.hujsaj-zdravo.si/zdravje/izracuni/kalorije\\_tabela\\_tabela\\_kalorij/#.UCjohZ0aPT9](http://www.hujsaj-zdravo.si/zdravje/izracuni/kalorije_tabela_tabela_kalorij/#.UCjohZ0aPT9)  
[zadnji obisk: 3. 8. 2012]
- [9] Ozmec T. Kako sem izklesala svoje telo. Ptuj: Trgovina mojStil, 2009
- [10] It&c solutions, How to add a menu, dostopno na:  
[http://www.zdravoznaravo.si/index.php?route=extras/blog/getblog&blog\\_id=107](http://www.zdravoznaravo.si/index.php?route=extras/blog/getblog&blog_id=107)  
[zadnji obisk: 3. 8. 2012]
- [11] Cenim se, glikemični indeks živil, dostopno na:  
<http://www.cenim.se/115-a.html>  
[zadnji obisk: 9. 8. 2012]
- [12] Cenim se, Te presnete kalorije, dostopno na:  
<http://www.cenim.se/130-pr.html>  
[zadnji obisk: 9. 8. 2012]
- [13] Google Play, dostopno na:  
[http://en.wikipedia.org/wiki/Google\\_Play](http://en.wikipedia.org/wiki/Google_Play)  
[zadnji obisk: 13. 8. 2012]
- [14] Smartphone statistic and marcet, dostopno na:  
<http://www.email-marketing-reports.com/wireless-mobile/smartphone-statistics.htm>  
[zadnji obisk: 12. 8. 2012]
- [15] Android, the world's most popular mobile platform, dostopno na:  
<http://developer.android.com/about/index.html>  
[zadnji obisk: 9. 8. 2012]
- [16] Android (operating system), dostopno na:  
[http://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))  
[zadnji obisk: 9. 8. 2012]

- [17] Android version history, dostopno na:  
[http://en.wikipedia.org/wiki/Android\\_version\\_history](http://en.wikipedia.org/wiki/Android_version_history)  
[zadnji obisk: 30. 7. 2012]
- [18] Android software development, dostopno na:  
[http://en.wikipedia.org/wiki/Android\\_software\\_development](http://en.wikipedia.org/wiki/Android_software_development)  
[zadnji obisk: 25. 7. 2012]
- [19] About SQLite, dostopno na:  
<http://www.sqlite.org/about.html>  
[zadnji obisk: 10. 8. 2012]
- [20] SQLite Database Browser, dostopno na:  
<http://sqlitebrowser.sourceforge.net/>  
[zadnji obisk: 9. 8. 2012]
- [21] Android menus, dostopno na:  
<http://developer.android.com/guide/topics/ui/menus.html>  
[zadnji obisk: 9. 8. 2012]
- [22] Androidhive, Android Tab Layout Tutorial, dostopno na:  
<http://www.androidhive.info/2011/08/android-tab-layout-tutorial/>  
[zadnji obisk: 31. 8. 2012]
- [23] Praprotnik U. Tekaški trener. Ljubljana: Palestra, 2007
- [24] Dervišević E., Vidmar J. Vodič športne prehrane. Ljubljana: Fakulteta za šport, 2009
- [25] Middleton H. Hrana in kalorije. Ljubljana: Prešernova družba, 1998
- [26] Meier R. Professional Android 2 application development. Indianapolis: Wiley, 2010
- [27] Darcey L., Conder S. Sams teach yourself Android application development in 24 hours. Indianapolis: Sams, 2010

- [28] Eclipse (software), dostopno na:  
[http://en.wikipedia.org/wiki/Eclipse\\_\(software\)](http://en.wikipedia.org/wiki/Eclipse_(software))  
[zadnji obisk: 6. 9. 2012]
- [29] Cepec Ž. Razvoj aplikacije Kinoteka na mobilni platformi Android, Diplomsko delo, Maribor 2011
- [30] Samsung Galaxy S II., dostopno na:  
<http://www.samsung.com/global/microsite/galaxys2/html/index.html>  
[zadnji obisk: 6. 9. 2012]

## **9 PRILOGE**

### 9.1 Kratek življenjepis

Rojen: 20. 10. 1988

Šolanje: 1995 – 2003 Osnovna šola Tončke Čeč, Trbovlje

2003 – 2007 Splošna gimnazija Litija

2007 – Fakulteta za elektrotehniko, računalništvo in informatiko  
Univerzitetni študijski program Računalništvo in informacijske  
tehnologije





Univerza v Mariboru

Fakulteta za elektrotehniko,  
računalništvo in informacijo

### IZJAVA O USTREZNOSTI DIPLOMSKEGA DELA

Spodaj podpisani/-a red. prof. dr. MILAN ZORMAN izjavljam, da je  
(ime in priimek mentorja/-ice)

študent ALEN SOTLAR izdelal diplomsko  
(ime in priimek študenta/-ice)

delo z naslovom: MOBILNI PRIROČNIK O VADBI  
IN PREHRANI ZA OPERACIJSKI  
SISTEM ANDROID  
(naslov diplomskega dela)

v skladu z odobreno temo diplomskega dela, Navodili za pisanje diplomskih del na dodiplomskih študijskih programih UM FERi in mojimi navodili.

Kraj in datum: 3. september 2012

Podpis mentorja:



Univerza v Mariboru

Fakulteta za elektrotehniko,

računalništvo in informatiko

## IZJAVA O AVTORSTVU

Spodaj podpisani/-a ALEN SOTLAR  
z vpisno številko E1000047  
sem avtor/-ica diplomskega dela z naslovom:  
MOBILNI PRIROČNIK O VADBI IN  
PREHRANI ZA OPERACIJSKI SISTEM ANDROID  
(naslov diplomskega dela)

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)

RED. PROF. DR. MILAN ZORMAN

in s mentorstvom (naziv, ime in priimek)

ASIS. BOJAN BUTOLEN univ. dipl. inž. rač. in inf.

- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela.
- soglašam z javno objavo elektronske oblike diplomskega dela v DKUM.

V Mariboru, dne 3. september 2012

Podpis avtorja/-ice:



Univerza v Mariboru

Fakulteta za elektrotehniko,  
računalništvo in informatiko

**IZJAVA O ISTOVETNOSTI TISKANE IN ELEKTRONSKE VERZIJE ZAKLJUČNEGA  
DELA IN OBJAVI OSEBNIH PODATKOV DIPLOMANTOV**

Ime in priimek avtorja-ice: ALEN SOTLAR  
Vpisna številka: E1000047  
Študijski program: RAČUNALNIŠTVO IN INFORMACIJSKE TEHNOLOGIJE  
Naslov zaključnega dela: MOBILNI PRIROČNIK O VADBI IN  
PREHRANI ZA OPERACIJSKI SISTEM ANDROID  
Mentor: RED.PROF. DR. MILAN ŽORMAN  
Somentor: ASIS. BOJAN BUTOLEN univ. dipl. inž. rač. in inf.

Podpisani-a ALEN SOTLAR izjavljam, da sem za potrebe arhiviranja oddal elektronsko verzijo zaključnega dela v Digitalno knjižnico Univerze v Mariboru. Zaključno delo sem izdelal-a sam-a ob pomoči mentorja. V skladu s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah dovoljujem, da se zgoraj navedeno zaključno delo objavi na portalu Digitalne knjižnice Univerze v Mariboru.

Tiskana verzija zaključnega dela je istovetna z elektronsko verzijo elektronski verziji, ki sem jo oddal za objavo v Digitalno knjižnico Univerze v Mariboru.

Podpisani izjavljam, da dovoljujem objavo osebnih podatkov, vezanih na zaključek študija (ime, priimek, leto in kraj rojstva, datum zaključka študija, naslov zaključnega dela), na spletnih straneh in v publikacijah UM.

Datum in kraj: 3.9.2012, Maribor Podpis avtorja-ice: 

Podpis mentorja: \_\_\_\_\_  
(samo v primeru, če delo ne sme biti javno dostopno)

Podpis odgovorne osebe naročnika in žig: \_\_\_\_\_  
(samo v primeru, če delo ne sme biti javno dostopno)