

Gimnazija Celje – Center

MOBILNA APLIKACIJA EMATURA

RAZISKOVALNA NALOGA

Avtor:

Uroš Hercog, 4. b

Mentor:

Otmar Uranjek, prof.

Mestna občina Celje, Mladi za Celje

Celje, 2014

ZAHVALA

Zahvaljujem se mentorju prof. Otmarju Uranjeku za pomoč in nasvete pri izdelavi projektne naloge.

Zahvaljujem se Dejanu Cenclju za vse programerske nasvete.

KAZALO VSEBIN

1. Uvod.....	7
2. Cilji naloge.....	9
3. Metode dela.....	9
4. Jedro – teoretični del.....	10
4.1 Pametne naprave.....	10
4.2 Android.....	10
4.2.1 Operacijski sistem android.....	10
4.2.2 Android SDK.....	10
4.2.3 Aplikacija.....	10
4.2.4 Android emulator.....	11
4.3 Razvojno okolje.....	12
4.3.1 DBDesigner4.....	12
4.3.2 phpMyAdmin.....	13
4.3.3 SQLite Database Browser Portable.....	13
4.3.4 Eclipse.....	14
4.3.5 Microsoft Expression Web 4.....	14
4.3.6 Cyberduck.....	14
4.4 Osnovni opis funkcionalnosti mobilne in spletne aplikacije.....	15
4.4.1 Tehnični opis mobilne aplikacije.....	15
4.4.2 Specifikacije spletne aplikacije.....	16
5. Jedro – praktični del.....	17
5.1 Načrtovanje aplikacij.....	17
5.2 Tabele v podatkovnih bazah.....	17
5.2.1 Razlaga povezanosti tabel.....	19
5.3 Spletna aplikacija.....	20
5.3.1 Uporabniška navigacija po spletni aplikaciji.....	21
5.4 Mobilna aplikacija.....	25
6. Zaključek.....	30
7. Literatura.....	31
8. Priloge.....	32
8.1 Seznam uporabljenih kratic in pojmov.....	32
8.2 Uporabniški vmesnik nove aplikacije.....	33

KAZALO SLIK

Slika 1: Shema osnovnega koncepta naloge	7
Slika 2: Predogled Android OS v Windowsih 7	11
Slika 3: Načrtovanje podatkovne baze v DBDesigner 4	12
Slika 4: phpMyAdmin uporabniški vmesnik	13
Slika 5: SQLite Database Browser Portable	13
Slika 6: Eclipse	14
Slika 7: Hierarhija predmeta	19
Slika 8: Struktura spletnega dela projektne naloge	20
Slika 9: Dejanska koda v datoteki <i>inc.config.php</i>	21
Slika 10: Definiranje konstant v <i>inc.constants.php</i>	22
Slika 11: Definiranje spremenljivk za <i>Smarty</i> znotraj metode <i>smartyVars()</i>	22
Slika 12: Funkcija <i>isLoggedIn()</i> preveri, če je uporabnik prijavljen	22
Slika 13: Funkcija <i>render()</i> skrbi za sproten predogled izraza	23
Slika 14: Primer uporabe <i>Smartyja</i> za prikazovanje obvestil uporabniku	24
Slika 15: Seznam paketov, razredov in XML-datotek mobilne aplikacije	25
Slika 16: Metoda, ki spremeni trenutni zaslon	26
Slika 17: Metoda <i>onCreateView()</i> znotraj fragmentov	26
Slika 18: Prenajanje identifikatorja predmeta	27
Slika 19: Metoda <i>onReceive()</i> razreda <i>BroadcastReceiver</i>	28
Slika 20: Vmesnik <i>EmaturaListener</i>	28
Slika 21: Začetni pojavni zaslon	33
Slika 22: Prijavni zaslon	33
Slika 23: Zaslon brskanja med predmeti	34
Slika 24: Zaslon brskanja po poglavjih	34
Slika 25: Zaslon brskanja po podpoglavjih	35
Slika 26: Zaslon brskanja po podpodpoglavjih	35
Slika 27: Prikaz vsebine	36
Slika 28: Seznam priljubljenih vsebin	36
Slika 29: Zaslon iskanja vsebin	37

POVZETEK

Cilj moje raziskovalne naloge je razviti mobilno androidno aplikacijo in spletno aplikacijo, ki uporabnikom omogočata enostaven dostop do maturitetnih vsebin. Uvod je namenjen predstavitvi tehnologij in orodij, ki sem jih uporabil pri razvoju sistema. Pri opisu uporabljenih tehnologij sem se osredotočil predvsem na operacijski sistem Android (v nadaljevanju Android OS) in njegove komponente.

V nadaljevanju teoretičnega dela raziskovalne naloge je opisano razvojno okolje in želene funkcionalnosti novo razvitega sistema.

Jedro raziskovalne naloge vsebuje opis praktičnega dela, v katerem je predstavljena mobilna aplikacija eMatura in njena podporna spletna aplikacija. Opisal sem tudi izvirne datoteke aplikacij. Za lažje razumevanje sem navedel tudi posamezne dele programske kode in primere načrtovanja podatkovnih baz, uporabljenih znotraj sistema.

Rezultat praktičnega dela raziskovalne naloge sta mobilna aplikacija eMatura in njena podporna spletna aplikacije, nameščena na strežniku. Aplikacija se lahko uporablja na vseh mobilnih napravah z operacijskim sistemom Android različice 2.1 ali novejše.

Ključne besede: eMatura, Android OS, androidna aplikacija, pametne naprave, Java

ABSTRACT

The objective of this project was to develop a mobile Android application and an online application to enable users an easier access to the matura content. The introduction contains the presentation of technologies and tools, which I used in the process of development. In the presentation of the technologies I focused mainly on the Android operating system (in continuation Android OS) and its components.

Development environment and desired functionalities of the newly developed system are presented in the theoretical part of the project.

The main body of the project includes a description of the practical part in which the eMatura mobile and online applications are presented alongside descriptions of the source files of the mentioned applications supported by actual code snippets for easier understanding.

The result of the practical part of the project is the mobile application eMatura and its support online application setup on a server. The mobile application can be installed on every mobile device with Android OS version 2.1 and upwards.

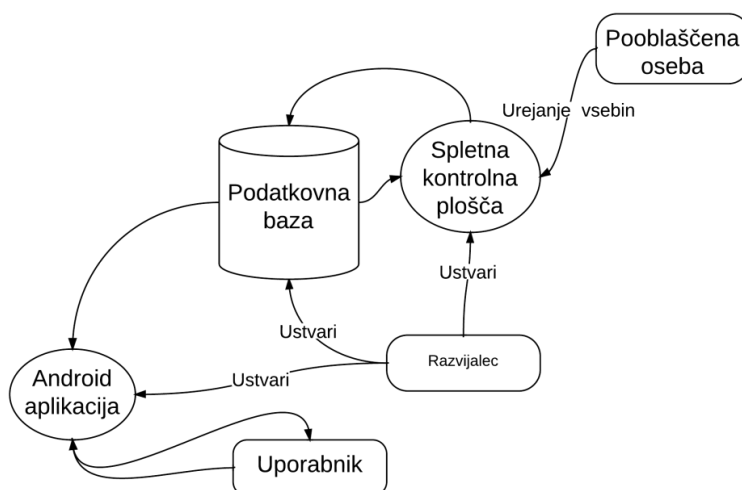
Keywords: eMatura, Android OS, Android application, smart devices, Java

1. UVOD

Pametni mobilni telefoni, tablični in osebni računalniki so postali sestavni del našega življenja. Večina ljudi ima v lasti vsaj eno izmed pametnih naprav, katere funkcionalnosti znamo bolj ali manj s pridom izkoristiti. Naprave so preproste za uporabo in cenovno zelo ugodne. Seveda si uporabe pametnih naprav ne moremo predstavljati brez različnih aplikacij, od predvajalnikov glasbe in filmov do aplikacij o financah in iger. Prav aplikacije dajejo uporabno vrednost t.i. pametnim mobilnim telefonom in tablicam.

Pametne naprave so lahko tudi idealni učni pripomoček, saj nudijo neomejeno število hitro dostopnih informacij, s čimer nam olajšajo učenje, omogočijo lažji dostop do natančnejših in kvalitetnejših vsebin. Ker pa me v letošnjem šolskem letu čaka opravljanje mature, sem s projektno nalogo želel napisati mobilno aplikacijo, ki bi dijakom omogočala lažji dostop do informacij, skrajšala čas iskanja vsebin in kvalitetno izboljšala tudi mojo pripravo na maturo. Nalogo sem zasnoval dolgoročno, z željo, da bi koristila tudi ostalim dijakom (v nadaljevanju uporabnikom).

Mobilna aplikacija eMatura temelji na prostem dostopu do vsebin katerega koli maturitetnega predmeta slovenske mature, povzetih po maturitetnih katalogih Republiškega izpitnega centra. Te vsebine vključujejo znakovno omejene vsebine, podprte s slikovnim gradivom in/ali matematičnim zapisom. Moja osnovna zamisel pri delu je bila ustvariti sistem, neodvisen od razvijalca. To sem dosegel z razvojem dveh med seboj povsem ločenih aplikacij; mobilne androidne aplikacije in spletne nadzorne aplikacije. Z androidno aplikacijo eMatura uporabnik dostopa do vsebin, ki jih lahko preko spletne aplikacije pooblaščen oseba poljubno dodaja, ureja in odstranjuje.



Slika 1: Shema osnovnega koncepta naloge

Najprej sem preveril trenutno ponudbo mobilnih aplikacij in literaturo o tej tematiki. Na osnovi analize trga in zbranih informacij sem nato zasnoval osnovni koncept naloge in napisal teoretično osnovo. V praktičnem delu pa sem ločeno razvil mobilno androidno aplikacijo in spletno nadzorno aplikacijo.

Pri razvojnem okolju sem se zaradi finančnih omejitev izvedbe projekta omejil na uporabo odprtokodnih oziroma brezplačnih razvojnih programov. Ker gre v tem smislu za precej obsežen projekt, sem se znotraj nabora maturitetnih predmetov omejil na testni predmet, in sicer fiziko. Ta predstavlja zgled za vse ostale maturitetne predmete.

V prihodnosti nameravam projekt nadaljevati tudi z vnosom vsebin za druge maturitetne predmete, če bodo izkušnje prvih testnih uporabnikov pozitivne.

2. CILJI NALOGE

Ker gre v tej projektni nalogi zgolj za razvoj mobilne in spletne aplikacije in ne za raziskavo, ne bom postavljajl hipotez, saj ne bi bile smiselne. Zato sem se raje osredotočil na več ciljev, ki sem jih želel uresničiti s to projektno nalogo.

- zasnovati androidno aplikacijo eMatura in spletno aplikacijo,
- napisati androidno aplikacijo za pregledovanje vsebin v programskem jeziku Java,
- napisati spletno kontrolno aplikacijo za urejanje in dodajanje vsebin,
- povezati androidno aplikacijo eMatura in spletno nadzorno aplikacijo v delujoč sistem,
- uporabiti in opisati brezplačna oziroma odprtokodna programska orodja, ki sem jih uporabil pri razvoju,
- opisati delovanje sistema,
- zagotoviti testno dodajanje vsebin za maturitetni predmet fizika (dodajanje izvede mentor),
- testirati sistem na testnih uporabnikih,
- testirati mobilno aplikacijo na čim večjem številu uporabnikov z različnimi pametnimi napravami z Android OS.

3. METODE DE LA

Za projektno nalogo sem uporabil več različnih metod dela. S preučevanjem strokovne literature o programskih jezikih sem si pomagal pri celotni nalogi. Na osnovi pridobljenih informacij in predhodnega znanja programiranja, pridobljenega v več kot štirih letih samoiniciativnega eksperimentiranja v različnih programskih jezikih, sem nato zasnoval in razvil dve aplikaciji, imenovani eMatura.

4. JEDRO – TEORETIČNI DEL

4.1 Pametne naprave

Pametne naprave so brezžične prenosne digitalne elektronske naprave, ki omogočajo povezljivost v internetno omrežje preko različnih brezžičnih (Wi-Fi) in mobilnih omrežij (2G/3G/4G). Naprave nam omogočajo glasovno komuniciranje, komuniciranje preko videa in predvsem podatkovni dostop do različnih multimedijskih izobraževalnih vsebin na daljavo. Uporabnost naprav ni omejena na izdelek, ki ga dobimo iz tovarne, a se lahko njena uporabnost poveča z dodatno programsko opremo oz. aplikacijo.

4.2 Android

4.2.1 Operacijski sistem android

Android je odprtokodni operacijski sistem, ki temelji na sistemu Linux. Operacijski sistem je namenjen prenosnim mobilnim napravam, ki imajo omejeno avtonomijo baterijskega napajanja. V to skupino sodijo mobilni telefoni, tablice in tudi vrsta ultra tankih prenosih računalnikov. Ta distribucija operacijskega sistema Linux je namenjena predvsem pametnim mobilnim telefonom in tabličnim računalnikom. Operacijski sistem Android je razvilo podjetje Android Inc. pod okriljem Googlovega poslovnega združenja OHA. Javnosti je bil predstavljen ob koncu leta 2007. Namen razvoja tega operacijskega sistema je bil uporabniku zagotoviti cenejše in zmogljivejše mobilne naprave. Android je odprtokoden in omogoča uporabnikom z ustreznim predznanjem razvijanje aplikacij, ki dodajajo napravam nove dimenzije uporabnosti in zmogljivosti.

4.2.2 Android SDK

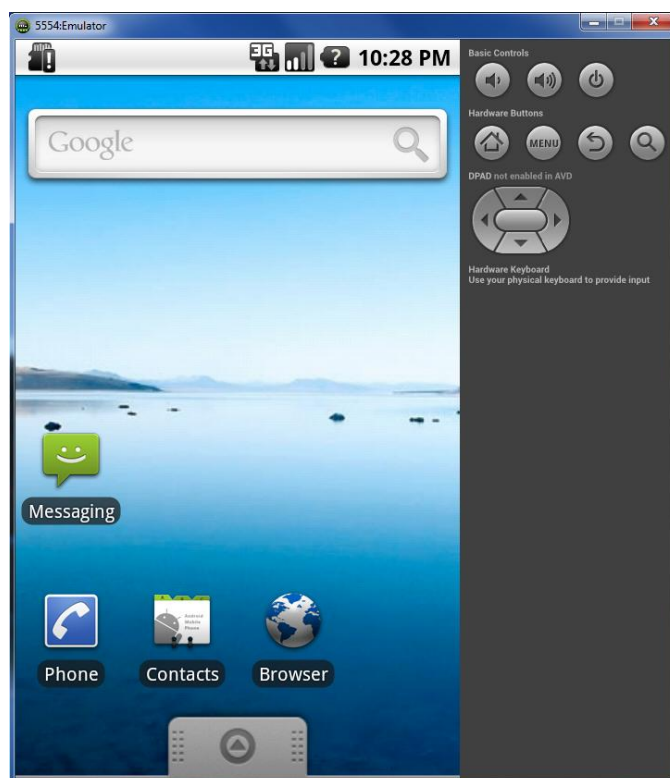
Android SDK je zbirka brezplačnih orodij, ki so namenjena razvoju aplikacij. V zbirko so poleg dokumentacije vključene razhroščevalnik, namenske programske knjižnice in program za emulacijo fizične naprave.

4.2.3 Aplikacija

Po definiciji iz literature je aplikacija *»zaključen in po navadi delujoči skupek kode, lahko tudi v različnih datotekah, spisanih v katerem koli od računalniških jezikov ali njihove kombinacije.«* (Matjaž Štrancar in Simon Klemen, stran 13, 2005)[1]. Android omogoča razvoj aplikacij v različnih programskih jezikih. Najbolj razširjen je programski jezik Java, ki sem ga tudi izbral za razvoj praktičnega dela seminarske naloge. Aplikacija ima končnico APK in vključuje tri komponente. Datoteko z vso prevedeno izvorno kodo v formatu DEX, multimedijske datoteke (slike, XML, ...), in dodatne knjižnice, ki omogočajo pravilno delovanje aplikacije npr. povezljivosti v internet. Osnovni namen aplikacij je povečanje uporabnosti naprave in ji dodati dodano vrednost.

4.2.4 Android emulator

Emulator je del Android SDK, ki podvaja funkcije Androida znotraj operacijskega sistema Windows. Njegova naloga je, da funkcionira enako, kot če bi bil nameščen na mobilni napravi. Uporablja se za zagon aplikacij operacijskega sistema Android v razvojno okolje in nam omogoča testiranje aplikacij brez fizične naprave oziroma v različnih verzijah Androida. Omogoča nam simulacijo različnih fizičnih naprav. Brez emulatorja bi bilo potrebno kupiti čisto vse naprave, ki podpirajo Android, kar pa je zelo nepraktično in drago.



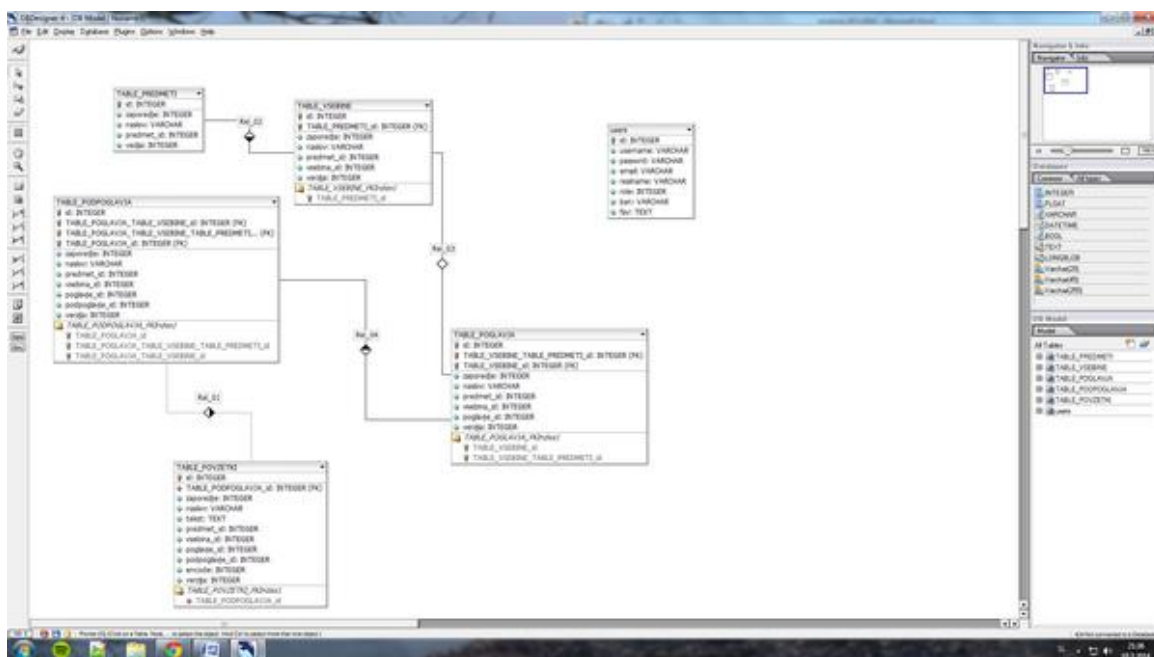
Slika 2: Predogled Android OS v Windowsih 7

4.3 Razvojno okolje

V tem poglavju se bom posvetil predstavitvi programov oziroma orodij, ki sem jih uporabil pri razvoju spletne in mobilne aplikacije. Za načrtovanje podatkovne baze sem uporabil program DBDesigner 4, same izgradnje spletne podatkovne baze pa sem se lotil neposredno v aplikaciji phpMyAdmin, nameščeni na strežniku Apache. Za razvoj mobilne aplikacije sem uporabil Eclipse IDE, z nameščenim ADT-vtičnikom in Android SDK, za pisanje in razhroščevanje aplikacije, SQLite Database Browser Portable, ki omogoča urejanje podatkovne baze znotraj fizičnih naprav, na katerih sem imel urejene administratorske pravice. Za testiranje aplikacije sem uporabil pametni mobilni telefon Samsung Galaxy Ace in HTC mini HD z operacijskim sistemom Android verzije 2.2.3 in tablico Samsung Galaxy Tab 2 10.1 in tablico Asus Eee pad Transformer Prime z nameščenim operacijskim sistemom Android verzije 4.1.2. Za razvoj spletne aplikacije sem poleg phpMyAdmin aplikacije uporabil tudi Microsoft Expression Web 4 za pisanje kode. Do strežnika sem si uredil dostop s programom Cyberduck, ki omogoča FTP-dostop do mape aplikacije na strežniku. Program Putty je služil kot TELNET klient za povezavo s šolskim spletnim strežnikom, na katerem se aplikacija nahaja.

4.3.1 DBDesigner4

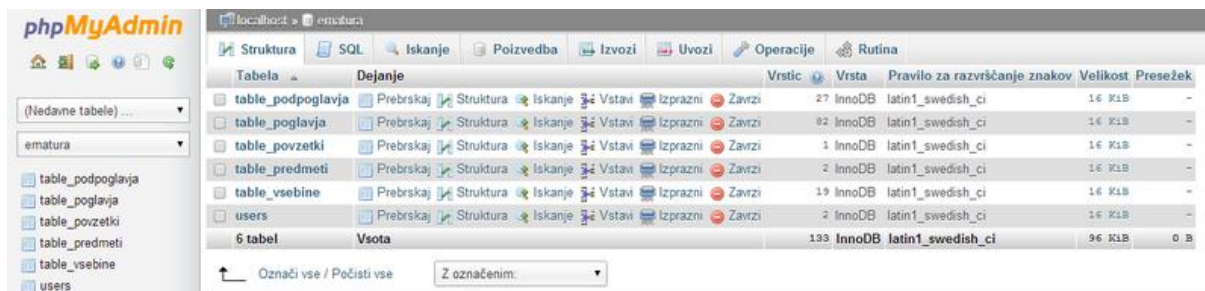
DBDesigner je odprtokodno orodje za modeliranje, ustvarjanje in vzdrževanje MySQL podatkovnih baz, ki je nastalo pod okriljem organizacije »fabFORCE.net«. Uporaba je brezplačna in ni omejena na preizkusno dobo, kot podobni programi. Orodje nam ponuja ogromno funkcionalnosti, od različnih tehnik modeliranja podatkovnih baz do njihovega grafičnega urejanja.



Slika 3: Načrtovanje podatkovne baze v DBDesigner 4

4.3.2 phpMyAdmin

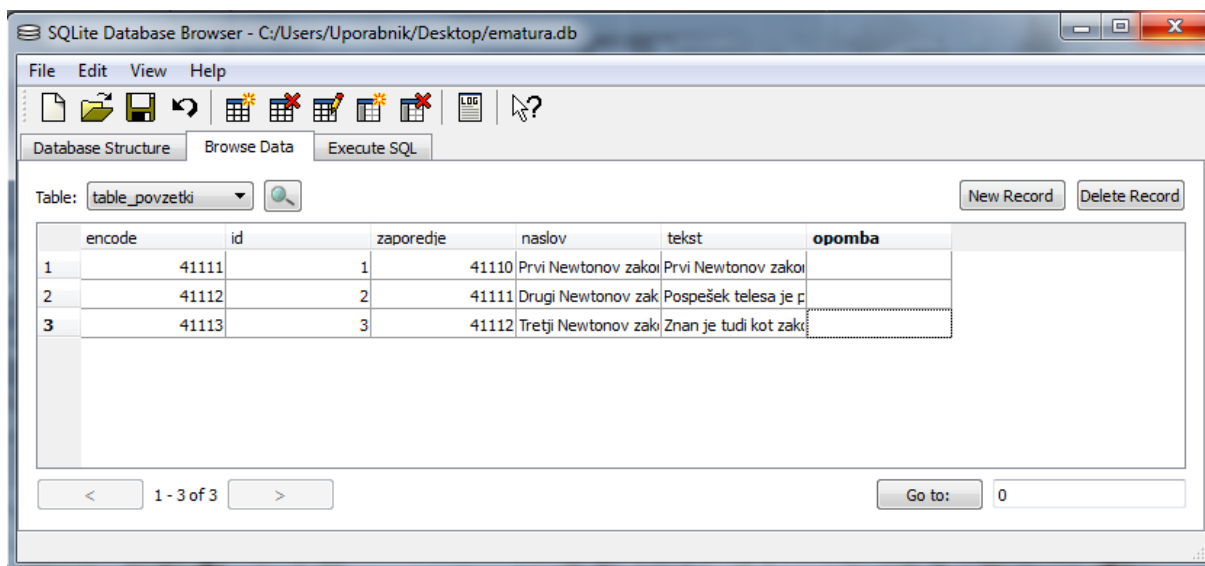
phpMyAdmin je brezplačno orodje, napisano v PHP namenjeno, administraciji MySQL podatkovnih baz preko spleta. Podpira organizacijo baz, tabel, relacij, uporabnikov in njihovih pravic. Vse operacije (dodajanje, pregledovanje, odstranjevanje baz, tabel in zapisov v njih ter uporabnikov) lahko izvedemo s pomočjo uporabniškega vmesnika v brskalniku ali neposredno z izvajanjem poizvedb na določeni podatkovni bazi.



Slika 4: phpMyAdmin uporabniški vmesnik

4.3.3 SQLite Database Browser Portable

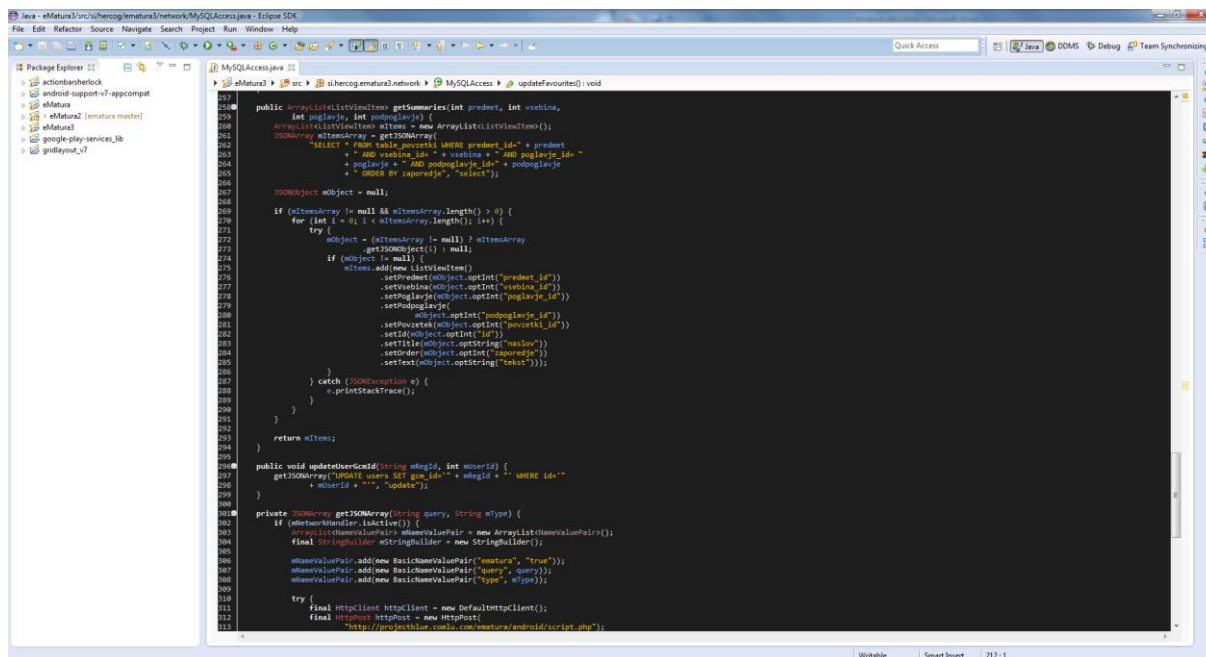
SQLite Database Browser Portable je brezplačen program, ki ga ni potrebno namestiti na računalnik. Namen tega programa je omogočiti tehnično neveščim uporabnikom upravljanje s podatkovnimi bazami s pomočjo različnih čarovnikov. Sam sem ta program uporabljal zgolj za pregledovanje lokalnih baz znotraj fizične naprave v procesu pisanja mobilne aplikacije.



Slika 5: SQLite Database Browser Portable

4.3.4 Eclipse

Eclipse je odprtokodni program z integriranim razvojnim okoljem. Uporablja se na različnih področjih, sam sem ga uporabil za razvoj naše mobilne aplikacije. Orodje vključuje sistem vtičnikov, s katerimi lahko poleg Jave omogočimo razvoj tudi v ostalih programskih jezikih npr. HTML5, Javascript in Python.



Slika 6: Eclipse

4.3.5 Microsoft Expression Web 4

Microsoft Expression Web 4 je brezplačen HTML-urejevalnik oziroma orodje za ustvarjanje spletnih strani, ki ga je razvilo podjetje Microsoft. Nudi nam možnost razvoja v XML, CSS, ASP.NET, PHP in JavaScript programskih jezikih.

4.3.6 Cyberduck

Cyberduck je odprtokodni FTP-klient napisan v Javi s Cocoa uporabniškim vmesnikom. Omogoča nam povezavo s strežnikom preko FTP-protokola (SSL-zaščita). Uporabniku omogoča enostaven prenos datotek na strežnik in s strežnika.

4.4 Osnovni opis funkcionalnosti mobilne in spletne aplikacije

V projektnem delu sem razvil zaprt sistem spletne in mobilne Android aplikacije. Mobilno aplikacijo je mogoče uporabljati na pametnih mobilnih napravah, spletno aplikacijo pa preko poljubnega spletnega brskalnika na računalniku. Aplikaciji sta preprosti za uporabo, prilagojeni povprečnemu uporabniku in upoštevata omejitve posameznih načinov dostopa. Uporabnika preko opozorilnih oken obveščata o dogajanju znotraj aplikacije. Registracija uporabnika je obvezna in je namenjena le dodajanju priljubljenih vsebin uporabnika.

Mobilna aplikacija ni popolnoma odvisna od internetne povezave. Priljubljene vsebine se uporabniku prenesejo in shranijo lokalno ter so mu na voljo kadarkoli. Internetna povezava je potrebna za prijavo uporabnika, sinhronizacijo priljubljenih vsebin, pregledovanje in iskanje vsebin. Podatki o trenutno prijavljenem uporabniku se shranijo. Mobilna aplikacija omogoča prikazovanje vsebin predmetov, poglavij, podpoglavij, podpodpoglavij. Vsebine so multimedijsko podprte z matematičnimi izrazi in slikovnim gradivom. Na voljo so tudi nastavitve intervala sinhronizacije in pomoč uporabnikom.

Spletna aplikacija omogoča registracijo uporabnika in njegovo prijavo. Na voljo so trije nivoji dostopa: dijak, profesor in administrator. Vsak nivo ima določene pravice urejanja vsebin. Nivo »dijak« ima najmanj pravic, »profesor« pravice, povezane s poglavji, podpoglavji, podpodpoglavji in vsebinami, nivo »administrator« pa popoln nadzor nad spletno aplikacijo. Urejevalnik vsebin ponuja možnost vnašanja matematičnih izrazov in dodajanje slikovnega gradiva. Pri razvoju te funkcionalnosti sem se precej časa ukvarjal z razvojem posebnega urejevalnika za vnos matematičnih izrazov, ki predstavlja enostaven vmesnik za ustvarjanje kvalitetnejših in interaktivnih vsebin.

4.4.1 Tehnični opis mobilne aplikacije

Glede na opisane zahteve mobilne aplikacije sem zasnoval osnovni koncept:

- Aplikacija je namenjena operacijskemu sistemu Android verzije 2.1 ali novejši, kar predstavlja več kot 95% vseh naprav z Android OS na tržišču.
- Primarno je optimizirana za delovanje na pametnih mobilnih telefonih, deluje tudi na tabličnih računalnikih.
- Aplikacija podatke shranjuje v lokalno bazo, kar omogoča delovanje brez internetne povezave.
- Aplikacija je brezplačna tako za prenos kot za uporabo. Ob namestitvi se je potrebno strinjati s pravicami za popoln dostop do interneta, dostopom do notranjega in zunanjega pomnilnika naprave (če je ta na voljo). Aplikacija bo ob prvem zagonu preverila, če je uporabnik prijavljen, sicer aplikacije ne bo mogoče uporabljati.

- Uporabniški vmesnik je privlačen in preprost za uporabo. Uporabnik hitro prepozna, čemu so namenjeni posamezni deli vmesnika. Ob zagonu aplikacije se pokaže pozdravni zaslon, nato osrednji meni. Ta je sestavljen iz gumba za brskanje po vsebinah, gumba za iskanje in gumba za dostop do priljubljenih vsebin. Na dnu zaslona je na voljo gumb za dostop do nastavitev, gumb za pomoč in gumb za sinhronizacijo na zahtevo.
- Tipka »meni« ni funkcionalna, saj je na nekaterih napravah ni več v fizični obliki. Klik na tipko »nazaj« uporabnika vrne na prejšnji zaslon.
- Podatki iz spletne baze se v aplikacijo prenesejo v obliki tekstovne datoteke JSON. Iz te oblike jih aplikacija nato samodejno pretvori v bolj organizirano in lažje berljivo obliko. Poglavlja, podpoglavja in podpodpoglavja se prikazujejo v seznamih. Prikaz vsebin je sestavljen iz naslova in teksta. Slednji je po določenem algoritmu pretvorjen v končno obliko (matematični izrazi znotraj $\langle \text{frm} \rangle \langle / \text{frm} \rangle$ značk se pretvori v LaTeX-obliko). Ker je znotraj teksta vsebin možno tudi slikovno gradivo, aplikacija za vsako datoteko (s končnico JPEG in maksimalne velikosti 1MB) preveri, če obstaja v pomnilniku telefona, sicer jo prenese s spleta in jo shrani. Vsako vsebino je možno dodati med priljubljene s pritiskom na ustrezen gumb ob vsaki vsebini.
- Aplikacija osvežuje podatke s strežnika tako, da primerja časovni odtis dodajanja priljubljene vsebine. Če je na strežniku novejša vsebina, se ta avtomatično posodobi.

4.4.2 Specifikacije spletne aplikacije

- Aplikacija deluje v vseh modernih brskalnikih brez kakršnih koli omejitev.
- Aplikacijo omogoča registracijo uporabnika in prijavo v sistem. Prijavljen uporabnik ima glede na svojo vlogo različne pravice. Uporaba spletne aplikacije brez prijave ni mogoča. Uporabnik z vlogo »dijak« lahko prosto brska po vsebinah in jih lahko tudi sam dodaja med priljubljene. Uporabnik z vlogo »profesor« ima na voljo možnost dodajanja, urejanja in odstranjevanja poglavij, podpoglavij, podpodpoglavij in vsebin, uporabnik z vlogo »administrator« pa poleg tega na voljo še dodajanje, urejanje in odstranjevanje uporabnikov ter pregled celotnega sistema.
- Urejevalnik vsebin vključuje poseben urejevalnik matematičnih izrazov, ki deluje s podporo brezplačne knjižnice MatJax. Z urejevalnikom lahko generiramo matematične izraze s sprotnim predogledom. Znotraj tega urejevalnika je na voljo še urejanje že obstoječih formul in grafični seznam najpogosteje uporabljenih simbolov ter funkcij.
- Uporabniški vmesnik spletne aplikacije sem izdelal s pomočjo brezplačne knjižnice Smarty.

5. Jedro – praktični del

5.1 Načrtovanje aplikacij

Po oblikovanju idej o funkcionalnosti mobilne in spletne aplikacije sem začel z udejanjanjem idej v prakso. Najprej sem s programom DBDesigner 4 naredil model podatkovne baze, ki jo uporablja spletna aplikacija. Bazo sem poimenoval »ematura« in obsega šest različnih tabel (table_predmeti, table_vsebine, table_poglavja, table_podpoglavja, table_povzetki, users).

5.2 Tabele v podatkovnih bazah

TABLE_PREDMETI

V tej tabeli so shranjeni zapisi za vse maturitetne predmete, ki se nahajajo na spletni strani Republiškega izpitnega centra – RIC-a. Tabela vsebuje naslednja polja:

- **ID** – identifikator za vnos v tabeli (INTEGER[11], primarni ključ, obvezen)
- **ZAPOREDJE** – številka zaporedja predmeta (INTEGER[11])
- **NASLOV** – naslov predmeta (VARCHAR[100])
- **PREDMET_ID** – zaporedna številka predmeta (INTEGER[11])
- **VERZIJA** – predstavlja trenutno verzijo predmeta (INTEGER[11])

TABLE_VSEBINE

V tej tabeli se shranjujejo vsa poglavja določenega predmeta, dodana preko spletne aplikacije.

- **ID** – identifikator za vnos v tabeli (INTEGER[11], primarni ključ, obvezen)
- **ZAPOREDJE** – številka zaporedja poglavja (INTEGER[11])
- **NASLOV** – naslov poglavja (VARCHAR[100])
- **PREDMET_ID** – zaporedna številka predmeta starša (INTEGER[11])
- **VSEBINA_ID** – zaporedna številka poglavja (INTEGER[11])
- **VERZIJA** – predstavlja trenutno verzijo predmeta (INTEGER[11])

TABLE_POGLAVJA

V tej tabeli se shranjujejo vsa podpoglavja določene vsebine, dodana preko spletne aplikacije.

- **ID** – identifikator za vnos v tabeli (INTEGER[11], primarni ključ, obvezen)
- **ZAPOREDJE** – številka zaporedja poglavja (INTEGER[11])
- **NASLOV** – naslov poglavja (VARCHAR[100])
- **PREDMET_ID** – zaporedna številka predmeta starša (INTEGER[11])
- **VSEBINA_ID** – zaporedna številka poglavja starša (INTEGER[11])
- **POGLAVJE_ID** – zaporedna številka poglavja (INTEGER[11])
- **VERZIJA** – predstavlja trenutno verzijo predmeta (INTEGER[11])

TABLE_PODPOGLAVJA

V tej tabeli se shranjujejo vsa podpodpoglavja določenega poglavja, dodana preko spletne aplikacije.

- **ID** – identifikator za vnos v tabeli (INTEGER[11], primarni ključ, obvezen)
- **ZAPOREDJE** – številka zaporedja poglavja (INTEGER[11])
- **NASLOV** – naslov poglavja (VARCHAR[100])
- **PREDMET_ID** – zaporedna številka predmeta starša (INTEGER[11])
- **VSEBINA_ID** – zaporedna številka poglavja starša (INTEGER[11])
- **POGLAVJE_ID** – zaporedna številka poglavja starša (INTEGER[11])
- **PODPOGLAVJE_ID** – zaporedna številka podpoglavja (INTEGER[11])
- **VERZIJA** – predstavlja trenutno verzijo predmeta (INTEGER[11])

TABLE_POVZETKI

V tabeli »table_povzetki« se shranjujejo vse vsebine določenega podpodpoglavja, dodane preko spletne aplikacije.

- **ID** – identifikator za vnos v tabeli (INTEGER[11], primarni ključ, obvezen)
- **ZAPOREDJE** – številka zaporedja poglavja (INTEGER[11])
- **NASLOV** – naslov poglavja (VARCHAR[100])
- **TEKST** – besedilo v vsebini (TEXT)
- **PREDMET_ID** – zaporedna številka predmeta starša (INTEGER[11])
- **VSEBINA_ID** – zaporedna številka poglavja starša (INTEGER[11])
- **POGLAVJE_ID** – zaporedna številka poglavja starša (INTEGER[11])
- **PODPOGLAVJE_ID** – zaporedna številka podpoglavja (INTEGER[11])
- **ENCODE** – krajši zapis staršev (INTEGER[11])
- **VERZIJA** – predstavlja trenutno verzijo predmeta (INTEGER[11])

USERS

V tej tabeli se shranjujejo vsa podpodpoglavja določenega poglavja, dodana preko spletne aplikacije.

- **ID** – identifikator za vnos v tabeli (INTEGER[11], primarni ključ, obvezen)
- **USERNAME** – uporabniško ime uporabnika (VARCHAR[50])
- **PASSWORD** – geslo uporabnika v MD5 enkripciji (VARCHAR[50])
- **EMAIL** – e-mail naslov uporabnika (VARCHAR[50])
- **REALNAME** – ime in priimek uporabnika (VARCHAR[50])
- **ROLE** – vloga uporabnika (INTEGER [11])
- **BAN** – predstavlja dostopnost računa (BINARY[1])
- **FAV** – predstavlja zapis priljubljenih vsebin uporabnika (TEXT)

5.2.1 RAZLAGA POVEZANOSTI TABEL

Vse tabele, razen tabele »users«, poleg stolpca za id, naslov in tekst vsebujejo še polja za posebne označevalce, ki omogočajo pravilno hierarhično strukturo vsebine. Ta polja (predmet_id, vsebina_id, poglavje_id, podpoglavje_id, povzetki_id) so namenjena sledenju relacij starš-otrok in s tem nam omogočajo veliko število vnosov v podatkovno bazo ter določajo pravilno hierarhijo.

Označevalec »starš« predstavlja vnos v tabeli, kateremu lahko pripišemo »otroke« oziroma podrejene vnose v drugi tabeli (glej sliko številka 7). Otrok je vnos v tabeli, ki je podrejena tabeli starša in je na nižjem nivoju v hierarhiji, na primer vsebina je otrok podpoglavja. Vsak starš in otrok imata svoj posebni označevalec in tudi označevalce vseh staršev nad njim. S tem načinom označevanja ne more priti do težav pri ugotavljanju relacij otrok-starš, poenostavi pa se tudi njihovo prikazovanje, dodajanje in urejanje znotraj spletne aplikacije.

Vsebine imajo še en dodaten označevalec, encode, ki je pa zgolj strnjena oblika staršev v obliki števila, katerega številke predstavljajo identifikatorje staršev. Uporablja se pri dodajanju vsebin med priljubljene. Problem takšnega zapisa je nezmožnost ugotovitve posameznih staršev, saj števila ni mogoče razpisati na posamezne identifikatorje staršev, ker ne vemo, koliko števk predstavlja identifikator posameznega nadrejenega vnosa.

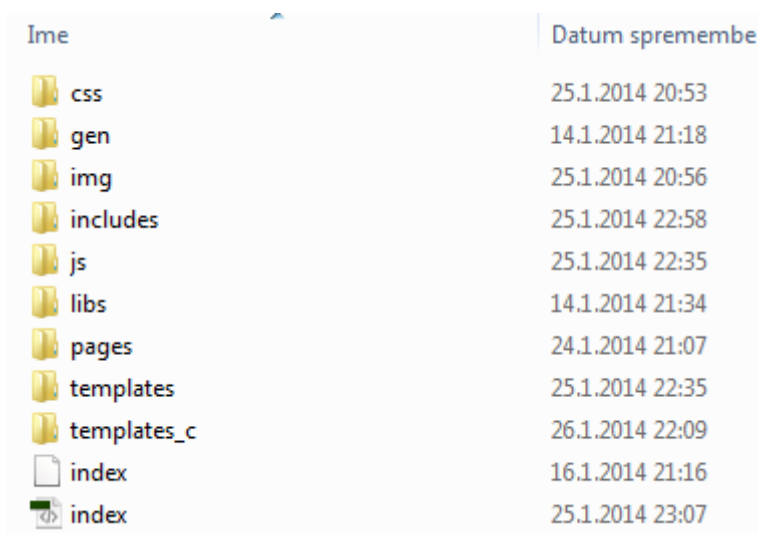
The screenshot displays the 'Matura' application interface, showing a hierarchical structure of subjects and topics. The interface is divided into four panes, each with a header containing 'BRSKAJ', 'PRILJUBIJE...', and 'IŠČI'. The first pane shows 'Anglščina' and 'Fizika'. The second pane lists physics topics, with '1. Premo gibanje' and '4. Premo in krivo gibanje' highlighted in red. The third pane lists kinematics topics, with '2. Trenutna in povprečna hitrost' highlighted in red. The fourth pane shows a detailed view of 'Povprečna hitrost' and 'Trenutna hitrost' with mathematical formulas and explanatory text.

Slika 7: Hierarhija predmeta

5.3 SPLETNA APLIKACIJA

Po končanem načrtovanju podatkovnih baz sem se lotil najprej spletnega dela projekta. Pred začetkom pisanja programske kode sem za lažjo organizacijo ustvaril mape, v katere sem nato sistematično shranjeval vse ustvarjene datoteke. Slika številka 8 prikazuje to strukturo.

Opis aplikacije in njeno delovanje bom združil v to poglavje, saj je enostavneje in pregledneje, če je razlaga delovanja prikazana ob slikah končane aplikacije.



Ime	Datum spremembe
css	25.1.2014 20:53
gen	14.1.2014 21:18
img	25.1.2014 20:56
includes	25.1.2014 22:58
js	25.1.2014 22:35
libs	14.1.2014 21:34
pages	24.1.2014 21:07
templates	25.1.2014 22:35
templates_c	26.1.2014 22:09
index	16.1.2014 21:16
index	25.1.2014 23:07

Slika 8: Struktura spletnega dela projektne naloge

MAPA CSS – vsebuje vse CSS-podloge, ki določajo izgled spletne strani.

MAPA GEN – v to mapo se nalagajo vse slikovne datoteke, ki so uporabljene znotraj vsebin.

MAPA IMG – vsebuje vse slikovne datoteke, ki se uporabljajo v zvezi s CSS-predlogami.

MAPA INCLUDES – v tej mapi so vse datoteke, ki predstavljajo temelj za delovanje spletne aplikacije.

MAPA JS – vsebuje vse javascript datoteke povezane s CSS-predlogami.

MAPA LIBS – vsebuje uporabljene brezplačne spletne knjižnice.

MAPA PAGES – vsebuje vse datoteke za posamezne podstrani.

MAPA TEMPLATES – vsebuje vse datoteke s končnico TPL (potrebne s strani Smarty knjižnice) za prikaz podstrani.

MAPA TEMPLATES_C – vsebuje vse datoteke predlog podstrani generirane s strani Smarty knjižnice.

DATOTEKA INDEX.PHP – glavna stran.

5.3.1 UPORABNIŠKA NAVIGACIJA PO SPLETNI APLIKACIJI

Ob obisku spletnega naslova (<http://ematura.gcc.si>) se nam pokaže vstopna stran z informacijami o aplikaciji. Po prijavi v sistem opazimo v meniju element »predmeti«, preko katerega lahko brskamo po vsebinah, in element »priljubljene«, kjer najdemo vsebine, ki smo jih označili kot priljubljene. Če ima naš račun vlogo »profesor« ali »administrator«, v meniju najdemo še element »admin« s podmenijem, preko katerega lahko urejamo poglavja, podpoglavja, podpodpoglavja in vsebine.

DATOTEKA INDEX.PHP

Ta datoteka predstavlja osrednjo stran in skrbi za navigacijo po spletni strani. V tej datoteki so tudi definirane vse datoteke iz mape »includes«, da se lahko nato njihove funkcije uporabljajo v vseh ostalih podstraneh brez potrebe njihovega ponovnega vključevanja.

MAPA INCLUDES

V tej mapi se nahajajo PHP-datoteke, ki služijo kot temelj za izgradnjo celotne spletne aplikacije. To so datoteke za konfiguracijo, datoteke s konstantami in datoteke s funkcijami, uporabljenimi skozi celoten spletni del projekta.

- **inc.config.php** - tu so nastavitve, od naslova aplikacije do osnovne poti na strežniku. Celotna konfiguracija se shrani v polje »\$config[]«. Takšna organizacija konfiguracije nam omogoča, da lahko hitro spreminjamo določene vrednosti in se to pozna v celotni aplikaciji.

```
$config = array();

$config['website_title']           = 'eMatura';
$config['website_slogan']         = 'Raziskovalna naloga';
$config['website_db_type']        = 'mysql';
$config['mysql']                  = array(
    'host' =>
    'user' =>
    'pass' =>
    'db' => '

```

Slika 9: Dejanska koda v datoteki *inc.config.php*

- ***inc.constants.php*** - v tej datoteki so splošno uporabljene konstante, kot so imena tabel in verzija aplikacije.

```
define('TABLE_USERS', 'users');
define('TABLE_SUBJECT', 'table_predmeti');
define('TABLE_CONTENT', 'table_vsebine');
```

Slika 10: Definiranje konstant v *inc.constants.php*

- ***inc.functions.php*** – tu so splošno uporabljene funkcije, kot je pošiljanje pošte. V tej datoteki se nahaja tudi metoda *smartyVars()*, ki knjižnici *Smarty* določi vrednosti spremenljivk, ki se nato uporabljajo skozi celotno predlogo.

```
$smarty->assign('website_title', $config['website_title']);
$smarty->assign('website_slogan', $config['website_slogan']);
```

Slika 11: Definiranje spremenljivk za *Smarty* znotraj metode *smartyVars()*

- ***inc.functions.db.php*** – koda znotraj te datoteke poskrbi za vzpostavitev povezave s spletno podatkovno bazo *MySQL*. Uporabljen je standard »*MySQLi*« in zagotavlja visok nivo varnosti.
- ***inc.functions.account.php*** – tu so funkcije, namenjene uporabniškemu računom, kot je spreminjanje gesla.
- ***inc.functions.security.php*** – tu so funkcije, povezane z zagotavljanjem varnosti sistema, kot je preverjanje prijavljenosti uporabnika, funkcija prijave v sistem in odjave iz njega, preverjanje nivojev dostopa in zaklenjenosti uporabniškega računa.

```
function isLoggedIn(){
    if(isset($_SESSION['eid']) && ((int)$_SESSION['eid'] > -1)){
        return true;
    }else{
        return false;
    }
}
```

Slika 12: Funkcija *isLoggedIn()* preveri, če je uporabnik prijavljen

- ***inc.functions.validate.php*** – ta datoteka vsebuje funkcije, namenjene preverjanju veljavnosti uporabniških računov, gesel in elektronskih naslovov.

MAPA JS

- *ematura.js* – v tej datoteki so splošne JavaScript funkcije, uporabljene skozi vse datoteke predlog spletne strani.
- *editor.js* – predstavlja osrednjo datoteko za delovanje urejevalnika matematičnih izrazov. Funkcije znotraj te datoteke skrbijo za njihovo sprotno generiranje glede na vnos v polje urejevalnika, kot tudi za brisanje, vstavljanje in urejanje izrazov.

```
function render(what,where){
    where.innerHTML = what;
    MathJax.Hub.Queue(["Typeset",MathJax.Hub],where);
}
```

Slika 13: Funkcija *render()* skrbi za sproten predogled izraza

MAPA PAGES

V tej mapi so datoteke podstrani spletne strani. To so datoteke, ki gradijo dejansko aplikacijo na temeljih datotek iz mape »includes«. Mapa vsebuje še dve podmapi, in sicer »account« in »admin«, v katerih so podstrani specifične za uporabniške račune in administracijo spletne strani. Vsaka od teh podstrani vsebuje funkcije, specifične zanjo.

Na primer datoteka »inc.summary.php« ima funkcijo »getSummary()«, ki skrbi za pridobivanje teksta in naslova vsebin iz podatkovne baze in jih vrača v polju \$array[]. Druga pomembna funkcija je preverjanje, ali je določena vsebina že dodana med priljubljene ali ne. Na podlagi poizvedbe na podatkovni bazi vrne logično vrednost »true« ali »false«. Funkciji »addFavourite()« in »removeFavourite()« skrbita za dodajanje in odstranjevanje priljubljenih vsebin.

MAPA TEMPLATES

Ta mapa je obvezna s strani *Smartyja*. V njej so datoteke predlog podstrani in imajo končnico TPL. Ker je *Smarty* neke vrste skriptni jezik, nam ponuja ogromno funkcionalnosti za delo s podatki in njihovim prikazovanjem v predlogah podstrani.

```
{if count($messages) > 0}
  <div class="message-wrapper">
    {foreach item=msg from=$messages}
      <div class="message-content">
        <div class="{ $msg.type}">
          
          <h3>{$msg.title}</h3>
          <p>{$msg.text}</p>
        </div>
      </div>
    {/foreach}
  </div>
```

Slika 14: Primer uporabe *Smartyja* za prikazovanje obvestil uporabniku

5.4 MOBILNA APLIKACIJA

Po končanem pisanju spletnega dela sem pričel s pisanjem mobilne aplikacije. Glede na zahteve, ki sem jih zasnoval, sem ugotovil, da bom potreboval kar nekaj različnih glavnih in pomožnih razredov (ter pripadajočih XML-datotek, ki predstavljajo uporabniške vmesnike). Mobilna aplikacija je napisana v programskem jeziku Java in je objektno orientiran jezik, kar pomeni, da je programska koda organizirana v strukture, imenovane razredi. Razredi so razporejeni v strukture, imenovane paketi. Slika številka 15 prikazuje hierarhijo teh struktur.

```
> si.hercog.ematura2
  > MainActivity.java
si.hercog.ematura2.data
  > SummaryData.java
si.hercog.ematura2.fragment
  > FragmentContent.java
  > FragmentFavourite.java
  > FragmentHome.java
  > FragmentHomeDisabled.java
  > FragmentLogin.java
  > FragmentSplash.java
  > FragmentSummary.java
  > FragmentUser.java
si.hercog.ematura2.http
  > ContentManager.java
  > JSONParser.java
  > NetworkHandler.java
si.hercog.ematura2.listener
  > EmaturaListener.java
si.hercog.ematura2.sql
  > SqlHelper.java
  > SqlManager.java
si.hercog.ematura2.utils
  > BitmapManager.java
  > EmaturaUtils.java
```

Slika 15: Seznam paketov, razredov in XML-datotek mobilne aplikacije

PAKET SI.HERCOG.EMATURA2

Znotraj tega paketa je osrednji razred za delovanje mobilne aplikacije in vsebuje naslednje razrede.

- **MainActivity.java** (*activity_main.xml*) – to je glavna in dejansko edina aktivnost (*ang.activity*) aplikacije, ki skrbi za celotno delovanje aplikacije. Aktivnost je del programske kode aplikacije, ki skrbi za zagon grafičnega vmesnika. Razred »MainActivity« preko vmesnikov skrbi za od preklapljanja med zasloni aplikacije (*ang.fragments*) do preverjanja aktivnosti internetne povezave in dodajanje vsebin med priljubljene. Deluje kot nekakšen nadzorni center za celotno mobilno aplikacijo.

```
@Override
public void onFragmentManager(Fragment f, String name) {
    fTransaction = fManager.beginTransaction();
    fTransaction.replace(R.id.main_container, f);
    fTransaction.addToBackStack(name);
    fTransaction.commit();
}
```

Slika 16: Metoda, ki spremeni trenutni zaslon

PAKET SI.HERCOG.EMATURA2.DATA

Ta paket vsebuje zgolj en razred, »SummaryData«. Objekt tega razreda se uporablja za hranjenje vsebin, dokler se te ne shranijo v napravo ali prikažejo na zaslonu.

PAKET SI.HERCOG.EMATURA2.FRAGMENT

V tem paketu so vsi razredi, namenjeni posameznim zaslonom aplikacij. Vsak od teh razredov razširja razred »Fragment«, ki je na voljo s strani Android podporne knjižnice »android.support.v4.app« (podporna knjižnica je uporabljena zato, ker so bili fragmenti predstavljeni šele v android verziji 3.0, aplikacije eMatura pa mora delovati že na verziji 2.1 in to nam »android.support.v4.app« podporna knjižnica omogoča). V vsakem od teh razredov je glavna metoda »onCreateView()«, znotraj katere je definirana predloga uporabniškega vmesnika.

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    View v = inflater.inflate(R.layout.fragment_content_predmet,
        container, false);
}
```

Slika 17: Metoda onCreateView() znotraj fragmentov

Razredi poglavij, podpoglavij in podpodpoglavij morajo med seboj komunicirati in si podajati identifikatorje staršev, saj lahko le na tako uporabnik najde ustrezno vsebino. Podatke si med seboj podajajo preko objekta razreda »Bundle« znotraj paketa »android.os«. Ob kliku na element (to so poglavja...) v seznamu v uporabniškem vmesniku trenutnega objekta, objekt razreda nato razbere njegov identifikator in ga shrani v ustvarjeni objekt razreda »Bundle«. Nato se ustvari ustrezen objekt – otrok (npr. znotraj poglavja se ustvari objekt razreda podpoglavje), objekt z identifikatorjem se shrani kot argument objekta otroka in nato se pokliče povratna metoda, preko katere razred »MainActivity« zamenja zaslon.

```
Poglavje poglavje = new Poglavje();
Bundle bundle = new Bundle();
bundle.putInt("predmet", arg2 + 1);
poglavje.setArguments(bundle);

mCallback.onFragmentClickListener(poglavje, "poglavje");
```

Slika 18: Prenašanje identifikatorja predmeta

PAKET SI.HERCOG.EMATURA2.HTTP

V tem paketu so razredi, povezani za nadzor nad internetno povezavo in komuniciranjem s spletno podatkovno bazo. Vsi podatki, ki jih aplikacija dobi s spleta so v obliki *JSON*. Te podatke nato razred »JSONParser« pretvori v »SparseArray« (razred znotraj paketa »android.util«; izpopolnjena različica polja) za enostavnejšo rabo v aplikaciji.

- **ContentManager.java** - vsebuje metode, ki vračajo polja (»SparseArray«), v katerih so informacije, pridobljene iz spletne podatkovne baze.
- **JSONParser.java** – ta razred vsebuje metodo, ki skrbi za pridobivanje informacij v obliki *JSON* in njihovo pretvorbo v »SparseArray« polje glede na podano *SQL* poizvedo znotraj njenih parametrov.
- **NetworkHandler.java** - ta razred skrbi za nadzor nad internetno povezavo tako, da registrira »BroadcastReceiver« poslušalca (razred znotraj paketa »android.content«). Znotraj metode »onReceive()« razreda poslušalca se glede na trenutno stanje internetne povezave (za to skrbita metodi »isActive()«, ki vrne logično vrednost aktivnosti povezave, in metoda »getConnectivityStatus()«, ki vrne tip trenutno aktivne povezave), izvede povratni klic metode »onNetworkStateChangedListener()« znotraj razreda »MainActivity« implementiranega vmesnika »EmaturaListener«.

```

@Override
public void onReceive(Context context, Intent intent) {
    if (!isActive()) {
        mCallback.onNetworkStateChangeListener(false);
    } else if (isActive()) {
        mCallback.onNetworkStateChangeListener(true);
    }
}
}

```

Slika 19: Metoda *onReceive()* razreda *BroadcastReceiver*

PAKET SI.HERCOG.EMATURA2.SQL

Ta paket vsebuje vse razrede, potrebne za upravljanje z lokalno podatkovno bazo *MySQLi*.

- *SqlHelper.java* – ta razred razširja razred »*SQLiteOpenHelper*« (ta je znotraj paketa »*android.database.sqlite*«) in skrbi za ustvarjanje podatkovne baze (če ta ne obstaja) in za njeno posodobitev, če želimo dodati novo tabelo ali stolpec v že obstoječi tabeli.
- *SqlManager.java* – ta razred je glavni razred pri dostopanju do podatkovne baze. Vsebuje metode vse od pridobivanja priljubljenih do njihovega odstranjevanja.

PAKET SI.HERCOG.EMATURA2.LISTENER

V tem paketu je zgolj vmesnik »*EmaturaListener*«. Ta vmesnik vsebuje metode poslušalcev ki jih implementira razred »*MainActivity*«.

```

public interface EmaturaListener {
    public void onFragmentClickListener(Fragment f, String name);
    public void onFavouriteAddListener(String title, String content,
        String remark, int location);
    public void onBackClickListener();
}

```

Slika 20: Vmesnik *EmaturaListener*

PAKET SI.HERCOG.EMATURA.UTILS

V tem paketu so razredi, povezani s splošnimi metodami, uporabljenimi skozi celotno aplikacijo. Najpomembnejši razred je »*BitmapManager*«, ki skrbi za nadzor nad datotekami slikovnega gradiva. Ta vsakič, ko se s spleta v aplikacijo naloži izbrana vsebina, preveri, če je slikovno gradivo (v primeru, da ga prenesena vsebina vsebuje) že shranjeno na napravi, v nasprotnem primeru pa sproži prenos in shranitev datoteke na napravo. To zagotavlja, da se datoteke slikovnega gradiva prenesejo samo enkrat in uporabijo tudi naslednjič. S tem se zmanjša uporaba internetne povezave. Glede na aktivnost internetne povezave in parametre, ki jih izbere uporabnik, razred »*Search*« nato izvede iskanje vsebin po spletu ali med priljubljenimi.

DATOTEKA ANDROIDMANIFEST.XML

Datoteka »*androidmanifest.xml*« predstavlja konfiguracijsko datoteko vsake androidne aplikacije. V njej je definirana verzija aplikacije, vsa dovoljenja, ki jih aplikacija potrebuje (v tem primeru poln dostop do interneta, stanja internetne povezave, pisanja in branja na napravo), glavno aktivnost aplikacije (v tem primeru »*MainActivity*«), temo in ikono aplikacije ter minimalno verzijo operacijskega sistema android, na kateri bo aplikacija še delovala.

6. ZAKLJUČEK

V projektne delu sem v teoretičnem delu predstavil vsa razvojna orodja, ki so povezana z razvojem mobilne aplikacije eMatura in tudi njenega spletnega dela. V praktičnem delu naloge sem podrobno predstavil način razvoja in opisal delovanje sistema s primeri konkretne izvorne kode. Pri razvoju sem naletel na različne ovire, ki pa sem jih s pomočjo literature in spletnih dokumentacij zelo hitro premostil. V mobilni aplikaciji so implementirane vse opisane in načrtovane specifikacije. Aplikacija eMatura deluje brezhibno na testiranih napravah. Da bi zagotovil delovanje na večjem številu naprav, bi potreboval dostop do več različnih naprav.

Končna mobilna aplikacija in spletna aplikacija imata za uporabnika povsem enostaven in intuitiven uporabniški vmesnik. Prednosti mobilne aplikacije so to, da lahko uporabnik vse vsebine, ki jih je dodal pod priljubljene, pregleduje tudi brez aktivne internetne povezave, da je brezplačna in popolnoma prilagojena za slovenski trg.

Vsekakor imam namen sistem še razvijati. V načrtu imam že posodobitev predvsem spletnega dela aplikacije. To pomeni uporabo najnovejših tehnologij in knjižnic, s katerimi lahko zagotovim še boljšo in varnejšo uporabniško izkušnjo. V načrtu je tudi objava mobilne aplikacije na Googlovi spletni trgovini »*Google Play*« in predstavitev spletnega dela aplikacije na lasten strežnik pod lastno domeno.

S pisanjem programerskega dela projektne naloge sem se seznanil z veliko najrazličnejšimi razvojnimi orodji in tehnikami, za katere menim, da mi bodo v prihodnosti koristile. Cilj projektne naloge, predstavitev in razvoj sistema eMatura, je bil dosežen. Ker pa sem pri tem še zelo razširil svoje znanje o programiranju in delovanju računalniških sistemov nasploh, menim, da bom v prihodnje ustvarjal še boljše, kvalitetnejše in zanimivejše mobilne in/ali spletne aplikacije.

7. LITERATURA

- [1] PHP in MySQL na spletnem strežniku Apache, druga dopolnjena izdaja, Matjaž Štrancar in Simon Klemen, Založba Pasadena, 2005
- [2] Spoznajmo JavaScript: programiranje spletnih strani, Peer Hribar, Flamingo, 1998
- [3] Naučite se PHP v 24 urah, Matt Zandstra, Založba Pasadena, 2004
- [4] Android Developers
<http://developer.android.com/index.html>
- [5] Vogella Android Development Tutorial
<http://www.vogella.com/tutorials/Android/article.html>
- [6] Android (operacijski sistem)
[http://sl.wikipedia.org/wiki/Android_\(operacijski_sistem\)](http://sl.wikipedia.org/wiki/Android_(operacijski_sistem))
- [7] Android (operating system)
[http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
- [8] MathJax knjižnica
<http://www.mathjax.org/>
- [9] Zasnova mobilne aplikacije na zahtevo, Dejan Cencelj, Fakulteta za elektrotehniko, računalništvo in informatiko v Mariboru, 2009

8. PRILOGE

8.1 SEZNAM UPORABLJENIH KRATIC IN POJMOV

ADT	ang. <i>Android Development Tools</i> – zbirka razvojnih orodij, namenjenih uporabi znotraj operacijskega sistema Android
APK	ang. <i>Android Package</i> – arhivska datoteka, ki se uporablja v operacijskem sistemu android za nameščanje aplikacij
CSS	ang. <i>Cascading Style Sheet</i> – skriptni jezik, namenjen opisovanju in definiranju izgleda spletne strani
HMTL	ang. <i>HyperText Markup Language</i> – označevalni skriptni jezik, predstavlja osnovno večini spletnih strani
IDE	ang. <i>Intergrated Development Environment</i> – integrirano razvojno okolje znotraj katerega lahko programer z ustreznimi orodji ustvarja programe
JS	ang. <i>JavaScript</i> – skriptni jezik, namenjen ustvarjanju interaktivnih in odzivnih spletnih strani
JSON	ang. <i>JavaScript Object Notation</i> - format strukturiranih informacij, ki temeljijo na besedilu
MYSQL	ang. <i>My Structured Query Language</i> – sistem upravljanja s podatkovnimi bazami, nadgradnja SQL– jezika
MATHJAX	brezplačna odprtokodna JavaScript knjižnica, namenjena generiranju matematičnih izrazov v MathML ali LaTeX obliki
OHA	ang. <i>Open Handset Alliance</i> – združenje, namenjeno določanju standardov za mobilne naprave
OS	ang. <i>Operating System</i> – operacijski sistem
PHP	ang. <i>PHP: Hypertext Preprocessor</i> – strežniški skriptni jezik, največkrat v uporabi skupaj z HTML– jem
QEMU	ang. <i>QuickEmulator</i> – brezplačen in odprtokodni program za virtualizacijo strojne opreme
REGEX	ang. <i>Regular Expressions</i> – izrazi, ki računalniku povedo, s katerimi vzorci naj analizira dano besedilo
SDK	ang. <i>Software Development Kit</i> – zbirka razvojnih orodij
TELNET	omrežni protokol, ki omogoča povezavo računalnika s strežnikom in izvajanje ukazov na njem
Wi-Fi	ang. <i>Wireless Fidelity</i> – neformalno sinonim za brezžično računalniško omrežje
XML	ang. <i>eXtensible Markup Language</i> – označevalni jezik, ki omogoča zapis strukturiranih podatkov

8.2 UPORABNIŠKI VMESNIK NOVE APLIKACIJE



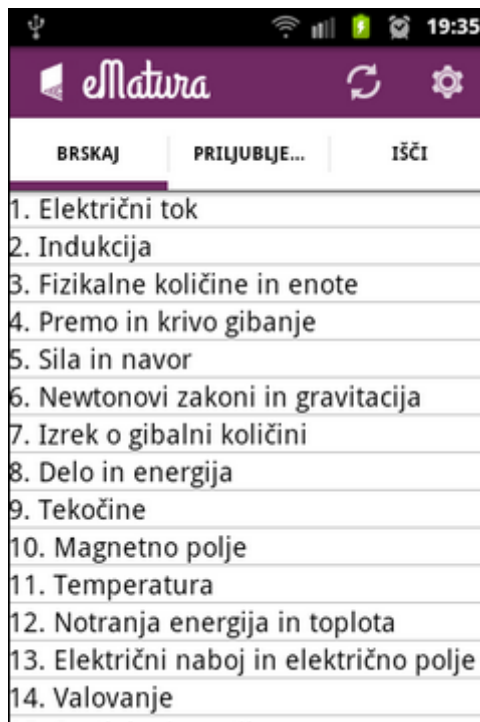
Slika 21: Začetni pojavni zaslon



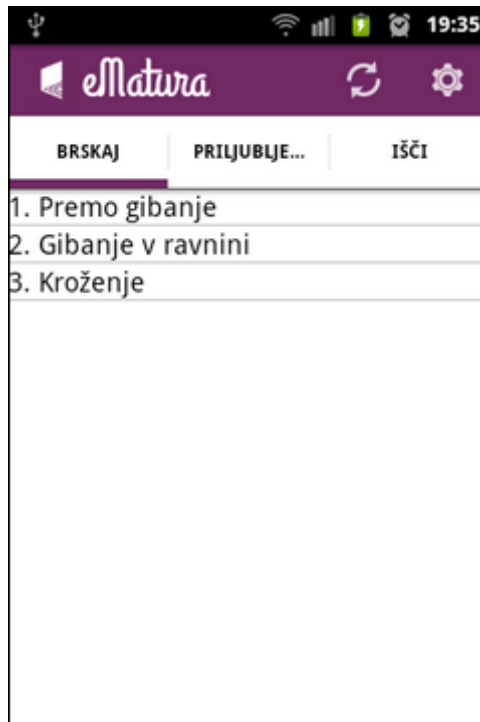
Slika 22: Prijavni zaslon



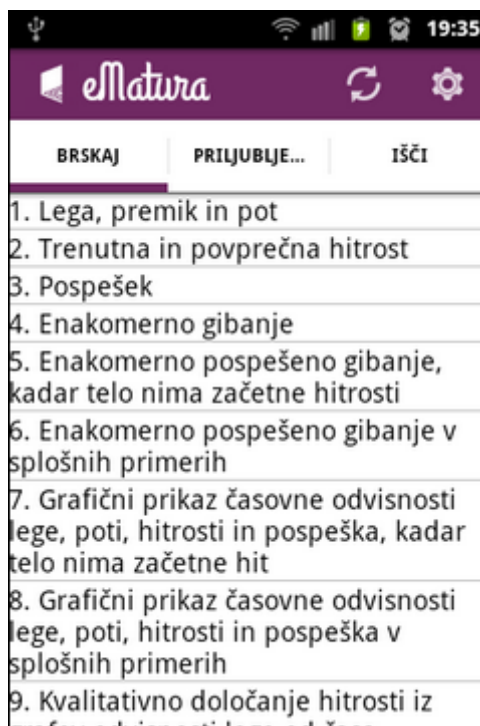
Slika 23: Zaslón brskanja med predmeti



Slika 24: Zaslón brskanja po poglavjih



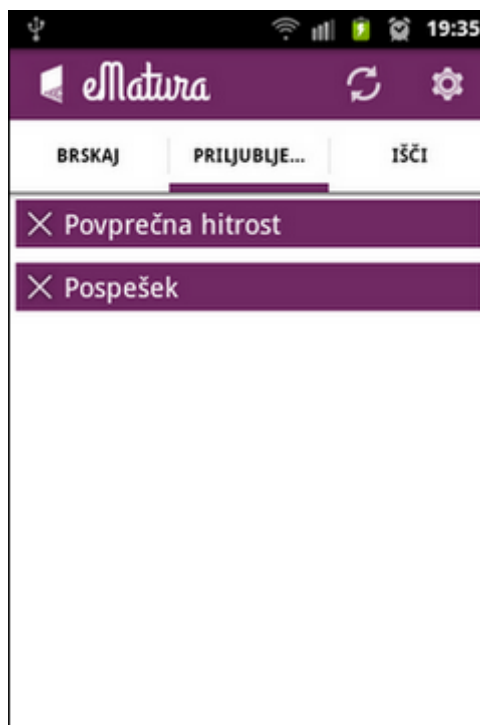
Slika 25: Zaslón brskanja po podpoglavjih



Slika 26: Zaslón brskanja po podpodglavjih



Slika 27: Prikaz vsebine



Slika 28: Seznam priljubljenih vsebin



Slika 29: Zaslón iskanja vsebin