

Šolski center Celje
Srednja šola za kemijo, elektrotehniko in računalništvo

WEB-KER ZA ANDROID

Raziskovalna naloga

AVTORJI:

Jan Dimnik

Žan Tadini

Tadej Kotar

Mentor:

Borut Slemenšek univ.dipl. inž.

Celje, 2016

Šolski center Celje
Srednja šola za kemijo, elektrotehniko in računalništvo

WEB-KER ZA ANDROID

Raziskovalna naloga

Avtorji:

Jan Dimnik

Žan Tadina

Tadej Kotar

Mentor:

Borut Slemenšek univ.dipl. inž.

Mestna občina Celje, Mladi za Celje

Celje, 2016

IZJAVA*

Mentor (-ica) , BORUT SLEMENŠEK, v skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi naslovom WEB-KER ZA ANDROID,

katere avtorji (-ice) so JAN DIMNIK, ŽAN TADINA, TADEJ KOTAR :

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo (-ičino) dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje

Celje, 14. 3. 2016

žig šole



Podpis mentorja(-ice)

Podpis odgovorne osebe

*

POJASNILO V skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja(-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja(-ice) fotografskega gradiva, katerega ni avtor(-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.

Kazalo vsebine

1. Uvod.....	3
1.1 Hipoteze raziskovalne naloge	3
1.2 Opis raziskovalnih nalog	3
2. Razvoj mobilne aplikacije	4
2.1 Koraki dela	4
3. Načrtovanje	4
3.1 Načrtovanje novega projekta ter začetne zaslonske slike.....	4
3.2 Načrtovanje zaslonskih slik ob različnih aktivnostih	4
4. Izdelava	5
4.1 Izdelava projekta ter začetne zaslonske slike	5
4.2 Prijava v šolski server.....	6
4.3 Urniki.....	7
4.4 Nadomeščanja.....	7
4.4.1 HTML koda strani	7
4.4.2 Programiranje	8
4.5ocene	9
4.6 Izostanki.....	9
5. Zaključek.....	10
5.1 Hipoteze.....	10
5.2 Težave pri delu	11
5.3 Kaj smo se novega naučili?	11
6. Viri	12

Kazalo slik

Slika 1 Postavitev gumbov na začetni zaslonski sliki	5
Slika 2 Koda za oblikovanje gumbov	5
Slika 3 Ikona za prijavo.....	6
Slika 4 Zgled prijave	6
Slika 5 Zgled urnikov	7
Slika 6 Primer HTML kode za nadomeščanja	8
Slika 7 Primer kode za pridobivanje podatkov za nadomeščanja	8
Slika 8 Klicanje spletne strani ter pridobitev podatkov	8
Slika 9 Zgled ocen.....	9
Slika 10 Zgled izostankov	9

Povzetek

V raziskovalni nalogi smo se odločili narediti mobilno aplikacijo za telefone v Android Studiu, ki uporablja programski jezik Java. Aplikacija bo poenostavila pregled podatkov objavljenih na šolski spletni strani. Program nudi možnost prijave v EviWeb SIS. Po prijavi lahko uporabnik pregleda le svoje ocene in izostanke. Brez prijave lahko vsak uporabnik pregleda ure nadomeščanj za tekoči in naslednji dan. Ker so javno objavljeni tudi urniki, je možen tudi ogled le-teh za vse oddelke.

Abstract

In this research paper we decided to make a mobile application in Android Studio, which uses Java programming language. The application will manage an easier access to information posted on our school website. The program offers the possibility to login into EviWeb SIS. After connection the user is able to check his own grades and absences. Without logging everyone is able to check out the substitute lessons for the present and the next day. While the schedules are public they can also be seen for every class.

Ključne besede

- Android studio – programsko okolje
- Java – programski jezik
- Layout – postavitev zaslonske slike
- Listview – polje, kjer je shranjenih več podatkov
- String – objekt v Javi, ki hrani znake

1. Uvod

Mobilni telefoni ter tablični računalniki so dandanes že nekaj vsakdanjega in predstavljajo sestavni del življenja vsakega izmed nas. Vsaka naprava ima naložene tudi določene aplikacije, ki so lahko namenjene opravljanju različnih nalog, od gledanja filmov, do prebiranja aktualnih novic. Imenujemo jih tudi »mali računalniki« in ta naziv tudi opravičujejo, kajti skorja vse vsebine, ki jih gledamo na računalniku, so dostopne tudi na naših malih napravah, njihove komponente pa so miniaturizirane komponente računalnika.

Vsakdo se je v življenju že znašel v situaciji, ko ga je preganjal čas in mu je zato šlo vse narobe. V takem položaju si vsekakor ne bi želel, da bi moral na hitro preveriti še kakšen podatek na internetu. Zaradi počasnega brskalnika ali slabo narejene spletne strani bi bilo to v takem trenutku zelo zamudno. Ta problem smo občasno občutili tudi mi dijaki, ko smo bili že pozni pri odpravljanju v šolo in je preverjanje nadomeščanj trajalo »celo večnost«. Da bi preprečili paniko ter omogočili hiter, enostaven, ter učinkovit dostop do nadomeščanj, urnikov, ocen in izostankov, smo se odločili, da naredimo mobilno aplikacijo v ta namen. Aplikacija je uporabniku zelo prijazna in enostavna za uporabo, kar omogoča preverjanje določenih podatkov že z nekaj kliki.

1.1 Hipoteze raziskovalne naloge

Postavili smo naslednje hipoteze raziskovalne naloge:

- Dostop do podatkov je z mobilno aplikacijo olajšan
- Uporabniki želijo imeti mobilno aplikacijo
- Mobilna aplikacija bo bolj v uporabi kot pa spletna verzija

1.2 Opis raziskovalnih nalog

Pri izdelavi raziskovalne naloge smo uporabili različna spletna gradiva za pripravo idej in reševanje programerskega dela naloge, izdelali smo pa tudi anketo, ki so jo reševali dijaki našega oddelka. Kot računalničarji so se nam zdeli najprimernejši, saj so potencialni uporabniki našega produkta, hkrati pa se spoznajo tudi na programiranje in so jim podobni izzivi blizu, kar je za nas in reševanje naloge zelo pozitivno v smislu pridobitve še kakšnega namiga za izboljšanje aplikacije.

2. Razvoj mobilne aplikacije

2.1 Koraki dela

Razvoj naše mobilne aplikacije je obsegal naslednje korake:

1. Načrtovanje:
 - Načrtovanje novega projekta ter začetne zaslonske slike
 - Načrtovanje zaslonskih slik ob različnih aktivnostih
2. Izdelava mobilne aplikacije v Android studiu
3. Testiranje aplikacije in njeno izpopolnjevanje

3. Načrtovanje

3.1 Načrtovanje novega projekta ter začetne zaslonske slike

Za izdelavo našega projekta smo imeli na izbiro dve programske okolji. To sta Xamarin in Android studio. Odločili smo se za Android studio, saj je programsko okolje brezplačno in ponuja več možnosti za razvijalce. Začetno zaslonsko sliko smo zastavili v istih barvah kot so značilne za našo šolsko spletno stran.

3.2 Načrtovanje zaslonskih slik ob različnih aktivnostih

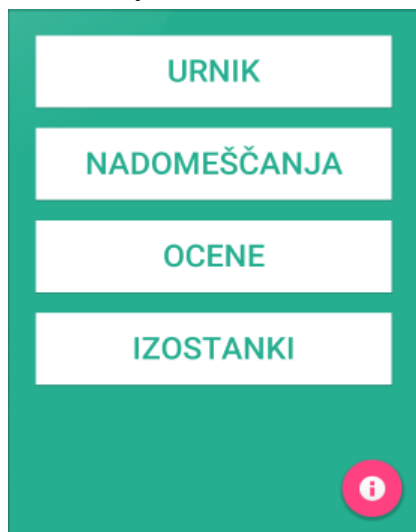
Zaslonske slike ob različnih aktivnostih smo načrtovali tako, da smo jih sprva narisali na list papirja, predebatirali o njihovi funkcionalnosti in jih zatem posebej oblikovali. Ker dobimo ogromno vhodnih podatkov, smo jih seveda morali omejiti (izbrati le tiste, ki so bili za naš prikaz potrebni).

4. Izdelava

4.1 Izdelava projekta ter začetne zaslonske slike

Na začetku, kot pri vsaki aplikaciji, se začne z oblikovanjem grafičnega vmesnika aplikacije. Android studio ima grafični vmesnik, ki omogoča lažje oblikovanje posameznih zaslonskih slik. Najnovejša verzija Android studia ima dve vrsti postavitev zaslonske slike (layout) in sicer »content« ter »activity«. Razlika med tema dvema je, da se content uporablja za vstavljanje gumbov, besedil, ..., activity pa se uporablja za prikaz menija in postavitev lebdečih gumbov. En tak gumb uporabljamo tudi mi v naši aplikaciji. Gradnjo naše aplikacije smo začeli tako, da smo izdelali nov projekt imenovan WEB-KER. Določili smo začetno verzijo OS Android, od katere naprej bo naš app podprt, ter izbrali prazno predlogo (blank activity), ki nam na zaslonsko sliko postavi dve začetni zasnovi zaslonskih slik (layouta content ter activity). Prva postavitev zaslonske slike je uporabljena kot začetna zaslonska slika, ki se najprej prikaže uporabniku.

V postavitev content smo vstavili štiri gumbe za nadomeščanja, ocene, urnike ter izostanke. Nato je sledilo oblikovanje oblike gumbov. Za oblikovanje gumbov naredimo v mapi drawable novo datoteko, v katero napišemo kodo ter nato kličemo to datoteko pri gumbih v content layoutu. Ko smo to naredili smo se posvetili lebdečemu gumbu, ki smo ga uporabili



za prikaz splošnih informacij o tej aplikaciji. Izdelali smo tudi ikono, katere namen je preusmeriti uporabnika v drugo zaslonsko sliko, kjer se lahko uporabnik poveže na šolski strežnik z uporabniškim imenom in geslom. Morali smo narediti novi zaslonski sliki, ki smo ju poimenovali »o_aplikaciji«. V content layoutu tega dela aplikacije smo vstavili sliko s splošnimi informacijami. Za prikaz teh informacij ob kliku na lebdeči gumb je bilo potrebno napisati kodo, ki ob kliku prikaže to zaslonsko sliko. Aktivnosti ob kliku na gumb je potrebno zapisati s programsko kodo, ki se nahaja v mapi »java«. Pri tem delu ustvarjanja aplikacije nismo imeli posebnih težav saj je na spletu dovolj primerov in pomoči, da je šlo vse po tekočem traku.

Slika 1 Postavitev gumbov na začetni zaslonski sliki

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item >
    <shape android:shape="rectangle" >
      <gradient android:angle="-90" android:startColor="@color/belaBarva" android:endColor="@color/belaBarva" />
    </shape>
  </item>
</selector>
```

Slika 2 Koda za oblikovanje gumbov

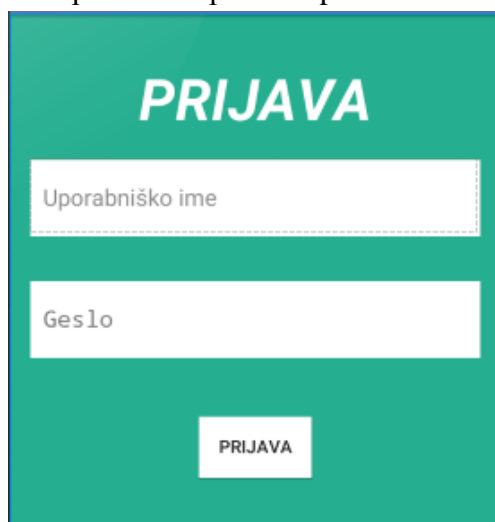
4.2 Prijava v šolski server

Ob kliku na ikono za prijavo, aplikacija preusmeri uporabnika na zaslonsko sliko prijave. Na njej sta dve polji za vstavljanje besedila ter en gumb. V prvo polje se vpiše uporabniško ime (le-to je naslov elektronske pošte), drugo polje pa je polje za vpis gesla.



Slika 3 Ikona za prijavo

Ko uporabnik vpiše vse podatke in klikne gumb prijava, se izvedeta dve metodi: GET in POST. Metoda GET preveri povezavo s strežnikom in priskrbi Connection ID. Izvede se z naslednjim klicem:



PRIJAVA

Uporabniško ime

Geslo

PRIJAVA

<https://web.sc-celje.si/cgi-pub/EviWebPUB.exe/getdata?CID=000:-:0000000000&ACT=getPUBCID>

Podatki s tega klica pridejo v obliki spletne strani, zato smo morali uporabiti JSOUP, to je zunanja knjižnica, ki jo vstavimo v našo aplikacijo in nam omogoča, da iz spletne strani dobimo točno določen del, ki ga potrebujemo, v našem primeru CID.

Slika 4 Zgled prijave

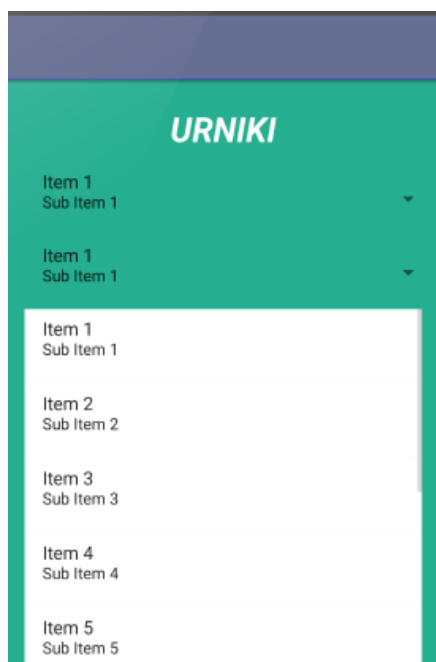
Po pridobitvi CID se izvede še POST metoda, ki poskrbi za prenos uporabniškega imena in gesla. Slednja podatka se skupaj z URL naslovom preneseta kot parametra. Če je prijava z identifikacijskimi elementi uspešna, dobimo v odgovoru naslednje podatke:

[CID=001:-:1609000193;APP=1111111100;PRI=1111101111;IDD=9727;AVT=2016-08-25;UAE=N;USR=D;](#)

Med vsemi prispelimi podatki potrebujemo v naši aplikaciji le CID in IDD, zato smo spet uporabili JSOUP za izločitev samo teh dveh podatkov. Po uspešni prijavi se aplikacija vrne na začetno zaslonsko sliko.

V tem delu komunikacije smo imeli v začetku nekaj težav, kako posredovati identifikacijske elemente z metodo POST. Odpravili smo jih z pomočjo mentorja.

4.3 Urniki



Slika 5 Zgled urnikov

Do urnikov lahko, za razliko od dostopa do ocen in izostankov, dostopa vsak uporabnik brez prijave. Ob kliku na začetnem zaslonu na gumb »Urniki« preusmeri aplikacija uporabnika na zaslonsko sliko urnikov. Tu smo dodali seznam (listview), ki bo prikazal za vsak razred dnevni urnik. Dodali smo tudi dva izbirnika (spinner), katerih funkcija je, da lahko izberemo oddelek, ter dan za katerega želimo videti urnik. Na osnovi teh izbirnikov aplikacija izloči potrebne podatke, nato pa jih vstavi v seznam in tako sestavi urnik po dnevih. Uporabnik lahko torej izbere poljuben oddelek in dan, za katerega hoče videti urnik, ki se izpiše v seznam in prikaže na zaslonu.

Pri izpisu urnika smo imeli nekaj težav z vstavljanjem podatkov v seznam, a smo jih hitro odpravili.

4.4 Nadomeščanja

Za izdelavo nadomeščanj smo poleg znanja Jave potrebovali tudi znanje HTML-ja, kajti vsi potrebni podatki so zapisani v spletni strani šole za nadomeščanja. Od tu jih je bilo potrebno izločiti in jih potem izpisati tako, da je izpis čim bolj pregleden. To delo je bilo zelo naporno, saj je bilo potrebno napisati veliko kode. Pri tem se je pogosto vmes tudi kaj zalomilo in tako upočasnilo nadaljnje delo.

S pripravo izpisa nadomeščanj smo imeli ogromno težav, saj sprva nismo vedeli, na kakšen način se naj sploh lotimo dela. Na internetu smo zasledili ogromno načinov, ki so se na koncu izkazali za neuspešne poizkuse.

4.4.1 HTML koda strani

Nadomeščanja so na spletni strani shranjena v tabeli. Tabela je zgrajena iz 7 stolpcev in sicer:

- ime in priimek odsotnega učitelja,
- ime in priimek učitelja ki nadomešča,
- šolska ura nadomeščanja,
- razred,
- učilnica,
- predmet,
- ter po izbiri še nazadnje dodana opomba, ki ni obvezna.

Ti podatki so v HTML-ju shranjeni na naslednji način:

```
<table id="nadomescanja">
  <thead>...</thead>
  <tbody>
    <tr>
      <td>Jančič, Jernej</td>
      <td>Delo doma</td>
      <td>0</td>
      <td>K4A-an/f • K3A/f • K4A-p/2f</td>
      <td>TV2</td>
      <td>ŠVZt</td>
      <td>Delo doma</td>
    </tr>
```

Slika 6 Primer HTML kode za nadomeščanja

4.4.2 Programiranje

Opazili smo, da so ti podatki edini, ki so zapisani v oznakah `<td>`. Tako smo se poglobili v izločanje teh podatkov in končno našli pravo rešitev. Z Jsoup-om smo poiskali vsak `<td>`, začeli pa smo s številko 0.

Naprej smo deklarirali osnovne podatke ki jih potrebujemo:

```
private Document htmlDocument;
private String htmlPageUrl = "http://ker.sc-celje.si/nadomescanja/";
private TextView IzpisanPodatek;
private String XX;
```

Slika 7 Primer kode za pridobivanje podatkov za nadomeščanja

Nato smo naš besedilni pogled (Textview) z id-jem povezali z objektom, katerega vsebino želimo prikazati.

Z uporabo Jsoup-a smo se povezali na spletno stran šolskega strežnika z nadomeščanji. Nato smo ustvarili element, ki prebere vse »<td>« objekte. V niz (string) poimenovan XX smo pretvorili ta niz z dodano izračunano številko (indeksom).

```
htmlDocument = Jsoup.connect(htmlPageUrl).get();
Elements elements = htmlDocument.select("td");
XX = elements.get(0).text();
```

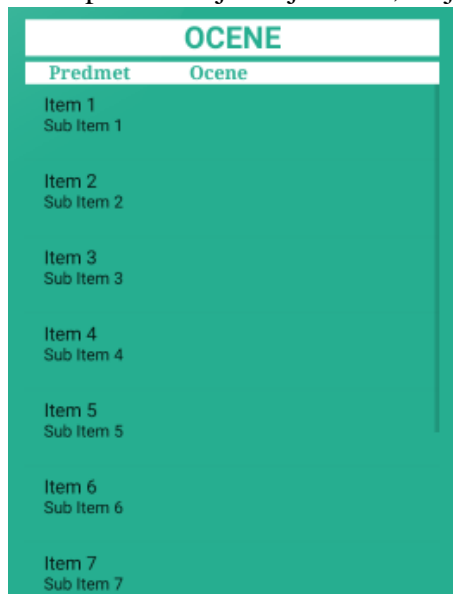
Slika 8 Klicanje spletne strani ter pridobitev podatkov

Ker so elementi razvrščeni po vrsti je možno le s številko (indeksom) vedeti, kateri element predstavlja kateri podatek. Tako smo ugotovili, da je vsak 8 element (pri vsakem 8 indeksu) prvi podatek sledečega nadomeščanja (0 predstavlja učitelja, ki je odsoten, 7 pa je zadnji podatek tega nadomeščanja in sicer vrsta nadomeščanja). 8 element predstavlja novo nadomeščanje, spet z odsotnim učiteljem.

Na koncu smo naredili le še urejen in estetsko primeren zaslon za prikazovanje nadomeščanj ločeno po datumu (za trenutni dan, ter za naslednji dan).

4.5 Ocene

Dostop do ocen je urejen tako, da jih lahko uporabnik dobi le, če je prijavljen v EviWeb SIS



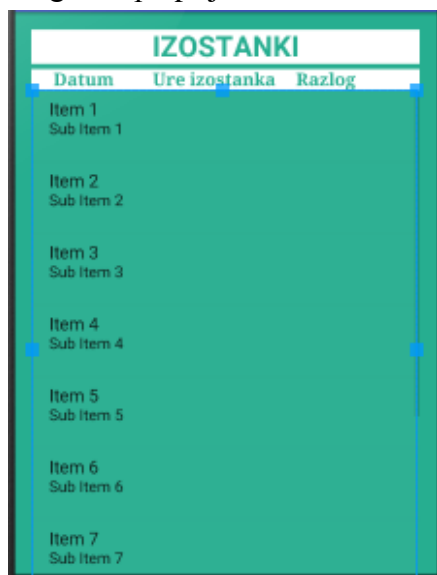
OCENE	
Predmet	Ocene
Item 1 Sub Item 1	
Item 2 Sub Item 2	
Item 3 Sub Item 3	
Item 4 Sub Item 4	
Item 5 Sub Item 5	
Item 6 Sub Item 6	
Item 7 Sub Item 7	

šolskega strežnika. Po kliku gumba »Ocene« na začetni strani, preusmeri aplikacija uporabnika na zaslon za prikaz ocen. Na njem je seznam, ki je razdeljen v dve koloni za vstavljanje besedila. V prvi so kratice predmetov, v drugi pa ocene. Ocene pridobi aplikacija na osnovi metode GET in ustrezne funkcije za prejetje podatkov iz strežnika. Za uspešno izveden klic rabi aplikacija IDD, ki ga prejme ob prijavi. V odgovoru v obliki spletne strani prejmemo veliko količino podatkov (podatki o učiteljih, predmetih redovalnici in ocenah), ki jih nato program obdela z knjižnico Jsoup in nam v listview naloži predmete in pripadajoče ocene.

Slika 9 Zgled ocen

4.6 Izostanki

Izostanki so izdelani podobno kot ocene, saj so tudi to osebni podatki. Pristop do nji je torej mogoč le po prijavi v EviWeb SIS na šolskem strežniku. Podobno kot ocene so tudi izostanki zgrajeni iz enega seznama, ki je razdeljen v tri kolone za vstavljanje besedil.



IZOSTANKI		
Datum	Ure izostanka	Razlog
Item 1 Sub Item 1		
Item 2 Sub Item 2		
Item 3 Sub Item 3		
Item 4 Sub Item 4		
Item 5 Sub Item 5		
Item 6 Sub Item 6		
Item 7 Sub Item 7		

V prvi so datumi, ko je uporabnik manjkal pri pouku, drugi je namenjen manjkajočim uram in zadnji služi izpisu razloga manjkanja, če ga je učitelj vpisal.

Tudi izostanke dobimo s klicem metode GET in IDD, ki od EviWeb SIS na šolskem strežniku pridobi podatke o izostankih v obliki spletne strani. Tudi tu uporabimo Jsoup knjižnico za izločanje tistih podatkov, ki jih potrebujemo in vstavimo v seznam.

Slika 10 Zgled izostankov

5. Zaključek

5.1 Hipoteze

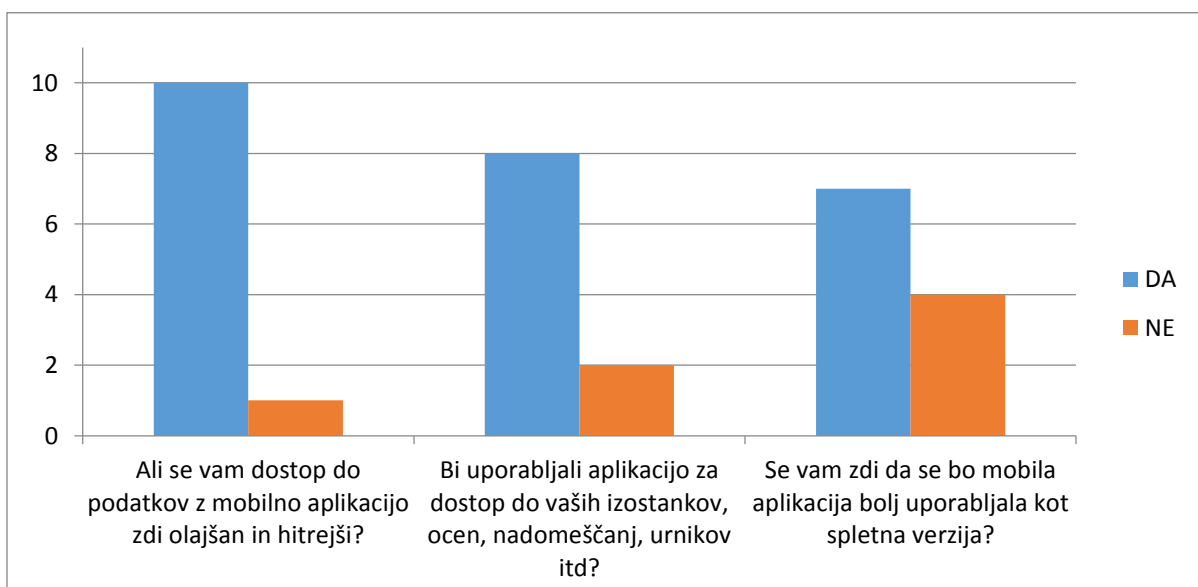
Naše hipoteze so se glasile:

- Dostop do podatkov je z mobilno aplikacijo olajšan
- Uporabniki želijo imeti mobilno aplikacijo
- Mobilna aplikacija bo bolj v uporabi kot pa spletna verzija

Pri potrjevanju hipotez smo si pomagali z že omenjeno anketo, ki smo jo izvedli v našem oddelku. Sicer je bila zelo kratka, a tudi v tem primeru se je izkazalo, da anketiranci niso vedno »navdušeni« nad izpolnjevanjem anketnih vprašanj. Tako po izvedbi ankete nismo dobili vseh izpolnjenih odgovorov od vseh anketiranih.

Kljub temu smo dobili dovolj glasov za potrditev naših hipotez. 90% anketirancev se zdi lažji dostop do virov z uporabo telefona, ker je vedno pri roki. Prav tako se večina (80%) strinja, da bi uporabljali našo aplikacijo. 63 % anketirancev pravi, da bi se naša aplikacija najverjetneje bolje uporabljala kakor spletna verzija.

Rezultati ankete so predstavljeni v spodnjem grafikonu.



Na dejanski odziv in uporabnost aplikacije bomo morali še nekoliko počakati in preveriti njeno dejansko učinkovitost in uporabnost.

Kot programerji in uporabniki mobilnih telefonov in aplikacij lahko zatrdimo, da je zagon aplikacije in dostop do podatkov hitrejši in enostavnejši kot brskanje za podatki po spletnih

straneh strežnika, za kar moramo predhodno pognati brskalnik in šele nato prebrskati spletne strani, da pridemo do zelenih podatkov.

Veliko sošolcev in drugih dijakov je pogosto omenjalo, da bi bila aplikacija za mobilni telefon, ki bi nudila osnovne podatke šolskega informacijskega sistema, zelo dobrodošla, seveda v primeru, da bi bila tudi preprosta in hitra za uporabo. S pripravo zaslonskih slik in preproste navigacije smo naredili aplikacijo zelo odzivno in enostavno za uporabo. Iz teh razlogov predvidevamo, da bo aplikacija res služila namenu.

Glede na rezultate ankete lahko predpostavimo, da bo aplikacija dosegla svoj namen in jo bodo dijaki naše šole uporabljali. K temu jih bomo vzpodbudili tudi z namestitvijo aplikacije na spletne strani šole, kjer jo bo mogoče prenesti na svoj mobilni telefon. Ker prevladujejo mobilni telefoni z operacijskim sistemom Android menimo, da bo prenosov in uporabe te aplikacije veliko. Vsekakor pa bodo o tej hipotezi imeli zadnjo besedo uporabniki.

5.2 Težave pri delu

Skozi izdelavo našega projekta smo naleteli na mnogo ovir in težav. S pomočjo spleta, video vodičev in pomočjo mentorja smo jih s skupnimi močmi eno za drugo rešili. Največ težav je bilo s klicem in pridobitvijo podatkov iz šolskega strežnika, saj je bil dostop do »žive« delujoče aplikacije za nas nekaj novega, pri vajah pri pouku in v laboratoriju pa tega nismo spoznali. Z vztrajnostjo pa smo na koncu le prebredli tudi te težave in uspeli vzpostaviti komunikacijo z aplikacijo.

Zaradi časovne stiske in precej zapletov, ki so nam vzeli veliko časa, v času oddaje te raziskovalne naloge nismo uspeli aplikacije dokončati v celoti. Postavili smo zaslonske slike in uredili prehajanje med njimi, vseh dobljenih podatkov iz aplikacije EviWeb SIS pa še nismo uspeli vključiti v prikaz zaslonskih slik. Slednje bomo še dopolnili do predstavitve raziskovalne naloge, ko naj bi bila aplikacija WEB KER za Android v polni funkcionalnosti.

5.3 Kaj smo se novega naučili?

Med delom smo se naučili veliko novega s področja programiranja v Javi, kot tudi o izdelavi aplikacij v Android studiu. Naučili smo se uporabljati klice metod GET in POST v Androidu v povezavi »živo« aplikacijo kot tudi spletnimi stranmi. Tudi uporaba Jsoup knjižnic je bila novost, pri katerih smo ugotovili visoko uporabnost pri iskanju podatkov iz »standardnih« zapisov. Vse novo znanje nam bo gotovo v pomoč v nadaljnjem šolanju, kot tudi kasneje v življenju in v službi, saj je število aplikacij za Android iz dneva v dan večje, možnosti za komuniciranje z mobilnimi napravami pa vse hitrejše in bolj uporabne.

6. Viri

GET metoda. Programnerguru. 2014. Dosegljivo na: <http://programnerguru.com/android-tutorial/android-restful-webservice-tutorial-how-to-call-restful-webservice-in-android-part-3/> (datum ogleda: 15.1.2016)

Razcepitev dveh podatkov. Stackoverflow. 2012. Dosegljivo na: <http://stackoverflow.com/questions/12121322/string-between-two-commas-java> (datum ogleda: 15.1.2016)

Dodajanje več vrstic v listview. Youtube. 2014. Dosegljivo na <https://www.youtube.com/watch?v=JLF7b0ftIEc> (datum ogleda: 8.2.2016)

POST metoda. wikipow.2014. Dosegljivo na: <http://www.wikihow.com/Execute-HTTP-POST-Requests-in-Android> (datum ogleda: 16.1.2016)

Spinner za urnike.2013. Dosegljivo na: <https://www.youtube.com/watch?v=aApS2W-j8oM> (datum ogleda: 20.2.2016)

Parsanje z Jsoup-om. 2015. Youtube. Dosegljivo na: <https://www.youtube.com/watch?v=OuMCtg5z5ns> (datum ogleda: 19.2.2016)