

Ponedeljek, 3. oktober 2011	začetek predavanj v zimskem semestru	Ura
Teden 43	Poročilo	1
Teden 44	Načrt dela	1
Teden 45	Razpoznavanje spletne strani	
Teden 46	Razpoznavanje spletne strani	
Teden 47	Razpoznavanje spletne strani	10
Teden 48	Integracija v android	3
Teden 49	Logika	
Teden 50	Logika	
Teden 51	Logika	8
Teden 52	Integracija v budilko	
Teden 53	Integracija v budilko	2
Teden 54	Grafični vmesnik	
		4
Teden 57	Konec	
Petek, 20. Januar 2012	konec predavanj v zimskem semestru	

Budilka, ki se bo odločila ali nas zbudi, glede na vreme.

Poročilo tedenskega dela <https://github.com/slovenec88/brezskrbnik>

1. Teden

Priprava poročila

Pri predmetu RAIN sem sprogramiral in v projekt dodal budilko, ki jo je mogoče nastavljati preko spletne storitve. Spletno storitev sem zapisal v C#. Zadeva zaenkrat deluje po principu, v spletni storitvi shrani nek čas v spremenljivko (prek vmesnika), nato pa na telefonu klikneš gumb »nastavi budilko preko spleta«. Ta se sinhronizira na spletni čas budilke.

Zadevo bi lahko nastavil tako, da bi delovala preko googlovih storitev. Dodatne implementacije, so npr nastavljanje budilke za različne dneve. Zanimiva uporaba, bi bila še uporabljanje vseh googlovih storitev, torej implementacija koledarja, mogoče kakšnega widgeta.. dodal bi lahko kakšno vreme. Nastavljanje ozadja preko interneta in podobne funkcionalnosti so bile že v prvotnem načrtu. Torej upravljanje telefona na daljavo, kot bi ga imel v rokah, vendar brez žive slike, torej preko menijev in ne dejanske slike zaslona telefona. Mogoče bi lahko delil tudi svojo lokacijo na internetu, glede na to da podpira aplikacija tudi neke vrste navigacijo (delitev lokacije je že mogoča preko sms sporočil).

2. Teden

Ponedeljek, 3. oktober 2011	začetek predavanj v zimskem semestru	Ura
Teden 43	Poročilo	1
Teden 44	Načrt dela	1
Teden 45	Parsanje spletne strani	
Teden 46	Parsanje spletne strani	
Teden 47	Parsanje spletne strani	10
Teden 48	Integracija v android	3
Teden 49	Logika	
Teden 50	Logika	
Teden 51	Logika	8
Teden 52	Integracija v budilko	
Teden 53	Integracija v budilko	2
Teden 54	Design	4
Teden 55	Spletna stran	
Teden 56	Spletna stran	10
Teden 57	Konec	
Petek, 20. Januar 2012	konec predavanj v zimskem semestru	

3. Teden

Priprava potrebnih informacij, ogled kode spletne strani.

```
<span id="ctl00_cphContent_repForecastDay_ctl00_lblDate">Monday 1/16/2012</span>
```

```
<span id="ctl00_cphContent_repForecastDay_ctl00_lblDesc">Brilliant sunshine</span>
```

```
<span id="ctl00_cphContent_repForecastDay_ctl00_Label1">High Temperature</span>
```

Takole so vpisani podatki (znotraj še več divov).

4. Teden

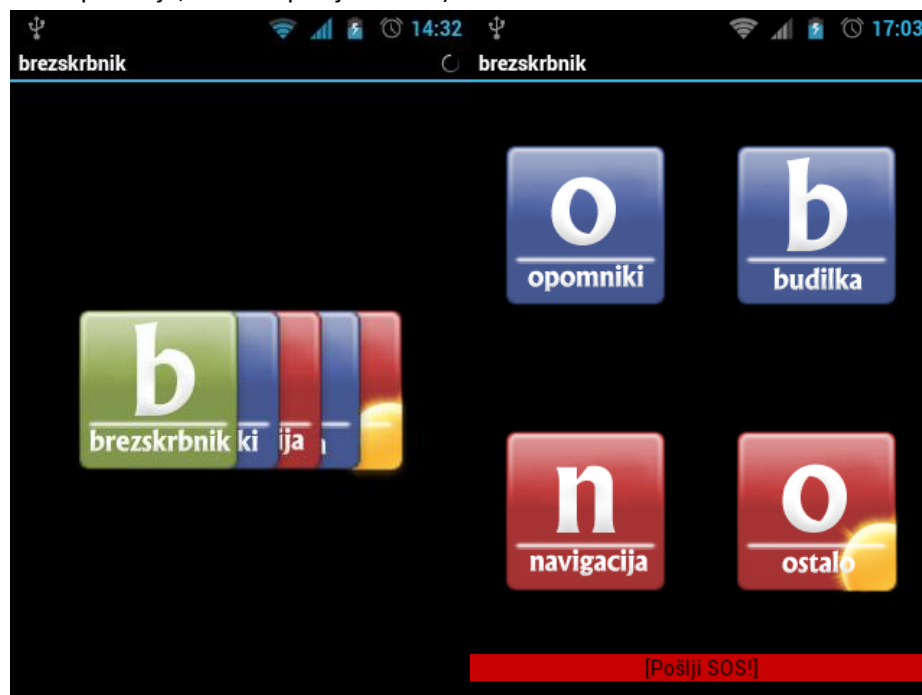
Inštalacija spletne storitve na računalnik in integracija htmlcleaner knjižnice. Imel sem problem z dodajanjem .jara v eclipse, saj ni hotel razparkirati knjižnice v sam strežnik. Asistent mi je pomagal pri problemu in zadevo smo rešili na način, da sem .jar htmlclenerja dodal v lib mapo tomcata (neposredno).

5. Teden

Napisal sem razred vreme, iz katerega kreiram instance za vsak dan posebaj.

```
public Vreme(String datum, String realfeel, String maxtemp, String opis, String slika){  
    this.datum = datum;  
    this.realfeel = realfeel;  
    this.maxtemp = maxtemp;  
    this.opis = opis;  
    this.slika = slika;  
}
```

In tudi metode, za vse možne nastavitve. (Pridobim več podatkov, kot jih zaenkrat aplikacija na androidu uporablja, zato ne pošljem vseh).



6. Teden

Izdelava razreda service2, pridobivanje podatkov iz spleta. Uporabil sem funkcijo za razpoznavanje, ki sem jo napisal že pri predmetu prevajalniki.

```
public TagNode xmlCleaner(String url) {
    CleanerProperties props = new CleanerProperties();
    props.setTranslateSpecialEntities(true);
    props.setTransResCharsToNCR(true);
    props.setOmitComments(true);
    TagNode tagNode;
    try {
        tagNode = new HtmlCleaner(props).clean(new URL(url));
        return tagNode;
    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return null;
}

private TagNode[] findInfo(TagNode node, String XPathExpression) {
    TagNode[] description_node = null;
    Object[] object = null;
    try {
        object = node.evaluateXPath(XPathExpression);
        description_node = new TagNode[object.length];
        for(int i=0; i<object.length; i++){
            description_node[i] = (TagNode) object[i];
        }
    } catch (XPatherException e) {
        e.printStackTrace();
    }
    return description_node;
}

TagNode stran = xmlCleaner("http://www.accuweather.com/en-us/si/" + kraj + "/" + kraj +
    "/forecast.aspx");
TagNode[] all = findInfo(stran,
    "//*[id='content_640']//div[@class='fltLeft'][1]//div[@style='margin-bottom: 10px;']");
```

To je klic metode, ki mi zgenerira xml kodo iz htmlja.

7. Teden

Predelava razreda v spletno storitev, dodan Bing translator API. Bing translator mi prevede opis iz spletne strani accuweather, ki jo pridobim s html cleanerjem, nato pa ga prevedem iz angleščine v slovenščino (poljubno).

```
String prevod = Translate.execute(vreme[1].getOpis().toString(), Language.SLOVENIAN);
```

8. Teden

Izdelava aplikacije na telefonu, ki preko soapa pridobi podatke iz spletnega strežnika in nam jih posreduje v obliki slik in tekstovnega niza vremena. Dodan asinhroni klic. Napisan custom dialog.

```
public class BackgroundAsyncTask2 extends AsyncTask<Void, Integer, String> {

    @Override
    protected String doInBackground(Void... params) {

        vremenskiPodatki = onDobiVreme();

        return vremenskiPodatki;
    }

    @Override
    protected void onPreExecute() {
        setProgressBarIndeterminateVisibility(true);
    }

    protected void onPostExecute(String arg) {
        setProgressBarIndeterminateVisibility(false);
        if ((vremenskiPodatki.substring(0, 6)).equalsIgnoreCase("napaka")) {
            Toast.makeText(AlarmActivity.this,
                "Napaka v komunikaciji!", Toast.LENGTH_LONG).show();
        }
        else {
            vremenskiPodatkiTabela = vremenskiPodatki.split(";");
            odpriDialogOdlocanja();
        }
    }
}

public String onDobiVreme() {
    try {
        SoapObject request = new SoapObject(NAMESPACE2, METHOD_NAME2);
        request.addProperty("kraj", kraj);
        SoapSerializationEnvelope envelope = new
        SoapSerializationEnvelope(SoapEnvelope.VERSION1);
        envelope.dotNet = true;
        envelope.setOutputSoapObject(request);
        HttpTransportSE androidHttpTransport = new HttpTransportSE(URL2);
        androidHttpTransport.call(SOAP_ACTION2, envelope);
        Object result = envelope.getResponse();
        return result.toString();
    }
    catch (Exception e) {
        e.printStackTrace();
        return "napaka: " + e.toString();
    }
}
```

9. Teden

Dodana logika, ki nam omogoča lažjo odločitev ali želimo nastaviti budilko glede na vremenske pogoje naslednjega dne.

```
if (vremenskiPodatkiTabela[0].toString().equalsIgnoreCase("13_int.jpg")) {  
    ivSlika.setImageDrawable(getResources().getDrawable(R.drawable.weather_13));  
    btnPrekllici.setTextColors(Color.parseColor("#177BBD"));
```

Gumb potrdi, oz. prekliči, se samodejno nastavi glede na predlagano situacijo. Napisan meni, ki nam omogoča prikaz naprednih ukazov.

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.SpremeniPogled:  
            tv1.setVisibility(View.VISIBLE);  
            tv2.setVisibility(View.VISIBLE);  
            tvIP.setVisibility(View.VISIBLE);  
            ...
```



10. Teden

Predstavitev projekta in dopisan načrt dela.