

RAZVOJ MOBILNE PROGRAMSKE PODRŠKE ZA IZRAČUN PROVJESA IZOLIRANIH ENERGETSKIH VODOVA

Vukadin, Luka

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:200:914381>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-07-10**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science
and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Sveučilišni studij

**RAZVOJ MOBILNE PROGRAMSKE PODRŠKE ZA
IZRAČUN PROVJESA IZOLIRANIH ENERGETSKIH
VODOVA**

Završni rad

Luka Vukadin

Osijek, 2022.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

Osijek, 20.09.2022.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Luka Vukadin
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	R4296, 26.07.2018.
OIB Pristupnika:	88786220926
Mentor:	Izv. prof. dr. sc. Damir Blažević
Sumentor:	,
Sumentor iz tvrtke:	
Naslov završnog rada:	Razvoj mobilne programske podrške za izračun provjesa izoliranih energetske vodova
Znanstvena grana rada:	Programsko inženjerstvo (zn. polje računarstvo)
Zadatak završnog rada:	Izraditi mobilnu aplikaciju za izračun provjesa u karakterističnim i proizvoljnim točkama (udaljenost, temperatura) za neizolirane energetske vodove. Oformiti bazu tipskih vodova te ostaviti mogućnost pregleda i unosa podataka o vodovima, snimanje i izvoz rezultata proračuna, te kreiranje tablica provjesa. Rješenje temeljiti na open source tehnologijama. Tema rezervirana za: Luku Vukadina
Prijedlog ocjene završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	20.09.2022.
Datum potvrde ocjene od strane Odbora:	
Potvrda mentora o predaji konačne verzije rada:	<i>Mentor elektronički potpisao predaju konačne verzije.</i>
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 21.09.2022.

Ime i prezime studenta:

Luka Vukadin

Studij:

Preddiplomski sveučilišni studij Računarstvo

Mat. br. studenta, godina upisa:

R4296, 26.07.2018.

Turnitin podudaranje [%]:

14

Ovom izjavom izjavljujem da je rad pod nazivom: **Razvoj mobilne programske podrške za izračun provjesa izoliranih energetske vodova**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Damir Blažević

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD	1
1.1. Zadatak završnoga rada	1
2. MEHANIČKI IZRAČUN PROVJESA	2
2.1. Određivanje parametara mehaničkog izračuna	2
2.2 Izračun provjesa	3
3. REALIZACIJA MOBILNE APLIKACIJE	6
3.1 Korisničko sučelje	6
3.2 Povezivanje korisničkog sučelja i varijabli	9
3.3 Java dio	9
4. USPOREDBA REZULTATA	15
4.1 Ručni izračun provjesa	15
4.2 Računanje pomoću aplikacije	19
5. ZAKLJUČAK	21
LITERATURA	21
SAŽETAK	23
ABSTRACT	24
ŽIVOTOPIS	25
PRILOZI	26

1. UVOD

Nadzemni vodovi kojima električna energija putuje su ključni za funkcioniranje današnjeg svijeta. Elektrane koje proizvode električnu energiju kao što su hidroelektrane i termoelektrane često se na udaljenim prostorima te se radi toga električna energija dovodi do kućanstva uz nadzemne vodove. Mreže nadzemnih vodova zahtijevaju projektiranje koje omogućuje distribuciju električne energije uz minimalne gubitke. Različiti metali od kojih su izrađeni nadzemni vodovi prilikom određenim temperaturama se skupljaju ili istežu. Upravo projektiranje takvih vodova zahtijeva precizno izračunavanje provjesa nadzemnog voda kako bi se povećala sigurnost i smanjio trošak izgradnje nadzemnih vodova. Za izračunavanje provjesa potrebni su brojni parametri koji se često mijenjaju te tako uzrokuju zahtjevna ponovna računanja. Ideja kako bi se minimiziralo vrijeme koje je potrošeno za izračun i izračun učinilo dostojno preciznim je upravo izrada aplikacije za izračun provjesa.

Zadatak ovoga završnoga rada je napraviti Android mobilnu aplikaciju koja se koristi za izračun provjesa izoliranih energetske vodova. Na početku rada opisan je način na koji se provjes nadzemnih vodova računa. Napisani su potrebne formule, podaci i postupak za izračun provjesa na posebnim temperaturama. Slijedeći dio rada opisuje izvedbu algoritma za dobivanje provjesa opisanog u ranijim dijelovima aplikacije za Android aplikacijski sustav u programskom jeziku Javi. Navedeni su korisničko sučelje, korišteni alati i opis programskog koda. U zadnjem dijelu su uspoređeni rezultati koji se dobivaju izračunom preko aplikacije s rezultatima koji su dobiveni ručnim proračunom na papiru.

1.1. Zadatak završnoga rada

Projektiranje Android mobilne aplikacije za izračunavanje provjesa izoliranih energetske vodova. Treba napraviti algoritam i mobilnu aplikaciju za računanje provjesa izoliranih energetske vodova za zadana ranije pomoću visine vodova, svojstava voda i temperature okoline. Povijest proračuna treba biti dostupna korisniku i podržana od strane Android platformi.

2. MEHANIČKI IZRAČUN PROVJESA

Mehaničkim izračunom nadzemnih vodova se izračunava naprezanje i provjes voda ovisno o temperaturama. Izračunom se osigurava može li mehaničko naprezanje voda i provjes preći dopuštenu granicu u izvanrednim uvjetima. Važni podaci koji su potrebni prilikom projektiranja nadzemnih vodova i njihovih mreža su provjes i naprezanje kojima se izračunavaju razmak između stupova i visine stupova. U obzir treba uzeti da prilikom nekih temperatura vod nosi i težinu snijega.

Tri su vrste točnosti mehaničkog izračuna:

1. Približni izračun
2. Točni izračun
3. Vrlo točni izračun

Približni izračun u inženjerskom projektiranju se koristi zbog svoje jednostavnosti. Približni izračun se koristi zato što u izračunu provjesa nadzemnih vodova nije očekivana milimetarska preciznost i krivulja se prikazuje parabolom dok vrlo točni izračunom krivulja provjesa se prikazuje elastičnom lančanicom, u točnom izračunu krivulja se prikazuje sa lančanicom.

Temperatura vodiča je glavni čimbenik te se prema [1] koriste tri temperature:

1. $t = -20 \text{ }^\circ\text{C}$ – najniža temperatura za koju se izgrađuju vodiči
2. $t = -5 \text{ }^\circ\text{C}$ – temperatura zaleđivanja
3. $t = 40 \text{ }^\circ\text{C}$ – najviša temperatura okoline

2.1. Određivanje parametara mehaničkog izračuna

Unaprijed izračuna potrebno je odrediti sve parametre koji utječu na izračun. Parametri vezani uz stupove i vodič. Parametri stupova su:

- visina prvog stupa – h_1 [m]
- visina drugog stupa – h_2 [m]
- denivelacija (razlika između prvog i drugog stupa) – h_{12} [m]
- raspon – a [m]
- spojnica – a' [m]

Parametri vodiča su:

- Promjer vodiča – d [mm]
- Presjek vodiča – A [mm]
- Uzdužna masa – m [kg/m]
- Modul elastičnosti – E [N/mm²]
- Koeficijent linearnog toplinskog istezanja – β [1/ ° C]
- Normalno dozvoljeno istezanje – σ_d [N/mm²]
- Iznimno dozvoljeno istezanje – σ_i [N/mm²]
- Maksimalno dozvoljeno istezanje – σ_m [N/mm²]

2.2 Izračun provjesa

Uz [2] za proračun provjesa potrebno je izračunavanje vlastite težine vodiča:

$$G_0 = m_1 * g \left[\frac{N}{m} \right] \quad (2 - 1)$$

m_1 – masa vodiča [kg / m]

g – gravitacija, $g = 9,81$ m/s

Za izračun provjesa koristi se reducirana težina vodiča:

$$g_0 = \frac{G_0}{A} \left[\frac{N}{m * mm^2} \right] \quad (2 - 2)$$

Normalno dodatno opterećenje treba iznositi manje:

$$G_{l0} = 0,18 * \sqrt{d} \left[\frac{N}{m} \right] \quad (2 - 3)$$

Stvarno dodatno opterećenje je:

$$G_l = k * G_{l0} \left[\frac{N}{m} \right] \quad (2 - 4)$$

Nakon toga se zaključuje da reducirana težina zaleđenog vodiča iznosi:

$$g_z = \frac{G_0 + G_l}{A} \left[\frac{N}{m * mm^2} \right] \quad (2 - 5)$$

Mora se odrediti idealni i kritični raspon vodiča.

Kritični raspon vodiča iznosi:

$$a_k = \sigma_{max} \sqrt{\frac{360 \cdot \beta}{g_z^2 - g_0^2}} [m] \quad (2-6)$$

σ_{max} je maksimalno naprežanje vodiča koje mora biti jednako ili manje dopuštenom naprežanju.

Idealni raspon iznosi:

$$a_{idealno} = \sqrt{\frac{\sum_{i=1}^n a_i^3}{\sum_{i=1}^n a_i^2} * \frac{\sum_{i=1}^n a_i^3}{\sum_{i=1}^n a_i^2}} [m] \quad (2-7)$$

Ako je $a_{idealno} < a_k$ onda za početno stanje uzima temperatura $t = -20^\circ C$

- $t = -20^\circ C$
- $g_1 = g_0$
- $\sigma_1 = \sigma_{max}$

Ako je $a_{idealno} > a_k$ onda za početno stanje uzimamo $t = -5^\circ C$

- $t = -5^\circ C$
- $g_1 = g_z$
- $\sigma_1 = \sigma_{max}$

Potrebno je izračunati i horizontalno naprežanje. U slučaju da nema denivelacije onda se horizontalno naprežanje računa po formuli:

$$\bar{\sigma}_1 = \sigma_1 = \sigma_{max} \left[\frac{N}{mm^2} \right] \quad (2-8)$$

Ako ima denivelacije onda vrijedi:

$$\bar{\sigma}_1 = \sigma_1 * \frac{\sum_{i=1}^n \frac{a_i^3}{a_i^2}}{\sum_{i=1}^n \frac{a_i^2}{a_i}} \left[\frac{N}{mm^2} \right] \quad (2-9)$$

Nadomjesno naprežanje određujemo uz raspisivanje jednadžbe za kosi raspon:

$$\frac{\bar{\sigma}_1 - \bar{\sigma}_2}{E} + \beta * (\theta_1 - \theta_2) = \frac{a_{idealno}^2}{24} * \left(\frac{g_1^2}{\bar{\sigma}_1^2} - \frac{g_2^2}{\bar{\sigma}_2^2} \right) \quad (2-10)$$

gdje je:

- $\bar{\sigma}_2$ nadomjesno naprezanje za danu temperaturu
- θ_2 željena temperatura
- g_2 je g_z u slučaju da je temperatura -5°C dok inače vrijedi $g_2 = g_0$

Uz formulu (2-9) određuje se stvarno naprezanje:

$$\sigma_2 = \bar{\sigma}_2 * \frac{\sum_{i=1}^n \frac{a_i^2}{a_i^3}}{\sum_{i=1}^n \frac{a_i}{a_i}} \left[\frac{N}{\text{mm}^2} \right] \quad (2-11)$$

Zadnji korak za raspon stupova i odabrano temperaturu računamo provjes:

- Za slučaj kada je $h_{12}=0$

$$f = \frac{a^2 * g_2}{8 * \sigma_2} [m] \quad (2-12)$$

- Za slučaj kada je $h_{12} \neq 0$

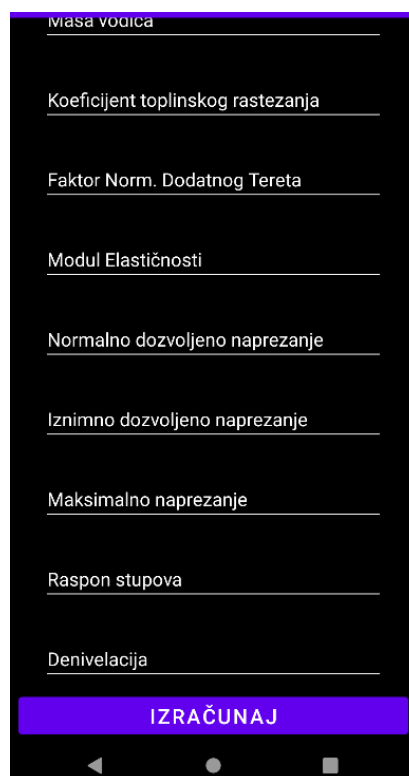
$$f = \frac{a^2 * g_2}{8 * \sigma_2} * \frac{a'}{a} [m] \quad (2-13)$$

3. REALIZACIJA MOBILNE APLIKACIJE

Računanje pomoću aplikacije nije nužno, ali računanje ubrzava proces i pojednostavnjuje računanja provjesa nadzemnog voda. Putem aplikacije omogućeno je da različite vodove provjes izračunava u kratkom vremenskom roku. Jednostavno je promijeniti parametre stupova i vodiča. Izrađena je u Java programskom jeziku i podrazmjeva korisničko sučelje u koje se unose parametri potrebni za proračun provjesa nadzemnih vodova i dio koda koji koristi algoritam za izračun.

3.1 Korisničko sučelje

Korisničko sučelje je rađeno u android studiju te omogućuje korisniku upravljanje aplikacijom. Aplikacija nudi odgovarajuća polja za unos odgovarajućih parametara. Pritiskom na gumb „IZRAČUNAJ“ ostvaruje se algoritam napisan u Java programskom jeziku koji vraća rezultat za izračun provjesa izoliranih energetske vodova. Korisničko sučelje se sastoji od EditText polja preko kojih se unose vrijednosti parametara i Buttona koji na pritisak vraća izračun provjesa za dane parametre.



Wisa vodiča

Kočeficijent toplinskog rastezanja

Faktor Norm. Dodatnog Tereta

Modul Elastičnosti

Normalno dozvoljeno naprezanje

Iznimno dozvoljeno naprezanje

Maksimalno naprezanje

Raspon stupova

Denivelacija

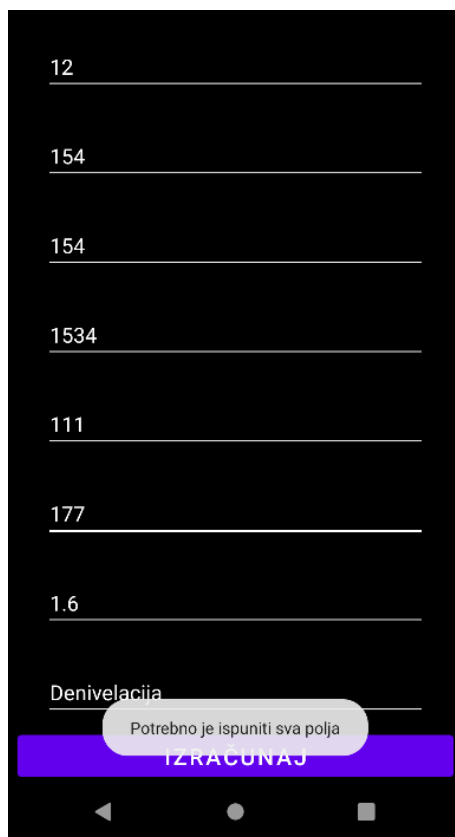
IZRAČUNAJ

Slika 3.1.1 Prikaz korisničkog sučelja

Kako bi rad sa aplikacijom bio jednostavniji postavljeni su uvjeti koji korisniku ne omogućuju da neko od odgovarajućih polja ostane ne ispunjeno. Slika 3.1.2 prikazuje metodu unutar aplikacije koja provjerava ispunjenost polja za unos parametara. Ukoliko sva polja nisu ispunjena, korisnik se prikazuje prigodna poruka kao na slici 3.1.3 i time mu se daje do znanja da je došlo do pogreške.

```
private Boolean checkInputFieldsAreEmpty() {  
    return presjekVodicaUnos.getText().toString().equals("")  
        || promjerVodicaUnos.getText().toString().equals("")  
        || masaVodicaUnos.getText().toString().equals("")  
        || koefToplinskogIstezanjaUnos.getText().toString().equals("")  
        || faktorNormDodatnogTeretaUnos.getText().toString().equals("")  
        || moduleElasticnostiUnos.getText().toString().equals("")  
        || normalnoDozvoljenoNaprezanjeUnos.getText().toString().equals("")  
        || iznimnoDozvoljenoNaprezanjeUnos.getText().toString().equals("")  
        || maksimalnoNaprezanjeUnos.getText().toString().equals("")  
        || rasponStupovaUnos.getText().toString().equals("")  
        || denivelacijaUnos.getText().toString().equals("");  
}
```

Slika 3.1.2 Uvjet da sva polja moraju biti ispunjena

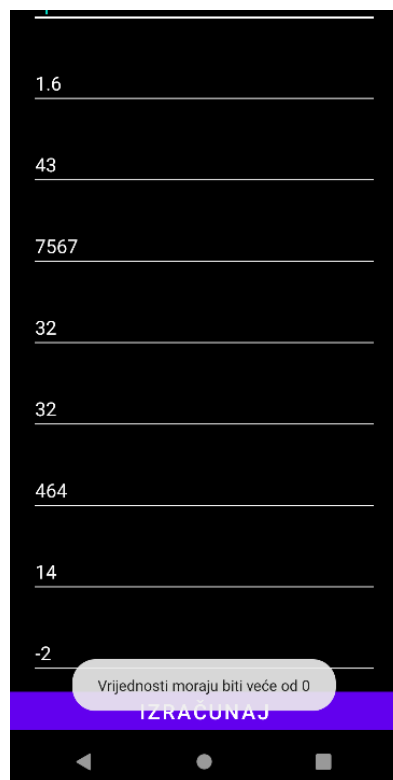


Slika 3.1.3 Korisničko sučelje uz neispunjene 1. uvjet

Drugi uvjet aplikacije sprječava korisnika da unosi vrijednosti manje ili jednake 0, dok vrijednost visinske razlike između stupova koja smije biti 0. Na slici 3.1.4 dan je kod koji prikazuje hvatanje takve greške i na slici 3.1.5 prigodnu poruku korisniku u slučaju pogreške.

```
try {
    Calculations calculations;
    calculations = new Calculations(
        Double.parseDouble(presjekVodicaUnos.getText().toString()),
        Double.parseDouble(promjerVodicaUnos.getText().toString()),
        Double.parseDouble(masaVodicaUnos.getText().toString()),
        Double.parseDouble(koefToplinskogIstezanjaUnos.getText().toString()),
        Double.parseDouble(faktorNormDodatnogTeretaUnos.getText().toString()),
        Double.parseDouble(moduleElasticnostiUnos.getText().toString()),
        Double.parseDouble(maksimalnoNaprezanjeUnos.getText().toString()),
        Double.parseDouble(rasponStupovaUnos.getText().toString()),
        Double.parseDouble(denivelacijaUnos.getText().toString())
    );
} catch (IllegalArgumentException exception) {
    Toast.makeText(context: MainActivity.this, text: "Vrijednosti moraju biti veće od 0",
        Toast.LENGTH_SHORT).show();
}
```

Slika 3.1.4 Prikaz koda koji hvata grešku



Slika 3.1.5 Prikaz Korisničkog sučelja u slučaju negativne vrijednosti

3.2 Povezivanje korisničkog sučelja i varijabli

XML jezik u kojem je napisan kod za korisničko sučelje se povezuje s glavnim djelom aplikacije koji je napisan u java programskom jeziku. Svako polje za unos dobiva svoju varijablu unutar aplikacije te omogućuje ispravan rad aplikacije.

```
private void initEditTexts() {
    presjekVodicaUnos = (EditText) findViewById(R.id.et_PresjekVodiča);
    promjerVodicaUnos = (EditText) findViewById(R.id.et_PromjerVodiča);
    masaVodicaUnos = (EditText) findViewById(R.id.et_LinijskaMasaVodiča);
    koefToplinskogIstezanjaUnos = (EditText)
        findViewById(R.id.et_koefToplinskogIstezanja);
    faktorNormDodatnogTeretaUnos = (EditText)
        findViewById(R.id.et_faktorNormDodatnogTereta);
    moduleElasticnostiUnos = (EditText) findViewById(R.id.et_ModulElastičnosti);
    normalnoDozvoljenoNaprezanjeUnos = (EditText)
        findViewById(R.id.et_NormDozvoljenoNaprezanje);
    iznimnoDozvoljenoNaprezanjeUnos = (EditText)
        findViewById(R.id.et_IznimnoDozvoljenoNaprezanje);
    maksimalnoNaprezanjeUnos = (EditText)
        findViewById(R.id.et_MaksimalnoNaprezanje);
    rasponStupovaUnos = (EditText) findViewById(R.id.et_RasponStupova);
    denivelacijaUnos = (EditText) findViewById(R.id.et_Denivelacija);
}
```

Slika 3.2.1 Prikaz povezanog varijabli i korisničkog sučelja

3.3 Java dio

Java dio sadrži sav kod koji je potreban za izračune provjesa nadzemnih vodova. Sve formule od (2-1) do (2-13) su iskorištene i mogu se vidjeti na slici 3.3.1 , 3.3.2 i 3.3.3 te kako izgledaju unutar java koda.

```

double temperature = -5.0;
double secCo1 = -(strain - (((Math.pow(idealSpan, 2) / 24) *
    (Math.pow(reducedConductorWeight, 2)
    / Math.pow(strain, 2)) - (koefIstezanja * (temperature - (-20)))) *
    modulElasticnosti));
double secCo2 = -(strain - (((Math.pow(idealSpan, 2) / 24) *
    (Math.pow(reducedConductorWeight, 2)
    / Math.pow(strain, 2)) - (koefIstezanja * (temperature - (-5)))) *
    modulElasticnosti));
double secCo3 = -(strain - (((Math.pow(idealSpan, 2) / 24) *
    (Math.pow(reducedConductorWeight, 2)
    / Math.pow(strain, 2)) - (koefIstezanja * (temperature - 40))) *
    modulElasticnosti));
double thirdCo1 = -((Math.pow(idealSpan, 2) * Math.pow(reducedWeight, 2)) / 24) *
    modulElasticnosti;
double thirdCo2 = -((Math.pow(idealSpan, 2) * Math.pow(reducedConductorWeight,
    2)) / 24) * modulElasticnosti;
double thirdCo3 = -((Math.pow(idealSpan, 2) * Math.pow(reducedWeight, 2)) / 24) *
    modulElasticnosti;
Cubic cub = new Cubic();
cub.solve(a: 1.0, secCo1, c: 0.0, thirdCo1);
double strainRaw1 = cub.x1;
Cubic cub2 = new Cubic();
cub2.solve(a: 1.0, secCo2, c: 0.0, thirdCo2);
double strainRaw2 = cub2.x1;
Cubic cub3 = new Cubic();
cub3.solve(a: 1.0, secCo3, c: 0.0, thirdCo3);
double strainRaw3 = cub3.x1;
double strain20 = strainRaw1 * ((Math.pow(directDistance, 2) / idealSpan) /
    ((Math.pow(directDistance, 3) / Math.pow(idealSpan, 2))));
provjes20 = ((Math.pow(idealSpan, 2) * reducedWeight) / (8 * strain20)) *
    (directDistance / idealSpan);
double strain5 = strainRaw2 * ((Math.pow(directDistance, 2) / idealSpan) /
    ((Math.pow(directDistance, 3) / Math.pow(idealSpan, 2))));
provjes5 = ((Math.pow(idealSpan, 2) * reducedConductorWeight) / (8 * strain5)) *
    (directDistance / idealSpan);
double strain40 = strainRaw3 * ((Math.pow(directDistance, 2) / idealSpan) /
    ((Math.pow(directDistance, 3) / Math.pow(idealSpan, 2))));
provjes40 = ((Math.pow(idealSpan, 2) * reducedWeight) / (8 * strain40)) *
    (directDistance / idealSpan);

```

Slika 3.3.1 kod u slučaju idealni raspon ima veću vrijednost od kritičnog raspona

Na slici 3.3.1 je kod u slučaju idealni raspon ima veću vrijednost od kritičnog raspona, dok na slici 3.3.2 je kod u slučaju idealni ima manju vrijednost od kritičnog raspona.

```

double temperature = -20.0;
double secCo1 = -(strain - (((Math.pow(idealSpan, 2) / 24) *
    (Math.pow(reducedConductorWeight, 2) / Math.pow(strain, 2)) - (koefIstezanja *
    (temperature - (-
        20)))) * modulElasticnosti));
double secCo2 = -(strain - (((Math.pow(idealSpan, 2) / 24) *
    (Math.pow(reducedConductorWeight, 2) / Math.pow(strain, 2)) - (koefIstezanja *
    (temperature - (-
        5)))) * modulElasticnosti));
double secCo3 = -(strain - (((Math.pow(idealSpan, 2) / 24) *
    (Math.pow(reducedConductorWeight, 2) / Math.pow(strain, 2)) - (koefIstezanja *
    (temperature -
        40)))) * modulElasticnosti));
double thirdCo1 = -((Math.pow(idealSpan, 2) * Math.pow(reducedWeight, 2)) / 24) *
    modulElasticnosti;
double thirdCo2 = -((Math.pow(idealSpan, 2) * Math.pow(reducedConductorWeight,
    2)) /
    24) * modulElasticnosti;
double thirdCo3 = -((Math.pow(idealSpan, 2) * Math.pow(reducedWeight, 2)) / 24) *
    modulElasticnosti;
Cubic cub = new Cubic();
cub.solve(a: 1.0, secCo1, c: 0.0, thirdCo1);
double strainRaw1 = cub.x1;
Cubic cub2 = new Cubic();
cub2.solve(a: 1.0, secCo2, c: 0.0, thirdCo2);
double strainRaw2 = cub2.x1;
Cubic cub3 = new Cubic();
cub3.solve(a: 1.0, secCo3, c: 0.0, thirdCo3);
double strainRaw3 = cub3.x1;
double strain20 = strainRaw1 * ((Math.pow(directDistance, 2) / idealSpan) /
    ((Math.pow(directDistance, 3) / Math.pow(idealSpan, 2))));
provjes20 = ((Math.pow(idealSpan, 2) * reducedWeight) / (8 * strain20)) *
    (directDistance / idealSpan);
double strain5 = strainRaw2 * ((Math.pow(directDistance, 2) / idealSpan) /
    ((Math.pow(directDistance, 3) / Math.pow(idealSpan, 2))));
provjes5 = ((Math.pow(idealSpan, 2) * reducedConductorWeight) / (8 * strain5)) *
    (directDistance / idealSpan);
double strain40 = strainRaw3 * ((Math.pow(directDistance, 2) / idealSpan) /
    ((Math.pow(directDistance, 3) / Math.pow(idealSpan, 2))));
provjes40 = ((Math.pow(idealSpan, 2) * reducedWeight) / (8 * strain40)) *
    (directDistance / idealSpan);

```

Slika 3.3.2Kod u slučaju idealni raspon ima veću vrijednost kritičnog raspona


```

double strain20 = strainRaw1 * ((Math.pow(directDistance, 2) / idealSpan) /
    ((Math.pow(directDistance, 3) / Math.pow(idealSpan, 2))));
provjes20 = ((Math.pow(idealSpan, 2) * reducedWeight) / (8 * strain20)) *
    (directDistance / idealSpan);
double strain5 = strainRaw2 * ((Math.pow(directDistance, 2) / idealSpan) /
    ((Math.pow(directDistance, 3) / Math.pow(idealSpan, 2))));
provjes5 = ((Math.pow(idealSpan, 2) * reducedConductorWeight) / (8 * strain5)) *
    (directDistance / idealSpan);
double strain40 = strainRaw3 * ((Math.pow(directDistance, 2) / idealSpan) /
    ((Math.pow(directDistance, 3) / Math.pow(idealSpan, 2))));
provjes40 = ((Math.pow(idealSpan, 2) * reducedWeight) / (8 * strain40)) *
    (directDistance / idealSpan);

```

Slika 3.3.3 Kranja formula za izračun

Također sadrži klasu Cubic slika 3.3.4 koja omogućuje rješavanje kubnih jednadžbi te tako omogućuje ispravan rad aplikacije.

```
double denom = a;
a = b/denom;
b = c/denom;
c = d/denom;

double a_over_3 = a / 3.0;
double Q = (3*b - a*a) / 9.0;
double Q_CUBE = Q*Q*Q;
double R = (9*a*b - 27*c - 2*a*a*a) / 54.0;
double R_SQR = R*R;
double D = Q_CUBE + R_SQR;

if (D < 0.0)
{
    nRoots = 3;
    double theta = Math.acos (R / Math.sqrt (-Q_CUBE));
    double SQRT_Q = Math.sqrt (-Q);
    x1 = 2.0 * SQRT_Q * Math.cos (theta/3.0) - a_over_3;
    x2 = 2.0 * SQRT_Q * Math.cos ((theta+TWO_PI)/3.0) - a_over_3;
    x3 = 2.0 * SQRT_Q * Math.cos ((theta+FOUR_PI)/3.0) - a_over_3;
    sortRoots();
}
else if (D > 0.0)
{
    nRoots = 1;
    double SQRT_D = Math.sqrt (D);
    double S = Math.cbrt (R + SQRT_D);
    double T = Math.cbrt (R - SQRT_D);
    x1 = (S + T) - a_over_3;
    x2 = Double.NaN;
    x3 = Double.NaN;
}
else
{
    nRoots = 3;
    double CBRT_R = Math.cbrt (R);
    x1 = 2*CBRT_R - a_over_3;
    x2 = x3 = CBRT_R - a_over_3;
    sortRoots();
}
```

Slika 3.3.4 kod klase Cubic

```
private void sortRoots()
{
    if (x1 < x2)
    {
        double tmp = x1; x1 = x2; x2 = tmp;
    }
    if (x2 < x3)
    {
        double tmp = x2; x2 = x3; x3 = tmp;
    }
    if (x1 < x2)
    {
        double tmp = x1; x1 = x2; x2 = tmp;
    }
}}
```

Slika 1.3.5 pomoćna metoda sortRoots()

4. USPOREDBA REZULTATA

Cilj ove usporedbe je prikazati postoji li razlika između izračuna ručno i izračuna preko aplikacije. Kod ručnog izračuna provjesa imat ćemo prikazan svaki korak izračuna, formule i parametre. Kod aplikacije je rješenje koje smo dobili upisivanje različitih parametara i pritisak gumba „IZRAČUNAJ“. Očekivanja su da će razlike u rezultatima biti minimalni jer u suprotnome aplikacija neće imati smisla. Do razlike u rezultatima bi trebalo doći zbog toga što aplikacija vraća rješenje sa dvije decimale. Smisao usporedbe je dokazati ispravnost aplikacije. Za ručno računanje bit će potrebne formule navedene u dijelovima ranije. Iz usporedbe bi trebalo poprilično biti jasno da postoji ogromna razlika u vremenu kod računanja provjesa na ručni način i kad aplikacija to napravi u sekundi.

4.1 Ručni izračun provjesa

Tablica 1. prikazuje vrijednosti vodiča korištenog u računanju provjesa.

Tablica 4.1 vrijednosti karakteristika vodiča

Promjer vodiča - d [mm]	22,4
Presjek vodiča - A [mm ²]	297,8
Uzdužna masa - m [kg/m]	1,003
Modul elastičnosti - E [N/mm ²]	68000
Koeficijent linearnog toplinskog rastezanja - β [1/°C]	$19,4 \cdot 10^{-6}$
Normalno dozvoljeno naprezanje - σ_d [N/mm ²]	100
Iznimno dozvoljeno naprezanje - σ_i [N/mm ²]	200
Maksimalno dozvoljeno naprezanje – σ_{max} [N/mm ²]	100

Potrebne su nam i vrijednosti raspona stupova i denivelacije(h12)

- $h_{12}=10\text{m}$
- $a=200\text{m}$

Prvo je potrebno izračunati vlastitu težinu vodiča:

$$G_0 = g * m_1 = 9,81 * 1,003 = 9,83943 \text{ N M}$$

Zatim se računa reducirana težina

$$g_0 = G_0/A = 9,83943/297,8 = 0,03304 \text{ N/(m * mm}^2\text{)}$$

Nakon toga izračun dodatnog opterećenja:

$$G_{10} = 0.18 * \sqrt{d} * g = 0.18 * \sqrt{22,4} * 9,81 = 8,3573 \text{ N/m}$$

Zatim izračunavanje stvarnog dodatnog opterećenja:

$$G_1 = k * G_{10} = 1 * 8,3573 = \text{N/m}$$

Potom izračun dodatnog opterećenja, odrediti reduciranu težinu zaleđenog vodiča:

$$g_z = (G_1 + G_0)/A = (9,83943 + 8,3573)/297,8 = 0,061104 \text{ N/(m * mm}^2\text{)}$$

U slučaju jednog raspona a , taj raspon se uzima kao idealni:

$$a_{idealno} = a = 200\text{m}$$

Potom slijedi računanje kritičkog raspona:

$$a_k = \sigma_{max} * \sqrt{\frac{360 * \beta}{g_z^2 - g_0^2}} = 100 * \sqrt{\frac{360 * 19.4 * 10^{-6}}{0.061104^2 - 0.03304^2}}$$

Provjerava se uvjet $a_{idealno} > a$, pošto je uvjet zadovoljen slijedi:

- $\sigma_1 = \sigma_{max} = 100 \text{ N/(m*mm}^2\text{)}$
- $g_1 = g_z = 0,061104 \text{ N/(m*mm}^2\text{)}$
- $t_1 = -5^\circ \text{C}$

Uz denivelaciju 10m računa se nadomjesno naprezanje:

$$\bar{\sigma}_1 = \sigma_1 * \frac{\sum_{i=1}^n \frac{a'_i{}^3}{a_i^2}}{\sum_{i=1}^n \frac{a'_i}{a_i}} = 100 * 1.00125 = 100.125 \frac{\text{N}}{\text{mm}^2}$$

Zatim slijedi određivanje stvarnog nadomjesnog naprežanja:

$$1) \theta_2 = -5^\circ \text{ C}$$

$$g_2 = g_z = 0.061104 \frac{N}{m * mm^2}$$

$$\frac{\bar{\sigma}_1 - \bar{\sigma}_2}{E} + \beta * (\theta_1 - \theta_2) = \frac{a_{idealno}^2}{24} * \left(\frac{g_1^2}{\bar{\sigma}_1^2} - \frac{g_2^2}{\bar{\sigma}_2^2} \right)$$

$$\frac{100,125 - \bar{\sigma}_2}{68000} + 19.4 * 10^{-6}(-5 - (-5)) = \frac{200^2}{24} * \left(\frac{0.061104^2}{100.125^2} - \frac{0.061104^2}{\bar{\sigma}_2^2} \right)$$

Nakon sređivanja slijedi kubna jednačba

$$\bar{\sigma}_2^3 + 57.9154\bar{\sigma}_2^2 + 423154.44 = 0$$

$$\bar{\sigma}_2 = 100.1250293 \frac{N}{mm^2}$$

$$\text{Stvarno nadomjesno naprežanje: } \sigma_2 = \bar{\sigma}_2 * \frac{\sum_{i=1}^n \frac{a_i'^2}{a_i^2}}{\sum_{i=1}^n \frac{a_i'^3}{a_i}} = 100.2501856 \frac{N}{mm^2}$$

$$2) \theta_2 = -20^\circ \text{ C}$$

$$g_2 = g_0 = 0.03304 \frac{N}{m * mm^2}$$

$$\frac{100,125 - \bar{\sigma}_2}{68000} + 19.4 * 10^{-6}(-5 - (-20)) = \frac{200^2}{24} * \left(\frac{0.061104^2}{100.125^2} - \frac{0.061104^2}{\bar{\sigma}_2^2} \right)$$

Nakon sređivanja slijedi kubna jednačba

$$\bar{\sigma}_2^3 + 77.7033396\bar{\sigma}_2^2 + 123719,2 = 0$$

$$\bar{\sigma}_2 = 92.24340733 \frac{N}{mm^2}$$

$$\text{Stvarno nadomjesno naprežanje: } \sigma_2 = \bar{\sigma}_2 * \frac{\sum_{i=1}^n \frac{a_i'^2}{a_i^2}}{\sum_{i=1}^n \frac{a_i'^3}{a_i}} = 92.1282 \frac{N}{mm^2}$$

$$3) \theta_2 = 40^\circ \text{ C}$$

$$g_2 = g_0 = 0.03304 \frac{N}{m * mm^2}$$

$$\frac{100,125 - \bar{\sigma}_2}{68000} + 19.4 * 10^{-6}(-5 - (40)) = \frac{200^2}{24} * \left(\frac{0.061104^2}{100.125^2} - \frac{0.061104^2}{\bar{\sigma}_2^2} \right)$$

Nakon sređivanja slijedi kubna jednačba

$$\bar{\sigma}_2^3 - 1.4486\bar{\sigma}_2^2 + 123719,38 = 0$$

$$\bar{\sigma}_2 = 49.35045 \frac{N}{mm^2}$$

Stvarno nadomjesno naprezanje: $\sigma_2 = \bar{\sigma}_2 * \frac{\sum_{i=1}^n \frac{a_i'^2}{a_i^2}}{\sum_{i=1}^n \frac{a_i'^3}{a_i}} = 49.2883895 \frac{N}{mm^2}$

Dobivši rezultate za sva stvarno nadomjesna rastezanja računamo provjese:

1) Provjes nadzemnog vodiča pri temperaturi -5° C

$$f' = f * \frac{a'}{a} = \frac{a^2 * g_2}{8 * \sigma_2} * \frac{a'}{a} = 3.05138m$$

2) Provjes nadzemnog vodiča pri temperaturi -20° C

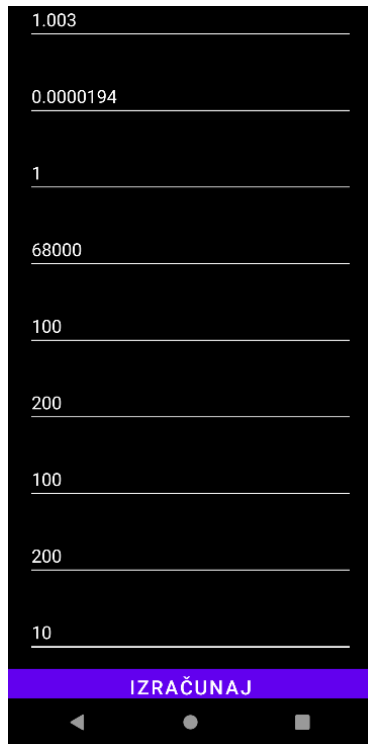
$$f' = f * \frac{a'}{a} = \frac{a^2 * g_2}{8 * \sigma_2} * \frac{a'}{a} = 1.795396m$$

3) Provjes nadzemnog vodiča pri temperaturi 40° C

$$f' = f * \frac{a'}{a} = \frac{a^2 * g_2}{8 * \sigma_2} * \frac{a'}{a} = 3.3558611m$$

4.2 Računanje pomoću aplikacije

Pomoću aplikacije računanje je jednostavno, samo se mora pravilno upisati parametre i nakon pritiska gumba „IZRAČUNAJ“ korisnik dobiva traženo rješenje. Slika 4.2.1 uneseni su svi parametri identični onima koji su korišteni kod ručnog izračuna.



Slika 4.2.1 Unos parametara u aplikaciju

Rješenje se prikazuje u novom prozoru, te imamo i prikaz prethodnik izračuna. Na slici 4.2.2 izračun 1 rješenje unesenih parametara.

Izračun 3: Provjesi za temperature(-5,-20-40) pri denivelaciji(10 m) i raspon stupova(160 m) su: 2.52, 1.8, 3.75



Izračun 2: Provjesi za temperature(-5,-20-40) pri denivelaciji(15 m) i raspon stupova(150 m) su: 1.97, 1.09, 2.32



Izračun 1: Provjesi za temperature(-5,-20-40) pri denivelaciji(10 m) i raspon stupova(200 m) su: 3.06, 1.8, 3.36



Slika 2.2.2 Rezultati aplikacije

Iz tablice je vidljivo da su razlike zanemarive u praktičnom svijetu, dok brzina izračuna koja se postiže je znatno brža od ručnog izračuna.

Tablica 4.2 Rezultati svih izračuna

Temperatura	Izračun Aplikacije	Ručni izračun
-5° C	3.06m	3.051m
-20° C	1.8m	1.795m
40° C	3.36m	3.356m

5. ZAKLJUČAK

Svakom kućanstvu je potrebna električna energija, a električna energija dolazi do svakog kućanstva uz električne vodove za prijenos električne energije. Postavljanje električnih vodova mora biti precizno zato što u suprotnom riskiramo sigurnost onih u blizini. Izračun provjesa je složen i ako se izvodi ručno može uzeti dosta vremena zato je napravljena ova aplikacija koja ubrzava izračun. Aplikacija je napravljena u programskom jeziku java pomoću razvojnog okruženja Android Studio. Aplikacija sadrži kompleksne algoritme koji brzo dolaze do rezultata te tako omogućuju uštedu vremena potrebnog za izračun provjesa. Rezultati dobiveni aplikacijom i ručnim izračunom se razlikuju u milimetrima, što znači da u ovom slučaju aplikacija je pouzdana i može se koristiti u svrhu izračuna provjesa. Do pogreške u rezultatima dolazi zato što aplikacija nije napravljena precizno kao ručni izračun, te su razlike minimalne. Aplikacija se može nadograditi tako da se dodaju SI jedinice uz parametre koji se unose, forma unosa se može poboljšati tako da se i kod upisa brojeva parametara vidi o kojem parametru zapravo radi. Moguće je poboljšati i računski dio aplikacije kako bi aplikacija bila još preciznija.

LITERATURA

[1] Pravilnik o tehničkim normativima za izgradnju nadzemnih elektroenergetskih vodova napona 1 kV do 400 kV, «Službeni list» broj 65/88, «Narodne novine» broj 53/91 – Zakon o standardizaciji 24/97, 1997.

[2] Brkić, N.: Mehanički proračun vodiča, Sveučilište u Rijeci, Tehnički fakultet, Rijeka 2015.

[3] Alan Kaminsky, Klasa cubic, dostupno na:

<https://www.cs.rit.edu/~ark/pj/doc/edu/rit/numeric/Cubic.html>, (19.7.2022.)

[4] Weisstein, Eric W., "Cubic Formula.", MathWorld--A Wolfram Web Resource, dostupno na: <https://mathworld.wolfram.com/CubicFormula.html>, (19.7.2022.)

[6] Neumark, S: Solution of Cubic and Quartic Equations, 1965.

[7] Java(programski jezik), [https://hr.wikipedia.org/wiki/Java_\(programski_jezik\)](https://hr.wikipedia.org/wiki/Java_(programski_jezik)) , (19.7.2022.)

[8] Razvojno okruženje „Android Studio“, https://en.wikipedia.org/wiki/Android_Studio, (19.7.2022.)

[9]“ Električni vodič“, https://hr.wikipedia.org/wiki/Elektri%C4%8Dni_vodi%C4%8D

[10] mehanički proračun vodiča Brkic_Nikica,

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjv3rCDm6H6AhXixQIHHWPkBdgQFnoECAsQAQ&url=http%3A%2F%2Fold.riteh.hr%2Fnast%2Fobrane%2Fstrucni_el%2FRadovi_17092015%2FBrkic_Nikica.pdf&usg=AOvVaw3e0snT8znN8ckRb_C3TBrT

[11] Matematički model konstrukcije:

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjv3rCDm6H6AhXixQIHHWPkBdgQFnoECAwQAQ&url=https%3A%2F%2Fwww.grad.unizg.hr%2F_download%2Frepository%2FNMK-skript.pdf&usg=AOvVaw2b_e0Ssy1x9et5IuoAN0dM

SAŽETAK

Cilj ovog završnog rada je izrada Android aplikacije za izračunavanje provjesa izoliranih energetskih vodova na brži i jednostavniji način od ručnog računanja provjesa. Aplikacija je namijenjena inženjerima i studentima koji će to tek postati kako bi mogli brže doći do željenih podataka ili pak kako bi provjerili svoje izračune. Aplikacija je rađena u razvojnom okruženju Android Studio, a kao programski jezik je korištena Java. Aplikacija uz manja odstupanja prikazuje željene podatke.

Ključne riječi: Android, Android Studio, Java, aplikacija, vodovi, provjes izoliranih energetskih.

ABSTRACT

The purpose of this thesis is the creation of an Android application for calculating the sag of overhead powerlines in a faster and simpler way than doing it manually. The application is meant for engineers and students who are yet to become one for the purpose of getting to wanted data faster or to verify their own calculations. The application was created using the Android Studio development environment with Java as the programming language. The application shows the expected data with slight deviations.

Key words: Android, Android Studio, Java, application, overhead powerlines, overhead powerline sag

ŽIVOTOPIS

Luka Vukadin rođen je 30.6.1999. u Đakovu. U rodnome gradu pohađao je i završio Osnovnu školu Vladimir Nazor. Nakon što je svaki razred završio odličnim uspjehom, upisuje matematičko-prirodoslovnu gimnaziju Antun Gustav Matoš u Đakovu. Po završetku srednjoškolskoga obrazovanja 2018. godine upisuje preddiplomski sveučilišni studij na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, smjer računarstvo. Uz hrvatski jezik govori i dobro poznaje engleski. Tijekom fakultetskoga školovanja proširio je znanja iz programskih jezika C-a i HTML-a te stilskoga jezika CSS-a, a stekao je osnovna znanja iz C++ i Java programskoga jezika.

PRILOZI

CD/DVD medij koji sadrži:

-Projekt mobilne aplikacije

-Završni rad u pdf. formatu