Aplikacija za interpretaciju rezultata prema normi HRN EN60599

Pandurić, Mario

Master's thesis / Diplomski rad

2024

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:200:869273

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-21**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

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

Sveučilišni studij

Aplikacija za interpretaciju rezultata prema normi HRN EN60599

Diplomski rad

Mario Pandurić

Osijek, 2024.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za ocjenu diplomskog rada na sveučilišnom diplomskom studiju

Ocjena diplomskog rada na sveučilišnom diplomskom studiju

Ime i prezime pristupnika:	Mario Pandurić
Studij, smjer:	Sveučilišni diplomski studij Računarstvo
Mat. br. pristupnika, god.	D1314R, 07.10.2022.
JMBAG:	0165079118
Mentor:	izv. prof. dr. sc. Alfonzo Baumgartner
Sumentor:	
Sumentor iz tvrtke:	Veljko Špica
Predsjednik Povjerenstva:	izv. prof. dr. sc. Tomislav Keser
Član Povjerenstva 1:	izv. prof. dr. sc. Alfonzo Baumgartner
Član Povjerenstva 2:	doc. dr. sc. Tomislav Galba
Naslov diplomskog rada:	Aplikacija za interpretaciju rezultata prema normi HRN EN60599
Znanstvena grana diplomskog rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak diplomskog rada:	Potrebno je izraditi aplikaciju koja će uvrstiti koncentracije plinova iz kromatograma DGA analize, Uvrstiti sliku kromatograma, provjeriti min/max vrijednosti, interpretirati rezultate prema normi HRN EN60599 te izraditi izvješće sa zaključkom.

Datum ocjene pismenog dijela diplomskog rada od strane mentora:	06.09.2024.
Ocjena pismenog dijela diplomskog rada od strane mentora:	Izvrstan (5)
Datum obrane diplomskog rada:	12.09.2024.
Ocjena usmenog dijela diplomskog rada (obrane):	Izvrstan (5)
Ukupna ocjena diplomskog rada:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije diplomskog rada čime je pristupnik završio sveučilišni diplomski studij:	17.09.2024.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O IZVORNOSTI RADA

Osijek, 17.09.2024.

Ime i prezime Pristupnika:	Mario Pandurić
Studij:	Sveučilišni diplomski studij Računarstvo
Mat. br. Pristupnika, godina upisa:	D1314R, 07.10.2022.
Turnitin podudaranje [%]:	10

Ovom izjavom izjavljujem da je rad pod nazivom**: Aplikacija za interpretaciju rezultata prema normi HRN EN60599**

izrađen pod vodstvom mentora izv. prof. dr. sc. Alfonzo Baumgartner

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 pristupnika:

Sadržaj

1. UVOD 1
1.1. Zadatak diplomskog rada 1
2. TEORIJSKI UVOD U TRANSFORMATORE 2
2.1. Transformator 2
2.2. Transformatorsko ulje 3
2.3. Analiza otopljenog plina 4
2.3.1. Interpretacija rezultata DGA analize metodom Duvalovog trokuta 4
2.3.2. Interpretacija rezultata DGA analize metodom Duvalovog peterokuta 5
3. KORIŠTENE TEHNOLOGIJE 7
3.1. Visual Studio Code 7
3.2. Python programski jezik 8
3.2.1. Tkinter biblioteka 8
3.2.2. Matplotlib biblioteka 9
3.2.3. ReportLab biblioteka 9
3.2.4. PyWin32 biblioteka 9
3.2.5. Pillow biblioteka 10
3.2.6. Shapely biblioteka 10
3.2.7. PyMuPDF biblioteka 10
3.2.8. Auto Py to Exe 10
3.3. CSV datoteka 11
3.4. PDF datoteka 12
4. PREGLED POSTOJEĆIH RJEŠENJA 13
5. PROGRAMSKO RJEŠENJE 15
5.1. Opis funkcionalnosti programskog rješenja 15
5.2. Razvoj programskog rješenja 16
5.2.1. Instalacija i priprema okoline 16

5.2.2. Pisanje programskog koda 16
5.2.3. Izgradnja aplikacije 21
5.3. Testiranje programskog rješenja 23
6. ZAKLJUČAK 28
LITERATURA 29
SAŽETAK 31
ABSTRACT 32
Application for interpretation of results according to HRN EN60599 32

1. UVOD

Transformatori su najvažniji uređaji u distribuciji električne energije. Koriste kako bi se vrijednost napona dovedena na primarnoj strani povećala ili snizila kao izlaz na sekundarnoj strani. Zbog toga nam je vrlo važno njihovo održavanje i ispravnost. Jedna od metoda u održavanja transformatora je analiza otopljenog plina ulja transformatora. Prekomjerna ili smanjena koncentracija određenog plina (ili plinova) može ukazivati na potencijalni kvar transformatora. Koncentracije plinova su definirane normom HRN EN 60599. U ovom diplomskom radu bit će kreirana desktop aplikacija za interpretaciju rezultata DGA analize transformatorskog ulja. Aplikacija će biti pisana u programskom jeziku Python.

Nakon uvoda, u drugom poglavlju je opisan transformator kao uređaj, kako funkcionira, neke vrste i čemu služi transformator. Također, u ovom poglavlju je opisano transformatorsko ulje te zašto je ono jedan od najbitnijih elemenata transformatora. Objašnjeno je što je DGA analiza te neke metode interpretacije DGA analize. U idućem poglavlju su spomenute tehnologije za razvoj aplikacije. Opisan je Python programski jezik i neke njegove biblioteke korištene za razvoj aplikacije. Spomenut je Visual Studio Code uređivač teksta, što je CSV i što je PDF datoteke. U četvrtom poglavlju je spomenuta i opisana najsličnija dostupna aplikacija iz ovog rada. U petom poglavlju je opisan i prikazan razvoj i rad aplikacije.

1.1. Zadatak diplomskog rada

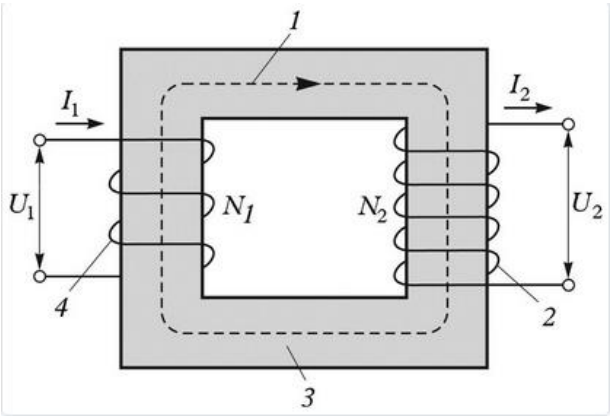
Potrebno je izraditi aplikaciju koja će uvrstiti koncentracije plinova iz kromatograma DGA analize, Uvrstiti sliku kromatograma, provjeriti min/max vrijednosti, interpretirati rezultate prema normi HRN EN60599 te izraditi izvješće sa zaključkom.

2. TEORIJSKI UVOD U TRANSFORMATORE

U ovom poglavlju će biti opisan transformator kao električni uređaj, bit će opisano što je transformatorsko ulje i zašto je ono važno te što je DGA analiza i kako interpretirati njezine rezultate.

2.1. Transformator

Transformator je električni uređaj bez pokretnih dijelova koji međuinduktivno povezuje dva električna kruga te jednu vrijednost izmjeničnog napona pretvara u drugu vrijednost izmjeničnog napona iste frekvencije. Također je važno naglasiti da je snaga oba strujnih krugova jednaka. Glavni dijelovi transformatora su: magnetska jezgra, i barem dva međusobno izolirana namota tzv. primar i sekundar. Primarni i sekundarni namotaji su često postavljeni jedan preko drugog ili jedan pokraj drugog kako bi se postigla što bolja međuinduktivna veza. Električna energija se prenosi elektromagnetskom indukcijom s primara na sekundar uz nepromjenjivu frekvenciju. Omjer napona na primarnoj i sekundarnoj strani jednak je omjeru broju namota na primarnoj i sekundarnoj strani. Transformatori se najčešće koriste za povećavanje ili sniženje vrijednosti napona u prijenosu električne energije. Postoje još specijalne vrste transformatora kao što su strujni mjerni transformatori koji se koriste za mjerenje velikih struja, izolacijski transformatori koji služe za odvajanje strujnih krugova, audiofrekvencijski transformatori koji omogućuje prijenos audiofrekvencijskih signala. S obzirom na vrstu izolacije transformatori mogu biti „suhi transformatori“ i uljni transformatori, a s obzirom na broj faza mogu biti jednofazni i trofazni. [1] Na slici 2.1. je prikazan jednostavan shematski prikaz transformatora.



Slika 2.1. Shematski prikaz transformatora (1. magnetski tok, 2. sekundarni namot, 3. mag. tok, 4. primarni namot) [1]

2.2. Transformatorsko ulje

Transformatorsko ulje je posebna vrsta ulja koja ima odlična izolacijska svojstva i stabilno je na visokim temperaturama [2]. Funkcije ulja u uljnim transformatorima su:

1. Izolacija – izolacijski materijal uronjen je u ulje što poboljšava čvrstoću izolacije te štiti od vlage.
2. Rasipanje topline – transformatorsko ulje ima veliku specifičnu toplinu pa se koristi kao rashladno sredstvo.
3. Suzbijanja luka - na prekidaču za ulje i prekidaču za regulaciju napona na opterećenju pojavit će se električni luk kod uključenja kontakata. S obzirom da transformatorsko ulje ima dobru toplinsku vodljivost i može odvojiti veliku količinu plina pod djelovanjem visoke temperature luka i stvoriti visoki tlak. [3]

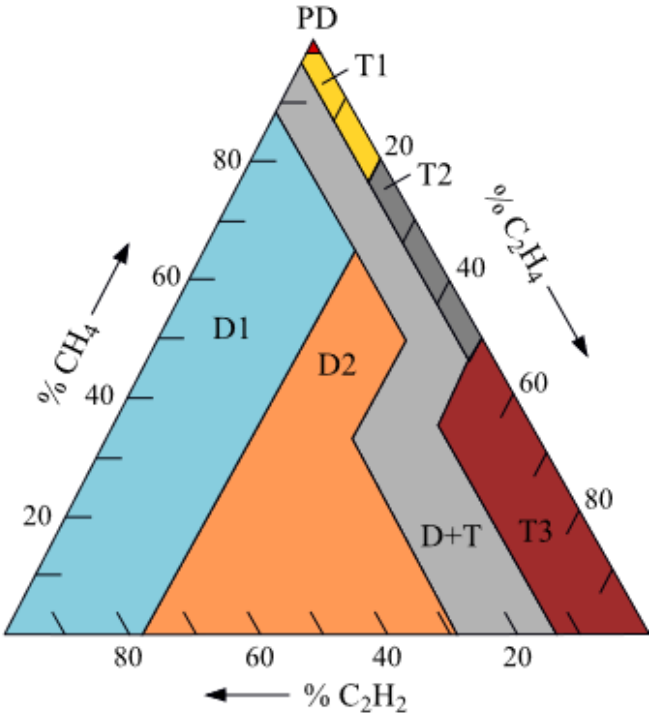
Da bi ulje bilo sigurno za korištenje treba zadovoljavati određena fizička, električna i kemijska svojstva. Pod fizička svojstva spadaju: količina vlage, međupovršinska napetost, plamište, viskoznost i stinište. U električna svojstva ubrajamo probojnu čvrstoću (omjer najveće vrijednosti napona kojeg dielektrik može izdržati, a da ne dođe do njegovo proboja i debljine toga dielektrika), otpor izolatora i faktor dielektričnih gubitaka. Oksidacijska stabilnost i djelovanje korozivnog sumpora na radne uvjete transformatora pripadaju kemijskim svojstvima transformatora. [4]

2.3. Analiza otopljenog plina

Analiza otopljenog plina (DGA analiza) je moćna analitička metoda korištena za praćenje stanja transformatora i ostale električne komponente koje koriste ulje [5]. Metoda analizira koncentracije otopljenih plinova u transformatorskom ulju koji nastaju zbog razgradnje ulja i izolacijskog papira kod djelovanja električnih i termičkih naprezanja. Prisustvo određenih plinova izvan dozvoljenih granica može ukazivati na neispravnost ili kvar transformatora kao što su pregrijavanje, proboj izolacije i sl. DGA analizom se promatra nekoliko kritičnih plinova kao primjerice vodik, ugljikov dioksid i monoksid, ugljikovodici kao što su metan, acetilen, etilen i etan. DGA analiza pomaže u detekciji kvarova prije nego što oni postanu katastrofalni te pomaže pri održavanju i popravci transformatora. Rezultati DGA analize se mogu interpretirati na različite načine. Neki od njih su: Doenenburgov omjer, IEC omjer, metoda ključnog plina te pomoću metoda Duvalovih trokuta i peterokuta.

2.3.1. Interpretacija rezultata DGA analize metodom Duvalovog trokuta

Metoda prikaza rezultata DGA analize Duvalovim trokutom je razvijena ranih 1970-ih godina od strane Michela Duvala. Metoda se temelji na promatranju triju: metana, etilena i acetilena. Stranice trokuta predstavljaju postotak koncentracije jednog od navedenih plinova. Trokut je podijeljen na 7 površina koje predstavljaju kvar. Nakon što se mjerenjem dobiju vrijednosti plinova izražene u ppm-u, izračunaju se njihovi relativni postotci koncentracije te se označi točka na stranici trokuta iz koje se crtaju pravci zadani određenim kutom. Pravac za metan se crta pod kutom od 0°, pravac za acetilen se crta pod kutom od 135° dok se za etilen pravac crta pod kutom od 225°. Kvar se iščitava iz područja u kojem se nalazi točka (ili površina) presjeka nacrtanih pravaca. Kvarovi koje je moguće iščitati iz trokuta su: niskoenergetsko pražnjenje (D1), visokoenergetsko pražnjenje (D2), toplinski kvar do 300°C (T1), toplinski kvar između 300°C i 700°C (T2), toplinski kvar veći od 700°C (T3), djelomično pražnjenje (PD) te kvar uzrokovan mješavinom toplinskim i električnim čimbenicima (DT). Duvalov trokut je prikazan na slici 2.2.



Slika 2.2. Prikaz Duvalovog trokuta [6]

2.3.2. Interpretacija rezultata DGA analize metodom Duvalovog peterokuta

Metoda Duvalovog peterokuta je slična metodi Duvalovog trokuta, ali se uz navedena 3 plina još promatraju vodik i etan. Za razliku od stranica trokuta koje su predstavljale relativnu koncentraciju plinova, ovdje relativne koncentracije plinova predstavljaju dužine koje spajaju središte i vrh pravilnog peterokuta. Kao i kod metode trokuta, potrebno je izračunati postotak relativnih koncentracija plinova te označiti točke na dužinama za pojedini plin. Nakon toga je potrebno izračunati koordinate centroida označenih točaka koristeći sljedeće formule:

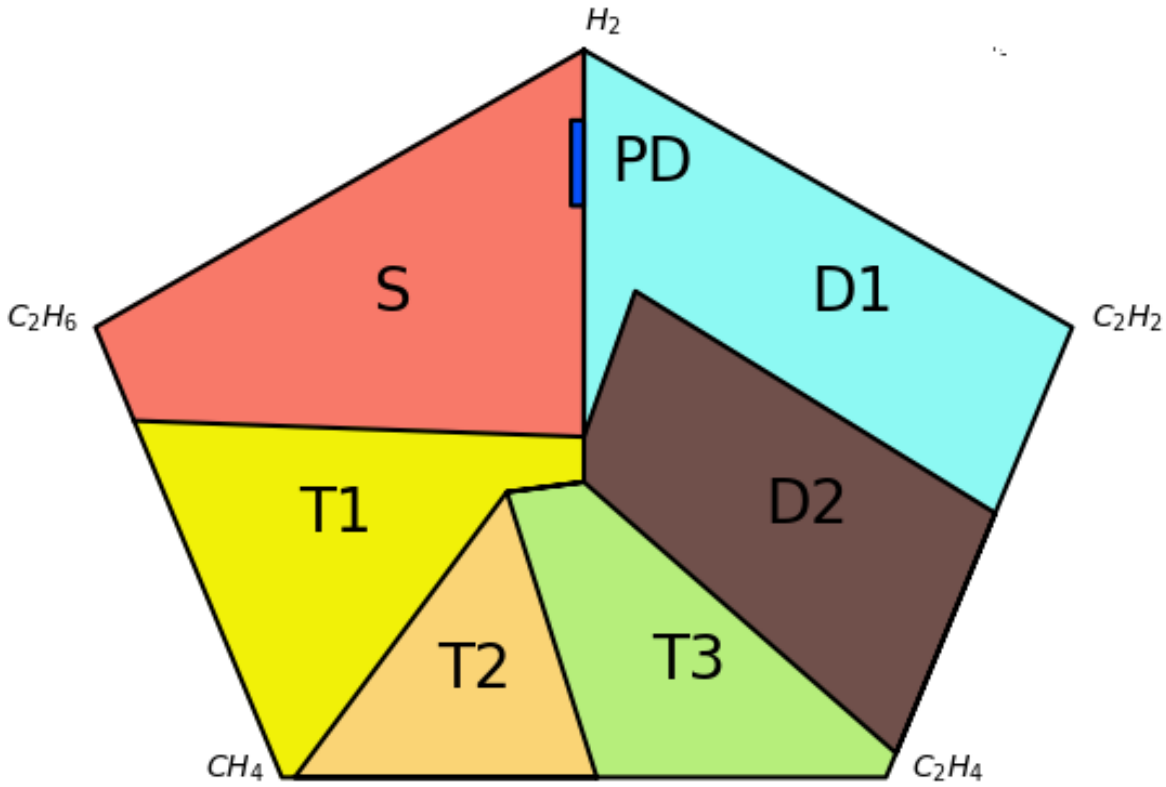
$$C_x = \frac{1}{6A}\sum_{i-0}^{n-1}(X_i + X_{i+1})(X_iY_{i+1} - X_{i+1}Y_i)$$

$$C_y = \frac{1}{6A}\sum_{i-0}^{n-1}(Y_i + Y_{i+1})(X_iY_{i+1} - X_{i+1}Y_i)$$

gdje je A površina novonastalog peterokuta. Površinu je moguće izračunati sljedećom formulom:

$$A = \frac{1}{2}\sum_{i-0}^{n-1}(X_iY_{i+1} - X_{i+1}Y_i)$$

Kvar se iščitava iz područja u kojem se nalaze koordinate centroida. Kao i Duvalov trokut, Duvalov peterokut ima sedam područja koji predstavljaju kvarove. Razlika među kvarovima je da Duvalov trokut ima područje koje prikazuje mješavinu toplinskog i električnog kvara (DT) dok peterokut ima područje koje označava pojavu stranih plinova u ulju transformatora (S). Na slici 2.3. prikazan je izgled Duvalov trokuta.



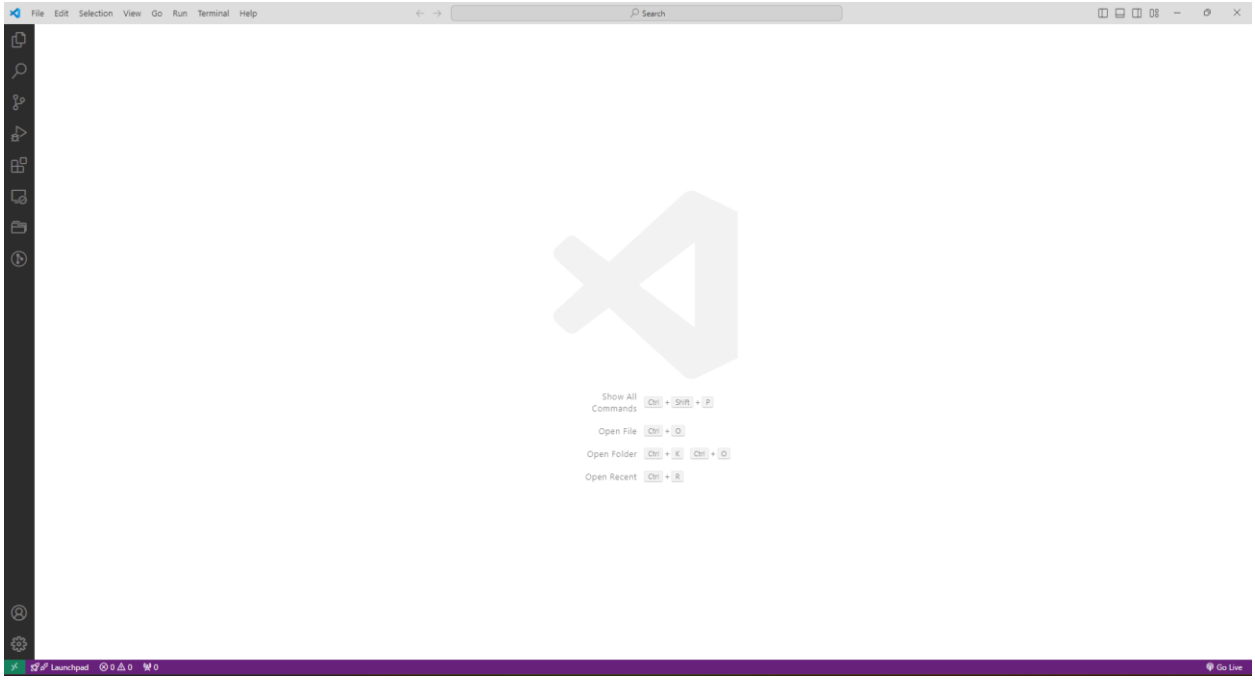
Slika 2.3. Prikaz Duvalovog peterokuta iz aplikacije

3. KORIŠTENE TEHNOLOGIJE

Kod razvoja softvera vrlo je važno odabrati odgovarajuće tehnologije i okruženje u kojem će se taj softver razvijati. Za ovu aplikaciju je zbog jednostavnosti i crtanja Duvalovog trokuta i peterokuta odabran programski jezik Python. Visual Studio Code je korišten kao uređivač koda.

3.1. Visual Studio Code

Visual Studio Code je besplatan, lagan i moćan uređivač izvornog koda koji se može koristiti na radnoj površini računala ili webu i dostupan je za Windows, macOS, Linux i Raspberry Pi operacijske sustave [7]. Razvio je ga Microsoft i prvi put je objavljen 29. travnja 2015. na Microsoft Build konferenciji. Izgrađen je na Electron radnom okviru koji se koristi za razvoj Node.js web aplikacija. Može se koristiti kao uređivač koda za mnogo različitih programski jezika, kao što su: C, C++, C#, Java, Python, JavaScript, Go, PHP itd. Visual Studio Code omogućava različita proširenja (eng. extensions) za dodavanje novih programskih jezika i različitih alata za podršku razvoja softvera. Visual Studio Code je prikazan na slici 3.1.



Slika 3.1. Prikaz početnog prozora aplikacije Visual Studio Code-a

3.2. Python programski jezik

Python je interpreterski, objektno-orijentirani programski jezik visoke razine sa dinamičkom semantikom [8]. Dobio je ime po britanskoj komičarskoj grupi Monty Python. Jedan je od najpopularnijih programskih jezika današnjice. Kreirao ga je Guido van Rossum 80-ih godina prošlog stoljeća. godine kao nasljednika za ABC programski jezik. Prvi put je izdan 1991. godine. Lako je čitljiv, varijablama se ne navodi tip nego se tip automatski dodjeljuje varijabli u ovisnosti o dodijeljenoj vrijednosti varijable. Memorija se oslobađa automatski tzv. *sakupljačem smeća* (garbage collector). Jedna od najvećih značajki Pythona je ta da osim standardnih biblioteka postoje biblioteke trećih strana sa širokim rasponom funkcionalnosti. Python se koristi za kreaciju web aplikacija, desktop aplikacija, automatizaciju, vizualizaciju podataka, koristi se u znanosti, strojnom učenju itd.

3.2.1. Tkinter biblioteka

Tkinter je standardna Python GUI biblioteka koja omogućava skup alata i widgeta za kreiranje desktop aplikacija s grafičkim sučeljem. Ime „Tkinter“ dolazi od „Tk interface“, referirajući se na Tk GUI alat na kojem se Tkinter bazira [9]. Biblioteka je uključena u samoj instalaciji Pythona za Microsoft Windows, Linux i MacOS operacijske sustave te ju nije potrebno dodatno instalirati. Osnove komponente Tkintera su widgeti, menadžeri geometrije i upravljanje događajima. Widgeti su komponente koje se nalaze na prozoru aplikacije te s njima korisnik ima neku vrstu interakcije. Primjer widgeta su gumbovi, polja za unos, *Combobox* itd. Menadžeri geometrije nam služi za pozicioniranje widgeta. Upravljanje događajima se odnosi na ponašanje i funkciju widgeta, npr. gumbova kad se klikne na njih, prijeđe pokazivačem miša preko njih i sl.

3.2.2. Matplotlib biblioteka

Matplotlib je Python biblioteka koja se koristi za kreaciju i crtanje statičkih, dinamičkih i interaktivnih vizualizacija [10]. Razvio ju je John D. Hunter 2003. godine. Matplotlib je besplatna i otvorenog je koda te održava ju zajednica developera. Matplotlib može generirati različite grafičke prikaze: linije, točke, stupčaste dijagrame, histograme itd. Te prikaze je moguće prilagođavati, mijenjati i uređivati. Također podržava dinamičke grafičke prikaze kroz različite widgete što omogućava korisnici da dinamički istražuju i pregledavaju podatke. Biblioteka je neovisna o platformi i moguće ju je pokrenuti na mnogim operacijskim sustavima uključujući Microsoft Windows, macOS i Linux.

3.2.3. ReportLab biblioteka

ReportLab biblioteka je biblioteka koja omogućava kreaciju PDF datoteka koristeći Python programski jezik. Moguće je kreirati grafove i vizualizirati podatke u različitim formatima [11]. Reportlab alat za PDF je besplatan i otvorenog koda za razliku od ReportLab Plusa koji znatno brže generira PDF datoteke te koristi Report Markup Language jezik koji je baziran na XML-u. Dio je standardnih Linux distribucija, dio je mnogih proizvoda i odabran je za pokretanje i izvoz značajki za Wikipedia-u. Biblioteka se sastoji od 3 glavna sloja: sloj za crtanje PDF stranica (canvas), grafovi i widgeti za kreiranje vizualizacije podataka te mehanizam za igradnju stranice *PLATYPUS* (Page Layout and TYPography Using Scripts) koji služi za izgradnju dokumenta od komponenata kao što su naslov, paragraf, tablice, font i slike [12].

3.2.4. PyWin32 biblioteka

PyWin32 je Python biblioteka koja omogućava pristup značajkama niske razine na Windows operacijskim sustavima, njihovim funkcijama i APIja [13]. Omogućava kontrolu i manipulaciju Windows elemenata pomoću Python skripti. Ovo se odnosi na procese, niti, datoteke, registre i usluge.

3.2.5. Pillow biblioteka

Pillow je Python biblioteka otvorenog koda posebno dizajnirana za obradu slike. Razvio ju je Fredrik Lundh 1995. godine pod nazivom PIL (Python Imaging Library). Projekt je preuzeo Alex Clark 2009. godine. Godinu kasnije je izdao novu verziju biblioteke i nazvao ju Pillow [14]. Pillow sadrži mnogo metoda za obradu slike, intuitivna je pa je stoga jedna od najkorištenijih biblioteka za obradu slike. Biblioteka podržava velik broj formata slika, neki od njih su: BMP, PNG, JPEG i TIFF.

3.2.6. Shapely biblioteka

Shapely je Python biblioteka koja služi za manipulaciju i analizu planarnih geometrijskih objekata [15]. Shapely je omotač za GEOS. GEOS je C/C++ biblioteka koja se koristi u računalnoj geometriji koristeći algoritme geografskog informacijskog sustava (GIS) [16]. Funkcije Shapely-a omogućavaju višenitnost što znači da je moguće izvršavati više zadataka paralelno.

3.2.7. PyMuPDF biblioteka

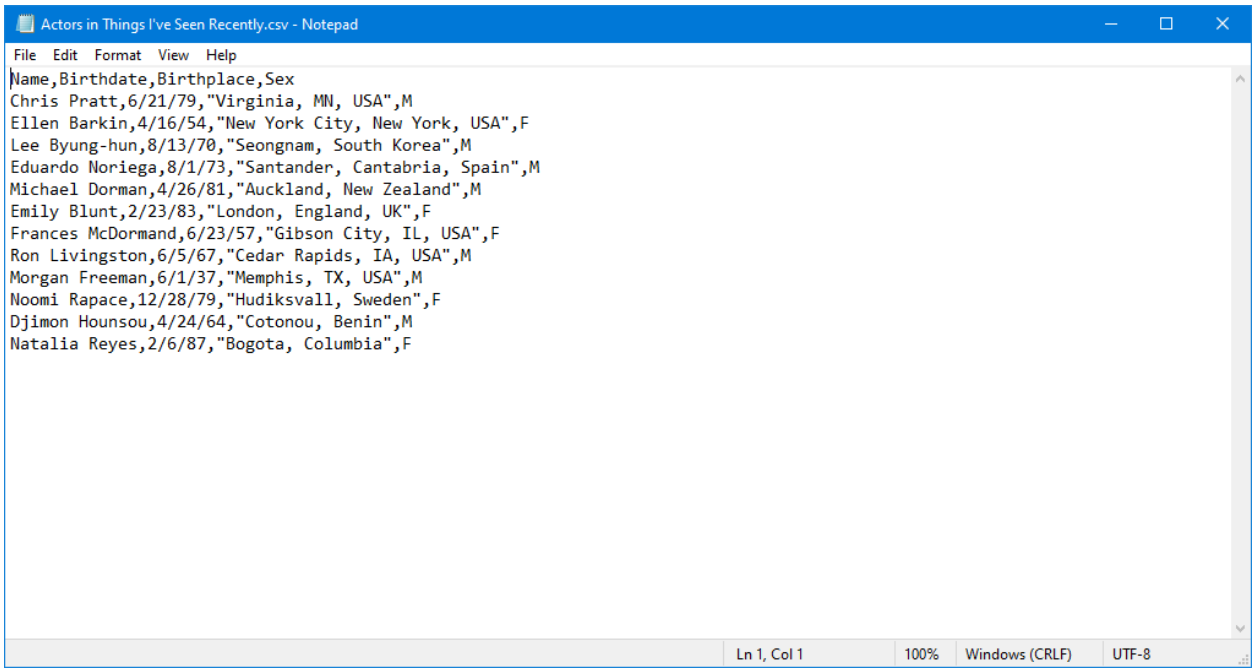
PyMuPDF je Python biblioteka otvorenog koda, visokih performansi koja omogućuje ekstrakciju podataka, analizu, konverziju i manipulaciju PDF dokumenata [17]. Osim PDF-ova podržava i druge formate kao što su XPS, EPUB, SVG, TXT, MOBI itd. Podržava i neke formate slika kao što su JPG/JPEG, PNG, PPM, PAM, PSD.

3.2.8. Auto Py to Exe

Auto Py to Exe je grafičko sučelje za kreaciju i izgradnju izvršnih datoteka iz Python koda i modula. Auto Py to Exe zapravo interno poziva PyInstaller. PyInstaller Python aplikaciju i sve njezine ovisnosti sakuplja u jedan paket. Stoga, korisniku omogućava pokretanje aplikacije bez instalacije Python prevoditelja. PyInstaller je moguće koristiti na mnogim operacijskim sustavima među kojima su Windows, Linux, macOS. PyInstaller je ovisan o platformi što znači ako želimo izgraditi aplikaciju za Windows moramo ju izgraditi na računalu s Windows operacijskim sustavom.

3.3. CSV datoteka

Comma Separated Values (CSV) datoteka je datoteka čistog teksta koja pohranjuje podatke na način da razdvaja unos podataka zarezom [18]. Svaki redak predstavlja jedan zapis podatka. Ove datoteke se koriste kad više različitih programa koriste iste podatke ili za pohranjivanje podataka. Prednost ovih datoteka je jednostavno kreiranje i brzina. Nedostatak je što može spremiti samo tekst, a ne neke složenije strukture. Ove datoteke je moguće otvoriti običnim uređivačem teksta ili programom za proračunske tablice. Primjer CSV datoteke prikazan je na slici 3.2.



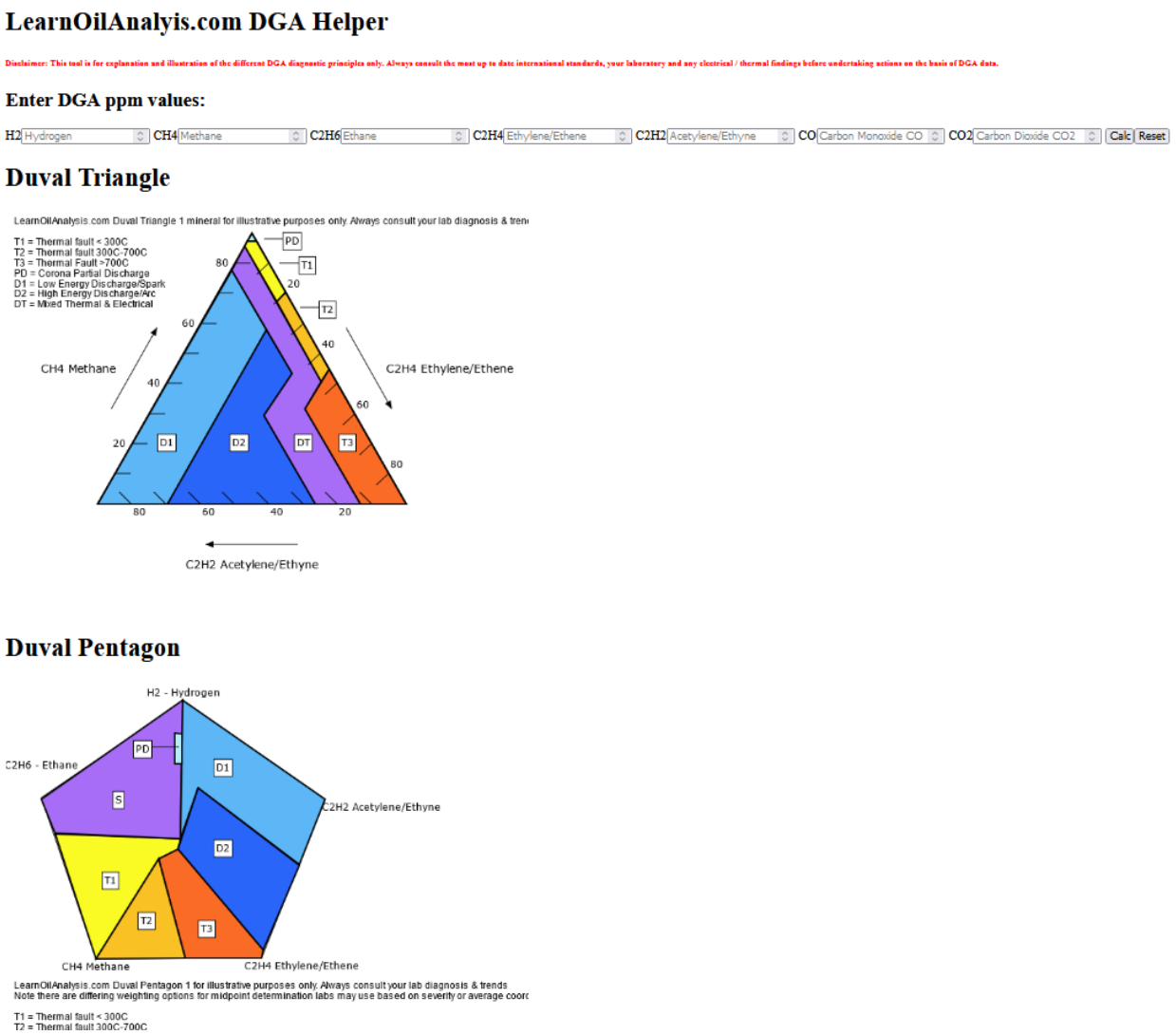
Slika 3.2. Primjer CSV datoteke [19]

3.4. PDF datoteka

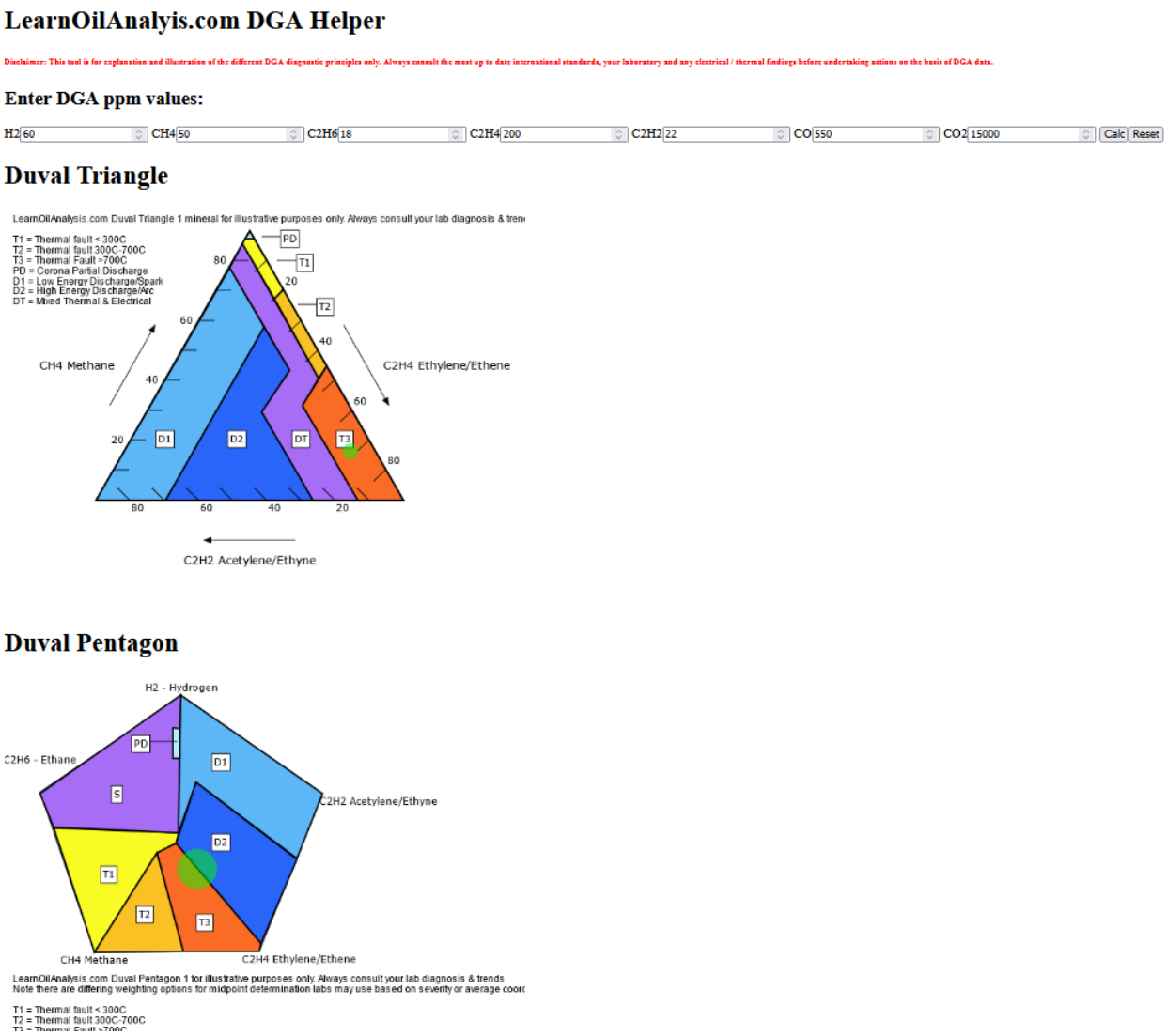
Portable Document Format (PDF) datoteka je standardizirana datoteka, održavana od strane Međunarodne organizacije za standardizaciju (ISO) koja služi za prijenos i prikaz dokumenata. Kreirao ga je Adobe s ciljem omogućavanja razmjene dokumenata neovisno sofveru, sklopovlju i operacijskom sustavu. PDF datoteke mogu sadržavati tekst, slike, videozapise, audiozapise, gumbove i moguće je elektronički potpisati dokumente. PDF datoteke je moguće otvoriti pomoću web preglednika ili posebnih PDF aplikacija kao što je Adobe Acrobat Reader.

4. PREGLED POSTOJEĆIH RJEŠENJA

Zbog specifičnih zahtjeva i funkcionalnosti ove aplikacije ne postoji potpuno ista aplikacija s istim zahtjevima. Postoji nekoliko aplikacija za određivanje kvarova transformatora DGA analizom, no većina njih nije javno dostupna. Ipak, postoji jedna javno dostupna web aplikacija za određivanje kvarova transformatora pod nazivom *LearnOilAnalysis. com DGA Helper*. Aplikacija je jednostavna za korištenje. Sastoji se od 7 inputa za unos vrijednosti plinova izraženih u ppm. Nakon unosa vrijednosti klikne se na gumb *Calc* te se na već nacrtanom Duvalovom trokutu i Duvalovom peterokutu nacrta kružić. Na slici 4.1. je prikazan početni izgled web aplikacije dok je na slici 4.2. prikazana web aplikacija nakon unosa vrijednosti.



Slika 4.1. Prikaz web aplikacije LearnOilAnalysis. com DGA Helper [20]



Slika 4.2. Prikaz DGA rezultata web aplikacije nakon unesenih vrijednosti [20]

Moguće je prikazati više DGA rezultata na istim likovima jer novim unosom rezultata, prethodni kružić se ne obriše, nego se doda novi. U slučaju da se želi obrisati prikaz prošlog DGA rezultata, potrebno je kliknuti na gumb *Reset*.

5. PROGRAMSKO RJEŠENJE

U ovom poglavlju je objašnjeno kako aplikacija funkcionira, kako je ona razvijana i izgrađena. Objašnjene su neke mogućnosti i značajke aplikacije.

5.1. Opis funkcionalnosti programskog rješenja

Aplikacija učitava XLSX datoteku koja sadrži vrijednosti koncentracije plinova, sliku kromatograma te oznaku uzorka. Te podatke je potrebno obraditi te usporediti vrijednosti s dozvoljenim vrijednostima HRN EN60599 normi odnosno vrijednostima postavljenima u postavkama. U slučaju kvara, potrebno je nacrtati i označiti područje kvara na Duvalovom trokuti i Duvalovom peterokutu i napisati o kojem mogućem kvaru je riječ. Izvještaj je potrebno generirati u PDF formatu. Moguće je imati više radnih listova u XLSX datoteci s više podataka što znači da za svaki pojedini radni list je potrebno generirati po jedan izvještaj. Nakon generiranja jednog ili više izvještaja (ovisno o broju radnih listova), u aplikaciji se prikazuju generirani izvještaji. Izvještaje je moguće ispisati pomoću odabranog pisača. U aplikaciji je omogućeno učitati datoteku pomoću pretraživanja datoteke ili je moguće „povući i ispustiti“ datoteku u predviđeno mjesto u aplikaciji. Aplikacija još sadrži i postavke u kojima je moguće mijenjati donje i gornje granice plinova u slučaju da se vrijednosti propisane norme promijene. Postavke se spremaju u CSV datoteku i čitaju iz nje. Kod otvaranja prozora postavki, moguću je vidjeti trenutno namještene vrijednosti. U aplikaciji se još nalazi traka napretka koja prikazuje napredak obavljenih operacija pri generiranju izvještaja nakon učitavanja ulazne XLSX datoteke.

Aplikacija funkcionira na način da se nakon učitane datoteke prvo kreira mapa unutar mape *Data* koja ima ime isto kao učitana datoteka. Nakon toga se unutar te novokreirane mape kreira još onaj broj mapa koliko ima radnih listova u učitanoj datoteci. Imena tih mapa su brojevi. U tim mapa se generira CSV datoteka koja sadrži podatke o koncentracijama, sliku kromatograma, sliku Duvalovog trokuta i sliku Duvalovog peterokut za taj radni list. Nakon generiranja slika i datoteke, ti podaci se koriste za generiranja izvještaja koji se također generira u toj mapi.

5.2. Razvoj programskog rješenja

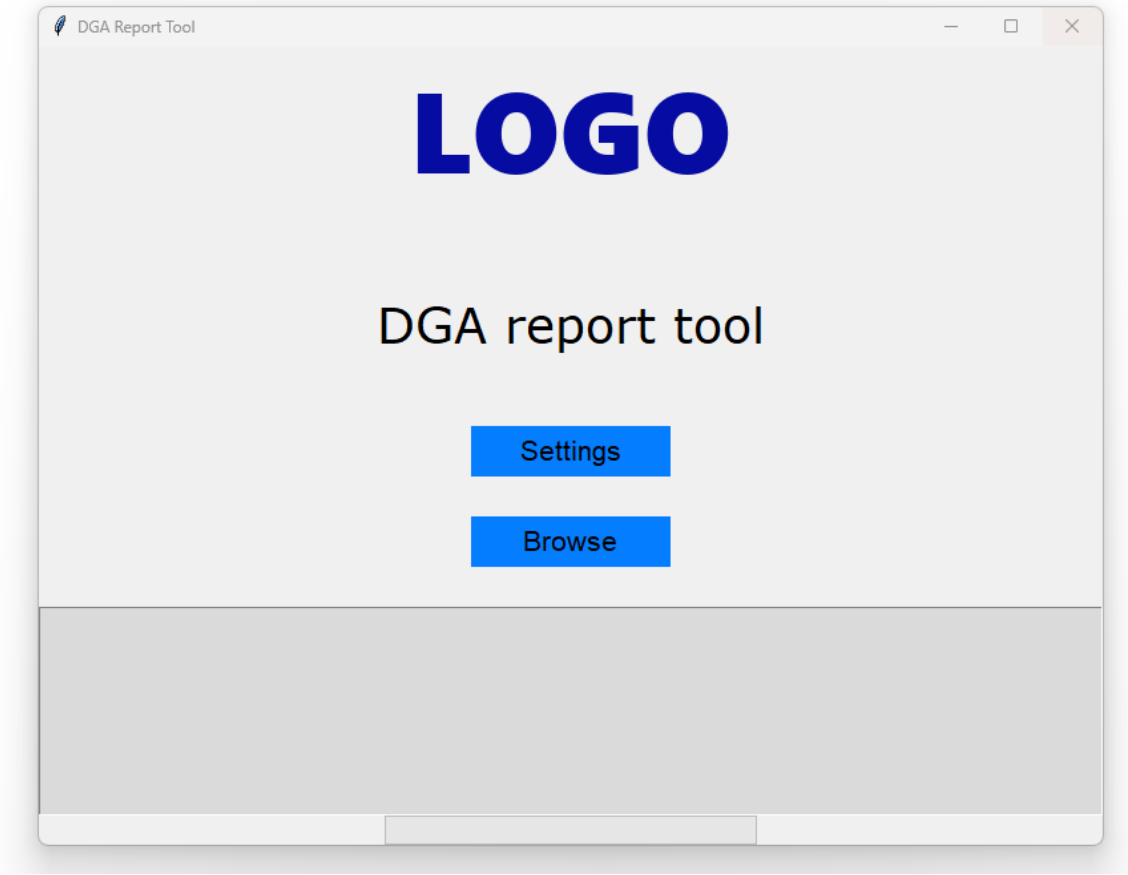
U ovom potpoglavlju je objašnjena instalacija okoline, struktura koda te izgradnja same aplikacije.

5.2.1. Instalacija i priprema okoline

Za razvoj aplikacije u Pythonu potrebno je instalirati Python okruženje. Instalacijsku datoteku Python okruženja je moguće preuzeti sa službene Python stranice. Nakon instalacije okruženja, potrebno je instalirati potrebne biblioteke. Biblioteke se instaliraju pomoću naredbe *pip install „ime biblioteke“*. Pip je sustav za upravljanje i instaliranje Python biblioteka i paketa. Dolazi već instaliran s instalacijom Pythona. Nakon instalacije Python biblioteka, instaliran je Visual Studio Code uređivač teksta. Nakon toga je kreirana mapa naziva *DGA report* koju smo otvorili u Visual Studio Code-u te u nju kreirali Python module za pisanje Python koda. Unutar te mape kreirane su još dvije mape *Data* u koju se generiraju podaci i izvještaj, i *Images* mapa gdje su smještene slike koje aplikacija koristi.

5.2.2. Pisanje programskog koda

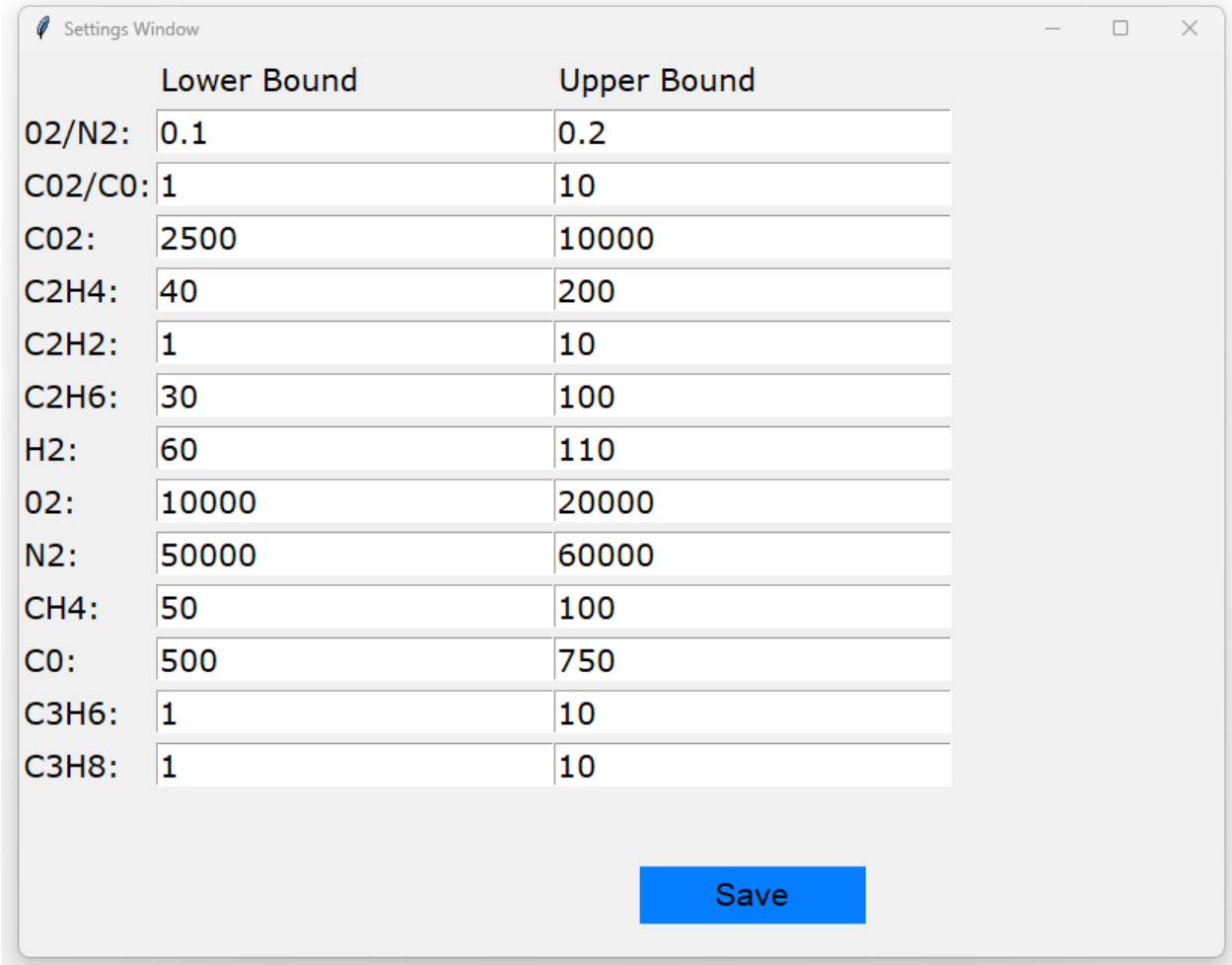
Kod je napisan u nekoliko modula koji sadrže klase za obavljanje nekog dijela aplikacije. U modulu *app.py* su definirane klase za izgled i funkcionalnosti glavnog prozora i prozora postavki. Grafičko sučelje je kreirano s već spomenutom tkinter bibliotekom. U klasi *Program* je definiran izgled glavnog programa dok je u klasi *Settings* definiran izgled prozora postavki. Na slici 5.1. je prikaz glavni prozor aplikacije dok je kod prikazan za glavni prozor prikazan na slici 5.2. Na slici 5.3. je prikazan izgled prozora postavki.



Slika 5.1. Prikaz izgleda glavnog prozora aplikacije

```python
class Program:
    def __init__(self):
        root = TkinterDnD.Tk()
        root.geometry("800x600")
        root.title("DGA Report Tool")
        root.state('zoomed')
        logo = Image.open(r"Images\logo.png")
        logo = ImageTk.PhotoImage(logo)
        logo_label = Label(root, image=logo)

        logo_label.pack(fill="both", expand=1)
        import_lbl = Label(root,
                           text="DGA report tool",
                           font=('Verdana', 28))
        import_lbl.pack(pady=50, fill="both", expand=1)
        settings_btn = Button(root,
                              text="Settings",
                              height=1, width=12,
                              border=0,
                              font=('Comic Sans', 16),
                              background="#047eff",
                              activebackground="#047eff",
                              command=self.settings_window
                              )
        settings_btn.pack()
        browse_btn = Button(root,
                            text="Browse",
                            height=1, width=12,
                            border=0,
                            font=('Comic Sans', 16),
                            background= "#047eff",
                            activebackground="#047eff",
                            command= self.browse_file
                            )
        browse_btn.pack(pady=30)
        drag_and_drop_entry = Entry(root,
                                    font=("default", 100),
                                    state=DISABLED,
                                    )
        drag_and_drop_entry.config(disabledbackground="#dadada")
        drag_and_drop_entry.pack(fill="both", expand=1)

        drag_and_drop_entry.drop_target_register(DND_ALL)
        drag_and_drop_entry.dnd_bind("<<Drop>>", self.import_file_dnd)
        self.pb = ttk.Progressbar(
            root,
            orient='horizontal',
            mode='determinate',
            length=280
        )

        self.pb.pack()
        self.pb['value'] = 0
        root.mainloop()
```

Slika 5.2. Prikaz koda za izgled glavnog prozora aplikacije



Slika 5.3. Prikaz prozora postavki popunjen s proizvoljnim vrijednostima

U modulu *report* se nalazi klasa *PDF_generator* za generiranje izvještaja. Klasa sadrži metode za prikupljanje podataka i generiranje teksta i slika te na kraju spremanja datoteke izvještaja. Na slici 5.4. je prikazana metoda za slaganje i generiranje PDF dokumenta.

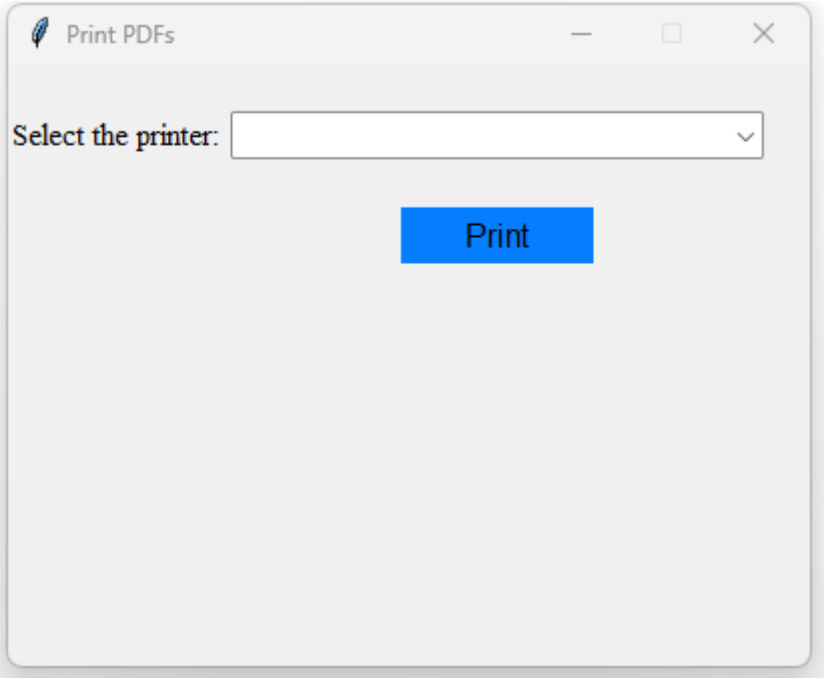
```python
def generete_pdf(self, folder_id, failure):
    c = canvas.Canvas(f"{folder_id}\Report.pdf", pagesize=A4)
    pdfmetrics.registerFont(TTFont('TNR', 'times.ttf'))
    logo_path = "Images/logo.png"
    logo_width = 7.5 * cm
    logo_height = 3 * cm
    c.drawImage(logo_path, 1.5 * cm, A4[1] - 3 * cm, width=logo_width, height=logo_height, mask='auto')

    c.setFont(psfontname="Times-Roman", size=10)

    title = "<b>DGA IZVJEŠTAJ</b>"
    style = ParagraphStyle(name="Title", fontName="TNR", fontSize=20, alignment=1)
    title_object = Paragraph(text=title, style=style, encoding="UTF-8")
    title_width, title_height = title_object.wrap(A4[0], A4[1])
    title_object.drawOn(c, 0, 25 * cm)

    chromatogram = f"{folder_id}\chromatogram.png"
    c.drawImage(chromatogram, 3 * cm, A4[1] - 14 * cm, width=15 * cm, height=7 * cm)

    table, sample_name = self.create_table(folder_id)
    table.wrapOn(c, aW=A4[0], aH=A4[1])
    table.drawOn(c, 2 * cm, 1 * cm)
    style = ParagraphStyle(name="Text", fontName="TNR", fontSize=11, alignment=0)
    sample_object = Paragraph("Oznaka uzorka: " + sample_name, style)
    sample_width, sample_height = sample_object.wrap(A4[0], A4[1])
    sample_object.drawOn(c, 1.5 * cm  , 23.5 * cm)
    if os.path.isfile(f"{folder_id}\\triangle.png"):
        c.showPage()
        pentagon_path = f"{folder_id}\pentagon.png"
        triangle_path = f"{folder_id}\\triangle.png"
        c.drawImage(pentagon_path, 1 * cm, A4[1] - 13 * cm, width=16 * cm, height= 12 * cm, mask='auto')
        c.drawImage(triangle_path, 1 * cm, A4[1] - 25 * cm, width=16 * cm, height= 12 * cm, mask='auto')
        if(failure != None):
            style = ParagraphStyle(name="Text", fontName="TNR", fontSize=11, alignment=0)
            sample_object = Paragraph(text="Kvar: " + self.get_triangle_failure_name(failure[0]) + "/" + self.get_pentagon_failure_name(failure[1]) , style=style, encoding="utf-16")
            sample_width, sample_height = sample_object.wrap(A4[0], A4[1])
            sample_object.drawOn(c, 1.5 * cm  , 4.5 * cm)
        else:
            style = ParagraphStyle(name="Text", fontName="TNR", fontSize=11, alignment=0)
            sample_object = Paragraph(text="Kvar: nema")
            sample_width, sample_height = sample_object.wrap(A4[0], A4[1])
            sample_object.drawOn(c, 1.5 * cm  , 4.5 * cm)
    c.save()
```

Slika 5.4. Prikaz koda metode za generiranje izvještaja u PDF formatu

U modulu *PDFHandler* kreirana je klasa *PDFHandle*. Ta klasa sadrži metode za prikaz generiranih izvještaja te elemente za mogućnost ispisa izvještaja putem pisača. Slika koda za ispis izvještaja se nalazi na slici 5.5. Na slici 5.6. prikazan je prozor za ispis izvještaja.

```python
def print_pdf(self, printer_name, pdf_path):
    mat = fitz.Matrix(700 / 72, 700 / 72)
    pdf_document = fitz.open(pdf_path)

    printer_handle = win32print.OpenPrinter(printer_name)
    try:

        hdc = win32ui.CreateDC()
        hdc.CreatePrinterDC(printer_name)

        hdc.StartDoc(pdf_path)

        for page_num in range(len(pdf_document)):
            page = pdf_document.load_page(page_num)
            pix = page.get_pixmap(dpi=600)

            img = Image.frombytes("RGB", [pix.width, pix.height], pix.samples)
            hdc.StartPage()

            dib = ImageWin.Dib(img)

            dib.draw(hdc.GetHandleOutput(), (0, 0, img.width, img.height))

            hdc.EndPage()

        hdc.EndDoc()
    finally:
        win32print.ClosePrinter(printer_handle)
```

Slika 5.5. Prikaz koda za ispis izvještaja

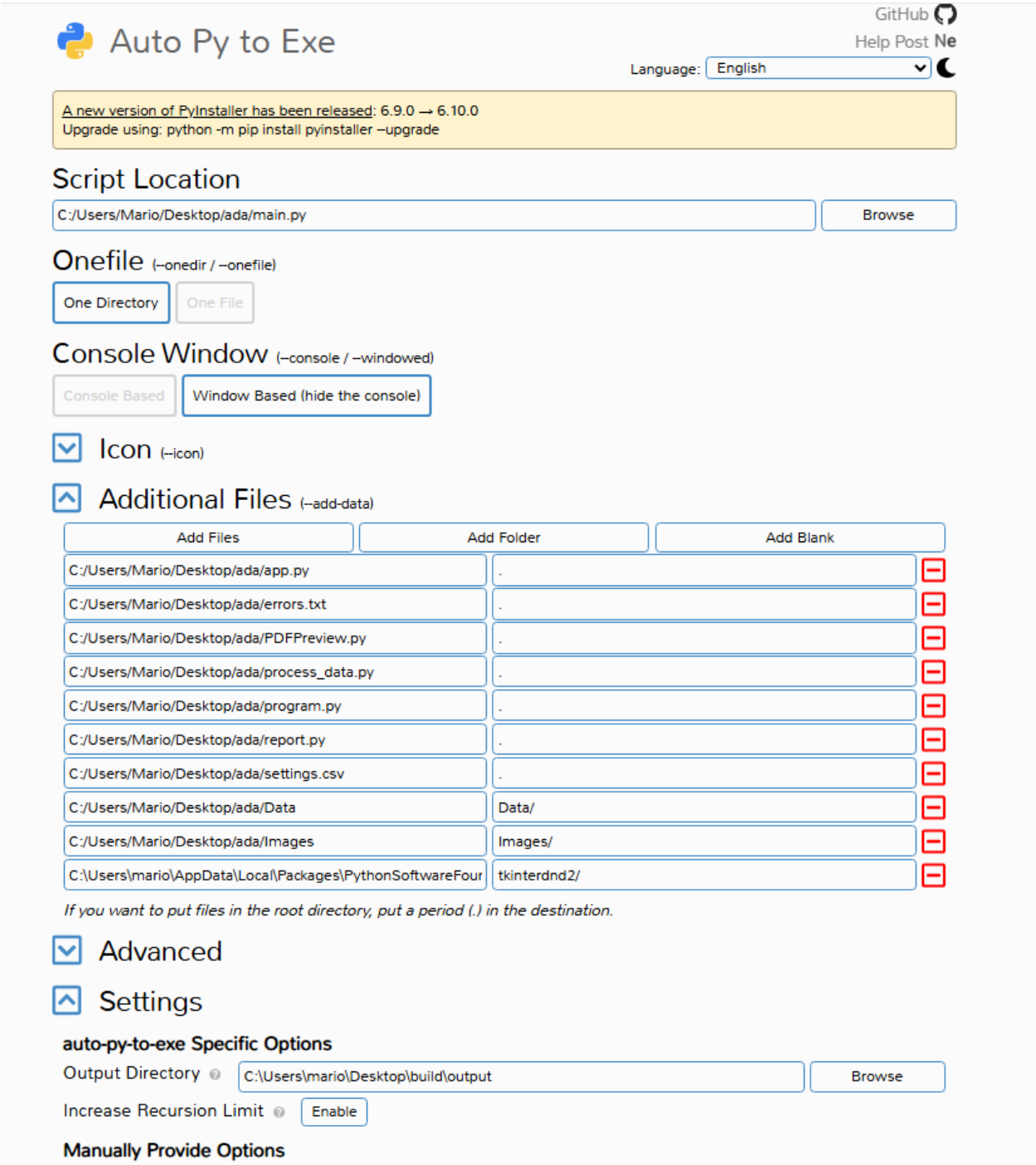


Slika 5.6. Prikaz prozora za ispis izvještaja

Kod za pronalazak i obradu podataka je napisan u modulu *process_data* u klasi *ProcessData*. Klasa sadrži metode za pronalazak i ekstrakciju podataka iz ulazne datoteke, generiranje potrebnih mapa i datoteka. U modulu *program.py* nalazi se apstraktna klasa *Polygon* i dvije klase koje ju nasljeđuju *Triangle* i *Pentagon*. U ovim klasama su definirane metode za crtanje Duvalovog trokuta i peterokuta te za pronalazak kvara.

5.2.3. Izgradnja aplikacije

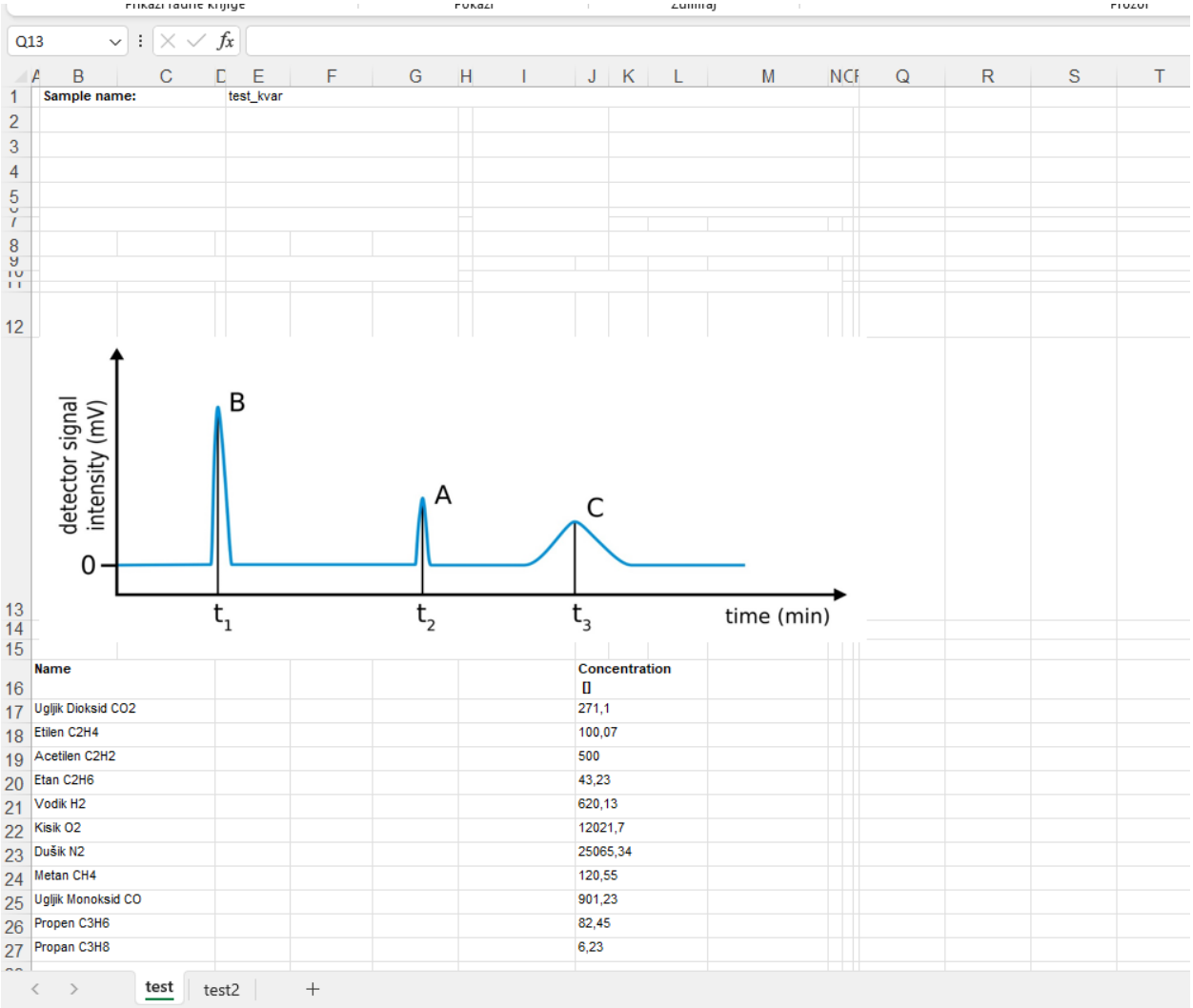
Za izgradnju aplikacije je korišten Auto Py To Exe. Prikaz postavki za izgradnju aplikacije prikazan je na slici 5.7.



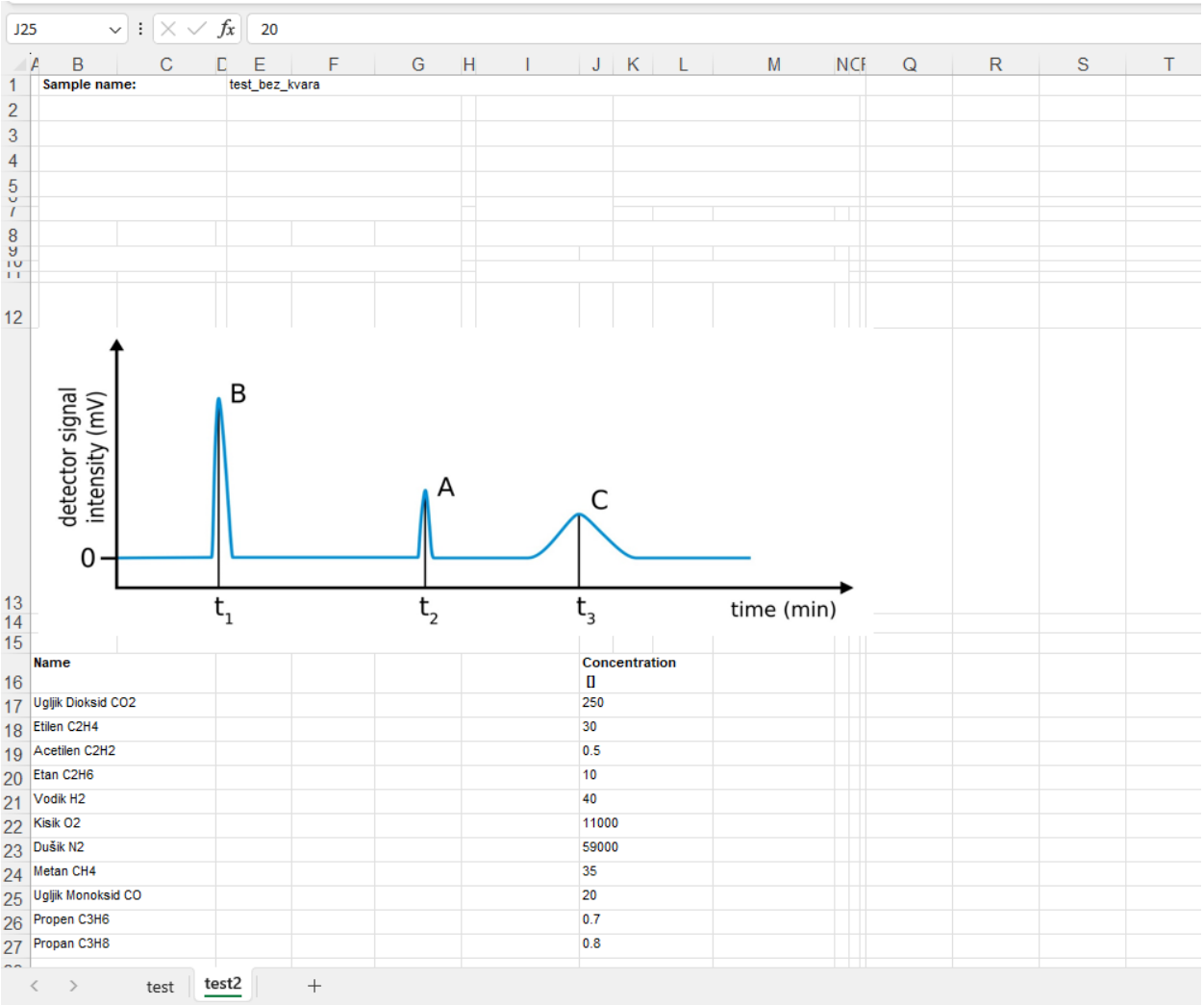
Slika 5.7. Prikaz postavki Auto Py To Exe-a za izgradnju aplikacije

5.3. Testiranje programskog rješenja

Kako bi testirali aplikaciju, prvo je potrebno kreirati ulaznu datoteku s potrebnim podacima. Kreirana datoteka sadrži 2 radna lista s različitim koncentracijama i nazivima uzorka. Jedino je slika kromatograma koja služi kao *placeholder* za pravu sliku kromatograma ista u oba radna lista [21]. Vrijednosti koncentracija su namještene na način da će se za prvi radni list prikazat kvar dok za drugi neće. Ime uzorka prvog radnog lista je *test_kvar* i za taj izvještaj se očekuje da će kvar biti prikazan na izvještaju. Ime uzorka drugog radnog lista je *test_bez_kvara* i na izvještaju se očekuje da kvar neće biti prikazan. Vrijednosti namještene u postavkama su iste kao na slici 5.3. Naziv ulazne datoteke je *test*. Na slikama 5.8. i 5.9. prikazana je datoteka odnosno radni listovi *test* datoteke.

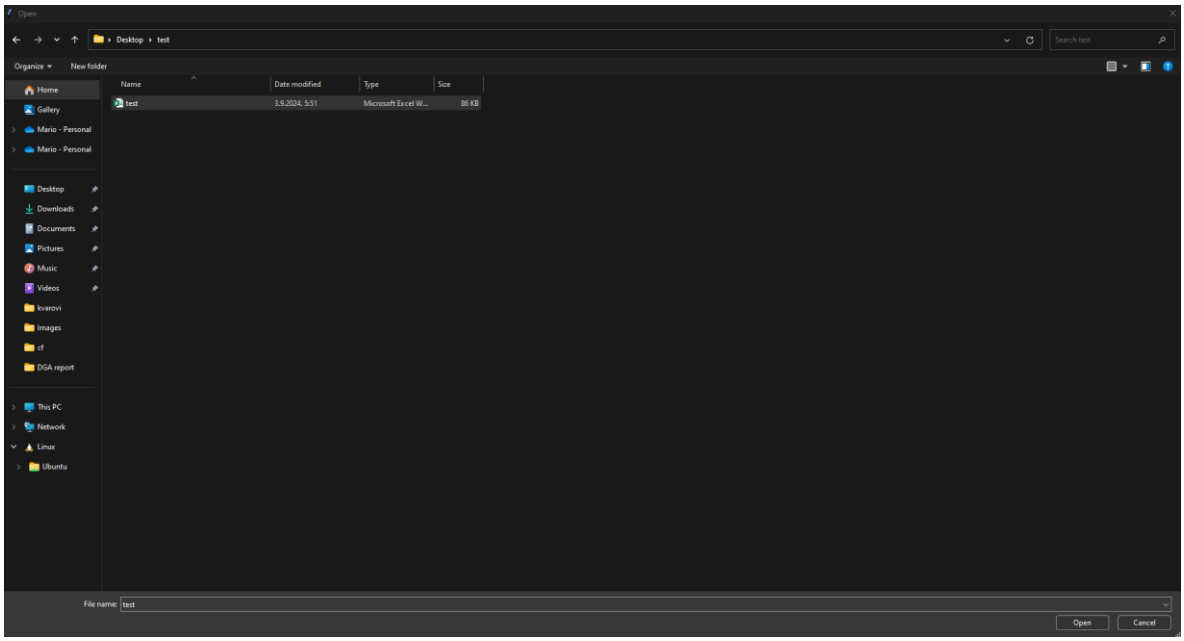


Slika 5.8. Prikaz prvog radnog lista *test* datoteke



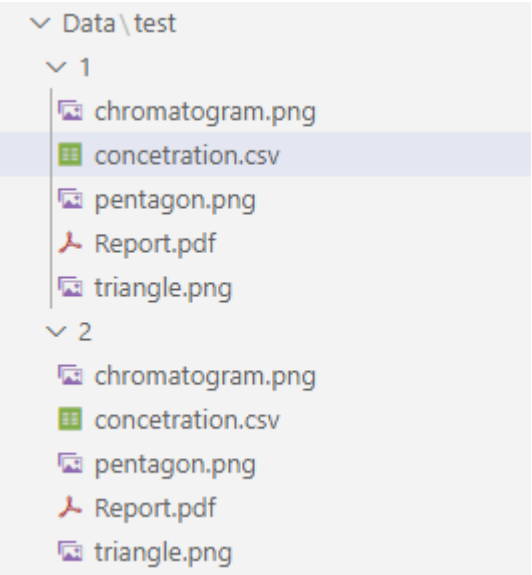
Slika 5.9. Prikaz drugog radnog lista *test* datoteke

Nakon pokretanja aplikacije otvara se prozor kao na slici 5.1. te klikom na gumb *Browse* otvara se Windows pretraživač za odabir datoteke (slika 5.10.). Moguće je datoteku učitati pomoću „povuci i ispusti“ funkcije.



Slika 5.10. Prikaz Windows pretraživača za odabir datoteke

Tokom generiranja sadržaja u mapi *Data* se trebaju generirati dvije nove mape naziva *1* i *2* te u njima se trebaju generiraju slike kromatograma, Duvalovog troukuta i peterokuta, CSV datoteka s koncentracijama i imenom uzorka te na kraju i sam izvještaj (slika 5.11.). Na slici 5.12. je prikazan generirana CSV datoteka za prvi radni list. Po završetku obrade podataka i generiranja izvještaja, otvara se prozor s prikazanim generiranim izvještajima (slika 5.13. i slika 5.14.).

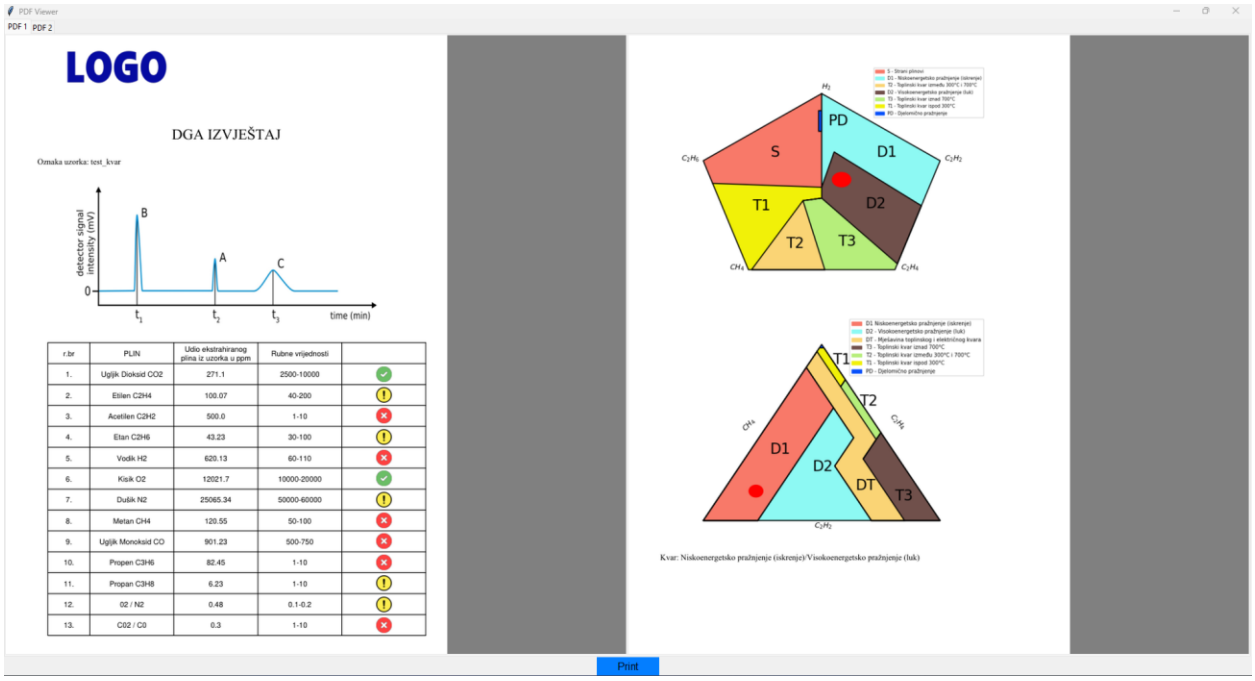


Slika 5.11. Prikaz generiranih mapa i datoteka unutar njih tokom generiranja izvještaja

Data > test > 1 > concetration.csv > data

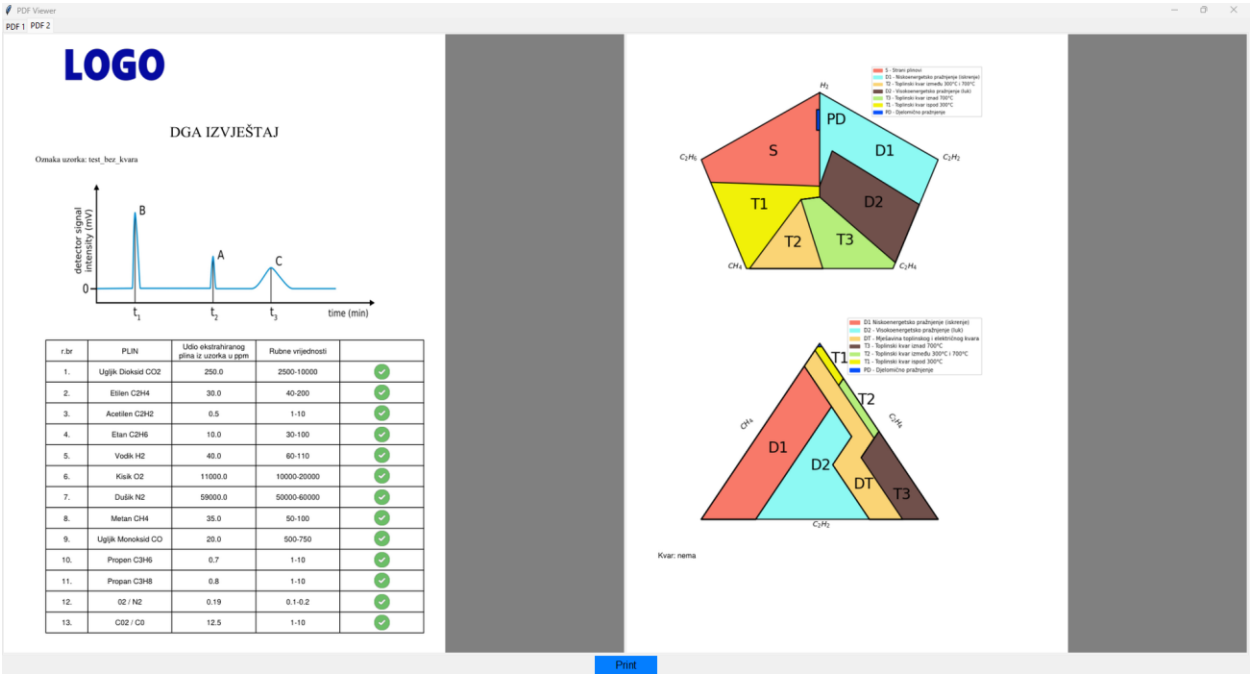
```
Sample name,test_kvar
Ugljik Dioksid CO2,271.1
Etilen C2H4,100.07
Acetilen C2H2,500.0
Etan C2H6,43.23
Vodik H2,620.13
Kisik O2,12021.7
Dušik N2,25065.34
Metan CH4,120.55
Ugljik Monoksid CO,901.23
Propen C3H6,82.45
Propan C3H8,6.23
```

Slika 5.12. Prikaz CSV datoteke s vrijednostima koncentracija za prvi radni list



r.br	PLIN	Udio ekstrahiranog plina iz uzorka u ppm	Rubne vrijednosti	
1.	Ugljik Dioksid CO2	271.1	2500-10000	✓
2.	Etilen C2H4	100.07	40-200	!
3.	Acetilen C2H2	500.0	1-10	✗
4.	Etan C2H6	43.23	30-100	!
5.	Vodik H2	620.13	60-110	✗
6.	Kisik O2	12021.7	10000-20000	✓
7.	Dušik N2	25065.34	50000-60000	!
8.	Metan CH4	120.55	50-100	✗
9.	Ugljik Monoksid CO	901.23	500-750	✗
10.	Propen C3H6	82.45	1-10	✗
11.	Propan C3H8	6.23	1-10	!
12.	02 / N2	0.48	0.1-0.2	!
13.	C02 / C0	0.3	1-10	✗

Slika 5.13. Prikaz prozora prvog generiranog izvještaja



Slika 5.14. Prikaz prozora drugog generiranog izvještaja

Iz prvog izvještaja se može vidjeti da su koncentracije nekih plinova narušene pa je zbog toga označen kvar na Duvalovom trokutu i peterokutu. Kvar označen na peterokutu je visokoenergetsko pražnjenje dok kvar označen na trokutu je niskoenergetsko pražnjenje. Ovo je normalna pojava jer u nekim slučajevima (kao u ovom) dolazi do rasipanja. Iz drugog izvještaja se vidi da su sve koncentracije i omjeri dobri pa kvarovi nisu označeni u trokutu.

6. ZAKLJUČAK

Cilj ovog rada je bilo razviti desktop aplikaciju za generiranje izvještaja na temelju podataka kromatograma DGA analize. Aplikacija učitava datoteku koja sadrži koncentracije pojedinih plinova, provjera vrijednosti te na temelju njih generira izvještaj. Vrijednosti se uspoređuju prema HRN EN 60599 no moguće je mijenjati dozvoljene granice plinova i omjera jer aplikacija ima postavke.

U ovom radu je objašnjen i prikazan rad same aplikacije, navedene su i opisane korištene tehnologije. Objašnjeno je što je DGA analiza i zašto je ona važna pri održavanju transformatora. Objašnjene su dvije metode interpretacije rezultata DGA analize. Prikazana je jedna slična web aplikacija za određivanje kvara. Aplikacija je jednostavna za korištenje

Aplikacija bi se mogla unaprijediti na način da se prilikom generiranja izvještaja može odabrati u kojem će se formatu izvještaj spremiti kao i lokaciju gdje će se spremiti. Također bi se mogle dodati opcije kod ispis izvještaja

LITERATURA

[1] Hrvatska enciklopedija, mrežno izdanje. Leksikografski zavod Miroslav Krleža, 2013. – 2024.: https://www.enciklopedija.hr/clanak/transformator, pristupljeno 21. 6. 2024.

[2] Vietnam Power Transformer, Transformer oil: Testing, Types, and Properties: https://vietnamtransformer.com/our-news/transformer-oil-types-mbt-transformer, pristupljeno 23. 6. 2024.

[3] Huazheng, Vrsta i funkcija transformatorskog ulja: https://hr.transformer-tester.com/info/type-and-function-of-transformer-oil-59880715.html, pristupljeno 23. 6. 2024.

[4] Huazheng, Karakteristike ulja transformatora: https://www.electric-test.com/info/transformer-oil-characteristics-33215154.html , pristupljeno 1. 8. 2024.

[5] Huayi Electric, What is The Dissolved Gas Analysis: https://www.sinohipot.com/info/what-is-the-dissolved-gas-analysis-96597518.html, pristupljeno 6. 8. 2024.

[6] Bustamante, Sergio & Mañana, Mario & Arroyo, Alberto & Castro, Pablo & Laso, Alberto & Martínez, Raquel. (2019). Dissolved Gas Analysis Equipment for Online Monitoring of Transformer Oil: A Review. Sensors.

[7] InfoWorld, What is Visual Studio Code? Microsoft's extensible code editor: https://www.infoworld.com/article/2335960/what-is-visual-studio-code-microsofts-extensible-code-editor.html, pristupljeno 15. 8. 2024.

[8] Python, What is Python? Executive Summary: https://www.python.org/doc/essays/blurb/?external-link=true, pristupljeno 17.8. 2024.

[9] Geeks for geeks, What is Tkinter for Python?: https://www.geeksforgeeks.org/introduction-to-tkinter/, pristupljeno 20. 8. 2024.

[10] Geeks for geeks, Introduction to Matplotlib, https://www.geeksforgeeks.org/python-introduction-matplotlib/, pristupljeno 22.8. 2024.

[11] ReporLab Docs, Introduction: https://docs.reportlab.com/reportlab/userguide/ch1_intro/, pristupljeno 23. 8. 2024.

[12] Gentoo linux, dev-python/reportlab: https://packages.gentoo.org/packages/dev-python/reportlab, pristupljeno 23. 8. 2024.

[13] Dana Scientest, PyWin32: Unveiling the Python Extension Exclusively for Windows Systems, https://datascientest.com/en/pywin32-unveiling-the-python-extension-exclusively-for-windows-systems, pristupljeno 23. 8. 2024.

[14] Dana Scientest, Pillow: How to process images with Python: https://datascientest.com/en/pillow-how-to-process-images-with-python, pristupljeno: 24. 8. 2024.

[15] Shapely 2.0.6. documentation, Shapely: https://shapely.readthedocs.io/en/stable/, pristpljeno 26.8. 2024.

[16] GEOS, GEOS: https://libgeos.org/, pristpupljeno 26. 8. 2024.

[17] PyMuPDF 1.24.9 documentation, Welcome to PyMuPDF: https://pymupdf.readthedocs.io/en/latest/, pristupljeno 26. 8. 2024.

[18] How-To Geek, What Is a CSV File, and How Do i Open it?: https://www.howtogeek.com/348960/what-is-a-csv-file-and-how-do-i-open-it/, pristupljeno 27.8. 2024.

[19] Modern CSV, How to Create CSV Files (WooCommerce CSV Import Example): https://www.moderncsv.com/how-to-create-csv-files-woocommerce-import-csv-example/, pristupljeno 27. 8. 2024.

[20] Oilanalysislab, LearnOilAnalysis. com DGA Helper: https://oilanalysislab.com/Oil_Analysis_Downloads/LW/DGA/, pristupljeno 25. 8. 2024.

[21] Wikipedia, Resolution (chromatography): https://en.wikipedia.org/wiki/Resolution_(chromatography)#/media/File:Chromatogram_in_English.svg, pristupljeno 2. 9. 2024.

SAŽETAK

Transformator je električni uređaj koji služi podizanju ili spuštanju iznosa izmjeničnog napona iste frekvencije. Zbog toga, transformatori su ključni elementi kod distribucije električne energije. Shodno tome, njihovo održavanje i ispravnost su jako bitni. Jedna od metoda za održavanje transformatora je DGA analiza. DGA analiza je analitička metoda koja se temelji na određivanju koncentracije otopljenih plinova u transformatorskom ulju. U ovom radu je kreirana desktop aplikacija za interpretaciju i generiranje izvještaja rezultata DGA analize. Aplikacija sadrži postavke gdje se mogu mijenjati minimalne i maksimalne dozvoljene granice plinova, nakon generiranja izvještaja aplikacija prikazuje generirani izvještaj te postoji mogućnost ispisa izvještaja izravno iz aplikacije. Aplikacija je razvijena pomoću Python programskog jezika.

Ključne riječi: desktop aplikacija, DGA analiza, Python, transformator, transformatorsko ulje

ABSTRACT

Application for interpretation of results according to HRN EN60599

A transformer is an electrical device that serves to raise or lower the amount of alternating voltage of the same frequency. That is why transformers are key elements in the distribution of electricity. Consequently, their maintenance and correctness are very important. One of the methods for transformer maintenance is DGA analysis. DGA analysis is an analytical method based on determining the concentration of dissolved gases in transformer oil. In this work, a desktop application was created for the interpretation and generation of DGA analysis results reports. The application contains settings where the minimum and maximum permitted limits of gases can be changed, after generating the report the application displays the generated report and there is an option to print the report directly from the application. The application was developed using the Python programming language.

Key words: desktop application, DGA analysis, Python, transformer, transformer oil