Aplikacija za ekstrakciju metapodataka iz PPTX formata i konverziju u Markdown sintaksu

Troha, Tomislav

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

Tomislav Troha

APLIKACIJA ZA EKSTRAKCIJU METAPODATAKA IZ PPTX FORMATA I KONVERZIJU U MARKDOWN SINTAKSU

Diplomski rad

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

Tomislav Troha

APLIKACIJA ZA EKSTRAKCIJU METAPODATAKA IZ PPTX FORMATA I KONVERZIJU U MARKDOWN SINTAKSU

Diplomski rad

JMBAG: 0303082820, redovni student
Studijski smjer: Informatika

Predmet: Suvremene tehnike programiranja
Znanstveno područje: Društvene znanosti
Znanstveno polje: Informacijske i komunikacijske znanosti
Znanstvena grana: Informacijski sustavi i informatologija

Mentor: Izv.prof.dr.sc. Siniša Sovilj
Komentor: izv.prof.dr.sc. Darko Etinger

Pula, _____, ___ godine

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani **Tomislav Troha**, kandidat za **magistra informatike**

ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljeni način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

_____

Student

U Puli, _____

IZJAVA O KORIŠTENJU AUTORSKOG DJELA

Ja, **Tomislav Troha** dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj Diplomski rad pod nazivom

"Aplikacija za ekstrakciju metapodataka iz pptx formata i konverziju u markdown sintaksu"

koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.
Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____

Potpis

_____

SVEUČILIŠTE JURJA DOBRILE U PULI

FAKULTET INFORMATIKE

Pula, 24. veljače 2023.

DIPLOMSKI ZADATAK

Pristupnik:	**Troha Tomislav (0303082820)**
Studij:	Sveučilišni diplomski studij Informatike
Naslov (hrv.):	**Aplikacija za ekstrakciju metapodataka iz PPTX formata i konverziju u Markdown sintaksu.**
Naslov (eng.):	Application for metadata extraction from PPTX format and conversion to Markdown syntax.
Opis zadatka:	Zadatak je izraditi aplikaciju za ekstrakciju metapodataka iz Microsoft PowerPoint dokumenata formata PPTX i konverziju u tekstualne dokumente s Markdown sintaksom. Aplikacija se treba temeljiti na tehnologijama Java/Kotlin. Omogućiti korisnicima uređivanje Markdown sintakse nakon ekstrakcije metapodatka. Istražiti postojeće programske metodologije pohrane, ekstrakcije i prevođenja metapodatka iz različitih formata. Opis sustav - korisničke scenarije, funkcionalnosti, izraditi potrebne UML dijagrame (klasne, use case, use sequence, i sl.), opisati implementaciju te na kraju izraditi kratke korisničke upute i postaviti aplikaciju u javni GitHub repozitorij fakulteta.

Zadatak uručen pristupniku: 24. veljače 2023.

Rok za predaju rada: 24. veljače 2024.

Mentor:	Komentor:
_____	_____
izv.prof.dr.sc. Siniša Sovilj	izv.prof.dr.sc. Darko Etinger

Sadržaj

1. Uvod

U današnjem svijetu sve se više oslanjamo na digitalne platforme za obavljanje svakodnevnih aktivnosti, od komunikacije s kolegama, do izrade prezentacija i izvješća za poslovne i akademske svrhe. U isto vrijeme, pretvaranje dokumenata iz jednog formata u drugi postala je uobičajena potreba. Na primjer, PowerPoint prezentacije (PPTX format) koriste se za pripremu i prezentiranje materijala u nekoliko polja. Međutim, upravljanje i manipuliranje tim prezentacijama može biti nepraktično, osobito kada podatke treba izdvojiti i pretvoriti u čitljiviji format kao što je Markdown. To je jednostavan format za stiliziranje teksta koji je popularan u web okruženjima zbog svoje čitljivosti i lakoće uređivanja. Pretvaranje složenih dokumenata kao što su PPTX datoteke u ovaj format može biti izazovno, predstavljati tehničke prepreke i ometati rad.

Imajući na umu ove izazove, razvijena je računalna aplikacija koja može učinkovito izvršiti transformaciju, izdvajajući ključne metapodatke iz PPTX formata i pretvarajući ih u Markdown sintaksu. Svrha ovog rada je objasniti razvoj aplikacije, njezinu funkcionalnost i tehničke prepreke. Zadatak ovog rada bio je izraditi alat koji bi omogućio brzu i učinkovitu konverziju između ovih formata, čime bi se eliminirala potreba za ručnim radom. Ovaj će alat dramatično poboljšati tijek rada i pojednostaviti proces konverzije, omogućujući vam da se usredotočite na sadržaj, a ne na tehničke aspekte konverzije. Aplikacija predstavljena u ovom radu korak je prema tom cilju.

2. Tehnologije korištene pri izradi računalne aplikacije

Java je objektno - orijentiran programski jezik visoke razine dizajniran da ima što manje ovisnosti o implementaciji. To je programski jezik opće namjene koji je namijenjen programerima da pišu jednom, pokreću bilo gdje, (eng. WORA) što znači da prevedeni Java kôd može raditi na svim platformama koje podržavaju Javu bez potrebe za ponovnim prevođenjem (1).

2.1. Biblioteke i alati korišteni pri izradi računalne aplikacije

2.1.1. Apache POI

Ekstrakcija metapodataka iz PPTX datoteka pomoću programskog jezika Java uvelike ovisi o biblioteci Apache POI, koja pruža bitne alate za ovaj zadatak. Za upute korištena je službena dokumentacija (2).

2.1.2. Flexmark

Flexmark, verzija 2.1.2, koristan je alat. Kompatibilan je s Javom i Kotlinom, a svrha mu je pretvaranje sadržaja u one koji su kompatibilni s HTML-om. Ovaj je alat također platforma otvorenog kôda, što znači da ga svatko može slobodno koristiti i modificirati prema vlastitim potrebama. Jedna od njegovih zanimljivih značajki je sposobnost preoblikovanja dokumenata prema korisničkim preferencijama, izrada makronaredbi za smanjenje tipografskih pogrešaka i poboljšanje ukupne točnosti sadržaja. Specifikacija CommonMark je ono na čemu se temelji Flexmark, Java implementacija. Pregledavajući dokumente, može se naučiti kako ispravno stvoriti Markdown koji odgovara izdvojenim podacima.(3).

2.1.3. JavaFX

JavaFX, nudi svestranu platformu za stvaranje bogatih grafičkih korisničkih sučelja (GUI) koja se mogu koristiti na različitim uređajima, uključujući stolna računala, preglednike i mobilne uređaje. To je izražajan jezik koji pruža širok raspon multimedijskih alata, kao što su 2D i 3D grafika, animacija, audio i video, koji se mogu ugraditi u bilo koju Java aplikaciju. Osim toga, JavaFX ima sveobuhvatnu biblioteku UI kontrola, izgleda i grafikona, što olakšava stvaranje složenih sučelja s minimalnim

kodom. Također ima jednostavan eng. API i kompatibilan je s drugim popularnim Java okvirima, kao što su Swing i Java 2D. Ukratko, JavaFX je snažna i fleksibilna tehnologija za razvoj interaktivnih i privlačnih aplikacija na bilo kojoj platformi. Za izradu sučelja aplikacije proučavani su online vodiči i službena dokumentacija.(4).

2.1.4. GitHub i StackOverflow

GitHub i StackOverflow dva su važna resursa za programere. Oni pružaju platformu za dijeljenje i suradnju na kodu i projektima. To je prostor kojim upravlja zajednica gdje korisnici mogu raspravljati o problemima vezanim uz kôd, dijeliti rješenja i tražiti povratne informacije. Ova web-mjesta često koriste programeri svih razina za učenje i razvoj na terenu. GitHub je posebno poznat po svom sustavu kontrole verzija koji programerima omogućuje upravljanje i praćenje promjena u svom kodu tijekom vremena. StackOverflow ima reputaciju izvornog resursa za rješavanje problema s programiranjem i dobro je poznat po svom formatu pitanja i odgovora. Sve u svemu, ovi su alati vrijedni za programere kako bi poboljšali svoje vještine i znanje, kao i za povezivanje s drugim stručnjacima u industriji. Tijekom procesa razvoja aplikacija, platforme StackOverflow i GitHub pokazuju se kao vrijedan izvor za pronalaženje rješenja i dobivanje praktičnih savjeta. Njihova je vrijednost u tom smislu nemjerljiva.

2.1.5. RFC 4648 - Base16, Base32, Base64 kodiranje podataka

Kodiranje podataka je ključan aspekt pri prijenosu podataka na internetu i drugim mrežama. Svrha kodiranja je pretvoriti podatke u format koji je optimalan za prijenos ili skladištenje. Tri uobičajene metode kodiranja koje se koriste na internetu su Base16, Base32 i Base64, koje su definirane u specifikaciji RFC 4648 (5).

2.1.6. IntelliJ IDEA

Proces razvoja aplikacije poduprlo je integrirano razvojno okruženje (IDE) tvrtke JetBrains. Njihove brojne značajke, uključujući inteligentno dovršavanje koda, refaktoriranje i (eng. Debugging) alate pomogli su u pisanju čistog kôda. Integracija s Git-om imala je isti učinak. Upravljanje verzijama je jednostavno kroz dokumentaciju i brojne online resurse za IntelliJ IDEA, to je omogućilo da se nauče najbolje prakse u

vezi s korištenjem alata (6).

2.1.7. OpenHtmlToPdf biblioteka

Koristeći CSS za kontrolu izgleda, biblioteka otvorenog kôda Openhtmltopdf olakšava stvaranje PDF dokumenata iz HTML sadržaja. Biblioteka također podržava razne web standarde, uključujući web fontove, SVG, HTML i CSS. (7).

3. Metodologije obrade metapodataka

3.1. Metapodaci i njihova pohrana

Metapodaci (ili metainformacije) su podaci koji pružaju informacije o drugim podacima, ali ne i sadržaj podataka, kao što je slika ili tekst poruke (13). Postoje mnoge vrste različite vrste metapodataka, uključujući:

1. Deskriptivni metapodaci - opisne informacije o izvoru. Koristi se za otkrivanje i identifikaciju. Sadrži elemente kao što su naslov, sažetak, autor i ključne riječi.
2. Strukturni metapodaci - ukazuje na to kako se složeni objekti sastavljaju, na primjer, kako su stranice poredane da tvore poglavlja. Opisuje vrste, verzije, odnose i druge karakteristike digitalnih materijala (9).
3. Administrativni metapodaci - informacije koje pomažu u upravljanju resursom, poput vrste resursa, dopuštenja te kada i kako su stvoreni (10).
4. Statistički metapodaci - nazivaju se i procesni podaci, mogu opisivati procese koji prikupljaju, obrađuju ili proizvode statističke podatke (11).
5. Pravni metapodaci - pruža informacije o kreatoru, nositelju autorskog prava i javnoj licenci, ako je navedena.

Pohrana metapodataka odnosi se na procese i sustave koji se koriste za spremanje i održavanje metapodataka. Ovo može uključivati različite tehnologije, uključujući baze podataka, datotečne sustave, usluge u oblaku i specijalizirane sustave za upravljanje metapodacima.

Obično se pohranjuju na jedan od dva načina:

1. Ugrađeni metapodaci - ovi metapodaci su ugrađeni izravno u datoteku ili objekt koji opisuju. Na primjer, digitalna fotografija može sadržavati ugrađene metapodatke koji opisuju kada je slika snimljena, koja kamera je korištena, i druge tehničke detalje.
2. Vanjski metapodaci - metapodaci se pohranjuju odvojeno od podataka koje opisuju. Mogu biti pohranjeni u posebnoj datoteci, bazi podataka ili drugom sustavu za upravljanje metapodacima. Vanjski metapodaci mogu biti korisni kada je potrebno opisati velike količine podataka ili kada je potrebno održavati metapodatke neovisno o podacima koje opisuju.

Rad "MaSQue: An Approach for Flexible Metadata Storage and Querying in RDF" predlaže pristup pod nazivom MaSQue (Metadata Storage and Querying) za fleksibilnu pohranu i upite podataka i njihovih metapodataka neovisno o primijenjenom modelu predstavljanja metapodataka. Autori uvode među format za (meta)podatke i anotacije upita kao sloj metapodataka na vrhu RDF-a i SPARQL-a. Ovaj pristup omogućuje fleksibilnu pohranu i upite podataka i njihovih metapodataka bez obzira na primijenjeni model predstavljanja metapodataka (12).

Također, Rad "Efficient Metadata Indexing for HPC Storage Systems" predstavlja Brindexer, alat za indeksiranje metapodataka i pretragu posebno dizajniran za velike HPC (High Performance Computing) sustave pohrane. Autori ističu da je Brindexer dizajniran da poboljša performanse indeksiranja metapodataka u HPC sustavima pohrane, koristeći pristup koji se temelji na eng. Bloom filterima i eng. R-trees. Ovaj pristup omogućuje efikasno indeksiranje i pretragu metapodataka, čime se poboljšava ukupna performansa HPC sustava pohrane (8).

3.2. Ekstrakcija metapodataka

U radu „Rule Based Metadata Extraction Framework from Academic Articles“ Azimjonov i Alikhanov navode: "Metapodaci znanstvenih članaka kao što su naslov, sažetak, ključne riječi ili indeksni pojmovi, tijelo teksta, zaključak, referenca i drugi igraju odlučujuću ulogu u prikupljanju, upravljanju i pohranjivanju akademskih podataka u znanstvenim bazama podataka, akademskim časopisima i digitalnim knjižnicama. Točno izdvajanje ovih vrsta podataka iz znanstvenih radova ključno je za

organizaciju i dohvaćanje važnih znanstvenih informacija za istraživače, kao i za knjižničare."

Neki od važnih aspekata prilikom ekstrakcije metapodaka su:

1. Prilagodljivost - jedna od ključnih prednosti ovog okvira je njegova prilagodljivost. Autori ističu da se pravila za ekstrakciju mogu lako prilagoditi za različite vrste dokumenata i formate. To znači da se okvir može koristiti za ekstrakciju metapodataka iz različitih izvora, što ga čini vrlo fleksibilnim alatom (8).
2. Automatizacija - okvir omogućava automatizaciju procesa ekstrakcije metapodataka. To može značajno ubrzati proces obrade dokumenata i smanjiti potrebu za ručnim radom (8).
3. Preciznost - autori su testirali okvir na velikom broju akademskih članaka i otkrili da ima visoku preciznost. To znači da je okvir u stanju precizno identificirati i izvući relevantne metapodatke (8).
4. Primjena u različitim domenama - iako je okvir prvobitno razvijen za akademske članke, autori ističu da se može koristiti i u drugim domenama. Na primjer, može se koristiti za ekstrakciju metapodataka iz pravnih dokumenata, medicinskih izvještaja, novinskih članaka i drugih vrsta dokumenata (8).
5. Unapređenje pretraživanja informacija - korištenje ovog okvira može poboljšati pretraživanje informacija. Ekstrakcija metapodataka može omogućiti bolje kategoriziranje i indeksiranje dokumenata, što može olakšati pretraživanje i pristup relevantnim informacijama (8).

4. Motivacija

U suvremenom digitalnom svijetu, upravljanje i manipuliranje podacima postao je temeljni aspekt svakodnevnog života. Bez obzira na to radi li se o akademskom, poslovnom ili osobnom kontekstu, sve veći broj ljudi koristi sofisticirane alate za obradu podataka kako bi stvorili, obradili i prezentirali informacije.

Međutim, veliki broj tih alata koristi različite formate za pohranu i prikaz podataka, što često otežava razmjenu informacija između različitih platformi. Također, unutar samih datoteka često postoje skupovi podataka, poznati kao metapodaci, koji pružaju dodatne informacije o sadržaju datoteke. Te informacije mogu biti vrlo korisne, ali

često su teško dostupne jer standardni alati za pregledavanje datoteka ne prikazuju te podatke ili ih prikazuju na način koji nije prikladan za korisnika.

Motivacija za izradu računalne aplikacije za ekstrakciju metapodataka iz PPTX formata i konverziju u Markdown sintaksu proizlazi iz potrebe za jednostavnim, učinkovitim alatom koji omogućuje korisnicima da lako izvuku, manipuliraju i dijele podatke iz PowerPoint prezentacija. To bi moglo biti posebno korisno za ljude koji rade s velikim brojem PowerPoint prezentacija, kao što su predavači, studenti, istraživači ili poslovni analitičari.

5. SWOT analiza

SWOT analiza omogućuje da objektivno sagledamo snagu (eng. Strength), slabosti (eng. Weaknesses), prilike (eng. Opportunities) i prijetnje (eng. Threats) koje se odnose na aplikaciju za ekstrakciju metapodataka iz PPTX formata i konverziju u Markdown sintaksu.

5.1. Snaga

- Omogućava korisnicima da izvuku metapodatke iz PPTX formata, koji su inače teško dostupni.
- Konverzija metapodataka u lako čitljiv i lako urediv Markdown format, što omogućava veću fleksibilnost i portabilnost podataka.
- Jednostavno i intuitivno korisničko sučelje, što ga čini pristupačnim za korisnike različitih razina tehničke pismenosti.
- Koristi otvorene tehnologije, što znači da se aplikacija može lako nadograditi ili prilagoditi.

5.2. Slabosti

- Može biti ograničen u pogledu vrsta metapodataka koje može ekstrahirati iz PPTX formata.
- Neke složenije PPTX značajke možda neće biti u potpunosti podržane u Markdown formatu.
- Korisnici možda neće biti svjesni svih funkcionalnosti ili kako da ih

maksimalno iskoriste.

5.3. Prilike

- Postoji rastuća potreba za alatima koji omogućavaju bolje upravljanje i manipulaciju podacima, osobito u obrazovnim i poslovnim okruženjima.
- Aplikaciju je moguće nadograditi da podržava druge formate datoteka, osim PPTX.
- Moguća suradnja s obrazovnim institucijama ili poslovnim organizacijama koje bi mogle koristiti ovu aplikaciju u svojim svakodnevnim operacijama.

5.4. Prijetnje

- Postoji konkurencija od drugih alata za ekstrakciju metapodataka ili konverziju formata.
- Tehnološki napredak može dovesti do promjena u načinu na koji se podaci pohranjuju i dijele, što bi moglo učiniti aplikaciju manje relevantnom.
- Aplikacija je osjetljiva na potencijalne promjene u PPTX i Markdown specifikacijama.

6. Razrada funkcionalnosti

6.1. USE CASE Dijagram

Veze između korisnika i svakog od Use Case-ova predstavljaju interakciju koja se odvija između korisnika i aplikacije. Linije koje povezuju korisnika i Use Case-ove pokazuju da korisnik inicira svaki od tih procesa.

Slika 1 Use Case diagram, autorski rad

6.1.1. Akter – korisnik

Korisnik predstavlja osobu koja koristi aplikaciju za ekstrakciju metapodataka i konverziju u Markdown sintaksu.

6.1.2. Učitavanje PPTX datoteke

Korisnik pokreće Use Case dijagram odabirom i učitavanjem PPTX datoteke u aplikaciji. Ova interakcija predstavlja početak procesa ekstrakcije i konverzije.

6.1.3. Ekstrakcija metapodataka

Nakon što je PPTX datoteka učitana, aplikacija automatski ekstrahira metapodatke iz datoteke. Ovaj korak ne zahtijeva dodatnu interakciju korisnika.

6.1.4. Konverzija u Markdown

Ekstrahirani metapodaci se potom konvertiraju u Markdown sintaksu. Ovaj korak

također se automatski odvija unutar aplikacije.

6.1.5. Pregled i uređivanje Markdown-a

Korisnik sada ima mogućnost pregledati ekstrahirane i urediti metapodatke koji su konvertirani u Markdown formatu. Ovo omogućava korisniku da vidi konačan rezultat konverzije i po potrebi uredi po njegovoj želji.

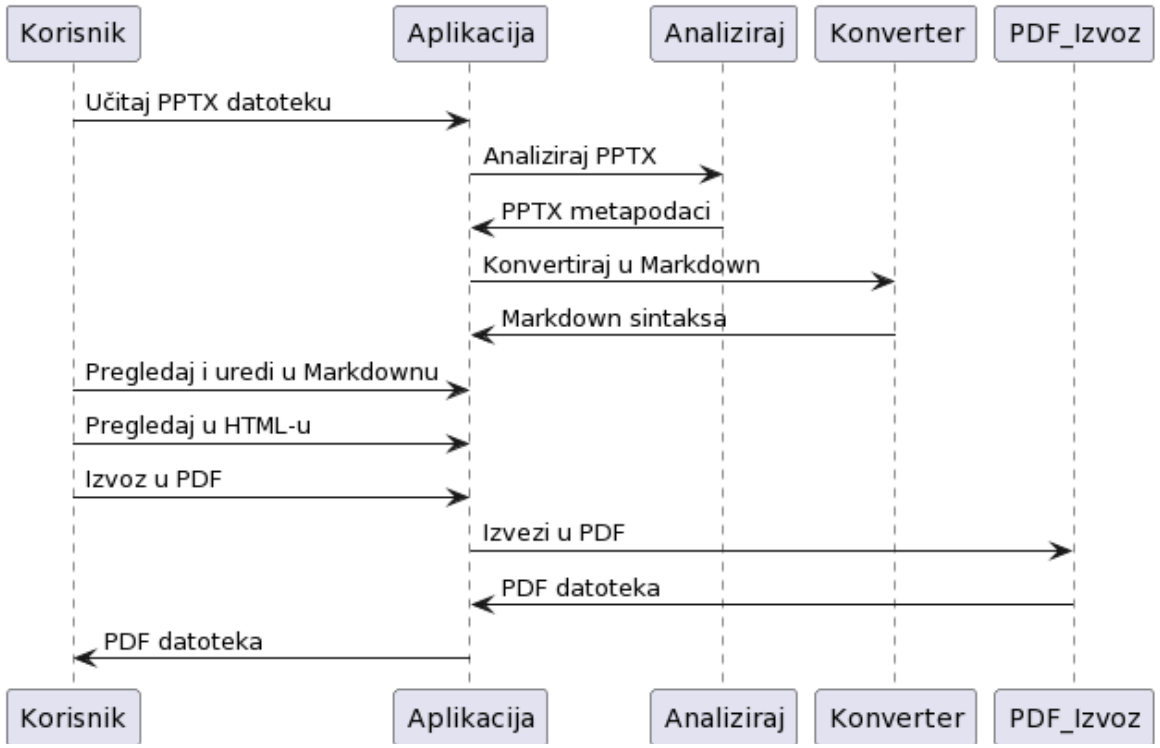
6.1.6. Pregled u HTML-u

Alternativno, korisnik može odabrati da pregleda konvertirane metapodatke u HTML formatu. To daje korisniku dodatnu fleksibilnost u pregledu konvertiranih podataka.

6.1.7. Izvoz u PDF

Konačno, korisnik može odabrati da izveze konvertirane metapodatke u PDF formatu. Ovo je posljednji korak koji omogućava korisniku da rezultate konverzije sprema za buduću upotrebu.

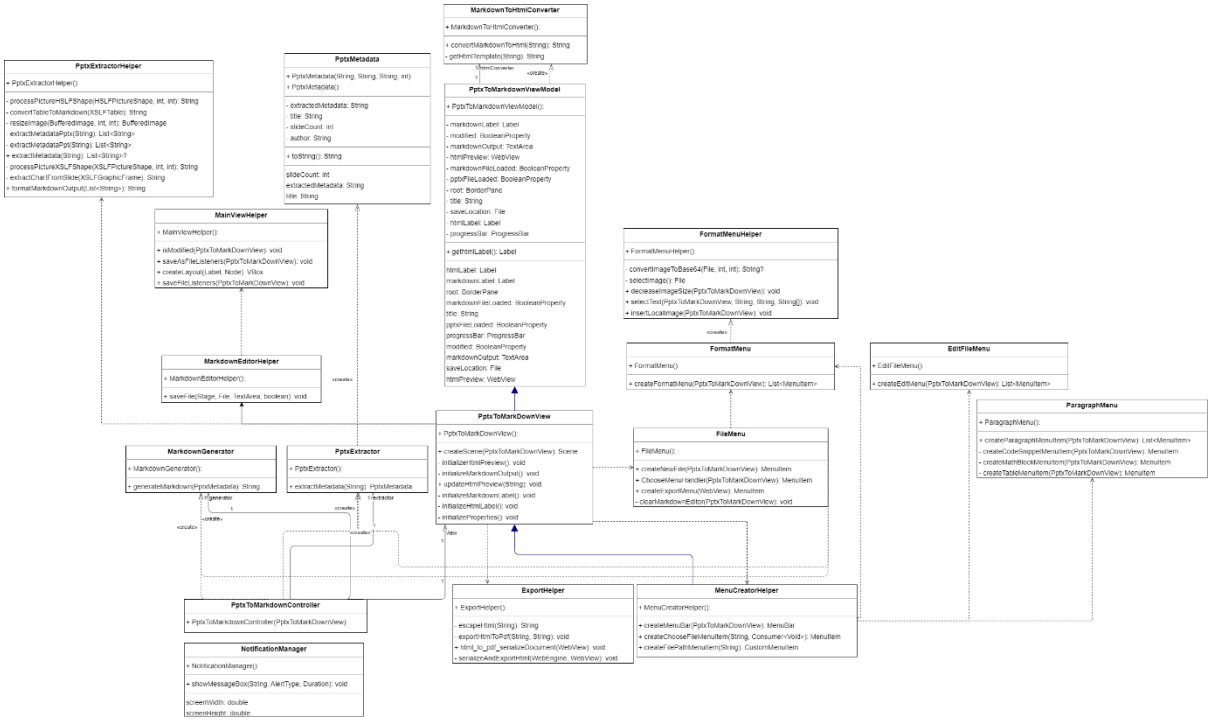
6.2. USE CASE SEQUENCE Dijagram



Slika 2 USE CASE SEQUENCE DIAGRAM, autorski rad

Korisnik započinje proces odabirom i učitavanjem željene PPTX datoteke. Nakon što je datoteka učitana, aplikacija koristi eng. Parser za ekstrakciju metapodataka iz PPTX datoteke. Zatim, eng. Parser vraća ekstrahirane metapodatke aplikaciji koju prosljeđuje konverteru. Nakon konverzije, konvertirani podaci se vraćaju u Markdown sintaksu. Korisnik zatim može pregledati ili urediti metapodatke, te iste ima kao pregled u HTML-u. Ako korisnik želi, može izvesti finalnu verziju u PDF.

6.3. UML CLASS Dijagram



Slika 3 UML Class Dijagram, autorski rad

6.3.1. Veze u klasnom dijagramu

Na temelju opisa dijagrama, možemo reprezentirati interakcije u UML dijagramu klasa kako slijedi:

a. Klasa "FormatMenu" posjeduje ovisnost prema klasi "FormatMenuHelper" s obzirom na to da koristi metode iz "FormatMenuHelper" klase. Ova veza se vizualizira kao isprekidana linija s otvorenim vrhom strelice koja izlazi iz klase "FormatMenu" i upućuje prema klasi "FormatMenuHelper".

b. Slično tome, klasa "MenuCreatorHelper" ovisi o klasi "FormatMenu" jer u svom kodu poziva "FormatMenu". Ovaj odnos se također prikazuje kao isprekidana linija s otvorenim vrhom strelice koja ide iz "MenuCreatorHelper" prema

"FormatMenu".

c. Klasa "MenuCreatorHelper" također ovisi o klasi "FileMenu" jer koristi "FileMenu" u svojim metodama. Ova veza se može vizualizirati kao isprekidana linija s otvorenim vrhom strelice koja ide iz "MenuCreatorHelper" prema "FileMenu".

d. Klasa "PptxToMarkdownView" ovisi o klasama "MenuCreatorHelper" i "FileMenu" jer koristi metode obiju klasa. Ovi odnosi se mogu prikazati kao isprekidane linije s otvorenim vrhovima strelica koji izlaze iz "PptxToMarkdownView" i upućuju prema "MenuCreatorHelper" i "FileMenu".

7. Implementacija

Za razvoj aplikacije "Aplikacija za ekstrakciju metapodataka iz PPTX formata i konverziju u Markdown sintaksu" korištena je IntelliJ IDEA, jedna od najpopularnijih integriranih razvojnih okruženja (Integrated Development Environments, IDE) za programiranje na Javi.

Kako bi se osiguralo stabilno i sigurno radno okruženje, za pokretanje i razvoj aplikacije korištena je Liberica JDK 11, distribucija OpenJDK-a koja je potpuno certificirana i usklađena s TCK-om za Java SE specifikaciju. Liberica JDK 11 pruža visokoučinkovito i sigurno okruženje za pokretanje Java aplikacija.

Konfiguracija projekta u IntelliJ IDEA-u započela je stvaranjem novog Java projekta. Nakon toga, postavljena je Liberica JDK 11 kao platformu za pokretanje projekta kroz Project Structure postavke. Zatim, kroz File > Project Structure > Modules > Dependencies dodane su potrebne vanjske biblioteke za rad s PPTX formatima, Markdown sintaksom, te generaciju PDF dokumenata.

7.1. Ulazna točka aplikacije

```
Tomislav-Troha
@Override
public void start(Stage primaryStage) {
    mainStage = primaryStage;
    PptxToMarkDownView view = new PptxToMarkDownView();
    PptxToMarkdownController controller = new PptxToMarkdownController(view);

    mainStage.getIcons().add(new Image( url: "/pptx-to-markdown-converter-logo.png"));

    Scene scene = view.createScene(view);
    scene.getStylesheets().add(getClass().getResource( name: "/styles.css").toExternalForm());

    primaryStage.setTitle("Pptx to Markdown Converter");
    primaryStage.setScene(scene);
    primaryStage.setMaximized(true);
    primaryStage.show();
}
}
```

Slika 4 Ulazna točka aplikacije (App.java), autorski rad

Ovaj dio kôda predstavlja ulaznu točku u JavaFX aplikaciju. Ona mora sadržavati klasu koja nasljeđuje "JavaFX Application" klasu, a metoda "start(Stage primaryStage) " predstavlja glavnu metodu koja se izvodi kada se pokrene aplikacija.

Kreira se instanca "view" klase "PptxToMarkDownView“ koja je odgovorna za definiranje izgleda korisničkog sučelja aplikacije. Zatim, sljedeća bitna stvar je da se kreira eng. scene objekt kojeg eng. stage (prozor aplikacije) prikazuje. U eng. scene objekt se dodaju elementi korisničkog sučelja.

7.2. Glavna scena aplikacije

```
1 inheritor  Tomislav-Troha
public class PptxToMarkDownView extends PptxToMarkdownViewModel {

    Tomislav-Troha
    public PptxToMarkDownView() {...}

    1 usage  Tomislav-Troha
    private void initializeMarkdownLabel() {...}
    1 usage  Tomislav-Troha
    private void initializeHtmlLabel() {...}
    1 usage  Tomislav-Troha
    private void initializeMarkdownOutput() {
        setMarkdownOutput(new TextArea());
        getMarkdownOutput().setPrefSize( prefWidth: 794,  prefHeight: 1123);
        getMarkdownOutput().setStyle("-fx-font-size: 18px;");
        getMarkdownOutput().setPromptText("Type markdown here...");
        getMarkdownOutput().setEditable(true);

        getMarkdownOutput().textProperty().addListener((observable, oldValue, newValue) -> updateHtmlPreview(newValue));
    }
    1 usage  Tomislav-Troha
    private void initializeHtmlPreview() {...}
    1 usage  Tomislav-Troha
    private void initializeProperties() {...}
```

Slika 5 Inicijalizacija svojstva aplikacije (PptxToMarkdownView.java), autorski rad

```
1 usage  Tomislav-Troha
public Scene createScene(PptxToMarkDownView view) {
    //create menu bar
    MenuBar menuBar = MenuCreatorHelper.createMenuBar(view);

    HBox fileInputLayout = new HBox( spacing: 10);
    VBox markdownLayout = MainViewHelper.createLayout(getMarkdownLabel(), getMarkdownOutput());
    markdownLayout.setAlignment(javafx.geometry.Pos.CENTER);

    VBox htmlLayout = MainViewHelper.createLayout(gethtmlLabel(), getHtmlPreview());
    htmlLayout.setAlignment(javafx.geometry.Pos.CENTER);

    SplitPane splitPane = new SplitPane(markdownLayout, htmlLayout);
    splitPane.setDividerPositions(0.5);

    VBox topLayout = new VBox(menuBar, fileInputLayout); // Combine the menuBar and fileInputLayout

    setRoot(new BorderPane());
    getRoot().setTop(topLayout);
    getRoot().setCenter(splitPane);
    getRoot().setBottom(getProgressBar());

    //if the markdownOutput is modified, set the modified flag to true
    getMarkdownOutput().textProperty().addListener((observable, oldValue, newValue) -> {
                getModified().set(true);
        getMarkdownFileLoaded().set(false);
    });

    MainViewHelper.isModified(view);

    return new Scene(getRoot());
}
```

Slika 6 Glavna metoda kreiranja scene, autorski rad

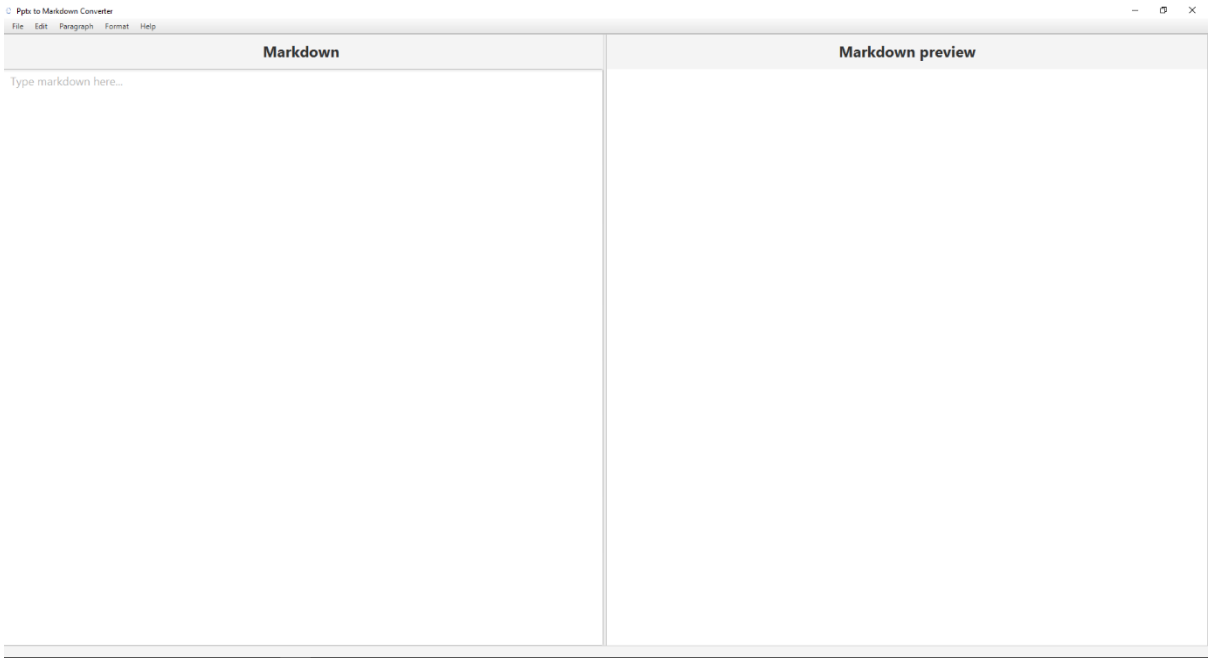
Na slici 5 se nalaze svojstva aplikacije koja se inicijaliziraju prilikom njenog pokretanja. Metoda "PptxToMarkDownView()" je konstruktor za klasu "PptxToMarkDownView". Ova metoda se poziva kada se stvara nova instanca ove klase.

Metoda "createScene(PptxToMarkDownView view)" koristi se za stvaranje glavnog prozora ili scene aplikacije. Scene predstavljaju kontejner za sve grafičke komponente koje se prikazuju na prozoru aplikacije.

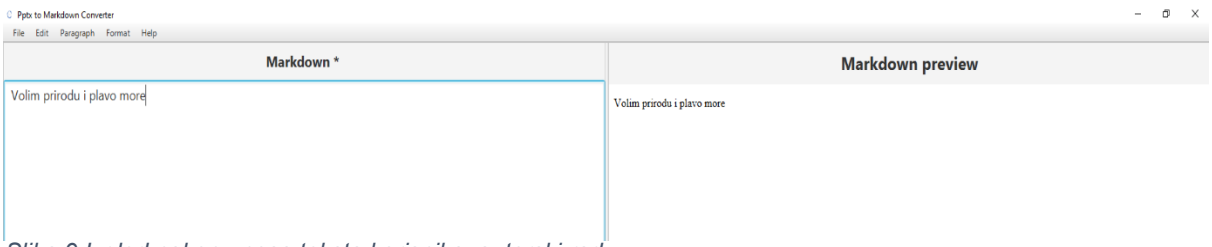
```
public void updateHtmlPreview(String markdown) {
    WebEngine webEngine = getHtmlPreview().getEngine();
    String cssFileUrl = getClass().getResource( name: "/markdown-preview.css").toExternalForm();
    webEngine.setUserStyleSheetLocation(cssFileUrl);
    String html = htmlConverter.convertMarkdownToHtml(markdown);
    webEngine.loadContent(html);
}
```

Slika 7 Ažuriraj pregled Markdown-a, autorski rad

Metoda "updateHtmlPreview()" služi za automatsko ažuriranje HTML pregleda svaki put kada se uneseni Markdown tekst promijeni. Ova funkcionalnost je ključna za korisničko iskustvo jer omogućava korisnicima da u realnom vremenu vide kako će njihov Markdown tekst izgledati kada se prevede u HTML. Bez ove funkcije, korisnici bi morali ručno pokretati pretvorbu svaki put kada bi željeli vidjeti ažurirani pregled. Stoga, ova funkcionalnost doprinosi dinamičnosti i interaktivnosti aplikacije, čineći je praktičnijom i intuitivnijom za korištenje.



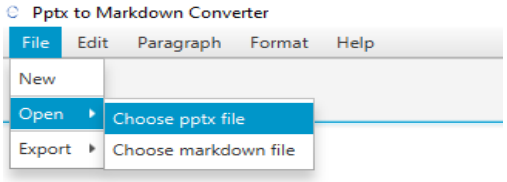
Slika 8 Početni izgled aplikacije, autorski rad



Slika 9 Izgled nakon unosa teksta korisnika, autorski rad

Na Slici 9 prikazan je unos teksta u polje za Markdown, s trenutnim prikazom kako će se taj unos manifestirati u konačnom pregledu. Budući da se u ovom slučaju radi o običnom tekstu, razlika nije vidljiva.

7.3. Učitavanje PPTX datoteke

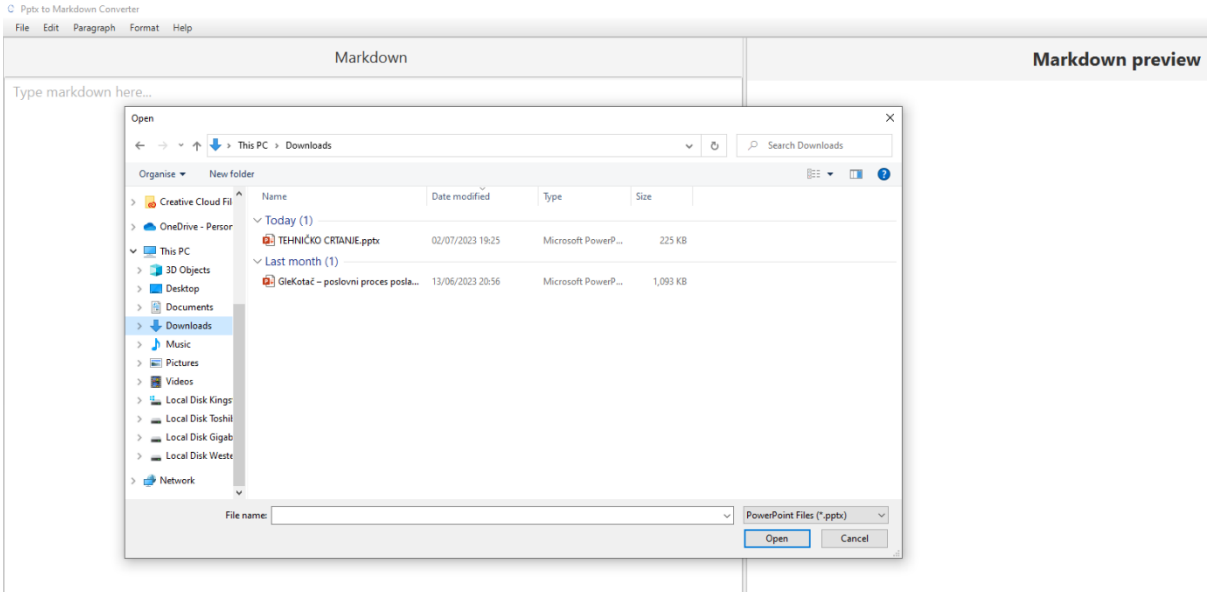


Slika 10 Odabir PPTX datoteke iz izbornika, autorski rad

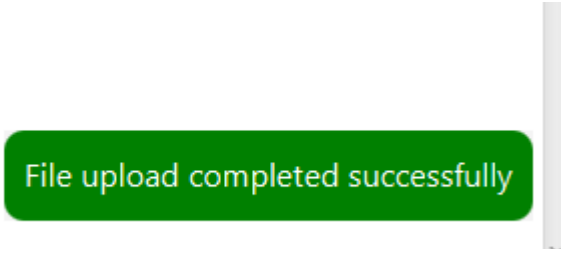
Učitavanje PPTX datoteke u aplikaciji se odvija putem opcije iz eng. File menija. Korisnik otvara eng. File meni klikom na "File" na glavnoj traci menija. Tu se prikazuje padajući meni s različitim opcijama, uključujući "Choose PPTX file" i "Choose markdown file". Ovaj događaj pokreće metodu za otvaranje eng. FileChooser dijaloga, koji korisniku omogućuje pretraživanje sustava datoteka i odabir željene PPTX datoteke. Kada korisnik odabere datoteku i klikne eng. "Open", metoda čita datoteku, ekstrahira metapodatke i konvertira sadržaj u Markdown sintaksu.

```
Task<Void> task = new Task<>() {
    @Override
    protected Void call() {
        PptxMetadata metadata = extractor.extractMetadata(selectedFile.getAbsolutePath());
        String markdown = generator.generateMarkdown(metadata);

        Platform.runLater(() -> {
            view.getMarkdownOutput().setText(markdown);
            view.updateHtmlPreview(markdown);
        });
        return null;
    }

    @Override
    protected void succeeded() {
        super.succeeded();
        NotificationManager.showMessageBox( message: "File upload completed successfully", Alert.AlertType.INFORMATION, Duration.seconds( s: 3));
        view.setSaveLocation(null);
        view.getPptxFileLoaded().set(true);
        progressBar.setVisible(false);
    }

    @Override
    protected void failed() {
        super.failed();
        try {
            progressBar.setVisible(false);
            throw new Exception("Error while uploading file!");
        } catch (Exception e) {
            NotificationManager.showMessageBox(e.getMessage(), Alert.AlertType.ERROR, Duration.seconds( s: 3));
        }
    }
};
new Thread(task).start();
}
} catch (Exception e) {
    NotificationManager.showMessageBox( message: e + " " + e.getMessage(), Alert.AlertType.ERROR, Duration.seconds( s: 3));
```

Slika 11 Dio metode za učitavanje PPTX datoteke, autorski rad



Slika 12 Odabir PPTX datoteke iz računala, autorski rad



Slika 13 Prikaz da je datoteka uspješno učitana

7.4. Ekstrakcija metapodataka

Nakon što se sadržaj datoteke pročita, aplikacija zatim ekstrahira metapodatke iz datoteke. Ovo uključuje informacije poput naslova slajda, sadržaja, bilješki, autora, vremena stvaranja i drugih metapodataka povezanih sa PPTX datotekom. Ove informacije se zatim pohranjuju u odgovarajuće objekte za daljnju obradu. Sadržaj svakog slajda, zajedno s ekstrahiranim metapodacima, se zatim konvertira u Markdown sintaksu. Ovaj korak uključuje transformaciju formatiranja teksta, eng. bullet lista, tablica, hiperlinka, itd., u ekvivalentni Markdown format.

U procesu ekstrakcije metapodataka iz PPTX datoteka, upotrijebljen je značajan broj metoda i algoritama. Ove metode obuhvaćaju složen proces čitanja i obrade binarnih datoteka, pravilno dekodiranje različitih tipova sadržaja unutar datoteka (uključujući tekst, slike, tablice i drugo), te konverziju tih sadržaja u Markdown

format.

Kako bi se održala konciznost i jasnoća ovog rada, neće biti prikazan cijeli kôd uključen u ovaj postupak. Razlog za to leži u činjenici da sam kôd može biti prilično obiman i specifičan za biblioteke koje su upotrijebljene (Apache POI). Detaljni prikaz kôda bi mogao ometati čitatelja od ključnih koncepta i svrhe rada, stoga je prikazan manji dio ekstrakcije.

```
//extract images from pptx
if (shape instanceof XSLFPictureShape) {
    //process image
    String base64Image = processPictureXSLFShape((XSLFPictureShape) shape, targetWidth: 350, targetHeight: 300);

    //get image type
    String pictureData = ((XSLFPictureShape) shape).getPictureData().getContentType();
    String markdownFormat = "![](data:%s;base64,%s)";

    String imageTag = String.format(markdownFormat, pictureData, base64Image);

    stringSlideContents.append("\n\n").append(imageTag);
}
```

Slika 14 Prikaz ekstrakcije metapodataka iz slike iz PPTX datoteke, autorski rad

```
//helper method for extract images from pptx
1 usage  Tomislav-Troha
private static String processPictureXSLFShape(XSLFPictureShape pictureShape, int targetWidth, int targetHeight) throws IOException {
    XSLFPictureData pictureData = pictureShape.getPictureData();
    BufferedImage originalImage = ImageIO.read(new ByteArrayInputStream(pictureData.getData()));
    BufferedImage resizedImage = resizeImage(originalImage, targetWidth, targetHeight);

    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    String imageFormat = pictureData.getContentType().split( regex: "/")[1];
    ImageIO.write(resizedImage, imageFormat, baos);

    return Base64.getEncoder().encodeToString(baos.toByteArray());
}
```

Slika 15 Pomoćna metoda za ekstrakciju metapodatka iz slike, autorski rad

Ovaj dio kôda posvećen je obradi i konverziji slika unutar PPTX datoteka u format koji je kompatibilan s Markdown sintaksom. Na samom početku, provjerava se je li određeni element (eng. shape) na slajdu slika, primjenjujući "instanceof" operator za utvrđivanje tipa objekta - u ovom slučaju, provjerava radi li se o objektu tipa XSLFPictureShape. Ako je to slučaj, slika se dalje obrađuje.

Metoda "processPictureXSLFShape" koristi se za obradu slika. Ulazni parametar ove metode je objekt "XSLFPictureShape", koji predstavlja sliku unutar PPTX datoteke. Također, metodi se prosljeđuju ciljane širine i visine, koje se koriste za određivanje dimenzija slike nakon što je njena veličina promijenjena.

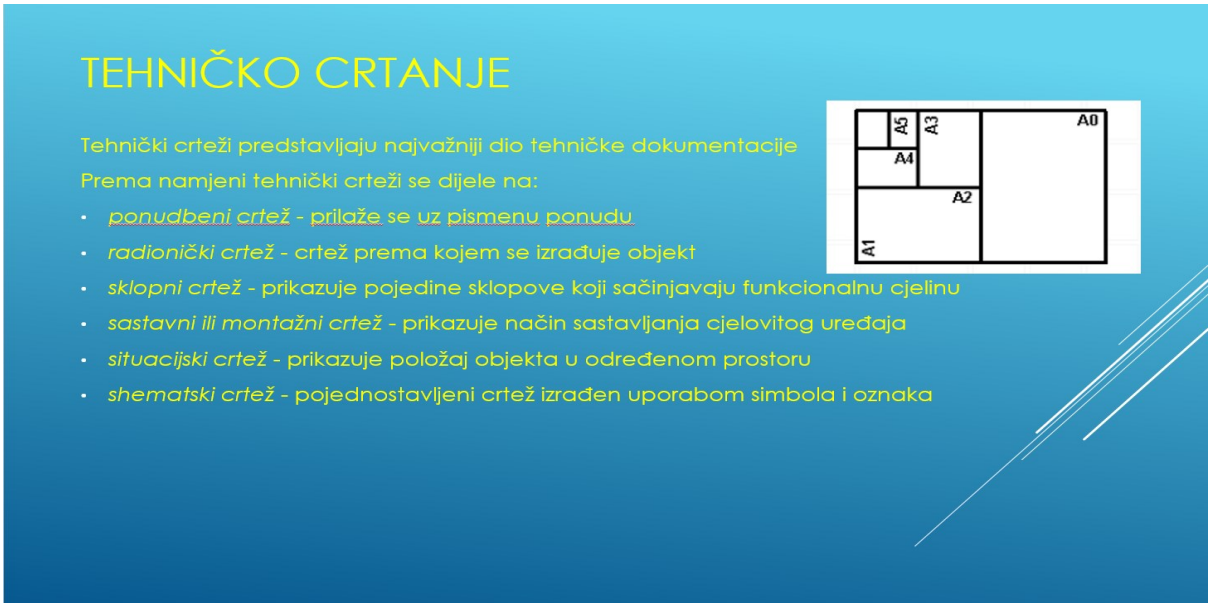
Iz objekta "XSLFPictureShape" ekstrahiraju se podaci o slici (XSLFPictureData), koji se zatim koriste za stvaranje objekta "BufferedImage", odnosno originalne slike. Ova slika se potom skalira korištenjem metode

"resizeImage" kako bi odgovarala ciljanim dimenzijama.

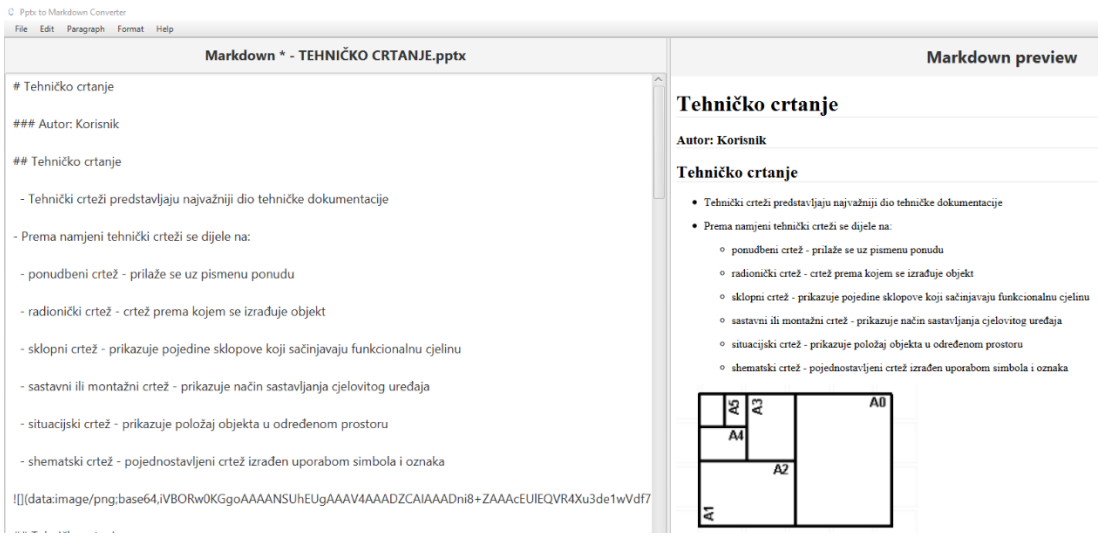
Nakon što je veličina slike promijenjena, primjenjuje se metoda "ImageIO.write" koja upisuje novu sliku u objekt "ByteArrayOutputStream". Izlazni podaci ovog toka kodiraju se u Base64 format pomoću metode "Base64.getEncoder().encodeToString".

Krajnji rezultat je slika kodirana u Base64 formatu, koja se koristi za kreiranje Markdown tag-a za sliku. Ovaj tag koristi specifični format koji omogućava ugrađivanje slika direktno u tekst kroz upotrebu Base64 kodirane slike.

Na posljetku, ovaj Markdown tag slike integrira se u sadržaj slajda, koji se potom konvertira u Markdown format. Na taj način, sve slike unutar PPTX datoteka mogu se uspješno konvertirati i prikazati unutar konvertiranog Markdown sadržaja.



Slika 16 Primjer slide-a sa prezentacije, autorski rad



Slika 17 Markdown sintaksa i pregled nakon ekstrakcije metapodataka, autorski rad

7.5. Pregled Markdown sintakse

```
3 usages  Tomislav-Troha *
public class MarkdownToHtmlConverter {
    1 usage  Tomislav-Troha *
    public String convertMarkdownToHtml(String markdown) {

        MutableDataSet options = new MutableDataSet();
        options.set(Parser.EXTENSIONS, List.of(TablesExtension.create()));

        Parser parser = Parser.builder(options).build();
        HtmlRenderer renderer = HtmlRenderer.builder(options).build();

        Document document = parser.parse(markdown);
        String html = renderer.render(document);

        String finalHtml = null;
        try {
            finalHtml = getHtmlTemplate(html);
        } catch (IOException e) {
            e.printStackTrace();
        }
        return finalHtml;
    }

    1 usage  Tomislav-Troha
    private String getHtmlTemplate(String content) throws IOException {
        InputStream inputStream = getClass().getResourceAsStream( name: "/main-html.html");
        BufferedReader reader = new BufferedReader(new InputStreamReader(inputStream));
        StringBuilder htmlTemplate = new StringBuilder();
        String line;

        while ((line = reader.readLine()) != null) {
            htmlTemplate.append(line);
        }A

        String html = htmlTemplate.toString();
        html = html.replace( target: "CONTENT", content);

        return html;
    }
}
```

Slika 18 Klasa za konverziju Markdown sintakse u HTML pregled, autorski rad

Klasa "MarkdownToHtmlConverter" koristi se za konverziju sadržaja pisanih u Markdown formatu u HTML. Ova klasa ima ključnu ulogu u prikazivanju Markdown sadržaja unutar korisničkog sučelja, omogućujući korisnicima da imaju vizualni pregled u realnom vremenu. Proces konverzije koristi nekoliko ključnih komponenti:

- "MutableDataSet" koristi se za postavljanje opcija za parser. U ovom

kontekstu, koristi se za omogućavanje podrške za Markdown tablice putem "TablesExtension".

- "Parser" i "HtmlRenderer" koriste se za konverziju Markdown sadržaja te njegovo prevođenje u HTML format.

Nakon što je HTML generiran, koristi se HTML predložak (main-html.html) kako bi se pružila potrebna struktura za sadržaj. Ovaj predložak se učitava iz datoteke resursa te se sadržaj uklapa unutar njega koristeći metodu "getHtmlTemplate".

Metoda "getHtmlTemplate" čita HTML predložak iz datoteke, zamjenjuje mjesto označeno kao "CONTENT" sa stvarnim sadržajem, i vraća finalni HTML string. Treba napomenuti da, ukoliko se prilikom čitanja HTML predloška dogodi greška, metoda će izbaciti "IOException". Ova iznimka se detektira i greška se ispisuje unutar metode "convertMarkdownToHtml".

Sveukupno, klasa "MarkdownToHtmlConverter" predstavlja vitalni dio aplikacije, omogućujući korisnicima da vizualiziraju kako će njihov Markdown sadržaj izgledati u HTML formatu.

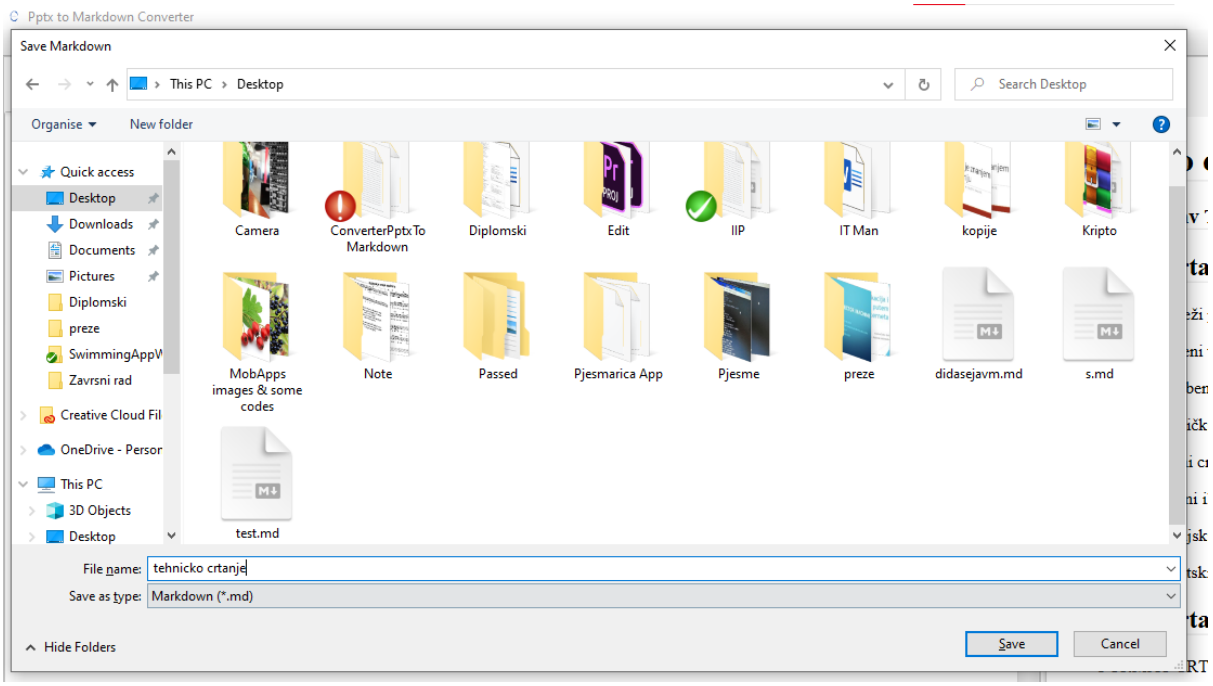
```
<!DOCTYPE html>
<html>
<meta charset="UTF-8" />
<head>
    <link href="resources:markdown-preview.css" rel="stylesheet" media="print" />
    <style>
      body {
        font-family: Arial Unicode MS;
        text-align: left;
      }
    </style>
    <script type="text/javascript" async src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/3.2.0/es5/tex-mml-chtml.js"></script>

    <style>
    img { -webkit-user-drag: none; user-drag: none; }

    h1, h2, h3, h4, h5, hó {
      border-bottom: 1px solid rgba(0, 0, 0, 0.1);
    }
    </style>

</head>
<body>
CONTENT
</body>
</html>
```

Slika 19 Predložak HTML-a, autorski rad

7.6. Proces spremanja Markdown sintakse

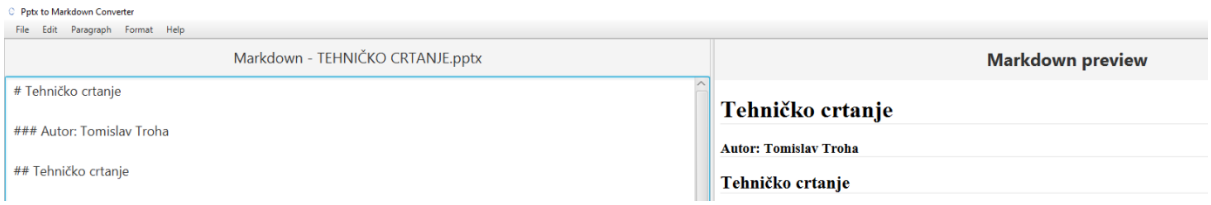
Kada korisnik klikne na opciju "Save" u izborniku ili koristi prečac CTRL+S, aktivira se akcija spremanja. U kontekstu ove aplikacije, postupak spremanja obuhvaća sljedeće korake.

Aplikacija prvo provjerava da li je korisnik izmijenio trenutni sadržaj od posljednjeg spremanja. Ako nije bilo promjena, nema potrebe za ponovnim spremanjem istog sadržaja, pa aplikacija preskače korake spremanja. Na slici 17 se jasno vidi da je prezentacija učitana i da promjene nisu spremljene.



Slika 20 Spremanje Markdown sintakse na disk računala, autorski rad

Spremljena datoteka se sad može otvoriti u nekom online uređivaču Markdown-a ili slično. Nakon što je sadržaj uspješno spremljen, stanje aplikacije se ažurira. Označava se da je trenutni sadržaj spremljen.



Slika 21 Ažurirano stanje aplikacije, autorski rad

7.7. Uređivanje Markdown sintakse

Dio aplikacije predviđen za uređivanje Markdown-a obuhvaća korisničko sučelje koje korisnicima omogućava manipulaciju Markdown dokumentom.

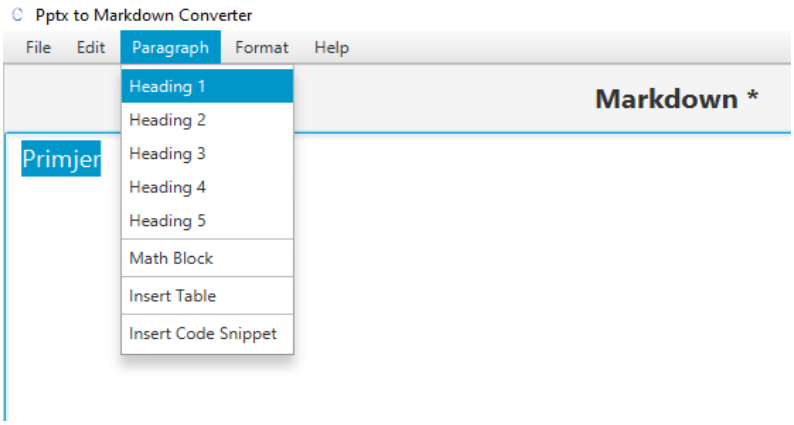
Funkcionalnosti ovog segmenta grupirane su u "MenuBar" koji se nalazi na vrhu aplikacije i sadrži razne opcije u padajućim izbornicima, poput "File", "Edit", "Format", i slično.

Izbornik "File" uključuje opcije kao što su "Open" "Save" i "Export", koje korisnicima omogućavaju otvaranje postojećih, spremanje trenutačnih i izvoz dokumenata pdf format.

Izbornik „Edit“ obuhvaća opcije poput "Undo", "Redo", "Cut", "Copy", "Paste", itd., koje korisnici mogu upotrijebiti za manipulaciju tekstom u dokumentu.

Izbornik „Format“ nudi niz opcija za formatiranje teksta, uključujući "Bold", "Italic", "Underline", "Strike" i slično. Svaka od ovih opcija automatski dodaje odgovarajuće Markdown oznake u tekst, olakšavajući korisnicima formatiranje vlastitog teksta bez potrebe za ručnim unošenjem Markdown sintakse.

Nadalje, funkcionalnost "Paragraph" omogućava korisniku lako unošenje novih naslova, matematičkih formula, tablica i dio kôda.

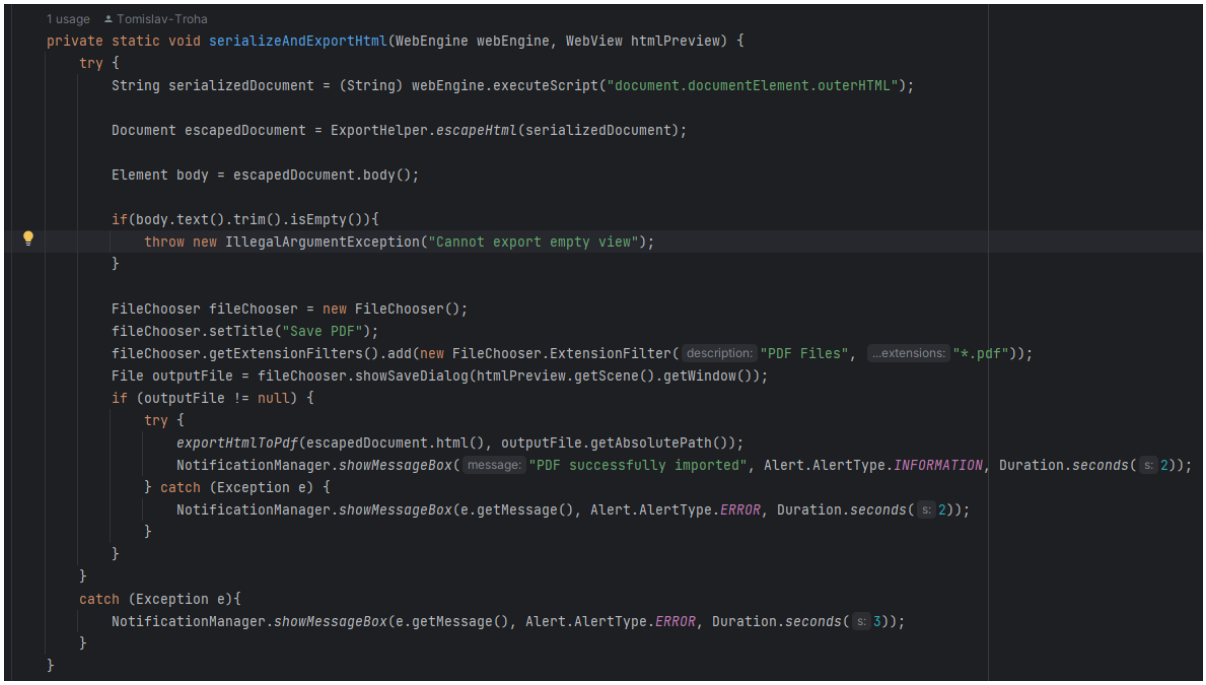


Slika 22 Paragraf funkcija za dodavanje naslova, autorski rad

Klikom na "Heading 1" aplikacija je dodala ispred označenog teksta odgovarajući znak, u ovom slučaju znak "#" te automatski napravila Markdown naslov.

7.8. Izvoz u PDF format

Izvoz dokumenta u PDF format postiže se koristeći funkcionalnost "Export to PDF" dostupnu u glavnom izborniku aplikacije pod opcijom "File". Kada korisnik odabere ovu opciju, otvara se dijalog za spremanje datoteke gdje korisnik može odabrati željenu lokaciju i ime za izlaznu PDF datoteku.



```
1 usage  Tomislav-Troha
private static void serializeAndExportHtml(WebEngine webEngine, WebView htmlPreview) {
    try {
        String serializedDocument = (String) webEngine.executeScript("document.documentElement.outerHTML");

        Document escapedDocument = ExportHelper.escapeHtml(serializedDocument);

        Element body = escapedDocument.body();

        if(body.text().trim().isEmpty()){
            throw new IllegalArgumentException("Cannot export empty view");
        }

        FileChooser fileChooser = new FileChooser();
        fileChooser.setTitle("Save PDF");
        fileChooser.getExtensionFilters().add(new FileChooser.ExtensionFilter( description: "PDF Files",  ...extensions: "*.pdf"));
        File outputFile = fileChooser.showSaveDialog(htmlPreview.getScene().getWindow());
        if (outputFile != null) {
            try {
                exportHtmlToPdf(escapedDocument.html(), outputFile.getAbsolutePath());
                NotificationManager.showMessageBox( message: "PDF successfully imported", Alert.AlertType.INFORMATION, Duration.seconds( s: 2));
            } catch (Exception e) {
                NotificationManager.showMessageBox(e.getMessage(), Alert.AlertType.ERROR, Duration.seconds( s: 2));
            }
        }
    }
    catch (Exception e){
        NotificationManager.showMessageBox(e.getMessage(), Alert.AlertType.ERROR, Duration.seconds( s: 3));
    }
}
```

Slika 23 Metoda za izvoz u pdf, 1.dio, autorski rad

```
1 usage  Tomislav-Troha
private static void exportHtmlToPdf(String htmlContent, String outputFile) throws Exception {
    try (OutputStream os = new FileOutputStream(outputFile)) {
        PdfRendererBuilder builder = new PdfRendererBuilder();
        builder.useFastMode();
        builder.useFont(new File(Main.class.getClassLoader().getResource( name: "export_fonts/Arial_Unicode_MS.ttf").getFile()),  fontFamily: "Arial Unicode MS");
        builder.withW3cDocument(new W3CDom().fromJsoup(Jsoup.parse(htmlContent)),  baseUri: null);
        builder.toStream(os);
        builder.run();
    }
}
```

Slika 24 Metoda za izvoz u pdf, 2.dio, autorski rad

Funkcionalnost izvoza u PDF temelji se ključnoj biblioteci - "PdfRendererBuilder". Koristi za pretvaranje HTML sadržaja u PDF.

Izvorni tekst unesen u aplikaciju prvo se obrađuje kako bi se generirao odgovarajući sadržaj za PDF. Osim samog teksta, prilikom izvoza u PDF također je važno obratiti pažnju na izgled dokumenta. Ovo uključuje postavke kao što su margine stranica, veličina i tip fonta, razmak između redaka i druge detalje koji utječu na izgled dokumenta. Spremanje dokumenta: Nakon što je sadržaj generiran i izgled dokumenta

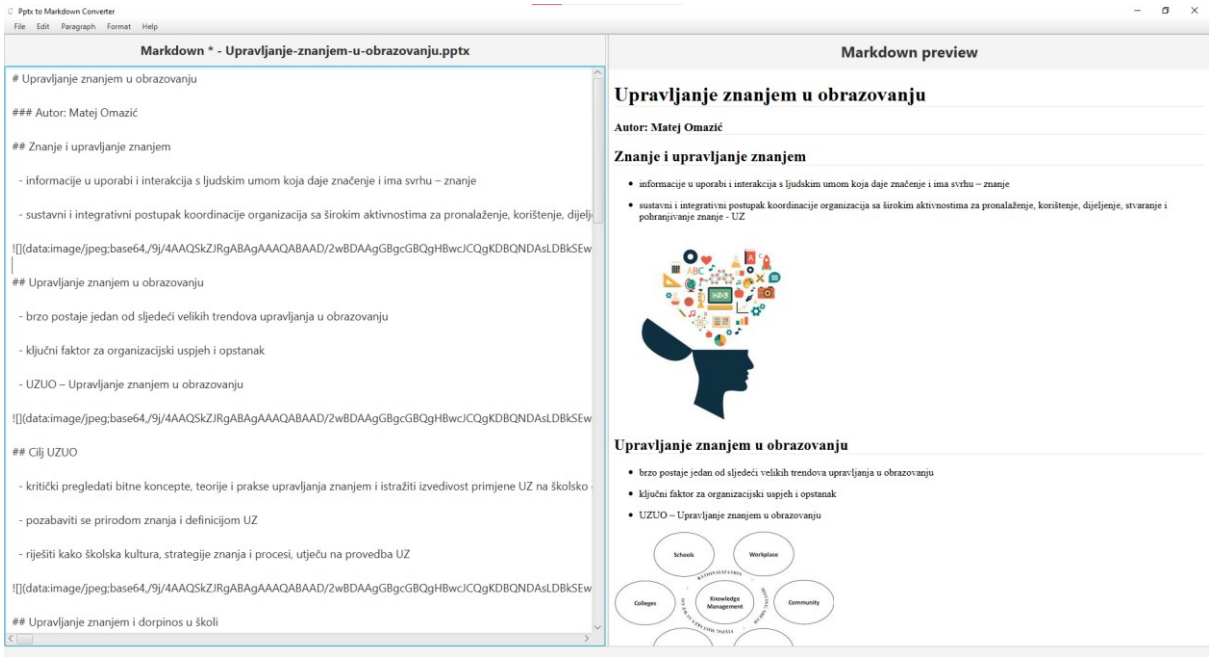
postavljen, stvoreni PDF dokument sprema se na lokaciju koju je korisnik odabrao.

Izvoz u PDF omogućuje korisnicima da prenose i dijele svoje radove na siguran i konzistentan način, osiguravajući da će dokument izgledati isto bez obzira na platformu na kojoj se otvara.

8. Korisničke upute

8.1. Otvaranje PPTX datoteke

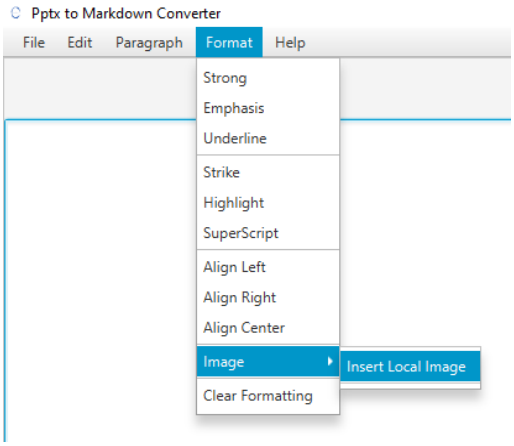
Nakon pokretanja aplikacije, korisničko sučelje će se prikazati na zaslonu. Gornji izbornik aplikacije sadrži nekoliko opcija, uključujući "File", "Edit", "Format" i "View" (Slika 8). Da bi se učitala PPTX datoteka, potrebno je kliknuti na "File" u gornjem izborniku. Nakon što se padajući izbornik otvori, odaberite opciju "Choose PPTX File" (Slika 10). Odmah nakon, otvorit će se prozor za odabir datoteke. U ovom prozoru mogu se pregledavati datoteke na vašem računalu (Slika 12). Nakon odabira prezentacije, aplikacija će automatski pretvoriti sadržaj u Markdown format. Ova pretvorba uključuje ekstrakciju teksta, slika i metapodataka iz prezentacije i njihovo formatiranje u odgovarajuću Markdown sintaksu. Ako prezentacija sadrži složene elemente, kao što su tablice ili matematički izrazi, oni će se također adekvatno prevesti. Isto tako, otvoriti se može direktno Markdown datoteka (Slika 10).



Slika 25 Primjer učitane prezentacije, autorski rad

8.2. Uređivanje teksta

Uređivanje teksta u aplikaciji jednostavno je i intuitivno, pružajući korisnicima potpunu kontrolu nad izgledom i strukturom Markdown sadržaja. Korisnik može odabrati dio teksta koji želi urediti pomoću kursora. Nakon što je tekst odabran, na raspolaganju su razne opcije za formatiranje dostupne u gornjem izborniku aplikacije. Te opcije uključuju "Bold" za podebljavanje teksta, "Italic" za korištenje kurzivnih slova, "Underline" za podvlačenje, "Strike" za precrtavanje teksta i "Highlight" za isticanje teksta. Osim toga, moguće je koristiti i opcije "Left align", "Center align" i "Right align" za poravnanje teksta. Kada se odabere neka od opcija za formatiranje, aplikacija automatski dodaje odgovarajuću Markdown sintaksu oko odabranog teksta. Na primjer, ako je odabrana opcija "Bold", aplikacija će dodati dva zvjezdasta znaka (**) na početak i kraj odabranog teksta. Osim osnovnih opcija formatiranja, dostupne su i druge opcije poput "Insert table", "Insert code snippet" i "Insert Math Block" za dodavanje specifičnih struktura u Markdown tekst. Također, pruža se mogućnost ručnog uređivanja. Korisnik može direktno pisati u editoru, koristeći Markdown sintaksu za formatiranje teksta prema vlastitim potrebama. Nakon uređivanja, korisnik može pregledati svoj tekst u "Preview" načinu rada. Ova opcija omogućava korisniku da vidi kako će tekst izgledati kada se prikaže kao formatirani Markdown, pružajući vizualni pregled svih promjena formatiranja koje su napravljene. Nakon završetka uređivanja, sve promjene mogu se spremiti odabirom opcije "Save" ili prečicom "Ctrl + S".

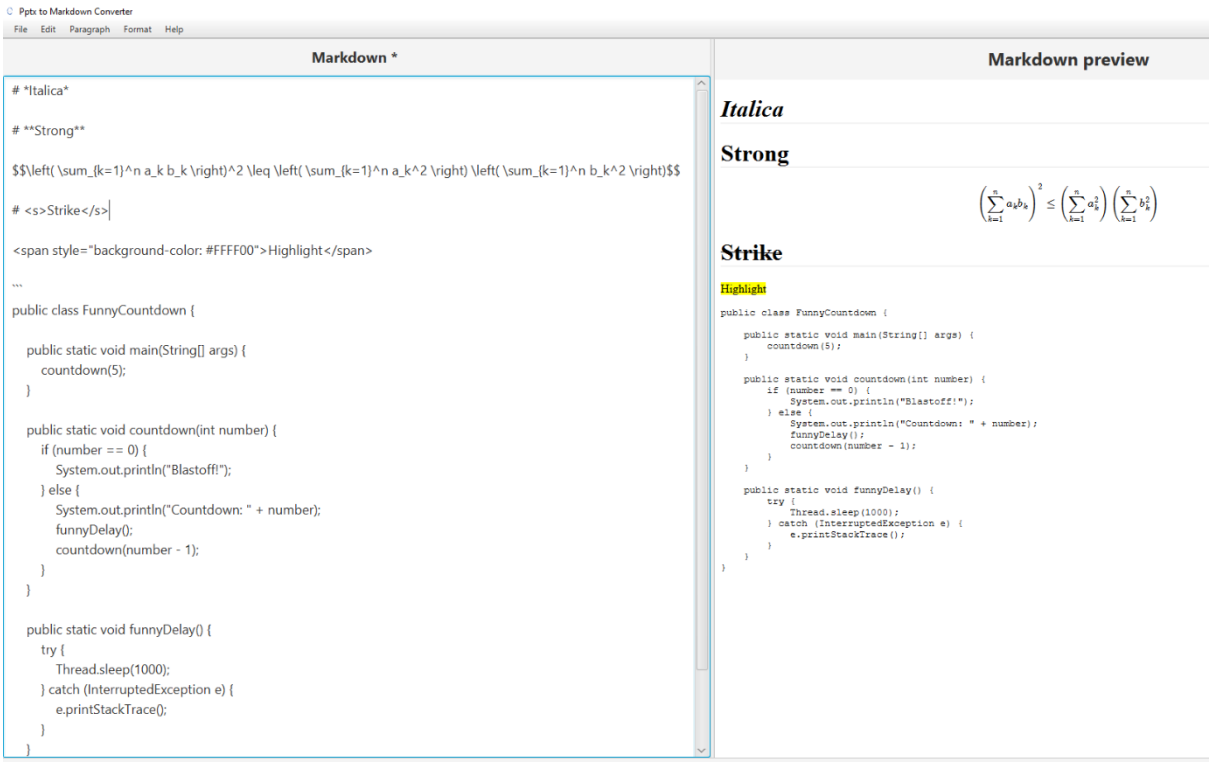


Slika 26 Dodavanje svoje slike u Markdown, autorski rad

Pored prethodno spomenutih opcija uređivanja, korisniku je omogućeno ubacivanje vlastitih slika direktno u Markdown tekst. Realizacija te funkcionalnosti ostvaruje se putem "Format" -> "Image" -> "Insert Local Image" u izborniku. Nakon što korisnik odabere tu opciju, prikazat će se dijaloški prozor koji omogućava pregled i odabir slike s lokalnog diska.

Prilikom selekcije slike, aplikacija automatski generira i ubacuje adekvatnu Markdown sintaksu za prikaz slike.

Na slici 26 se vidi da aplikacija nudi i značajku za brzo uklanjanje svih formata s odabranog teksta, vraćajući ga u njegov izvorni, neformatirani oblik. Funkcionalnost pod nazivom "Clear Format" dostupna je u izborniku pod "Format" -> "Clear Format". Po njenom odabiru, svi Markdown formati (npr. podebljano, kurzivno, podcrtano, naslovi, tablice itd.) bit će eliminirani iz trenutno odabranog teksta. Ova funkcionalnost posebno je korisna kada korisnik želi brzo očistiti određeni segment teksta od svih formata, vraćajući ga na izvorni oblik.



Slika 27 Prikaz nekih od mogućnosti uređivanja teksta, autorski rad

Zaključak

U svijetu digitalne komunikacije, Markdown je postao popularan zbog svoje jednostavnosti i čitljivosti. Njegova integracija s modernim alatima za stvaranje sadržaja, kao što je PowerPoint, otvara brojne mogućnosti za stvaranje efikasnih i vizualno privlačnih prezentacija.

U sklopu ovog diplomskog rada, razvijena je aplikacija koja integrira funkcionalnosti PowerPointa i Markdown-a, omogućavajući korisnicima da jednostavno pretvore svoje PowerPoint prezentacije u Markdown format. Dodatno, aplikacija omogućava uređivanje i prilagođavanje Markdown-a teksta, dodajući mu funkcionalnosti poput umetanja slika, tablica, blokova kôda i drugih formata, koji su uobičajeni u modernim digitalnim dokumentima. Koristeći niz sofisticiranih biblioteka i algoritama, aplikacija pruža korisnicima mogućnost da pretvore svoje Markdown dokumente u PDF format.

Zaključno, ovaj diplomski rad predstavlja značajan doprinos razvoju alata za stvaranje sadržaja, kombinirajući funkcionalnosti PowerPointa, Markdown-a i PDF formata u jednom intuitivnom sučelju. Njegova primjena može značajno olakšati stvaranje, uređivanje i dijeljenje prezentacija i digitalnih dokumenata, čime se poboljšava efikasnost i produktivnost korisnika. Kroz daljnji razvoj i poboljšanje, ova aplikacija ima potencijal da postane standardni alat za stvaranje sadržaja u digitalnom dobu.

Popis literature

1. https://web.archive.org/web/20210813193857/https://www.computerweekly.com/feature/Write-once-run-anywhere (Langley, 2002) [Pristupljeno: 4.5.2023.]
2. https://poi.apache.org/ [Pristupljeno: 4.5.2023.]
3. https://github.com/vsch/flexmark-java [Pristupljeno: 4.5.2023.]
4. https://openjfx.io/ [Pristupljeno: 4.5.2023.]
5. https://tools.ietf.org/html/rfc4648 [Pristupljeno: 4.5.2023.]
6. https://www.jetbrains.com/idea/ [Pristupljeno: 5.5.2023.]
7. https://github.com/danfickle/openhtmltopdf [Pristupljeno: 5.5.2023.]
8. Paul, S., Wang, T., Radke, T., Sim, A., & Butt, A. R. (2020). Efficient Metadata Indexing for HPC Storage Systems. In Proceedings of the 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID). https://dblp.org/rec/conf/ccgrid/PaulWRSB20 [Pristupljeno: 7.5.2023.]
9. William Y. Arms, Christophe Blanchi, Edward A. Overly - An Architecture for Information in Digital Libraries [Pristupljeno: 10.5.2023.]
10. National Information Standards Organization, 2004, https://web.archive.org/web/20141107022958/http://www.niso.org/publications/press/UnderstandingMetadata.pdf [Pristupljeno: 10.5.2023.]
11. Dippo, Cathryn; Sundgren, Bo. "The Role of Metadata in Statistics"
12. Frey, J., & Hellmann, S. (2017). MaSQue: An Approach for Flexible Metadata Storage and Querying in RDF. In Proceedings of the International Conference on Semantic Systems (i-Semantics). https://dblp.org/rec/conf/i-semantics/FreyH17 [Pristupljeno: 11.5.2023.]
13. Gonzalez-Perez C (2018). "Metainformation". In Gonzalez-Perez C (ed.). Information modelling for archaeology and anthropology: software engineering principles for cultural heritage (1st ed.) [Pristupljeno: 15.5.2023.]

Popis slika

Sažetak

Ovaj diplomski rad fokusira se na razvoj aplikacije koja omogućava konverziju PowerPoint prezentacija u Markdown format i njihovo daljnje uređivanje. Alat integrira funkcionalnosti PowerPointa i Markdown-a, omogućavajući transformaciju prezentacija u čitljive i lako prilagodljive Markdown dokumente. Pretvorba uključuje različite elemente kao što su slike, tablice i blokovi kôda. Razvijena je mogućnost izvoza Markdown dokumenata u PDF format, koristeći kombinaciju Flexmark, Flying Saucer i PdfRendererBuilder biblioteka. Rezultat ovog rada je efikasna aplikacija koja predstavlja značajan doprinos alatima za stvaranje digitalnog sadržaja, olakšavajući stvaranje, uređivanje i dijeljenje prezentacija i dokumenata.

Ključne riječi: PowerPoint prezentacije, Markdown format, konverzija dokumenta, uređivanje teksta, PDF izvoz

This thesis focuses on the development of an application that enables the conversion of PowerPoint presentations into Markdown format and their subsequent editing. The tool integrates PowerPoint and Markdown functionalities, allowing the transformation of presentations into readable and easily adjustable Markdown documents. The conversion includes various elements such as images, tables, and code blocks. Additionally, the ability to export Markdown documents into PDF format was developed, utilizing a combination of Flexmark, Flying Saucer, and PdfRendererBuilder libraries. The result of this work is an efficient application that represents a significant contribution to digital content creation tools, simplifying the creation, editing, and sharing of presentations and documents.

Key words: PowerPoint presentation, Markdown format, Document conversion, text editing, PDF export