

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Projektová dokumentácia
Časť č. 1 - Inžinierske dielo



Document Wizard

Predmet: Tímový projekt
Ročník: 2021/22

Tím 18

Bc. Andrej Belák
Bc. Martin Jankuliak
Bc. Dávid Korman
Bc. Martin Kukučka
Bc. Blažej Rypák
Bc. Marek Víťaz

Téma: Document Wizard

Vedúci tímu: Ing. Peter Bakonyi

Kontakt: timovy.projekt18@gmail.com

Webová stránka: <http://team18-21.studenti.fiit.stuba.sk/>

Obsah

1. Letný semester	1
1.1. Big picture	1
1.1.1. Úvod	1
1.1.2. Pohľad na systém po zimnom semestri	1
1.1.3. Globálne ciele projektu na letný semester	3
1.1.4. Celkový pohľad na systém	5
1.2. Moduly systému	10
1.2.1. Analýza	10
1.2.2. Návrh	11
1.2.3. Implementácia	12
1.2.4. Testovanie	13
2. Zimný semester	16
2.1. Big picture	16
2.1.1. Úvod	16
2.1.2. Globálne ciele projektu na zimný semester	16
2.1.3. Celkový pohľad na systém	17
2.2. Moduly systému	18
2.2.1. Analýza	18
2.2.2. Návrh	19
2.2.3. Implementácia	19
2.2.4. Testovanie	20
3. Príručky	21
3.1. Používateľská príručka	21
3.1.1. Úvod	21
3.1.2. Použitie funkcionalít	22
3.1.2.1. Kreslenie	22
3.1.2.2. Watermark	23
3.1.2.3. 2D objekty	24
3.1.2.5. Rotácia jednej strany	25
3.1.2.6. Rotácia všetkých strán	26
3.1.2.7. Split	27
3.1.2.8. Merge	27
3.1.2.9. Export citácií	28
3.1.2.10. Úprava metadát	29

3.2.	Inštaláčn prručka	30
3.2.1.	vod	30
3.2.2.	Postup inštalácie	30
4.	Technick dokumentcia	30
5.	Zoznam prloh.....	31
A.	Vygenerovan dokumentcia	31
B.	Protokol o pouzvatel'skom testovan	31
C.	Pouzvatel'sk prručka.....	31
D.	Inštaláčn prručka	31

1. Letný semester

1.1. Big picture

1.1.1. Úvod

Tento dokument slúži na popis inžinierskeho diela vzniknutého po tímovom úsilí vykonanom v letnom semestri v rámci predmetu Tímový projekt.

Produkt má názov Document Wizard a bol vytvorený tímom č.18.

V nasledujúcich sekciách dokument ponúka stručný prehľad globálnych cieľov projektu pre prácu v letnom semestri a vyjadrenie k ich postupnom dosahovaní. Dôležitou sekciou je celkový pohľad na systém, v ktorom sa dokument zaoberá uceleným pohľadom na vytvorený produkt. Ponúka vysvetlenie použitej architektúry vo forme návrhov a diagramov. V ďalšej kapitole dokumentácie nasleduje popis procesu vývoja týkajúci sa produktu, ktorý je rozdelený na základe jednotlivých fáz vývoja na analýzu, návrh, implementáciu a následné testovanie vzniknutého produktu. V posledných častiach dokumentu nájdeme príručky na inštaláciu a používanie produktu a aj automaticky vygenerovanú technickú dokumentáciu.

1.1.2. Pohľad na systém po zimnom semestri

Výsledná podoba prototypu zo zimného semestra zahŕňala funkčný zobrazovač PDF s podmnožinou dostupných a implementovaných plánovaných funkcionalít. Tabuľka nižšie znázorňuje dostupné funkcionality po zimnom semestri, ich textový popis a príslušný stav.

Názov funkcionality	Popis funkcionality	Stav funkcionality
View	otvorenie a zobrazenie dokumentu	kompletná
Rotate	rotácia dokumentu	kompletná – menšie chyby
Delete	zmazanie strán dokumentu	rozpracovaná – väčšie chyby
Watermark	pridanie vodoznaku	rozpracovaná – chýbajúce vlastnosti
Drawing	kreslenie a zvýrazňovanie textu do dokumentu	hrubý prototyp – väčšina vlastností chýba
Split	rozdelenie dokumentu	kompletná
Merge	zlúčenie dokumentov	kompletná

Tabuľka č.1 – Zoznam a stav dostupných funkcionalít po zimnom semestri

Všetky funkcionality nenachádzajúce sa v kompletne dokončenom stave bude cieľom pretaviť do tohto stavu v priebehu letného semestra. Okrem už spomenutých funkcionalít evidujeme ďalšie funkcionality, ktoré plánujeme implementovať v priebehu letného semestra. Plánované funkcionality zobrazuje tabuľka nižšie.

Názov funkcionality	Popis funkcionality	Odhadovaná náročnosť
Citation	generovanie automatických citácií v rôznych formátoch	stredná
Metadata edit	úprava metadát zobrazovaného dokumentu	stredná
2D Objekty	umiestňovanie dvojrozmerných objektov do dokumentu v podobe nálepiek a textových polí	vysoká
Watermark	pridanie vodoznaku – dopracovať chýbajúce vlastnosti ako napríklad upload vlastného vodoznaku	vysoká
Drawing	kreslenie a zvýrazňovanie textu do dokumentu – dopracovanie zmeny veľkosti štetca a farby	veľmi vysoká

Tabuľka č.2 – Plánované funkcionality na implementáciu v letnom semestri

1.1.3. Globálne ciele projektu na letný semester

- **Úspešné nasadenie produktu na knižničný systém Elvíra** – najvyššia priorita – Nasadenie výsledného produktu do knižničného systému Elvíra je tým najhlavnejším dôvodom, prečo bol tento projekt vôbec vypísaný. Na tento cieľ sa musí myslieť v rámci celej práce v letnom semestri, no najvyššiu prioritu naberie ku koncu letného semestra, kedy plánované úlohy do šprintov sa budú týkať najmä tohto cieľa. Projekt musí byť zaobalený v takej forme, aby mohol byť následne na systéme Elvíra spustiteľný ako samostatný modul. Tento cieľ sa s menšími komplikáciami podarilo zrealizovať včas a už v súčasnej dobe je Document Wizard v knižničnom systéme Elvíra použiteľný.
- **Vyriešenie korektného zaznamenávania akcií v editore** – vysoká priorita – Korektné zaznamenávanie akcií vykonaných pomocou dostupných funkcionalít v zobrazovači PDF súborov sa stalo veľkým problémom pre náš projekt, ktorý vznikol po práci v zimnom semestri. Cieľom bolo tento problém správne vyriešiť čo najskôr, pretože ovplyvňoval vývoj ostatných funkcionalít a spôsoboval viacero chýb a nežiadúceho správania. Tento problém sa podarilo úspešne vyriešiť v strede letného semestra, pomocou správnej väzby a komunikácie medzi súbormi nachádzajúcimi sa v moduloch store a utils.
- **Implementácia funkcionality Drawing** – vysoká priorita – Funkcionalita Drawing zahŕňa voľné kreslenie po zobrazenom dokumente, ktoré môže byť následne uložené kliknutím na príslušné tlačidlo. Pri kreslení môžeme meniť veľkosť štetca a taktiež jeho farbu. Táto funkcionalita by mala pomôcť študentom pri poznámkovaní a vyznačovaní si podstatných častí v texte alebo grafoch a obrázkoch nachádzajúcich sa v dokumente. Vypracovanie tejto funkcionality sa ukázalo byť problematické. Implementácia bola odovzdaná podľa plánu ešte v strede letného semestra, no bolo nájdených veľa chýb a nežiadúceho správania naviazaného na túto funkcionalitu, ktoré boli náročné na odstránenie. Do konca semestra sa odstránenie chýb však úspešne podarilo.
- **Implementácia funkcionality Stickers** – stredná priorita – Pod touto funkcionalitou myslíme umiestňovanie predvolených dostupných dvojrozmerných objektov priamo do otvoreného dokumentu. Umiestňované objekty dokážeme presúvať, rotovať, zväčšovať a zmenšovať. Taktiež zahŕňa možnosť pridania a následného písania do textového poľa alebo zvolenie a následné nahratie vlastného dvojrozmerného objektu ako obrázku. Túto funkcionalitu sa podarilo implementovať už v rannom priebehu letného semestra. Funkcionalita bola následne refaktorovaná a úspešne hotová pred skončením prvej polovice letného semestra.
- **Implementácia funkcionality Cite** – stredná priorita – Funkcionalita citácií zahŕňa automatické generovanie citácií z prezeraného dokumentu vo formátoch plain text alebo BibTeX. Funkcionalitou sme sa zaoberali krátko po skončení prvej polovice letného semestra, pričom bola v priebehu jedného šprintu úspešne vypracovaná. Chyby ani nežiadúce správanie v rámci tejto konkrétnej funkcionality zaznamenané neboli.
- **Implementácia funkcionality Metadata** – stredná priorita – Metadata edit zahŕňa úpravu dostupných metadát otvoreného PDF dokumentu. Upravené metadáta

dokážeme následne uložiť a tieto zmeny majú následne okamžitý účinok. Dokument sa v zobrazovači nachádza v pozmenenom stave a dokážeme s ním pracovať rovnakým spôsobom, ako predtým. Podobne ako pri citáciách, aj funkcionality zmeny metadát sa podarilo úspešne implementovať bez väčších problém na začiatku druhej polovice semestra.

- **Vypracovanie vedeckého článku na konferenciu IIT.SRC** – vysoká priorita – Vypracovanie vedeckého článku je aj pre vedúceho projektu jeden z hlavných cieľov tohto projektu. Očakávaný deadline pre túto prácu bol stanovený na koniec prvej polovice letného semestra. Vedecký článok bol úspešne vypracovaný a včas odovzdaný. Témou článku bol inovatívny prístup k získavaniu a následnej analýze požiadaviek, ktorý bol našim tímom použitý v prvej polovice zimného semestra.
- **Odprezentovanie projektu na IIT.SRC** – stredná priorita – Kvalitná prezentácia práce na vedeckej konferencii je kľúčová pre dobré umiestnenie tímu v rámci súťaže TP Cup. Je potrebné pripraviť propagačný plagát, ktorý bude stručne a precízne informovať záujemcov o vznikajúcom produkte, a tiež aj prezentáciu ako podklad k ústnemu výkladu podanému potenciálnym záujemcom. Všetky artefakty boli včas vypracované a naplno využité priamo pri prezentácii na vedeckej konferencii.
- **Úspešné umiestnenie v TP Cupe** – stredná priorita – Ciele a ambície vedúceho projektu sú vysoké a snaží sa náš tím priebežne smerovať k vysokému umiestneniu v rámci súťaže TP Cup. Pre dobré umiestnenie je potrebné kvalitné vypracovanie vedeckého článku, odprezentovania práce na vedeckej konferencii a vytvorenie propagačného článku na blog a videa. Všetky tieto body boli v rámci práce v letnom semestri úspešne vypracované a na čas odovzdané. Výsledky umiestnení v súťaži v čase písania dokumentácie ešte neboli zverejnené.
- **Dokončenie implementácie funkcionality Delete** – nízka priorita – Implementácia funkcie zmazania strán po zimnom semestri síce existuje, no nespĺňa viaceré dodatočné požiadavky a spôsobuje hneď niekoľko chýb a rovnako aj nežiaduce správanie v programe. Úspešné riešenie tohto problému sa neočakávanie pretiahlo až ku koncu práce v letnom semestri, ale nakoniec sa problém podarilo do finálnej verzie produktu vyriešiť a funkcionality zmazania strán je tak plne funkčná.
- **Prepracovanie funkcionality Watermark** – nízka priorita – Prvotná verzia funkcionality vodoznaku bola prevzatá už z výsledkov zo zimného semestra. Chýba však možnosť pre voľbu úpravy prehľadnosti vodoznaku a tiež aj voľba pre nahratie vlastného vodoznaku z lokálneho úložiska. Túto funkcionality sa podarilo prepracovať už po druhom šprinte letného semestra.
- **Konzultácie s administrátormi knižničného systému Elvíra** - stredná priorita - po vytvorení určitej časti finálnej podoby výsledného produktu po zimnom semestri dokážeme túto časť odprezentovať priamo našim pomyselným zákazníkom, teda administrátorom knižničného systému Elvíra. Stretnutia boli priebežne plánované počas letného semestra a prebiehali formou krátkej prezentácie súčasného stavu vyvíjaného produktu, pričom

1.1.4. Celkový pohľad na systém

1.1.4.1. Čo je to Document Wizard?

Document Wizard je samostatný webový modul pre zobrazovanie a prácu so súbormi vo formáte PDF. Okrem zobrazovania dokumentov prináša aj množstvo funkcií pre bežnú, ale aj rozšírenú prácu a potrebu modifikácie prezeraného dokumentu, napríklad funkciu vymazávania strán, rozdeľovania a zlučovania PDF súborov, kreslenia a zvýrazňovania obsahu, umiestňovania dvojrozmerných objektov a pridávania poznámok. Vykonané zmeny v dokumente je možné uložiť a následne modifikovaný dokument stiahnuť ako lokálnu kópiu.

Projekt vznikol s cieľom nasadenia výsledného produktu na existujúci školský knižničný systém s názvom Elvíra. Knižničný systém bude obsahovať rôzne druhy dokumentov vo formáte PDF, od záverečných prác študentov, až po dostupné skriptá. Vytvorený modul Document Wizard bude v tomto knižničnom systéme využívaný jeho používateľmi na zobrazovanie a prácu s dostupnými dokumentami.

Produkt tak bude využívaný najmä študentami pri procese výskumu a analýzy odbornej literatúry, pričom by im práca v Document Wizarde mala tento proces zjednodušiť a sprehladniť. K naplneniu tejto vízie napomáha väčšina vyvinutých funkcionalít. Veľmi užitočným nástrojom pre študentov môžu byť automatické citácie z otvoreného dokumentu, ktoré sú automaticky generované z metadát dokumentu v rôznych formátoch podľa vlastného výberu. Vytvorenú citáciu je možné nakopírovať do systémovej schránky alebo stiahnuť v samostatnom textovom súbore. Ďalšou užitočnou funkcionalitou pre študentov môže byť najmä pridávanie poznámok priamo do dokumentu, či zvýrazňovanie dôležitých častí v texte alebo obrázkov nachádzajúcich sa v dokumente pomocou módu kreslenia. Príkladom praktického využitia môže byť tiež rozdelenie obsiahleho dokumentu výlučne iba na strany, ktoré zahŕňajú kapitoly dôležité pre študenta.

1.1.4.2. Aké technológie sa používajú?

- Typescript – primárny jazyk pre implementáciu plánovaných funkcionalít
- Javascript – podporný jazyk, alternatíva pri určitých špecifických prípadoch pre Typescript, používa sa napríklad ak si použitá knižnica vyžaduje čistý Javascript
- HTML – štruktúra a obsah frontendu zobrazovača
- CSS – vzhľad a štýl frontendu zobrazovača
- Mozilla PDF.js – knižnica, ktorá zabezpečuje zobrazovanie dokumentov, interagujú s ňou naprogramované funkcionality
- pdf-lib – knižnica zabezpečujúca rendering do formátu PDF, využíva sa pri viacerých funkcionalitách
- fabric.js – knižnica používaná pri funkcionalitách kreslenia a umiestňovania 2D objektov

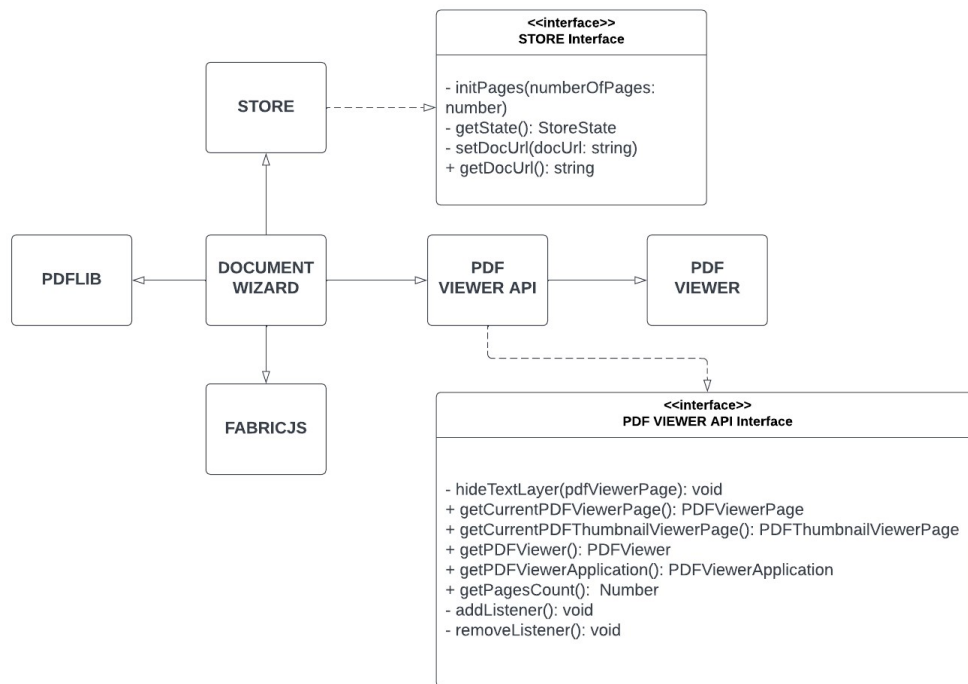
1.1.4.3. Interakcia technológií

Vyvíjaný modul ako celok je vytváraný pomocou jazyka TypeScript, ktorý funguje ako nadstavba k jazyku JavaScript. Projekt je kompilovaný ako webový modul, pričom pri jeho vytváraní využívame aj zdrojové súbory v jazykoch HTML a CSS.

V projekte používame knižnice tretej strany PDF.js, pdf-lib.js a fabric.js. PDF.js je knižnica, ktorá vykresľuje súbory vo formáte PDF a tvorí základ nášho viewera. Knižnica pdf-lib.js slúži na vykonávanie zmien, ich ukladanie a exportovanie. Knižnica fabric.js je knižnica priamo venovaná práci s kreslením a vkladaním rôznych 2D objektov a obrázkov do dokumentu.

Výstup z procesu kompilácie sa zobrazuje na lokálnom hostiteľovi, ktorého port je v rámci tohto projektu nastavený na predvolenú hodnotu 8080.

1.1.4.4. Architektúra riešenia



Obrázok č.1 – Diagram architektúry produktu

- **STORE** - slúži na dočasné ukladanie zmien pri práci
- **PDFLIB** - tento modul slúži na modifikáciu a export zmien pri práci
- **FABRICJS** - slúži na kreslenie a prácu s obrázkami
- **PDF VIEWER** - slúži na zobrazenie PDF dokumentu

Hlavný modul nášho riešenia je DOCUMENT WIZARD, ktorý komunikuje pomocou nami vytvoreného API rozhrania s PDF Viewer, ktorý zobrazuje PDF dokument. Tento modul taktiež pracuje s FabricJS pomocou, ktorého dokážeme kresliť a pridávať obrázky, ktoré ďalej pomocou modulu PDFLIB exportujeme a následne prekreslíme aktuálny canvas.

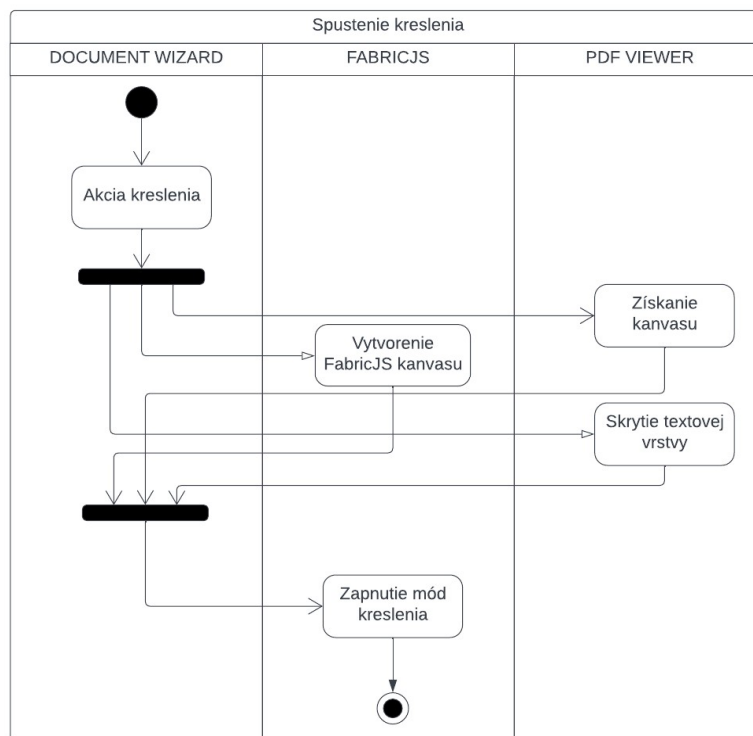
1.1.4.5. Základná štruktúra súborov

- **example**
- **src**
 - **document-wizard**
 - **icons** - ikony použité v paneli nástrojov
 - **store** - funkcionálna časť modulu STORE
 - **utils** - adresár so súborami pre každú funkcionálnu časť
 - **actions.ts** - api rozhranie na spustenie nástrojov
 - **index.ts** - inicializácia panelu nástrojov
 - **styles.css** - štýly document-wizardu
 - **viewer.ts** - pridaná vlastná funkcionálna časť do tried PDF.js
 - **pdfjs** - adresár knižnice mozilla PDF.js
 - **content.ts** - panel nástrojov našej knižnice
 - **index.ts** - inicializácia Document Wizardu
- **tsconfig.ts** - konfigurácia TypeScript projektu
- **webpack.config.ts** - konfigurácia na generovanie knižnice Document Wizard

1.1.4.6. Vysvetlenie na praktickom príklade

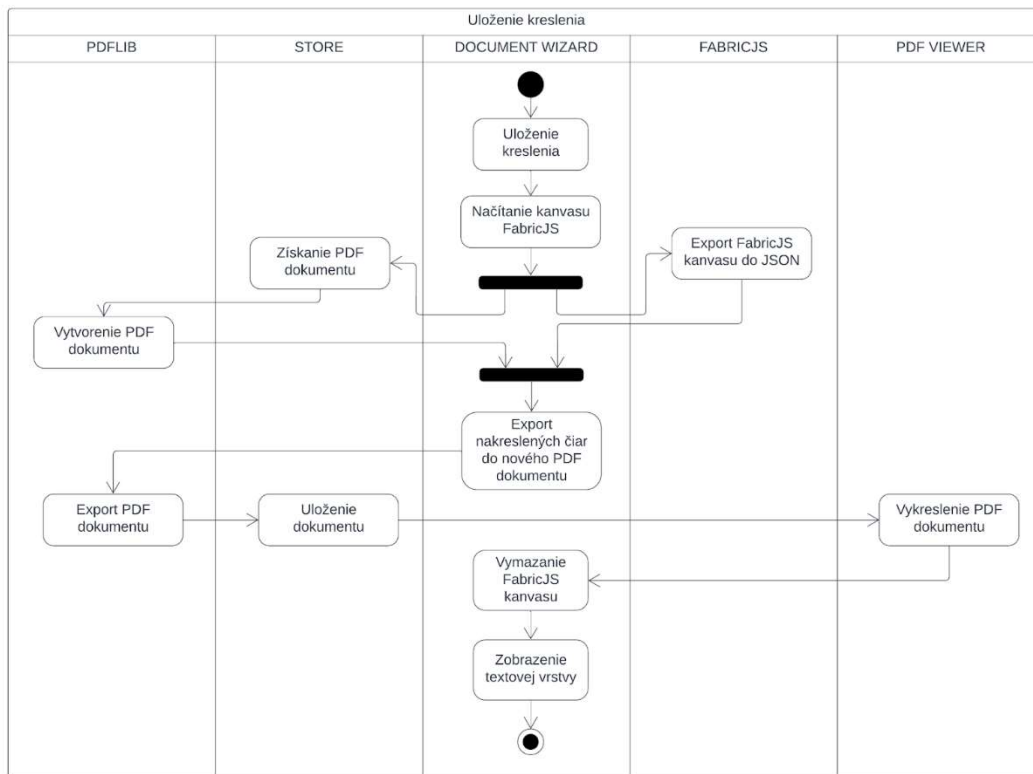
Scenár použitia funkcionality kreslenia

Prvý activity diagram zobrazuje spustenie kreslenia, resp. je na diagrame zobrazená komunikácia medzi modulmi, ktoré sa podieľajú na funkcionálnosti spustenia kreslenia.



Obrázok č.2 – Diagram aktivít popisujúci spustenie kreslenia

Druhý activity diagram reprezentuje funkcionality uloženia kreslenia, reps. export pomocou knižnice PDFLIB a opätovné vykreslenie modifikovaného PDF dokumentu.



Obrázok č.3 – Diagram aktivít popisujúci uloženie kreslenia

1.1.4.7. Aké funkcionality Document Wizard ponúka?



Obrázok č.4 – Propagačný plagát dostupných funkcionalít produktu

Rotácia – modul ponúka štyri tlačidlá pre funkcionality rotácie dokumentu. Prvé dve tlačidlá ponúkajú rotáciu jednej, práve aktívnej strany proti alebo po smere hodinových ručičiek v rotácii deväťdesiatich stupňov. Ďalšie dve tlačidlá ponúkajú obdobnú funkcionality, avšak nie iba pre jednu, práve aktívnu stranu, ale pre všetky strany nachádzajúce sa v prezeranom dokumente.

Zmazanie – funkcionality zmazania po kliknutí na zodpovedajúce tlačidlo umožňuje používateľovi vymazať stranu, ktorú si v zobrazovači momentálne prezerá. Jedná sa o práve aktívnu stranu, ktorou je strana nachádzajúca sa v strede používateľovho zorného poľa.

Rozdelenie – poskytuje jednoduché kontextové menu v podobe troch polí, v ktorom používateľ zdefinuje konkrétny súbor na rozdelenie a následne tiež dve hodnoty, ktoré označujú prvú a poslednú stranu nového, rozdeleného dokumentu. Document Wizard zvolený dokument podľa zadaného rozsahu strán rozdelí a následne je možné si takto rozdelený dokument stiahnuť do lokálneho úložiska.

Zlúčenie – je možné vybrať dve oddelené PDF súbory, ktoré budú následne v Document Wizarde zlúčené do jedného uceleného dokumentu. Takto zlúčený dokument je následne možné zo zobrazovača stiahnuť do lokálneho úložiska.

Kreslenie – ponúka kreslenie po stranách otvoreného dokumentu, pričom je možné ľubovoľne meniť veľkosť používaného štetca a taktiež jeho farbu. Následne, po dokončení kreslenia, je možné zmeny uložiť vo forme vyrenderovania nakreslených ciest priamo na stranu dokumentu, alebo je tiež možné kreslenie zrušiť a obnoviť tak pôvodnú podobu strany.

2D Objekty – pridávanie dvojrozmerných objektov do otvoreného dokumentu vo forme predom dostupných nálepiek, prípadne je možné nahráť akékoľvek obrázky vo formáte .jpg alebo .png, ktoré je možné umiestniť do dokumentu pomocou tlačidla custom. Je taktiež možné pridávať textové polia do otvoreného dokumentu, v ktorých je možné napísať vlastný

text so zmeniteľnou farbou. Tieto dvojrozmerné objekty je možné ľubovoľne presúvať po stranách dokumentu, rotovať s nimi a meniť ich veľkosť.

Citácie – je možné automaticky vygenerovať citácie z prezeraného dokumentu vo formátoch plain text alebo BibTex. Tieto citácie sa automaticky generujú z dostupných metadát dokumentu. Vygenerované citácie je možné nakopírovať priamo do lokálnej schránky počítača, prípadne stiahnuť ako samostatný textový súbor.

Vodoznak – pridanie vodoznaku do prezeraného dokumentu. Je možné si definovať predvolený vodoznak alebo nahrať z lokálneho úložiska vlastný obrázok vo formáte .jpg alebo .png, ktorý bude zobrazovaný v pozadí každej strany so zmeniteľnou hodnotou priehľadnosti.

Uloženie – po zvolení príslušného tlačidla dokáže webový modul korektne uložiť všetky vykonané zmeny v otvorenom dokumente. Tieto zmeny sú priebežne zaznamenávané a po zvolení tejto akcie sú odoslané príslušným knižniciam na rendering, ktoré vyprodukurujú zmodifikovaný dokument. Stav zobrazovaného dokumentu sa ihneď obnoví a pozmenený dokument je dostupný v zobrazovači. Dokument ja následne možné si stiahnuť ako lokálnu kópiu.

1.2. Moduly systému

1.2.1. Analýza

Časť analýzy pre náš projekt bola podstatná najmä v zimnom semestri, v letnom semestri táto fáza projektu nemala skoro vôbec žiadnu dôležitosť. V letnom semestri bola preberaná analýza iba pri príležitosti prezentácie produktu kľúčovým stakeholderom, ktorými boli manažéri knižničného systému Elvíra. Z ich následnej spätnej väzby vyplynulo niekoľko pripomienok a návrhov na nové funkcionality.

Táto spätná väzba bola zaznamenaná a zapísaná, následkom čoho došlo k analytickej práci vo forme malej štúdie uskutočniteľnosti pre každú z chýbajúcich, navrhovaných funkcionalít. Výsledky štúdie uskutočniteľnosti boli nasledovné:

Názov funkcionality	Rozhodnutie o uskutočniteľnosti
Automatické citácie	prijatá
OCR	zamietnutá
Zmena metadát otvoreného dokumentu	prijatá
Priama editácia obsahu dokumentov	zamietnutá

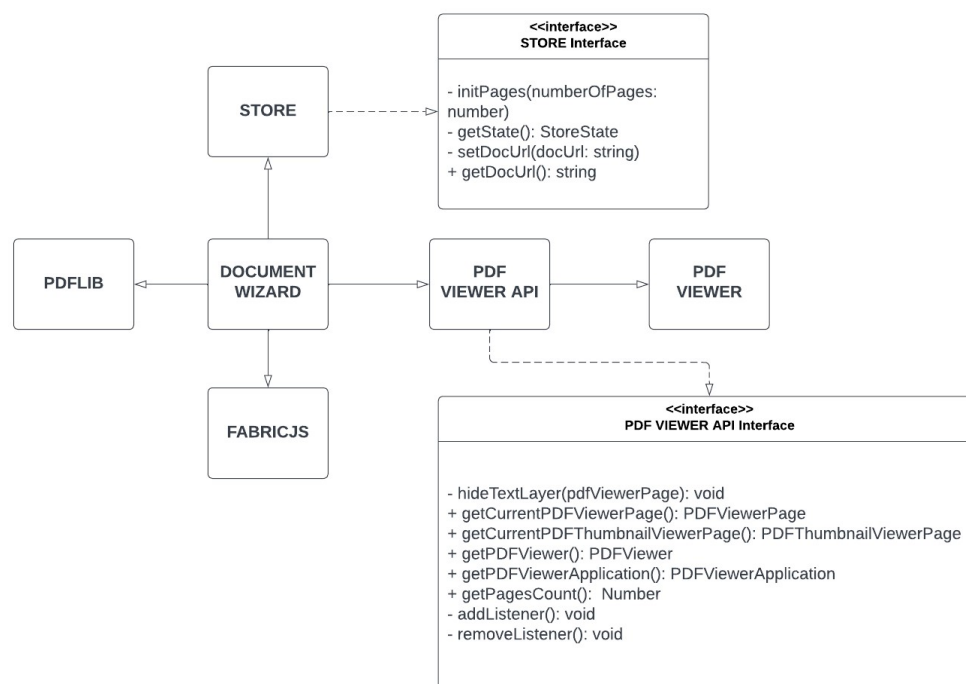
Tabuľka č.3 – Prehľad rozhodnutia uskutočniteľnosti navrhovaných funkcionalít

Prijaté funkcionality boli následne zaradené do produktového backlogu. V nasledujúcich šprintov boli tieto úlohy z backlogu prijaté a naplánované na ich realizáciu v konkrétnych šprintoch. Funkcionality zamietnuté vykonanou štúdiou uskutočniteľnosti do produktového backlogu zaradené neboli.

Štúdia uskutočniteľnosti spočívala najmä vo výskume existujúcich riešení ponúkajúcich konkrétne navrhované funkcionality, a najmä výskum nástrojov a knižníc, pomocou ktorých by boli navrhované funkcionality neskôr realizovateľné. Po nájdení vhodných nástrojov a knižníc následne prebehol test importu týchto softvérových riešení do projektu. Po úspešnom prebehnutí celého procesu bolo rozhodnuté o uskutočniteľnosti konkrétnej funkcionality.

1.2.2. Návrh

Celkový návrh architektúry bol prebraný zo stavu našej práce zo zimného semestra. K tejto architektúre bolo pridaných niekoľko nových elementov. Návrh architektúry sa menil vzhľadom na pridané moduly knižníc, ktorou bola napríklad knižnica fabric.js, ďalej bolo pridané API rozhranie PDF viewera, a tiež prišlo aj k prepracovaniu modulu store, ktorý slúži na ukladanie zmenených akcií, súčasných stavov a všeobecnú podporu komunikácie medzi funkcionalitami.

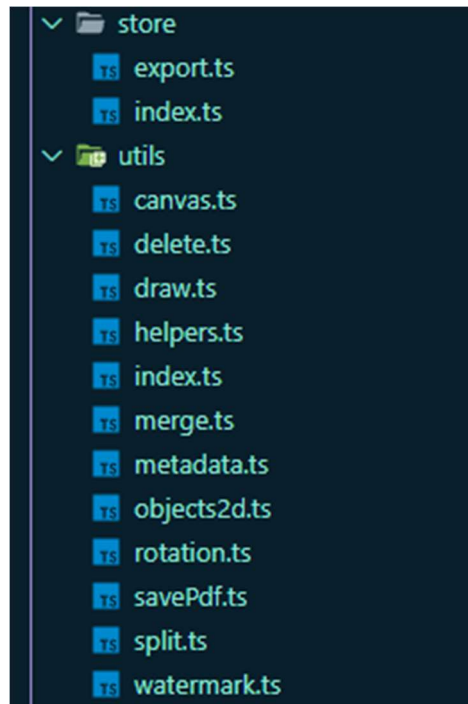


Obrázok č.1 – Diagram architektúry produktu

Samotné implementácie jednotlivých funkcionalít sú uložené v module, ktorý je na diagrame vyššie zobrazený pod názvom Document Wizard. Funkcionality podľa potreby dokážu komunikovať s podpornými knižnicami pri dosiahnutí žiadanych výsledkov, prípadne môžu požiadať o doplňujúce informácie samotný zobrazovač, pomocou PDF Viewer API. Modul store slúži ako podporný modul pre funkcionality, napríklad pri cacheovaní výsledkov, vďaka vzájomnej komunikácii medzi funkcionalitami a podobne.

1.2.3. Implementácia

Pri implementácii postupujeme podľa návrhu uvedeného vyššie. Väčšina práce v letnom semestri sa týkala implementovania naplánovaných funkcionalít nachádzajúcich sa v produktovom backlogu. Pri implementácii dodržiavame pravidlá oddelenia modulov store, prídavných knižníc, súčastí zobrazovača a modulu funkcionalít, označenom v návrhu ako Document Wizard. Pri nahrávaní implementácií postupujeme podľa konvencie verziovania kódu, ktorá je popísaná v sekcii metodík v dokumentácii pre riadenie projektu.



Obrázok č.5 – Znáozornenie implementačnej štruktúry funkcionalít

Obrázok vyššie znázorňuje popisovanú skutočnosť implementovania funkcionalít. Logika funkcionalít je prehľadne pomenovaná a uložená v príslušnom adresári, zatiaľ čo ostatná logika, najmä logika modulu store, je vzájomne oddelená.

```
import { PDFDocument, degrees, rgb, StandardFonts } from 'pdf-lib'
import { rerenderViewer } from '../pdfjs/web/viewer'
import { store } from '../store'

export async function savePdf() {
  const url = store.getState().pdfUrl
  const existingPdfBytes = await fetch(url).then((res) => res.array
```

Obrázok č.6 – Spôsob implementácie komunikácie medzi funkcionalitami

Vzájomná komunikácia medzi oddelenými modulmi sa realizuje pomocou importov a exportov potrebných funkcií. Podobne, aj pri implementovaní konkrétnej funkcionality, člen tímu odkrýva funkcie určené na komunikáciu pomocou kľúčového slova export.

```
const colorListener = (event: any) => {
  if (event.data['event_id'] === 'sticker_color') {
    hex_color = event.data['data']
    let result = hexToRgb(event.data['data'])
    if (result !== null) {
      rgb_color[0] = result['r']
      rgb_color[1] = result['g']
      rgb_color[2] = result['b']
      rgb_string_color =
        'rgb(' + rgb_color[0] + ',' + rgb_color[1] +
        ',' + rgb_color[2] + ')'
    }
  }
}
```

Obrázok č.7 – Náhľad do vnútornej logiky implementácie funkcionalít

Samotné funkcie sa implementujú pomocou logického združenia konštánt, rozhraní, funkcií (synchronných aj asynchronných) a event listenerov. Naprogramovanou logikou týchto elementov a ich následnou spolupracou dosahujeme žiadaných výsledkov vo forme kompletne funkčných funkcionalít, dostupných v našom produkte zobrazovača.

1.2.4. Testovanie

Základné testovanie vyvíjaného kódu je veľmi podobné, ako bolo popísané v zimnom semestri. V letnom semestri sme si však prehľadnejšie zadefinovali rozlíšenie viacerých kategórií testovania.

- **Jednotkové testovanie** pri vývoji (in-dev unit testing) – vykonáva samotný programátor, teda člen tímu, ktorý pracuje na svojej implementačnej úlohe. Vykonáva sa najmä pomocou dostupných debugovacích nástrojov, ktoré sú v prípade webového vývoja dostupné vo forme integrovaného debuggera v prehliadači Google Chrome, ktorý sa zobrazí po stlačení klávesy F12.

Člen tímu pracujúci na implementácii vykonáva takéto jednotkové testovanie pravidelne a priebežne ho strieda s aktivitou písania kódu. Zaznamenáva si akékoľvek nájdené chyby a neočakávané správanie, ktoré v najlepšom prípade vyrieši ešte pred odovzdaním kódu a dokončením konkrétnej úlohy.

V prípade, že sa určitým chybám nepodari počas vývoja predísť, člen tímu o tejto skutočnosti informuje v podobe komentára pod vykonaný merge requeste na GitLabe, a taktiež o tejto skutočnosti v dostatočnej miere informuje ostatných členov tímu pri prezentovaní jeho dokončenej úlohy na najbližšom tímovom stretnutí.

- **Integračné testovanie** (integration testing) – je aktivita vykonávaná väčšinou rolou hlavného testera, ktorý priebežne vykonáva kľúčové scenáre na vytvorenom produkte, pričom zaznamenáva akékoľvek náznaky chýb alebo nežiadúceho správania. Takáto aktivita sa vykonáva v projekte priebežne, aspoň jedenkrát v priebehu každého šprintu.

Po identifikovaní chýb je kontaktovaná osoba, ktorej úloha, ktorej sa nájdená chyba, bola v predchádzajúcom priebehu projektu pridelená. Táto osoba je následne poverená vyriešením identifikovanej chyby. V závažných prípadoch je možné osobe do ďalšieho šprintu naplánovať úlohu, ktorej primárnym cieľom bude riešenie konkrétnej chyby.

- **Systémové testovanie** (system testing) – vykonávané najmä pred dôležitými termínmi alebo milníkmi, čo zahŕňa napríklad:
 - odprezentovanie produktu stakeholderom
 - odovzdanie produktu na používateľské testovanie
 - odovzdanie produktu po letnom semestri
 - odprezentovanie produktu na študentskej konferencii IIT.SRC

Proces systémového testovania je vedený rolou manažéra testovania, ktorý podľa potreby rozdeľuje jednotlivým členom tímu úlohy týkajúce sa tejto aktivity. Cieľom tohto testovania je identifikovanie a následné vyriešenie čo najväčšieho počtu chýb v bežnom používaní vytváraného produktu a zvýšenie dôvery vo vytváraný produkt.

- **Používateľské testovanie** (user testing) – je aktivita, ktorá je podľa harmonogramu letného semestra vykonávaná v poslednom šprinte. Vytvorený produkt bude odovzdaný na používateľské testovanie odlišnému tímu, ktorý nám poskytne spätnú väzbu v podobe výsledného protokolu o testovaní. Rola manažéra testovania je druhému tímu v prípade potreby plne dostupná.

Rovnako tak aj členovia nášho tímu budú vykonávať aktivitu používateľského testovania na produkte, ktorý bude poskytnutý odlišným tímom. Cieľom je vyskúšať si poskytnutú aplikáciu a jej bežné používateľské scenáre, pričom každý z členov dáva zvýšený ohľad na identifikáciu akýchkoľvek chýb a potenciálne vzniknutých nezrovnalostí. Rola manažéra testovania vedie evidenciu nájdených chýb pri používateľskom testovaní a taktiež navrhuje výsledné ohodnotenie tejto aktivity, ktoré je neskôr odsúhlasené v rámci celého tímu.

Výsledný dokument protokolu z testovania nášho produktu je priložený v prílohách.

2. Zimný semester

2.1. Big picture

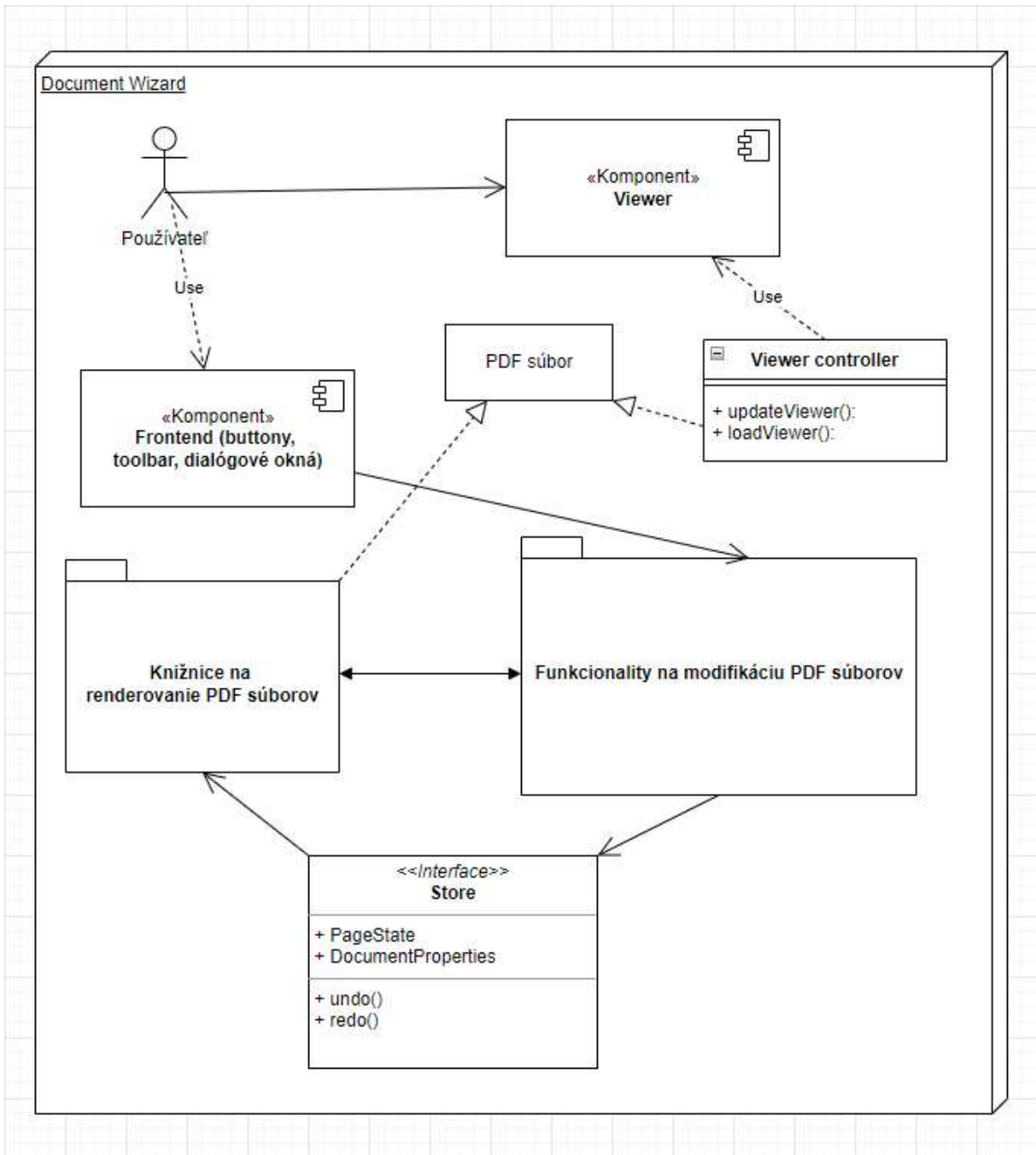
2.1.1. Úvod

Tento dokument slúži na oboznámenie sa a bližší popis s inžinierskym dielo, ktoré vzniká v rámci predmetu Tímový projekt. Poskytuje prehľad globálnych cieľov projektu na zimný semester týkajúcich sa tímového úsilia na tomto predmete, následne aj celkový pohľad na architektúru vyvíjaného systému. V ďalších kapitolách popisujeme naše získané vedomosti a postrehy v najdôležitejších fázach vývoja softvéru, akými sú návrh, dizajn, implementácia a testovanie. Opis diania vo vymenovaných fázach sa týka vyvíjaného produktu. Na konci dokumentu taktiež nájdeme odkaz na vygenerovanú technickú dokumentáciu inžinierskeho diela, ktorá je v rámci odovzdania priložená v samostatnom súbore. V tomto dokumente nenájdeme popis riadenia projektu a používaných metodík. Tieto informácie sú mimo rozsah tohto dokumentu a nájdeme ich v druhej časti dokumentácie.

2.1.2. Globálne ciele projektu na zimný semester

- Úspešné identifikovanie a následná integrácia všetkých potrebných komponentov, knižníc, a iných potrebných súčastí na realizovanie zadaných funkcionálnych požiadaviek
- Návrh, dizajn, následný vývoj a úspešná integrácia modulu na zobrazovanie PDF súborov (modul budeme v kontextu nášho projektu nazývať viewer)
- Vyriešená otázka správnej šablóny a postupu ako vyvíjať a do už vytvoreného modulu na zobrazovanie PDF súborov integrovať ďalšie doplnkové funkcionality definované vo funkcionálnych požiadavkách
- Vyskúšanie si, získanie skúsenosti a celkové osvojenie si metodík vývoja a riadenia projektu SCRUM
- Získať efektivitu v plánovaní a riadení vývoja projektu metodík SCRUM
- Pravidelne a úspešne prinášať progres na projekte v rámci celého tímu
- Úspešne, včas, a efektívne identifikovať a prekonať vzniknuté blokery a problémy pri procese vývoja produktu
- Úspešný vývoj a následná integrácia viacerých pridaných funkcionalít na modifikovanie PDF súborov
- Pravidelné aktualizovanie dokumentácie a zaznamenávanie získaných poznatkov z vykonaného vývoja na úrovni každého člena tímu
- Pravidelné a efektívne predávanie si informácií o vytváranom produkte vo forme prezentácií prínosov jednotlivcov do tímového projektu

2.1.3. Celkový pohľad na systém



Obrázok č.8 - pohľad na architektúru systému v zimnom semestri

Zjednodušený pohľad na architektúru vyvíjaného systému poskytuje obrázok č.1. Zobrazuje kľúčové triedy, objekty, rozhrania, balíky a komponenty systému, a tiež aj načrtáva príslušné vzťahy medzi vyobrazenými entitami.

- Viewer - všetky súčasti systému, ktoré sa podieľajú na zobrazovaní PDF súborov
- Viewer controller - súbor funkcionalít, ktoré viewer ovládajú - pri vykonaných zmenách alebo pri nahraní nového PDF súboru môžu viewer aktualizovať, prípadne nahrádzať zobrazované súbory
- PDF súbor - objekt konkrétneho súboru PDF, s ktorým používateľ chce pracovať – používateľ môže želaný PDF súbor uploadovať z vlastného zariadenia, po dokončení práce je možné ho uložiť a stiahnuť naspäť do zariadenia
- Frontend - zahŕňa všetky viditeľné a funkčné súčasti systému pri bežnom používaní okrem zobrazovania PDF súborov
 - sem patria všetky tlačidlá, dialógové okná, rolovacie okná a panel nástrojov s ktorými používateľ môže interagovať
- Knižnice na renderovanie PDF súborov - zahŕňa všetky funkcionality systému, ktoré sa zaoberajú renderovaním PDF súborov, či už sú to podporné knižnice, alebo iné naprogramované súčasti a skripty
- Funkcionality na modifikáciu PDF súborov - tento balík kompletne zahŕňa naprogramovanú logiku doplnkových funkcionalít ako jeden celok, ktoré vyvolávame pomocou elementov z frontendu
- Store - špeciálne rozhranie používané na zaznamenávanie žiadaných zmien od používateľa, z takto zaznamenaných stavov následne svoj vstup čerpajú knižnice na generovanie PDF súborov
- Používateľ - znázorňuje rolu nášho zákazníka, typického používateľa vyvíjaného systému

2.2. Moduly systému

2.2.1. Analýza

Pri analýze sme sa najmä snažili identifikovať čo najviac už existujúcich alebo podobných riešení, z ktorých konkrétne požadované artefakty a fragmenty môžeme neskôr skombinovať alebo zmodifikovať do nášho riešenia. Analýza mala tiež priniesť prvotnú ideu a náčrt nášho pracovného postupu pri vytváraní multifunkčného modulu na prezeranie a modifikáciu PDF súborov.

Dôležitou časťou analýzy tiež bolo pochopiť formát súboru PDF aj z hľadiska jeho vnútornej štruktúry. Hlavnou ideou nášho vyvíjaného produktu bude zaznamenávať žiadané modifikácie PDF súborov od používateľa a následne renderovať zmodifikované PDF súbory so zaznamenanými zmenami.

Súčasne musí modul zabezpečiť správne zobrazovanie a priebežné aktualizovanie náhľadu konkrétneho PDF súboru. K aktivite analýzy sa periodicky v priebehu projektu vraciame najmä pri metodike backlog groomingu, kedy analyzujeme naplánované úlohy takým spôsobom, aby sme si overili ich už pridelenú prioritu a kroky.

2.2.2. Návrh

Prvotný návrh vytváraného systému pozostával z hlavnej časti, ktorou bol samotný PDF viewer, a následne vedľajších častí, ktorými budú požadované funkcionality, ktoré budú rozširovať celkovú použiteľnosť vytváraného modulu. Požadované funkcionality budú v priebehu tímového projektu postupne vyvíjané nezávisle na sebe rôznymi členmi tímu. Bude sa tak diať v rámci naplánovaných šprintov. Hotové funkcionality budú následne integrované k samotnému vieweru.

Z celkového pohľadu tak viewer musí vedieť pracovať s vyvinutými funkcionalitami a rovnako aj naopak, vyvíjané funkcionality musia byť už v priebehu procesu vývoja vytvárané so zreteľom na kompatibilitu s viewerom.

Viewer PDF súborov so všetkými jeho zdrojovými súbormi sa bude nachádzať v samostatnej zložke. Zložka bude mať dočasný názov src, anglicky source, čo znamená v slovenčine zdroj. Tento priečinok bude obsahovať všetky potrebné súčasti zabezpečujúce správnu funkcionality viewera. Nájdeme v nej všetky kľúčové triedy a ich metódy modulu viewera. Pri vyvíjaní jednotlivých doplnkových funkcionalít programátor nemusí disponovať hĺbkovými vedomosťami týkajúcimi sa obsahu tohto priečinku, preto je pri vývoji možné použiť techniku čiernej skrinky (black box technique).

Všetky prídavné funkcionality k vieweru sa budú nachádzať v samostatnej zložke oddelenej od komponentu viewera. Zložka bude mať zatiaľ iba dočasný názov example. Konkrétny názov a tiež aj usporiadanie funkcionalít sa bude meniť v priebehu projektu po získaní väčšieho množstva skúseností. Do spomínanej zložky budú patriť všetky typescript súbory, ktoré definujú funkcionality prídavných funkcií, a taktiež ich správnu integráciu k vieweru.

2.2.3. Implementácia

Dôležitým komponentom z hľadiska implementácie je najmä priečinok store. Tu sa definujú všetky dôležité rozhrania ohľadom stránok modifikovaného PDF dokumentu.

Každý dokument má svoj stav (state), pričom evidujeme zoznam všetkých jednotlivých stránok dokumentu a celkový počet stránok dokumentu.

Rovnako ako dokument, aj každá stránka dokumentu má svoj vlastný stav, kde uvádzame dôležité vlastnosti konkrétnej stránky, akými sú napríklad identifikačné číslo stránky, jej rotácia, skrytie pri vymazaní stránky a tiež aj pole rôznych objektov (môžu to byť priložené obrázky, vodoznak, alebo kreslenie tvarov na konkrétnej stránke), ktoré je z iníciaľneho stavu prázdne.

Definujeme tak určitú hierarchiu stavov, do ktorých sa dokument ako celok môže pri práci s ním dostať. Jednotlivé objekty, ktoré môžeme na stránku vkladať pri modifikácii dokumentu sú pomenované ako placeholders. Každý placeholder si udržuje o sebe informácie. Môžu to byť napríklad jeho umiestnenie na konkrétnej stránke (koordináty x a y), jeho šírka a výška, či hodnota rotácie v stupňoch.

Ak na stránku vkladáme obrázok, udržujeme si o ňom rovnaké informácie, ako o placeholderi (je medzi nimi vzťah dedenia). Navyše definujeme cestu ku konkrétnemu obrázku a jeho priehľadnosť vzhľadom na text.

Všetky doplnkové funkcionality na modifikáciu PDF súborov vo vieweri sú vypracované ako samostatné funkcie, ktoré získavajú informácie od používateľa, spracovávajú ich, ukladajú vykonané zmeny do stavu stránok a prepájajú svoju funkcionality s viewerom, to znamená, že po spracovaní konkrétneho príkazu od používateľa a vykonaní zmien viewer zmenenú stránku obnoví už s požadovanými zmenami.

2.2.4. Testovanie

Testovanie je priebežne vykonávanou aktivitou, ktorá nasleduje vždy po dokončení vývoju doplnkových funkcionalít k vieweru. Konkrétne sa jedná o extenzívne jednotkové testy (unit tests) za pomoci debugovacích pomôcok integrovaných priamo v prehliadači, väčšinou v prehliadači Google Chrome pomocou stlačenia klávesy F12.

Jednotkové testy vykonávajú samotný programátori pri procese vývoja, teda pri zapracovávaní pridelených úloh. Pri takomto testovaní tester najmä dbá na vyhľadanie potenciálnych chýb, prípadne hľadanie ideí na nové funkcionality alebo zlepšenie testovanej časti modulu. Všetky poznatky z vykonaného jednotkového testovania vyvíjaných častí člen tímu odprezentuje pri odovzdávaní svojej úlohy.

Vyššie úrovne testov okrem jednotkových testov, akými sú napríklad integračné, systémové alebo akceptačné testy budú zahrnuté v neskorších fázach vývoja nášho produktu, v priebehu letného semestra. Na implementácii vyšších úrovni testov sa začína pracovať až po získaní konsenzu v rámci celého tímu týkajúceho sa dokončenia kľúčových komponentov vyvíjaného systému.

3. Príručky

Rozoznávame dva druhy dostupných príručiek:

- Používateľská príručka – ponúka stručný prehľad o produkte bežnému používateľovi
- Inštalčná príručka – ponúka postup pre správnu inštaláciu a integráciu produktu

3.1. Používateľská príručka

3.1.1. Úvod

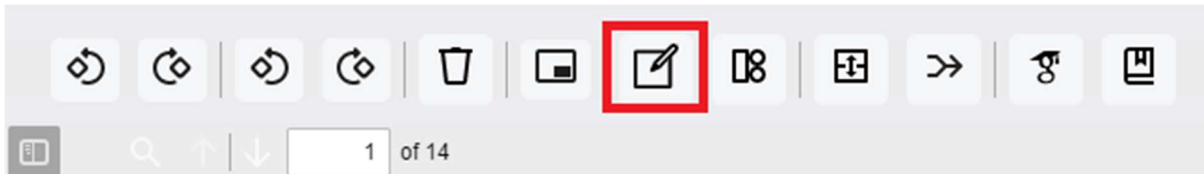
Document Wizard ponúka množstvo funkcií, ktoré sú veľmi intuitívne na použitie. Aj neskúsený používateľ by mal dokázať využiť plný potenciál produktu bez akýchkoľvek výraznejších problémov.

Táto používateľská príručka by budúcim používateľom nášho produktu mala poskytnúť základný prehľad o dostupných funkcionalitách. V primeranej úrovni detailu vysvetľuje a popisuje použitie funkcionalít, ktoré produkt ponúka.

3.1.2. Použitie funkcionalít

3.1.2.1. Kreslenie

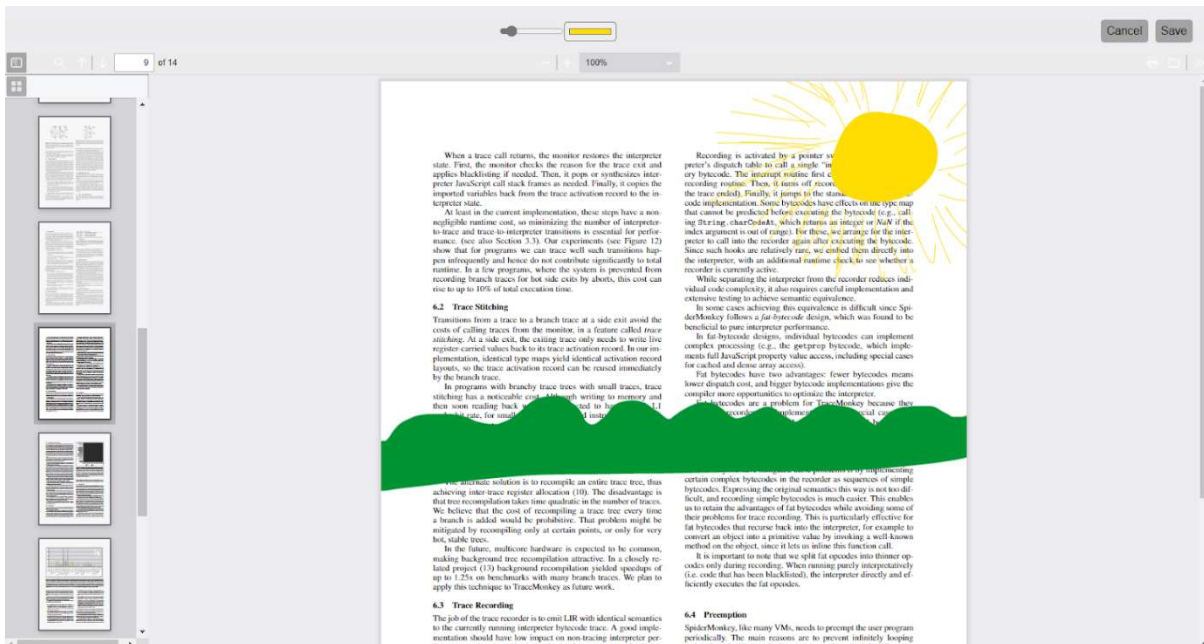
Funkcia kreslenia ponúka upravovať dokument, konkrétne jeho canvas. Tento canvas môže byť upravovaný štýlom kreslenia voľnou rukou.



Kreslenie sa začína kliknutím na vyznačenú ikonu z hlavného menu. Následne sa spustí takzvaný drawing mode. V tomto móde sa hlavné menu zmení na menu, ktoré je exkluzívne pre kreslenie.



V tomto menu sa nachádza slider, ktorým sa mení veľkosť brushu a výber farby. Následne sa tu nachádzajú dva buttony Cancel a Save. Stlačením Cancel sa zahodia všetky kreslené zmeny. Stlačením Save sa nakreslené zmeny uložia do canvasu.

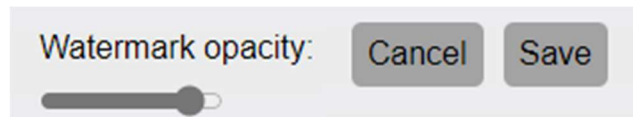


3.1.2.2. Watermark

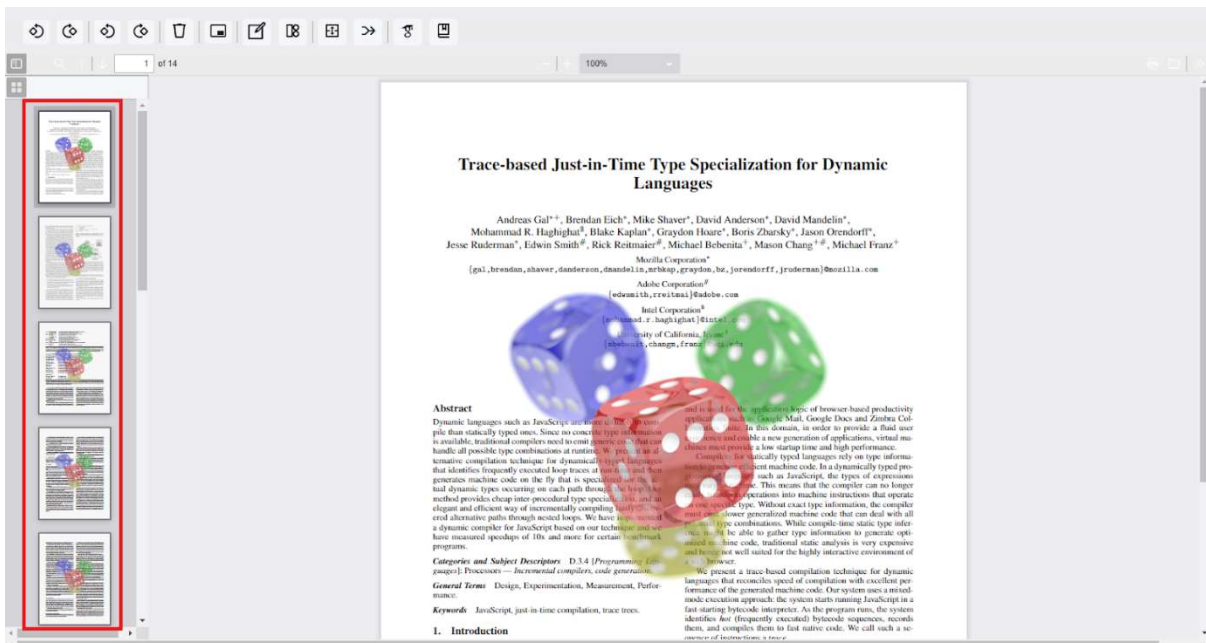
Funkcia watermarku je zameraná na vloženie vodoznaku na všetky strany pdf dokumentu. Tento watermark je následne uložený do canvasu všetkých strán dokumentu.



Watermark sa začína kliknutím na vyznačenú ikonu z hlavného menu. Následne program požiada používateľa o nahranie obrázku z jeho počítača vo formáte png. Taktiež sa hlavné menu zmení na menu, ktoré je exkluzívne len pre watermark.



V tomto menu sa nachádza slider, ktorým sa mení opacity obrázku. Následne sa tu nachádzajú dva buttony Cancel a Save. Stlačením Cancel sa zahodia všetky zmeny. Stlačením Save sa zmeny uložia do canvasu všetkých strán.

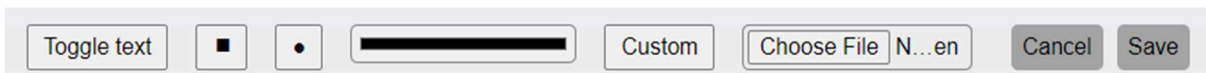


3.1.2.3. 2D objekty

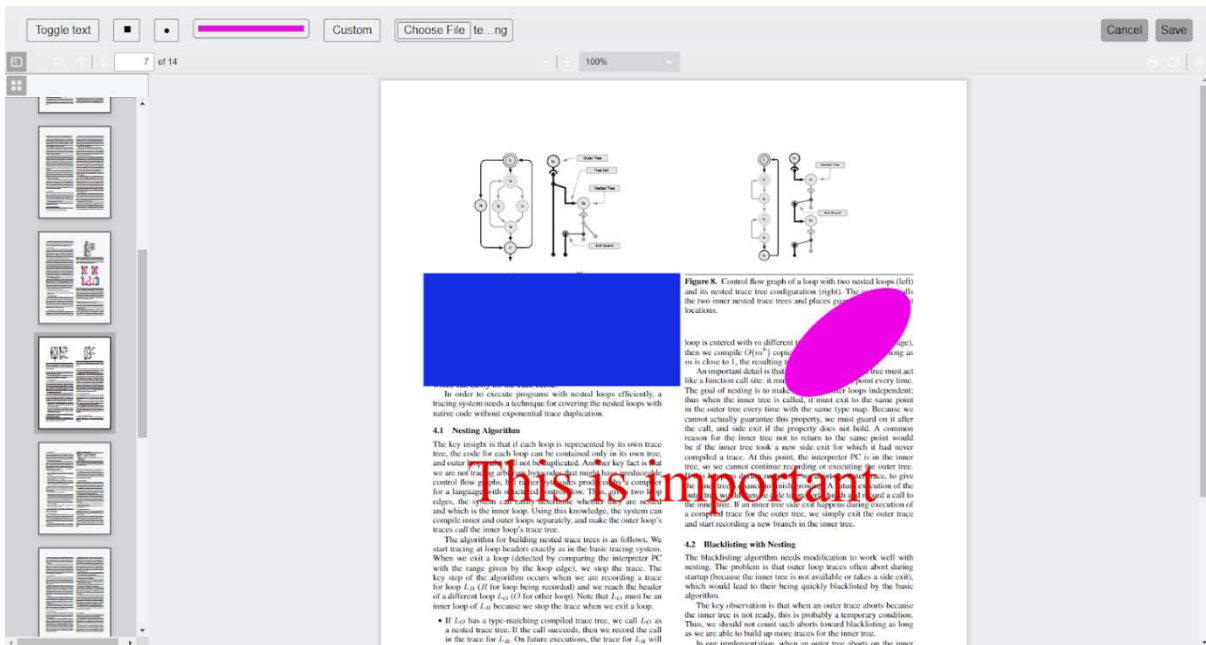
Funkcia 2D je zameraná na vloženie rôznych 2D objektov do dokumentu. Náš produkt ponúka vložiť štvorec, kruh, text a obrázok z počítača.



2D objekty sa začínajú kliknutím na vyznačenú ikonu z hlavného menu. Hlavné menu sa zmení na menu, ktoré je exkluzívne len pre 2D objekty.

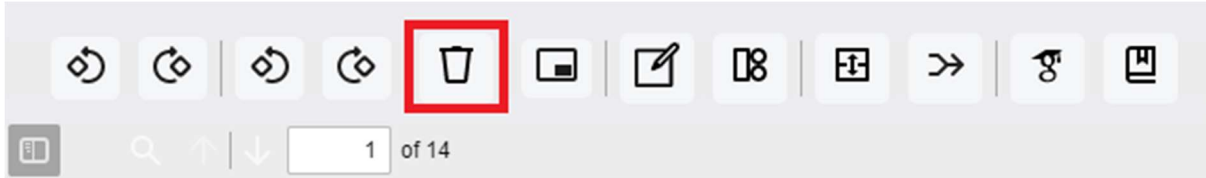


Používateľ má možnosť vybrať z nasledujúcich buttonov. V strede sa nachádza color picker, ktorý určuje farbu zvoleného objektu. Po kliknutí na choose file sa používateľovi ponúkne vybrať obrázok z počítača, ktorý chce vložiť. Objekty je možné rôzne upravovať. Je možné ich ľubovoľne zväčšovať a zmenšovať. To isté platí aj pre objekt textu, ale ten je možné aj prepísať na ľubovoľný text, ktorý používateľ potrebuje. Následne sa tu nachádzajú dva buttony Cancel a Save. Stlačením Cancel sa zahodia všetky zmeny. Stlačením Save sa zmeny uložia do canvasu zvolenej strany.



3.1.2.4. Delete

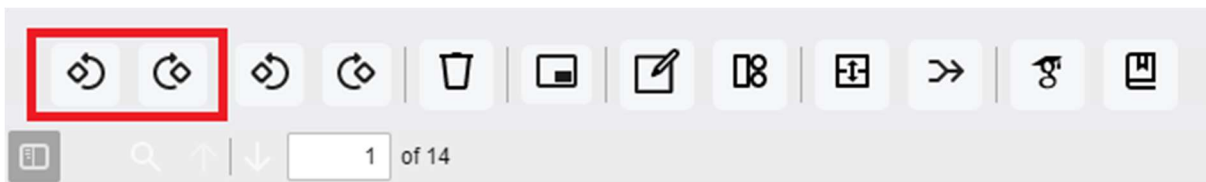
Funkcia delete je jednoduchá funkcia na odstránenie jednej strany.



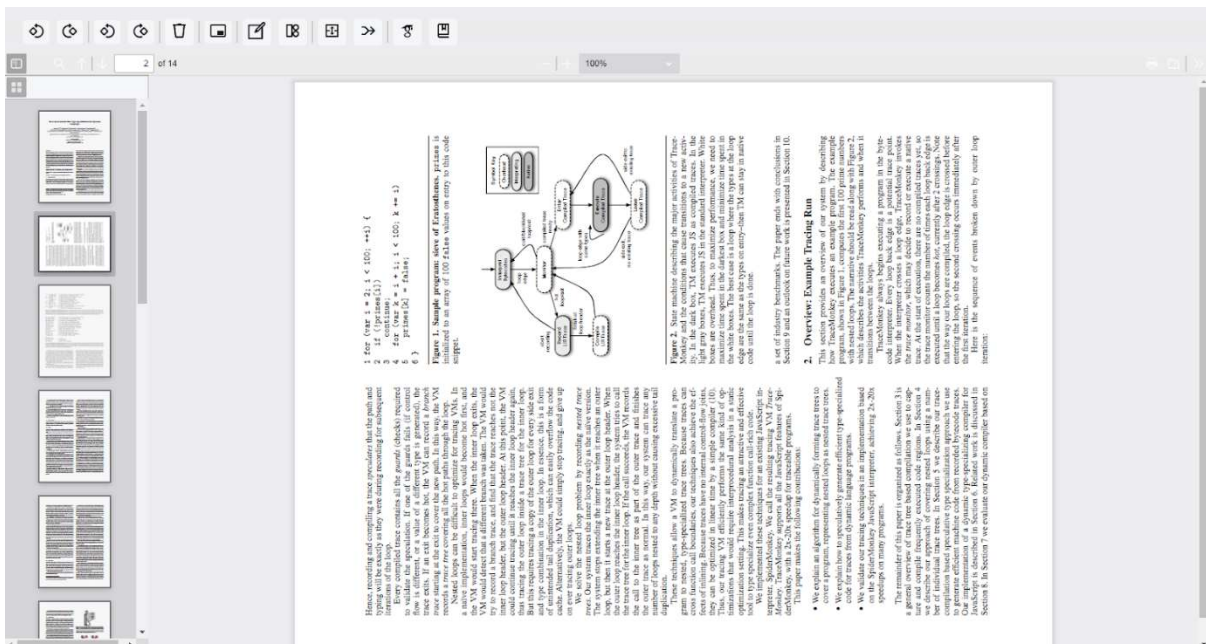
Začína kliknutím na vyznačenú ikonu. Po stlačení sa vymaže aktuálna strana.

3.1.2.5. Rotácia jednej strany

Funkcia slúži na rotáciu jednej strany. Či už v smere, alebo proti smeru hodinových ručičiek.

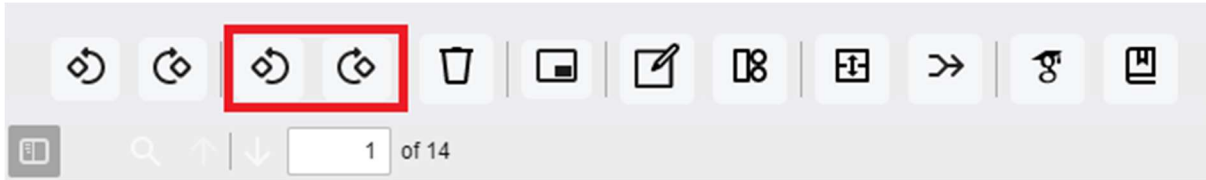


Funkcia začína kliknutím na jednu ikonu z dvoch vyznačených. Akcia sa vykoná instantne a používateľ okamžite vidí výsledok.

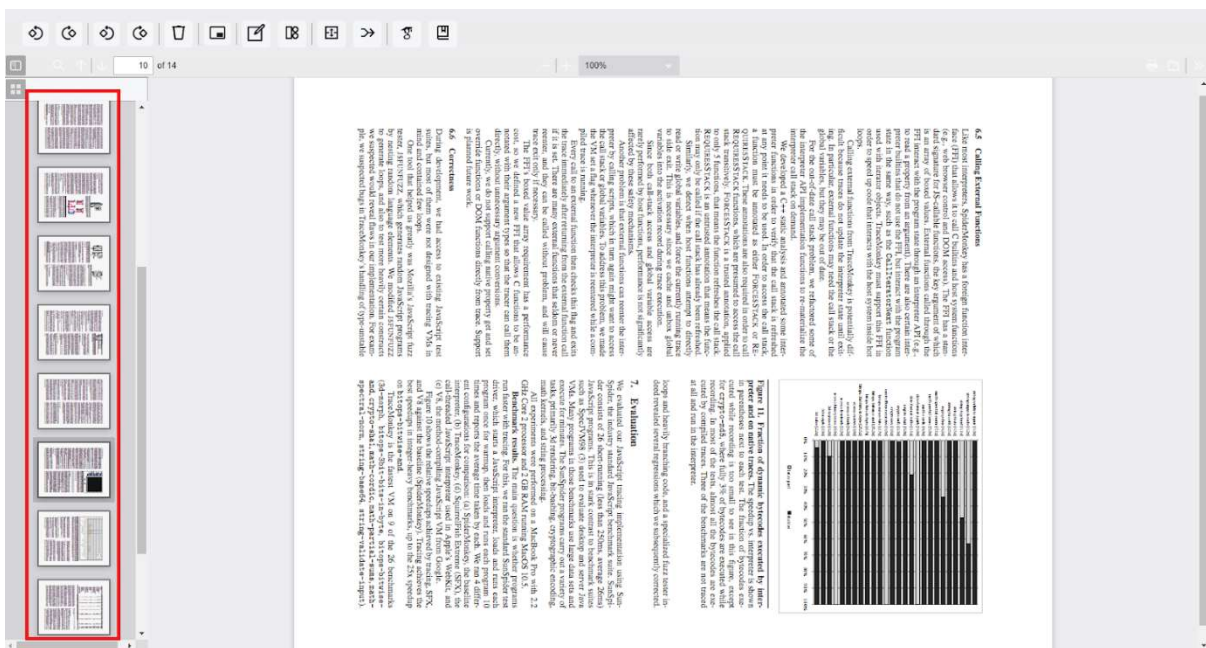


3.1.2.6. Rotácia všetkých strán

Funkcia slúži na rotáciu všetkých strán. Či už v smere, alebo proti smeru hodinových ručičiek.

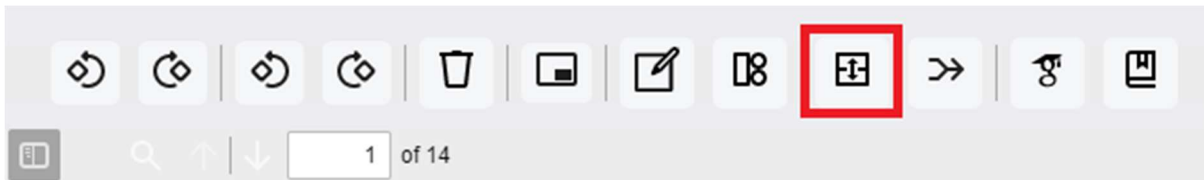


Funkcia začína kliknutím na jednu ikonu z dvoch vyznačených. Akcia sa vykoná instantne a používateľ okamžite vidí výsledok. Akcia sa aplikuje na všetky strany dokumentu.



3.1.2.7. Split

Funkcia slúži na extrakciu niekoľkých strán z dokumentu. Vyberie sa podmnožina strán, ktoré sa následne stiahnu ako samostatný dokument.



Split sa začína kliknutím na vyznačenú ikonu z hlavného menu. Hlavné menu sa zmení na menu, ktoré je exkluzívne pre split.



Používateľ vyplní údaje o novom dokumente. Zadá rozsah strán, ktoré majú byť v novom dokumente. Napríklad v tomto prípade bude nový dokument od strany 4 po stranu 9. Následne zadá názov súboru, ktorý chce uložiť. Nachádzajú sa tu 2 buttony, Split a Cancel. Po stlačení Split sa dokument oreže podľa údajov, ktoré zadal používateľ a stiahne sa do počítača. Po stlačení Cancel sa zahodia zmeny a aplikácia sa vráti do hlavného menu.

3.1.2.8. Merge

Funkcia slúži na spojenie dvoch pdf dokumentov. Dokumenty sa spoja za sebou v poradí, v ktorom ich používateľ nahrá do aplikácie.



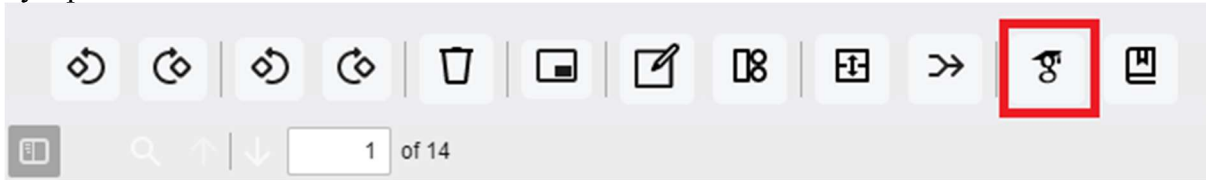
Merge sa začína kliknutím na vyznačenú ikonu z hlavného menu. Hlavné menu sa zmení na menu, ktoré je exkluzívne pre merge.



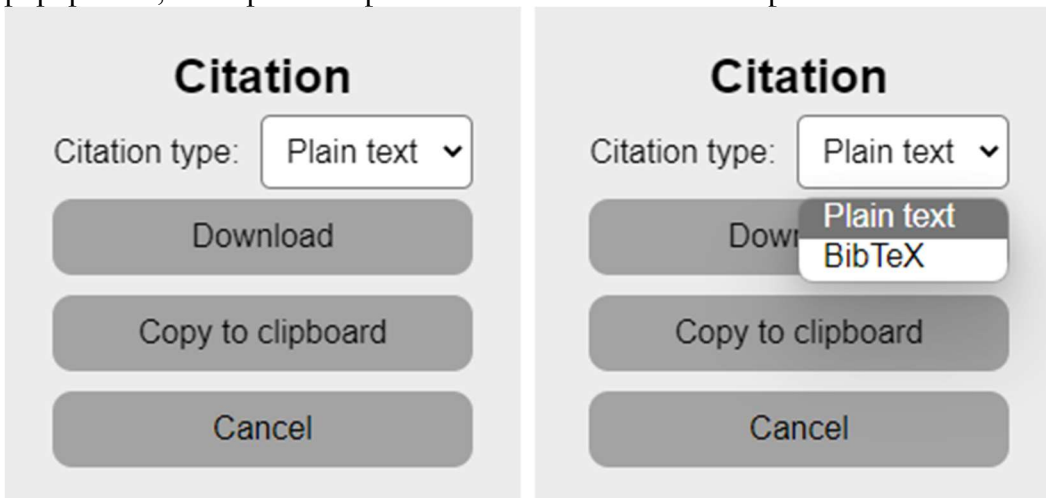
Používateľ vloží 2 dokumenty pdf, ktoré chce spojiť do jedného. do prvého poľa vloží súbor, ktorý chce na začiatku. Súbor z druhého poľa sa napojí na koniec prvého súboru. Nachádzajú sa tu 2 buttony, Save a Cancel. Po stlačení Save sa výsledný dokument stiahne do počítača. Po stlačení Cancel sa zahodia zmeny a aplikácia sa vráti do hlavného menu.

3.1.2.9. Export citácií

Vyexportovanie citácie z daného dokumentu na základe jeho metadát. Citácia môže byť vyexportovaná vo forme Plain text alebo BibTeX.



Export citácií sa začína kliknutím na vyznačenú ikonu z hlavného menu. Následne vyskočí popup okno, ktoré ponúkne používateľovi viac možností exportu.



Používateľ si zvolí či chce citáciu vo forme Plain text alebo BibTeX. Následne sa tu nachádzajú 3 buttony. Button Download stiahne citáciu do textového súboru do počítača. Button Copy to clipboard vloží citáciu do clipboardu a používateľ ju môže vložiť na ľubovoľné miesto napríklad pomocou ctrl+v. Po stlačení Cancel sa zahodia vykonané akcie a aplikácia sa vráti do hlavného menu.

3.1.2.10. Úprava metadát

Funkcia slúži na zmenu metadát konkrétneho PDF súboru. Umožňuje meniť základné informácie o súbore ako názov, meno autora, predmet, kľúčové slová a dátum vytvorenia.



Úprava metadát sa začína kliknutím na vyznačenú ikonu z hlavného menu. Následne vyskočí pop-up okno, ktoré ponúkne používateľovi možnosti na zmenu metadát.

A screenshot of a dialog box titled 'Metadata settings'. The dialog box has a light gray background and contains several input fields and buttons. The fields are: 'Title:' with an empty text box; 'Author:' with an empty text box; 'Subject:' with an empty text box; 'Keywords:' with an empty text box; and 'Creation date:' with a text box containing '04/02/2009' and a calendar icon. At the bottom of the dialog box, there are two buttons: 'Save' and 'Cancel'.

Používateľ vyplní informácie, ktoré chce zmeniť. Button Save slúži na uloženie zmien. Dokument bude obsahovať nové metadáta, ktoré zadal používateľ. Po stlačení Cancel sa zahodia vykonané akcie a aplikácia sa vráti do hlavného menu.

3.2. Inštalačná príručka

3.2.1. Úvod

Produkt Document Wizard je ideálnou voľbou na použitie pre systémy evidencie dokumentov vo formáte PDF. Príkladom môžu byť knižničné systémy, rozsiahle dokumentačné servery, zdravotnícka evidencia, evidencia checklistov a smerníc pre letecké spoločnosti alebo aj pre bežnú administratívu dokumentov v akejkoľvek modernej spoločnosti.

3.2.2. Postup inštalácie

Inštalácia modulu na zobrazovanie PDF súborov Document Wizard je jednoduchá. Vďaka spôsobu, akým je Document Wizard vyvinutý a následne zaobalený do uceleného balíka, ho pri inštalovaní považovať za samostatnú knižnicu.

Knižnicu je možné jednoducho nainštalovať pomocou nasledovných príkazov rozdelených podľa druhu package managera, ktorý vo vašom systéme používate.

Pre npm package manager: „npm install document-wizard“

Pre yarn package manager: „yarn add document-wizard“

Po inštalácii knižnice je už iba potrebné importovať náš modul spolu s jeho štýlmi a spustiť, resp. načítať modul s PDF dokumentom.

Pre viac informácií sa obráťte na stránku balíka document-wizard v npm alebo yarn package manageri.

V prípade vzniknutých problémov neváhajte kontaktovať vývojársky tím produktu Document Wizard na mailovej adrese timovy.projekt18@gmail.com

Radi vám ochotne poradíme.

4. Technická dokumentácia

Za technickú dokumentáciu sme zvolili možnosť vygenerovanej dokumentácie. Technická dokumentácia sa nachádza v priečinku „Vygenerovana_dokumentacia“. Vygenerovaná dokumentácia je vo formáte HTML a otvoriť ju možno cez súbor „index.html“.

Technická dokumentácia bola vygenerovaná pomocou nástroja Typedoc.

5. Zoznam príloh

A. Vygenerovaná dokumentácia

- na automaticky vygenerovanú dokumentáciu sme použili nástroj TypeDoc
- vygenerovaná dokumentácia obsahuje základné informácie ohľadom naprogramovaných tried a metód
- nachádza sa v priečinku s názvom „Vygenerovana_dokumentacia“

B. Protokol o používateľskom testovaní

- výsledný protokol poskytujúci súhrn hodnotenia z vykonaného používateľského testovania
- nachádza sa v priečinku s názvom „Ine_dokumenty“

C. Používateľská príručka

- používateľská príručka stručným spôsobom vysvetľuje používanie produktu Document Wizard
- okrem zodpovedajúcej sekcie v tomto dokumente je možné ju nájsť v samostatnom dokumente, ktorý je umiestnený v priečinku s názvom „Ine_dokumenty“

D. Inštalačná príručka

- inštalačná príručka poskytuje návod na úspešnú inštaláciu produktu Document Wizard do cieľového systému
- okrem zodpovedajúcej sekcie v tomto dokumente je možné ju nájsť v samostatnom dokumente, ktorý je umiestnený v priečinku s názvom „Ine_dokumenty“