

**Dokumentácia**  
**Theses PDF viewer**

**Tvorba informačných systémov 2019/2020**

**Iveta Krempaská**

**Matej Kormuth**

**Bálint Áron Zajíc**

**Peter Tamáš**

## Obsah

Dokumentácia.....	1
Theses PDF viewer .....	1
Katalóg požiadaviek.....	3
2.1. Perspektíva systému.....	6
2.2. Funkcie systému .....	6
2.3. Charakteristika používateľa .....	6
2.4. Všeobecné obmedzenia .....	6
2.5. Predpoklady a závislosti .....	7
4 Návrh implementácie.....	18
4.1 Prehľad použitých technológií.....	18
4.2 Diagramy .....	18
4.2.1 Triedny diagram .....	18
4.2.2 Stavový diagram.....	19
4.2.3 Komponentový diagram .....	20
4.2.4 Funkcie vo Frond-Ende.....	20
4.3 Funkcie v Back-Ende.....	21
Testovacie Scenáre .....	22
5 Testovanie .....	23
5.1 Testovanie Stránkovania .....	23
5.2 Testovanie Navigácie.....	23
5.3 Testovanie zobrazovania autora, názov práce a druh (t. j. diplomová, bakalárska, dizertačná alebo rigorózna) práce .....	24
5.4 Testovanie kopírovania dokumentu .....	24
5.5 Testovanie Informácie o načítavaní.....	24
5.6 Testovanie Cache pamäte.....	24

# **Časť I**

## **Katalóg požiadaviek**

# **1. Úvod**

## **1.1. Účel tohto katalógu požiadaviek**

Tento katalóg požiadaviek popisuje požiadavky zadávateľa na funkcionality projektu „PDF viewer pre záverečné práce“, ako návrh na schválenie a pre vývojový tím ako referencia pre vývoj informačného systému.

Predmetom tohto dokumentu je určiť a špecifikovať požiadavky na softvérový projekt.

Obsah dokumentu vznikol po dohode so zadávateľom softvérového projektu. Špecifikované požiadavky budú počas vývoja, ako aj po ukončení vývoja softvérového produktu slúžiť na vyhodnocovanie správnosti softvéru zadávateľom aj realizátorom.

## **1.2. Rozsah využitia systému**

Produktom je systém pre Akademickú knižnicu UK umožňujúci sprístupňovať e-verzie záverečných prác v PDF formáte v prostredí webu tak aby ich nebolo možné jednoducho tlačiť, kopírovať a šíriť.

Záverečné práce v e-verzii (PDF) sú sprístupňované prostredníctvom knižničného online katalógu, pričom záznamy o prácach obsahujú odkaz (URL) na sprístupňovaný PDF súbor.

## **1.3. Slovník pojmov**

Vysvetlenie pojmov:

Cache – pamäť, ktorá slúži na prechodné ukladanie dát

API – konkrétne rozhranie pre programovanie aplikácii

CORS – mechanizmus na zdieľanie zdrojov webovej stránky pre aplikáciu na inej doméne

Stránkovanie - rozdelenie dokumentu na jednotlivé strany

Parsovanie – proces analýzy vstupu na určenie gramatickej štruktúry vzhľadom na danú gramatiku

#### **1.4. Odkazy a referencie**

Pri práci a tvorbe systému sa budú autori inšpirovať podobným projektom, a využijú rôzne dostupné technológie.

Centrálny register záverečných a kvalifikačných prác - <https://cms.crzp.sk>

#### **1.5. Prehľad nasledujúcich kapitol**

V ďalších kapitolách je opísaná základná funkcionálna a požiadavky, ktoré musí systém spĺňať.

## **2. Všeobecný popis**

### **2.1. Perspektíva systému**

Projekt s názvom „PDF viewer pre záverečné práce“, predstavuje systém, ktorý bude slúžiť pre Akademickú knižnicu UK. Oblasťou záujmu bude vytvoriť také technologické riešenie, ktoré umožní splniť klauzulu o ochrane a tak sprístupniť všetky záverečné práce v e-verzii.

### **2.2. Funkcie systému**

Systém bude mať prístup k záverečným prácam v elektronickej podobe z knižničného serveru. Ak si užívateľ zvolí na prehliadanie takú prácu, ktorú jej autor označil za ochranu proti kopírovaniu, systém neumožní užívateľovi priamy prístup k PDF súboru s možnosťou stiahnutia. Namiesto toho z každej strany PDF súboru vytvorí fotokópiu, ktorej zníži rozlíšenie, prípadne pridá vodoznak a tú sprístupní užívateľovi, bez možnosti ju kopírovať alebo uložiť. Zároveň tieto fotokópie dočasne uloží na úložisko, pre prípad, že by si danú prácu chcel pozrieť iný užívateľ, bez potreby znova prerábať pôvodný PDF súbor na fotokópie. V prípade, že fotokópie nebude nikto iný prehliadať, budú automaticky v určitom intervale odstránené. V prípade, že na úložisku určenom na fotokópie už nebude dostatok miesta na ďalšiu prácu, budú niektoré fotokópie taktiež okamžite odstránené.

### **2.3. Charakteristika používateľa**

V systéme bude len jeden druh používateľa, ktorý bude mať prístup ku všetkým funkciám softvéru.

Primárne je softvér určený pre akademickú knižnicu, v ktorej záujme je sprístupniť záverečné práce v elektronickej podobe a súčasne ich chrániť pred plagiátorstvom, tak ako to vyplýva zo zákona. Ďalej je softvér určený pre užívateľov, ktorí chcú práce prehliadať.

### **2.4. Všeobecné obmedzenia**

Náš softvér bude obmedzovať jedno obmedzenie:

- Autor môže podmieniť sprístupnenie záverečnej práce len za podmienky využitia takých technológií ktoré zabránia tlačiť, rozmnožiť, kopírovať, uložiť na ďalšie médium danú prácu

## **2.5. Predpoklady a závislosti**

Predpokladáme, že systém bude bežať nepretržite. Server, na ktorom sú uložené záverečné práce je prístupný iba zo siete UK. Systém bude mať obmedzený priestor na uchovávanie fotokópií, čiže bude potrebná správa pamäte. Ďalej predpokladáme, že k systému bude môcť pristupovať viacero užívateľov naraz. V závislosti od dĺžky prípravy fotokópií z PDF súboru môže byť užívateľ odradený, preto ak nebudú ešte vytvorené, bude ich treba sprístupňovať postupne okamžite po vytvorení, namiesto čakania na premenu celého PDF súboru.

### 3. Špecifické požiadavky

Vysvetlenie označení:

BE - backend

FE – frontend

IR – požiadavky rozhrania

NF – požiadavky nevzťahujúce sa na funkcionálnosť

#### 3.1. Funkčné požiadavky

BE-01: Aplikácia umožní načítať PDF súbor z lokálneho disku z adresáru určeným konfiguračným súborom (v súbore bude napísané, v akom adresári sa adresárová štruktúra s prácami na disku nachádza) podľa URL parametra aplikácie podľa bodu 3.4.

BE-02: Aplikácia sa spustí načítaním webovej stránky aplikácie s URL parametrom, ktorý bude obsahovať URL adresu zadaného dokumentu v online katalógu (v tvare napr. <http://davinci.fmph.uniba.sk/aplikacia/?document=http://alis.uniba.sk/storage/ddp/dostupne/FM/2013/2013-FM-16975/>). Podľa tejto URL adresy načíta PDF súbor.

BE-03: Aplikácia umožní vyrendrovanie načítaného PDF dokumentu do viacerých obrázkov vo formáte JPEG v rozlíšení 72 DPI po jednotlivých stranách dokumentu (každá strana dokumentu je samostatný obrázok).

BE-04: Aplikácia cachuje vyrendrované obrázky aby sa zabránilo preťažovaniu procesora a aby sa zrýchlila odozva pri prezeraní.

BE-05: Aplikácia pravidelne čistí cache aby sa zabránilo preplneniu disku.

BE-06: Aplikácia vystaví API endpoint pomocou vhodnej knižnice (express.js) do internetu tak aby bolo možné requestovať jednotlivé stránky jednotlivých dokumentov.

BE-07: (optional) Aplikácia vystaví API endpoint, ktorý umožní načítať metadáta o PDF dokumentoch (počet strán, názov, autor, kapitoly a prislúchajúce strany kapitol)



BE-08: Aplikácia zabezpečí nastavenie cross-origin resource sharing (CORS) tak aby bolo možné pristupovať k API z frontendovej časti.

FE-01: Aplikácia poskytne komponent, ktorý umožňuje stránkovanie (užívateľ sa môže dostať jedným kliknutím o 1, 2, 3 alebo 4 strany dopredu a 1, 2, 3 alebo 4 strany dozadu. okrem toho sa môže vždy dostať jedným kliknutím na prvú a poslednú stranu dokumentu).

FE-02: Aplikácia nadviaže spojenie a nakonfiguruje API klienta na komunikáciu s backendom.

FE-03: Aplikácia rozparsuje informácie o dokumente, ktorý sa má prehliadať podľa parametrov, ktoré sú predané našej aplikácii pomocou parametrov URL adresy (parametrom je URL adresa na knižničný server, kde sa dokument nachádza).

FE-04: Aplikácia zabráni kopírovania (right-click).

FE-05: Aplikácia poskytne informáciu o tom, že prebieha načítavanie (užívateľ uvidí na obrazovke točiace sa koliesko, ktoré bude indikovať že prebieha načítavanie obsahu a musí byť trpezlivý).

FE-06: (optional) Aplikácia môže zobrazit' obsah s názvami kapitol v paneli, ktorý sa bude nachádzať na ľavej alebo pravej strane obrazovky na obrazovkách so šírkou rozlíšenia väčšou ako 500px. Na mobilných zariadeniach sa zobrazovať nebude lebo ich obrazovky sú zvyčajne veľmi malé a to by mohlo znehodnotiť používateľský zážitok.

FE-07: (optional) Aplikácia môže zobrazit' obsah s názvami kapitol takým spôsobom, že sa po presunutí kurzora na názov kapitoly a následným stlačením ľavého tlačidla myši prehliadač presunie na tú stranu dokumentu, na ktorej začína kapitola na názov ktorej užívateľ klikol.

FE-08: Aplikácia môže zobrazit' názov práce a autora v záhlaví prehliadača.

FE-09: Aplikácia bude obsahovať odkaz späť na online katalóg.

### **3.2. Požiadavky nevzťahujúce sa na funkcionality**

NF-01: Systém bude dodaný ako 2 aplikácie (backend a frontend) nasadené na takom serveri, ktorý bude mať k dispozícii PDF dokumenty prostredníctvom lokálneho disku (filesystému).

NF-02: Počas písania kódu sa budú dodržiavať zásady čistého kódu.

NF-03: K softvéru bude dodaná používateľská príručka, ktorá dostatočným spôsobom vysvetlí ako je možné backendovú časť konfigurovať a aplikáciu používať.

### **3.3. Požiadavky rozhrania**

IR-01: Systém bude fungovať na univerzitnom serveri, so systémom solaris, na ktorom bude mať prístup k záverečným prácam a bude mať vyhradený priestor na operáciu.

IR-02: Užívateľ k nemu bude pristupovať pomocou web stránky,

IR-03: Zvolená záverečná práca bude užívateľovi sprístupnená bez možnosti ju kopírovať.

IR-04: Rozhranie bude dostatočne prehľadné a bude dbať na čo najrýchlejšiu odozvu zároveň bude šetriť prostriedky.

IR-05: Rozhranie bude ovládané myšou a klávesnicou, prípadne dotykom.

### **3.4. Požiadavky na načítanie PDF súborov**

Všeobecná štruktúra sprístupňovaných prác je takáto (podrobnejšie nižšie):

/zkp/storage/typ\_prace/dostupne/FF/RRRR/RRRR-FF-identifikator/subor.xxx

resp. pre prace s embargom v počte mesiacov (max 36) odo dňa obhajoby

(rozdiel v názve posledného podadresára):

zkp/storage/typ\_prace/dostupne/FF/RRRR/RRRR+MM-FF-identifikator/subor.xxx

Konkrétne napr.:

/zkp/storage/ddp/dostupne/FM/2019/2019-FM-93761/125249v1.pdf

resp.:

/zkp/storage/ddp/dostupne/FM/2019/2019-FM-93761

Keďže URL sa v online katalógu linkuje na úroveň posledného  
podadresára, nie na konkrétne PDF:

<http://alis.uniba.sk/storage/ddp/dostupne/FM/2019/2019-FM-93761/>

Chránené práce sú umiestňované v podobnej štruktúre, ale do adresára  
/zkp/storage-chranene, ktorý je nedostupný pre web server.

Adresár "storage-chranene" však NEMA úroveň 2) podľa typu licencie  
(dostupne), adresár indikujúci typ práce sa delí na fakulty:

/zkp/storage-chranene/dzb/FM/2019/2019-FM-99715/133292v1.pdf

/zkp/storage-chranene/dzb/FM/2019/2019+12-FM-97736/130504v1.pdf

## **4. Vysvetlivky**

/zbp/storage sa delí na podadresáre:

### **4.1. Podľa typu práce:**

- dzb = bakalárske práce
- ddp = diplomové práce
- dpg = dizertačne práce
- drz = rigorózne práce
- other = iné typy práce (licenčné , habilitačne, atestačne a pod.)

### **4.2. Každý podadresár podľa typu práce má podadresáre podľa licencie**

- UK = prístup len pre IP z autorizovaného rozsahu UK
- archív = nesprístupňovane prace
- dostupne = prace sprístupňovane bez obmedzenia

Pričom len "dostupne" je stále aktívny podadresár; UK a archív už nie sú priebežne naplňane, vzťahujú sa k LZ pred roku 2012.

### **4.3. Podadresár "dostupne" sa ďalej delí na 13 podadresárov podľa fakulty**

Názov podadresárov = 2-znakový kód fakulty z 3-4 pozície UKO kódu (viac pozri

<https://uniba.sk/o-univerzite/fakulty-a-dalsie-sucasti/akademicka-kniznica-uk/kody-pracovisk-uk/>):

EB, FA, FI, FM, FS, JL, LF, MA, PA, PD, PR, RK, TV

#### **4.4. Fakultné podadresáre obsahujú podadresáre podľa rokov obhájenia práce: 2007, 2008 ... 2018, 2019**

#### **4.5. V rámci podadresára roku je adresár so samotnými prácami**

Nie sú iba PDF súbory, lebo práce majú často aj prílohy v inom formáte (xlsx, docx, jpeg, mp3 apod.). Názov adresára vždy obsahuje rok (prípadne po znamienko + dobu embarga v mesiacoch), 2-znakový kód fakulty a jednoznačne identifikačné číslo práce ako celku generované AiS2.

Napr.:

- 2019-FM-93761
- 2019-FI-96719
- 2019+6-FI-100729

#### **4.6. Podadresár s pracou obsahuje už samotné súbory s pracou (PDF, prípadne prílohy), názvy sú jedinečné, generuje ich AiS2**

Napr.:

- 2019-FM-93761
- 2018+6-FI-91945
- 2018+12-FI-86963

Výpis obsahu posledného podadresára štruktúry:

```
# ls /zkp/storage/ddp/dostupne/FM/2019/2019-FM-93761
```

```
125249v1.pdf    priloha_22743.zip
```

**Čast' II**

**Návrh**

# **1. Úvod**

## **1.1. Účel dokumentu**

Tento dokument predstavuje podrobný popis návrhu na systém PDF viewer záverečných prác. Pomocou diagramov a detailných popisov v tomto dokumente je vysvetlené, akým spôsobom bude systém vyvinutý a ako bude následne fungovať tak, aby splnil všetky požiadavky uvedené v katalógu požiadaviek.

## **1.2. Zameranie a rozsah**

Predpokladáme, že čitateľ tohto dokumentu má dôkladne prečítaný katalóg požiadaviek a teda má ucelenú predstavu ako má softvér fungovať. Veľkú časť tohto dokumentu tvoria UML diagramy, v ktorých je dôkladne predstavené, na aké moduly a triedy bude systém rozdelený. V časti návrh implementácie sú predstavené všetky diagramy na základe, ktorých je dobre vysvetlený priebeh celého systému.

## **1.3. Definície a skratky**

Frontend – vrstva softvéru na prezentáciu dát

Backend – vrstva softvéru na prístup k a spracovania dát.

# **2. Špecifikácia vonkajších interfejsov**

## **2.1. Frontend**

Frontend modul tvorí vizuálnu časť programu a slúži na jednoduchšiu manipuláciu s programom

Implementácia:

Náš modul sa bude skladať z jednej stránky:

- Modul dáta posiela do „backendu“ pomocou navigačných tlačidiel na listovanie dokumentu
- Aplikácia poskytne informáciu o tom, že prebieha načítavanie ( po kliknutí na jeden z navigačných tlačidiel), užívateľ uvidí na obrazovke točiac sa koliesko (ktoré bude indikovať, že prebieha načítanie obsahu).
- Aplikácia bude obsahovať v bočnom paneli obsah s názvami kapitol, taktiež bude zobrazovať názov práce a autora
- Aplikácia bude obsahovať tlačidlo na vrátenia sa do online katalógu

## **2.2. Backend**

Backend vystaví API, ktoré odpovedá cez protokol http na žiadosti.

Backend modul pracuje s dátami, ktoré sú zaslané z navigačných tlačidiel (frontend) cez metódu GET.

Implementácia:

- Stránka vyrenderuje požadovaný PDF dokument
- Uloží ho a následne, sú obrázky dostupné na zobrazenie
- Aplikácia po nejakej dobe tieto obrázky vymaže aby sa zabránilo preplneniu disku
- Aplikácia poskytne rozhranie na prezeranie týchto dokumentov (frondend)

## **2.3. Detaily parsovania URL parametra (vonkajší interfejs)**

Dostaneme: <http://alis.uniba.sk/storage/ddp/FM/2013/2013-FM-16975/>

Zoberieme koncovu časť FM/2013/2013-FM-16975/ a podľa nej na disku otvoríme adresár /zbp/storage-chranene/dzb/FM/2013/2013-FM-16975/ ,ňom nájdeme všetky PDF súbory. Zo všetkých PDF súborov vyberieme taký, ktorý nezačína prefixom “príloha” a ten bude PDF súbor práce a následne z neho vygenerujeme strany.



V tejto časti predstavujeme náčrt používateľského rozhrania, spoločne s popisom jednotlivých častí. Bližšie informácie k funkciám možno nájsť aj v katalógu požiadaviek, z ktorého návrh vychádza.

**Jozko Mrkvicka - Význam voľnočasových aktivít pre deti a mládež zo sociálne znevýhodnených rodín**

## Abstract

## Uvod

## UML Diagramy

## Navrh

## Implementacia

Zaver

« Previous 1 2 **3** 4 5 .... 12 13 14 Next »

Na mobile bude vyzerat' rovnako len tam nebude voliteľný bočný panel s názvami kapitol.

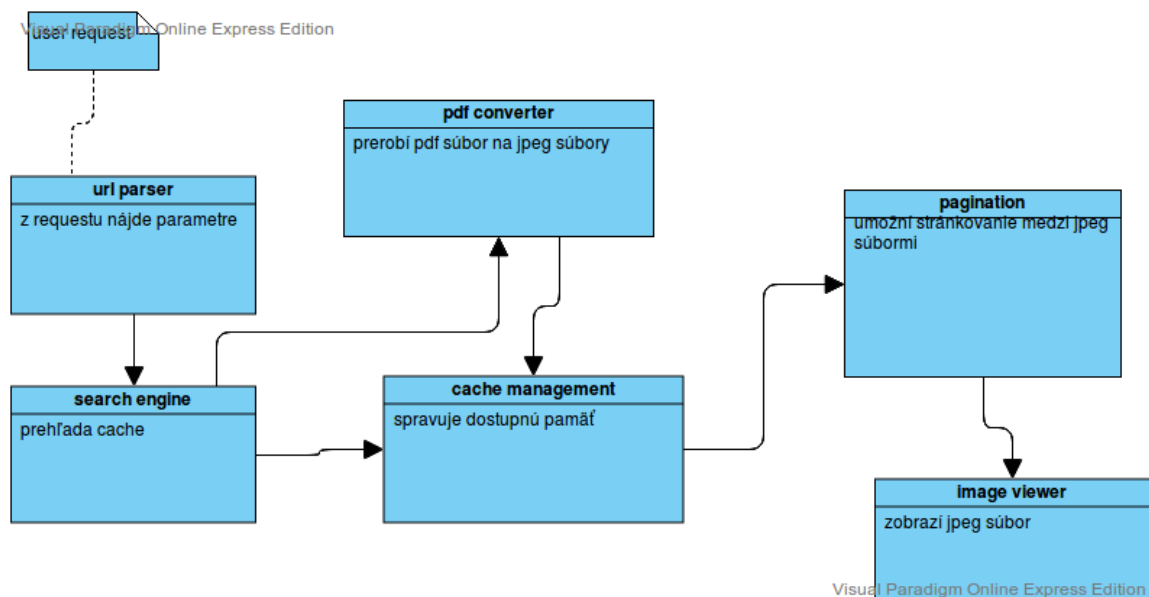
## 4 Návrh implementácie

### 4.1 Prehľad použitých technológií

HTML/CSS, JavaScript, Python/CherryPy

### 4.2 Diagramy

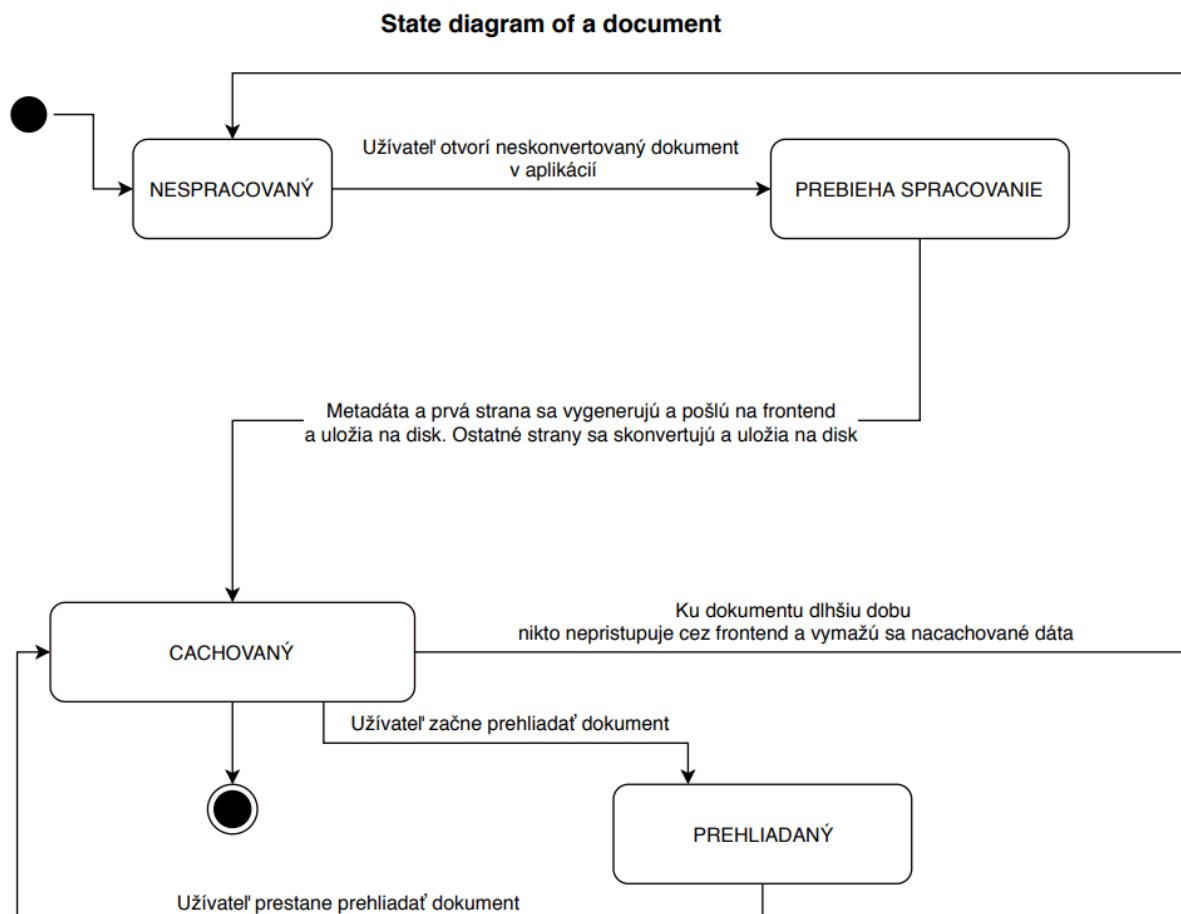
#### 4.2.1 Triedny diagram



Popis:

Triedny diagram popisuje vzťahy medzi triedami.

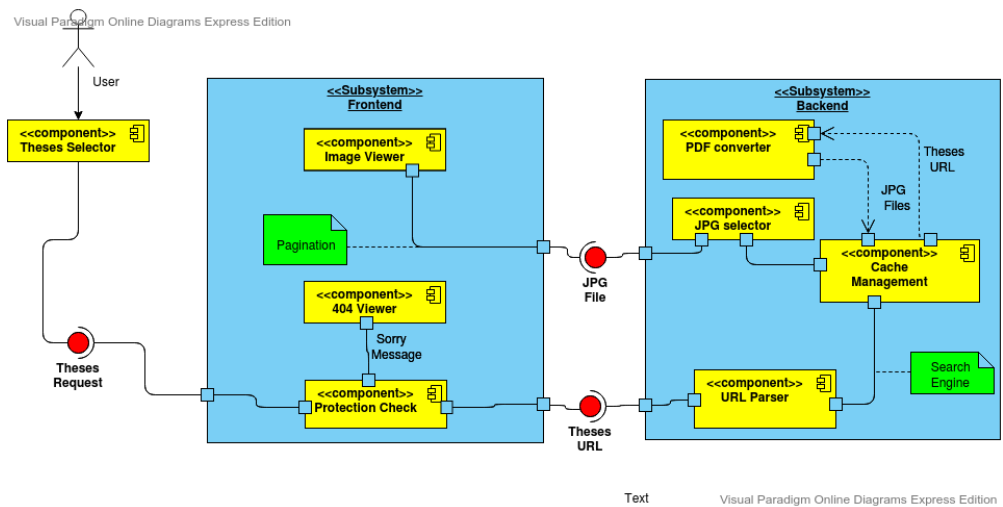
#### 4.2.2 Stavový diagram



Popis:

Stavový diagram reprezentuje životný cyklus entít dokumentu. Každý stav entity je vyjadrený oválom. Vstup entity do stavového diagramu sa realizuje cez počiatkový stav (kruh) a výstup cez koncový stav (kruh v kružnici). Prechody medzi stavmi vyvolávajú používatelia a pravidlá aplikácie.

### 4.2.3 Komponentový diagram



Popis:

Frontend zodpovedný za správu údajov aplikácie, dostáva používateľské vstupy, reprezentuje údaje z daného modelu ako výstupný vo formáte HTML/CSS.

Backend dostáva vstupy od používateľa a premieňa ich na príkazy pre frontend.

### 4.2.4 Funkcie vo Frond-Ende

- **App()** - funkcia vyrendruje Reactovský komponent aplikácie
- **Header()** - funkcia vyrendruje Reactovský komponent hlavičky
- **Chapters()** - funkcia vyrendruje Reactovský komponent zoznamu kapitol
- **getMetadata(publication)** - funkcia získa metadata o dokumente z backendu
- **getPage(publication, page)** - funkcia získa URL / obrázok zadanej strany z backendu

- **goToPage(page)** - funkcia zobrazí danú stranu v prehliadací
- **loadDocument(publication)** - funkcia načíta a zobrazí prvú stranu zadaného dokumentu
- **parseDocumentIdFromUrl()** - funkcia vyparsuje ID dokumentu z URL adresy

### 4.3 Funkcie v Back-Ende

Využijeme už naprogramovaný konverter, ktorý využíva Linuxové knižnice `convert`, `gs` a `pdftinfo`.

- **def index(self, praca, strana)** - vráti obrázok jednotlivej prace podľa strany
- **def metadata(self, praca)** - vráti metadáta ako počet strán

## **ČASŤ III.**

### **Testovacie Scenáre**

## 5 Testovanie

### 5.1 Testovanie Stránkovania

- **Vstup:** Kliknutie ľavým tlačidlom myši na ľubovoľné číslo strany, ktoré sú v aplikácii práve zobrazené
- **Výstup :** Zobrazenie daného dokumentu o x ( x je hodnota z 1-4 strán) strán ďalej

**FUNGUJE SPRÁVNE**

- **Vstup:** Kliknutie ľavým tlačidlom myši na poslednú stranu
- **Výstup:** Zobrazenie sa posledná strana dokumentu

**FUNGUJE SPRÁVNE**

- **Vstup:** Kliknutie ľavým tlačidlom myši na prvú stranu
- **Výstup:** Zobrazenie sa prvá strana dokumentu

**FUNGUJE SPRÁVNE**

### 5.2 Testovanie Navigácie

- **Vstup:** Napísanie čísla 10 pomocou klávesnice a kliknutie ľavým tlačidlom myši na „Prejsť na stranu“
- **Výstup:** Zobrazenie dokumentu na 10 strane

**FUNGUJE SPRÁVNE**

- **Vstup:** Napísanie neexistujúcej strany napr. 10 126 a kliknutie ľavým tlačidlom myši na „Prejsť na stranu“
- **Výstup:** Vypísanie hlášky „Neexistujúca strana“

### **5.3 Testovanie zobrazovania autora, názov práce a druh (t. j. diplomová, bakalárska, dizertačná alebo rigorózna) práce**

- **Vstup:** Načítanie dokumentu
- **Výstup:** Zobrazenie v záhlaví autora, názov práce a druh práce

**FUNGUJE SPRÁVNE**

### **5.4 Testovanie kopírovania dokumentu**

- **Vstup:** Onačenie textu pomocou myši alebo klávesnice
- **Výstup:** Text sa nedá označiť

**FUNGUJE SPRÁVNE**

- **Vstup:** Kliknutie pravým tlačidlom myši v aplikácii na „preskúmať“
- **Výstup:** Zobrazený zdrojový HTML kód neobsahuje informácie o dokumente

**FUNGUJE SPRÁVNE**

### **5.5 Testovanie Informácie o načítavaní**

- **Vstup:** Kliknutie na zobrazenie dokumentu
- **Výstup:** Zobrazená informácia o priebehu načítavania, točiac sa koliesko

**FUNGUJE SPRÁVNE**

### **5.6 Testovanie Cache pamäte**

- **Vstup:** Načítanie dokumentu, následné zatvorenie, a znovu otvorenie
- **Výstup:** Práca sa otvorí bez načítavania

**FUNGUJE SPRÁVNE**

-