



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**

DEPARTMENT OF INTELLIGENT SYSTEMS

**KLIENT SERVER APLIKACE PRO TVORBU SPISOVÝCH  
STRÁNEK PRO PČR**

CLIENT SERVER APPLICATION FOR THE CREATION OF FILE PAGES FOR THE PČR

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. FILIP TERBR**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. ŠTĚPÁN RYDLO**

BRNO 2023

## Zadání diplomové práce



146220

Ústav: Ústav inteligentních systémů (UITS)  
Student: **Terbr Filip, Bc.**  
Program: Informační technologie a umělá inteligence  
Specializace: Informační systémy a databáze  
Název: **Klient server aplikace pro tvorbu spisových stránek pro PČR**  
Kategorie: Uživatelská rozhraní  
Akademický rok: 2022/23

### Zadání:

1. Seznamte se s problematikou vytváření spisových zpráv k případům a převážně se soustředte na vytváření fotodokumentací ke spisům.
2. Prostudujte tvorbu aplikací typu klient-server a různé programovací jazyky pro návrh grafických uživatelských rozhraní (GUI). Dále prostudujte možnosti exportování do formátů vhodných pro tisk s předem definovanou kvalitou tisku.
3. Navrhněte serverovou aplikaci pro správu a tvorbu uživatelů, ukládání spisových zpráv a vytváření záznamů o provedených změnách. Navrhněte způsob zabezpečení komunikace mezi serverem a klientskou aplikací.
4. Navrhněte klientskou aplikaci s GUI, která umožní přihlášení uživatel do systému s následnou možností vytváření, editace a sdílení spisových stránek. V případě vložení obrázku, implementujte předvyplnění šablony podle nalezených označení stop na obrázku. Součástí aplikace vytvořte možnost exportu spisu nebo částí spisu s možností definice kvality pro tisk.
5. Implementujte serverovou a klientskou aplikaci podle předchozích bodů zadání.
6. Nasadte vámi vytvořenou aplikaci. Zhodnoťte dosažené výsledky a navrhněte případná rozšíření aplikace.

### Literatura:

- *Building Client/Server Applications with VB .NET: An Example-Driven Approach*. Ilustrované vydání. Apress, 2003. ISBN 1590590708, 9781590590706.
- *Object-oriented GUI Design: Object Technology Series*. McGraw-Hill Inc., US, 1998. ISBN 0070592748, 978-0070592742.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 až 4.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Rydlo Štěpán, Ing.**  
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.  
Datum zadání: 1.11.2022  
Termín pro odevzdání: 17.5.2023  
Datum schválení: 3.11.2022

## Abstrakt

Diplomová práce se věnuje návrhu a implementaci systému pro editaci fotografických dokumentací Policie ČR. V práci je obsažen rozbor aktuálně používaných technologií, návrh klientské aplikace, návrh serverové části systému a popis realizace klientské a serverové části systému. Výsledná implementace serverové části je napsána v jazyce JavaScript s podporou frameworku Express.JS, výsledná klientská aplikace je napsána v jazyce JavaScript s použitím frameworku Electron.

## Abstract

The diploma thesis is devoted to the design and implementation of a system for editing photographic documentation of the Police of the Czech Republic. The work includes an analysis of currently used technologies, a design of the client application, a design of the server part of the system and a description of the implementation of the client and server parts of the system. The resulting implementation of the server part is written in JavaScript with support for the Express.JS framework, the resulting client application is written in JavaScript using the Electron framework.

## Klíčová slova

fotografická dokumentace, fotografie, editor, uživatelský systém, klient-server architektura, klient, server, JavaScript, PostgreSQL, rozpoznání objektů, TensorFlow.JS, Express.JS, Electron, HTML, CSS, PDF

## Keywords

photo documentation, photography, editor, user system, client-server architecture, client, server, JavaScript, PostgreSQL, object recognition, TensorFlow.JS, Express.JS, Electron, HTML, CSS, PDF

## Citace

TERBR, Filip. *Klient server aplikace pro tvorbu spisových stránek pro PČR*. Brno, 2023. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Štěpán Rydlo

# Klient server aplikace pro tvorbu spisových stránek pro PČR

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Štěpána Rydla. Další informace mi poskytl Bc. Martin Blahák a pracovníci fotografických expertiz OKTE KŘP JmK. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Filip Terbr  
13. května 2023

# Obsah

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Úvod</b>                                   | <b>3</b>  |
| <b>2</b> | <b>Rozbor problému</b>                        | <b>4</b>  |
| 2.1      | Požadavky na řešení . . . . .                 | 4         |
| 2.2      | Vzhled fotografické dokumentace . . . . .     | 5         |
| 2.3      | Štítky pro označování stop . . . . .          | 6         |
| 2.4      | Desktopová aplikace . . . . .                 | 7         |
| 2.5      | Uživatelské rozhraní . . . . .                | 7         |
| 2.6      | Frontend a backend . . . . .                  | 8         |
| 2.7      | Architektura klient-server . . . . .          | 8         |
| 2.8      | JSON Web Tokeny . . . . .                     | 9         |
| 2.9      | Zpracování hesel . . . . .                    | 10        |
| 2.10     | REST API . . . . .                            | 11        |
| 2.11     | Zabezpečení komunikace . . . . .              | 12        |
| 2.12     | Počítačové vidění a detekce objektů . . . . . | 12        |
| <b>3</b> | <b>Návrh aplikace</b>                         | <b>14</b> |
| 3.1      | Aktéři systému . . . . .                      | 14        |
| 3.2      | Navrhované technologie . . . . .              | 16        |
| 3.2.1    | Tvorba nativních aplikací . . . . .           | 16        |
| 3.2.2    | Frontend . . . . .                            | 17        |
| 3.2.3    | Databáze . . . . .                            | 18        |
| 3.2.4    | Backend . . . . .                             | 19        |
| 3.3      | Použité knihovny . . . . .                    | 20        |
| 3.4      | Obrazovky . . . . .                           | 20        |
| 3.5      | Návrh databáze . . . . .                      | 24        |
| <b>4</b> | <b>Realizace</b>                              | <b>26</b> |
| 4.1      | Databáze . . . . .                            | 26        |
| 4.2      | Server . . . . .                              | 27        |
| 4.2.1    | Ověřování uživatele . . . . .                 | 28        |
| 4.2.2    | Správa útvarů . . . . .                       | 29        |
| 4.2.3    | Správa uživatelů . . . . .                    | 30        |
| 4.2.4    | Správa dokumentací . . . . .                  | 31        |
| 4.2.5    | Správa tisku . . . . .                        | 34        |
| 4.3      | Klientská aplikace . . . . .                  | 34        |
| 4.3.1    | Rozpoznání čísel ve fotografii . . . . .      | 51        |

|   |           |
|---|-----------|
| <b>5 Závěr</b>                                  | <b>54</b> |
| <b>Literatura</b>                               | <b>55</b> |
| <b>A Požadavky od Policie ČR</b>                | <b>59</b> |
| <b>B Navrhovaný vzhled obrazovek</b>            | <b>65</b> |
| B.1 Přihlašovací obrazovka . . . . .            | 65        |
| B.2 Přehled aktivních dokumentací . . . . .     | 67        |
| B.3 Přehled archivovaných dokumentací . . . . . | 70        |
| B.4 Náhled dokumentace . . . . .                | 72        |
| B.5 Editor dokumentace . . . . .                | 74        |
| B.6 Správa uživatelů . . . . .                  | 77        |
| B.7 Správa útvarů . . . . .                     | 78        |
| <b>C Hierarchie odevzdaného média</b>           | <b>80</b> |

# Kapitola 1

## Úvod

V rámci práce Policie ČR se provádějí i kriminalistické činnosti, jako jsou vyšetřování a objasňování trestné činnosti. K těmto činnostem patří i vytváření kriminalistické dokumentace, která může obsahovat i fotografickou dokumentaci, jenž se zaměřuje na dokumentování pomocí pořízených fotografií na místě činu. Vytváření takové dokumentace se řídí danými pravidly, avšak neexistuje jakékoliv aplikační řešení pro zjednodušení práce s fotografickou dokumentací.

Cílem této práce je vytvoření systému, jenž by mohli policisté používat při vytváření fotografických dokumentací. Výsledný systém by měl poskytovat urychlení a zefektivnění práce s fotografickými dokumentacemi.

Tato práce obsahuje kompletní popis vyvíjení systému pro Policii ČR. Bylo nutné seznámit se s požadavky na aplikaci přímo od Policie ČR, které zahrnovaly i požadavky na výstup aplikace. Následně bylo potřeba vytvořit návrh systému, jenž by splňoval získané požadavky na aplikaci. Po úspěšném schválení návrhu systému Policií ČR se mohlo přejít k samotné implementaci systému.

## Kapitola 2

# Rozbor problému

Účelem této práce je zjednodušení práce s fotografickou dokumentací Policie ČR, pro kterou aktuálně neexistuje žádné aplikační řešení. Fotografická dokumentace je obrazovou přílohou spisového materiálu, jenž má za úkol fixovat dokumentované skutečnosti pro osoby, které nemohou být na místě v čase dění. Dle zákona č. 141/1961 Sb., o trestním řízení soudním (trestní řád), § 113 odst. 2 [45] má protokol o ohledání poskytovat úplný a věrný obraz předmětu ohledání, toho lze docílit pomocí fotografické dokumentace. Dle zákona č. 273/2008 Sb., o Policii ČR, § 69-70 [46] má policista před zahájením úkonů trestního řízení, za účelem získání poznatků o trestné činnosti, vyhledávat, odhalovat, dokumentovat skutečnosti nasvědčující tomu, že byl spáchán trestný čin a vyhodnocuje poznatky o zájmovém prostředí a osobách v něm se pohybujících. Zájmovým prostředím je myšleno prostředí, ve kterém lze důvodně předpokládat získání poznatků o trestném činu. Dokumentování fotografiemi lze tedy provádět při procesních úkonech jako je ohledání, vyšetřování, проверка na místě, rekonstrukce, znalecké zkoumání a další.

### 2.1 Požadavky na řešení

Řešení má být formou desktopové aplikace, založené na principech klient-server, běžící hlavně na operačním systému Windows. Aplikace má sloužit jak policistům, tak obsluze tiskáren. K tomu by měla sloužit správa uživatelů a přidělování rolí, ke kterým bude mít přístup pouze administrátor. Administrátor dokáže vytvářet nové uživatele, ale heslo si musí každý uživatel nastavit sám při přihlášení heslem, které mu nastavil administrátor.

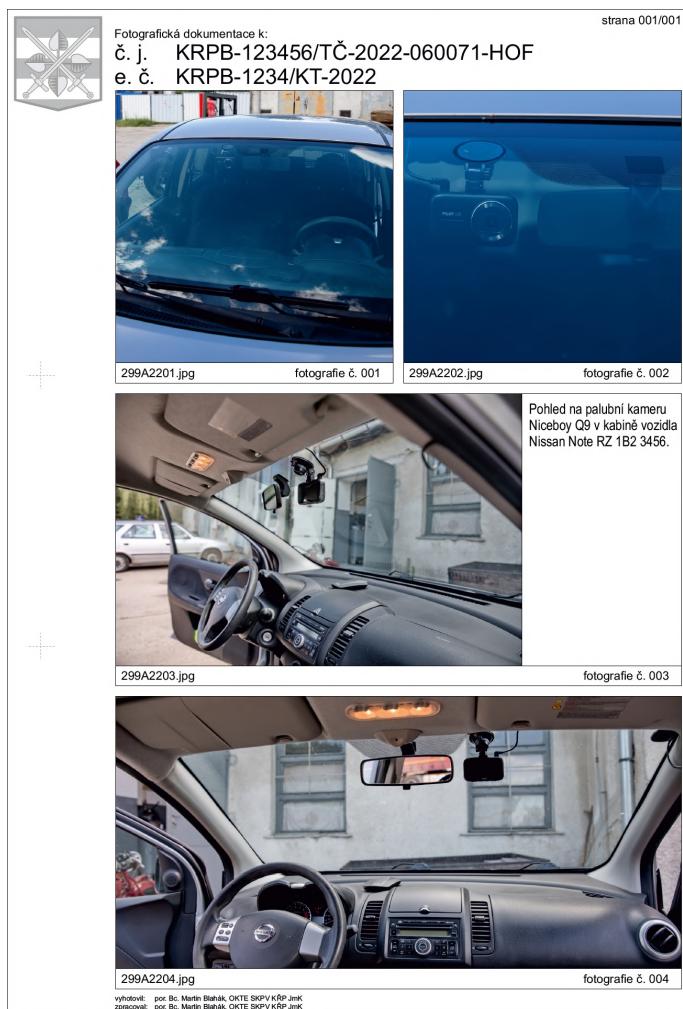
Hlavní předností aplikace má být editor dokumentací, který má nabízet Drag&Drop mechaniku pro vytváření dokumentací. Editor nabízí několik předem vytvořených předloh, které se do dokumentace vkládají a do kterých se následně mohou přidávat fotografie. Editor má obsahovat i funkci automatického vložení fotografií do dokumentace, kdy se označí fotografie, které má dokumentace obsahovat a po kliknutí na tlačítko se přidají do dokumentace. Při vkládání fotografií do dokumentace má být obsah fotografií prozkoumán detekcí objektů, při kterém hledá objekty vypadající jako štítky pro označování stop. Při nalezení shody má být tato informace předaná uživateli v podobě předvyplnění textového pole.

Po úspěšném vytvoření dokumentace musí být dokumentace zkontrolována některým vedoucím ze stejného útvaru jako je uživatel, jenž dokumentaci vytvořil. Po schválení dokumentace může vedoucí vytvořit požadavek k tisku. Požadavek se zobrazí obsluze tisku, obsluha ji vytiskne a po předání žadateli ji v systému označí jako vyřízenou. Schválené



dokumentace se následně nacházejí v archivu dokumentací, kde lze zobrazovat náhledy, vytvářet dodatečné požadavky na tisk, sdílet dokumentace útvarům či uživatelům.

## 2.2 Vzhled fotografické dokumentace



Obrázek 2.1: List fotografické dokumentace

Na základě konzultace s pracovníky fotografických expertiz OKTE KŘP JmK bylo zjištěno, že každá fotografická dokumentace musí obsahovat několik povinných náležitostí, které jsou definovány v neveřejném interním aktu řízení. Těmi jsou číslo jednací, kdo dokumentaci vyhotovil, kdo dokumentaci zpracoval, evidenční číslo nebo číslo výjezdu.

V hlavičce každého listu dokumentace se nachází text „Fotografická dokumentace k:“, kde pod tímto textem je v prvním řádku číslo jednací a v druhém řádku je evidenční číslo nebo číslo výjezdu. V levém horním rohu se nachází vodoznak Krajského ředitelství Policie ČR. V pravém horním rohu se nachází stránkování ve formátu „strana \_\_/\_\_“, kde označení aktuální strany a celkový počet mají pevný formát zobrazující 3 číslice. Každá fotografie má svou patičku, kde se nachází název fotografie a číslování fotografie v dokumentaci ve formátu

„fotografie č. \_“, číslo je zobrazeno v pevných třech číslicích. Pokud se u fotografie nachází text popisující fotografii, tak je umístěn buď vpravo od fotografie nebo pod fotografií.

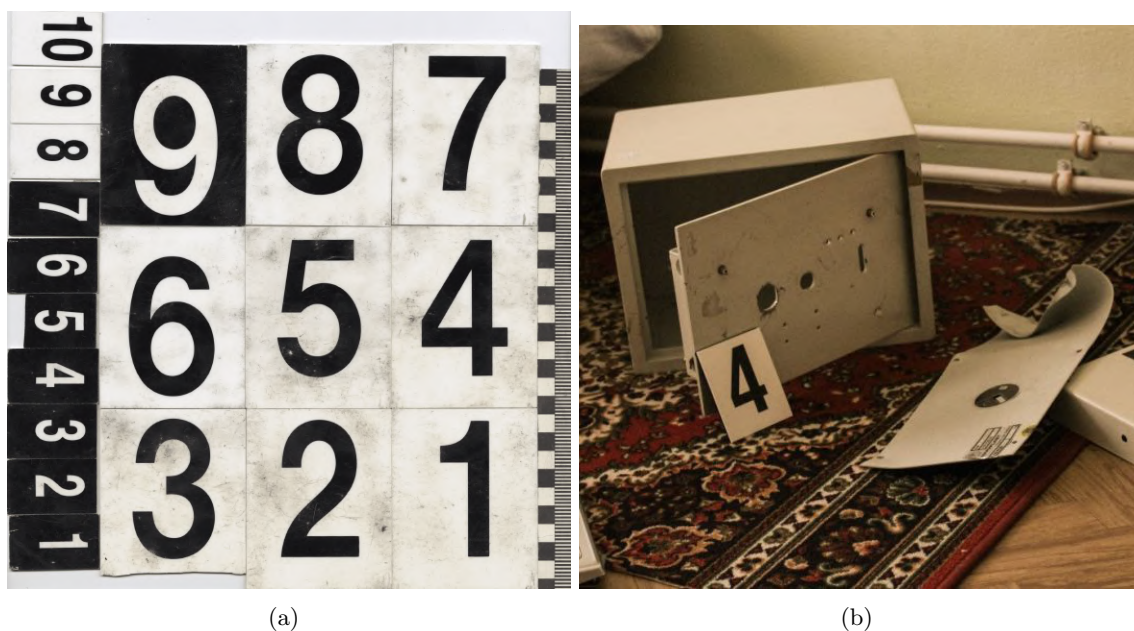
Na levé straně listu se nachází dvě označení pro spojování s dalšími listy a v patičce každého listu se nachází informace o tom, kdo dokumentaci vyhotovil a zpracoval.

Vzhled a rozměry jednotlivých částí listů fotografické dokumentace se podařilo definovat na konzultaci s pracovníky fotografických expertiz OKTE KŘP JmK. Všechny rozměry a požadavky na list dokumentace jsou zdokumentovány v příloze A.11. Na konzultaci bylo také zjištěno, že export a tisk se provádí do jediného formátu, aby se dosáhlo konzistence při vytváření dokumentace. Požadovaný formát výstupu musí mít kvalitu 300DPI<sup>1</sup>. Při tisku se tisknou pouze všechny stránky, na výšku, barevně a pouze jednostranně.

## 2.3 Štítky pro označování stop

Jedná se kriminalistickou techniku používanou při dokumentování zájmového prostředí. Používají se plastové štítky, na kterých je z jedné strany černě zbarvená číslice na bílém pozadí a z druhé strany invertovaně, tedy bíle zbarvená číslice na černém pozadí. Číslice se dělají pouze v rozsahu 0-9, při nutnosti označení 10 a více se štítky skládají k sobě. Po předchozí konzultaci bylo zjištěno, že se zřídka používají i číslice se žlutým pozadím a při označování míst s hromadným neštěstím se mohou použít i číslice zbarvená růžově, zeleně nebo modře, ale ty se zpracovávají jinak.

V rámci této aplikace má být počítačové vidění schopné rozpoznat štítky ve fotografiích a při úspěšně detekovaném štítku by se mělo předvyplnit textové pole u fotografie.



Obrázek 2.2: (a) Vzhled štítků pro označování stop<sup>1</sup> (b) Praktické využití štítků<sup>2</sup>

<sup>1</sup>Zkratka DPI určuje kolik obrazových bodů se vejde do délky jednoho palce.

<sup>1</sup>převzato z předchozí konzultace

<sup>2</sup>převzato z <https://www.gymskutec.cz/o-skole/fotogalerie/ucebny/odborna-ucebna-kriminalistiky-190cs.html>

## 2.4 Desktopová aplikace

Jedná se o aplikace, jenž jsou vyvíjené tak, aby běžely samostatně na stolním nebo přenosném počítači a jsou přímo instalované u uživatele. Instalace aplikace a aktualizace se provádějí na každém systému zvlášť, na rozdíl od webových aplikací, které mají přístupnou aktuální verzi pro všechny uživatele. Velkou výhodou je rychlost a výkon desktopových aplikací, jelikož výpočetní výkon je závislý na systému, kde je aplikace nainstalovaná. U webových aplikací může rychlost práce se systémem ovlivňovat i rychlosti internetového připojení. Pokud má cílový systém malý výpočetní výkon, tak bude složité používat oba typy aplikací. Desktopové aplikace mohou být nezávislé na internetovém připojení a mohou tak být funkční i bez internetu. Díky tomu může uživatel pracovat se starší verzí programu. Bezpečnost desktopových aplikací bývá silnější, hlavně kvůli možnosti odpojit se od internetového připojení a pracovat offline. [11] [17]

V této práci je požadavkem vytvoření desktopové aplikace, která by pracovala s daty na příslušném serveru. Nemělo by být možné pracovat s aplikací bez připojení do sítě, jelikož jsou vyžadována data ze serveru, alespoň do lokální sítě musí být uživatel připojen.

## 2.5 Uživatelské rozhraní

Stará se o interakci mezi člověkem a počítačem, webovou stránkou nebo aplikací. Mezi hlavní funkce uživatelského rozhraní patří přijetí uživatelských vstupů a zobrazování uživateli výstup aplikace. Existuje několik různých typů uživatelského rozhraní, mezi nejznámější patří grafické uživatelské rozhraní, textové uživatelské rozhraní, ale také například uživatelské rozhraní ovládané hlasem nebo gesty. V případě této práce bude navržen grafické uživatelské rozhraní, složené z interaktivních grafických ovládacích prvků, které budou ovládány primárně pomocí klávesnice a myši. Uživatelské rozhraní se primárně zaměřuje na vzhled a dojem z prezentovaného prostředí aplikace nebo webové stránky. Příkladem mohou být webové stránky Airbnb<sup>2</sup> nebo Dropbox<sup>3</sup>, které nabízejí velmi příjemný vzhled uživatelského rozhraní. Uživatelé mají tendenci velmi rychle hodnotit vzhled uživatelského rozhraní a chtějí, aby bylo uživatelské rozhraní příjemné na pohled, tak příjemné na používání. [6][41] O vytvoření uživatelského rozhraní, které by bylo příjemné na používání, se stará uživatelská zkušenost. Účelem uživatelského rozhraní je tedy, jak aplikace vypadá, přičemž uživatelská zkušenost se stará o to, jak se aplikace používá. [24]

V rámci této práce by se podle požadavků Policie ČR mělo docílit jednoduchého používání aplikace s příjemně vypadajícím vzhledem.

---

<sup>2</sup><https://www.airbnb.cz/>

<sup>3</sup><https://www.dropbox.com/>

## 2.6 Frontend a backend

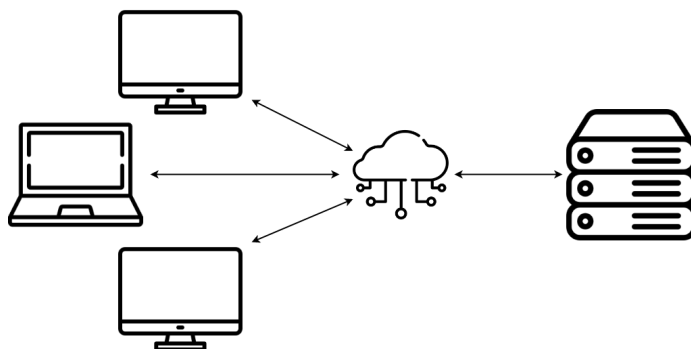
Frontend je část aplikace, která určuje co uživatel vidí a s čím dokáže provádět akce. Mezi to se počítá písmo, barvy, tlačítka, menu, tabulky a další. Snaží se o zjednodušení uživatelské interakce s backendem, která by byla jinak uživatelsky nepřívětivá. Backend je často v aplikaci označovaná jako vrstva přístupující k datům. Obecně není viditelná uživateli a uživatel je odkázán pouze na frontend, pomocí kterého je schopen použít backend. Backend obsahuje většinu programovacích částí systému a stará se jimi, aby vše fungovalo tak, jak má. Snaží se frontendu poskytovat všechny vyžadované zdroje, se kterými chce uživatel pracovat. Frontend i backend musí tedy spolu úspěšně komunikovat, aby se docílilo plně funkčního systému. [16] [34]

## 2.7 Architektura klient-server

Jedná se o výpočetní model, ve kterém existují dva hlavní aktéři, server a klient. Server se stará o poskytování a spravování většiny prostředků, služeb, které požaduje klient. Klienti vytváří požadavky na server, který jim na ně odesílá odpovědi. Typicky je klientů více a jsou umístěni na pracovních stanicích nebo na osobních počítačích, zatímco server je umístěn na výkonnějších strojích a nachází se odděleně od klientů. Klienti zasílají požadavky na server přes síť a i proto je tento model někdy označován jako síťový výpočetní model. Požadavky od klientů je vhodné nějakým způsobem zabezpečit, k tomu lze například využít JSON Web tokeny. Klienti mezi sebou nijak nesdílí informace, které odesílají. [7]

Klient-server architektura poskytuje horizontální škálovatelnost, která zvyšuje počet klientů a vertikální škálovatelnost, která přesouvá procesy na výkonnější servery nebo na více serverů. Škálovatelnost může probíhat separátně jak pro klienty, tak i pro server. Jedná se o centralizovaný systém, který udržuje všechny data a služby na jednom místě. Umožňuje znovuzískání dat nebo vyvážení zátěže na server. Nevýhodou může být zranitelnost na DoS útoky (odepření služby serveru) nebo nakažení serveru. Pokud se na server dostane červ, virus nebo trojský kůň, je pravděpodobné, že budou napadeni i klienti. Příkladem této architektury můžou být emailové servery, datové servery nebo webové servery. [40]

Výsledný systém má být založený právě na architektuře klient-server. Uživatelská aplikace má pracovat pouze se dokumentacemi a jinými daty, které jsou uloženy na serveru.

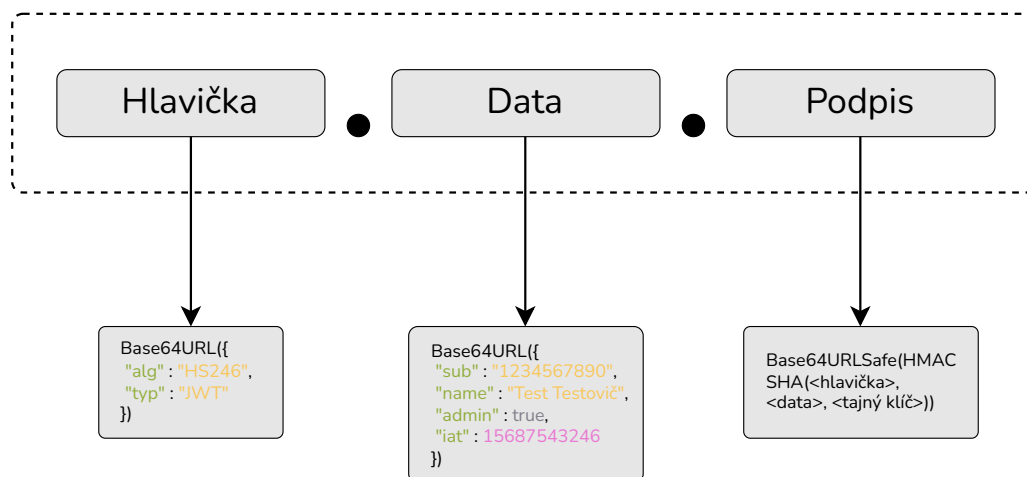


Obrázek 2.3: Klient-server architektura

## 2.8 JSON Web Tokeny

Jedná se o otevřený standard, RFC 7519[21], který definuje kompaktní a samostatný způsob pro bezpečný přenos informací mezi komunikujícími stranami. Využívá objektu JSON a informace v něm lze ověřit a důvěřovat jim, protože jsou digitálně podepsány. Tokeny mohou být podepsány pomocí tajemství, s použitím HMAC algoritmu nebo pomocí párů veřejného a soukromého klíče, s použitím RSA nebo ECDSA šifrování. Podepsané tokeny mohou ověřit integritu nároků v nich obsažených, šifrované tokeny tyto nároky skryjí před ostatními stranami. Pokud jsou tokeny podepsány pomocí párů veřejného a soukromého klíče, tak podpis také potvrzuje, že pouze strana, jenž drží soukromý klíč, jej mohla podepsat.

Struktura tokenu se skládá ze tří částí oddělené tečkami (xxxxx.yyyyy.zzzzz). Obsahuje v pořadí, hlavičku, anglicky Header, data, anglicky Payload a podpis, anglicky Signature. Hlavička se typicky skládá ze dvou částí, těmi jsou typ tokenu a použitý podpisový algoritmus. Před vložením hlavičky do tokenu je zakódována pomocí Base64Url. Data většinou obsahují prohlášení o entitě (typicky uživateli) a další údaje, někdy také označované jako nároky. Tyto nároky se dělí do tří kategorií, registrované, veřejné a soukromé. Registrované nároky obsahují sadu předem definovaných tvrzení, které nejsou povinné, ale doporučované, patří mezi ně například exp, který nese dobu vypršení platnosti. Veřejné nároky jsou libovolně definované uživateli JWT a aby se předešlo kolizím, měly by být definovány v registru webových tokenů IANA JSON nebo definovány jako URI, které obsahuje jmenný prostor odolný proti kolizím. Soukromé nároky obsahují vlastní nároky vytvořené za účelem sdílení informací mezi stranami, které se dohodly na jejich použití a nejedná se o registrované či veřejné nároky. Přenášena data jsou před vložením do tokenu zakódována pomocí Base64Url. K vytvoření podpisové části je potřeba zakódovaná hlavička, zakódovaná přenášena data, tajemství a algoritmus specifikovaný v hlavičce. Pomocí specifikovaného algoritmu se vytvoří podpis, který slouží hlavně pro ověření, že zpráva nebyla po cestě modifikovaná. V případě soukromého klíče, může také ověřit, že odesílatel JWT je opravdu, ten který má být. [3]



Obrázek 2.4: Struktura JSON Web Tokenu

V tomto projektu budou JSON Web tokeny použity hlavně k autentizaci uživatele a následně k autorizaci při přístupu k požadavkům serveru. Po úspěšném přihlášení obdrží

uživatel svůj token, který bude používat pro pohyb v aplikaci, každý požadavek vyžadující autorizaci bude obsahovat tento token, aby bylo možné uživatele autorizovat.

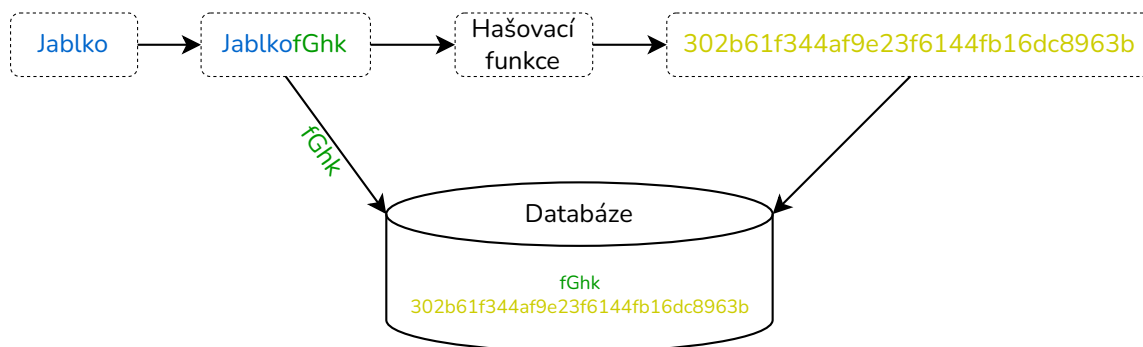
## 2.9 Zpracování hesel

Hesla by se neměla nikdy ukládat v otevřené formě, jelikož to přináší větší rizika než výhody. Mezi výhody můžeme zařadit dohledání a odeslání hesla uživateli, pokud uživatel zapomene heslo. Avšak musíme hlídat, kdo má přístup do databáze, protože vzniká riziko zneužití osobou s přístupem do databáze. Při úniku dat z databáze, útočník získá neomezený přístup k uživatelským účtům a heslům. Při takových zneužitích nemusí být ohrožen pouze přístup do aktuální aplikace, ale také do jiných služeb, jelikož lidé mohou používat stejná hesla ve více službách.

Pro ukládání a ověřování hesel byla zvolena technika „salting“ a hašování hašovací funkcí. Hašovací funkce jsou takové, které mají následující vlastnosti:

- Hašovací funkce musí vždy pro stejný vstup vrátit stejný výsledek.
- Hašovací funkce musí pro libovolně dlouhý vstup vždy vrátit výsledek o pevné délce.
- I sebemenší změna ve vstupu hašovací funkce vytvoří velkou změnu ve výsledku.
- Odolnost vůči nalezení vzor: Pro známý výsledek hašovací funkce je složité nalézt vstup hašovací funkce.
- Odolnost vůči nalezení druhého vzoru: Pokud je znám vstup hašovací funkce a následně i výsledek hašovací funkce, je složité nalézt druhý vstup hašovací funkce, jenž by měl stejný výsledek hašovací funkce.
- Odolnost proti kolizi: Je složité nalézt dva vstupy hašovací funkce, které by měly stejný výsledek hašovací funkce.

Jelikož mnoho lidí stále používá velmi běžná hesla a výsledek hašovací funkce bude pro daný vstup vždy stejný. Útočník může vytvořit hashe těchto běžných hesel a následně testovat uniklá data z databáze proti těmto předem vytvořeným hashům. Ve skutečnosti už existují celé databáze takto vytvořených hashů, nazývající se anglicky rainbow tables. Jednou z technik řešící tento problém, je takzvaný „salting“. Jedná se o generování krátkého náhodného textu, jenž bude přidán k heslu, ještě před samotným hašováním. Toto generování textu se provádí pro každého uživatele zvlášť a generovaný text je uložen v databázi. Při přihlašování se heslo ověří, tak že se načte vstup od uživatele, načte se salt z databáze, ke vstupu od uživatele se přidá salt, vytvoří se hash a porovná se s uloženým hashem v databázi. Pokud se hashe rovnají, uživatel zadal správné heslo a může přistoupit do systému. [38]



Obrázek 2.5: Postup Saltingu

## 2.10 REST API

REST je zkratka Representational State Transfer a jedná se o architektonický styl, jenž má své hlavní zásady a omezení. API, neboli rozhraní pro programování aplikací, je sada pravidel definující, jak se mohou aplikace nebo zařízení vzájemně propojit a komunikovat spolu. Aby bylo API možné nazývat REST, musí splňovat určité principy [30] [33]:

**Jednotné rozhraní** Může být získáno splněním vícero architektonických omezení, jedním z omezení je identifikace zdrojů, které znamená, že rozhraní musí jednoznačně identifikovat každý zdroj zapojený do interakce mezi klientem a serverem. Dalším omezením může být manipulace se zdroji pomocí reprezentací, které je definováno tak, že zdroje by měly mít jednotnou reprezentaci v odpovědi serveru, aby jej mohli uživatelé API používat k úpravě prostředků na serveru.

**Klient-server** Návrhový vzor umožňující samostatný vývoj jak klientské aplikace, tak serverové části. Zlepší se tím přenositelnost uživatelského rozhraní mezi více platformami a zvýší se škálovatelnost serverové části. Více o klient-server 2.7. Při vývoji těchto částí nezávisle, musí být rozhraní mezi částmi neporušené.

**Bezstavovost** Každý požadavek od klienta nese všechny potřebné informace k pochopení a zpracování požadavku. Serverová část nesmí využít informací z předchozích požadavků a klientská aplikace musí udržovat stav relace.

**Mezipaměť** Každá odpověď musí být implicitně nebo explicitně označena jako uložitelná nebo neuložitelná do mezipaměti. Pokud je odpověď uložitelná do mezipaměti, klientská aplikace může po určitou dobu využít data z odpovědi pro další stejné požadavky.

**Vrstvený systém** Žádná komponenta nevidí za bezprostřední vrstvu, se kterou pracuje.

**Kód na vyžádání** Volitelný princip, umožňující poskytnout klientovi funkcionalitu dostupnou na serveru. Klientská aplikace tuto funkcionalitu neimplementuje a na vyžádání si ji stáhne ze servera a spustí.

Zdroj v REST je abstrakce pro informaci, pokud dokážeme nějakou informaci pojmenovat, tak ji můžeme označovat jako zdroj. Zdrojem může tedy být obrázek, soubor, dokument, služba, osoba a další. Pro manipulaci s těmito zdroji se většinou používají HTTP metody,

kterými jsou GET, PUT, POST, DELETE. Význam operací spojených s metodami by měl být stejný napříč celým systémem, aby se udržel princip jednotného rozhraní. [32]

Výsledný systém by měl využívat REST API pro komunikaci mezi klientem a serverem.

## 2.11 Zabezpečení komunikace

Při přenosu citlivých dat mezi klientem a serverem je vhodné přenášet data a připojení zabezpečit. Základní komunikační protokol HTTP, definovaný v RFC 2616 [27], neumožňuje šifrování ani zabezpečení integrity dat. Pro zabezpečení HTTP se používají SSL/TLS protokoly, které zajišťují větší bezpečnost a důvěrnost dat uživatelů. Po zabezpečení komunikačního protokolu HTTP, je protokol označován jako HTTPS.

SSL je zkratka pro Secure Sockets Layer a jedná se o protokol, jenž vloží další vrstvu mezi transportní a aplikační vrstvu, která poskytuje zabezpečení komunikace šifrováním a umožní autentizaci komunikujících stran. Tím brání útočnickům ve čtení a úpravě přenášených informací. SSL může být implementováno pouze pokud má server SSL certifikát, tento certifikát je veřejně dostupný klientům. V certifikátu se nachází veřejný klíč, který použije pro navázání bezpečných šifrovacích klíčů se serverem. Server si udržuje svůj soukromý klíč, který používá pro rozšifrování dat, které klient zašifroval veřejným klíčem. SSL je přímým předchůdcem protokolu TLS, který je definován v RFC 8446 [31]. TLS je pouze aktualizovaná, bezpečnější verze SSL. SSL je již zastaralý a nepoužívá se, avšak bývá někdy zmiňován jako použitý protokol, ale ve skutečnosti je používán nový protokol TLS.[12]

V tomto projektu bude komunikace mezi klienty a serverem zabezpečena právě pomocí HTTPS s využitím TLS certifikátu.

## 2.12 Počítačové vidění a detekce objektů

Počítačové vidění je obor umělé inteligence, který umožňuje počítačům odvozovat smysluplné informace z obrázků, videí a jiných vizuálních vstupů. Umělá inteligence je vědní obor zabývající se vytvářením strojů a počítačů, jenž mohou uvažovat, učit se a jednat způsobem, který by vyžadoval lidskou inteligenci. Počítačové vidění umožňuje počítačům vidět, sledovat a chápat.

Aby počítač docílil počítačového vidění, zpracovává velké množství dat, na kterých provádí mnoho analýz, dokud není schopný rozeznat rozdíly. Využívá k tomu typ strojového učení nazývaný hluboké učení a konvoluční neuronové sítě. Strojové učení využívá algoritmičké modely, které umožňují počítači, aby se sám učil o kontextu vizuálních dat. Konvoluční neuronové sítě pomáhají hlubokému učení rozdělením obrázku na části, které mají definováno co se na nich nachází pomocí štítků. Neuronová síť provádí konvoluce a v jednotlivých iteracích kontroluje svou přesnost předpovědí. Pro pochopení jednotlivých obrázků se používají konvoluční neuronové sítě, u kterých nejprve rozeznává ostré hrany a jednoduché tvary, následně doplňuje informace při provádění iterací. Pro pochopení, jak spolu obrázky souvisí v rámci série snímků se používají rekurentní neuronové sítě. [14]

Detekce objektů se následně provádí tak, že se na vstup přijímá obrázek, na kterém se má provést rozpoznání objektů, a výstupem jsou pravděpodobnosti jednotlivých naučených objektů. Pokud se daný objekt nachází v obrázku bude pravděpodobnost vysoká. Detekce objektů často vystupuje jako kombinace klasifikace obrázku a lokalizace objektu, kde klasifikace obrázku bere jako vstup obrázek a vrací výstup obsahující klasifikační metriky, jako jsou pravděpodobnost, ztráta, přesnost apod. Zatímco lokalizace objektu hledá



v obrázku objekt a ten označí pomocí souřadnic a hran na místě, kde si myslí, že se objekt nachází. Detekce objektů tedy přijímá obrázek a vrací jeden nebo více ohraničujících boxů s výsledkem klasifikace. [29] [13]

Použití konvolučních neuronových sítí se řadí mezi pokročilejší metody detekce objektů, které jsou schopné dosáhnout vyšší přesnosti detekce a lépe zvládají různé změny v obrázku (rotace, deformace, změny světelných podmínek). Tradiční metody většinou nevyužívají neuronové sítě a zaměřují se na extrakci vlastností, ze kterých následně provádějí rozpoznávání. Mezi nejznámější metody detekce objektů patří metoda Viola-Jones, která k detekci využívá Haarovské příznaky a kaskádový klasifikátor. Metoda je využívána hlavně k detekci obličejů. Nevýhodou je citlivost na velké změny expozice a metoda je omezena na binární klasifikaci. [1] Mezi další tradiční metody detekce objektů lze zařadit metodu HOG, histogram orientovaných gradientů. Jedná se o jednoduchou a efektivní metodu pro detekci objektů, avšak přesnost detekcí není tak vysoká jako u metod s konvolučními neuronovými sítěmi a při zpracovávání obrázků s vysokým rozlišením je výpočetní rychlost velmi malá. [23]

V tomto projektu bude využita detekce objektů v obrázku pro rozpoznávání štítků s kriminalistickým číslem ve fotografiích. Výsledek detekce bude použit pro předvyplnění textového pole u fotografie. Detekce by měla využívat model YOLOv7<sup>4</sup>, který využívá konvoluční neuronové sítě a překonává všechny známé detektory objektů jak rychlostí, tak přesností detekcí. Díky tomu se stal velmi oblíbeným modelem pro detekci objektů. [4] Podrobnosti modelu jsou dostupné v dokumentu [43]. K detekci bude nutné vytvořit data set z vlastních fotografií, které budou obsahovat kriminalistické štítky s anotacemi. Model následně využije data set k natrénování detekce a po natrénování bude nutné převést model do formátu, který bude klientská aplikace schopna zpracovat a použít.

---

<sup>4</sup><https://github.com/WongKinYiu/yolov7>

## Kapitola 3

# Návrh aplikace

Úkolem je navrhnout desktop aplikaci, jenž by splňovala požadavky od Policie ČR a zároveň obsahovala frontend složený z funkčního uživatelského rozhraní tak, aby byla práce se systémem co nejvíce uživatelsky přívětivá. Pro backend by měla být navržena databáze, jenž by obsahovala všechny vyžadované zdroje aplikace. Pro systém by měly být definováni aktéři, jenž budou systém při nasazení využívat, i s jejich funkcemi v systému. Návrh by měl obsahovat všechny technologie, které bude systém zahrnovat a využívat. Měly by být navrženy všechny obrazovky, ke kterým se uživatel může dostat a za jakých podmínek.

### 3.1 Aktéři systému

Systém obsahuje několik aktérů:

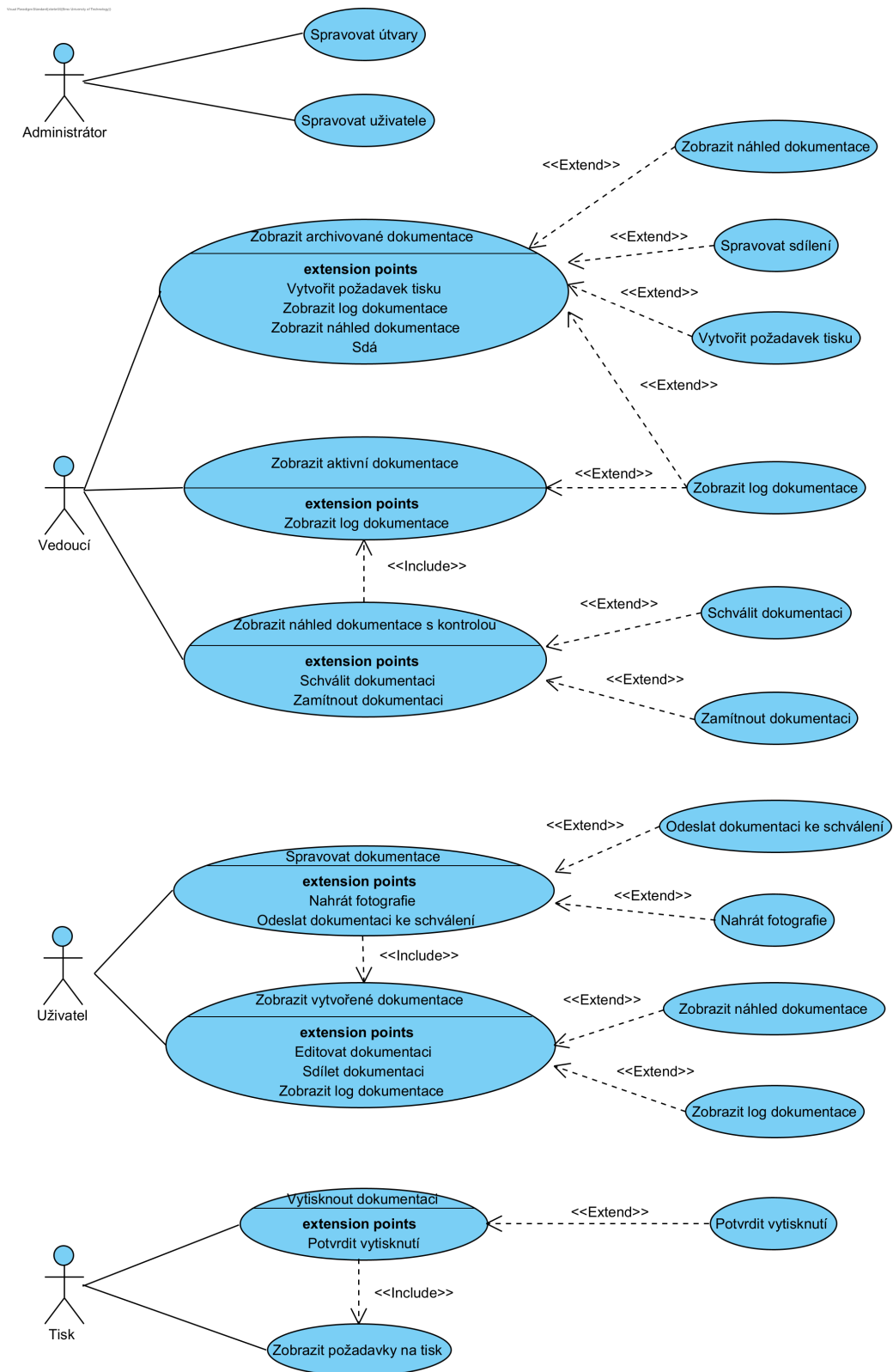
**Administrátor** - Disponuje schopností spravovat uživatele a útvary systému. Dokáže vytvářet, editovat uživatele a útvary.

**Uživatel** - Pracuje právě pod jedním útvarem, který může mít více vedoucích. Má možnost vytvářet dokumentace, které následně musí odeslat na schválení vedoucími. Se schvalováním souvisí také možnost opravovat zamítnuté dokumentace. Po přihlášení vidí všechny aktivní dokumentace, na kterých se podílí. U dokumentací může zobrazovat záznam změn dokumentace.

**Vedoucí** - Jedná se o uživatele, jenž má schopnost schvalovat fotografické dokumentace, vytvořené uživateli. Může přistoupit do archivu dokumentací, kde vidí schválené dokumentace, může vytvářet požadavky na tisk, sdílet dokumentace jiným uživatelům či útvarům. Po přihlášení vidí všechny aktivní dokumentace, které čekají na schválení, u těchto dokumentací může zobrazit náhled dokumentace s kontrolou.

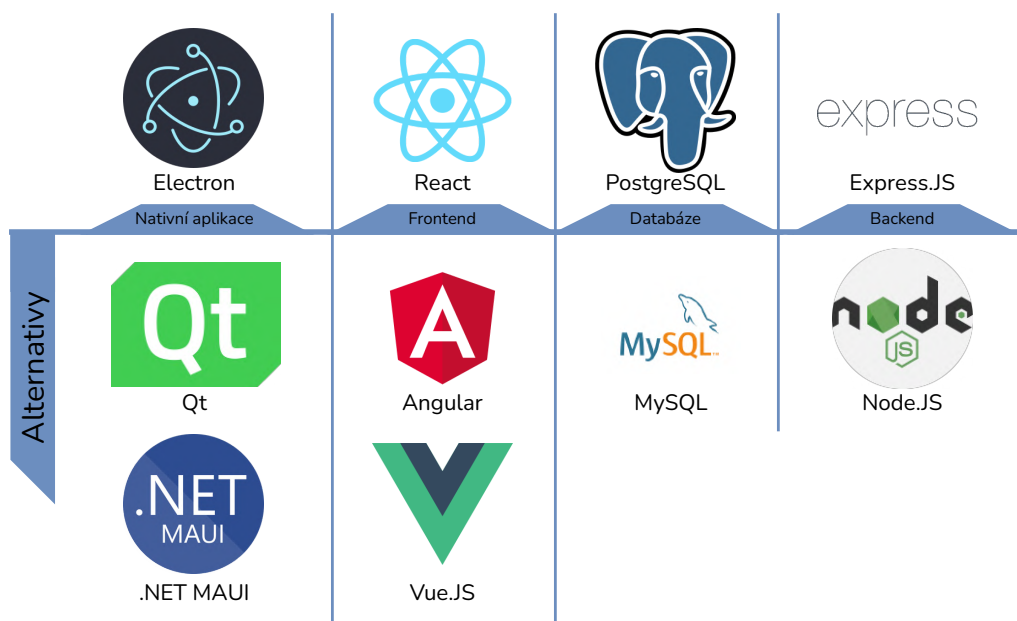
**Tisk** - Uživatel, jenž má přístup k tiskárnám, aby mohl tisknout požadavky na tisk od vedoucích. Po přihlášení vidí všechny aktivní požadavky k tisku. Po úspěšném vytisknutí a předání žádajícímu je tiskař povinen potvrdit tuto skutečnost v systému.

K těmto aktérům je definován diagram případů užití a podrobné specifikace případů užití, které se nacházejí v souboru `specifikace_use_case.pdf` na odevzdaném médiu.



Obrázek 3.1: Diagram případů užití

## 3.2 Navrhované technologie



Obrázek 3.2: Navrhované technologie

### 3.2.1 Tvorba nativních aplikací

#### Electron<sup>1</sup>

Jedná se o framework pro tvorbu nativních aplikací pro více platform. K vývoji využívá webové technologie, jako jsou JavaScript, HTML a CSS. Díky zabalení Node.js a Chromium přímo do spustitelného souboru, dokáže poskytovat aplikace pro více platform bez jakékoliv znalosti vývoje na těchto platformách. I kvůli těmto vlastnostem využívá Electronu mnoho známých aplikací, například Visual Studio Code, Facebook Messenger, Twitch, Discord, Microsoft Teams a další.

Mezi pozitiva Electronu patří určitě obrovská komunita vývojářů, takže při dohledávání nějakých odpovědí na otázky by neměl být problém. Dalším pozitivem může být nízká křivka učení, to ve zkratce znamená, že Electron není nijak složitý na naučení. Pro někoho a i pro mě může být velkým pozitivem, že používá technologie pro vývoj webu, které mi jsou celkem blízké. Mezi negativa můžeme zařadit větší velikost instalace a to hlavně kvůli přibalování Chromium a Node.JS. Dalším negativem může být samotný Chromium, který bývá náročnější na RAM a CPU. [2]

Pomocí hlavního souboru, definovaného v package.json, jsou vytvářena okna aplikace. S každým oknem je vytvářen proces, nazvaný jako „renderer“, tento proces se stará o zobrazování HTML, JavaScript souborů, obrázků a dalších v přiděleném okně. Hlavní proces spravuje všechna vytvářená okna a jejich procesy. Jen hlavní proces má přístup k interakcím s grafickým uživatelským prostředím, dokáže přistoupit ke všem NodeJS modulům, jako je například práce se souborovým systémem nebo vytváření HTTP požadavků.

<sup>1</sup>Logo v přehledu 3.2 převzato z <https://uxwing.com/electron-icon/>

Aby mohly mezi sebou procesy komunikovat musí být zavedena meziprocesorová komunikace. Proto Electron nabízí dva moduly nazvané jako „ipcMain“ a „ipcRenderer“. Modul ipcMain je použit pro asynchronní komunikaci z hlavního procesu procesům starajících se o renderování. Pokud je použit přímo v hlavním procesu, stará se o přicházející zprávy do hlavního procesu. Modul ipcRenderer je použit pro asynchronní komunikaci z procesů starajících se o renderování hlavnímu procesu. Zároveň je schopen přijímat odpovědi od hlavního procesu. [15]

## Alternativy

**Qt**<sup>1</sup> Je framework pro tvorbu aplikací pro více platforem obsahující komplexní sadu vysoce intuitivních a modulárních C++ knihoven. Poskytuje API pro Python a Javascript pro práci s těmito knihovnamy. Qt nabízí velmi čitelný, znovupoužitelný a jednoduše udržitelný kód s velkým běhovým výkonem. Příklady aplikací, využívající právě Qt, jsou například VLC media player, VirtualBox, Wireshark.

Mezi pozitiva můžeme zařadit například vyspělost frameworku, i díky tomu jej využívá mnoho velkých společností. Další pozitivem, navazující na vyspělost frameworku, je výborná dokumentace a pokud selže i dokumentace, velký počet uživatelů může zodpovědět jakékoliv otázky. Mezi negativa můžeme zařadit fakt, že Qt používá některé části, které nejsou součástí jazyku C++, jenž musí být překonvertovány speciální kompilací pomocí „Meta-Object Compiler“, to se musí provést před skutečnou kompilací aplikace. [42]

**.NET MAUI**<sup>2</sup> Framework poskytující tvorbu aplikací pro více platforem. Využívá jádra .NET 6, jazyku C# a vychází z projektu Xamarin, který už dříve umožňoval vývoj multiplatformních aplikací, avšak nebyl příliš oblíbený. Jelikož se jedná o celkem čerstvý počín, tak nemá moc příkladných aplikací. Framework byl představen na podcastové aplikaci .NET Podcasts a aplikaci s počasím WeatherTwentyOne.

Mezi pozitiva .NET MAUI patří použití jedné báze zdrojového kódu pro aplikace na různé platformy. Dalším pozitivem může být „Hot-reload“, který umožňuje rychle zobrazovat změny ve zdrojovém kódu. Mezi negativa patří například, že nepodporuje aplikace pro platformu Linux. Dalším negativem může být také, že se jedná o nový framework a tak může být složité hledat odpovědi na otázky u uživatelů používající .NET MAUI. [8]

### 3.2.2 Frontend

#### React<sup>3</sup>

Je open-source framework pro tvorbu uživatelského rozhraní založený na JavaScriptu. Zaměřuje se pouze na pohledovou vrstvu aplikace, jenž prezentuje data uživateli a na rozhraní, ve kterých se rychle mění data. Stejně jako Angular, nabízí techniku jednostránkových aplikací. Využívá komponenty a deklarativní vlastnosti, díky kterým dovoluje vývojářům vytvářet interaktivní a komplexní UI. React využívají aplikace jako jsou například Dropbox, Facebook, Instagram, Netflix, Airbnb a jiné.[19]

<sup>1</sup>Logo v přehledu 3.2 převzato z <https://brand.qt.io/design#logos>

<sup>2</sup>Logo v přehledu 3.2 převzato z [https://styles.redditmedia.com/t5\\_2odyx7/styles/communityIcon\\_19sk0x18irz41.png](https://styles.redditmedia.com/t5_2odyx7/styles/communityIcon_19sk0x18irz41.png)

<sup>3</sup>Logo v přehledu 3.2 převzato z <https://w7.pngwing.com/pngs/79/518/png-transparent-js-react-js-logo-react-react-native-logos-icon-thumbnail.png>

Využívá souborů JSX, JavaScript syntactic extension, do kterých je možné psát JavaScript kód a zároveň HTML struktury.

```
const name = 'React';  
const greet = <h1>Hello, {name}</h1>;
```

HTML dokumenty se mohou reprezentovat jako stromová struktura, kde každý objekt reprezentuje jednu část dokumentu, tato stromová struktura se nazývá DOM, objektový model dokumentu. React tvoří virtuální DOM, který je vlastně odlehčená verze reálného DOMu a to z důvodu, že manipulace s reálným DOMem je výrazně pomalejší než s virtuálním. Pokud se změní stav objektu, virtuální DOM se aktualizuje, následně se provede porovnání mezi virtuálním a reálným DOMem, pokud se nalezne rozdíl, je v reálném DOMu aktualizován pouze nalezený objekt.

React rozděluje uživatelské rozhraní do komponent, kde každá komponenta má vlastní atributy a funkce. Tímto rozdělením udělá ladění kódů jednodušší, umožní znovupoužití, které zrychlí vývoj celé aplikace. Komponenty lze vkládat do sebe, musí definovat metodu renderování v DOMu a může definovat atributy, které může rodičovský komponent využít pro specifikování hodnot komponenty. [9]

### Alternativy

**Angular**<sup>1</sup> Frontend framework, založený na komponentách, postavený na TypeScriptu, určený pro tvorbu škálovatelných aplikací. Nabízí techniku jednostránkových aplikací, ve kterých se mění pouze obsah. Nabízí kolekci integrovaných knihoven a funkcí, jako jsou například klient-server komunikace, směrování a další. Jedná se o konstantně aktualizovanou technologii vedenou společností Google. Aplikace, jež využívají Angular framework, jsou například PayPal, Google, HBO, Forbes a jiné. [10]

**Vue.js**<sup>2</sup> Patří do trojice, React, Angular, Vue.js, v současnosti nejoblíbenějších a nejvyužívanějších JavaScript frameworků. Nabízí deklarativní a na komponentách založený programovací model. Snaží se poskytovat většinu základních funkcí, které by mohly být u frontend vývoji potřeba. Nabízí stejně jako Angular a React, techniku jednostránkových aplikací. Na rozdíl od Reactu, kde se celý frontend píše v JSX, tak Vue.js využívá normálního HTML, které následně ožíví. Mezi aplikace, které využívají Vue.js, patří například Grammarly, Adobe Portfoliom, Trivago, GitLab a další. [22]

### 3.2.3 Databáze

#### PostgreSQL<sup>3</sup>

Je open-source výkonný databázový systém využívající jazyk SQL s rozšířeními pro bezpečné ukládání a škálování komplikovaných pracovních zatížení daty. Získala si silnou reputaci díky své osvědčené architektuře, spolehlivosti, integritě dat, robustní sadě funkcí a rozšiřitelnosti. Podporuje všechny hlavní operační systémy a plně podporuje ACID už od roku 2001.

<sup>1</sup>Logo v přehledu 3.2 převzato z <https://angular.io/presskit>

<sup>2</sup>Logo v přehledu 3.2 převzato z <https://vue.js.org/images/logo.png>

<sup>3</sup>Logo v přehledu 3.2 převzato z <https://download.logo.wine/logo/PostgreSQL/PostgreSQL-Logo.wine.png>

Mezi pozitiva lze zařadit podporu velkých systému a s tím související zápis velkého množství dat, avšak strádá v rychlosti čtení. Dalším pozitivem může být podpora JSON a ostatních NoSQL funkcí, jako je například XML. Podporuje jak materializované pohledy, tak dočasné tabulky. [20][25]

### Alternativa

**MySQL**<sup>1</sup> Je světově nejpopulárnější databáze a to i díky tomu, že se jedná o databázi, která je jednoduchá na použití a je optimalizovaná na rychlost. Je navrhnutá hlavně pro práci se systémy s velkým zatížením. Podporuje mnoho operačních systémů a je částečně kompatibilní s jazykem SQL. Stejně jako PostgreSQL, drží se modelu ACID, ale to jen za předpokladu, že jsou použity formáty úložiště jako InnoDB a NDB Cluster. Vyniká v rychlosti čtení velkého množství dat, kde PostgreSQL strádá. Podporuje datový typ JSON, avšak nepodporuje jiné NoSQL funkce.[18] [20]

### 3.2.4 Backend

#### Express<sup>2</sup>

Jedná se o minimalistický a robustní framework pro NodeJS, který nabízí kolekci funkcí pro vývoj webových aplikací. Je postavený nad NodeJS a snaží se hlavně zjednodušit a pomáhat při správě serveru a jeho požadavků. Dovoluje nastavit prostředníky, kteří se starají o zpracování HTTP požadavků od klienta. Definuje tabulku cest k požadavkům, které provádějí různé akce podle HTTP metody a URL. [36][37]

### Alternativa

**Node.JS**<sup>3</sup> Node.JS je běhové prostředí pro spouštění JavaScript kódu mimo prohlížeč. Nejedná se o framework ani programovací jazyk. Umožňuje vývoj aplikací na straně serveru díky svým rozsáhlým a všestranným funkcím. Mnoho známých společností, jako jsou například PayPal, Netflix, Twitter, LinkedIn a jiné, používají Node.JS pro svůj backend. Je postaven na modelu řízený asynchronními událostmi, díky čemuž lze nad ním vyvíjet škálovatelné síťové aplikace, to umocní ještě Google Chrome V8 engine, který poskytuje Node.JS rychlost a efektivitu.

Mezi pozitiva patří obrovská knihovna JavaScript modulů, které mohou rozšířit funkcionalitu Node.JS. Dalším pozitivem může být jednoduché lazení kódu, díky Google Chrome V8 engine. Mezi negativa můžeme zařadit používání asynchronního programovacího modelu, který může být pro některé vývojáře složitý. [28][37]

---

<sup>1</sup>Logo v přehledu 3.2 převzato z <https://1000logos.net/wp-content/uploads/2020/08/MySQL-Logo.png>

<sup>2</sup>Logo v přehledu 3.2 převzato z <https://www.pngwing.com/en/free-png-iuofv>

<sup>3</sup>Logo v přehledu 3.2 převzato z [https://cdnlogo.com/logo/nodejs\\_42584.html](https://cdnlogo.com/logo/nodejs_42584.html)

### 3.3 Použité knihovny

#### PDF.js

Je knihovna umožňující zobrazovat PDF dokumenty. Je postavena na HTML5, řízena komunitou a podporovaná společností Mozilla. Jejich cílem je vytvořit univerzální platformu pro zpracování a zobrazování PDF dokumentů založenou na webových standardech. V tomto projektu by měla být knihovna použita pro vytváření PDF a následného zobrazování PDF pro náhled dokumentace. [26]

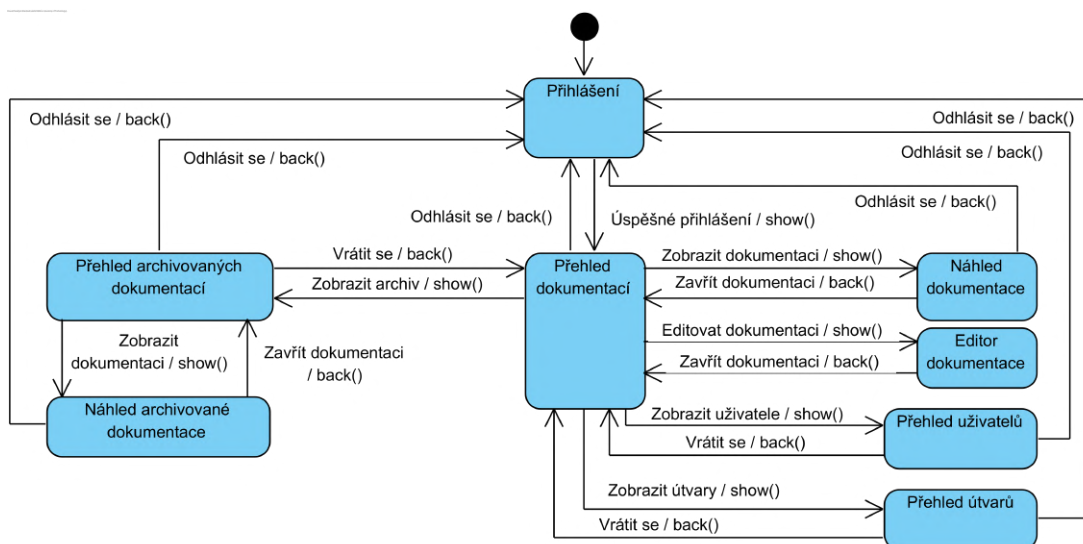
#### TensorFlow.js

Open-source hardwarově akcelerovaná Javascript knihovna pro učení a nasazování modelů strojového učení ve webovém prohlížeči nebo pomocí Node.js. Poskytuje flexibilní a intuitivní rozhraní k vytváření nových modelů, dovoluje spustit již vytvořené modely TensorFlow nebo dokáže přetrénovat existující model pomocí dat na straně klienta. V tomto projektu by měla být knihovna použita pro načtení vytvořeného modelu TensorFlow a k provedení detekce kriminalistických štítků ve fotografiích. [39]

#### CamanJS

Jedná se o knihovnu pro manipulaci s obrázky pomocí HTML5 a Javascript. Používá pokročilé a efektivní techniky pro úpravu obrázků. Funkcionalita knihovna může být rozšířena pomocí dodatečných rozšíření. Samotná knihovna je nezávislá na jiných knihovnách a funguje za použití jen NodeJS. V projektu by měla být použita hlavně kvůli editoru, ve kterém se dají vkládat uživatelem vytvořené fotografie. Pomocí této knihovny by měl být uživatel schopen udělat jednoduché grafické korekce fotografie. [5]

### 3.4 Obrazovky



Obrázek 3.3: Stavový diagram návaznosti obrazovek



Stavový diagram návaznosti obrazovek zobrazuje všechny možné obrazovky, ke kterým se dokáže uživatel v aplikaci dostat. Hlavní obrazovkou je přehled dokumentací, do které se dostane každý uživatel hned po přihlášení. Uživatel může v přehledu vybrat dokumentaci nebo vytvořit novou a tak se dostat na obrazovku s editorem dokumentace. Vedoucí se může, výběrem dokumentace v přehledu, dostat na obrazovku s náhledem dokumentace. Tiskový personál může v přehledu dokumentací vidět tiskové požadavky. Pomocí navigace v horní části obrazovky se mohou uživatelé dostat na přehled archivovaných dokumentací, ze kterého se mohou dostat na náhled archivovaných dokumentací. Administrátor se může pomocí navigace v horní části obrazovky dostat na přehled uživatelů a na přehled útvarů.

## Přihlašování

Aby uživatel mohl začít pracovat s aplikací musí se nejprve přihlásit, bez přihlášení by nemohl být identifikován a autorizován k provádění akcí v systému. K přihlášení se využívá přidělený login a nastavené heslo. **B.1** Při dalším provádění akcí není nutné zadávat login ani heslo, uživatel bude interně ověřen pomocí JWT.

Pokud uživatel zapomene heslo, bude muset kontaktovat administrátora, ten mu nastaví nové dočasné heslo. **B.3** Při dalším přihlášení uživatele do systému, je vyzván ke změně hesla, bez změny hesla nelze pokračovat dále do aplikace. **B.4**

Pokud uživatel zadá špatné heslo nebo není dostupný ověřovací server, nad tlačítkem „Přihlásit se“ se zobrazí informační zpráva o aktuální chybě. **B.2**

## Přehled dokumentací

Po úspěšném přihlášení se uživatelé dostanou na přehled aktivních dokumentací. **B.5** To znamená, že uvidí dokumentace, které jsou rozpracované, čekající na schválení, zamítnuté od vedoucího. Úspěšně schválené dokumentace se archivují, z archivu může vedoucí vytvořit požadavek na tisk a tiskaři dokumentaci vytisknou a předají.

Jednotlivé dokumentace u sebe uvádějí číslo jednacích, evidenční číslo nebo číslo výjezdu, podle toho, které je specifikované, informace o uživatelích, kteří dokumentaci vyhotovily, zpracovaly a schválily. Dále má každá dokumentace u sebe uveden stav, v jakém se aktuálně nachází. Je možné si u každé dokumentace zobrazit záznam provedených akcí, například vytvoření, zamítnutí, schválení. **B.6** Dokumentace lze zobrazovat v náhledu, otevírat editor dokumentací a zobrazovat záznam akcí. V celém přehledu dokumentací lze vyhledávat podle všech zobrazovaných údajů.

Pokud nastane jakákoliv chyba, uživatel je na to upozorněn pomocí upozornění s vypanou chybou.

K této obrazovce se mohou dostat všichni uživatelé systému. Uživatel zde uvidí své dokumentace. Vedoucí zde uvidí dokumentace, které čekají na jeho schválení. Tiskový personál zde uvidí dokumentace, které čekají na vytisknutí. Administrátor zde uvidí všechny dokumentace, které ještě nebyly archivované.

## Archiv dokumentace

Po úspěšném schválení dokumentace vedoucím se dokumentace přesune do archivu dokumentací a je jí nastaven stav „Archivováno“. V tomto přehledu je umožněno dokumentace zobrazovat, sdílet, dodatečně tisknout, stahovat PDF dokumentace a zobrazovat záznam provedených akcí s dokumentací. **B.11** V tomto stavu již nelze dokumentace upravovat.

U každé dokumentace je zobrazováno číslo jednací, evidenční číslo nebo číslo výjezdu, kdo dokumentaci vyhotovil, kdo dokumentaci zpracoval, kdo dokumentaci schválil. Dále je u každé dokumentace uváděno datum vytvoření, datum schválení a aktuální stav. V přehledu archivovaných dokumentace lze vyhledávat podle všech uváděných údajů.

K této obrazovce se mohou dostat pouze vedoucí a administrátor.

## Editor dokumentace

Umožňuje uživateli vytvářet nebo upravovat dokumentace. K této obrazovce má přístup pouze běžný uživatel. **B.19**

V levé horní části obrazovky se nachází informace o případu, které musí uživatel vyplnit, aby mohl dokumentaci uložit. **B.22** Musí vyplnit jednací číslo, evidenční číslo nebo číslo výjezdu, kdo dokumentaci vyhotovil. Podle přihlášeného uživatele se předvyplní zpracovatel dokumentace.

Pod informacemi lze vybírat a přesouvat předlohy, které tvoří dokumentaci a do kterých lze následně přesouvat fotografie. Předlohy se dělí do tří kategorií. Předloha umožňující vložit pouze fotografii, předloha umožňující vložit fotografii a vedle ní popis a posledním typem předlohy je pouze vložení textu. Všechny možné varianty předloh jsou definovány od Policie ČR.

Ve spodní části je přehled listů právě vytvářené dokumentace. Listy lze přidávat, mazat nebo měnit jejich pořadí. Pomocí přehledu lze přepínat listy, které se zobrazí na editovacím plátně. U aktuálně zobrazeného listu se zobrazí tlačítka pro přesun v rámci dokumentace, přidání listu před nebo za aktuální, odstranění listu a pořadí listu v dokumentaci. **B.20**

Na pravé straně obrazovky je prohlížeč fotografií nahraných k dokumentaci. Uživatel musí fotografie nejdříve nahrát k dokumentaci než je bude moct použít v dokumentaci. Po nahrání fotografií k dokumentaci lze fotografie přetahovat na dané předlohy. U jednotlivých fotografií se nachází zaškrtačací tlačítko, pomocí kterého lze vybírat fotografie a následně nad těmito fotografiemi provést automatické vložení. Uživatel musí mít zvolenou předlohu kompatibilní s fotografiemi, vybrané fotografie a musí kliknout na tlačítko „Automatické vložení“. **B.21** Tlačítkem na fotografiemi lze dodatečně nahrávat fotografie. Mazání fotografií může uživatel provést označením fotografie zaškrtačacím tlačítkem a kliknutím na tlačítko nad fotografiemi.

Uprostřed obrazovky se nachází editovací plátno, které zobrazuje aktuálně zvolený list dokumentace a lze na něj přesouvat předlohy. Předlohy a plátno mají určenou velikost, takže pokud se uživatel pokusí vložit předlohu na nedostatečné místo, je akce neprovedena a musí buď zvolit jinou předlohu, vytvořit místo na plátně odstraněním vložených předloh nebo přidat list, na který je možné předlohu přidat. Do vložených předloh lze přetahovat fotografie. Vložení doplní název fotografie do patičky předlohy. V horní části editovacího plátna se nachází panel nástrojů pro práci s listem. Akce umožňují přesun předloh, pohyb listu na plátně, úprava textu předloh, úprava obrazových vlastností fotografií, změna měřítka zobrazovaného listu a vycentrování listu v okně.

## Náhled dokumentace

Slouží k zobrazení náhledu dokumentace. **B.15** Zobrazuje se uložené PDF dokumentace na serveru. Pokud otevřel náhled dokumentace vedoucí, jenž může dokumentaci kontrolovat, tak má v pravé dolní části dvě tlačítka, pomocí kterých buď odsouhlasí dokumentaci nebo zamítne a přidá poznámku pro uživatele, aby mohl uživatel chybu napravit. **B.16 B.17** Potvrzování a zamítnutí vyžaduje potvrzení v dialogu.

K této obrazovce mají přístup všichni uživatelé systému, kromě tisku. Vedoucí zde uvidí případné schvalovací tlačítka.

## Přehled uživatelů

Administrátor je schopen zobrazit přehled všech uživatelů, kteří se mohou přihlásit do systému. **B.24** Může uživatele vytvářet nebo upravovat. **B.25 B.26** Při vytváření uživatele administrátor zadá dočasné heslo, které při přihlášení vyvolá nastavení nového hesla. Pokud za administrátorem dorazí uživatel s požadavkem na změnu hesla, může mu nastavit heslo na předem domluvené. Uživatel bude při dalším přihlášení vyzván ke změně hesla. Nastavení hesla je při úpravě uživatele volitelná. Při úpravě uživatele má administrátor nastavovat uživateli přiřazené role. Uživateli je možné přiřadit více rolí. V přehledu má každý uživatel zobrazen login, jméno, email, telefon, útvar, role, datum vytvoření, datum posledního přihlášení.

Jakékoliv vytváření, upravování nebo potvrzování je realizované pomocí vyskakovacích dialogů.

K této obrazovce má přístup pouze administrátor.

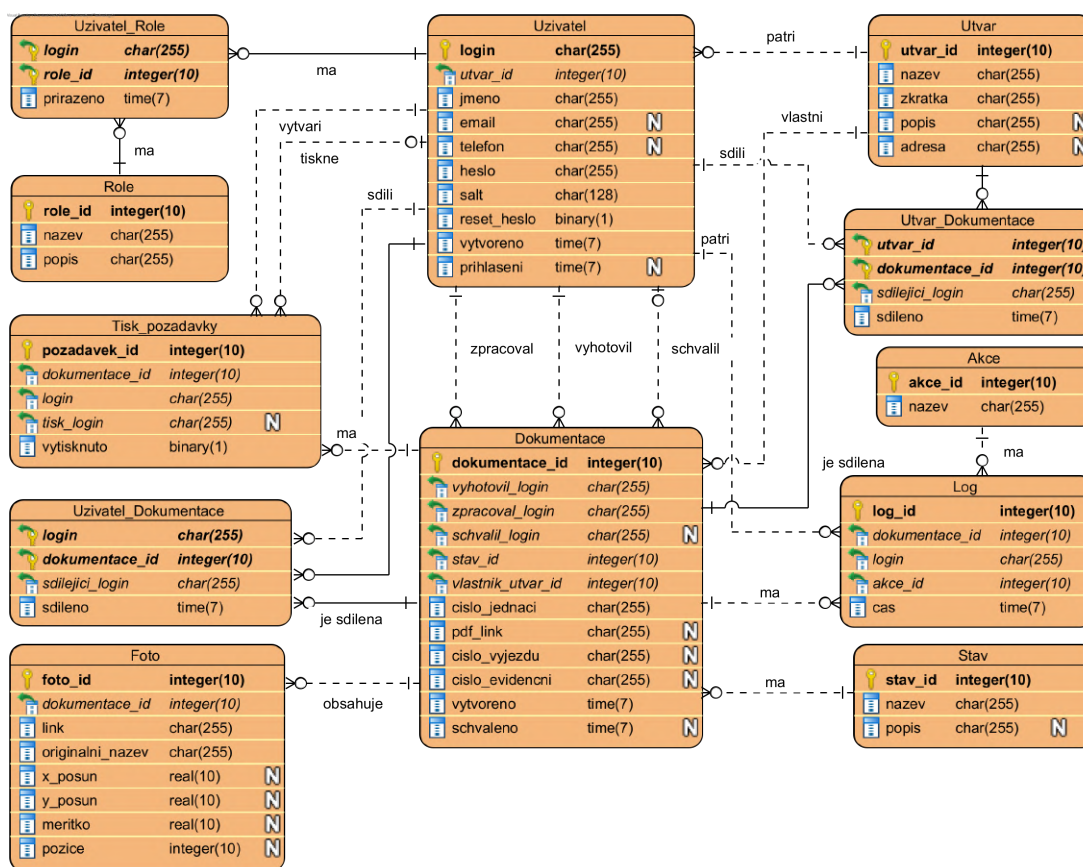
## Přehled útvarů

Administrátor může zobrazit přehled útvarů v systému, do kterých jsou následně přiřazování uživatelé systému. **B.27** Přiřazení do útvaru je pro uživatelé velmi důležité, protože uvidí pouze dokumentace vytvořené v rámci útvaru. Útvary lze vytvářet a upravovat. **B.28 B.29** U každého útvaru je uveden název, zkratka útvaru, případný popis, případná adresa, počet uživatelů v útvaru, počet vedoucích v útvaru a kdy byl útvar vytvořen.

Jakékoliv vytváření, upravování nebo potvrzování je realizované pomocí vyskakovacích dialogů.

K této obrazovce má přístup pouze administrátor.

### 3.5 Návrh databáze



Obrázek 3.4: Diagram návrhu databáze

Hlavní tabulkou v databázi je tabulka s názvem **Dokumentace**, reprezentující vytvářené dokumentace uživateli. Obsahuje své označení pomocí primárního klíče a odkazuje se, pomocí cizích klíčů, na tabulky **Stav** a **Uzivatel**. Tabulka **Stav** udržuje informaci o všech možných stavech, které jsou v systému definovány a kterých mohou dokumentace nabývat. Tabulka **Uzivatel** udržuje informace o všech uživatelích systému a z tabulky **Dokumentace** je na ni odkazováno třikrát a to kvůli informacím, kdo dokumentaci vyhotovil, kdo dokumentaci zpracoval a kdo dokumentaci schválil. Dokumentace mají svého vlastníka, ten je určen pomocí propojení na tabulku **Utvar**. Každá dokumentace může mít u sebe uloženo několik fotografií, informace o nich jsou ukládány v tabulce **Foto**. Takto uložené fotografie může uživatel používat ve vytvářené dokumentaci. Dokumentace si udržují informace o tom, jak je s nimi zacházeno, a to ukládají do tabulky **Log**. Na dokumentaci může uživatel vytvářet požadavky na tisk, tyto požadavky jsou ukládány v tabulce **Tisk\_pozadavky**. Jelikož má být v systému funkční sdílení dokumentací mezi uživateli a útvary, tak existují M:N propojení mezi tabulkou **Dokumentace** a tabulkami **Utvar** a **Uzivatel**.

Tabulka **Uzivatel** udržuje informace o uživatelích, kteří mohou přistoupit do systému. Každý uživatel je identifikován pomocí primárního klíče **Login**. Pomocí cizího klíče na tabulku **Utvar** je uživatel zařazen do útvaru. Uživatel disponuje rolemi, které mu jsou přiřazeny pomocí M:N spojení na tabulku **Role**, díky tomu může mít uživatel variabilní kombi-

naci rolí. Spojení na tabulku *Dokumentace* je zmíněno výše. Uživatel může být přiřazen do tabulky *Log*, pokud udělá nějakou akci s dokumentací, která je definována v systému. Může vytvářet požadavky na tisk, poté je odkazován v tabulce *Tisk\_pozadavky* jako ten, kdo požadavek vytvořil. Pokud má uživatel roli umožňující tisk, může se objevit v tabulce *Tisk\_pozadavky* jako uživatel, jenž požadavek tiskne.

Jelikož má systém umožňovat sdílení dokumentací mezi uživateli a útvary, tak je vytvořeno M:N propojení mezi tabulkou *Dokumentace* a tabulkou *Uzivatel*, u kterého je uložena časová známka sdílení a kdo sdílení vytvořil. Pro sdílení mezi útvary je vytvořeno M:N propojení mezi tabulkou *Dokumentace* a tabulkou *Utvár*.

O uživateli je v databázi uloženo jméno, heslo a nepovinně email, telefonní číslo. Z technického hlediska je o uživateli ještě uložena časová známka vytvoření uživatele, časový známka posledního přihlášení, příznak pro resetování hesla a „salt“ pro hašovací algoritmus hesla. U M:N spojení s rolemi je ukládána časová známka přiřazení dané role. Samotné role mají u sebe uloženou informaci s názvem role a popisu role. Každý uživatel patří do útvaru, kde o útvaru je uložen název a zkratka útvaru. Nepovinně může útvar mít uložen popis a adresa.

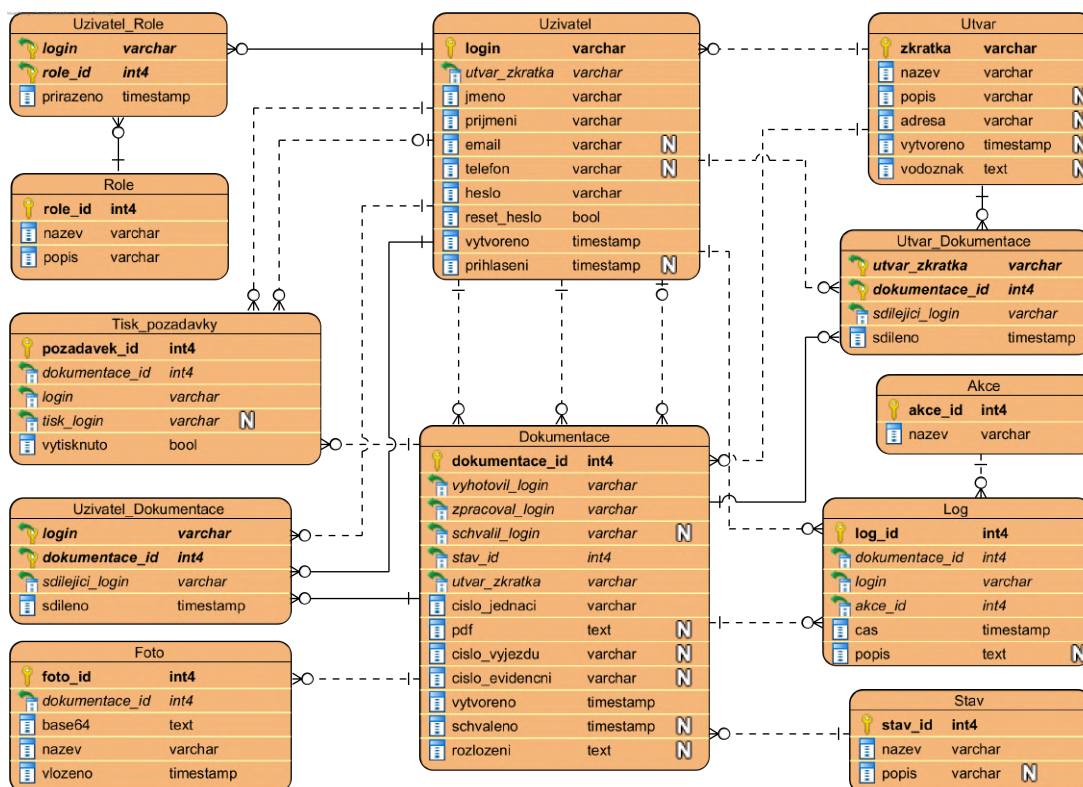
Dokumentace mají v databázi o sobě uložené jednacím číslo, číslo výjezdu nebo evidenční číslo. Z technického hlediska mají dokumentace u sebe časovou známku vytvoření, časovou známku schválení. Každá dokumentace může obsahovat několik fotografií, o fotografiích je v databázi uložen originální název souboru nahraného uživatelem a odkaz na místo na serveru. Z technického hlediska je u každé fotografie uložen posun v ose x, posun v ose y, měřítko v jakém má fotografie být a pozice, kde se nachází v dokumentaci. Dokumentace mají stav v jakém se právě nachází, stavy mají u sebe uložen název stavu a nepovinně popis stavu. Dokumentace mají záznam o provedených akcích s dokumentacích, kde každý tento záznam má svou časovou známku provedení a je spojen s tabulkou akcí, kde každá akce má svůj název. K dokumentaci může být přiřazen požadavek na tisk, u kterého je v databázi uložen příznak vytisknutí, který se změní poté co tiskový personál potvrdí vytisknutí.

# Kapitola 4

## Realizace

Úkolem bylo implementovat serverovou a klientskou aplikaci podle vytvořených návrhů se všemi vyžadovanými funkcemi. Avšak kvůli některým funkcím se muselo od návrhu odklonit a najít jinou cestu, jak funkci implementovat.

### 4.1 Databáze



Obrázek 4.1: Diagram databáze

Pro realizaci byla využita PostgreSQL databáze verze 12. Postup instalace a inicializace databáze je popsán v souboru `instalace.pdf`, jenž je přiložen k zdrojovým kódům backendu.

Struktura databáze se od návrhu příliš nezměnila, avšak došlo ke některých atributů přímo v tabulkách.

- V tabulce *Uzivatel* je odstraněn atribut *salt*, jelikož použitá knihovna pro hašování hesel *argon2*, si *salt* ukládá přímo k vygenerovanému hash hesla a při následném ověření si *salt* z něj načte. Do databáze tedy stačí ukládat pouze tento vygenerovaný hash.
- Tabulka *Utvár* změnila svůj primární klíč na *zkratka* a obsahuje atribut *vodoznak*. Atribut obsahuje vodoznak útvaru, jenž je poté využíván v editoru dokumentací pro automatické doplnění na listy dokumentace.
- Tabulka *Dokumentace* nyní obsahuje atribut *rozlozeni*, do kterého se ukládá aktuální rozložení elementů na listech. To umožňuje jednoduché opětovné načtení editoru dokumentace.
- Do tabulky *Log* je přidán atribut *popis*, jenž má za úkol ukládat například zprávu od vedoucího při zamítnutí dokumentace.
- Tabulka *Foto* neobsahuje atributy *x\_posun*, *y\_posun*, *meritko*, jelikož jsou fotografie zarovnané na střed elementu a ořezávání řeší editor fotografií. Neobsahuje ani atribut *pozice*, jelikož pozice elementů a příslušných fotografií jsou uloženy v atributu *rozlozeni* přímo u dokumentace.

## 4.2 Server

Základem serverové aplikace je Express.JS, který se stará o zpřístupnění a poskytování REST API klientům. Serverový skript vyžaduje k instalaci správce balíčků *npm* a ke svému spuštění běhové JavaScript prostředí *Node.js* alespoň verze 19. Cesta k SSL certifikátu se nastavuje pomocí systémové proměnné *DP\_SSL\_KEY* a *DP\_SSL\_CERT*. Aplikace se připojuje na PostgreSQL databázi, ale je nutné specifikovat detaily připojení pomocí systémových proměnných.

```
DP_DB_HOST=<adresa databáze>
DP_DB_PORT=<port databáze>
DP_DB_NAME=<jméno databáze>
DP_DB_USER=<jméno uživatele>
DP_DB_USER_PASSWORD=<heslo uživatele>
```

Po spuštění se skript pokusí připojit k databázi a začíná poslouchat, na portu 3000 pomocí protokolu HTTPS, požadavky od klientů. Kompletní postup instalace i s příkazy pro Linux Ubuntu jsou obsaženy v souboru *instalace.pdf*.

## Hašování hesel

Pro hašování uživatelských hesel je využita knihovna `argon2`<sup>1</sup>. Knihovna poskytuje funkci *hash*, jenž přijímá vstup od uživatele, nastavení hašování a salt. Na tento vstup vrátí vytvořený hash, který může být bezpečně uložen v databázi. Pro ověření hesla je nutné načíst uložený hash z databáze a použít další funkci z knihovny `argon2`, která se jmenuje *verify*. Funkce přijímá na vstupu nastavení hašování, načtený hash a heslo, které chceme ověřit. Nastavení hašování lze aplikaci upravit systémovými proměnnými `DP_PARALLELISM` (určuje počet použitých vláken pro hašování), `DP_MEMORY_COST` (určuje velikost paměti použité pro hašování) a `DP_TIME_COST` (určuje kolikrát bude hašování probíhat). Salt je 16 náhodných bajtů, jenž se generují pomocí knihovny *crypto*, která je součástí Node.JS.

## Prostředník pro ověření přístupu

Jedná se o funkci, jenž se zpracovává před funkcí samotného požadavku. Stará se o ověření JWT tokenu v požadavku. Vstupem je požadavek, ze kterého zkontroluje existenci cookie s přístupovým JWT. Pokud existuje, zkontroluje se pravost a expirace tokenu. Při neexistenci přístupového JWT nebo chybě při kontrole tokenu, je požadavek zamítnut a je odeslána odpověď s chybovým kódem 401. Zpracování JSON Web tokenů probíhá pomocí knihovny `jsonwebtoken`<sup>2</sup>. Pokud je vše úspěšně ověřeno, jsou k požadavku přidána rozšifrovaná data o uživateli a požadavek pokračuje dále do systému.

## Prostředník pro ověření rolí

Požadavky mohou mít vytvořené omezení, jaká role může požadavek využít. O kontrolu těchto omezení se stará tato funkce, která je zavolána ještě před zpracováním samotného požadavku. Pro administrátory neexistují žádná omezení a mají přístup ke všem požadavkům. Funkce načte uživatelské role z rozšifrovaných detailů o uživateli z přístupového JWT a porovná je na existenci alespoň jedné povolené role požadavku. Pokud je role nalezena, je požadavek předán dále ke zpracování požadavku, jinak je požadavek ukončen odesláním odpovědi s chybovým kódem 403.

### 4.2.1 Ověřování uživatele

#### POST /auth/login

Přihlášení uživatele do klientské aplikace. Bez úspěšného přihlášení nebude moct uživatel pokračovat do aplikace a pracovat s daty na serveru.

Systém ověří existenci údajů *login* a *heslo* v požadavku, pokud se údaje v požadavku nenachází, je požadavek ukončen a je odeslána odpověď s chybovým kódem 400. Pokud se údaje v požadavku nachází, systém zkusí vyhledat uživatele v databázi pomocí předaného přihlašovacího jména. Při nenalezení uživatele je požadavek ukončen a je odeslána odpověď s chybovým kódem 404. Při úspěšném nalezení uživatele, systém ověří jestli předané heslo souhlasí s heslem uloženým v databázi. Pokud uživatel zadal špatné heslo, je požadavek ukončen a v odpověď obsahuje chybový kód 400. Pokud bude heslo úspěšně ověřeno, tak systém zjistí, jestli nemá uživatel čekající požadavek na změnu hesla pomocí nastaveného příznaku *reset\_heslo* u záznamu v databázi. Při existenci čekajícího požadavku na změnu hesla je odeslána odpověď na požadavek s informací o existenci požadavku o změnu hesla.

<sup>1</sup><https://github.com/ransalt/node-argon2>

<sup>2</sup><https://github.com/auth0/node-jwebtoken>



Avšak pokud uživatel nemá čekající požadavek na změnu, jsou načteny role uživatele, které jsou následně přidávány do přístupového JSON Web Tokenu a JSON Web tokenu pro obnovu přístupového JWT. Přístupový i obnovovací token se skládá ze jména uživatele, loginu uživatele, seznam rolí a zkratky útvaru. Tokeny používají různé tajné klíče pro generování. Přístupový token má nastavenou dobu expirace na 30 minut a obnovovací má nastavenou dobu expirace na 1 den. Přístupový token je používán při dalších požadavcích od klientů a je obnovován při aktivním používání pomocí obnovovacího tokenu. Při aktivním používání aplikace může být uživatel přihlášen teoreticky až 1 den a 30minut. Po uplynutí této doby se bude muset uživatel znovu přihlásit. Tokeny jsou předávány v odpovědi na požadavek v podobě cookie s nastavenými atributy `httpOnly`, `secure` a `sameSite=strict`. Při úspěšném přihlášení je v databázi u uživatele aktualizován čas posledního přihlášení.

### **POST /auth/refresh**

Požadavek umožňující obnovu přístupového tokenu pro přístup do aplikace. Pokud v požadavku existuje cookie s obnovovacím tokenem, tedy ještě nevypřichala doba expirace, je ověřen pomocí tajného klíče. Při nenalezení cookie nebo při neúspěšném ověření obnovovacího tokenu je požadavek ukončen a je odeslána odpověď s chybovým kódem 401. Když se podaří ověřit obnovovací token, je vytvořen nový přístupový token z dat obnovovacího tokenu a přidá se opět do cookies odpovědi.

### **POST /auth/reset**

Požadavek využíván, když má uživatel čekající požadavek na změnu hesla a je v aplikaci vyzván na změnu hesla. V požadavku musí být obsažen login uživatele, heslo nastavené administrátorem a nové heslo uživatele. Při nenalezení vyžadovaných dat je požadavek ukončen a je odeslána odpověď s chybovým kódem 400. Systém se pokusí nalézt uživatele pomocí předaného loginu a následně ověří, zda má uživatel nastavený příznak pro změnu hesla. Při pokusu o nastavení hesla u uživatele bez nastaveného příznaku je požadavek ukončen a je odeslána odpověď s chybovým kódem 403. Následně se ověří i předané heslo nastavené administrátorem, pokud je vše ověřeno, je vytvořen hash nového hesla a je nastaven jako heslo uživatele. Pokud uživatel zadal špatné heslo nastavené administrátorem je požadavek ukončen a je odeslána odpověď s chybovým kódem 403. S nastavením nového hesla zaniká příznak pro změnu hesla a uživatel je schopen přihlásit se do aplikace novým heslem.

## **4.2.2 Správa útvarů**

### **POST /utvar**

Slouží pro tvorbu nových útvarů v systému a je omezen na použití pouze administrátorem. Součástí dat předaných požadavku musí být název, zkratka a vodoznak, bez těchto náležitostí je požadavek zamítnut a je odeslána odpověď s chybovým kódem 400. Jelikož by měla být zkratka jednoznačným identifikátorem útvaru, je nejdříve zjištěno, jestli neexistuje útvar s žádanou zkratkou. Při existenci útvaru s žádanou zkratkou je odeslána odpověď s chybovým kódem 400 a zprávou pro systém notifikací aplikace. Jinak je útvar přidán do databáze a v odpovědi je navrácena zkratka útvaru.

### **PUT /utvar/<zkratka>**

Úprava údajů o útvarech, vyžaduje předání povinných náležitostí o útvaru, kterými jsou název a vodoznak. Jinak je požadavek ukončen a je odeslána odpověď s chybovým kódem 400. Lze měnit název, vodoznak, popis a adresu útvaru. Zkratka je pevně vázaná na útvar, nelze ji změnit. Pokud požadavek proběhne úspěšně je v odpovědi navrácena zkratka útvaru. Požadavek může použít pouze administrátor.

### **GET /utvar**

Načítání dat do přehledu útvarů pro administrátora. Požadavek využívá parametry v adrese požadavku, kterými jsou *page*, *size* a *search*. Pomocí těchto parametrů lze omezovat velikost dat v odpovědi, protože se vrací pouze položky pomocí stránkování *page*, o velikosti stránky *size*. Parametr *search* slouží pro vyhledávání v útvarech a vyhledávat lze pomocí zkratky, názvu, popisu a adresy. Požadavek vrací objekt obsahující samotné útvary a dále celkové množství útvarů neohledně na stránkování. Každý záznam útvaru obsahuje zkratku, název, popis, adresu, počet uživatelů, počet vedoucích a datum vytvoření útvaru.

### **GET /utvar/vodoznak/<zkratka>**

Načtení vodoznaku útvaru. Požadavek vyžaduje zkratku v URL požadavku, podle této zkratky probíhá vyhledávání v databázi. Při nalezení záznamu v databázi je navrácen v odpovědi pouze vodoznak útvaru. Hlavní využití je v editoru, do kterého se načítá a doplňuje vodoznak útvaru. Požadavek je omezen na uživatele.

### **GET /utvar/list**

Sloužící pro získávání zjednodušeného seznamu útvaru, kde data o útvaru obsahují pouze zkratku a název. Požadavek je používán při přidělování útvaru uživateli nebo při sdílení dokumentace útvaru. Je omezen na použití vedoucím. Přijímá *page*, *size* a *search* parametr pro omezování vyhledávání. Vyhledává se pouze podle zkratka a názvu útvaru.

## **4.2.3 Správa uživatelů**

### **POST /uzivatel**

Vytvoření nového uživatele v systému, požadavek je omezen pouze na administrátora. Vyžadovanými údaji o uživateli jsou jméno, příjmení, zkratka útvaru, heslo a role. Při nalezení chybějícího údaje je požadavek ukončen odesláním odpovědi s chybovým kódem 400. Požadavek vytváří nový login pro uživatele, který se skládá ze zkratky útvaru a příjmení uživatele, ve tvaru:

`<zkratka>-<prijmeni>-<cislo>`

Následně se vytvoří hash hesla (4.2), který je poté ukládán společně s dalšími údaji o uživateli. Při vytváření uživatele se nastaví příznak pro změnu hesla, který způsobí, že si uživatel bude muset při prvním přihlášení nastavit nové heslo. Po úspěšném uložení uživatele se uživateli přidělí role. Po úspěšném vytvoření uživatele je v odpovědi vrácen login vytvořeného uživatele.

## **PUT /uzivatel/<login>**

Úprava údajů o uživateli, omezeno na použití pouze administrátorem. V požadavku se očekává jméno, příjmení, zkratka útvaru a role. Jestliže nějaký chybí, je požadavek ukončen odesláním odpovědi s chybovým kódem 400. Pokud se v požadavku nachází heslo, znamená to, že je vyžadována změna hesla. Je vytvořen nový hash hesla (4.2), který se použije při úpravě dat v databázi, kde je hash nastaven jako nové heslo a je nastaven příznak pro změnu hesla. Příznak pro změnu hesla určuje, že si uživatel bude muset při příštím přihlášení nastavit nové heslo. Po úspěšné úpravě uživatele jsou načteny aktuálně nastavené role a jsou porovnány s rolemi v požadavku, aby se určilo, které role se mají přidat a které odebrat. Po porovnání dojde k úpravě přidělených rolí.

## **GET /uzivatel**

Načtení dat pro přehled uživatelů využívající omezené odesílání dat pomocí parametrů *page*, *size* a *search* v požadavku. Návratovou hodnotou je objekt složený z vyhledaných záznamů a počtu celkových záznamů bez ohledu na stránkování. Záznam uživatele obsahuje login, jméno, příjmení, email, telefon, název útvaru, ve kterém se aktuálně nachází, role uživatele, datum vytvoření uživatele a datum posledního přihlášení. Vyhledávání je možné podle loginu, jména, příjmení, emailu uživatele nebo názvu útvaru. Požadavek může použít pouze administrátor.

## **GET /uzivatel/list**

Využívaný pro načítání zjednodušených dat o uživateli pro vyhledávání při sdílení dokumentace nebo při přidělování zhotovitele dokumentace. Záznam o uživateli obsahuje jméno, příjmení, login a zkratku útvaru. Používá parametr *search* v požadavku pro zúžení výběru dat, pomocí kterého lze vyhledávat podle loginu, jména a příjmení. V odpovědi se nachází pouze seznam vyhledaných záznamů.

### **4.2.4 Správa dokumentací**

#### **POST /dokumentace**

Vytváření nových dokumentací v systému. Omezeno na uživatele. Požadavek musí obsahovat login zhotovitele, login zpracovatele, zkratku útvaru, číslo jednacích, evidenční číslo nebo číslo výjezdu, rozložení prvků v dokumentaci a ukládané fotografie. Jinak je požadavek ukončen a je odeslána odpověď s chybovým kódem 400. Požadavek se pokusí vložit dokumentaci do databáze, s tím, že vložení vrací identifikátor vložené dokumentace. Identifikátor je následně využit pro přiřazení vkládaných fotografií. Po vložení všech fotografií je vytvořen záznam v logu dokumentací. Při úspěšném provedení všech vkládání do databáze je v odpovědi vrácen identifikátor dokumentace.

### **PUT /dokumentace/share/<id>**

Požadavek pro úpravu sdílení dokumentace, který je omezen na použití vedoucím. Sdílení dokumentace je možné vytvořit mezi uživateli nebo útvary. Vstupem požadavku jsou dva seznamy, seznam útvarů a seznam uživatelů. Pokud v požadavky tyto seznamy nenachází je požadavek ukončen a je odeslána odpověď s chybovým kódem 400. Seznamy obsahují uživatele a útvary, jenž mají mít dokumentaci sdílenou. Systém porovná existující sdílení se seznamy v požadavku a zahájí přidávání a odebírání sdílejících záznamů v databázi.

### **PUT /dokumentace/pdf/<id>**

Odeslání dokumentace ke kontrole vedoucím. Omezeno na použití uživatelem. Požadavek musí obsahovat PDF dokumentace, kterou má vedoucí kontrolovat, jinak je požadavek zamítnut a je odeslána odpověď s chybovým kódem 400. Dokumentaci je v databázi změněn stav a přiřazeno PDF. Po upravení záznamu dokumentace je vytvořen záznam v logu dokumentace informující o této činnosti.

### **PUT /dokumentace/<id>**

Aktualizování informací o dokumentaci. Omezeno na použití uživatelem. V požadavku je vyžadováno aktuální rozložení, fotografie, číslo jednacích, login fotografa a login zpracovatele. Jinak je požadavek ukončen a je odeslána odpověď s chybovým kódem 400. Požadavek aktualizuje informace o dokumentaci včetně rozložení elementů v dokumentaci. Součástí požadavku je i aktualizace fotografií přiřazených k dokumentaci. Aktualizování dokumentace je zaznamenáno v logu dokumentace. Při úspěšném provedení požadavku je v odpovědi identifikátor dokumentace.

### **PUT /dokumentace/review/<id>**

Schválení či zamítnutí dokumentace vedoucím. Omezeno na použití vedoucím. Pokud se jedná o zamítnutí dokumentace, je změněn stav dokumentace na „Zamítnuto“ a přidán záznam do logu dokumentace s případnou zprávou s důvodem zamítnutí od vedoucího. Při schválení dokumentace je změněn stav dokumentace na „Archivováno“, je odstraněno pracovní rozložení, jsou odstraněny pracovní fotografie, je přidán k dokumentaci uživatel, jenž dokumentaci schválil, a je vytvořen záznam v logu dokumentace informující o schválení dokumentace. Při úspěšném provedení požadavku je v odpovědi identifikátor dokumentace.

### **GET /dokumentace**

Získání dat pro přehled aktivních dokumentací. Omezeno na uživatele a vedoucího. Návratovou hodnotou je objekt obsahující celkový počet záznamů a samotné záznamy, kde záznam obsahuje pouze důležité údaje zobrazované uživateli. Jeden záznam obsahuje identifikátor dokumentace, číslo jednacích, evidenční číslo, číslo výjezdu, datum vytvoření dokumentace, zkratka vlastního útvaru, celé jméno fotografa, celé jméno zpracovatele, název stavu a příznak existence PDF dokumentace. Záznamy jsou omezovány pomocí parametrů *page*, *size* a *search*. Vyhledávání je možné podle čísla jednacích, čísla výjezdu, evidenčního čísla, celého jména fotografa, celého jména zpracovatele nebo stavu dokumentace. Pokud požadavek odeslal vedoucí, jsou mu navraceny aktivní dokumentace jen z jeho útvaru. Běžnému uživateli jsou navraceny jen aktivní dokumentace daného uživatele.

### **GET /dokumentace/archiv**

Požadavek získání dat pro přehled archivovaných dokumentací. Omezeno na vedoucího. Vrací objekt obsahující počet možných záznamů a samotné záznamy. Jeden záznam obsahuje identifikátor dokumentace, číslo jednacích, evidenční číslo, číslo výjezdu, datum vytvoření dokumentace, datum schválení dokumentace, zkratka vlastnického útvaru, celé jméno fotografa, celé jméno zpracovatele, celé jméno schvalujícího vedoucího a příznak existence PDF dokumentace. Vracené záznamy jsou omezeny pomocí parametrů *page*, *size* a *search*. Vyhledávat lze pomocí čísla jednacích, čísla výjezdu, evidenčního čísla, jména fotografa, jména zpracovatele a jména schvalujícího vedoucího. Zobrazované dokumentace se omezují dle přihlášeného vedoucího a aktivního sdílení dokumentací.

### **GET /dokumentace/<id>**

Načtení rozložení dokumentace pro funkce editoru. Omezeno na uživatele. Vrací kompletní informace o dokumentace i s rozložením prvků v editoru dle identifikátoru v URL.

### **GET /dokumentace/foto/<id>**

Získání pracovních fotografií přiřazených k dokumentaci pro editor dokumentace. Omezeno na použití uživatelem. Načte fotografie přiřazené k dokumentaci dle identifikátoru z URL požadavku. V odpovědi je seznam načtených záznamů databáze.

### **GET /dokumentace/log/<id>**

Získání záznamu akcí dokumentace. Požadavek načte všechny záznamy přiřazené k dokumentaci dle identifikátoru z URL. Každý záznam obsahuje čas, popis, login uživatele, název akce a celé jméno uživatele. Záznamy jsou seřazeny dle času provedení akce. V odpovědi je seznam načtených dat z databáze.

### **GET /dokumentace/share/<id>**

Načtení uživatelů a útvarů, kterým je dokumentace sdílena. Omezeno na vedoucího. Požadavek načítá seznam útvarů, kterým je dokumentace sdílena dle identifikátoru z URL. Záznam útvaru obsahuje zkratku a název útvaru. Následně se načítají sdílení jednotlivým uživatelům. Záznam uživatele obsahuje login, jméno a příjmení uživatele. V odpovědi se odesílají dva seznamy, seznam uživatelů a seznam útvarů.

### **GET /dokumentace/pdf/<id>**

Načtení PDF dokumentace. Dle identifikátoru předaného v URL se vyhledá odpovídající PDF. V aplikaci se používá pro načtení náhledu dokumentace a při tisknutí dokumentace tiskovým personálem. V odpovědi se nachází PDF dokumentace.

## 4.2.5 Správa tisku

### POST /tisk

Vytvoření nového tiskového požadavku k dokumentaci. Omezeno pouze na vedoucí. V požadavku je vyžadován identifikátor dokumentace, ke které se má vytvořit požadavek tisku. Jinak je požadavek ukončen a je odeslána odpověď s chybovým kódem 400. Nový požadavek se vloží do databáze, i s loginem přihlášeného uživatele, a do logu dokumentace je vložena akce vytvoření nového požadavku. Při úspěšném vytvoření je v odpovědi na požadavek identifikátor požadavku.

### PUT /tisk/confirm/<id>

Potvrzení úspěšného vytisknutí a předání požadavku na tisk. Omezeno na použití tiskovým personálem. V databáze se u požadavku na tisk nastaví login tiskového personálu a nastaví se příznak vytisknuto, tím ho skryje pro další vytisknutí. Akce je zaznamenána v logu dokumentace.

### GET /tisk

Načtení dat tiskových požadavků pro přehled. Omezeno na tiskový personál. Využívá omezení zobrazených výsledků pomocí parametrů *page*, *size* a *search*. Odpověď obsahuje seznam vyhledaných záznamů a celkový počet nezávisle na stránkování. Záznam obsahuje identifikátor požadavku, identifikátor dokumentace, číslo jednací, evidenční číslo, číslo výjezdu, časovou známku vytvoření, časovou známku schválení, celé jméno zhotovitele, celé jméno zpracovatele, celé jméno uživatele, jenž dokumentaci schválil a celé jméno žádajícího. Vyhledávání je možné podle čísla jednacího, čísla výjezdu, evidenčního čísla, celého jména fotografa, celého jméno zpracovatele, celého jména schvalujícího uživatele a celého jméno žádajícího uživatele.

## 4.3 Klientská aplikace

Základem klientské aplikace je Electron React Boilerplate<sup>3</sup>, jenž je základem škálovatelných multiplatformních aplikací, které využívají Electron (3.2.1) a jako frontend React (3.2.2). Součástí je i knihovna pro směrování React Router<sup>4</sup>, knihovna React Fast Refresh<sup>5</sup>, která umožňuje upravovat kód aplikace za běhu bez nutnosti opakovaného kompilování aplikace. Využívá Webpack 5<sup>6</sup> pro zabalení vyžadovaných modulů do jednoho nebo více balíčků využívaných jako statická aktiva.

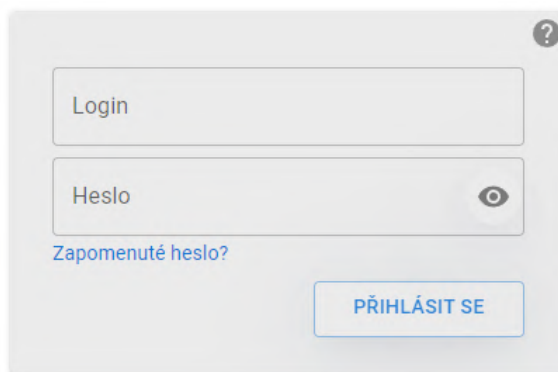
<sup>3</sup><https://electron-react-boilerplate.js.org/>

<sup>4</sup><https://github.com/remix-run/react-router>

<sup>5</sup><https://www.npmjs.com/package/react-refresh>

<sup>6</sup><https://webpack.js.org/>

## Přihlášení

The image shows a login form with a light gray background. At the top right corner, there is a small question mark icon. Below it, there are two text input fields. The first field is labeled 'Login' and contains the text 'Login'. The second field is labeled 'Heslo' and contains the text 'Heslo'. To the right of the 'Heslo' field, there is an eye icon. Below the 'Heslo' field, there is a blue link that says 'Zapomenuté heslo?'. At the bottom right of the form, there is a blue button with the text 'PŘIHLÁSIT SE'.

Obrázek 4.2: Přihlášení

Přihlašovací obrazovka je jedinou obrazovkou, jenž uživatel uvidí při spuštění aplikace. Po uživateli je vyžadováno, aby zadal svůj login, který mu databáze vygenerovala a heslo. V textovém poli hesla lze tlačítkem přepínat mezi viditelnou a skrytou variantou hesla. Pokus o přihlášení se odesílá na server skrze požadavek **POST** */auth/login* (4.2.1). Pokud má uživatel dočasné heslo, je po uživateli vyžadované nastavení nového hesla, které bude moci dále používat. Tlačítkem v pravém horním rohu se otevře dialog s podrobnostmi o aplikaci (4.3). Kliknutím na tlačítko „Zapomenuté heslo?“ se uživateli zobrazí dialog informující uživatele, že heslo je možné obnovit pouze skrze administrátora, který může nastavit nové dočasné heslo (4.4). Po úspěšném přihlášení jsou v aplikaci uloženy detaily o uživateli získané ze serveru a uživatel je přesměrován na jeden z přístupných přehledů podle rolí uživatele. Uživatel nebo vedoucí je přesměrován na přehled aktivních dokumentací, přičemž tiskový personál je přesměrován na přehled požadavků na tisk.

### O programu

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam commodo dui eget wisi. Fusce consectetur risus a nunc. Nunc dapibus tortor vel mi dapibus sollicitudin. Duis pulvinar. In enim a arcu imperdiet malesuada. Etiam bibendum elit eget erat. In enim a arcu imperdiet malesuada. Phasellus rhoncus. Nunc dapibus tortor vel mi dapibus sollicitudin. In laoreet, magna id viverra tincidunt, sem odio bibendum justo, vel imperdiet sapien wisi sed libero. Duis condimentum augue id magna semper rutrum. Aenean id metus id velit ullamcorper pulvinar. Duis sapien nunc, commodo et, interdum suscipit, sollicitudin et, dolor. Integer imperdiet lectus quis justo. Duis viverra diam non justo. Vestibulum erat nulla, ullamcorper nec, rutrum non, nonummy ac, erat. Fusce tellus odio, dapibus id fermentum quis, suscipit id erat. Maecenas sollicitudin. Curabitur ligula sapien, pulvinar a vestibulum quis, facilisis vel sapien.

ZAVŘÍT

Obrázek 4.3: Podrobnosti aplikace

## Zapomenuté heslo?

Zapomenuté heslo musí být řešeno prostřednictvím administrátora.

ROZUMÍM

Obrázek 4.4: Zapomenuté heslo

## Je vyžadovaná změna hesla

Právě jste použili heslo nastavené administrátorem. Nastavte si nyní své nové heslo.

Původní heslo \*



Nové heslo \*



Potvrzení nového hesla \*



POTVRDIT

Obrázek 4.5: Nastavení nového hesla

Dialog pro změnu nastaveného hesla, ve kterém je po uživateli vyžadováno znovu zadání dočasného hesla a dvojité zadání nového hesla. Pokud nová hesla v obou textových polích nesouhlasí, je na tuto skutečnost uživatel upozorněn, požadavek na server není odeslán a je po uživateli vyžadováno, aby problém odstranil. Po zadání souhlasných hesel je odeslán požadavek **POST /auth/reset** (4.2.1). Při úspěšné změně hesla na serveru je uživateli znovu otevřena přihlašovací obrazovka, kde se již může přihlásit novým heslem.

## Hlavní menu



Obrázek 4.6: Hlavní menu

Slouží hlavně pro navigaci uživatele po aplikaci. Tlačítka jsou zobrazována podle rolí přihlášeného uživatele. Role uživatele umožní přístup do přehledu s aktivními dokumentacemi, role vedoucího umožní přístup do přehledu aktivních dokumentací a archivovaných dokumentací. Uživatel s rolí tiskového personálu má přístup do přehledu s tiskovými požadavky. Administrátor má přístup do všech zmíněných přehledů a také má přístup do přehledu



s uživateli a útvary. Po kliknutí na některé z navigačních tlačítek je změněna právě zobrazovaná komponenta pod menu. Na pravé straně menu je zobrazeno celé jméno přihlášeného uživatele, tlačítko pro odhlášení uživatele a tlačítko pro zobrazení dialogu s podrobnostmi o aplikaci (4.3). Při kliknutí na tlačítko pro odhlášení je zobrazen potvrzovací dialog (4.7), ve kterém je uživatel upozorněn na možnou ztrátu neuložených změn.

### Odhlásit se?

Opravdu se chcete odhlásit? Neuložené změny nebudou provedeny.



Obrázek 4.7: Potvrzení odhlášení

## Přehled aktivních dokumentací

| Číslo jednací   | Evidenční číslo | Číslo výjezdu | Vyhotovil     | Zpracoval     | Vytvořen dne | Stav      | Akce |
|-----------------|-----------------|---------------|---------------|---------------|--------------|-----------|------|
| KPRS-6541654165 | KPRS-65445      |               | Test Uživatel | Test Uživatel | 29. 4. 2023  | Vytvořeno |      |

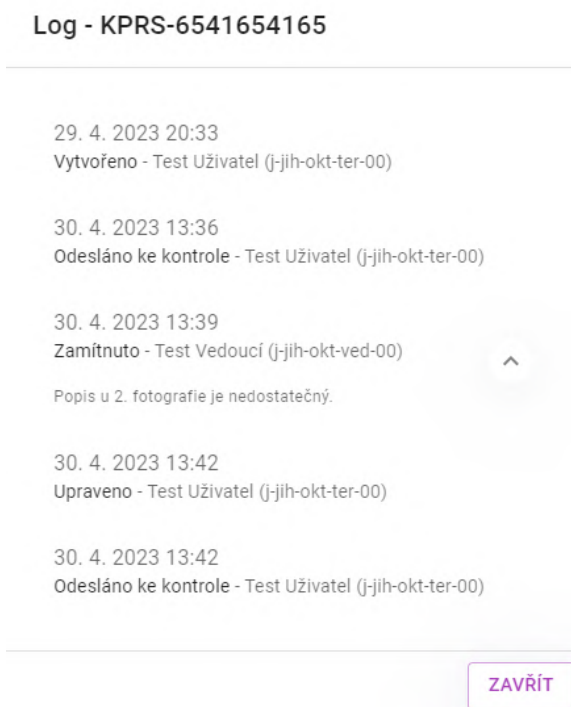
Below the table, there is a search bar with a green '+' button on the left and a search icon. To the right, it shows 'Řádků na stranu 10' and '1-1 ze 1' with navigation arrows.

Obrázek 4.8: Přehled aktivních dokumentací

Jedná se o hlavní přehledovou obrazovku pro uživatele a vedoucí. Uživatelé zde vidí své aktuálně zpracovávané dokumentace a vedoucí zde vidí dokumentace útvaru, které čekají na jeho schválení. Vedoucí může kontrolovat pouze dokumentace z útvaru, do kterého je aktuálně přiřazen, dokumentace z ostatních útvarů nevidí. Administrátor zde může vidět všechny aktivní dokumentace, které ještě nebyly archivovány. V levém dolním rohu se nachází zelené tlačítko, které je přístupné pouze uživateli a administrátorovi a slouží pro otevření editoru nové dokumentace. Přehled využívá dynamické načítání podle tří parametrů, mezi které patří obsah vyhledávacího pole, počet zobrazovaných položek na stranu a aktuální strana. Pro načítání dat do přehledu je využit požadavek **GET /dokumentace** (4.2.4). Před předáním dat univerzální komponentě starající se o zobrazování dat v přehledech, je k datům přidán sloupec s možnými akcemi u dokumentace a upraví se reprezentace stavu z jednoduchého textu na vzhled štítku.

Prvním tlačítkem u dokumentace lze dokumentaci otevřít v editoru (4.23), dostupné pouze uživateli a administrátorovi. Aby mohl vedoucí vytvářet či upravovat dokumentace, musí získat roli uživatele. Druhým tlačítkem lze otevřít náhled připravené dokumentace (4.20), tuto funkci používají primárně vedoucí a to z důvodu, že pouze z náhledu mohou dokumentaci schválit nebo zamítnout. Avšak náhled může otevřít každý s přístupem k dokumentaci, pokud je náhled dostupný, tím se zpřístupní i kliknutí na tlačítko. Náhled dokumentace se běžně stane dostupným až je dokumentace odeslána ke kontrole. Posledním

tlačítkem je otevření dialogu se záznamem aktivit, toto tlačítko může použít každý, kdo má přístup k dokumentaci.








Obrázek 4.9: Záznam aktivit dokumentace

Záznam aktivit zobrazuje všechny akce prováděné s dokumentací. Kompletní záznam aktivit je získán z požadavku **GET /dokumentace/log/<id>** (4.2.4). Záznamy jsou seřazeny od nejstarších a u každého záznamu je zobrazena časová známka provedení akce, druh provedené akce, celé jméno a login uživatele, jenž akci provedl. U záznamu zamítnutí dokumentace a vytvoření požadavku na tisk je dostupný i detail akce, kde u zamítnutí se nachází zpráva od vedoucího, jenž dokumentaci zamítl a u vytvoření požadavku na tisk je uveden identifikátor požadavku pro možné zpětné dohledání. Zaznamenávané akce jsou:

- Vytvoření dokumentace.
- Odeslání dokumentace ke kontrole.
- Schválení dokumentace.
- Zamítnutí dokumentace.
- Upravení dokumentace.
- Vytvoření požadavku na tisk.
- Vyřízení požadavku na tisk.

## Přehled archivovaných dokumentací

| Číslo jednací | Evidenční číslo | Číslo vyjezdu | Vyhotovil     | Zpracoval     | Schválil     | Vytvořen dne | Schváleno dne | Akce  |
|---------------|-----------------|---------------|---------------|---------------|--------------|--------------|---------------|---|
| KPRS-564564   | KPRS-561654     |               | Test Uživatel | Test Uživatel | Test Vedoucí | 28. 4. 2023  | 28. 4. 2023   |      |

Search: \_\_\_\_\_

Řádků na stranu 10 1-1 ze 1 < >

Obrázek 4.10: Přehled archivovaných dokumentací

Obrazovka zobrazovaná vedoucím, jenž zobrazuje již schválené dokumentace. Vedoucí zde vidí schválené dokumentace útvaru a dokumentace, které mu ostatní vedoucí z jiných útvarů sdílely. Pokud má vedoucí přístup ke sdílené dokumentaci, může s ní nakládat, jak s dokumentací z vlastního útvaru. Administrátorovi jsou zobrazeny všechny schválené dokumentace ze všech útvarů. Od navrhovaného přehledu se zobrazovaná data liší pouze v nezobrazování aktuálního stavu, jelikož se všechny dokumentace nacházejí ve stavu „Archivováno“. Dokumentace se do přehledu načítají postupně, podle tří parametrů, obsah vyhledávacího pole, počet řádků na stránku a zvolená stránka. Pro načtení je využíván požadavek **GET /dokumentace/archiv** (4.2.4). Při získání požadovaných dat ze server a před předáním dat komponentě, starající se o zobrazování, jsou k datům přidány všechny možné akce s dokumentací.

První tlačítko u dokumentací umožňuje zobrazení náhledu dokumentace (4.20). Druhé tlačítko umožňuje vytvoření požadavku na tisk, po kliknutí je zobrazen potvrzovací dialog vytvoření požadavku. Po potvrzení dialogu uživatelem je odeslán požadavek **POST /tisk** (4.2.5). Následně je tiskový požadavek přiřazen k přihlášenému vedoucímu a je zobrazen tiskovému personálu z útvaru, do kterého je vedoucí přiřazen. Třetím tlačítkem je stažení PDF souboru vybrané dokumentace na klientské zařízení. Jelikož se samotné PDF dokumentace nenachází v datech pro přehled, je PDF dodatečně staženo ze serveru pomocí požadavku **GET /dokumentace/pdf/<id>** (4.2.4). Vyvolání systémového dialogu pro uložení je provedeno pomocí skryté HTML značky <a>. Dokumentace je exportována v kvalitě 300DPI. Čtvrtým tlačítkem je sdílení dokumentace uživatelům či útvarům, které otevře nový dialog s možností upravovat sdílení dané dokumentace. Posledním tlačítkem je zobrazení dialogu se všemi zaznamenanými akcemi, které byly prováděny s dokumentací (4.9).

### Požadavek na tisk KPRS-564564

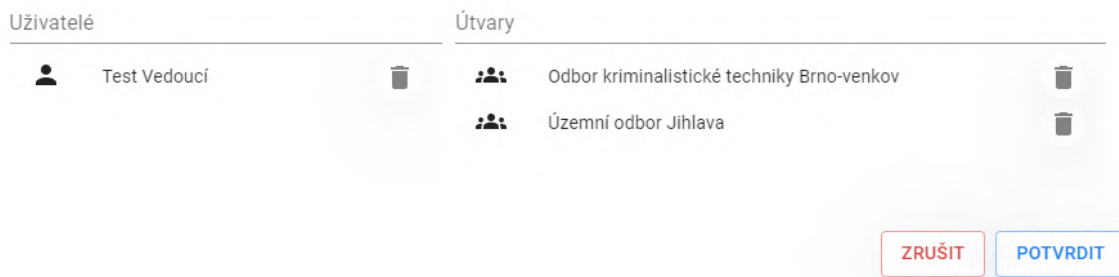
Opravdu chcete vytvořit požadavek na tisk dokumentace. Požadavek bude předán tiskovému personálu.

ZRUŠIT

VYTVŮRIT

Obrázek 4.11: Dialog vytvoření nového požadavku

## Sdílení KPRS-564564



Obrázek 4.12: Dialog pro sdílení dokumentace

V horní části dialogového okna pro sdílení dokumentace se nachází vyhledávací pole pro výběr uživatelů či útvarů. Po zvolení uživatele nebo útvaru se výběr přidá do příslušného seznamu pod vyhledávací pole. Do těchto seznamů se při otevření dialogu načítají aktuálně přiřazené sdílení a je možné sdílení odstraňovat ikonou u položek v seznamech. Načítání dat pro seznamy probíhá pomocí požadavku *GET /dokumentace/share/<id>* (4.2.4). Uložení sdílení dokumentace je provedeno až po kliknutí na tlačítko „Potvrdit“, kterým se vyvolá odeslání požadavku *PUT /dokumentace/share/<id>* (4.2.4).

## Přehled požadavků na tisk

| ID | Číslo jednací | Evidenční číslo | Číslo výjezdu | Vyhotovil     | Zpracoval     | Schválil     | Vytvořen dne | Schváleno dne | Žadající     | Akce |
|----|---------------|-----------------|---------------|---------------|---------------|--------------|--------------|---------------|--------------|------|
| 1  | KPRS-564564   | KPRS-561654     |               | Test Uživatel | Test Uživatel | Test Vedoucí | 28. 4. 2023  | 28. 4. 2023   | Test Vedoucí | ✓ 🖨  |
| 2  | KPRS-564564   | KPRS-561654     |               | Test Uživatel | Test Uživatel | Test Vedoucí | 28. 4. 2023  | 28. 4. 2023   | Test Vedoucí | ✓ 🖨  |

Search:  Rádků na stranu 10 1-2 ze 2 < >

Obrázek 4.13: Přehled Požadavků na tisk

Přehled zobrazovaný tiskovému personálu. Původně se tiskové požadavky měly zobrazovat v přehledu dokumentací, ale kvůli možnému kombinování rolí by tento přístup byl nepraktický a nepřehledný. V přehledu jsou zobrazeny všechny čekající požadavky na tisk od vedoucích ze stejného útvaru, do kterého je aktuálně přiřazen přihlášený uživatel. Od navrhované verze se v datech nezobrazuje pouze stav dokumentace, jelikož je stav vždy „Archivováno“. Data pro přehled se získávají ze serverového požadavku *GET /tisk* (4.2.5). Po přijetí dat ze serveru, se k datům přidá sloupec akcí a následně se data pošlou komponentě pro zobrazování dat. Požadavky se postupně načítají podle uživatelského vstupu ve vyhledávacím poli, zvolených řádků na stranu a zvolené strany. Prvním tlačítkem se potvrzuje vytisknutí dokumentace, finálnímu potvrzení předchází potvrzovací dialog, ve kterém je uživatel upozorněn, aby dokumentaci předal žadajícímu. Po uživatelském potvrzení se odesílá požadavek na server pomocí *PUT /tisk/confirm/<id>* (4.2.5). Druhým tlačítkem uživatel zahájí tisk dokumentace.

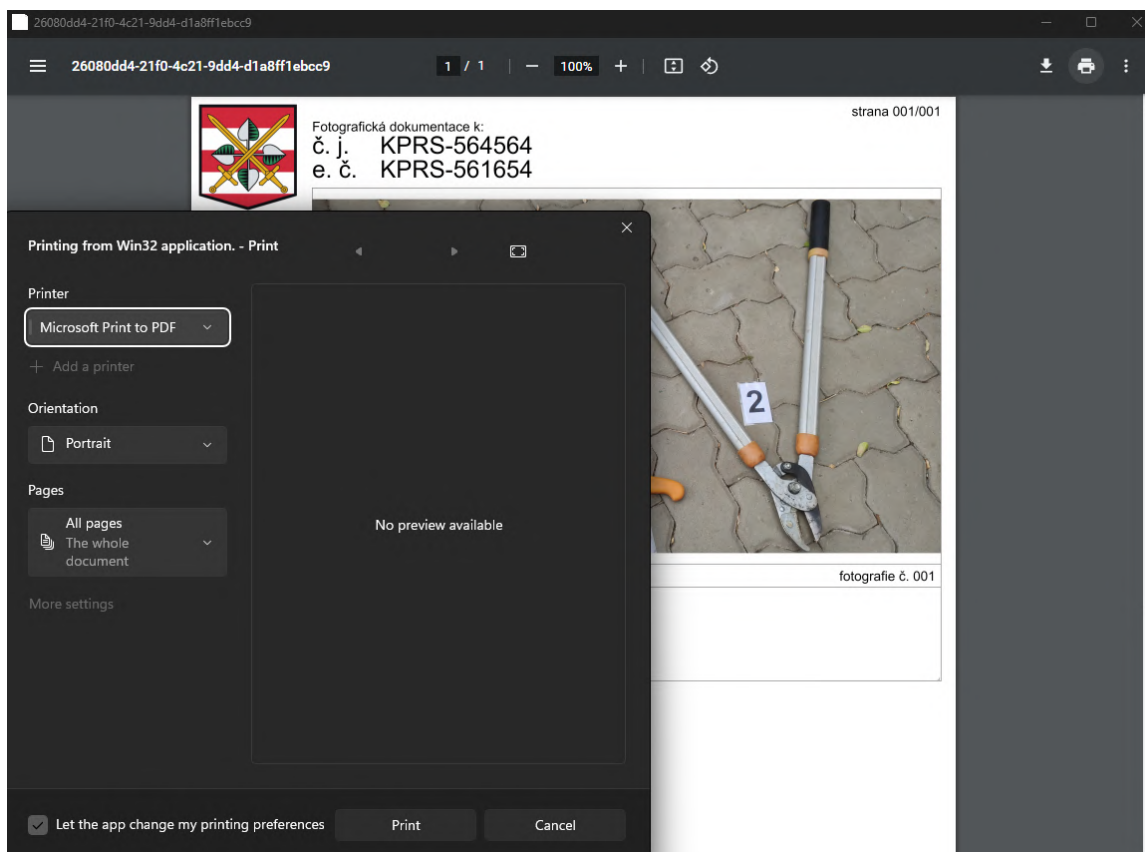
## Opravdu chcete potvrdit vytisknutí dokumentace?

Požadavek potvrzujte až bude dokumentace předána žádajícímu. Po potvrzení požadavku nebude možné vytisknout dokumentaci a požadavek bude označen jako splněný.

ZRUŠIT

POTVRDIT




Obrázek 4.14: Dialog pro potvrzení vytisknutí



Obrázek 4.15: Nové okno pro tisk dokumentace

Po zahájení tisku dokumentace se otevře nové okno s náhledem dokumentace, ze kterého je možné vyvolat systémový dialog tisku. Tento přístup byl zvolen z důvodu, že se nepodařilo zprovoznit náhled listů dokumentace přímo v systémovém dialogu pro tisk, jak je vidět na obrázku 4.15. Tiskový personál by tedy neměl náhled, jak má dokumentace vypadat a nemohl by porovnat předpokládaný vzhled dokumentace s vytištěnou dokumentací. Tiskový dialog umožňuje tisknout různé rozmezí stránek dokumentace, měnit orientaci stránek a další vlastnosti tisku. Avšak této vlastnosti nebude využíváno, jelikož z požadavku od Policie ČR bylo zjištěno, že se provádí tisk pouze všech stránek, na výšku, barevně a jednostranně. Tištěné PDF dokumentace odpovídá vyžadované kvalitě 300DPI.

## Správa útvarů

| Zkratka       | Název                                      | Popis | Adresa | Počet uživatelů | Počet vedoucích | Vytvořen dne | Akce  |
|---------------|--|-------|--------|-----------------|-----------------|--------------|---|
| ADMINISTRÁTOR | Útvar hlavního administrátora              |       |        | 1               | 0               | 01. 1. 2020  |  |
| b-bv-okt      | Odbor kriminalistické techniky Brno-venkov |       |        | 3               | 1               | 28. 4. 2023  |  |
| jjih-okt      | Územní odbor Jihlava                       |       |        | 4               | 1               | 28. 4. 2023  |  |

+

🔍

Řádků na stranu 10 1-3 ze 3 < >

Obrázek 4.16: Přehled útvarů

Data zobrazované ve správě útvarů se drží navrhované verze, avšak změna proběhla v pozici vyhledávacího pole a tlačítka vytvoření nového útvaru. Nyní se nachází vlevo dole a tlačítko pro přidání je výraznější než v navrhované verzi. Zobrazované útvary nejsou načteny všechny najednou, ale načtení probíhá postupně, podle obsahu vyhledávacího pole, nastavených řádků na stránku a nastavené stránce. Lze vyhledávat pomocí zkratky, názvu, popisu a adresy. Získání dat pro přehled probíhá skrze serverový požadavek **GET /utvar** (4.2.2). Po získání dat ze serveru, jsou k datům přidány tlačítka pro úpravu útvarů. Data jsou následně předány univerzální komponentě pro zobrazování přehledů.

Po kliknutí na zelené tlačítko se zobrazí dialog pro vytvoření nového útvaru. Vytvoření nového útvaru vyžaduje nahrání vodoznaku pro dokumentace, zvolení zkratky a názvu útvaru. Ostatní pole jsou volitelná. Od návrhu proběhla změna v nutnosti nahrát vodoznak útvar pro následné používání v dokumentacích. Po kliknutí na tlačítko „Vytvořit“ jsou data odeslána na server pomocí požadavku **POST /utvar** (4.2.2). Stejný dialog existuje i pro úpravu dat o útvaru, který je vyvolán kliknutím na tlačítko u záznamu útvaru. Při otevření dialogu jsou načtena aktuální data o útvaru do příslušných polí. Po úpravě dat a potvrzením dialogu dojde k odeslání požadavku **PUT /utvar/<zkratka>** (4.2.2).

Nový útvar

No file chosen

Zkratka \*









Název \*


Popis

Adresa

Obrázek 4.17: Nový útvar

## Správa uživatelů

| Login           | Jméno | Příjmení    | Email | Telefon | Útvar                                      | Role          | Vytvořen dne | Poslední přihlášení  | Akce  |
|-----------------|-------|-------------|-------|---------|--|---------------|--------------|----------------------|---|
| admin           | ADMIN | ADMIN       |       |         | Útvar hlavního administrátora              | Administrátor | 01. 1. 2020  | 30. 4. 2023 13:28:37 |  |
| b-bv-okt-tis-00 | Test  | Tisk        |       |         | Odbor kriminalistické techniky Brno-venkov | Tisk          | 28. 4. 2023  | 28. 4. 2023 13:09:39 |  |
| b-bv-okt-uzi-00 | Test  | Uživatel    |       |         | Odbor kriminalistické techniky Brno-venkov | Uživatel      | 28. 4. 2023  | 28. 4. 2023 13:04:21 |  |
| b-bv-okt-ved-00 | Test  | Vedoucí     |       |         | Odbor kriminalistické techniky Brno-venkov | Vedoucí       | 28. 4. 2023  | 28. 4. 2023 13:08:52 |  |
| jjlh-okt-res-00 | Test  | Reset Heslo |       |         | Územní odbor Jihlava                       | Uživatel      | 30. 4. 2023  |                      |  |
| jjlh-okt-ter-00 | Test  | Uživatel    |       |         | Územní odbor Jihlava                       | Uživatel      | 28. 4. 2023  | 30. 4. 2023 13:27:27 |  |
| jjlh-okt-tis-00 | Test  | Tisk        |       |         | Územní odbor Jihlava                       | Tisk          | 28. 4. 2023  | 28. 4. 2023 13:09:30 |  |
| jjlh-okt-ved-00 | Test  | Vedoucí     |       |         | Územní odbor Jihlava                       | Vedoucí       | 28. 4. 2023  | 28. 4. 2023 13:08:23 |  |

 Řádků na stranu 10 1-8 ze 8 < >

Obrázek 4.18: Přehled uživatelů

Zobrazovaná uživatelská data jsou identická navrhované verzi. Tlačítko vytvoření nového uživatele a vyhledávací pole je přesunuto doleva dolů. Uživatelské data jsou načítány dynamicky podle zvolených řádků na stranu, zvolené strany a obsahu vyhledávacího pole. V datech lze vyhledávat pomocí loginu, jména, příjmení, emailu, telefonu a jména útvaru. Data jsou získána ze serverového požadavku *GET /uzivatel* (4.2.3), ke kterým je následně přidáno vyvolání akce pro úpravu a reprezentace uživatelských rolí je změněna z prostého textu na štítky. Kliknutím na zelené tlačítko se otevře dialog s vytvořením nového uživatele.

## Nový uživatel

Jméno \*

---

Příjmení \*

---

Email

---

Telefon (s předvolbou)

např. +420123456789

Útvar \*

---

První heslo \*

---

Role\*

- Administrátor
- Vedoucí
- Uživatel
- Tisk

ZRUŠIT

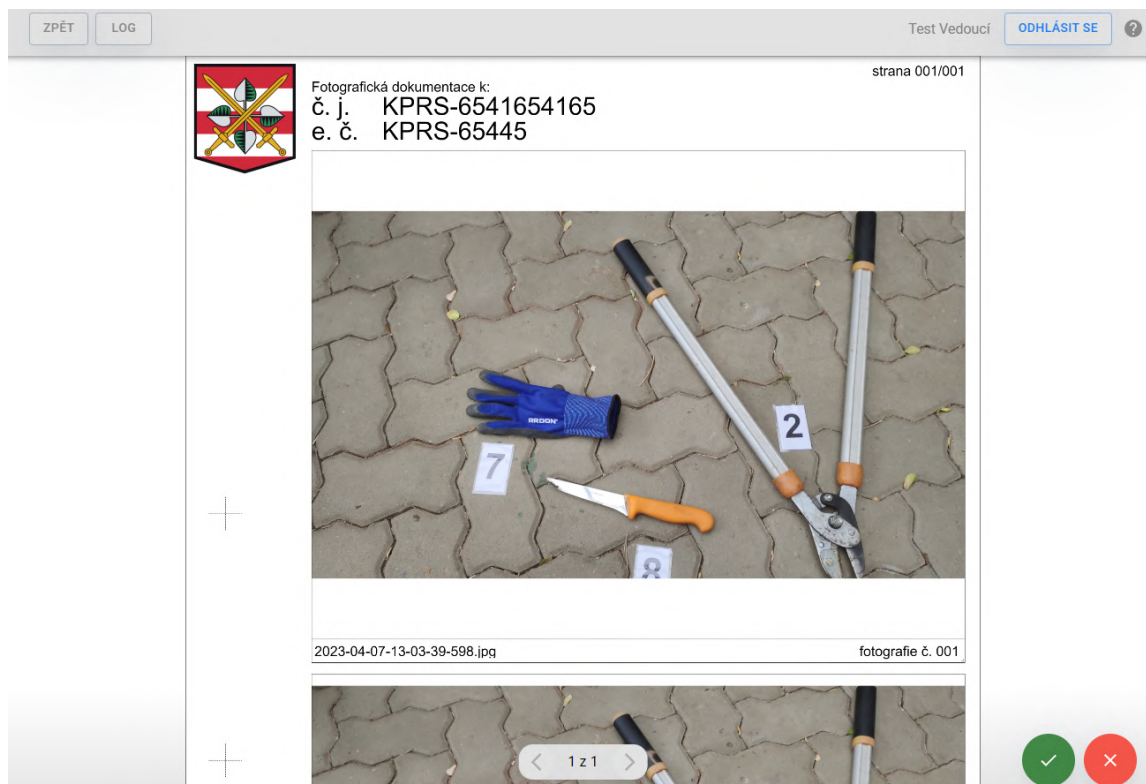
VYTVOŘIT

Obrázek 4.19: Nový uživatel

V dialogu je vyžadované zadání jména, příjmení, útvaru, prvního hesla a rolí. Volitelně se může zadat email a telefon. Pokud uživatel zadává email nebo telefon, probíhá ověření zda uživatelský vstup odpovídá danému formátu. Zadávání útvaru využívá vyhledávací pole, do kterého uživatel zadává vstup, na který mu je zobrazován seznam nalezených výsledků. Z těchto výsledků si může vybrat útvar, do kterého uživatele přiřadí. Vyhledávání využívá serverového požadavku **GET /utvar/list** (4.2.2). Uživateli je možné přiřadit více rolí. Po kliknutí na tlačítko „Vytvořit“ je vytvořen požadavek na server **POST /uzivatel** (4.2.3). Při následné pokusu o přihlášení, nově vytvořeného uživatele, je vyžadována změna hesla. Obdobný dialog je použit i pro úpravu uživatele, do kterého jsou při zobrazování načtena aktuální uživatelská data, upravená data jsou odesílána požadavkem **PUT /uzivatel/<login>** (4.2.3).



## Náhled dokumentace



Obrázek 4.20: Náhled dokumentace

Obrazovka zobrazuje náhled dokumentace načtené přímo z generovaného PDF, které odpovídá vyžadovaným parametrům. K náhledu dokumentace se mohou dostat všichni uživatelé aplikace, pokud mají přístup k nějaké dokumentaci s vytvořeným PDF. V horním menu s obrazovkou náhledu se nachází tlačítko pro návrat do přehledu a tlačítko pro zobrazení záznamu aktivit s dokumentací (4.9). Pokud otevře náhled dokumentace vedoucí z útvaru, do kterého patří i dokumentace a dokumentace je ve stavu „Čeká na schválení“, jsou mu zobrazena dvě tlačítka pro schválení nebo zamítnutí dokumentace. Zeleným tlačítkem dokumentaci schválí, přičemž mu je nejdříve otevřen potvrzovací dialog s poučením. Červeným tlačítkem může dokumentaci zamítnout, avšak nejdříve je mu otevřen dialog s poučením a s prostorem pro vložení zprávy pro uživatele, aby byl uživatel informován, proč byla dokumentace zamítnuta. Po potvrzení dialogu je odeslán požadavek *PUT /dokumentace/review/<id>* (4.2.4), ve kterém se nachází podrobnosti o provedené kontrole.

### Opravdu chcete schválit dokumentaci?

Schválení potvrdí správnost dokumentace a bude umožněn tisk dokumentace.

ZRUŠIT

POTVRDIT

Obrázek 4.21: Potvrzovací dialog schválení dokumentace

### Opravdu chcete zamítnout dokumentaci?

Po zamítnutí dokumentace bude zpracovatel upozorněn, aby udělal úpravy dokumentace, které lze specifikovat níže.

Zpráva pro uživatele...

---

ZRUŠIT

POTVRDIT

Obrázek 4.22: Potvrzovací dialog zamítnutí dokumentace

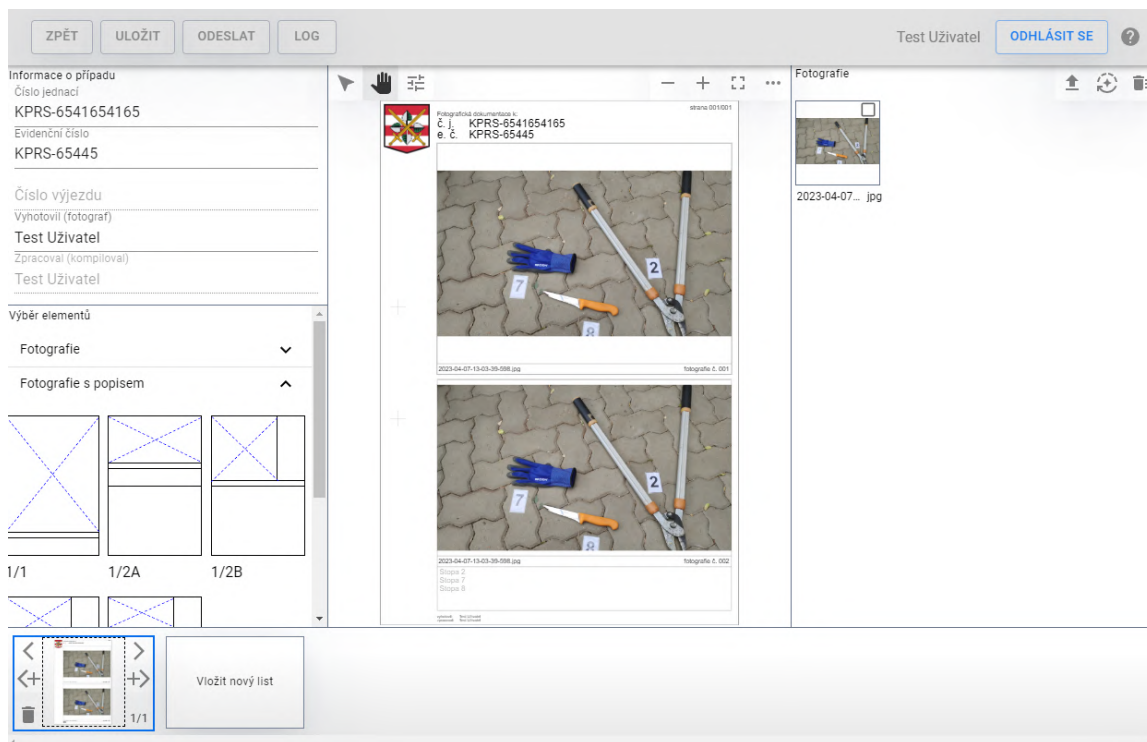
Po otevření náhledu dokumentace je ze serveru stažena aktuální verze uloženého PDF u dokumentace, pomocí požadavku **GET** `/dokumentace/pdf/<id>` (4.2.4). Pro samotné zobrazení je využita knihovna **react-pdf**<sup>7</sup>, která poskytuje komponenty **Document** a **Page**. Komponentě **Document** je předáno stažené PDF, která si jej načte a po načtení nastaví do proměnné počet načtených stránek dokumentu. Komponentě **Page** je nastavena proměnná s identifikátorem pro zobrazení stránky. V rámci prezentace stránek dokumentace byla zvolena varianta zobrazení pouze jedné stránky a změna právě zobrazené stránky se provádí pomocí tlačítek stránkování uprostřed dole. Tlačítka pro stránkování zpracovávají celkový počet stránek a posuvnými tlačítky mění identifikátor zobrazované stránky. Samotnou stránku lze posouvat a přibližovat, aby bylo možné prohlédnout si detaily stránky i při různých velikostech okna. Tato funkcionality je umožněna pomocí knihovny **react-zoom-pan-pinch**<sup>8</sup>, která poskytuje dvě komponenty, **TransformWrapper** a **TransformComponent**. Prvním komponentou se vytvoří prostor, ve kterém je možné posouvat a přibližovat. Druhá komponenta slouží pro obalení samotné komponenty, jenž má mít funkce posouvání a přibližování, v tomto případě je to samotná stránka, tedy komponenta **Page**.

---

<sup>7</sup><https://projects.wojtekmaj.pl/react-pdf/>

<sup>8</sup><https://github.com/prc5/react-zoom-pan-pinch>

## Editor



Obrázek 4.23: Editor dokumentací

Slouží uživatelům na tvorbu a úpravu dokumentací. Skládá se z pěti částí, v levé horní části se nachází hlavní informace o případu, pod informacemi je výběr elementů, uprostřed obrazovky se nachází upravovaná dokumentace, na pravé straně se nacházejí nahrané fotografie k dokumentaci a na spodní části obrazovky je přehled stránek dokumentace. V horním menu nad editorem je tlačítko pro návrat na přehled, tlačítko pro uložení dokumentace, tlačítko pro odeslání dokumentace ke kontrole a tlačítko pro zobrazení záznamu aktivit s dokumentací (4.9). Pro uložení dokumentace musí být vyplněny informace o dokumentaci. Uložení využívá serverový požadavek **POST /dokumentace** nebo **PUT /dokumentace/<id>** (4.2.4), podle existence dokumentace v databázi. Odeslání dokumentace ke kontrole vyžaduje, aby byla dokumentace uložena na serveru, následně lze dokumentaci odeslat dokumentaci ke kontrole pomocí požadavku **PUT /dokumentace/pdf/<id>** (4.2.4). Kvalita ukládaného PDF je 300DPI.

## Informace o případě

Hlavní informace o případě obsahují pět textových polí pro definici dokumentace. Uživatel musí zadat číslo jednacích případů, evidenční číslo nebo číslo výjezdu, které nelze zadat zároveň, pokud se nachází text v jednom, druhé je zablokováno. Dále je nutné zadat fotografa dokumentace, ten je vyhledáván na serveru pomocí zadaného textu v poli a výsledky jsou zobrazovány v podobě seznamu, ze kterého se následně provádí výběr uživatele. Pro vyhledávání se používá serverový požadavek **GET /uzivatel/list** (4.2.3). Poslední pole zobrazuje jméno uživatele jenž dokumentaci zpracoval, v případě nové dokumentace se jedná o jméno přihlášeného uživatele.

## Výběr elementů

Výběr elementů obsahuje všechny elementy, které lze přesouvat do dokumentace. Definice elementů odpovídají vyžadovaným přímo od Policie ČR a elementy jsou rozděleny podle informace, kterou poskytují. Rozměry všech elementů jsou popsány v příloze A s požadavky od Policie ČR. Kategorie „Fotografie“ obsahuje elementy, jenž obsahují pouze samotnou fotografii. Kategorie „Fotografie s popisem“ obsahují elementy, které poskytují jak prostor pro fotografii, tak místo pro případný popis. Tyto elementy poskytují automatickou detekci kriminalistických čísel ve fotografii, kde výsledek je zapsán do textového pole u fotografie, avšak pouze pokud textové pole neobsahuje už nějaký text. Pokud se detekce překrývají je vybrána pouze detekce s největší pravděpodobností. K tomuto filtrování byl využit skript od Hugo Zanini<sup>9</sup>. Elementy obsahující pouze text se nacházejí v kategorii „Text“, do těchto elementů nelze vkládat fotografie a slouží pouze jako textové pole. Elementy umožňují přesun do dokumentace technikou táhni a pusť nebo dvojklikem na element je jimi vyplněn zvolený list.

## Nahrání fotografie

Na pravé straně je prostor pro nahrávání fotografií k dokumentaci. Nahrávání je možné vyvolat kliknutím do prostoru části, pokud se zde nenacházejí žádné fotografie, prvním tlačítkem v pravém horním rohu nebo přesunem fotografií do aplikace. Pro nahrávání fotografií do aplikace je využita knihovna **react-dropzone**<sup>10</sup>. Zpřístupňuje funkci `useDropzone`, pomocí které je možné nastavit `Box` komponentě funkci přijímat fotografie při přesunu na komponentu a zviditelňuje funkci pro vyvolání systémového okna, ve kterém lze vybírat fotografie pro nahrání. Funkce je namapována na první tlačítko v pravém horním rohu nebo na prostor, pokud není nahrána žádná fotografie. Do aplikace je možné nahrávat pouze fotografie ve formátu jpg, toto omezení si definovala Policie ČR. Fotografie jsou v prostoru zobrazovány v ohraničeném čtverci, pod kterým se nachází původní název fotografie. Fotografie lze přesouvat do vložených elementů v dokumentaci. Každá fotografie má u sebe přidáno zaškrtačkové pole, pomocí kterého lze vybírat fotografie pro další akce v pravém horním rohu. První akcí je automatické přidání vybraných fotografií do dokumentace s využitím vybraného elementu, vybraný element je označen modrým ohraničením ve výběru elementů. Přidání probíhá na aktuálně zvolenou stránku, pokud už není možné vložit na aktuální stránku, vygeneruje se nová stránka, na kterou se pokračuje ve vkládání. Poslední akcí je hromadné odstranění nahraných fotografií z aplikace.

<sup>9</sup><https://github.com/hugozanini/yolov7-tfjs/blob/master/src/Utils/nonMaxSuppression.js>

<sup>10</sup><https://github.com/react-dropzone/react-dropzone>

## Přehled stránek dokumentace

Ve spodní části se nachází navigace napříč dokumentací. Navigace obsahuje celkově tři možné komponenty, tlačítko pro přidání nové stránky, tlačítko s náhledem stránky dokumentace a tlačítko s náhledem stránky dokumentace s akcemi. Tlačítkem s náhledem dokumentace bez akcí lze měnit aktuálně zvolenou stránku. Tlačítko s akcemi existuje vždy pouze jedno a symbolizuje právě zvolenou stránku. Vrchní tlačítka akcí umožňují přesunout zvolenou stránku v rámci dokumentace, další tlačítka umožňují přidání stránky před či za aktuálně zvolenou, posledním tlačítkem je odstranění stránky z dokumentace. V pravém dolním rohu se nachází číslo pořadí aktuální strany z celkového počtu stránek. Po kliknutí na tlačítka s náhledy je editor vycentrován na zvolenou stránku. Náhledy stránek dokumentace jsou zmenšené varianty reálné dokumentace, jenž uživatel upravuje.

## Editor dokumentace

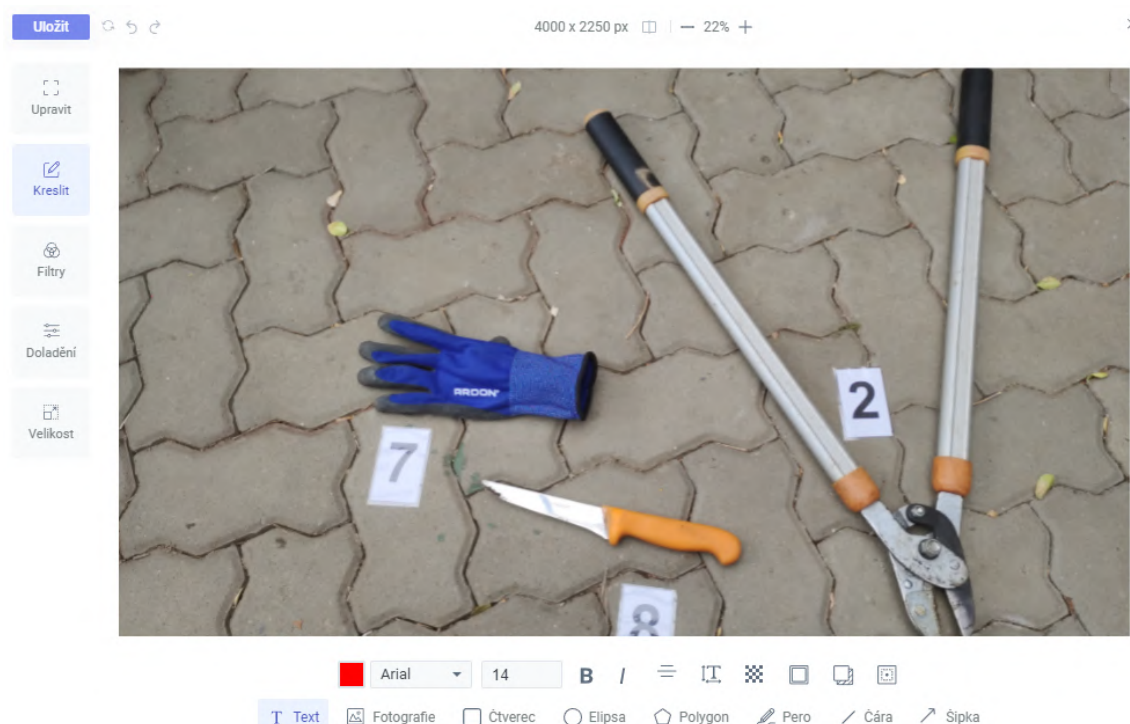
Hlavní část ve středu obrazovky zobrazuje samotný editor dokumentace. Pohyb a přibližování stránkou je umožněno knihovnou *react-zoom-pan-pinch* z poznámky pod čarou 8. Stránka je vytvořena pomocí rozměrů, které byly získány z konzultace s pracovníky fotografických expertiz OKTE KŘP JmK (A.11) a rozměrově je stránka vytvořena, aby odpovídala vyžadovaným 300DPI. Do hlavičky stránky se propisují proměnné číslo jednacích, evidenční číslo nebo číslo výjezdu. V hlavičce je použitým fontem textu Arial ve velikostech 10 a 18 bodů. Vodoznak je doplňován podle útvaru zpracovatele dokumentace. Načtení vodoznaku je prováděno pomocí požadavku *GET /utvar/vodoznak/<zkratka>* (4.2.2). Do patičky stránky se propisují proměnné se jménem fotografa a zpracovatele. Text v patičce používá font Arial ve velikosti 6 bodů.

Pro přesouvání a vkládání elementů je využita knihovna *react-grid-layout*<sup>11</sup>. Knihovna poskytuje komponentu `GridLayout`, do které lze následně přesouvat elementy a přesouvat je i v ní. Vkládání je omezeno tak, že každý element má dané rozměry v rámci stránky, díky kterým není možné na stránku přesunout více elementů než umožňuje prostor stránky. Vložené elementy jsou vlastní komponentou `LayoutElement`, která si udržuje detaily vložené fotografie a text v elementu. Tlačítka v levé horní části editoru slouží pro přepínání aktuálně zvoleného nástroje. První nástroj umožňuje přesouvat elementy na stránce, zablokuje se použití komponenty `TransformWrapper` pro pohyb se stránkou a povolí se pohyb elementů v komponentě `GridLayout`. Druhý nástroj umožňuje pohyb a přibližování stránky, to zablokuje přesouvání elementů v komponentě `GridLayout` a povolí přesouvání stránky pomocí komponenty `TransformWrapper`. Poslední nástroj slouží pro úpravu elementu, to znamená, že se zablokuje jakékoli pohyby se stranami nebo elementy a kliknutím na textové pole, lze do pole vpisovat a kliknutím na fotografii se zobrazí editor fotografie. Tlačítka v pravé horní části slouží pro modifikaci zobrazení. Jedná se o oddálení, přiblížení, zarovnání zvolené stránky na velikost editoru a změna stylu zobrazení stránek (horizontálně, vertikálně).

---

<sup>11</sup><https://github.com/react-grid-layout/react-grid-layout>

## Editor fotografií



Obrázek 4.24: Editor fotografií

Otevírá se v dialogu nad samotným editorem dokumentace po kliknutí editovacím nástrojem na fotografii. Editovací prostředí poskytuje komponenta `FilerobotImageEditor` z knihovny `filerobot-image-editor`<sup>12</sup>. Původně se o úpravu fotografií měla starat knihovna `Caman.JS` (3.3), ale nakonec byla zvolena tato knihovna a to hlavně z důvodu velkých možností přizpůsobení. Je možné zvolit, jaké funkce budou uživateli přístupné a je možné vytvoření vlastního překladu všech prvků v editoru. Pro tuto implementaci byly vybrány funkce pro oříznutí, změna velikosti, použití filtrů, doladění a kreslení do fotografie. Překlad je vytvořen pouze do českého jazyka a nelze jazyk přepínat. Po kliknutí na tlačítko „Uložit“ je přepsán zdroj fotografie v elementu na novou upravenou fotografii, avšak původní fotografie stále existuje v seznamu nahraných fotografií, aby se mohl uživatel případně vrátit k původní fotografii.

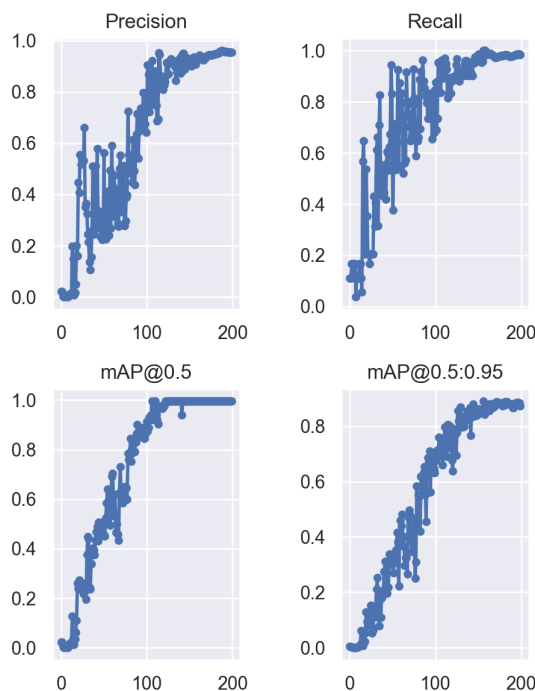
<sup>12</sup><https://github.com/scaleflex/filerobot-image-editor>

### 4.3.1 Rozpoznání čísel ve fotografii

Při vkládání fotografií do elementů s popisem probíhá detekce kriminalistických čísel ve fotografii a předvyplnění popisu výsledkem detekce. K tomu slouží detekční model, který je založený na modelu *YOLOv7* (4). Postup učení modelu a převodu do formátu, který je schopný přijmout knihovna *TensorFlow.JS* (3.3) je převzat od Hugo Zanini[44].

Data set je vytvořen z vlastních fotografií, focených na mobilní telefon do formátu JPG. Na fotografiích jsou foceny štítky kriminalistických čísel, které mají být detekovány. Fotografie jsou následně zmenšeny na rozlišení 480 na 640 pixelů a rozděleny na trénovací a testovací data v poměru 9:1. Jelikož samotné fotografie nenesou informaci o tom, co se na nich nachází, je nutné vytvořit anotace ke každé fotografii. Na vytvoření anotací byl použit nástroj CVAT<sup>13</sup>, který poskytuje prostředí pro anotaci fotografií a exportu do požadovaného formátu YOLOv7. Výsledkem jsou dvě složky, pro trénovací a testovací data, které obsahují fotografie a textové soubory s popisem anotací.

Pro trénování a export modelu bylo použito prostředí Google Colab a Python Notebook. Po nainstalování všech závislostí se klonuje repositář YOLOv7 (4) a instalují se závislosti samotného repositáře. Následně jsou staženy váhy data setu COCO2017<sup>14</sup>, jenž je schopen detekovat 80 tříd objektů. Tyto váhy jsou následně používány k inicializaci modelu, tím dosáhneme urychleného učení. Pro nastavení učení modelu je potřeba vytvořit konfigurační soubory. Po vytvoření konfiguračních souborů lze začít model učit, učení modelu podporuje využití grafické karty.



Obrázek 4.25: Průběh trénování

Grafy z průběhu trénování zobrazují celkem tři metriky. První metrikou je *Precision*, která udává jak dobře umí model identifikovat skutečná pozitiva ze všech pozitivních před-

<sup>13</sup><https://github.com/opencv/cvat>

<sup>14</sup><https://cocodataset.org/>

povědí. Další metrikou je **Recall**, jenž udává schopnost identifikovat skutečná pozitiva ze všech předpovědí. Poslední metrikou je **mAP**, jenž je metrika používaná pro vyhodnocování modelů detekce objektů. Pro výpočet se používá vzorec zpracovávající matici zmatení, metriku IoU, metriku Precision a metriku Recall. Metrika IoU označuje překrytí predikovaného souřadnic ohraničujícího rámečku a reálných souřadnic ohraničení. [35]

Po získání natrénovaného modelu je vhodné provést reparametrizaci vah, repositář YOLOv7 nabízí několik skriptů pro práci s modely a mezi nimi je i skript na reparametrizaci. Následně je možné začít převádět model do formátu TensorFlow.js. Prvně se model převede do formátu ONNX, který lze následně použít pro převod do formátu TensorFlow. Když je model ve formátu TensorFlow, tak je možné jej převést do formátu TensorFlow.js, který už je možné použít ve výsledné aplikaci. Výsledný model je uložen na serveru a klientské aplikace si k němu přistupují pomocí požadavku na server.

## Zhodnocení realizace

Výsledná klientská aplikace je vytvořena pomocí Electron (3.2.1) a obsahuje veškerou funkcionalitu z návrhu. Pro frontend byl využit React (3.2.2) a vzhled z většiny odpovídá návrhu, avšak proběhlo pár menších změn neovlivňujících požadovanou funkcionalitu aplikace. Byl přidán přehled tiskových požadavků, společně s hlavní navigací v horní části aplikace. Tento přístup byl zvolen z důvodu nepřehledného přehledu dokumentací při přiřazení více rolí uživateli. Navrhované funkce u požadavků k tisku zůstaly stejné jako v návrhu, jen se separovaly tiskové požadavky od přehledu dokumentací. Další změna proběhla při vytváření a úpravě útvarů, kde je nyní vyžadované vložení vodoznaku, jenž bude používán při generování fotografické dokumentace. Vodoznak nebyl v návrhu vůbec vyřešen. Funkcionalita byla testována vůči vytvořeným přijímacím testům z návrhu, které se nacházejí v souboru `prijimaci_testy.pdf` na odevzdaném médiu. Instalace aplikace na operační systém Windows se provádí pomocí instalátoru, který umožňuje jednoduché nasazení.

Serverová realizace je vytvořena pomocí Express (3.2.4) a odpovídá funkcionálně vytvořenému návrhu a kompletně podporuje funkce klientské aplikace. Použitou databází je PostgreSQL (3.2.3), ve které byl použit diagram databáze (4.1). Diagram se téměř drží navrhovaného ER diagramu (3.4), ale proběhlo pár malých změn v attributech některých tabulek, přičemž se jednalo spíše o nevyužití navrhovaných atributů a v případě tabulky s útvary se přidal atribut pro ukládání vodoznaku. Komunikaci mezi klientem a serverem se podařilo úspěšně zabezpečit pomocí TLS certifikátu a protokolu HTTPS. Instalace serverové realizace je popsána v souboru `instalace.pdf` ve složce Server se zdrojovými kódy. Implementace klientské aplikace a serverové části kompletně odpovídá požadavkům Policie ČR.



## Možná rozšíření aplikace

**Extrakce fotografie z video záznamu** Aplikace aktuálně přijímá fotografie ve formátu jpg. S tímto rozšířením by byl uživatel schopen extrahovat fotografii z video záznamu. Aplikace by načetla video záznam a zobrazila uživateli přehrávač videa s možností podrobného procházení po snímcích.

**Import PDF do aplikace** Uživatel má k dispozici PDF soubor již vytvořené dokumentace. Aplikace by byla schopna PDF načíst, extrahovat důležitá data a vytvořit záznam dokumentace v systému.

**Sdílení dokumentace pomocí odkazu** Vedoucí je schopen sdílet dokumentace dalším vedoucím a útvarům systému. S tímto rozšířením by měl schopnost vytvořit odkaz na PDF dokumentace, jenž by PDF zpřístupnil i mimo aplikaci. K PDF by bylo možné přidat heslo.

**Dvou-faktorové přihlašování** Všichni uživatelé aplikace se nyní přihlašují pomocí loginu a hesla. Rozšíření by přidalo dvou-faktorové ověření, například pomocí služebního telefonu.

**Kontrola dokumentací s grafickými prvky** Vedoucí je v aplikaci schopen přidat k zamítnutí dokumentace text, jenž popisuje důvod zamítnutí. Toto rozšíření přidá možnost komentovat problémy dokumentace přímo v PDF dokumentace. Možnost využití grafického tabletu pro označování problémů.

**Automatické aktualizace** Aplikace by byla schopna zkontrolovat dostupné aktualizace, stáhnout potřebné soubory a aplikovat aktualizaci aplikace bez nutnosti přehledání.

**Webová verze aplikace** Aplikace využívá běžně využívané webové technologie. Zpřístupnění aplikace, například v omezeném režimu, by rozšířilo použitelnost aplikace.

## Kapitola 5

# Závěr

Cílem diplomové práce bylo vytvoření a nasazení serverové a klientské aplikace, která by odpovídala požadavkům Policie ČR a zadání diplomové práce. K tomu bylo potřeba prostudovat techniky používané Policií ČR při vytváření fotografické dokumentace. Bylo nutné porovnat aktuální možnosti vývoje klient-server aplikace na operační systém Windows a vytvořit návrh systému, složený ze serverové části a klientské aplikace. Následně bylo nutné navržený systém implementovat.

Výsledná implementace využívá PostgreSQL databázi a Express skript jako serverovou část systému. Klientská aplikace je založená na Electron a je spustitelná na Windows operačním systému. Aplikace umožňuje spravovat dokumentace, tiskové požadavky, uživatele a útvary systému. Editor zjednodušil tvorbu fotografické dokumentace a detekce kriminalistických čísel ve fotografiích je schopna předvyplnit šablonu dokumentace. Komunikace mezi klientem a serverem je zabezpečena pomocí TLS certifikátu a protokolu HTTPS. Uživatelská hesla jsou na serveru ukládána v zabezpečené formě, která využívá hašování a „salting“. Nad rámec zadání a požadavků od Policie ČR, aplikace obsahuje editor vložených fotografií, jenž umožňuje úpravu fotografií v aplikaci.

Implementovaný systém odpovídá vytvořeným návrhům, funkcionality aplikace odpovídá přijímacím testům a výstupní PDF splňují požadavky na vzhled fotografické dokumentace. Do budoucna má aplikace potenciál být rozšířená o mnoho funkcí, v textu je zmíněno pár příkladů možného rozšíření aplikace. Avšak nyní by se aplikace měla začít používat, aby se našly případné slabiny aplikace a mohly se tyto slabiny opravit před rozšiřováním funkcionality.

# Literatura

- [1] AARONWARD. *Facial Detection — Understanding Viola Jones' Algorithm* [online]. Medium, Jan 2020 [cit. 2023-05-11]. Dostupné z: <https://medium.com/@aaronward6210/facial-detection-understanding-viola-jones-algorithm-116d1a9db218>.
- [2] ASYNCBANANA. *Is electron the best desktop framework to use in 2021?* [online]. Jul 2021 [cit. 2022-11-23]. Dostupné z: <https://javascript.plainenglish.io/is-electron-the-best-desktop-framework-to-use-in-2021-e525638b9b6a>.
- [3] AUTH0.COM. *JSON Web Tokens - jwt.io* [online]. Auth0, 2013 [cit. 2023-05-02]. Dostupné z: <https://jwt.io/introduction>.
- [4] BOESCH, G. *YOLOv7: The Most Powerful Object Detection Algorithm (2023 Guide) - viso.ai* [online]. Feb 2023 [cit. 2023-05-11]. Dostupné z: <https://viso.ai/deep-learning/yolov7-guide/>.
- [5] *CamanJS - Javascript Image Manipulation* [online]. 2022 [cit. 2022-12-10]. Dostupné z: <http://camanjs.com/>.
- [6] CHURCHVILLE, F. *User interface (UI)* [online]. TechTarget, 2021 [cit. 2022-12-10]. Dostupné z: <https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI>.
- [7] *What is client server architecture? - differences, types, example* [online]. Mar 2022 [cit. 2022-11-23]. Dostupné z: <https://intellipaat.com/blog/what-is-client-server-architecture/>.
- [8] DAVIDBRITCH, BANOVVV a JCONREY. *What is .net maui? - .net maui* [online]. Nov 2022 [cit. 2022-11-23]. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui?view=net-maui-7.0>.
- [9] DESHPANDE, C. *The Best Guide to Know What Is React* [online]. Simplilearn, Jun 2020 [cit. 2022-11-25]. Dostupné z: <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>.
- [10] DESHPANDE, C. *What is angular?: Architecture, features, and advantages [2022 edition]* [online]. Simplilearn, Oct 2022 [cit. 2022-11-23]. Dostupné z: <https://www.simplilearn.com/tutorials/angular-tutorial/what-is-angular>.
- [11] *What is Desktop Software?* [online]. Jun 2019 [cit. 2022-12-6]. Dostupné z: <https://www.educba.com/what-is-desktop-software/>.
- [12] DIGICERT®. *What is SSL, TLS and HTTPS?* [online]. 2017 [cit. 2023-04-14]. Dostupné z: <https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https>.

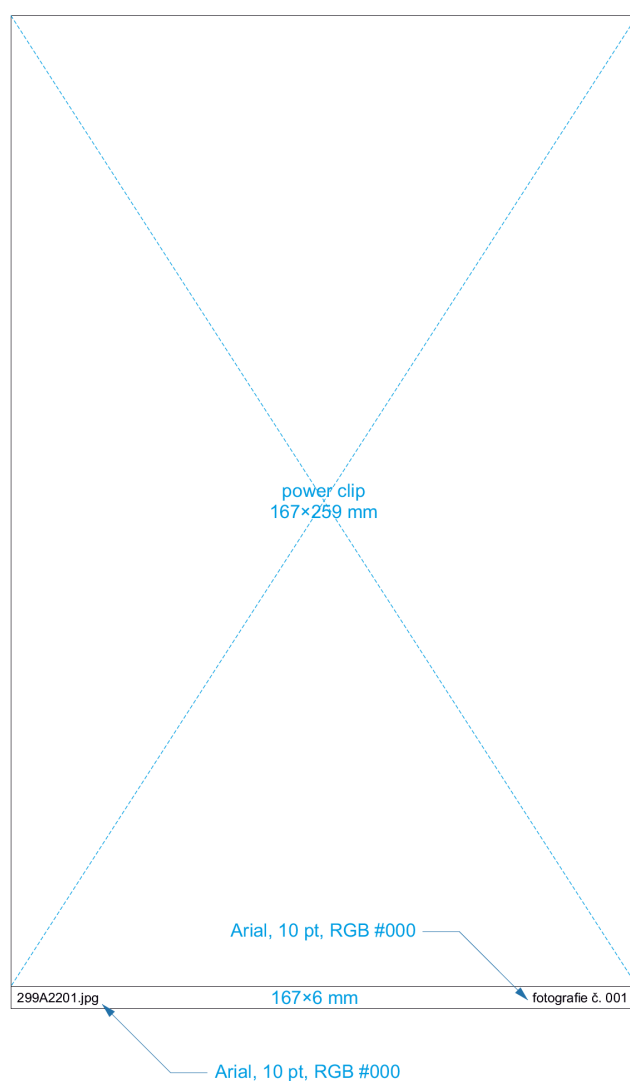
- [13] EDUCATION, I. C. *What are Convolutional Neural Networks?* [online]. Oct 2020 [cit. 2022-12-19]. Dostupné z: <https://www.ibm.com/cloud/learn/convolutional-neural-networks>.
- [14] EDUCATION, I. C. *What is computer vision?* [online]. 2022 [cit. 2023-04-27]. Dostupné z: <https://www.ibm.com/topics/computer-vision>.
- [15] *Electron Tutorial* [online]. 2022 [cit. 2022-11-25]. Dostupné z: <https://www.tutorialspoint.com/electron/>.
- [16] *Frontend vs Backend - GeeksforGeeks* [online]. Jul 2019 [cit. 2022-12-7]. Dostupné z: <https://www.geeksforgeeks.org/frontend-vs-backend/>.
- [17] GAILLE, B. *15 Desktop vs Web Application Pros and Cons* [online]. Mar 2019 [cit. 2022-12-6]. Dostupné z: <https://brandongaille.com/15-desktop-vs-web-application-pros-and-cons/>.
- [18] *Learn mysql tutorial - javatpoint* [online]. 2021 [cit. 2022-11-23]. Dostupné z: <https://www.javatpoint.com/mysql-tutorial>.
- [19] *Pros and cons of reactjs - javatpoint* [online]. 2021 [cit. 2022-11-23]. Dostupné z: <https://www.javatpoint.com/pros-and-cons-of-react>.
- [20] JETHVA, H. *MySQL VS postgresql - what's the difference (pros and cons)* [online]. Mar 2022 [cit. 2022-11-23]. Dostupné z: <https://cloudinfrastructureservices.co.uk/mysql-vs-postgresql/>.
- [21] JONES, M., BRADLEY, J. a SAKIMURA, N. *JSON Web Token (JWT)* [RFC 7519]. RFC Editor, may 2015. DOI: 10.17487/RFC7519. Dostupné z: <https://www.rfc-editor.org/info/rfc7519>.
- [22] KOŘOUSKOVÁ, B. *Vue Js: Výhody, nevýhody a možnosti Využití* [online]. Jul 2021 [cit. 2022-11-23]. Dostupné z: <https://www.rascasone.com/cs/blog/co-je-framework-vuejs>.
- [23] KYAW, W. Y. *Histogram of Oriented Gradients - Wai Yan Kyaw - Medium* [online]. Medium, May 2021 [cit. 2023-05-11]. Dostupné z: <https://waiyankyawmc.medium.com/histogram-of-oriented-gradients-90567ea6490a>.
- [24] LEŠKA Šimon. *Co je to UX a UI design?* [online]. Nov 2020 [cit. 2023-05-03]. Dostupné z: <https://www.blueghost.cz/clanek/co-je-to-ux-a-ui-design/>.
- [25] MOMJIAN, B. a GREGOR, J. *PostgreSQL: praktický průvodce*. Computer Press, 2003. ISBN 9788072269549. Dostupné z: <https://books.google.cz/books?id=tv4AAAACAAJ>.
- [26] MOZILLA. *Pdf.js/README.md at master · mozilla/pdf.js* [online]. Oct 2022 [cit. 2023-04-20]. Dostupné z: <https://github.com/mozilla/pdf.js/blob/master/README.md>.
- [27] NIELSEN, H., MOGUL, J., MASINTER, L. M., FIELDING, R. T., GETTYS, J. et al. *Hypertext Transfer Protocol – HTTP/1.1* [RFC 2616]. RFC Editor, červen 1999. DOI: 10.17487/RFC2616. Dostupné z: <https://www.rfc-editor.org/info/rfc2616>.
- [28] *Node.js Introduction* [online]. 2023 [cit. 2022-11-23]. Dostupné z: [https://www.w3schools.com/nodejs/nodejs\\_intro.asp](https://www.w3schools.com/nodejs/nodejs_intro.asp).

- [29] *Object Detection vs Object Recognition vs Image Segmentation - GeeksforGeeks* [online]. Feb 2020 [cit. 2022-12-19]. Dostupné z: <https://www.geeksforgeeks.org/object-detection-vs-object-recognition-vs-image-segmentation/>.
- [30] *What is a REST API?* [online]. 2022 [cit. 2022-12-5]. Dostupné z: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>.
- [31] RESCORLA, E. *The Transport Layer Security (TLS) Protocol Version 1.3* [RFC 8446]. RFC Editor, srpen 2018. DOI: 10.17487/RFC8446. Dostupné z: <https://www.rfc-editor.org/info/rfc8446>.
- [32] *What is REST - REST API Tutorial* [online]. May 2020 [cit. 2022-12-5]. Dostupné z: <https://restfulapi.net/>.
- [33] RICHARDSON, L. a AMUNDSEN, M. *RESTful Web APIs*. O'Reilly, 2013. ISBN 9781449358068. Dostupné z: <https://books.google.cz/books?id=i3a7mAEACAAJ>.
- [34] SCHINKEL, P. *What is Frontend? What is Backend?* [online]. Frontend GmbH, Nov 2019 [cit. 2022-12-7]. Dostupné z: <https://www.frontend-gmbh.de/en/blog/what-is-frontend-what-is-backend/>.
- [35] SHAH, D. *Mean Average Precision (mAP) Explained: Everything You Need to Know* [online]. Mar 2022 [cit. 2023-05-02]. Dostupné z: <https://www.v7labs.com/blog/mean-average-precision>.
- [36] SHARMA, A. *Express JS tutorial: What is express in node JS?* [online]. Simplilearn, Nov 2022 [cit. 2022-11-23]. Dostupné z: <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-express-js>.
- [37] SHUBHADARSHIE a VINAYEDULA. *Node.js vs express.js* [online]. Jun 2022 [cit. 2022-11-23]. Dostupné z: <https://www.geeksforgeeks.org/node-js-vs-express-js/>.
- [38] SUPERTOKENS. *Hash, salt and verify passwords - Node, Python, Go and Java* [online]. SuperTokens, 2022 [cit. 2022-12-1]. Dostupné z: <https://supertokens.com/blog/password-hashing-salting>.
- [39] TENSORFLOW. *TensorFlow Object Detection API* [online]. Nov 2022 [cit. 2023-04-20]. Dostupné z: [https://github.com/tensorflow/models/blob/master/research/object\\_detection/README.md](https://github.com/tensorflow/models/blob/master/research/object_detection/README.md).
- [40] TERRA, J. *What is client-server architecture? everything you should know: Simplilearn* [online]. Simplilearn, Oct 2022 [cit. 2022-11-23]. Dostupné z: <https://www.simplilearn.com/what-is-client-server-architecture-article>.
- [41] *What is User Interface (UI) Design? | IxDF* [online]. UX courses, 2022 [cit. 2022-12-10]. Dostupné z: <https://www.interaction-design.org/literature/topics/ui-design>.
- [42] VIKTOR. *Cross-platform app development or why should I choose qt?* [online]. Jun 2017 [cit. 2022-11-23]. Dostupné z: <https://blog.vakoms.com/qt-cross-platform-application-framework-is-it-worth-it/>.

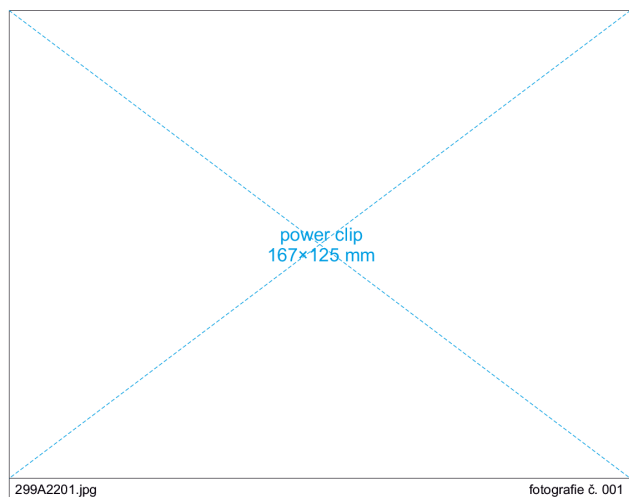
- [43] WANG, C.-Y., BOCHKOVSKIY, A. a LIAO, H.-Y. M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *ArXiv preprint arXiv:2207.02696*. 2022, [cit. 2023-05-11].
- [44] ZANINI, H. *Custom YOLOv7 Object Detection with TensorFlow.js* [online]. Mar 2023 [cit. 2023-04-20]. Dostupné z: <https://towardsdatascience.com/training-a-custom-yolov7-in-pytorch-and-running-it-directly-in-the-browser-with-tensorflow-js-96a5ecd7a530>.
- [45] § 113 odst. 2 zákona č. 141/1961 Sb., o trestním řízení soudním (trestní řád) - znění od 28. 6. 2022 [online]. ČESKO, © AION CS 2010-2022 [cit. 2022-11-23]. Dostupné z: <https://www.zakonyprolidi.cz/cs/1961-141#p113-2>.
- [46] § 69-70 zákona č. 273/2008 Sb., o Policii České republiky - znění od 28. 6. 2022 [online]. ČESKO, © AION CS 2010-2022 [cit. 2022-11-23]. Dostupné z: <https://www.zakonyprolidi.cz/cs/2008-273#p69>.

# Příloha A

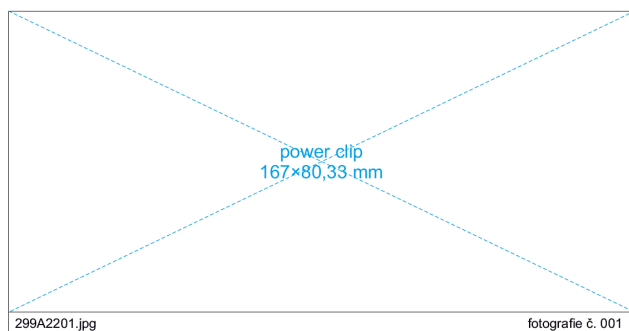
## Požadavky od Policie ČR



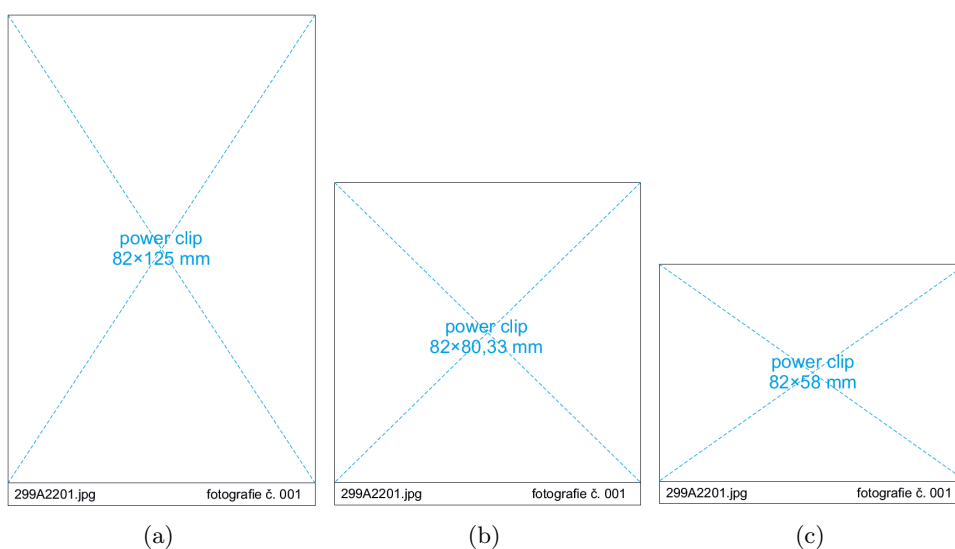
Obrázek A.1: Rozměry 1/1 elementu



Obrázek A.2: Rozměry 1/2 elementu

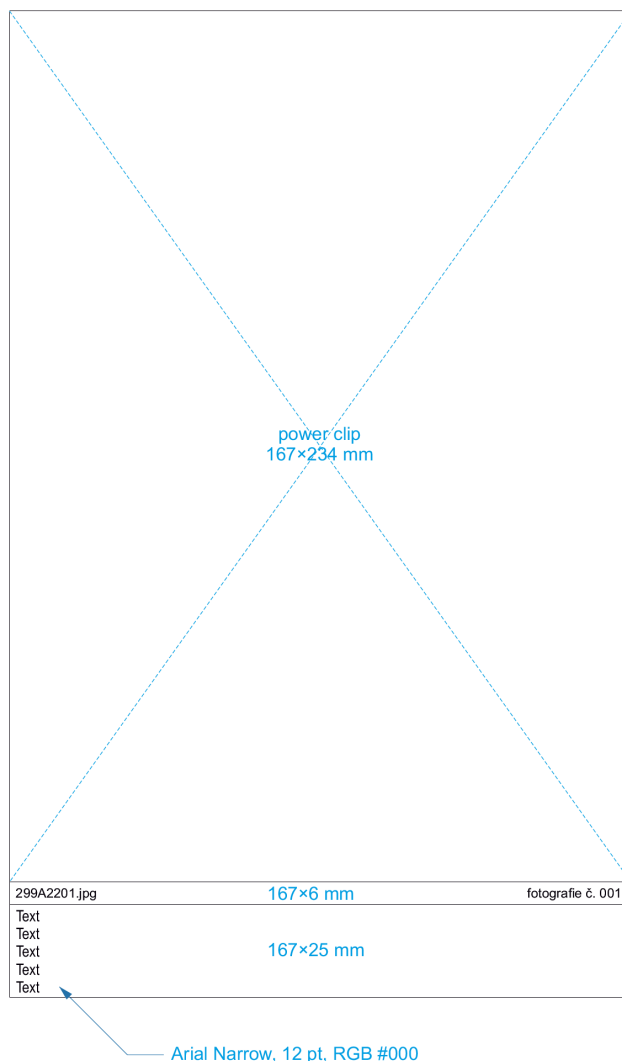


Obrázek A.3: Rozměry 1/3 elementu

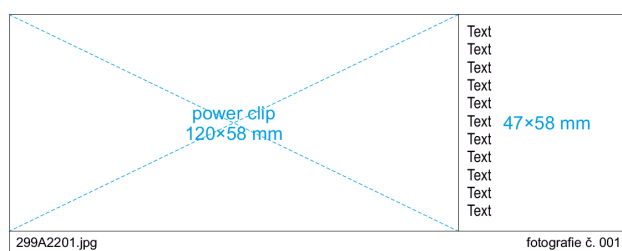


Obrázek A.4: Rozměry 1/4, 1/6 a 1/8 elementu

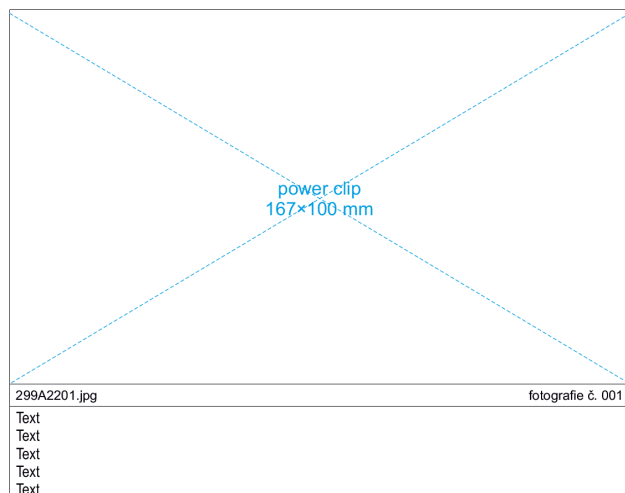




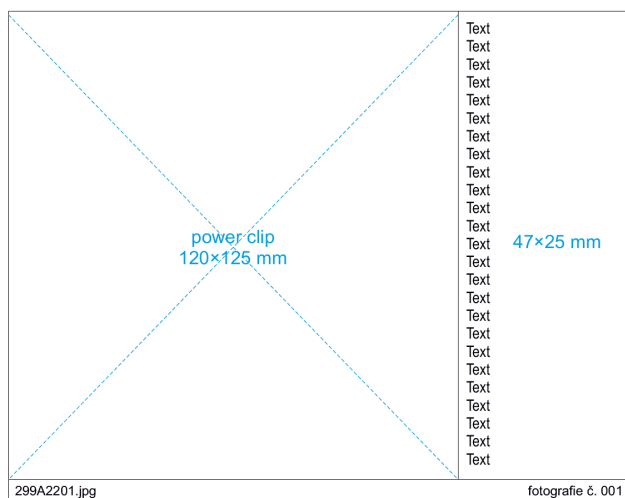
Obrázek A.5: Rozměry 1/1 elementu s textem



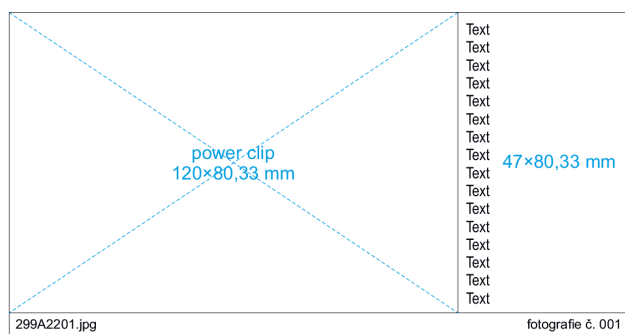
Obrázek A.6: Rozměry 1/4 elementu s textem



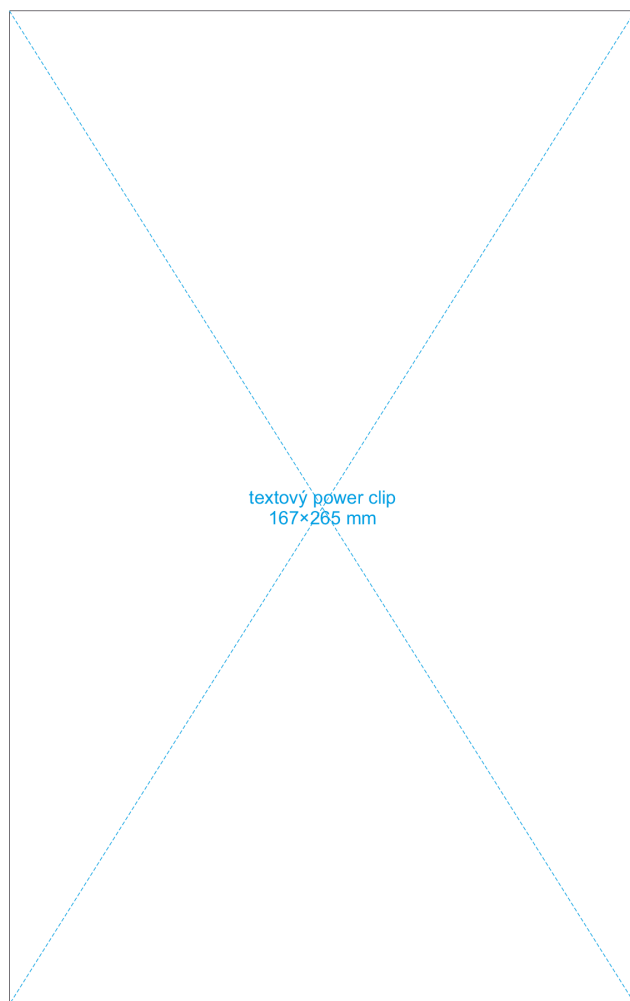
Obrázek A.7: Rozměry 1/2 elementu s textem



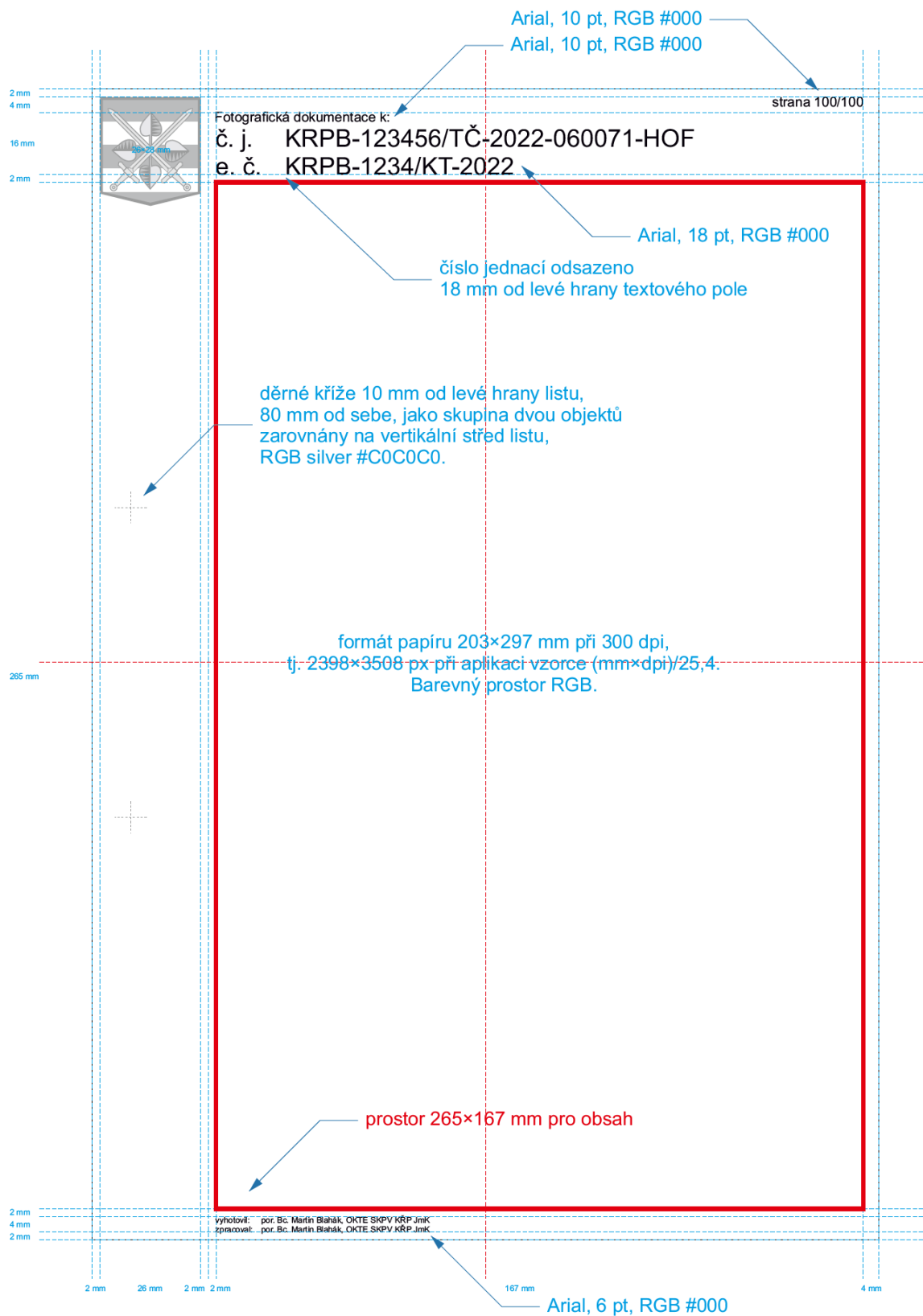
Obrázek A.8: Rozměry 1/2 elementu s textem



Obrázek A.9: Rozměry 1/3 elementu s textem



Obrázek A.10: Rozměry 1/1 textového elementu

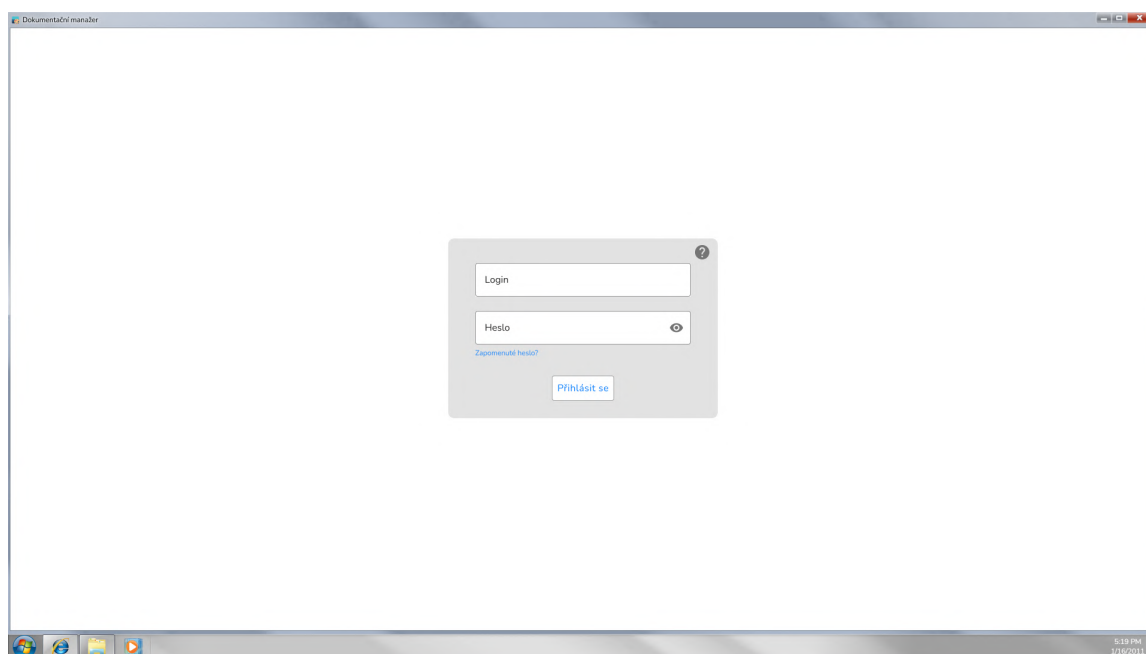


Obrázek A.11: Rozměry listu

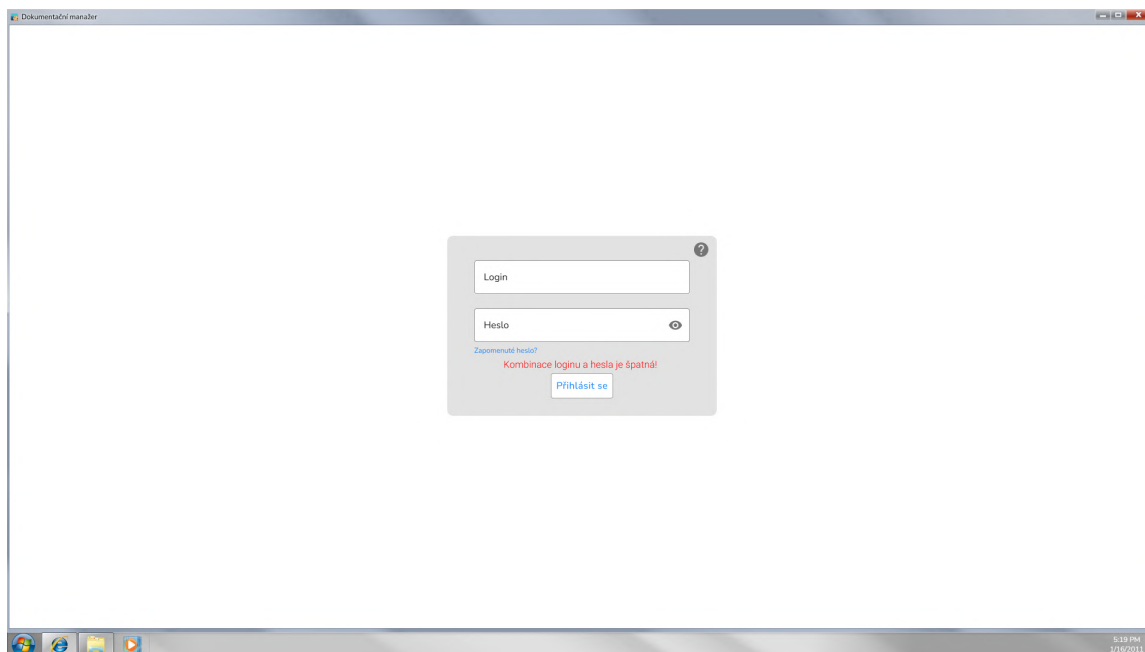
## Příloha B

# Navrhovaný vzhled obrazovek

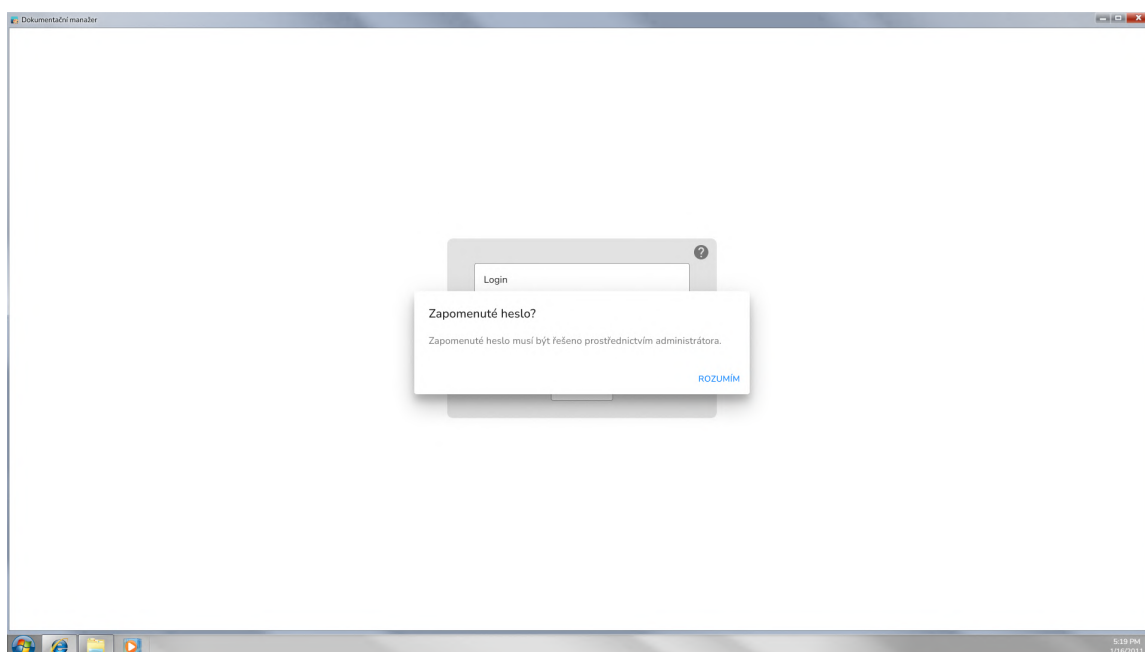
### B.1 Přihlašovací obrazovka



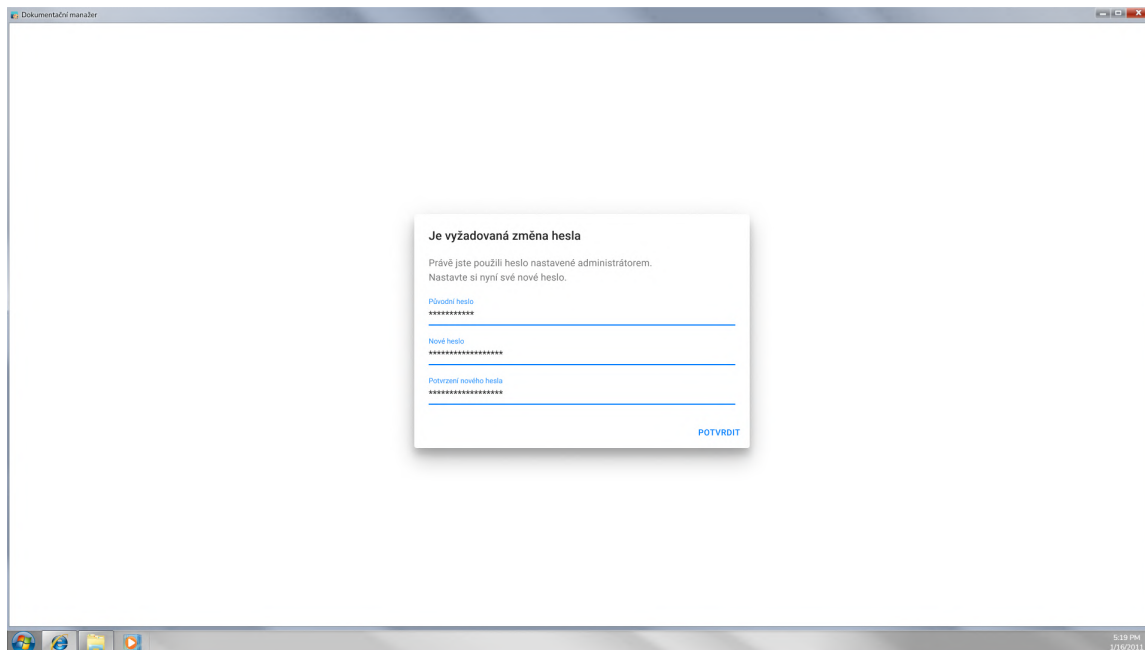
Obrázek B.1: Přihlašovací obrazovka



Obrázek B.2: Přihlašovací obrazovka se špatným heslem

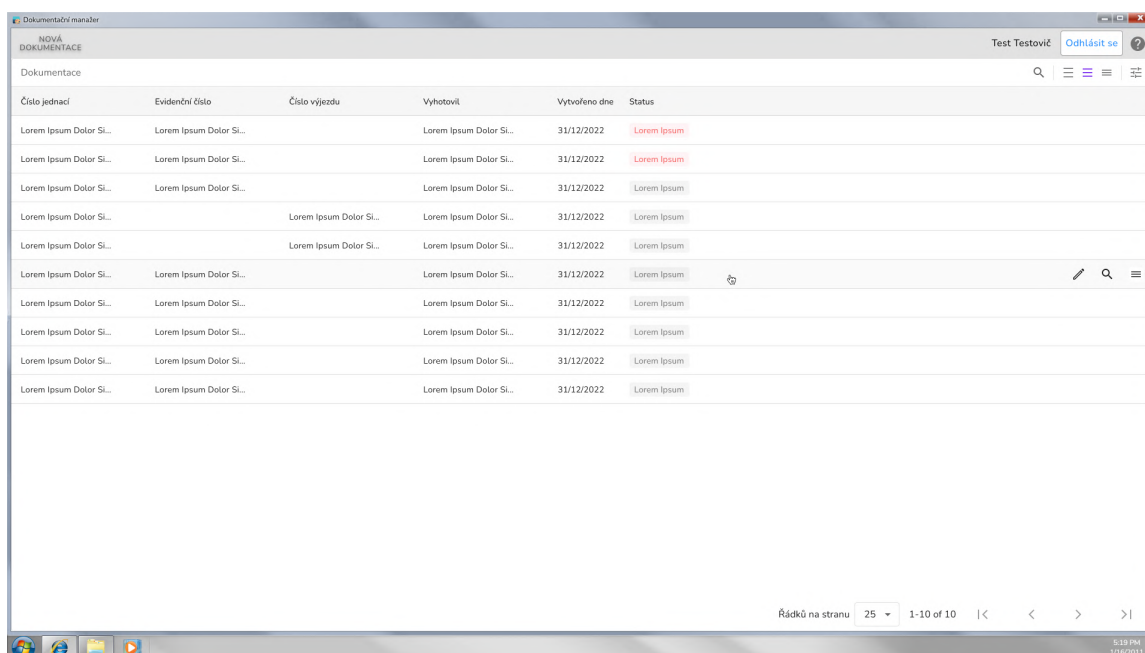


Obrázek B.3: Přihlašovací obrazovka - zapomenuté heslo

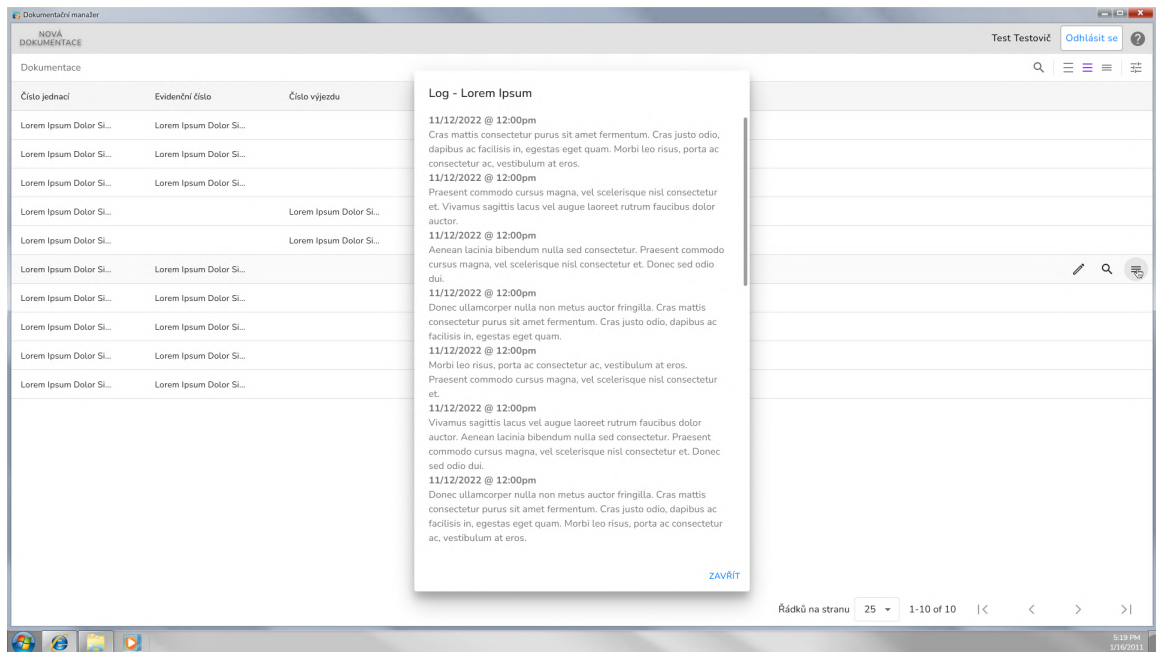


Obrázek B.4: Přihlašovací obrazovka - nastavení nového hesla

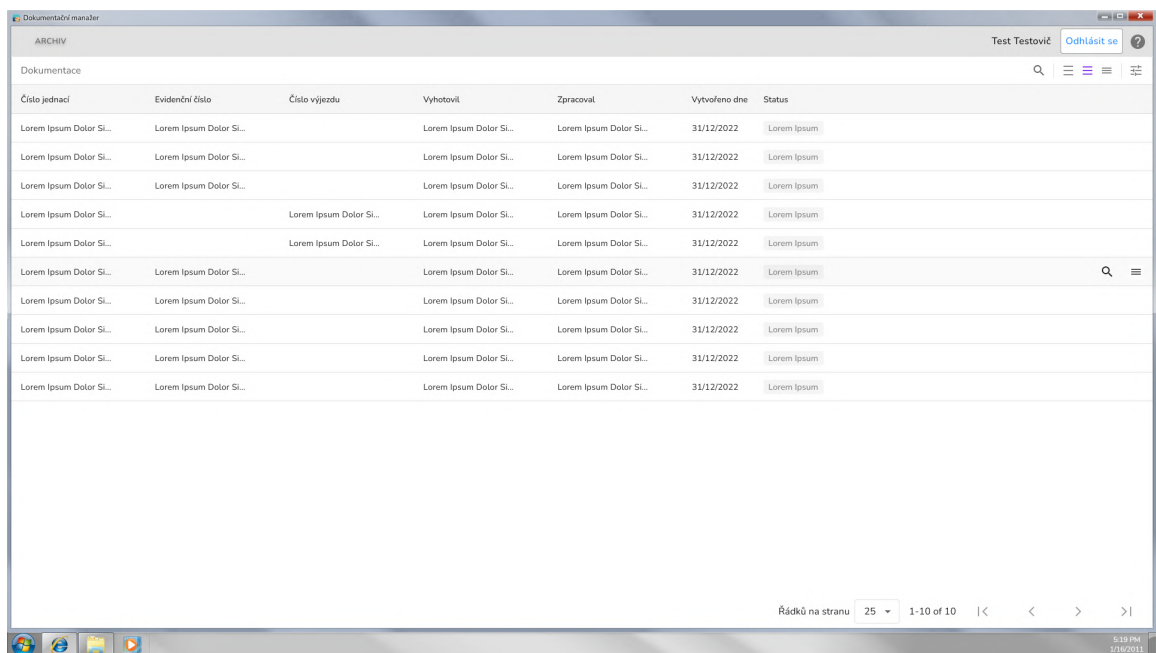
## B.2 Přehled aktivních dokumentací



Obrázek B.5: Pohled uživatele na přehled dokumentací

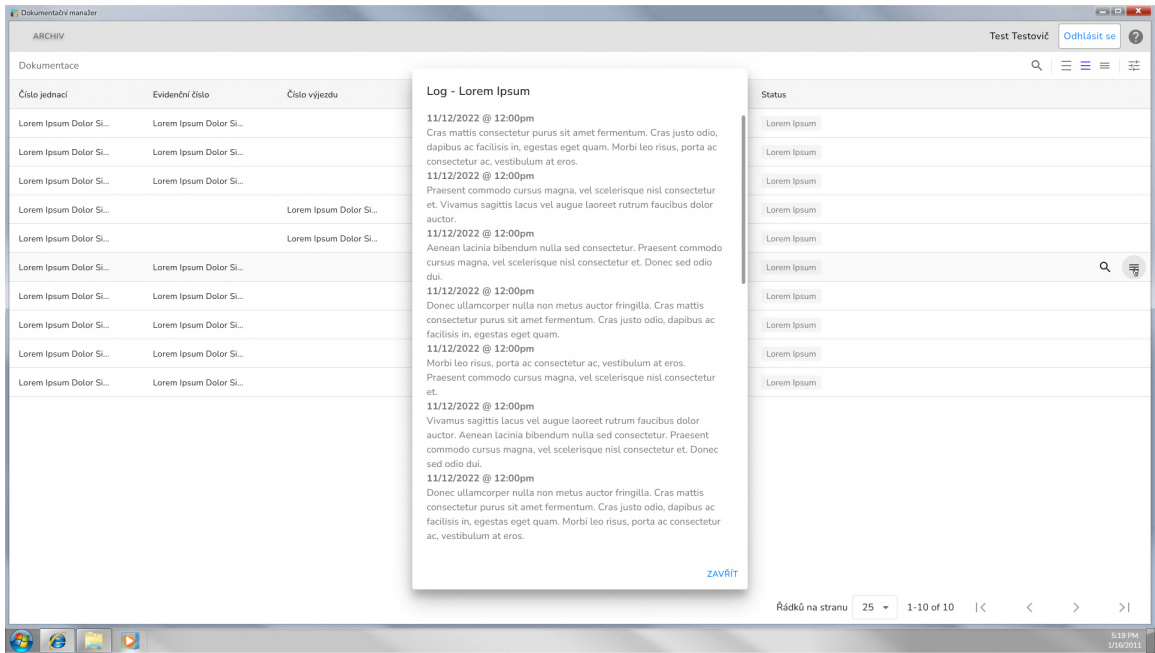


Obrázek B.6: Pohled uživatele na přehled dokumentací - log

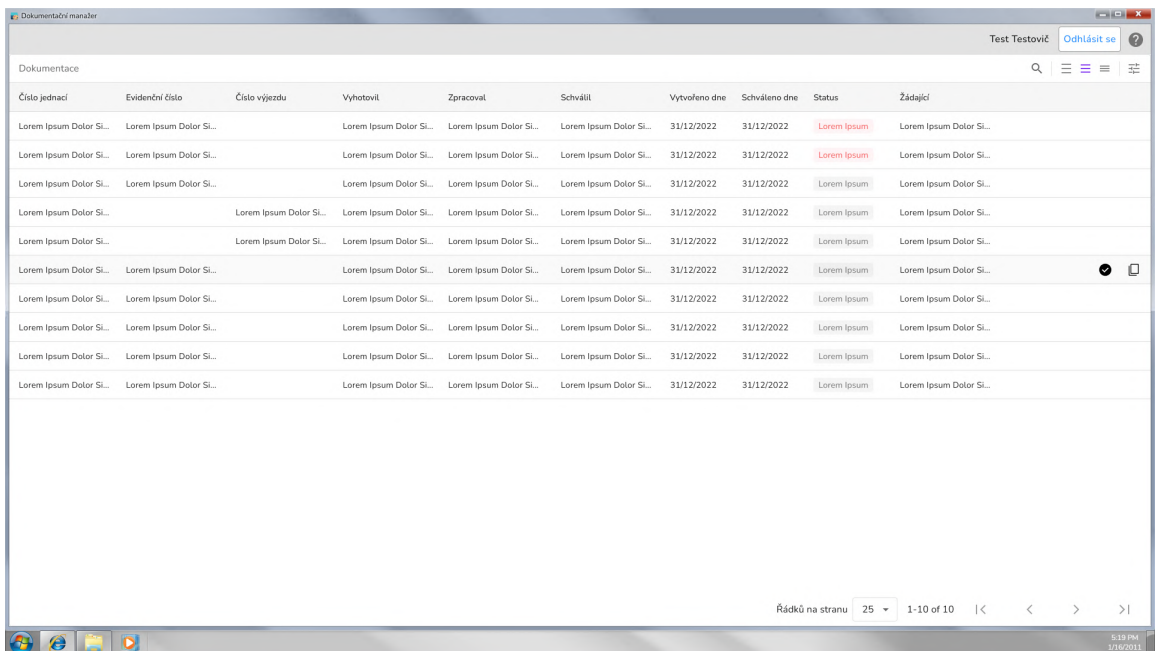


Obrázek B.7: Pohled vedoucího na přehled dokumentací

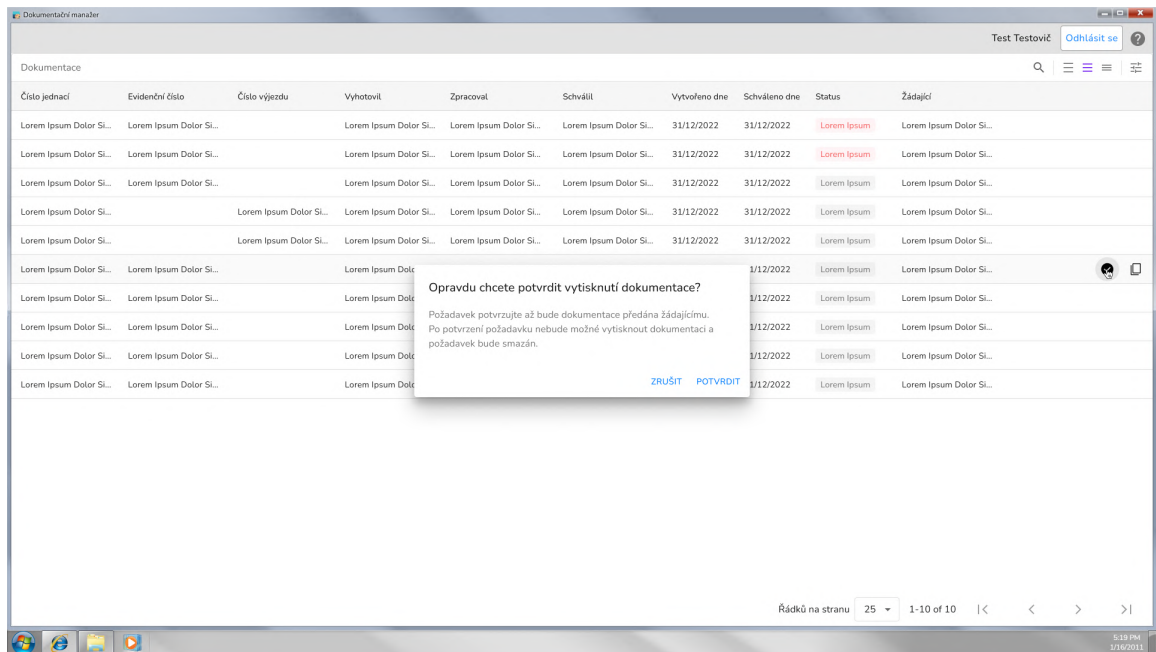




Obrázek B.8: Pohled vedoucího na přehled dokumentací - log

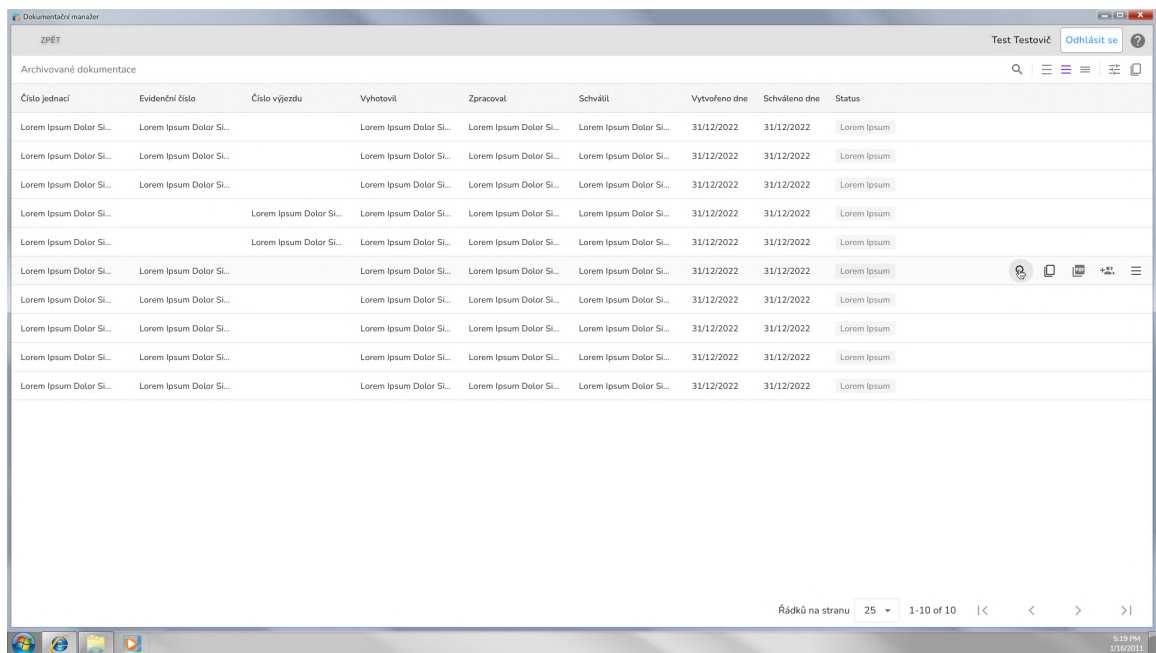


Obrázek B.9: Pohled tisku na přehled dokumentací

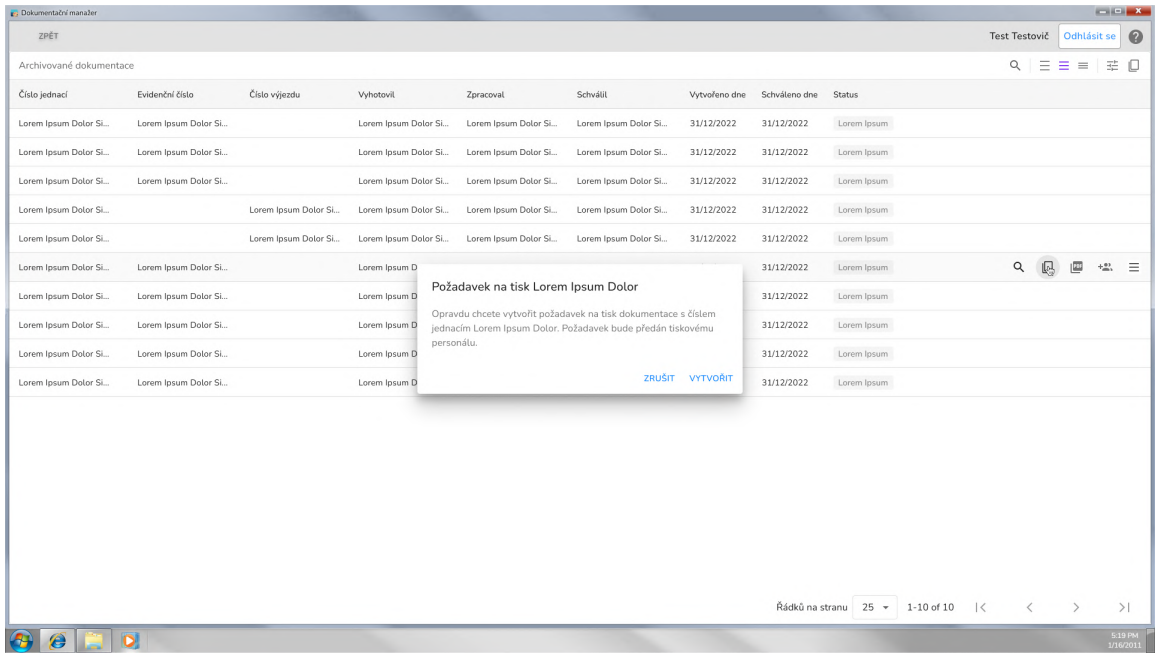


Obrázek B.10: Pohled tisku na přehled dokumentací - potvrzení vytisknutí

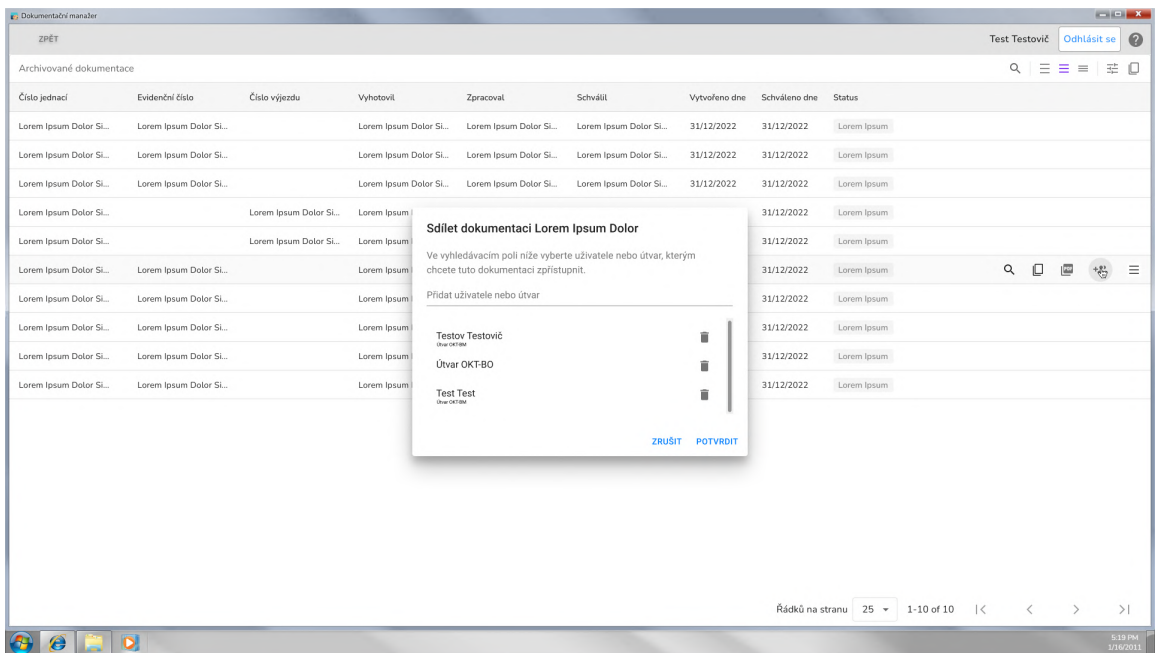
### B.3 Přehled archivovaných dokumentací



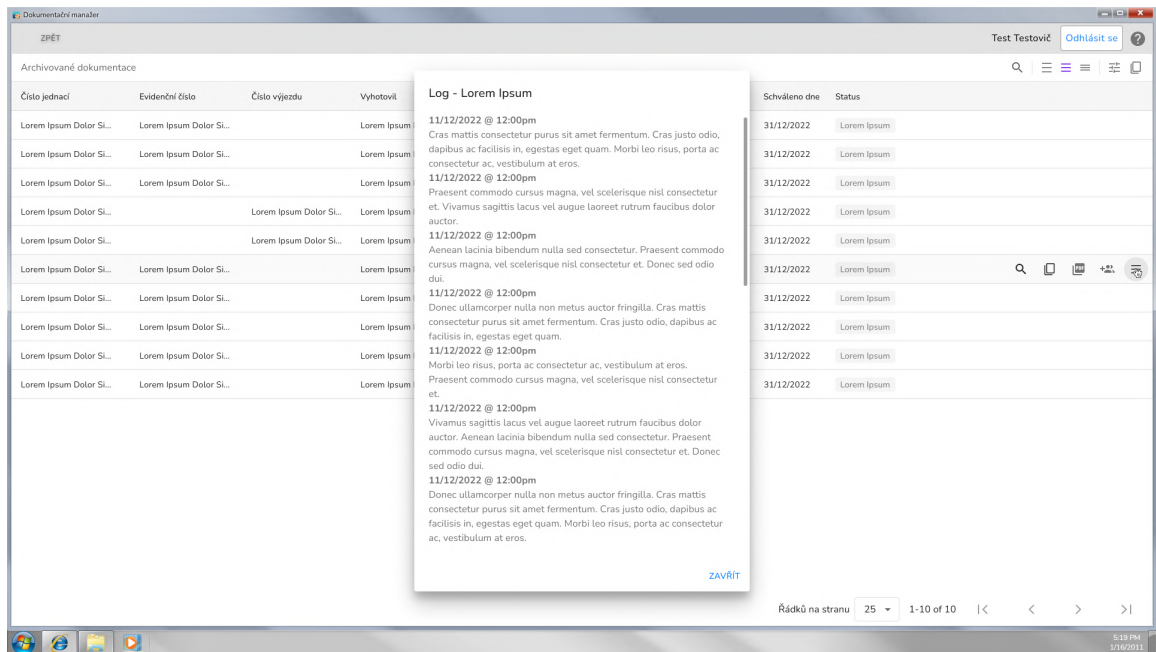
Obrázek B.11: Přehled archivovaných dokumentací



Obrázek B.12: Přehled archivovaných dokumentací - požadavek tisku

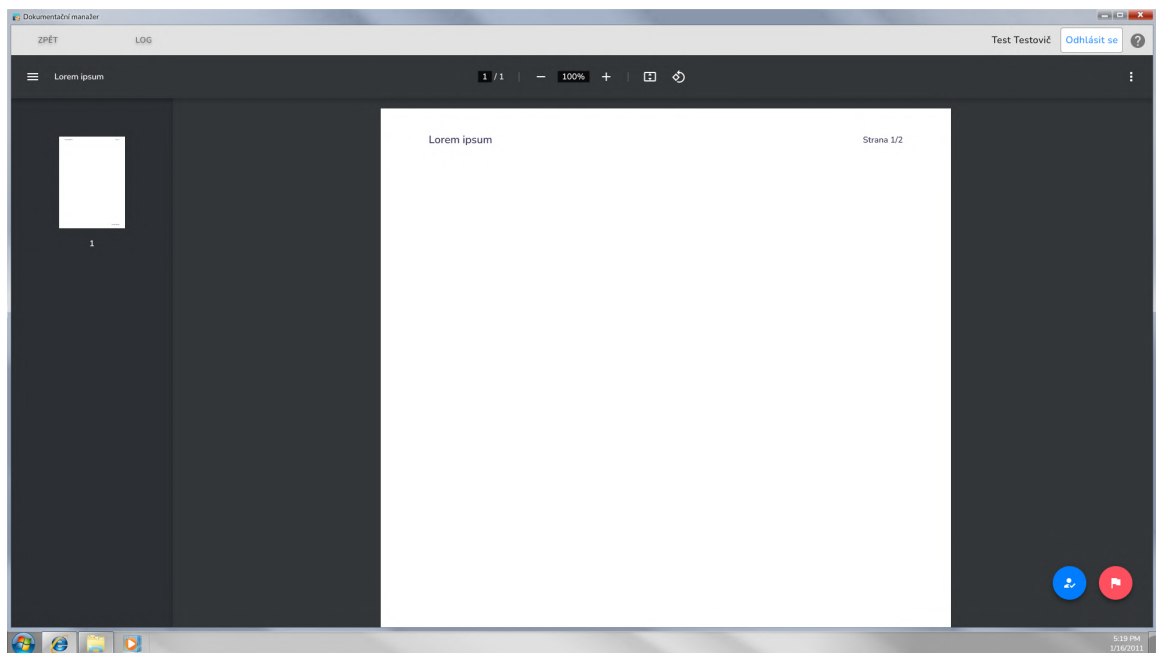


Obrázek B.13: Přehled archivovaných dokumentací - sdílení

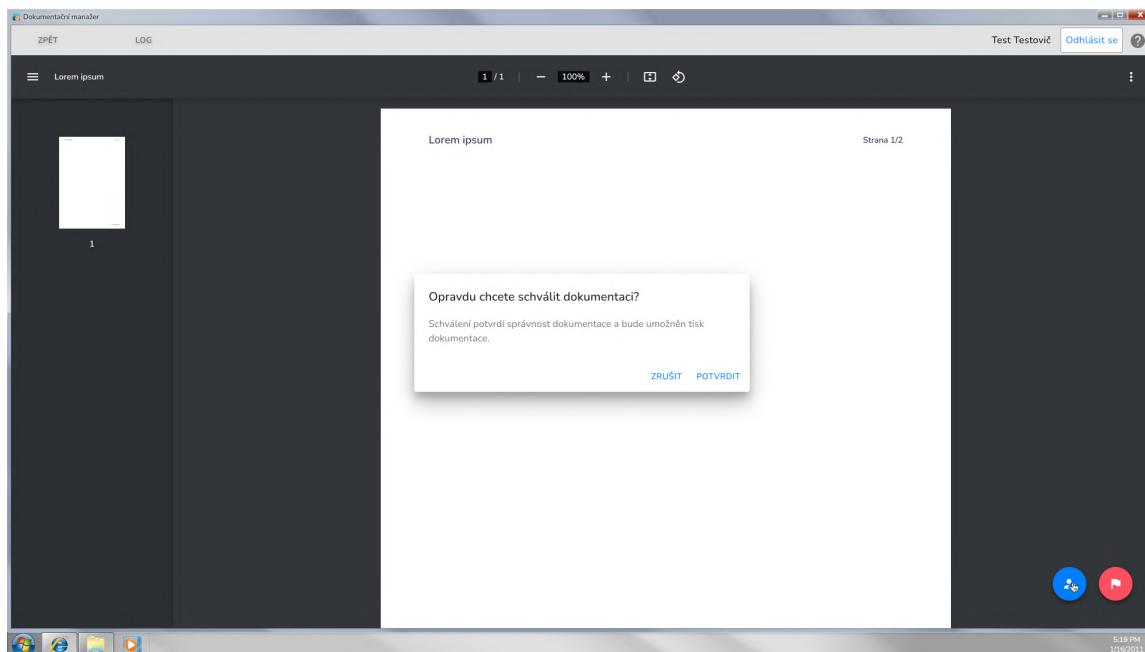


Obrázek B.14: Přehled archivovaných dokumentací - log

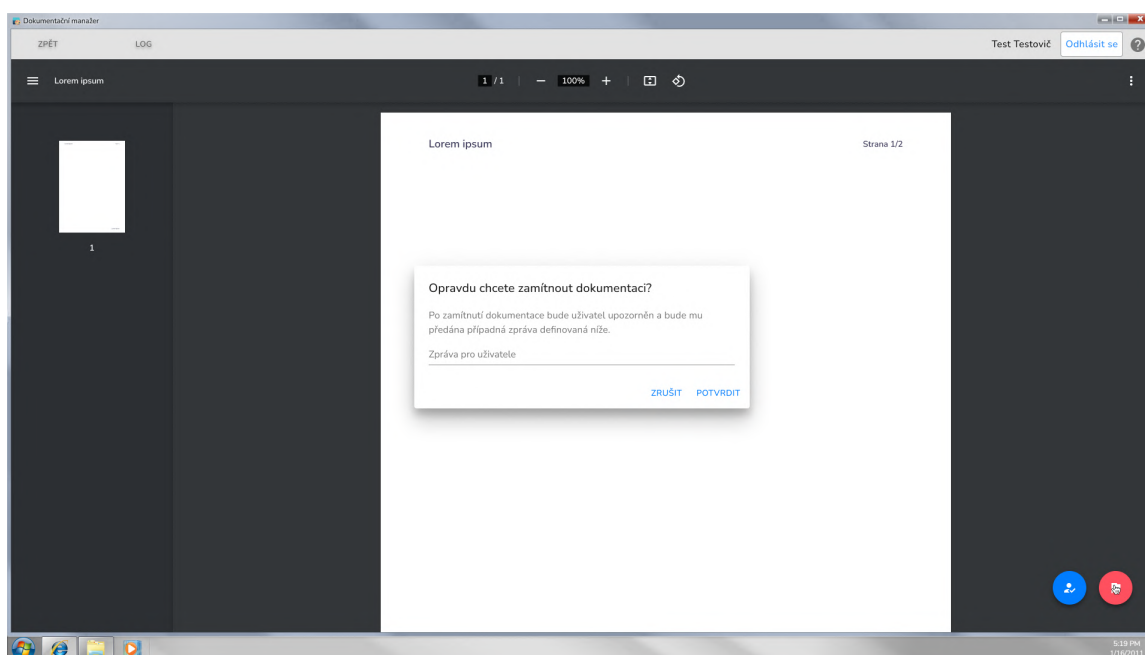
## B.4 Náhled dokumentace



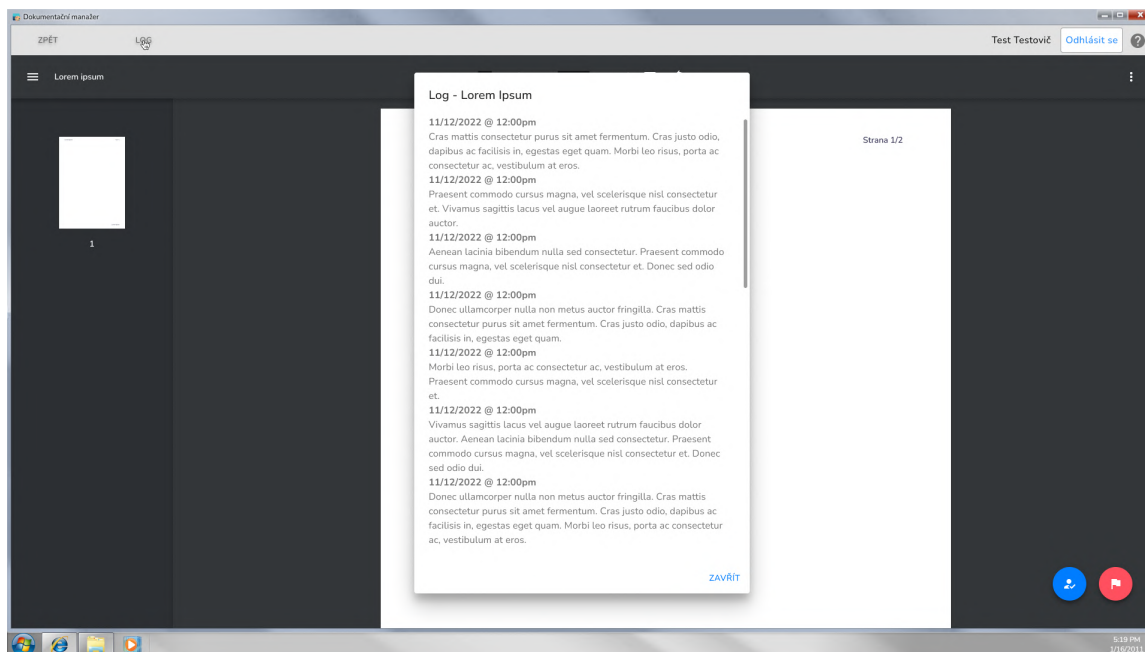
Obrázek B.15: Náhled dokumentace



Obrázek B.16: Náhled dokumentace - schválení

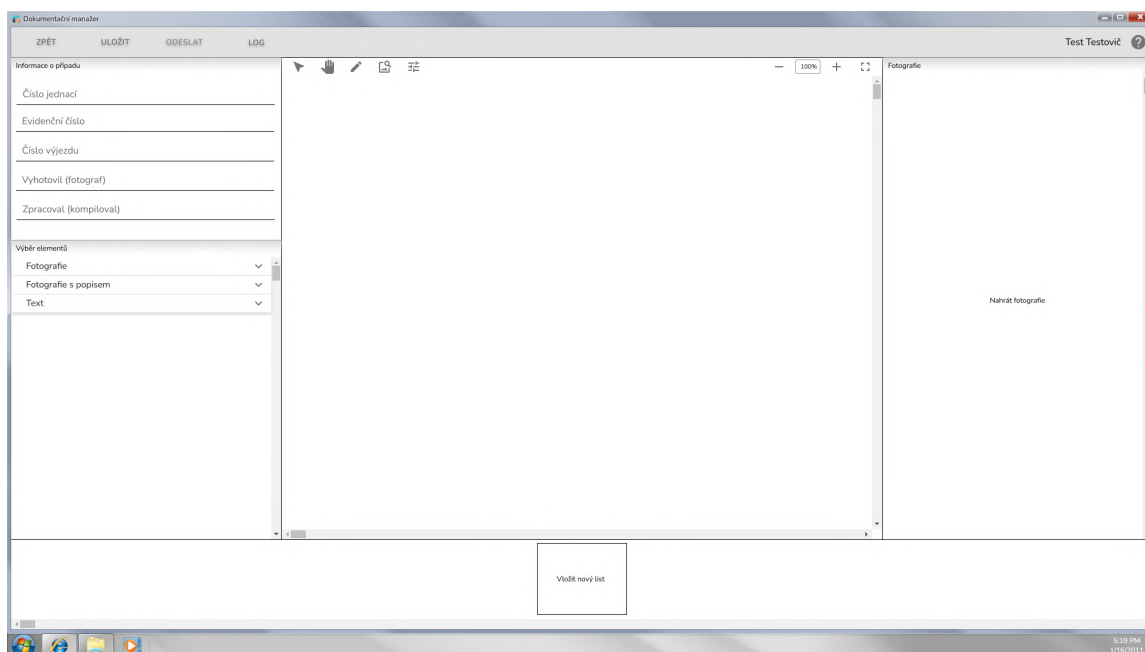


Obrázek B.17: Náhled dokumentace - zamítnutí

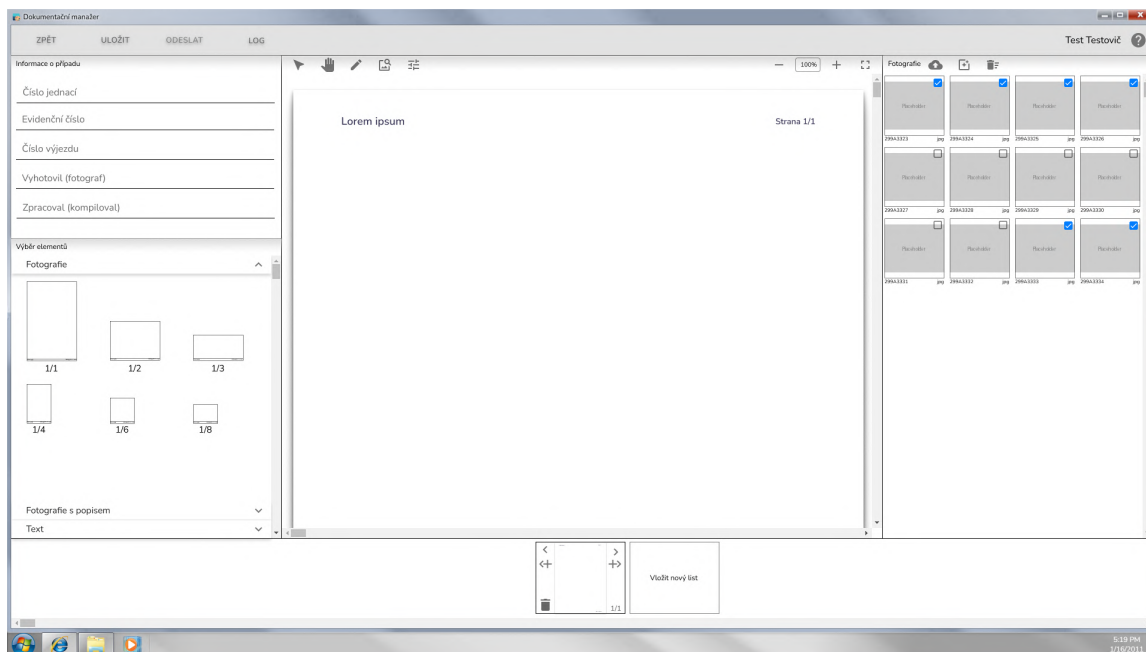


Obrázek B.18: Náhled dokumentace - log

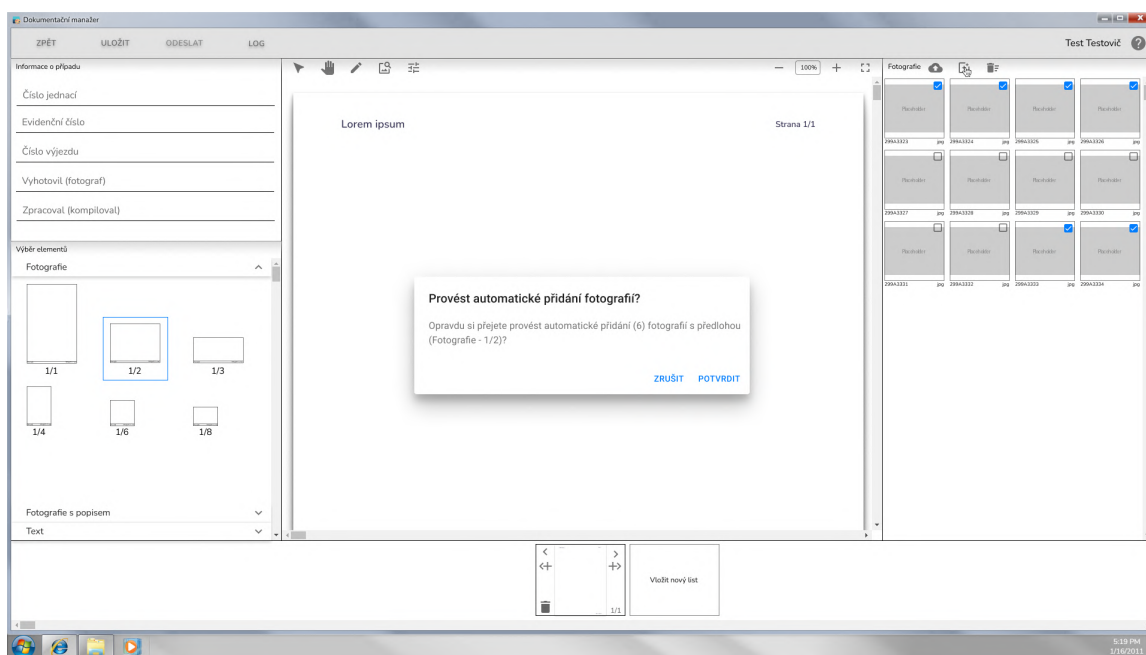
## B.5 Editor dokumentace



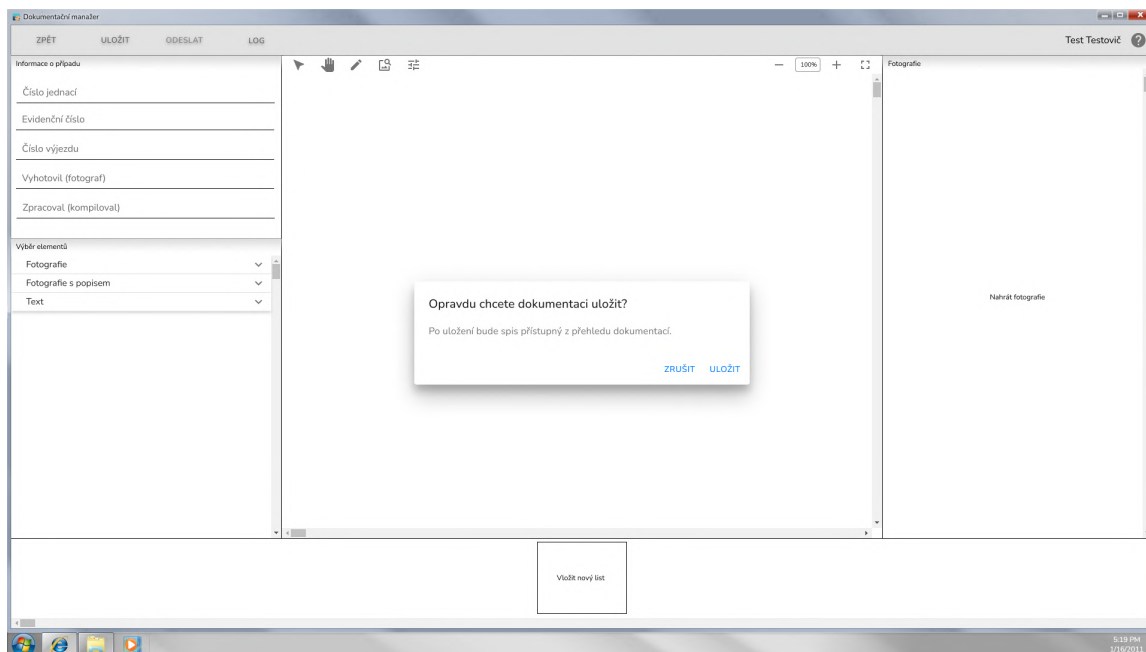
Obrázek B.19: Editor dokumentace



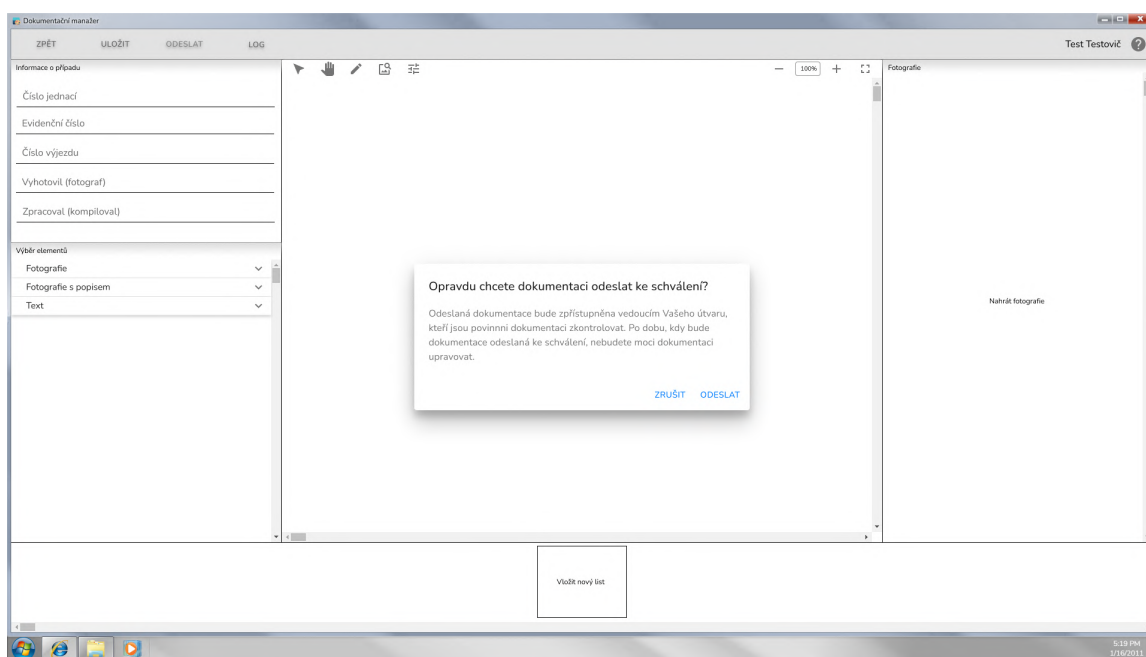
Obrázek B.20: Editor dokumentace - vložený list a fotografie



Obrázek B.21: Náhled dokumentace - automatické přidání



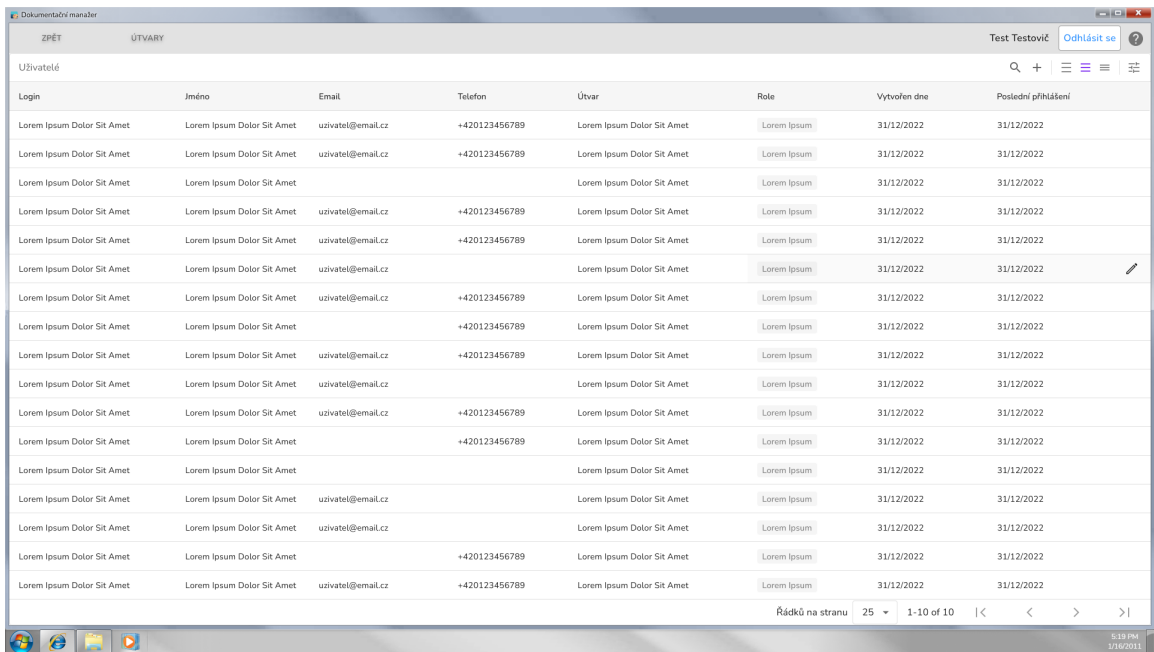
Obrázek B.22: Editor dokumentace - uložení



Obrázek B.23: Náhled dokumentace - odeslání ke schválení



## B.6 Správa uživatelů

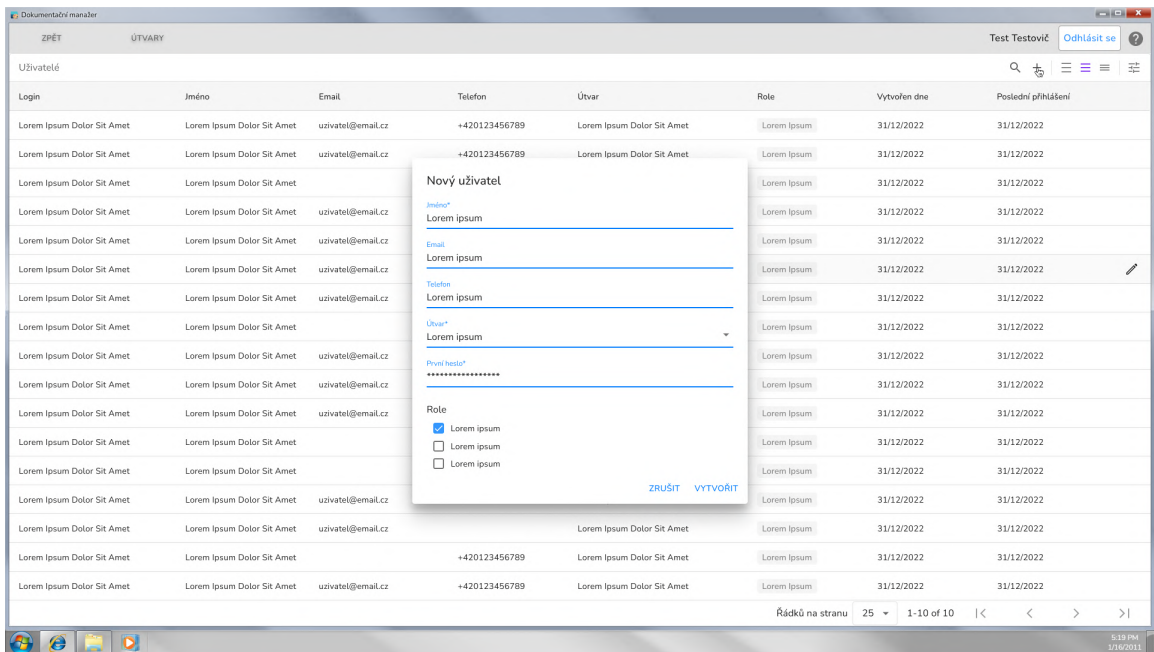


The screenshot shows a web application window titled 'Dokumentární manažer'. The main content area displays a table of users. The table has the following columns: Login, Jméno, Email, Telefon, Úvar, Role, Vytvořen dne, and Poslední přihlášení. The data is as follows:

| Login                      | Jméno                      | Email             | Telefon       | Úvar                       | Role        | Vytvořen dne | Poslední přihlášení |
|----------------------------|----------------------------|-------------------|---------------|----------------------------|-------------|--------------|---------------------|
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |
| Lorem Ipsum Dolor Sit Amet | Lorem Ipsum Dolor Sit Amet | uzivatel@email.cz | +420123456789 | Lorem Ipsum Dolor Sit Amet | Lorem Ipsum | 31/12/2022   | 31/12/2022          |

At the bottom of the table, there is a pagination control: 'Řádků na stranu 25', '1-10 of 10', and navigation arrows. The system tray shows the time '5:18 PM' and date '1/16/2011'.

Obrázek B.24: Přehled uživatelů

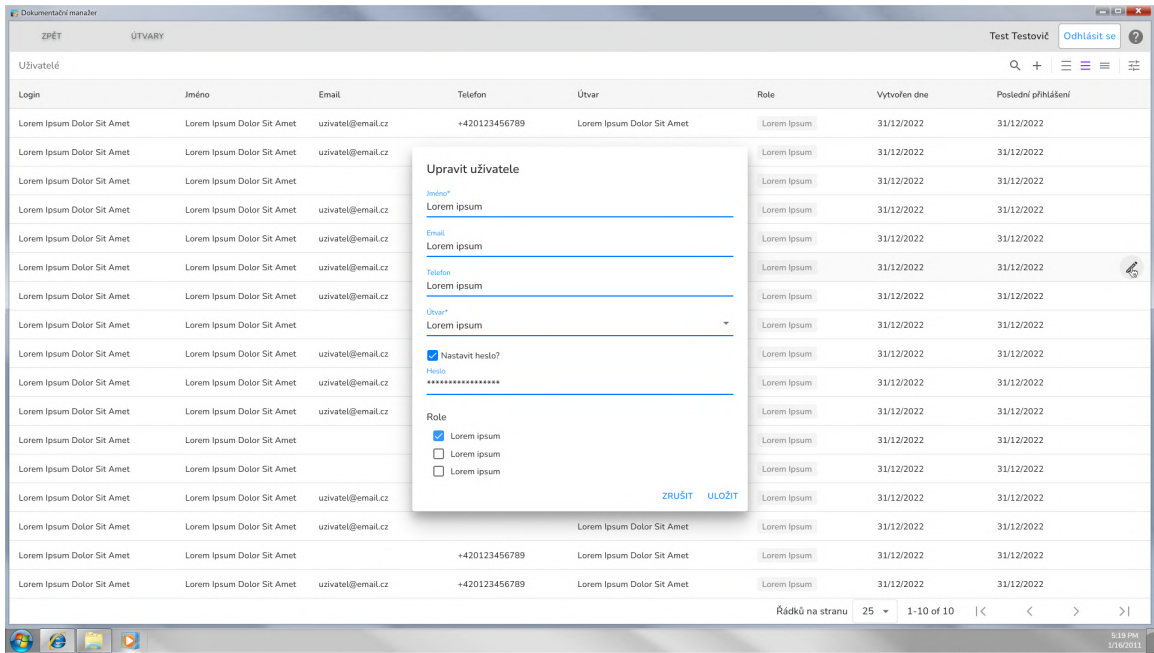


The screenshot shows the same 'Uživatelé' management interface as in B.24, but with a 'Nový uživatel' dialog box open. The dialog box contains the following fields and options:

- Jméno\*: Lorem ipsum
- Email: Lorem ipsum
- Telefon: Lorem ipsum
- Úvar\*: Lorem ipsum
- První heslo\*: \*\*\*\*\*
- Role:  Lorem ipsum,  Lorem ipsum,  Lorem ipsum

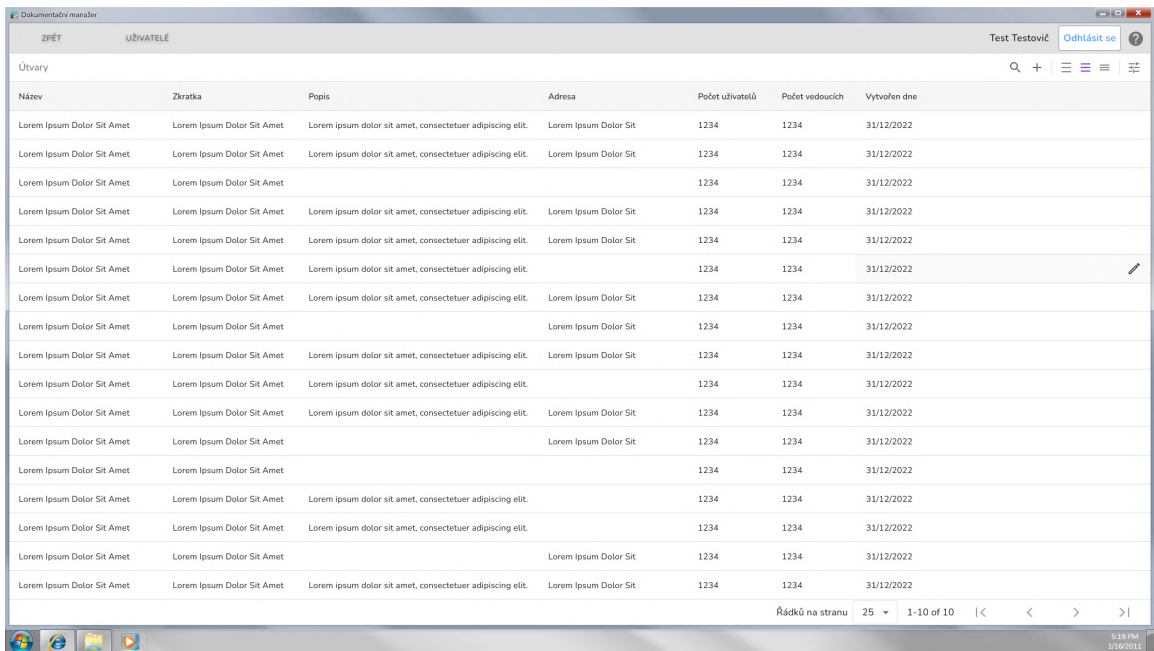
At the bottom of the dialog box, there are two buttons: 'ZRUŠIT' and 'VYTVOŘIT'. The background table is partially obscured by the dialog box.

Obrázek B.25: Přehled uživatelů - vytvoření

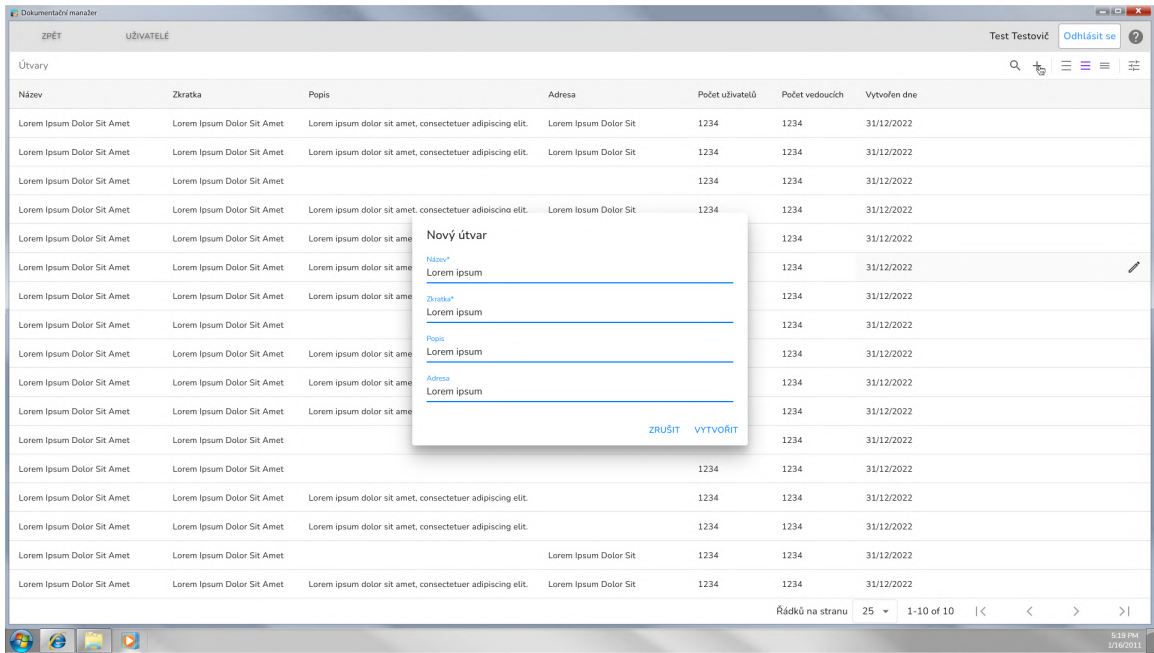


Obrázek B.26: Přehled uživatelů - upravení

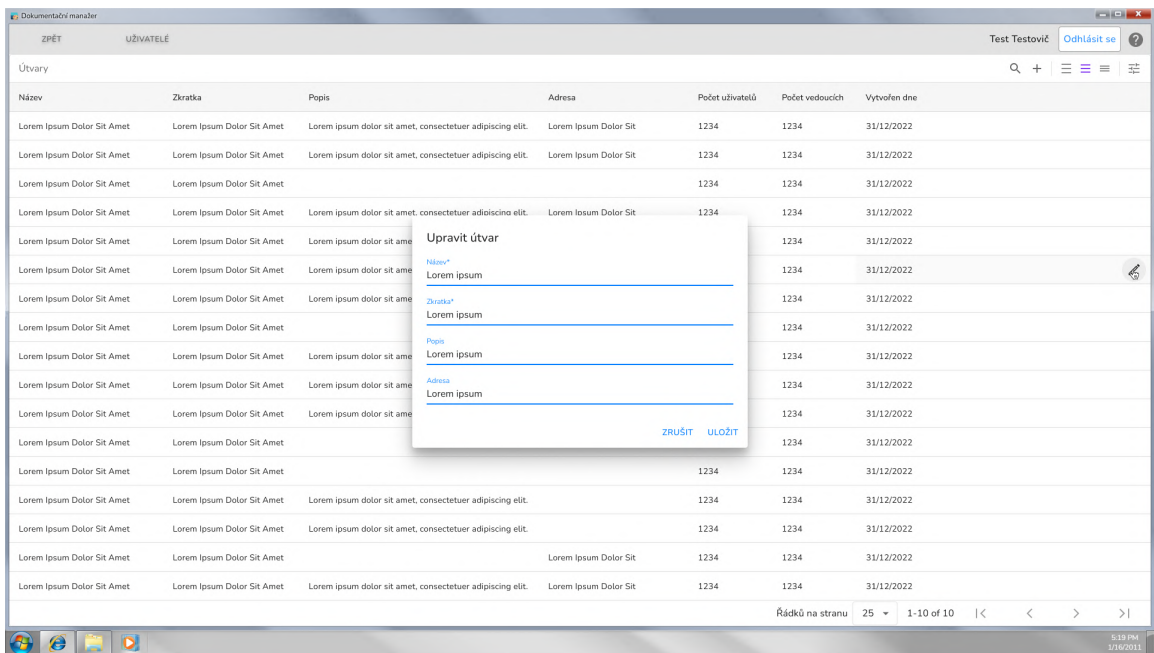
## B.7 Správa útvarů



Obrázek B.27: Přehled útvarů



Obrázek B.28: Přehled útvarů - vytvoření



Obrázek B.29: Přehled útvarů - upravení

## Příloha C

# Hierarchie odevzdaného média

- /Aplikace
  - \* Obsahuje spustitelný soubor klientské aplikace.
  - /Zdrojové kódy
    - \* Obsahuje zdrojové soubory aplikace a soubor README s příkazy pro přeložení aplikace.
- /Detekční model
  - /Dataset
    - \* Obsahuje dataset složený z fotografií a vytvořených anotací.
  - YOLOv7\_training.ipynb
- /Server
  - /Instalace
    - \* Obsahuje soubor instalace.pdf, ve kterém je popsána instalace serveru.
  - /Zdrojové kódy
    - \* Obsahuje zdrojové soubory serverové implementace.
- /Návrh
  - /obrazovky
    - \* Obsahuje všechny navržené obrazovky aplikace.
  - dp\_diagramy\_xterbr00.vpp
  - er\_diagram.pdf
  - prijimaci\_testy.pdf
  - sekvencni\_diagramy.pdf
  - specifikace\_use\_case.pdf
  - stavovy\_diagram\_obrazovek.pdf
  - use\_case\_diagram.pdf
- /Text práce
  - \* Obsahuje všechny soubory pro vytvoření textu práce.