

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra kvantitativních metod



Univerzita Hradec Králové
Fakulta informatiky a managementu

Aplikace pro podporu klasifikace medicínských dat
Bakalářská práce

Autor: Michal Vaďák
Studijní obor: Aplikovaná informatika

Vedoucí práce: doc. Ing. Filip Malý Ph.D.

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a čerpal jsem ze zdrojů uvedených v seznamu použité literatury.

V Pardubicích dne

Michal Vad'ák

Poděkování

Rád bych touto cestou vyjádřil poděkování panu doc. Ing. Filipovi Malému Ph.D. za metodické vedení, podnětné rady a odbornou pomoc při vypracování této bakalářské práce.

Anotace

Cílem bakalářské práce je v rámci projektu IT4NEURO popsat webovou aplikaci Endo-neuro-analytics a její rozšíření pro podporu klasifikace medicínských dat z endoskopické kapsle. Práce popisuje aktuální architekturu, použité technologie a využití aplikace. Dále čtenáře seznámí s novým rozšířením pro podporu klasifikace medicínských dat a princip jeho fungování. Detailně popisuje nově přidanou samostatnou aplikaci a rozšíření těch původních.

Klíčová slova

Endoskopie, endokapsle, kapsle endoskopická, Java, Javascript, Spring Boot, Spring Security, React, REST API

Title: Application to support the classification of medical data

Annotation

The aim of the bachelor's thesis is to describe the web application Endo-neuro-analytics and its extension to support the classification of medical data from the endoscopic capsule within the IT4NEURO project. The thesis describes the current architecture, used technologies and application usage. It will also introduce readers to the new extension to support the classification of medical data and the principle of its operation. It describes in detail the newly added standalone application and the extension of the original ones.

Key words

Endoscopy, endocapsule, capsule endoscopic, Java, Javascript, Spring Boot, Spring Security, React, REST API

Obsah

1	Úvod	1
2	Endo-neuro-analytics	2
2.1	Představení.....	2
2.2	Použité technologie.....	2
2.2.1	React	2
2.2.2	Spring Boot.....	4
2.3	Architektura aplikace.....	5
2.3.1	Analysis-management	5
2.3.2	Gui	6
2.3.3	User-management.....	6
2.3.4	Gateway	7
2.3.5	Image-analysis	7
2.3.6	Neural-network-mock.....	7
3	Rozšíření webové aplikace	8
3.1	Představení.....	8
3.2	Stavy	12
3.3	Rozšíření architektury aplikace	12
3.3.1	Manual-image-classifier	13
3.3.2	Analysis-management	15
3.3.3	Gui	16
3.3.4	User-management.....	23
3.3.5	Gateway	24
4	Shrnutí výsledků.....	25
5	Závěr.....	26
6	Seznam obrázků.....	27
7	Seznam zdrojových kódů.....	28
8	Literatura	29
9	Přílohy	1
9.1	Zprovoznění aplikace na lokálním zařízení.....	1
9.1.1	Manual-image-classifier	1

1 Úvod

Bakalářská práce je rozdělena do dvou částí pro představení původní aplikace a rozšíření pro klasifikaci medicínských dat. Představení původní aplikace popisuje základní funkcionality, její mikroservisy a technologie použité při vývoji. Rozšíření dále představuje proces manuální klasifikace, úpravu původních mikroservis a princip funkčnosti nově vytvořené mikroservisy.

Aplikace bude poskytovat možnost manuálního hodnocení jednotlivých snímků. Klasifikované obrázky roztřídí do složek dle názvu nálezů na snímku, který u nich lékař označil. Složky spolu s informacemi o nastalých chybách a komentářích zabalí do ZIP souboru a ten uloží do databáze.

Údaje o klasifikacích budou dostupné mimo jiné pro prohlížení či stažení a následné učení neuronové sítě, díky čemuž lékaři ušetří čas i finance.

2 Endo-neuro-analytics

V této části si představíme aktuální webovou aplikaci a popíšeme si její použití.

2.1 Představení

Webová aplikace Endo-neuro-analytics pro práci se snímky pořízené endokapslí je vyvinutá pro lékaře, jejichž práce souvisí s oborem endoskopie. V aplikaci je uživateli přidělena role správce nebo doktora. Uživatel s rolí doktor může spravovat údaje svého účtu, svých pacientů a jejich analýz. Správce má stejná oprávnění jako doktor, navíc má oprávnění pro správu aplikace a všech údajů. Při praktickém použití si lékař vytvoří záznam o svém pacientovi, který má podstoupit vyšetření endoskopickou kapslí, a vyplní jeho údaje. Pacient při vyšetření spolkně kapsli, která nahrává celý svůj průchod trávícím traktem a snímky odesílá do záznamového zařízení. Ty jsou ze záznamového zařízení přeneseny do počítače. Lékař následně vytvoří pacientovi analýzu a v případě potřeby doplní dodatečné informace. K analýze je potřeba nahrát snímky získané z endoskopického vyšetření, které se nacházejí v souboru s příponou gvf. Jedná se o komprimovaný soubor obsahující veškeré obrázky pořízené endoskopickou kapslí při vyšetření. Při nahrávání do aplikace jsou postupně zpracovány a pro každý z nich je vytvořen vlastní záznam obsahující informace o pořadí, datu vytvoření, ke které analýze se vztahuje a další. Tyto záznamy jsou spolu s obrázky uloženy do databáze [2].

Po úspěšném zpracování je lékaři, který soubor nahrál, zasláno oznámení oznamující úspěšné zpracování nahraného souboru. To obsahuje počet zpracovaných snímků a odkazy na stránky, na kterých je možné jejich zobrazení. Lékař má následně možnost spustit analýzu, která postupně odešle obrázky neuronové síti na zpracování a výsledek uloží do databáze. Ten je možné zobrazit na stránce s obrázky [2].

2.2 Použité technologie

V této kapitole si představíme technologie použité v aplikaci a jejich výhody.

2.2.1 React

Pro **Gui** aplikaci je využita technologie React [1]. React (také známý jako React.js) je javascriptová open-source front-endová knihovna a slouží k tvorbě uživatelského rozhraní.

Na vývoji se podílí Facebook a komunita vývojářů. React je optimální pro vytváření „single-page“ aplikací, které mění zobrazení dat na stránce bez toho, aniž by bylo nutné obnovit stránku. Primární výhody této knihovny jsou rychlost, jednoduchost a možnost vytváření znovupoužitelných komponent. Tyto komponenty pak můžeme využít napříč celým projektem, aniž bychom museli psát celý kód znovu.

Čím větší je aplikace, tím více dat se nám při prvotním načtení stránky musí stáhnout. To může mít za důsledek pomalejší stažení. React ale zároveň umožňuje takzvaný „Code splitting“. Při využívání této funkce se aplikace rozděluje do jednotlivých částí, které není potřeba stáhnout při prvotním zobrazení, pokud se zrovna na stránce nenachází. To je možné provést až v moment, kdy uživatel danou stránku navštíví. Tím zajistíme rychlejší prvotní načtení stránky.

Knihovna využívá tzv. JSX, také známé jako Javascript XML, které mu dovoluje strukturování komponent podle syntaxe podobné HTML. Toto má za výsledek jednodušší a rychlejší učení knihovny a celkovou implementaci na projektu. Samozřejmostí je ovšem předpoklad, že daný vývojář má už zkušenosti se základy HTML. Bez využití JSX bychom kód v Reactu museli psát následovně.

```
1 export const Komponenta = () => {
2   return createElement('div', null, 'Toto je komponenta')
3 }
```

Zdrojový kód 1: React komponenta bez použití JSX (zdroj: autor)

Při psaní webové aplikace v Reactu bychom měli správně rozkouskovat jednotlivé prvky do takzvaných komponent. Ty můžou být definovány mnoha způsoby. Starší psaní komponent v Reactu probíhalo za použití tříd (anglicky „class“), které lze použít i dnes.

```
1 class Komponenta extends Component {
2   render() {
3     return (
4       <div>
5         <h1>Toto je komponenta</h1>
6       </div>
7     )
8   }
9 }
```

Zdrojový kód 2: React komponenta s použitím třídy (zdroj: autor)

Novější psaní komponent nabízí použití takzvaných funkcionálních komponent. Ty lze deklarovat dvěma způsoby. První způsob je za použití konstanty. Druhý způsob je za použití funkce.

```
1 export const Komponenta = () => {
2   return (
3     <div>
4       <h1>Toto je komponenta</h1>
5     </div>
6   )
7 }
```

Zdrojový kód 3: React funkční komponenta za použití konstanty (zdroj: autor)

```
1 function Komponenta() {
2   return (
3     <div>
4       <h1>Toto je komponenta</h1>
5     </div>
6   )
7 }
```

Zdrojový kód 4: React funkční komponenta za použití funkce (zdroj: autor)

Zároveň je do sebe můžeme dále skládat a vytvořit tím složitější komponenty. Skládáním komponent můžeme předejít zbytečné složitosti a nečitelnosti kódu.

```
1 export const Aplikace = () => {
2   return (
3     <>
4       <Menu/>
5       <Obsah/>
6       <Pata/>
7     </>
8   )
9 }
```

Zdrojový kód 5: React složená komponenta (zdroj: autor)

2.2.2 Spring Boot

Spring Boot [3] je open-source framework pro vývoj aplikací v programovacím jazyce Java, který se používá pro tvorbu mikroservis. Výhoda Spring Bootu je v rychlém založení a zprovoznění aplikace. Také poskytuje správnou konfiguraci zabezpečení.

Spring Boot je použit na psaní back-endových mikroservis. Velice příhodná je automatická konfigurace aplikace a také to, že je schopný ji měnit podle závislostí. Vzhledem k tomu, jaký databázový jazyk použijeme, nám je schopný nakonfigurovat

potřebný konektor. Další výhodou je to, že v sobě obsahuje integrovaný Apache Tomcat, který poskytuje okamžité spuštění aplikace bez toho, abychom museli aplikaci nasazovat na webový server.

Mikroservis služby dále používají Spring Security [4], což ovlivňuje zabezpečení aplikace tak, že kontroluje práva uživatele, zda má pravomoc volat danou URL adresu.

2.3 Architektura aplikace

2.3.1 Analysis-management

Tato aplikace je napsána ve Spring Bootu. Poskytuje správu nad analýzami pacientů a uloženými snímky. Všechny tyto údaje jsou uchovávány v databázi, která je postavena na MongoDB [5].

Aplikace poskytuje REST API metody pro vytváření a správu analýz pacientů. Tyto metody jsou určeny pro správce a lékaře, aby byli schopni vytvořit analýzu, zobrazit si její detail, upravit její údaje, označit ji jako smazanou nebo aktivní či si načíst seznam všech dostupných analýz daného pacienta. Metoda pro seznam také poskytuje možnost volby pro filtrování, řazení a počet záznamů na stránku.

Analysis-management také obstarává správné fungování analýzy neuronovou sítí. Při zahájení nastaví stav na **ANALYSIS_IN_PROGRESS** a odešle na **Image-analysis** požadavek spolu s unikátním ID. Ta obstará celý proces vyhodnocení neuronovou sítí. Následně po dokončení přijme požadavek, jenž nastaví stav analýzy na **ANALYSIS_COMPLETED** a odešle uživateli, který ji spustil, oznámení o jejím úspěšném / neúspěšném dokončení.

2.3.1.1 Stav

Možné stavy analýzy, které mohou nastat při analýze obrázků neuronovou sítí.

CREATED

Výchozí stav, který se nastaví při vytvoření analýzy.

ANALYSIS_IN_PROGRESS

Stav reprezentující aktuálně probíhající vyhodnocování obrázků analýzy neuronovou sítí.

ANALYSIS_PAUSED

Stav reprezentující aktuální proces pozastaveného vyhodnocování neuronovou sítí u analýzy.

ANALYSIS_COMPLETED

Stav reprezentující dokončený proces vyhodnocování neuronovou sítí u analýzy.

2.3.2 Gui

Aplikace pro front-end, která je napsána v Reactu za použití jazyka Javascript. Slouží jako grafické prostředí pro uživatele. Ten má na výběr ze dvou rolí, pod kterými se může v aplikaci pohybovat. Podle toho mu aplikace zobrazí dostupné stránky a funkce. Aplikace využívá Tabler React [6], aby zachovala jednotný vizuální styl. Šablona nabízí responzivní design, přehledné komponenty pro vizualizaci dat a práci se seznamy.

2.3.3 User-management

Jedná se o Spring Boot aplikaci napsanou pro operace nad uživateli, kteří jsou ukládáni do databáze MySQL [7].

Aplikace poskytuje REST API metody pro vytvoření a správu uživatelů s rolí správce nebo lékaře. Vzhledem k zabezpečení jsou k dispozici i metody pro aktivaci nebo zablokování jejich účtů, ověření identity a změnu hesel. Ověření identity a zadaných údajů probíhá odesláním emailu s tokenem a url adresou na zadaný emailový účet při registraci. Jakmile uživatel potvrdí svoji identitu, je mu udělen přístup pro přihlášení do aplikace a možnost s ní pracovat.

Aplikace také umožňuje práci s oznámeními. Poskytuje REST API metody pro jejich vytváření a správu. Uživatel má možnost zobrazit si seznam všech svých oznámení a také mezi nimi filtrovat. Lze je vyfiltrovat podle shody s názvem a stavu přečtené / nepřečtené. Dále má možnost označit všechny nebo pouze jedno jako přečtené nebo nepřečtené.

2.3.4 Gateway

Tato aplikace je napsána ve Spring Bootu. Pro celý projekt představuje „vstupní bránu“, která přijímá požadavky od uživatel z **Gui**. Vzhledem k zabezpečení projektu zde byla implementována Spring Security. Její konfigurace zajišťuje obstarání požadavků, jež přijdou pouze ze správné domény, na které běží právě **Gui** aplikace. Zároveň jsou tyto požadavky kontrolovány, zda je vykonává uživatel s dostatečným oprávněním.

Pokud jsou výše popsané podmínky splněny, požadavek je přeměřován na konkrétní mikroservisu, kde dojde k jejímu obslužení. V případě, že jsou podmínky porušeny, je příkaz zamítnut.

2.3.5 Image-analysis

Jedná se o Spring Boot aplikaci napsanou pro odesílání obrázků na a přijímání výsledků z neuronové sítě. Při prvním spuštění je do databáze uložena výchozí konfigurace, jež představuje maximální počet snímků, který může být zpracovaný na jednu iteraci.

Z **Analysis-management** přijme požadavek na zahájení analýzy spolu s unikátním ID. Podle něho si z **Analysis-management** vyžádá seznam snímků pro danou analýzu v počtu předem stanoveném v databázi. Následně postupně každý snímek odešle do neuronové sítě a čeká na výsledek vyhodnocení. V rámci lokálního vývoje jsou snímky odesílány do **Neural-network-mock**, která slouží pro testování. V produkčním prostředí jsou již snímky odesílány na skutečnou neuronovou síť. Obdržený výsledek se uloží do databáze. Jakmile je analýza u konce, celý proces se ukončí a na **Analysis-management** je odeslán požadavek na dokončení analýzy.

2.3.6 Neural-network-mock

Jedná se o Spring Boot aplikaci, která má simulovat proces vyhodnocování neuronovou sítí.


Pro každý obdržený snímek z **Image-analysis** náhodně vygeneruje hodnoty od 0-1 pro každý z nálezů. Celý tento objekt s vygenerovanými hodnotami následně odešle zpět.

3 Rozšíření webové aplikace

V této části si představíme rozšíření aplikace, které má za účel dát lékařům možnost manuálního vyhodnocení snímků z endoskopické kapsle a z ní následnému vytvoření jednotlivých klasifikací. Ty jsou pak zařazeny do fronty, kde jsou jednotlivé snímky roztrženy do příslušných adresářů a zabaleny do ZIP souboru, který slouží pro učení neuronové sítě. Aplikace bude obsahovat možnost vytvoření a správu uživatelů s novou rolí pro možnost stažení všech klasifikačních souborů.

3.1 Představení

Rozšíření aplikace uděluje lékařům možnost vyhodnotit každý snímek individuálně podle toho, co se na něm nachází. Mají možnost si vybrat z předem definovaných nálezů nebo si přidat vlastní. Ten však musí splnit určitá pravidla, aby se pomocí nich mohla neuronová síť úspěšně učit. Zároveň bude implementován našeptávač, který předejde zbytečnému vznikání redundantních názvů podobného významu. U snímku je také možnost vybrat, že se zde žádný nález nenachází. Pokud lékař vybere tuto možnost, není možné zvolit jakýkoliv jiný. Všechny ostatní je možné spolu kombinovat. Po označení jedné z možností je potřeba kliknout na tlačítko pro uložení. Tím dojde k uložení dat jednotlivého snímku do databáze a předejde se tím případné ztrátě. Pokud by lékař zaznamenal chybné vyhodnocení, je možné kliknout na tlačítko pro resetování klasifikace samostatného snímku. Tak dojde k obnovení původních hodnot.



1

- Nothing
- Angioectasias
- Aphthae
- Bleeding
- Chylous Ccsts
- Lymphangiectasias
- Polypoids
- Stenoses
- Ulcers
- Villous oedemas

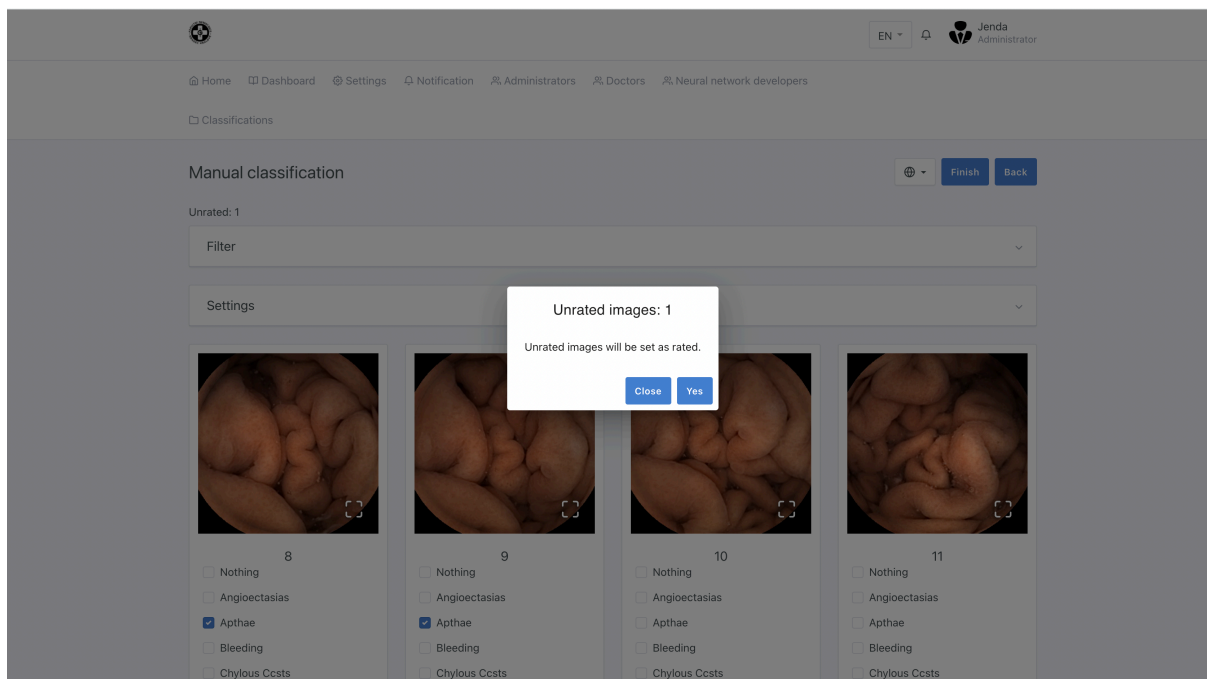
Add custom

Submit

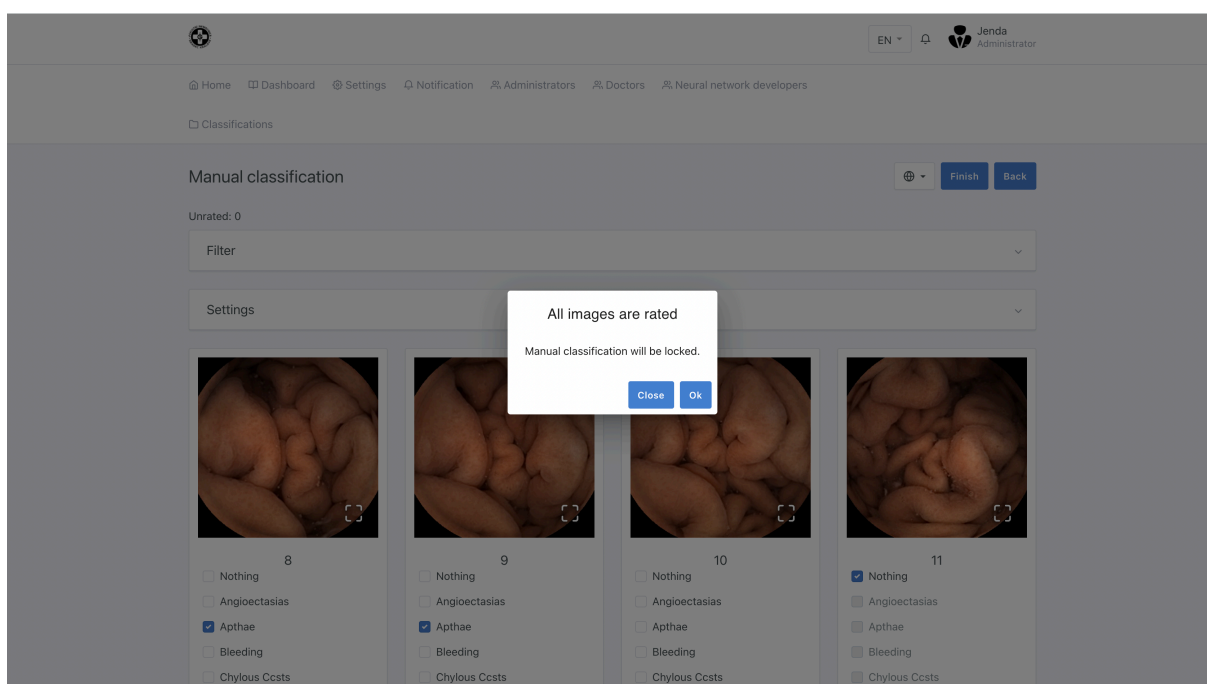
Reset

Obrázek 1: Ukázka formuláře pro vybrání nálezu (zdroj: autor)

Před dokončením klasifikace je potřeba vyhodnotit všechny obrázky v analýze. Pokud by se lékař pokusil dokončit klasifikaci před vyhodnocením všech, systém ho upozorní varovným dialogem. V něm se nachází informace o počtu nevyhodnocených snímků a také možnost označit je jako bez nálezu. Souhlasí-li, systém je uloží jako bez nálezu a zobrazí lékaři další dialog, který ho informuje o tom, že celá analýza je nyní vyhodnocená a je možné danou klasifikaci zamknout.



Obrázek 2: Ukázka dialogu varujícího o neohodnocených snímcích



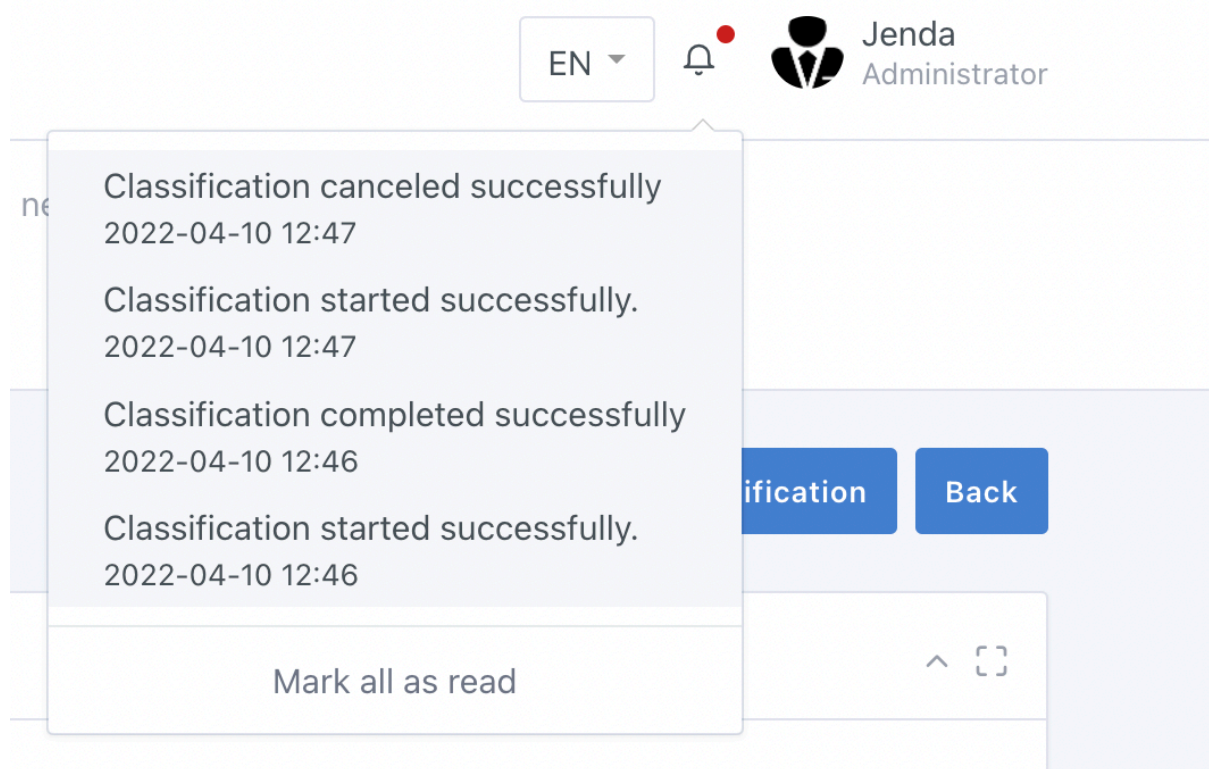
Obrázek 3: Ukázka dialogu oznamujícího uzamčení klasifikace

Pokud by lékař klasifikaci neuzamkl, není možné z ní vytvořit ZIP soubor pro neuronovou síť. U zamčené klasifikace není dostupná jakákoliv změna nálezů u snímků. Pro případnou změnu je potřeba ji znovu odemknout.

Pokud je klasifikace uzamčena a na dané analýze není spuštěna analýza neuronovou sítí nebo jiné vytváření ZIP souboru, je možné spustit vytváření ZIP souboru. Lékař může vyplnit název klasifikace, poznámku a komentář a ten bude uložen do ZIP souboru.

Po vyplnění těchto údajů se uloží záznam do databáze a přidá se do fronty požadavek na vytvoření ZIP souboru pro tuto klasifikaci.

Lékař bude informován o úspěšném započítání a následně také o jeho úspěšném dokončení. Během vytváření má lékař možnost proces přerušit bez možnosti jej znovu obnovit. Pokud by ho zrušil, je nutné zadat nový požadavek na vytvoření.



Obrázek 4: Ukázka možných oznámení o manuální klasifikaci (zdroj: autor)

Všechny vytvořené záznamy jsou uloženy v databázi a zobrazeny v tabulce. U každého z nich je reprezentován stav barevným puntíkem, který signalizuje, zda byl vytvořen, zrušen nebo právě probíhá jeho zpracování. Dále jsou zde informace o názvu klasifikace, popisu, komentáři, datu vzniku a poslední úpravy a jestli byl soubor smazán. Lékař má mimo jiné možnost zobrazit si detail klasifikace, kde je dostupné upravení názvu a popisu. Úprava komentáře není možná vzhledem k tomu, že se daný text ukládá do zazipovaného souboru. Dokončenou klasifikaci je možné stáhnout ve formátu ZIP.

Do aplikace byla přidána také nová role pro uživatele. Slouží pro vývojáře neuronové sítě, aby se mohli přihlásit do systému, zobrazit si seznam všech vygenerovaných klasifikací a stáhnout si zazipované soubory. Ty následně nahrají do neuronové sítě, která se

ze snímků bude učit, jak je vyhodnocovat, a v budoucnu převezme celý tento proces. Tím ulehčí lékařům čas, který musí strávit manuální klasifikací a procházením analýz.

3.2 Stavý

Možné stavy rozšiřující původní aplikaci, které mohou nastat při vytváření ZIP souboru.

ZIP_CREATION_IN_PROGRESS

Stav označující aktuální vytváření ZIP souboru u dané klasifikace.

ZIP_CREATION_CANCELED

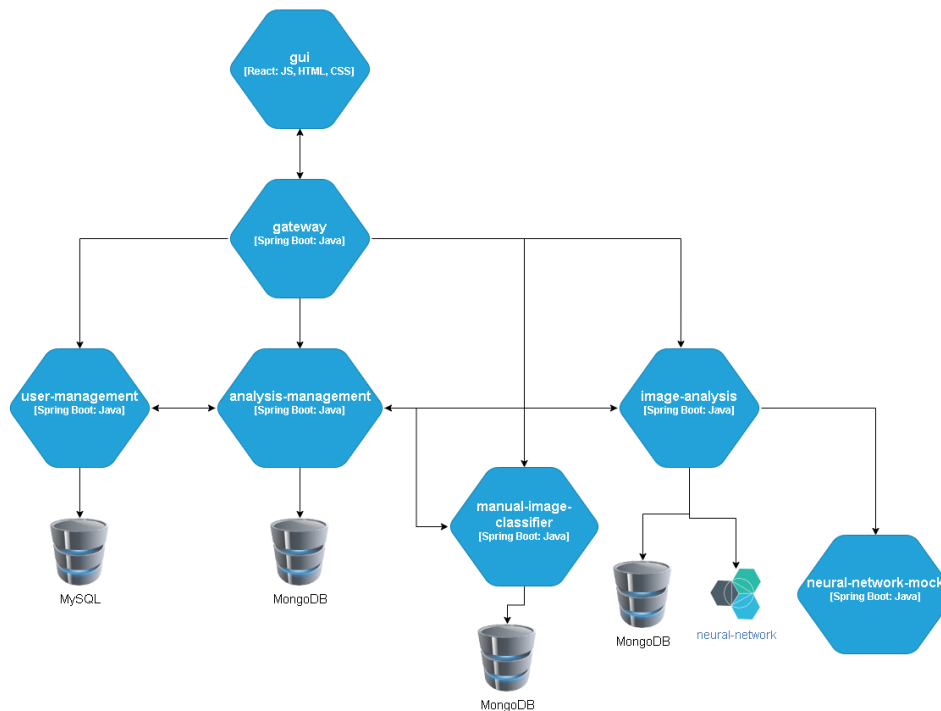
Stav označující zrušené vytváření ZIP souboru uživatelem u dané klasifikace.

ZIP_CREATED

Stav označující dokončené vytváření ZIP souboru u dané klasifikace.

3.3 Rozšíření architektury aplikace

Původní webová aplikace, která se skládala z celkem 6 samostatných aplikací, byla rozšířena o jednu samostatnou aplikaci **Manual-image-classifier**, která má na starosti vyhodnocování snímků. Ke správné implementaci rozšíření ale bylo potřeba upravit i původní aplikace, zejména pak **Gui** a **Analysis-management**.



Obrázek 5: Schéma architektury s aplikací Manual-image-classifier (zdroj: autor)

3.3.1 Manual-image-classifier

Jedná se o aplikaci napsanou v Spring Boot. Poskytuje správu nad klasifikacemi a jejich ZIP soubory, které se ukládají do databáze MongoDB. Tato aplikace je zároveň generuje podle předem dostupných dat o vyhodnocení snímků.

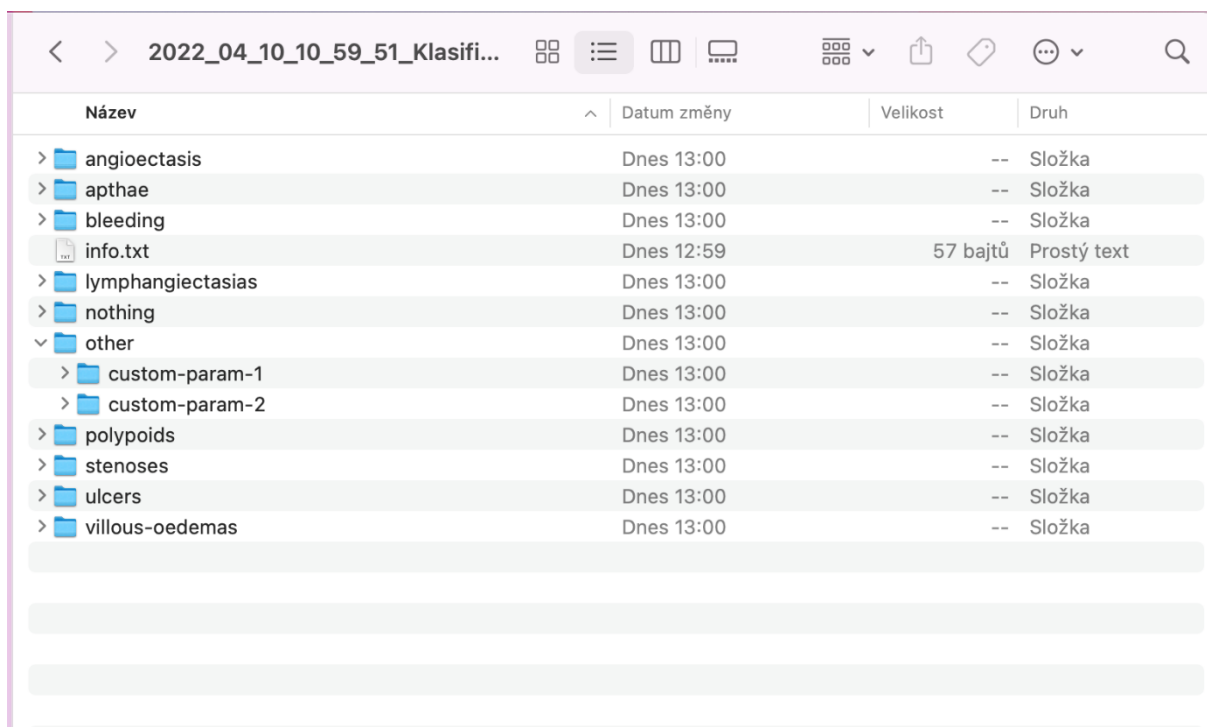
Při prvním spuštění aplikace je do databáze uložena hodnota nastavující maximální povolené množství snímků, které může aplikace zpracovávat na jednu iteraci.

K dispozici jsou REST API metody pro vytvoření a správu klasifikací. Pomocí těchto metod může lékař a správce vytvořit klasifikaci, načíst si její detail a upravit údaje. Dále má možnost označit ji jako aktivní / smazanou nebo si načíst seznam klasifikací na základě požadavků na řazení, stránkování a filtrování podle názvu nebo statusu dokumentu.

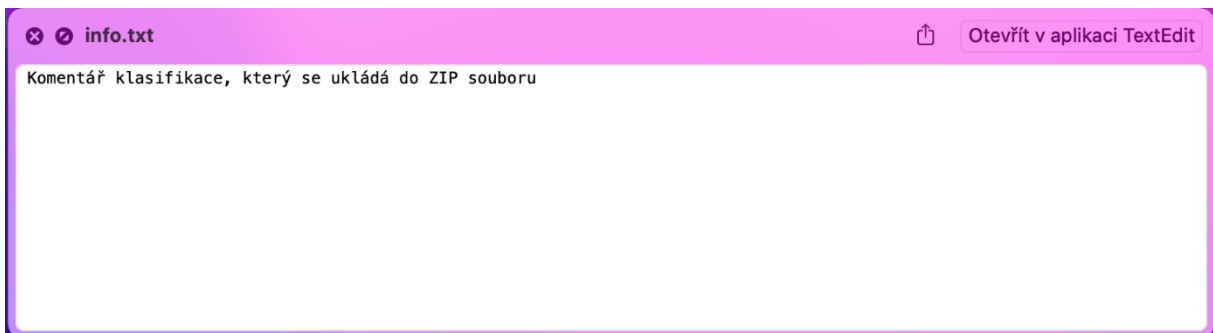
U stránkování je možné nastavit číslo stránky a počet položek na jednu stránku. Řazení můžeme zvolit vzestupně nebo sestupně podle data, kdy byla klasifikace vytvořena. Filtrování dle stavu dokumentu znamená, že budou vyfiltrovány pouze klasifikace, které byly nebo nebyly označeny jako smazané.

Při obdržení požadavku na vytvoření klasifikace z **Analysis-management** dojde k přidání záznamu do databáze, kde se nastaví její stav na **ZIP_CREATION_IN_PROGRESS**. Samotný příkaz na vytvoření ZIP souboru se pak

zařadí do fronty, která se zpracovává asynchronně. Jakmile daný požadavek přijde na řadu, zavolá se REST API metoda z **Analysis-management**, která vrátí seznam snímků z dané analýzy v počtu, nastaveném při prvotním spuštění **Manual-image-classifier**. Každý jednotlivý záznam v tomto seznamu nese informace o názvu, vyhodnocení manuální klasifikací a také samotný soubor. Ten je pak zpracován sérií funkcí, které zjistí označený nálezný, podle toho založí potřebné adresáře, pokud již neexistují, a nakonec tento obrázek do těchto adresářů nakopírují. V případě, že je snímek označen vlastním nálezem, dojde k vytvoření adresáře **other**, do kterého se zařadí všechny takovéto adresáře s vlastním názvem. Po zpracování celé analýzy se zkontroluje přítomnost volitelného komentáře, ten má uživatel možnost vyplnit. Pokud existuje, vytvoří se souboru **info.txt**, kam se vloží obsah. Zároveň se zkontroluje existence chyb při ukládání. V případě, že některé během procesu nastaly, vznikne CSV soubor, do kterého jsou zapsány údaje o chybě a daném záznamu. Tyto soubory jsou následně vloženy do ZIP souboru. Jeho vytváření je tímto dokončené a dojde k jeho uložení do databáze, kde se následně nastaví stav klasifikace na **ZIP_CREATED**.



Obrázek 6: Ukázka obsahu staženého ZIP souboru (zdroj: autor)



Obrázek 7: Ukázka komentáře nacházejícího se v ZIP souboru (zdroj: autor)

A screenshot of a CSV file named "errors.csv" opened in Microsoft Excel. The table has three columns: "_id", "imageId", and "error". It contains 13 rows of data, each representing an error where a file could not be saved to a ZIP archive.

_id	imageId	error
0	623c0ef82c39dc120a1b9b75	File '623c0ef82c39dc120a1b9b75' cannot be saved to zip.
1	623c0ef92c39dc120a1ba02e	File '623c0ef92c39dc120a1ba02e' cannot be saved to zip.
2	623c0ef82c39dc120a1b9b78	File '623c0ef82c39dc120a1b9b78' cannot be saved to zip.
3	623c0ef82c39dc120a1b9b72	File '623c0ef82c39dc120a1b9b72' cannot be saved to zip.
4	623c0ef22c39dc120a1b8fd2	File '623c0ef22c39dc120a1b8fd2' cannot be saved to zip.
5	623c0ef22c39dc120a1b8fd5	File '623c0ef22c39dc120a1b8fd5' cannot be saved to zip.
6	623c0ef92c39dc120a1ba02b	File '623c0ef92c39dc120a1ba02b' cannot be saved to zip.
7	623c0ef22c39dc120a1b8fc9	File '623c0ef22c39dc120a1b8fc9' cannot be saved to zip.
8	623c0ee92c39dc120a1b83a5	File '623c0ee92c39dc120a1b83a5' cannot be saved to zip.
9	623c0ef22c39dc120a1b8fc6	File '623c0ef22c39dc120a1b8fc6' cannot be saved to zip.
10	623c0ef42c39dc120a1b93fb	File '623c0ef42c39dc120a1b93fb' cannot be saved to zip.
11	623c0ef82c39dc120a1b9b6f	File '623c0ef82c39dc120a1b9b6f' cannot be saved to zip.
12	623c0ef92c39dc120a1ba049	File '623c0ef92c39dc120a1ba049' cannot be saved to zip.
13	623c0ef82c39dc120a1b9b6e	File '623c0ef82c39dc120a1b9b6e' cannot be saved to zip.

Obrázek 8: Ukázka CSV tabulky s chybami nacházející se v ZIP souboru (zdroj: autor)

3.3.2 Analysis-management

V této aplikaci byly k datům o analýze přidány informace o stavu klasifikace lékařem, označující, zda je klasifikace uzamčena. Také její stavy byly rozšířeny o `ZIP_CREATION_IN_PROGRESS`, `ZIP_CREATION_CANCELED` a `ZIP_CREATED`.

Přibyly další REST API metody, které slouží k úpravě vyhodnocení jednotlivých snímků, obnově výchozích hodnot u manuální klasifikace snímků, označení neohodnocených obrázků jako ohodnocených bez nálezu, vytvoření klasifikace a její dokončení.

Vytvoření klasifikace ověří všechna dostupná data a následně zavolá REST API metodu z aplikace **Manual-image-classifier**, která se postará o zbytek vytváření. Po dokončeném vytváření ZIP souboru **Manual-image-classifier** zavolá REST API metodu z aplikace **Analysis-management**, ta obstará správné nastavení stavu a informuje uživatele o dokončení.

3.3.3 Gui

Tato aplikace byla rozšířena o několik nových stránek, které bylo potřeba přidat pro novou funkci manuální klasifikace snímků.

První z přidanych stránek slouží přímo pro manuální klasifikaci. Nabízí stejné funkce jako původní sekce „Obrázky“. Je rozšířena o možnost u každého obrázku individuálně vybrat nález, který se na něm může nacházet. V případě, že se na snímku nic závadného nenachází, lékař má možnost označení jako bez nálezu.

Manual classification

Finish Back

Unrated: 100+

Filter

Order

Image status Document status

Manual classification status

Cancel Filter

Settings

Number of decimal places of diagnosis values: 3

Limit the maximum number of columns:

Maximum number of columns: 4

Show fullscreen button:

Adjust the size of the images:

Maximal width: 320 Maximal height: 320

8

9

10

11

12

13

14

15

- Nothing
- Angioectasias
- Apthae
- Bleeding
- Chylous Ccsts
- Lymphangiectasias
- Polypoids
- Stenoses
- Ulcers
- Villous oedemas

Add custom

Submit

Reset

Up Back

Obrázek 9: Ukázka stránky manuální klasifikace snímků (zdroj: autor)

Další přidaná stránka obsahuje seznam klasifikací dané analýzy. Zde je přehled všech takovýchto záznamů, které vzešly z předem vybrané analýzy spolu s jejich stavem klasifikace, názvem, popisem, komentářem, stavem dokumentu a počtem stažení. Lékař zde má možnost je upravit, vytvořit nové, stáhnout jejich zazipovaný soubor a filtrovat si mezi nimi. Stažení ZIP souboru je ovšem možné pouze u takových klasifikací, které byly dokončeny. Při prvotním zobrazení této stránky jsou zobrazeny pouze tyto sloupce: stav klasifikace, název, stav dokumentu a možnosti. Ostatní sloupce jsou skryté kvůli ušetření místa, responzivitě a přehlednosti, ale je možné je zobrazit přes dodatečné nastavení tabulky.

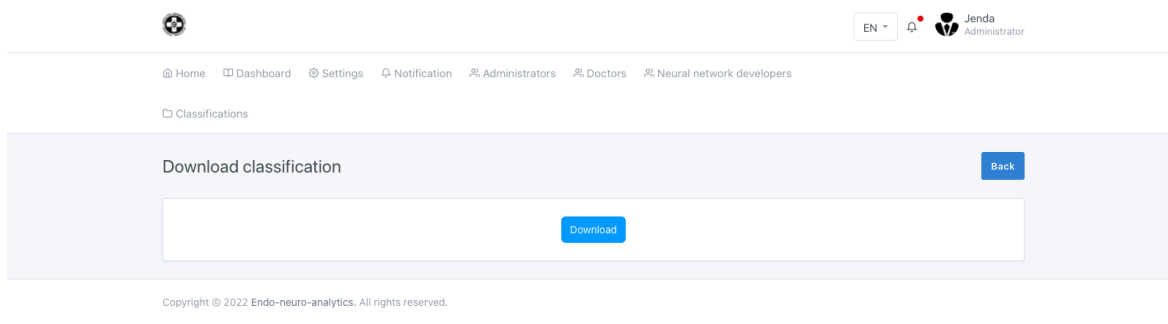
The screenshot shows a web application interface for managing classifications. At the top, there is a navigation bar with links for Home, Dashboard, Settings, Notification, Administrators, Doctors, and Neural network developers. The user is logged in as Jenda Administrator. The main content area is titled 'Classification' and features a 'Manual classification' button and a 'Back' button. Below this is a table with the following columns: State, Name, Status, and Options. The table contains three rows of data:

State	Name	Status	Options
●	Klasifikace1	ACTIVE -	⊕
●	Klasifikace2	ACTIVE -	⊕
●	Klasifikace3	ACTIVE -	⊕

At the bottom of the table, there is a pagination control showing '1' of 1 page and a 'Rows per page' dropdown set to '10'. There are also 'Up' and 'Back' buttons at the bottom right of the table area.

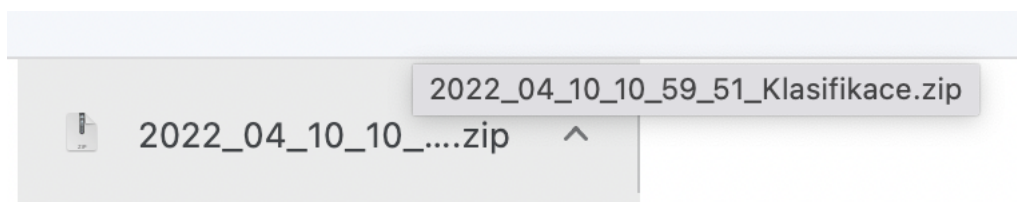
Obrázek 10: Ukázka stránky se seznamem klasifikací dané analýzy (zdroj: autor)

Po kliknutí na možnost pro stažení se nám otevře nové okno, aby uživatel mohl pracovat s aplikací i během doby stahování většího ZIP souboru. V tomto okně se nachází pouze jedno tlačítko, které po zmáčknutí inicializuje přípravu a stažení.

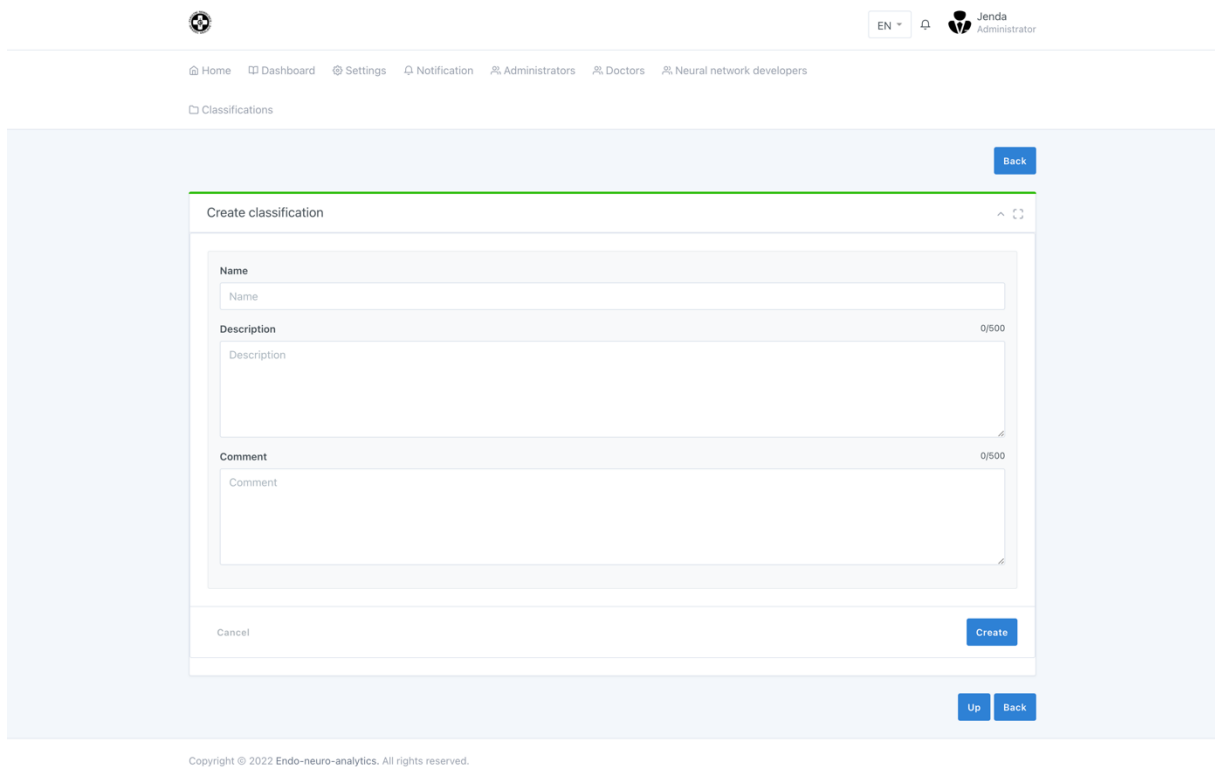


Obrázek 11: Ukázka stránky pro stažení ZIP souboru klasifikace (zdroj: autor)

Při vytváření klasifikace je možné vyplnit její název, popis a komentář. Když se hotový ZIP soubor ukládá do databáze, nastaví se jeho jméno podle aktuálního času ve formátu **YYYY_MM_DD_HH_MM_SS**. Takto pojmenovaný soubor bude stažen, když název u vytvořené klasifikace zůstane nevyplněný. Pokud však uživatel název vyplní, tak při stažení bude název ve formátu **YYYY_MM_DD_HH_MM_SS_NÁZEV_SOUBORU**.



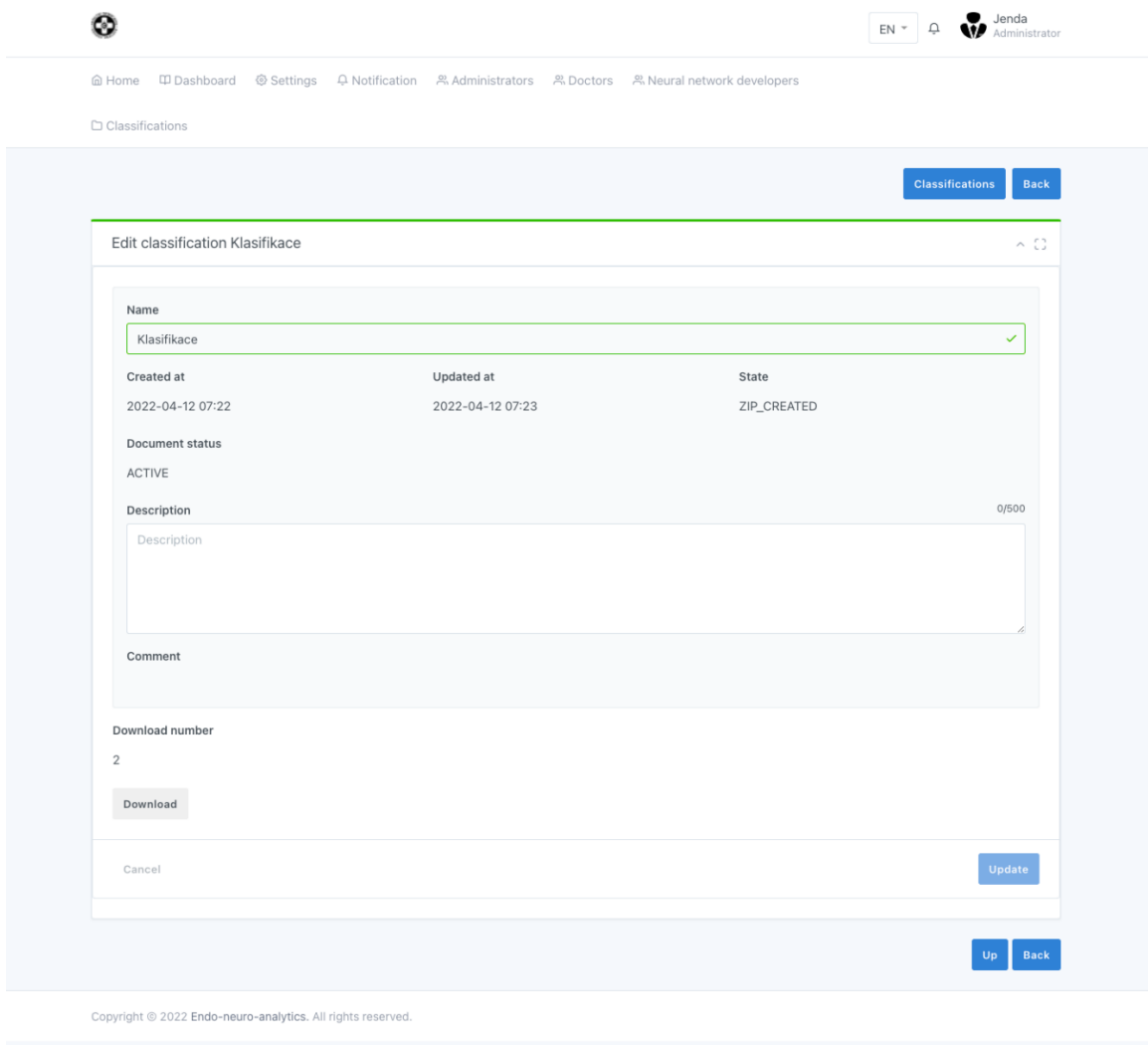
Obrázek 12: Ukázka názvu staženého ZIP souboru (zdroj: autor)



Copyright © 2022 Endo-neuro-analytics. All rights reserved.

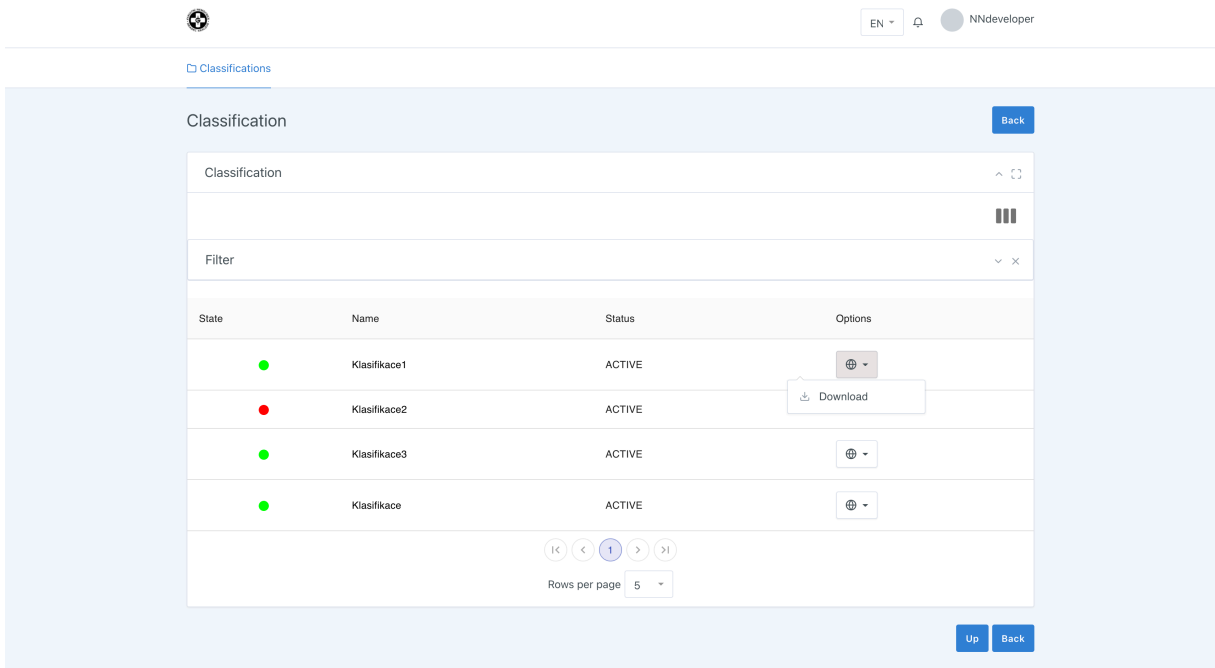
Obrázek 13: Ukázka stránky pro vytvoření nové klasifikace (zdroj: autor)

V detailu klasifikace pak můžeme zjistit všechny potřebné informace jako název, datum a čas vytvoření a poslední úpravy, stav klasifikace, stav dokumentu, popis a komentář. Na rozdíl od vytváření však můžeme upravit pouze název a popis. To je způsobeno především tím, že komentář se ukládá přímo do vytvořeného ZIP souboru, takže není možné jej měnit po dokončení. Samotná stránka nám pak umožňuje i stažení tohoto souboru, pokud je klasifikace dokončená a zobrazuje nám počet stažení.



Obrázek 14: Ukázka detailu klasifikace (zdroj: autor)

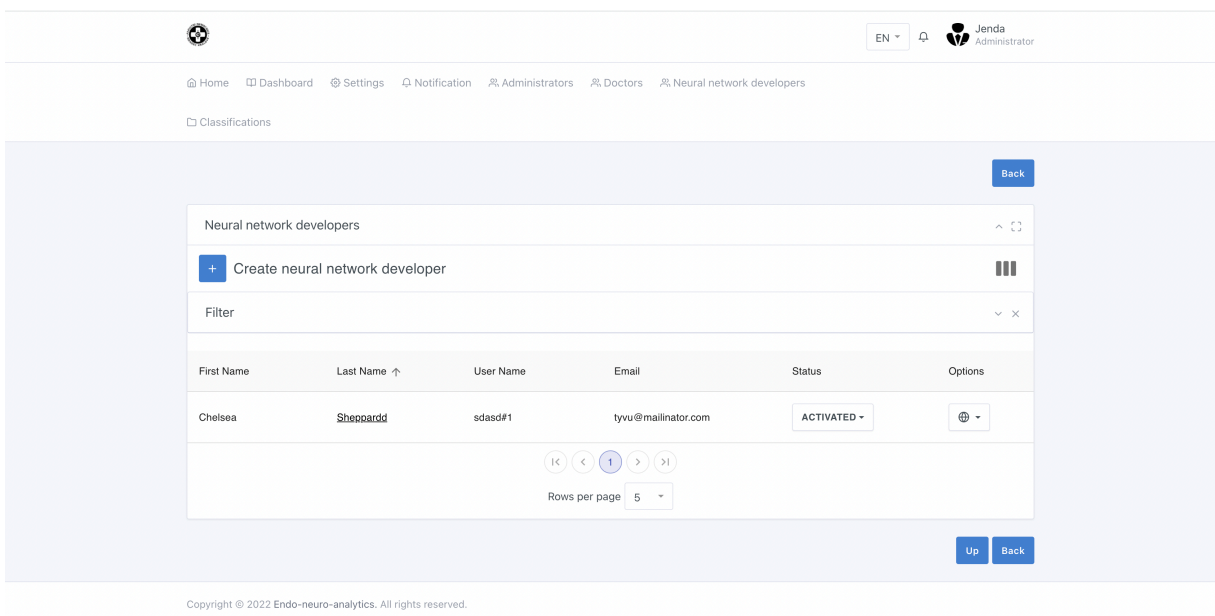
Do aplikace bylo také potřeba přidat stránku se seznamem všech klasifikací nezávisle na analýze, ke které patří. Tento seznam bude primárně využíván uživateli s rolí vývojář neuronové sítě, kteří budou z této tabulky stahovat vytvořené ZIP soubory a následně je používat pro učení neuronové sítě. Tabulka má stejně jako všechny předešlé možnost filtrování. Uživatel může filtrovat podle názvu a stavu dokumentu. Možnost stažení vytvořeného souboru je dostupná pouze u dokončených klasifikací.



Copyright © 2022 Endo-neuro-analytics. All rights reserved.

Obrázek 15: Ukázka stránky se seznamem všech klasifikací (zdroj: autor)

Pro správu a vytváření uživatelů s novou rolí je do aplikace přidána stránka, na které se nachází tabulka se všemi těmito uživateli. Oprávnění ke vstupu na tuto stránku má pouze správce, jenž je také může vytvářet a spravovat.



Copyright © 2022 Endo-neuro-analytics. All rights reserved.

Obrázek 16: Ukázka stránky se seznamem uživatelů s rolí vývojář neuronové sítě (zdroj: autor)

The screenshot shows a web application interface for creating a new user. The page title is "Create neural network developer". The form contains the following fields and elements:

- First Name ***: A text input field with a red border and a red error message below it: "First name is not specified." There is a small 'x' icon in the top right corner of the input field.
- Middle name (/ names)**: A text input field.
- Last Name ***: A text input field.
- User Name ***: A text input field containing the value "jan.kruncik@uhk.cz".
- Email ***: A text input field.
- About me**: A text area with a character count of 0/500.
- Passwords ***: A section with a "Show passwords" toggle switch. It contains two columns:
 - Password 1**: A text input field with a "Generate password" button below it.
 - Password 2**: A text input field with a "Generate password" button below it.
- Repeat password 1 ***: A text input field.
- Repeat password 2 ***: A text input field.
- At the bottom of the form, there are "Cancel" and "Create" buttons.

The interface also includes a navigation menu at the top with items like Home, Dashboard, Settings, Notification, Administrators, Doctors, and Neural network developers. A user profile "Jenda Administrator" is visible in the top right corner. A "Back" button is located in the top right of the form area, and "Up" and "Back" buttons are at the bottom right.

Obrázek 17: Ukázka stránky pro vytvoření nového uživatele s rolí vývojář neuronové sítě (zdroj: autor)

3.3.4 User-management

Tato aplikace byla rozšířena o novou roli pro uživatele určenou pro vývojáře neuronové sítě. Uživatel s touto rolí má práva na zobrazení seznamu všech vytvořených klasifikací a může si stáhnout vytvořené soubory pro učení neuronové sítě.

K dispozici jsou REST API metody pro jejich přidání, správu a výpis všech takovýchto uživatelů z databáze. Stejně jako u ostatních rolí je i zde pro lepší zabezpečení k dispozici možnost aktivace nebo zablokování účtu, ověření identity nebo změny hesel. Používá se stejné ověření identity a zadaných údajů jako u předešlých rolí, tedy zasláním adresy spolu s ověřovacím tokenem na email, který byl zadaný při registraci uživatele. Potvrzením tohoto emailu dojde ke schválení registrace. Do té doby, dokud není uživatel nově registrovaného účtu schválen, nemá možnost se přihlásit do aplikace a pracovat s ní.

3.3.5 Gateway

Do této aplikace byly přidány REST API metody z nové samostatné aplikace **Manual-image-classifier** tak, aby nedošlo k neoprávněnému použití aplikace. Metody byly omezeny na uživatele s určitou rolí. Některé byly dokonce zakázány pro volání uživatelem, protože jsou zamýšleny pouze pro použití ostatními mikroservis službami.

Zároveň bylo přidáno zabezpečení REST API metod pro aplikace **Analysis-management** a **User-management**. Pro **Analysis-management** to byly metody pro úpravu dat analýz a jejich snímků. Pro **User-management** metody týkající se vytváření a úpravy záznamů uživatelů s novou rolí vývojář neuronové sítě.

4 Shrnutí výsledků

Při implementaci projektu bylo využito vývojové prostředí IntelliJ IDEA pro práci na back-endových službách aplikace a WebStorm pro front-end aplikace.

Aplikace naplnila cíl zadání práce s jednou výjimkou. Z časových důvodů se nepodařilo do projektu naimplementovat našeptávač, který by nabízel na **Gui** již vyplněné vlastní nálezy v manuální klasifikaci u snímků jednotlivých analýz. To však nijak neomezuje funkčnost rozšíření a uživatel má stále možnost vlastní nález ke snímku vyplnit.

Aplikace umožňuje manuální klasifikaci snímků analýzy. Z ní následné vytváření klasifikací a jejich správu. Je také schopná samostatně vygenerovat ZIP soubor s vytvořenými adresáři dle označených nálezů u snímků a ty do nich roztřídit. Pokud jsou v procesu klasifikace vytvořeny i soubory pro dodatečné poznámky nebo zaznamenané chyby, jsou také přibaleny do ZIP souboru. Ten je možné stáhnout a předat neuronové síti.

Dále poskytuje možnost pro vytvoření a správu uživatelů s rolí vývojář neuronové sítě, jeho přihlášení do aplikace a práce s ní.

Za předpokladu, že by rozšíření mělo být nasazeno na produkční prostředí stejně jako zbytek aplikace, bylo by dobré podrobit tuto část uživatelskému testování.

5 Závěr

Práci lze rozdělit na dvě části. První z nich představuje původní aplikaci **Endo-neuro-analytics**, její využití, architekturu a použité technologie. V druhé části je popsána rozšiřující část manuální klasifikace a samostatná aplikace obstarávající proces vytváření ZIP souborů. Zároveň jsou zde i vysvětlena rozšíření, která bylo potřeba udělat k již existujícím aplikacím.

Uživatelé aplikace by v budoucnu jistě uvítali našeptávač, který se, jak jsem již zmínil, bohužel z časových důvodů nepovedlo naimplementovat. Pomohl by jim předejít duplikaci podobných názvů vlastního nálezu v manuální klasifikaci snímků.

Díky této aplikaci jsem rozšířil své vědomosti v odvětví vývoje webových aplikací a rozdělení back-endových služeb do mikroservis. Zároveň jsem ke svým znalostem v oblasti vývoje přidal nový programovací jazyk Java s použitím frameworku Spring Boot. V poslední řadě pak zlepšil své programovací schopnosti v knihovně React.

V praktické části byly odebrány funkce pro generování hesel a JWT tokenu z aplikací **User-management** a **Gateway** z důvodu bezpečnosti.

6 Seznam obrázků

Obrázek 1: Ukázka formuláře pro vybrání nálezu (zdroj: autor)	9
Obrázek 2: Ukázka dialogu varujícího o neohodnocených snímcích.....	10
Obrázek 3: Ukázka dialogu oznamujícího uzamčení klasifikace.....	10
Obrázek 4: Ukázka možných oznámení o manuální klasifikaci (zdroj: autor)	11
Obrázek 5: Schéma architektury s aplikací Manual-image-classifier (zdroj: autor)	13
Obrázek 6: Ukázka obsahu staženého ZIP souboru (zdroj: autor)	14
Obrázek 7: Ukázka komentáře nacházejícího se v ZIP souboru (zdroj: autor).....	15
Obrázek 8: Ukázka CSV tabulky s chybami nacházející se v ZIP souboru (zdroj: autor)..	15
Obrázek 9: Ukázka stránky manuální klasifikace snímků (zdroj: autor)	17
Obrázek 10: Ukázka stránky se seznamem klasifikací dané analýzy (zdroj: autor)	18
Obrázek 11: Ukázka stránky pro stažení ZIP souboru klasifikace (zdroj: autor).....	19
Obrázek 12: Ukázka názvu staženého ZIP souboru (zdroj: autor).....	19
Obrázek 13: Ukázka stránky pro vytvoření nové klasifikace (zdroj: autor).....	20
Obrázek 14: Ukázka detailu klasifikace (zdroj: autor).....	21
Obrázek 15: Ukázka stránky se seznamem všech klasifikací (zdroj: autor)	22
Obrázek 16: Ukázka stránky se seznamem uživatelů s rolí vývojář neuronové sítě (zdroj: autor).....	22
Obrázek 17: Ukázka stránky pro vytvoření nového uživatele s rolí vývojář neuronové sítě (zdroj: autor).....	23

7 Seznam zdrojových kódů

Zdrojový kód 1: React komponenta bez použití JSX (zdroj: autor).....	3
Zdrojový kód 2: React komponenta s použitím třídy (zdroj: autor).....	3
Zdrojový kód 3: React funkční komponenta za použití konstanty (zdroj: autor).....	4
Zdrojový kód 4: React funkční komponenta za použití funkce (zdroj: autor)	4
Zdrojový kód 5: React složená komponenta (zdroj: autor)	4

8 Literatura

- [1] React – A JavaScript library for building user interfaces [online]. Facebook, 2022 [cit. 2022-04-01]. Dostupné z: <https://reactjs.org/>
- [2] KRUNČÍK, Jan. Aplikace pro podporu zpracování dat z endokapsle [online]. Hradec Králové, 2021 [cit. 2022-03-31]. Dostupné z: <https://theses.cz/id/sepgef5/>. Diplomová práce. Univerzita Hradec Králové, Fakulta informatiky a managementu. Vedoucí práce doc. Ing. Filip Malý, Ph.D.
- [3] Spring Boot. Spring [online]. Palo Alto: VMware, c2022 [cit. 2022-04-05]. Dostupné z: <https://spring.io/projects/spring-boot>
- [4] Spring Security. Spring [online]. Palo Alto: VMware, c2022 [cit. 2022-04-05]. Dostupné z: <https://spring.io/projects/spring-security>
- [5] MongoDB: The Application Data Platform [online]. New York City: MongoDB, c2022 [cit. 2022-04-05]. Dostupné z: <https://www.mongodb.com/>
- [6] GitHub - tabler/tabler-react at typescript. GitHub: Where the world builds software · GitHub [online]. San Francisco: GitHub, c2022 [cit. 2022-04-08]. Dostupné z: <https://github.com/tabler/tabler-react/tree/typescript>
- [7] MySQL [online]. Austin: Oracle Corporation, c2022 [cit. 2022-04-08]. Dostupné z: <https://www.mysql.com/>

9 Přílohy

9.1 Zprovoznění aplikace na lokálním zařízení

Tato příloha popisuje prvotní konfiguraci aplikace na lokálním zařízení pro účely vývoje. Popis předpokládá již stažené zdrojové kódy aplikace a otevřené v programu vývojového prostředí.

9.1.1 Manual-image-classifier

Jelikož se jedná o Spring Boot aplikaci, která má v sobě integrovaný aplikační server Apache Tomcat, je spuštění opravdu jednoduché. Stačí nám pouze v rámci konfigurace nastavit tyto systémové proměnné.

Aplikace využívá profily pro vývojové a pro produkční prostředí, která obsahují všechny systémové proměnné pro konfiguraci aplikace a připojení k databázi.

Systémové proměnné pro zprovoznění aplikace v rámci vývojového profilu:

- REGISTER_CLIENT_URL
 - Adresa, kde je spuštěn Eureka server, což je v tomto případě Gateway aplikace
 - Například: `http://localhost:8761`
- ANALYSIS_MANAGEMENT_URL
 - Adresa, kde je spuštěna aplikace Analysis-management
 - Například: `http://localhost:8082`
- MAX_IN_MEMORY_SIZE
 - Hodnota, která nám nastavuje maximální velikost paměti, kterou může aplikace využít při vytváření ZIP souboru
 - Například: 524288000 (500 MB)

Dodatečné systémové proměnné pro zprovoznění aplikace v rámci produkčního profilu:

- MANUAL_IMAGE_CLASSIFIER_DB_AUTHENTICATION
 - Název databáze / kolekce, ve které se nachází uživatel, jímž se chceme přihlašovat do databáze

- Například: admin
- MANUAL_IMAGE_CLASSIFIER_DB_USERNAME
 - Přihlašovací jméno k připojení do databáze
 - Například: root
- MANUAL_IMAGE_CLASSIFIER_DB_PASSWORD
 - Přihlašovací heslo k připojení do databáze
 - Například: root
- MANUAL_IMAGE_CLASSIFIER_DB_DATABASE
 - Název databáze / kolekce, ve které se budou uchovávat data
 - Například: manual-image-classifier
- MANUAL_IMAGE_CLASSIFIER_DB_PORT
 - Port k připojení na MongoDB server
 - Například: 27017
- MANUAL_IMAGE_CLASSIFIER_DB_HOST
 - Adresa, kde je spuštěn MongoDB server
 - Například: localhost



Zadání bakalářské práce

Autor: Michal Vaďák

Studium: I1900269

Studijní program: B1802 Aplikovaná informatika

Studijní obor: Aplikovaná informatika

Název bakalářské práce: Aplikace pro podporu klasifikace medicínských dat

Název bakalářské práce AJ: Application to support the classification of medical data

Cíl, metody, literatura, předpoklady:

Cíl: vytvořit webovou aplikaci pro podporu ruční klasifikace medicínských dat.

Osnova:

- Úvod
- Analýza a návrh aplikace
- Popis aplikace
- Implementace
- Závěr
- Literatura

Garantující pracoviště: Katedra informatiky a kvantitativních metod,
Fakulta informatiky a managementu

Vedoucí práce: doc. Ing. Filip Malý, Ph.D.

Datum zadání závěrečné práce: 26.1.2021