

Univerzita Jana Evangelisty Purkyně
v Ústí nad Labem
Přírodovědecká fakulta



Aplikace pro anotaci fotografických snímků
geografické povahy

BAKALÁŘSKÁ PRÁCE

Vypracoval: Ján Páleník

Vedoucí práce: Ing. Jakub Trojánek

Studijní program: Aplikovaná informatika

Studijní obor: Informační systémy

ÚSTÍ NAD LABEM 2023

UNIVERZITA JANA EVANGELISTY PURKYNĚ V ÚSTÍ NAD LABEM
Přírodovědecká fakulta
Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Jméno a příjmení: **Ján PÁLENÍK**
Osobní číslo: **F20222**
Studijní program: **B1802 Aplikovaná informatika**
Studijní obor: **Informační systémy**
Téma práce: **Aplikace pro anotaci fotografických snímků geografické povahy**
Zadávající katedra: **Katedra informatiky**

Zásady pro vypracování

Cílem této bakalářské práce je návrh webové aplikace pro anotování fotografických snímků opatřených geografickými údaji. Aplikace má podporovat workflow management a správu uživatelů a jejich rolí v rámci systému. Výsledné anotace budou sloužit jako vstupní data pro strojové učení ve formátu GeoJSON. Klíčovým aspektem aplikace proto je, aby zajistila, že tato vstupní data budou kvalitní ve smyslu minimální chybovosti při maximální kvantitě.

Osnova:

1. anotace geodat se zaměřením na fotografické snímky
2. požadavky na data pro strojové učení
3. popis použitých technologií
4. analýza procesů a workflow managementu
5. návrh aplikace
6. implementace aplikace
7. zhodnocení aplikace

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

- BUTLER, H., M. DALY, A. DOYLE, S. GILLIES, S. HAGEN a T. SCHaub. *RFC 7946: The GeoJSON Format*. 2016. DOI: 10.17487/RFC7946. Dostupné také z: <https://www.rfc-editor.org/info/rfc7946>
- ARLOW, J. a I. NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
- KONECNY, G. *Geoinformation: Remote Sensing, Photogrammetry and Geographic Information Systems*. Second edition. Taylor & Francis, 2014. ISBN 978-1420068566.

Vedoucí bakalářské práce: **Ing. Jakub Trojánek**
METRANS, a.s., Fun on run s.r.o.

Datum zadání bakalářské práce: 27. října 2022
Termín odevzdání bakalářské práce: 8. prosince 2023

L.S.

doc. RNDr. Michal Varady, Ph.D.
děkan

RNDr. Jiří Škvor, Ph.D.
vedoucí katedry

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a použil jen pramenů, které cituji a uvádím v přiloženém seznamu literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., ve znění zákona č. 81/2005 Sb., autorský zákon, zejména se skutečností, že Univerzita Jana Evangelisty Purkyně v Ústí nad Labem má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Jana Evangelisty Purkyně v Ústí nad Labem oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladu, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

V Ústí nad Labem dne 8. prosince 2023

Podpis:

Děkuji vedoucímu práce Ing. Jakubovi Trojánkovi za neocenitelné rady
a pomoc při tvorbě bakalářské práce.

APLIKACE PRO ANOTACI FOTOGRAFICKÝCH SNÍMKŮ GEOGRAFICKÉ POVAHY

Abstrakt:

Cílem této bakalářské práce je návrh webové aplikace pro anotování fotografických snímků opatřených geografickými údaji. Aplikace má podporovat workflow management a správu uživatelů a jejich rolí v rámci systému. Výsledné anotace budou sloužit jako vstupní data pro strojové učení ve formátu GeoJSON. Klíčovým aspektem aplikace proto je, aby zajistila, že tato vstupní data budou kvalitní ve smyslu minimální chybovosti při maximální kvantitě.

Klíčová slova: anotace, strojové učení, workflow management

APPLICATION FOR ANNOTATION OF PHOTOGRAPHIC IMAGES OF A GEOGRAPHICAL NATURE

Abstract:

This bachelor thesis aims to design a web application for annotating photographic images with geographic data. The application is to support workflow management and management of users and their roles within the system. The resulting annotations will serve as input data for machine learning in GeoJSON format. A key aspect of the application is therefore to ensure that this input data is of high quality in the sense of minimum error rate with maximum quantity.

Keywords: annotation, machine learning, workflow management

Obsah

| | |
|---|-----------|
| 1. Úvod | 13 |
| 2. Základní teoretické koncepty | 15 |
| 2.1. Strojové učení | 15 |
| 2.2. Klasifikace dat v obraze | 17 |
| 2.3. Workflow management | 18 |
| 2.4. UX - Uživatelská zkušenost | 19 |
| 2.5. Uživatelské role | 19 |
| 2.6. Metodika vývoje | 21 |
| 2.7. Použité technologie | 22 |
| 3. Aplikačně-implementační návrh | 27 |
| 3.1. Struktura aplikace | 27 |
| 3.2. Analýza procesů | 27 |
| 3.3. Uživatelské role | 30 |
| 3.4. Datasetsy | 33 |
| 3.5. Projekt | 36 |
| 3.6. 2D Editor | 42 |
| 3.7. Datový model | 46 |
| 4. Závěr | 51 |
| A. Externí přílohy | 59 |

1. Úvod

Tato bakalářská práce se zaměřuje na návrh a vývoj webové aplikace určené pro efektivní anotace a organizaci shromažděných dat, která jsou nezbytná pro strojové učení. Dalším důležitým aspektem je implementace různých uživatelských rolí, které umožní vícefázovou kontrolu nad označenými daty.

Spolupráce s firmou Melown Technologies SE dala práci reálný kontext a umožnila aplikaci inovací v praxi. Motivací k vývoji byla nutnost optimalizovat proces shromažďování dat v porovnání s tradičními metodami, jako je ruční anotace v geografických informačních systémech (GIS). Projekt vycházel z potřeby efektivně zpracovávat velké objemy snímků, jejichž manuální anotace v GIS byla neefektivní a vyžadovala součinnost více expertů.

Tradiční metody anotací se ukázaly jako časově náročné a neposkytovaly adekvátní kontrolu nad prací anotátorů, což bylo zvláště problematické u projektů vyžadujících detailní shromažďování dat pro strojové učení. Proto bylo nezbytné vyvinout nové řešení, které by bylo flexibilní, škálovatelné, umožnilo efektivní management datových sad a nabídlo možnost sledování historie dat pro detekci regresí a provádění zpětných úprav.

Vývoj této aplikace je klíčový, zejména kvůli narůstající potřebě efektivně zpracovávat a kategorizovat rozsáhlá data pro strojové učení. Proces je výzvou nejen technicky, ale i organizačně. Proto je zásadní mít nástroj pro správu, kontrolu a rozdělování úkolů mezi uživatele s různými rolemi. Tato bakalářská práce přináší užitek nejen firmě Melown Technologies SE, ale i širší komunitě v oblasti strojového učení.

První část práce poskytuje teoretický základ o strojovém učení, klasifikaci dat a uvádí čtenáře do problematiky. Dále se věnuje workflow managementu, uživatelské zkušenosti (UX), metodikám vývoje softwaru a v závěru kapitoly jsou představeny zvolené technologie použité při vývoji aplikace.

Druhá část práce se zaměřuje na analýzu procesu anotace a podrobně rozebírá důležitost uživatelských rolí v systému. Kapitola také obsahuje důkladný popis správy dat a klíčových prvků aplikace, včetně představení editoru a datového modelu aplikace.

2. Základní teoretické koncepty

Tato kapitola se zabývá analýzou základních teoretických konceptů, které jsou fundamentální pro pochopení motivace a obsahu této práce. Kapitola zahrnuje následující klíčové prvky:

- **Strojové učení:**

Zabývá se vysvětlením základů strojového učení a jeho různých typů na základě vstupních dat.

- **Klasifikace dat v obrazu:**

V této části jsou vysvětleny základy klasifikace dat v obrazu, což je často klíčový úkol ve strojovém učení.

- **Workflow management:**

Zaměřuje se na nastavení a analýzu pracovního postupu, který jsou zásadní pro efektivní zpracování dat a organizaci anotačních procesů.

- **UX - Uživatelská zkušenost:**

Poskytuje úvod do UX návrhu, který je důležitý zhlediska použitelnosti aplikace.

- **Uživatelské role:**

Vysvětluje funkci a důležitost uživatelských rolí v rámci systému..

- **Metodika vývoje:**

Soustředí se na proces vývoje webové aplikace, která je jádrem této práce, od analýzy po implementaci a testování.

- **Použité technologie:**

Představuje konkrétní technologie a nástroje, jež byly využity během vývoje aplikace.

2.1. Strojové učení

Strojové učení je obor informatiky, který se zabývá vývojem algoritmů a matematických modelů. Tyto algoritmy se opírají o soubor příkladů určitého jevu. Příklady mohou pocházet z přírody, mohou být vytvořené člověkem nebo jiným algoritmem. Strojové učení lze také charakterizovat jako proces řešení praktických problémů pomocí 2 kroků – nejdříve musí být data shromážděna, poté se na jejich základě algoritmicky vytvoří statistický model. Tento model později slouží k řešení praktických problémů. Souboru vstupních dat v kontextu strojového učení se říká dataset. [1]

Typy strojového učení

Strojové učení lze rozdělit do několika kategorií, a to supervizované, částečně supervizované, nesupervizované a zpětnovazební učení. Tato práce se zabývá získáváním vstupních dat pro supervizované učení, nicméně níže jsou stručně popsány hlavní rozdíly vstupních dat mezi jednotlivými typy strojového učení.[2]

Supervizované učení

Supervizované strojové učení je oblast strojového učení, kde se modely učí z označených trénovacích dat a aplikují se na nová neoznačená data k predikci výstupních hodnot. Požadavky na vstupní data jsou následující:

- **Označená data:**

Supervizované učení vyžaduje soubor dat s přesně definovanými výstupními hodnotami.

- **Kvalitní a relevantní data:**

Data musí být reprezentativní a bez chyb.

- **Předzpracovaná data:**

Nutné provést čištění, normalizaci a kódování dat.

Supervizované strojové učení zahrnuje algoritmy, které se učí vztah mezi vstupními a výstupními proměnnými. Používá se například v klasifikaci, regresi a doporučovacích systémech. [1]

Částečně supervizované učení

Tento typ učení se pohybuje na pomezí supervizovaného a nesupervizovaného učení, kdy je k dispozici velké množství neoznačených dat a menší množství označených dat. Částečně supervizované učení využívá informace z obou typů pro vytvoření efektivnějšího modelu než v případě supervizovaného nebo nesupervizovaného učení samostatně.[2]

Nesupervizované učení

Algoritmy nesupervizovaného učení se soustředí na identifikaci skrytých struktur nebo vzorců v neoznačených datasetech. Tato kategorie algoritmů je zásadní pro úlohy, kde je potřeba detekovat anomálie nebo provádět segmentaci dat bez předchozího označení.[3]

Zpětnovazební učení

Zpětnovazební učení představuje přístup v oblasti strojového učení, kde model interaguje s daným prostředím a učí se na základě zpětné vazby, která může být ve formě odměn nebo trestů. Uplatnění nachází zejména v oblastech robotiky a hraní her. Cílem zpětnovazebního učení je vyvinout

optimální strategii chování, která maximalizuje kumulativní součet odměn získaných v průběhu interakce s prostorem. [1]

2.2. Klasifikace dat v obraze

V oblasti strojového učení je klasifikace dat v obraze klíčovou disciplínou, která má široké uplatnění v různých průmyslových a vědeckých oblastech, jako je zdravotnictví, bezpečnost, doprava a zemědělství [4]. Tato oblast se zabývá rozpoznáváním a kategorizací objektů a vzorců v digitálních obrazech.

Strojové učení, zejména metody jako konvoluční neuronové sítě (CNN), se ukázalo být velmi efektivní v řešení těchto úkolů [5]. CNN, inspirované biologickými procesy, se dokáží učit hierarchickým reprezentacím dat. Toto je zásadní pro práci s obrazy, které lze považovat za hierarchické struktury dat [6].

Fáze klasifikace dat v obraze se zabývá identifikací a kategorizací obsahu obrazových dat. Stěžejní aspekty této fáze zahrnují:

- předzpracování obrazu,
- extrakce rysů,
- trénování modelu,
- klasifikace.

Předzpracování obrazu

Předzpracování obrazu se soustředí na přípravu obrazu pro další zpracování. Zahrnuje odstranění šumu, normalizaci intenzity, kontrastní úpravy a další techniky, které zlepšují kvalitu obrazu pro analýzu. Cílem je odstranit nežádoucí variace v datech, které by mohly negativně ovlivnit proces klasifikace.

Extrakce rysů

Po předzpracování následuje extrakce rysů, kde jsou z obrazu identifikovány a extrahovány charakteristiky, které jsou relevantní pro klasifikaci. Tyto rysy mohou zahrnovat textury, tvary, barvy, kontury nebo jiné vizuální atributy. Účelem extrakce rysů je převést surová obrazová data na formu, která je vhodnější pro analýzu a klasifikaci.

Trénování modelu

Ve fázi trénování modelu se používají algoritmy strojového učení k vytvoření modelu, který dokáže rozpoznat a klasifikovat různé kategorie nebo objekty v obraze. Model se trénuje na sadě označených dat, kde je každá část snímku přiřazena ke konkrétní kategorii nebo třídě.

Klasifikace

Po trénování modelu se provádí klasifikace nových, neviděných obrázků. Model používá naučené vzory a charakteristiky k identifikaci a klasifikaci těchto snímků do relevantních kategorií. [7]

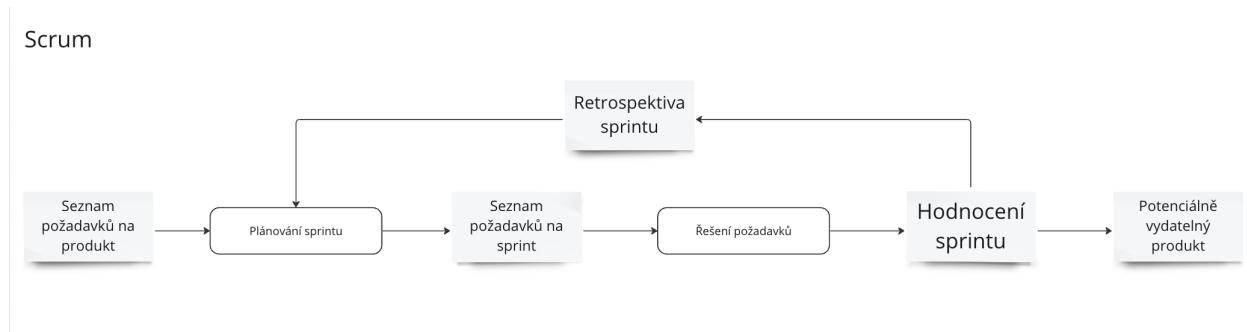
2.3. Workflow management

Workflow management je termín používaný pro definování procesu opakujících se činností. Zahrnuje popis procesu pro návrh, vytvoření, provedení a sledování pracovních postupů. Jeho úlohy zahrnují:

- nastavení a úpravy kroků procesu,
- řešení problémů,
- přizpůsobení se měnícím podmínkám,
- vytváření dokumentačních požadavků,
- informování zainteresovaných stran,
- udržování standardizovaného komunikačního systému.

Důležitost workflow managementu spočívá v prevenci ztráty času a zdrojů, která může být způsobena neefektivními pracovními postupy, a má významný dopad na úspěšnost a efektivitu v jakémkoli odvětví. Správně nastavený systém workflow managementu vede k minimalizaci a efektivnějšímu průběhu procesů. [8]

Příkladem workflow může být plánování v rámci později zmíněné agilní metodiky vývoje – scrum viz obrázek 2.1.



Obrázek 2.1.: scrum

Metody workflow managementu

Vizuální reprezentace a modelování

Workflow management často zahrnuje použití vizuálních reprezentací. Například diagramy toku práce, které ukazují sekvenční a paralelní procesy. Tyto modely mohou zahrnovat šipky a další

grafické prvky pro znázornění toku informací a pracovních úkolů. Vizualizace pracovních procesů umožňuje lepší porozumění a identifikaci potenciálních úzkých míst a zlepšení procesů.

Rozlišování rolí a odpovědností

V rámci workflow managementu jsou rozlišovány různé role. V kontextu této práce jsou to role účetní, anotátora, supervizora a administrátora, každá s vlastními odpovědnostmi a úkoly. Správná koordinace těchto rolí je nezbytná pro efektivní tok práce a zajištění, že každý úkol je správně adresován a vykonáván.

Integrace technologie a informačních systémů

Moderní workflow management často zahrnuje integraci technologických nástrojů a informačních systémů, jako jsou systémy pro správu dokumentů, databáze a softwarové aplikace. Tyto nástroje a systémy usnadňují automatizaci, sledování a analýzu pracovních procesů, což vede ke zvýšení efektivity a snížení chyb v rámci organizace. [9]

2.4. UX - Uživatelská zkušenost

UX (z anglického user experience) je klíčovým faktorem při vytváření interaktivních produktů a aplikací. Odráží to, jak se uživatel cítí při interakci s produktem, systémem nebo službou. Zahrnuje vnímání užitečnosti, jednoduchosti použití a efektivity. Správné navržení UX je důležité pro většinu společností, designérů a tvůrců při vytváření a zdokonalování produktů.

Podle Nielsen Norman Group 'uživatelská zkušenost' zahrnuje všechny aspekty interakce koncového uživatele se společností, jejími službami a produkty. Mezinárodní standard ISO 9241 definuje UX jako "vnímání a reakce uživatele vzniklé z používání nebo očekávaného používání systému, produktu nebo služby". [10]

Raný vývoj UX lze vysledovat až do strojové éry 19. a počátku 20. století. Frederick Winslow Taylor a Henry Ford byli průkopníky ve výzkumu efektivity práce a nástrojů. Termín UX byl zaveden Donaldem Normanem v polovině 90. let. [11]

2.5. Uživatelské role

Role-based access control (RBAC), neboli řízení přístupu založené na uživatelských rolích, je zásadním prvkem v oblasti bezpečnosti počítačových systémů. Jeho cílem je omezit přístup k systémovým zdrojům jen pro autorizované uživatele, a to na základě specifických rolí a oprávnění přiřazených těmto rolím. RBAC implementuje povinné (MAC) nebo diskreční (DAC) přístupové kontroly.

- **Mandatory Access Control (MAC):**

Přístupová kontrola typu MAC je založena na pevně stanovených pravidlech, která určuje, jaký přístup mají uživatelé k určitým systémovým zdrojům. V MAC systémech nelze pravidla přístupu jednoduše měnit bez specifických oprávnění. Tento přístup je typický pro prostředí vyžadující vysokou míru bezpečnosti, jako jsou vojenské nebo vládní organizace.

- **Discretionary Access Control (DAC):**

Diskreční přístupová kontrola umožňuje uživatelům s určitými oprávněními určovat a měnit pravidla přístupu. Uživatelé tak mají větší kontrolu nad tím, kdo může přistupovat k jejich datům. Tento systém nabízí větší flexibilitu, ale může přinášet vyšší bezpečnostní rizika kvůli možnosti individuálního nastavení přístupových práv.

RBAC definuje role pro různé pracovní funkce v organizaci a přiřazuje oprávnění těmto rolím. Uživatelé získávají oprávnění prostřednictvím svých rolí, které zjednodušují správu práv a bezpečnostní politik. RBAC uplatňuje tři základní pravidla:

- přiřazení role,
- autorizace role,
- autorizace oprávnění.

Tyto principy zajišťují, že uživatelé mohou využívat pouze oprávnění, pro která jsou autorizováni.

Význam RBAC pro skrývání prvků aplikace

RBAC umožňuje znepřístupnit prvky aplikace, které daný uživatel nemá oprávnění vidět. Příkladem může být skrytí prvku z uživatelského rozhraní, nebo deaktivace tlačítka nebo elementu. Schopnost znepřístupnit prvky je zásadní pro udržení bezpečného a efektivního aplikačního prostředí, kde má každý uživatel přístup jen k těm informacím a funkcím, které jsou pro jeho roli určené.

Role v kontextu informačních systémů lze popsat jako definovaný soubor přístupových oprávnění, který určuje, co může uživatel v systému dělat. Každá role může zahrnovat různá oprávnění, například čtení, úpravy, vytváření nebo mazání konkrétních dat. Uspořádání do rolí umožňuje efektivně spravovat přístupová práva různých uživatelů, neboť každé oprávnění nemusí být přidělováno individuálně, ale může být součástí předdefinované role.

Různé role mohou obsahovat stejná oprávnění. Například, role "anotátor" a "administrátor" mohou obě zahrnovat oprávnění k editaci anotací, ale role administrátor může mít navíc oprávnění k správě uživatelských účtů, zatímco role anotátora ne. Umožňuje větší flexibilitu a atomickou kontrolu nad tím, co jednotlivé role ve skutečnosti umožňují.

Role-based access control (RBAC) představuje efektivní metodu pro správu přístupu k systémovým zdrojům a je zásadní pro zajištění bezpečnosti a efektivity v moderních informačních systémech. Omezení přístupu, k určitým částem aplikace pouze pro autorizované uživatele, je důležité pro

ochranu citlivých dat a zajištění bezpečného uživatelského prostředí. RBAC rovněž napomáhá udržovat konzistenci a stabilitu aplikace tím, že zabraňuje uživatelům bez potřebných oprávnění a znalostí provádět změny v nastavení nebo konfiguraci. [12]

2.6. Metodika vývoje

Pro tento projekt byl zvolen agilní přístup vývoje. Agilní metodika vývoje představuje moderní přístup k softwarovému inženýrství, který klade důraz na flexibilitu, spolupráci a schopnost přizpůsobit se měnícím se podmínkám. Agilní vývoj je založen na principech obsažených v Agilním Manifestu pro Softwarový Vývoj a zdůrazňuje následující body: [13] [14]

- **Rychlé a iterativní vývojové cykly:**

Projekt je organizován do krátkých časových úseků nazývaných "sprinty," během nichž vývojový tým pracuje na konkrétních úkolech a funkcích. Tento přístup umožňuje flexibilní adaptaci k novým požadavkům a změnám.

- **Minimal viable product (MVP):**

Neboli minimalní životaschopný produkt je koncept vývoje produktu, který se zaměřuje na vytvoření produktu s dostatečnými funkcemi, které splňují základní potřeby uživatelů a umožňují shromáždění zpětné vazby pro další iterace a vylepšení. Příkladem může být zadání výroby automobilu. Místo toho, aby byl zákazník nespokojený do doby, než se automobil celý vyrobí, tak se mu nejdříve dodá koloběžka. Poté pokračuje vývoj, v další iteraci dostane motorku a nakonec auto. Stejný postup se dá aplikovat při vývoji aplikací.

- **Spolupráce a komunikace:**

Klade silný důraz na týmovou spolupráci a komunikaci mezi všemi zainteresovanými stranami, včetně zákazníků a uživatelů. Pravidelná komunikace a spolupráce přispívají k lepšímu porozumění požadavkům a zajišťují, že výsledný produkt nejlépe odpovídá potřebám uživatelů.

- **Flexibilita:**

Agilní metodologie vývoje poskytuje prostor pro adaptaci směru projektu v reakci na nově získané informace. Tato schopnost je obzvláště cenná v případech, kdy jsou požadavky nejasné nebo se mění.

- **Schopnost reagovat na změny:**

Agilní vývoj poskytuje rámec pro rychlé reagování na změny, který je důležitý pro projekty, kde se požadavky mohou vyvíjet nebo rozšiřovat s časem.

2.7. Použité technologie

Pro vývoj této aplikace byly vybrány technologie, které umožňují vytvoření výkonné a uživatelsky přívětivé aplikace. Následující sekce popisuje hlavní technologie použité při vývoji.

Node.js

Node.js je asynchronní běhové prostředí, které je řízeno událostmi a umožňuje vývoj a provoz serverových aplikací v jazyce JavaScript. Tento framework¹ nabízí výhodu použití jednoho programovacího jazyka pro vývoj na straně klienta i serveru. Node.js využívá koncept asynchronního programování, což je technika, která umožňuje provádění potenciálně časově náročných úkolů bez nutnosti blokovaní běhu programu a čekání na jejich dokončení. Díky tomu může aplikace reagovat na další události a zpracovávat výsledky úkolů, jakmile jsou k dispozici. [15]

Mezi časově náročné úkoly, které Node.js umožňuje zpracovávat asynchronně, patří například:

- **Čtení z databáze:**

Získávání dat z databáze může být proces s významnou časovou náročností, ovlivněný faktory jako je zpoždění sítě při připojení k databázovému serveru, nebo komplexností samotného dotazu.

- **Čtení ze souborového systému:**

Proces čtení dat ze souborů uložených na pevném disku může být značně časově náročný. Tato časová náročnost je způsobena různými faktory, včetně rychlosti samotného disku, efektivity souborového systému a také velikosti a formátu čtených dat.

- **Komunikace se vzdáleným serverem:**

Komunikace s jinými servery nebo externími službami přes síť může podléhat nepředvídatelným zpožděním, která jsou důsledkem různých faktorů v síťovém prostředí. Tato zpoždění mohou být způsobena omezenou propustností síťového připojení, vysokou zátěží na komunikačních uzlech, případně fyzickou vzdáleností mezi komunikujícími systémy.

Vue.js

Vue.js je progresivní JavaScriptový framework vyvinutý Evanem Youem, který byl poprvé představen v únoru 2014. Tato knihovna se zaměřuje na deklarativní vykreslování a skládání prvků. Vývojářům umožňuje efektivně vytvářet uživatelská rozhraní ze znova použitelných komponent.

Vue.js také poskytuje podporu pro jednosměrné i dvousměrné datové vazby, což usnadňuje správu stavu aplikace. Tento rys umožňuje vývojářům jednoduše sledovat a aktualizovat data ve své

¹Framework je strukturovaná sada nástrojů a knihoven v softwarovém inženýrství, která poskytuje předdefinovanou šablonu pro vývoj a nasazování aplikací, zvyšuje opakovanou použitelnost kódu a usnadňuje vývojářům práci díky standardizovaným konvencím a modulům.

aplikaci. Dalším významným prvkem Vue.js je jeho schopnost snadné integrace s různými projekty a knihovnami. To dává vývojářům flexibilitu používat jen ty části kódu, které potřebují pro svůj konkrétní projekt.

Vue.js se rychle stává oblíbeným frameworkm pro vývoj moderních webových aplikací díky své jednoduchosti a efektivitě při tvorbě uživatelských rozhraní.[16]

Architektura aplikací ve Vue.js

Pro tuto aplikaci, která je popisována v této bakalářské práci, byla zvolena architektura jednostránkové aplikace (SPA - Single-Page Application). Jedná se o moderní přístup vývoje webových aplikací, který využívá technologii JavaScriptu. SPA nabízí uživatelům přívětivější uživatelské rozhraní a rychlejší odezvu ve srovnání s klasickými webovými aplikacemi.

SPA se odlišuje od tradičních webových stránek tím, že veškerý obsah je načten a zobrazen na jediné stránce, a to za použití různých technik JavaScript API, jako jsou ajax, fetch a manipulace s DOM. Tímto způsobem může dynamicky aktualizovat obsah bez nutnosti opakovaného načítání celé stránky ze serveru. Data a celé uživatelské rozhraní jsou ukládány a spravovány lokálně. SPA rovněž poskytuje možnost routování, které umožňuje změnu adresy URL v prohlížeči podle aktuálního stavu aplikace. Tento přístup se liší od klasických webových aplikací, kde se při každé interakci s aplikací musí stahovat a načítat nové stránky ze serveru.

Je však třeba zdůraznit, že jedna z nevýhod použití SPA spočívá v horší možnosti indexování obsahu vyhledávači a optimalizaci pro vyhledávače (SEO²). Další výzvou je implementace správy navigace v rámci aplikace a sledování výkonu, pro co nejlepší uživatelský zážitek. [17]

Paper.js

Pro výběr knihovny pro 2D editor byla nezbytná volba takového nástroje, který podporuje manipulaci s vektorovou grafikou, zejména s ohledem na jednoduchou konverzi do formátu GeoJSON. Vektorová grafika se zakládá na matematických vztazích a křivkách, což umožňuje uchovat kvalitu a detaily obrazu i při různých úrovních zvětšení.

Pro dosažení stanoveného cíle byla zvolena knihovna Paper.js z důvodu jejího jednoduchého použití a rozšiřitelnosti. Tato open-source knihovna je postavena na technologiích HTML5 Canvas, což je HTML5 prvek umožňující vykreslování grafiky pomocí JavaScriptu. Paper.js poskytuje objektový model dokumentu a umožňuje snadné vytváření prvků, jako jsou body, trojúhelníky, křivky a cesty. Mezi důležité funkce knihovny Paper.js patří také poskytování rozhraní pro zachycení různých událostí, včetně kliknutí nebo pohybu myši, stisku klávesy a dalších interakcí. [18]

²SEO, nebo "Search Engine Optimization" je soubor praktik a technik zaměřených na optimalizaci webových stránek pro vyhledávače s cílem zvýšit jejich viditelnost a pozici ve výsledcích vyhledávání.

Git

Git představuje distribuovaný systém správy verzí, který je navržen k efektivnímu sledování změn v souborech a koordinaci práce mezi programátory během procesu vývoje softwaru. Jeho hlavními cíli jsou zajištění vysoké rychlosti, zabezpečení integrity dat a podpora pro distribuované, ne-lineární pracovní postupy. Klíčovou vlastností toho nástroje je schopnost vytvářet větve, díky kterým mohou programátoři pracovat na rozdílných funkcích nezávisle a následně tyto větve efektivně sloučit do hlavní větve projektu. [19]

Git si získal širokou popularitu a je aplikován v různých odvětvích, včetně podnikání, financí, zdravotnictví a vědeckého výzkumu, pro správu a verzování kódu. Jako open-source software je volně dostupný pro užití a modifikaci. Otevřený kód podporuje rozšíření a adaptabilitu v různých prostředích. [20]

V této bakalářské práci byla zmíněná knihovna využita pro verzování kódu vyvíjené aplikace. Vedle toho se uplatnila také pro verzování anotací. Demonstruje to její flexibilitu a rozmanité využití v oblasti softwarového inženýrství. [21]

MySQL

MySQL je open-source relační databázový systém (RDBMS), jehož název vznikl spojením jména "My", což je jméno dcery spoluzakladatele Michaela Wideniuse, a akronymu "SQL" pro Structured Query Language. Tento relační databázový systém umožňuje organizovat data do jedné či více datových tabulek, ve kterých mohou být data vzájemně propojena, což napomáhá efektivně strukturovat data. MySQL je kompatibilní s řadou operačních systémů, včetně Linuxu, macOS a Windows, a podporuje programovací jazyky jako C a C++.

Nejnovější verze MySQL Server 8.0 byla oznámena v dubnu 2018 a přinesla řadu významných inovací. Mezi tyto novinky patří NoSQL Document Store, atomické a bezpečné DDL příkazy a rozšířená JSON syntaxe. Dále nabízí vylepšené možnosti řazení a částečné aktualizace. Tato verze zaznamenala významné vylepšení v oblastech výkonu, bezpečnosti a správy. V roce 2019 byl MySQL vyhlášen DBMS roku podle hodnocení DB-Engines. [22]

GeoJSON

GeoJSON je otevřený standardní formát navržený pro reprezentaci geografických datových struktur s využitím JavaScript Object Notation (JSON). Tento formát umožňuje reprezentovat jak prostorové regiony (Geometrie), tak prostorově ohraničené entity (Prvky), nebo seznamy prvků (Kolekce prvků). GeoJSON podporuje široké spektrum aplikací, od webového mapování po navigační systémy.

GeoJSON definuje následující typy geometrie:

- bod (Point),

- liniový řetězec (LineString),
- vícebod (MultiPoint),
- víceliniový řetězec (MultiLineString),
- vícepolygon (MultiPolygon),
- kolekce geometrií (GeometryCollection).

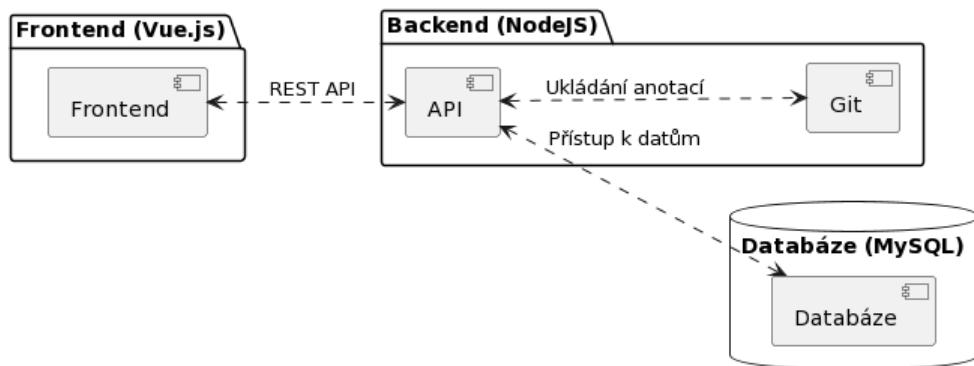
Prvky v GeoJSON obsahují geometrický objekt a další vlastnost. Kolekce prvků obsahuje seznam prvků.

Pracovní skupina pro formát GeoJSON byla zahájena v březnu 2007 a specifikace formátu byla dokončena v červnu 2008. V dubnu 2015 založila Internet Engineering Task Force pracovní skupinu Geographic JSON, která v srpnu 2016 vydala GeoJSON jako RFC 7946. Od té doby si formát GeoJSON získal širokou popularitu a je využíván v mnoha webových technologiích a aplikacích. [23]

3. Aplikačně-implementační návrh

3.1. Struktura aplikace

Aplikace je strukturována do tří hlavních komponent, jak je znázorněno na UML diagramu komponent (viz obrázek 3.1). Uživatelské rozhraní je implementováno prostřednictvím aplikace vytvořené ve frameworku Vue.js. Serverová část aplikace je realizována v Node.js a komunikuje s frontendem prostřednictvím HTTPS protokolu RESTful API¹. Pro efektivní ukládání a verzování postupu anotací byl použit lokální Git repozitář.



Obrázek 3.1.: Diagram komponent

V následující části je detailněji popsáné rozvržení uživatelského rozhraní. Aplikace má 4 hlavní části:

- projekty,
- datasety,
- supervize,
- správa uživatelů.

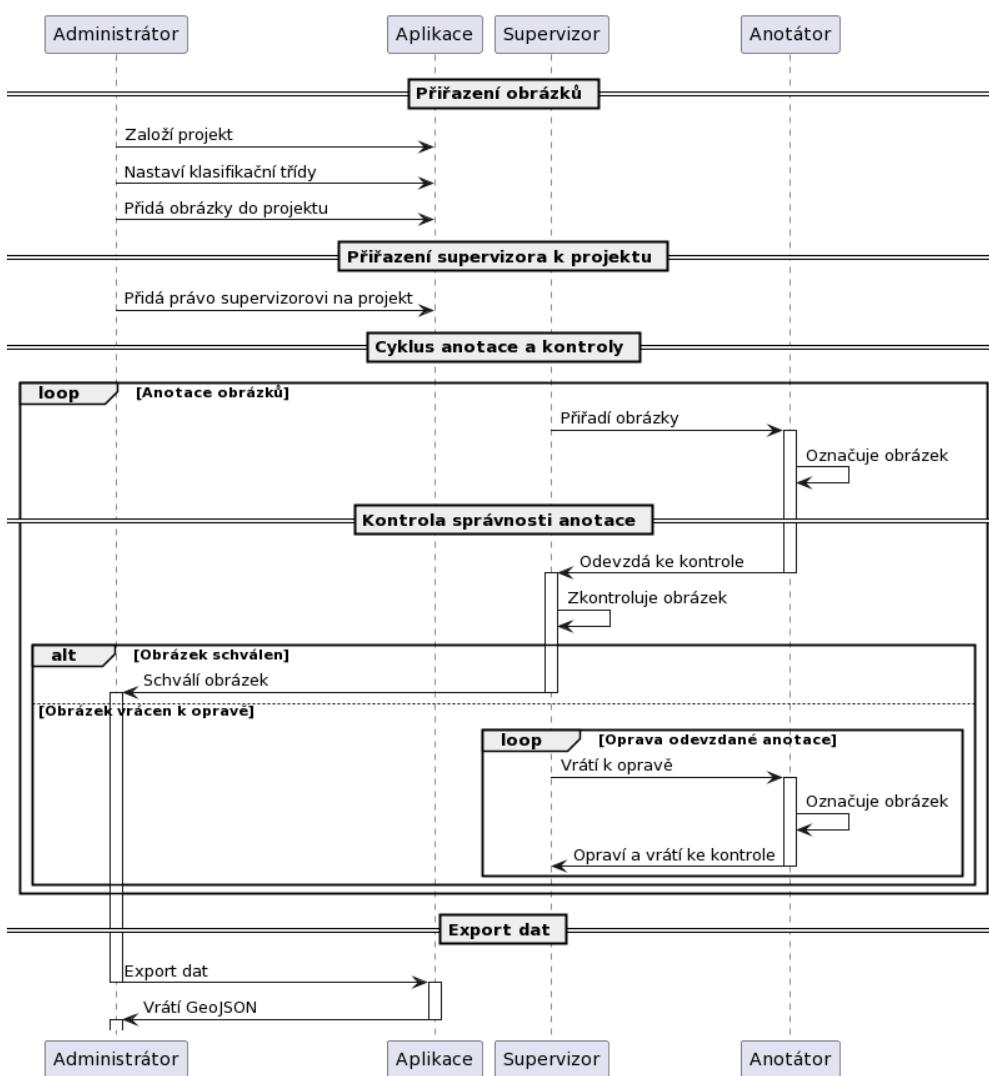
3.2. Analýza procesů

Analýza procesu anotace obrázků je zásadní pro pochopení celého procesu získávání dat. Proces je rozdělen do čtyř hlavních fází, s důrazem na praktické aspekty každého kroku. Proces je vizuálně

¹RESTful API (Representational State Transfer) je architektonický styl pro návrh síťových aplikací. Využívá HTTPS požadavky pro přístup a manipulaci s datovými reprezentacemi, čímž umožňuje snadnou a intuitivní komunikaci mezi klientem a serverem.

zobrazen sekvenčním diagramem, viz 3.2.

Hlavním důvodem pro definování procesu anotování obrázků je potřeba nastavit jej tak, aby byl co nejfektivnější a nejpřesnější. Pro každý projekt je nutné vybrat reprezentativní vzorek dat, který je následně anotován určenými třídami. Tyto třídy jsou později využity pro automatické anotování. Aby byla zajištěna vysoká kvalita a přesnost automatických anotací, je nezbytné, aby data byla nejprve správně a pečlivě označena člověkem. Vzhledem k možným chybám ze strany anotátorů je nezbytné do procesu zahrnout etapu kontroly, která zajišťuje, že data splňují všechny požadavky na kvalitu – například přesnost překryvů, pořadí vrstev a celková kvalita anotací. Tento kontrolní krok je realizován pracovníkem s příslušnými znalostmi, který zajišťuje, že data jsou vhodná pro další zpracování strojovým učením. Jakákoli chyba v počátečních anotacích může vést k nepřesnostem v automatickém označování pomocí konvolučních neuronových sítí (CNN).



Obrázek 3.2.: Analýza procesu anotace obrázků

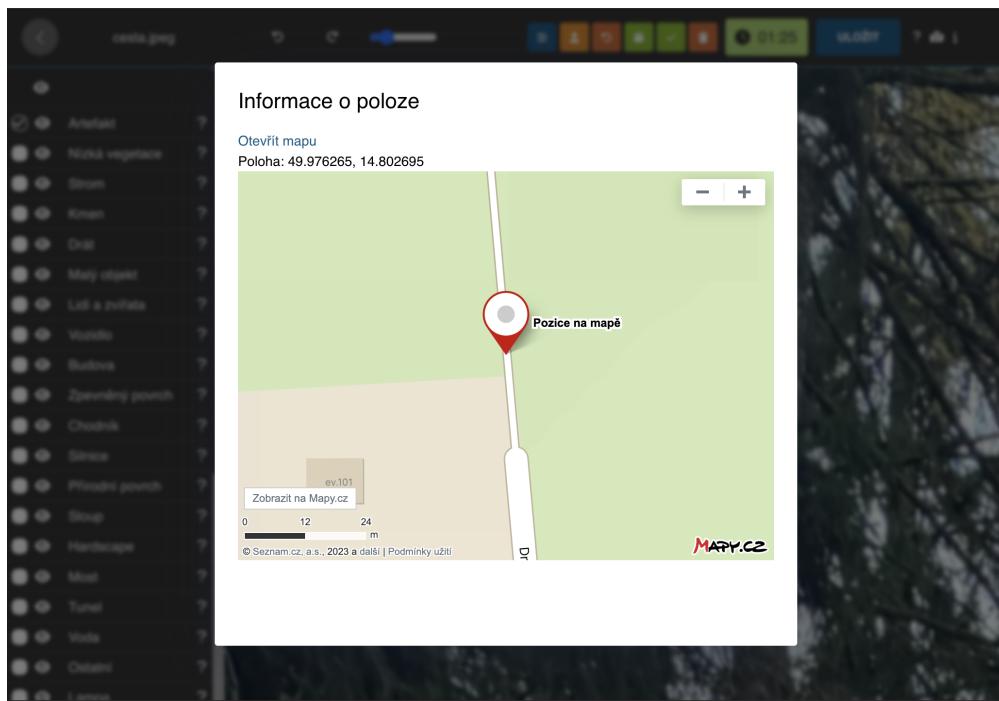
Příprava a nastavení projektu

Nejprve je nutné připravit správná data pro trénink modelu strojového učení, což závisí na specifických požadavcích na označená data. Tuto úlohu plní administrátor, který vybírá obrázky

odpovídající požadovaným anotačním třídám tak, aby zajistil, že data jsou reprezentativní a rozmanitá pro konkrétní trénovací úkol. Po výběru obrázků vytvoří administrátor nový projekt v aplikaci a definuje klasifikační třídy. Přesná definice těchto tříd je zásadní pro anotační fázi, neboť jakékoli chyby v této fázi mohou vést k nepřesnostem ve výsledcích strojového učení.

Přidělení úloh a začátek anotace

Po nastavení projektu přiřadí administrátor projekt supervizoru, který je zodpovědný za správu a dohled nad anotačním procesem. Supervizor poté nastaví přístupová práva pro anotátory a přidělí jim obrázky k anotaci. Anotátoři, vyškoleni v rozpoznávání a kategorizaci objektů na obrázcích, začnou anotovat podle definovaných tříd. Jejich úkolem je pečlivě označovat relevantní prvky na obrázcích, což vyžaduje pozornost k detailům a porozumění kontextu obrázků. Pro lepší pochopení kontextu jim mohou pomoci GPS souřadnice obrázků, ty může administrátor nastavit v datasetech. Po nastavení GPS souřadnic mají anotátoři možnost zobrazit vyskakovací okno s mapou, viz 3.3.



Obrázek 3.3.: Zobrazení modalu s polohou

Kontrola a iterace anotace

Jakmile anotátor úspěšně dokončí anotaci obrázku, je její výsledek předán supervizoru ke kontrole. Supervizor následně důkladně prochází každou anotaci, aby zaručil jejich přesnost a shodu s definovanými klasifikačními třídami. V případě zjištění chyb nebo nepřesností je obrázek vrácen anotátorovi k opravě. Vrácení k opravě obsahuje popis nalezené nepřesnosti. V nutných případech supervizor upřesní chybu anotátorovi přes chat s připojeným snímkem obrazovky. Proces kontroly se může opakovat několikrát, dokud nebude obrázek anotován správně.

Opakování kontroly je nezbytná pro zajištění kvality a spolehlivosti anotací, důležitých pro správnost tréninkových dat v oblasti strojového učení.

Finální schválení a export dat

Po schválení anotace supervizorem jsou obrázky předány zpět administrátorovi. Administrátor provádí konečnou kontrolu a dává souhlas k použití obrázků v tréninkovém datasetu. Jakmile administrátor schválí obrázky, se data vyexportují do formátu GeoJSON. Tento formát je široce využíván pro geoprostorová data a je vhodný pro účely strojového učení. Exportovaná data se následně používají jako dataset pro trénink algoritmů strojového učení.

3.3. Uživatelské role

V kontextu vyvíjené aplikace je zásadní implementace uživatelských rolí, díky kterým je možné dodržet efektivní kontrolu kvality práce a kontrolu přístupu do systému podle RBAC. Mezi hlavní role patří administrátor, supervizor, anotátor a účetní, přičemž každá z nich zastává specifickou funkci a odpovědnost. V následujícím textu bude podrobněji rozebrán význam a činnosti těchto rolí v rámci systému. Role jsou seřazeny podle oprávnění, od nejnižší po nejvyšší.

Anotátor

Role anotátora zahrnuje několik klíčových funkcí a pracovních postupů, které zajišťují efektivní zpracování obrázků a jejich anotací. Anotátor je uživatel, který edituje označení obrázků. Po přihlášení do aplikace vidí přehled jemu přiřazených obrázků k editaci. Pro zajištění efektivního zpracování obrázků by měl anotátor pracovat sekvenčně, což znamená, že může začít editovat další obrázek až po odevzdání aktuálně rozpracovaného ke kontrole.

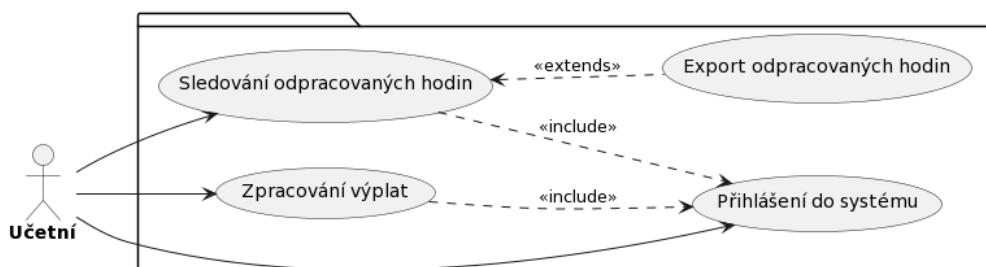
Pokud je anotátorovi vrácen obrázek k opravě, zobrazí se mu v přehledu se zvýrazněním obrázku, který má editovat prioritně. Může editovat a prohlížet pouze ty obrázky, které jsou mu aktuálně přiřazeny. Po odevzdání všech rozpracovaných obrázků může žádat o další soubory z fronty. Kliknutím na tlačítko v aplikaci se anotátorovi automaticky přiřadí obrázek, který nebyl nikým editován. Přiřazení obrázku se provádí na základě projektů, které má anotátor přiřazené. Vybere se obrázek z datasetu a projektu, který má nejvyšší prioritu.

Anotátor musí mít přístup k přehledu o operacích, které sám v systému provedl. Patří sem uložení anotace na server a odevzdání anotací ke kontrole supervizorem. Anotátor by měl mít možnost zobrazit si jednoduchý přehled, který si může filtrovat podle rozsahu dat. V tomto přehledu by měl vidět čas strávený anotováním, počet editovaných obrázků a počet projektů, ze kterých obrázky pocházely.

Anotátor nemá právo vidět informace o ostatních uživatelích a jejich projektech. Nemá možnost editovat ani prohlížet anotace, které mu nejsou aktuálně přiřazeny. Nemůže prohlížet projekty ani jejich historii.

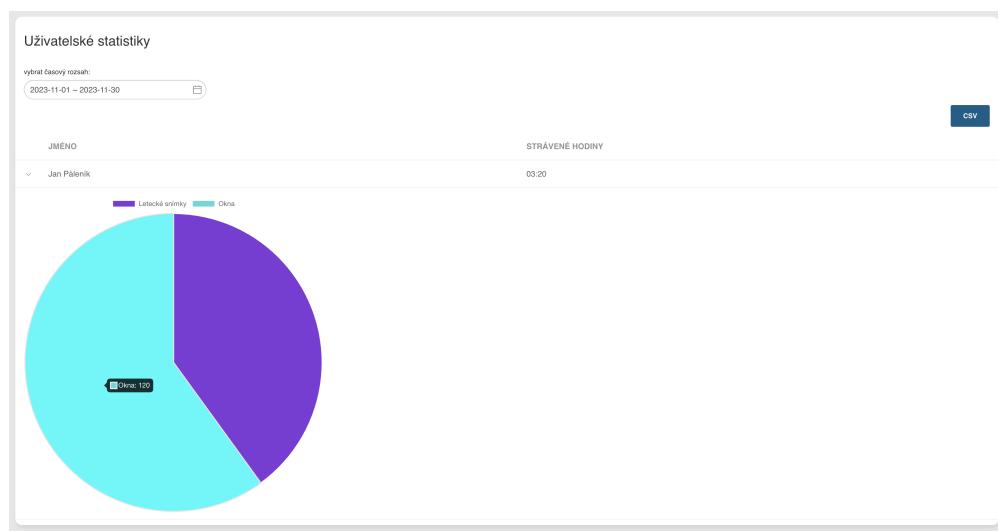
Účetní

Role účetní je v systému důležitá, neboť jsou anotátoři za svou odvedenou práci finančně odměňováni. Aby účetní mohl zpracovávat výplaty a spravovat kontaktní údaje anotátorů, musí mít přístup do aplikace. Přihlášení do aplikace mu umožňuje získat seznam všech uživatelů, zobrazit si a upravit jejich kontaktní informace. Pro správné zpracování výplat je důležité, aby účetní měl jednoduchý přehled o odpracovaných hodinách anotátorů za daný měsíc. Use case diagram učetní, viz 3.4.



Obrázek 3.4.: Use case diagram role účetní

Přehled odpracovaných hodin je zajištěn prostřednictvím tabulky viz tabulka 3.5, která obsahuje seznam všech uživatelů s celkovým součtem odpracovaných hodin. V rámci výplatního procesu je odpracovaný čas zaokrouhlen na nejbližší půlhodinu nahoru. Z důvodu zpracování výplat v účetním systému musí mít účetní možnost stáhnout přehled v CSV formátu (viz tabulka 3.1). Exportovaný soubor obsahuje jméno anotátora, e-mailovou adresu a součet odpracovaných hodin za vybrané období.



Obrázek 3.5.: Tabulka s přehledem odpracovaných hodin

Obsah souboru může být následovný:

Listing 3.1: Exportovaný výpis hodin

```
name,email,hours
```

Jan Páleník, jan.palenik@microsoft.example.com, 03:20

Jakub Marek, jakub.marek@microsoft.example.com, 12:20

Účetní má v systému omezený přístup, který zahrnuje pouze možnost zobrazovat kontaktní údaje uživatelů a mít přístup k sekci s přehledem odpracovaných hodin. Ovšem není oprávněn prohlížet projekty ani jiné části systému, které jsou spojené s anotacemi a označováním dat. Toto omezení reflektuje pracovní povinnosti účetní, které se zaměřují na účetnictví a sledování odpracovaného času, a nikoli na správu projektů či anotací.

Supervizor

Jednou z klíčových rolí supervizora je kontrola obrázků v projektech, které konkrétní supervizor spravuje. Supervizor nese zodpovědnost za ověření kvality anotací na odevzdaných označených obrázcích. V případě nedostatků může obrázky vrátit anotátorům k opravě. V opačném případě je schvaluje.

Tímto způsobem supervizor zajistí, že výsledné anotace dosáhnou nejvyšší možné kvality a budou splňovat požadavky projektu. Supervizor má také možnost upravovat prioritu souborů, které jsou již přiřazeny anotátorovi. Tato funkce umožňuje supervizorovi brát v úvahu důležitost jednotlivých souborů a přizpůsobit jejich pořadí v rámci anotátorovy fronty obrázků.

Supervizor může prohlížet projekty a jejich historii pouze v případě, že mu byli přiděleny k supervizi. Nemůže editovat nastavení projektu ani přidávat nové obrázky.

Administrátor

Administrátor má veškeré pravomoci ostatních rolí, což mu umožňuje efektivně spravovat a kontrolovat celý systém, zajišťovat kvalitu anotací a efektivně řídit práci anotátorů a supervizorů. Následující práva má navíc.

- Správa uživatelů a oprávnění k projektům:**

Může editovat oprávnění uživatelů, měnit jejich role, aktivovat či deaktivovat jejich účty a nastavovat povolené projekty pro jednotlivé uživatele. To zajišťuje, že anotátoři a supervizori mají přístup pouze k relevantním informacím a projektům.

- Práce s datasety:**

Administrátor může vytvářet nové datasety a nahrávat do nich obrázky. Pro lepší organizaci a přehlednost může ke každému typu datasetu definovat atributy, například GPS souřadnice, jejichž hodnoty může upravovat přímo v datasetu nebo u jednotlivých obrázků.

- Správa projektů:**

Administrátor je zodpovědný za vytváření projektů, přidávání obrázků z datasetů a vytváření seznamu tříd pro označování obrázků. Může také exportovat anotace a měnit prioritu projektů či datasetů v rámci projektu, což ovlivňuje frontu obrázků pro anotátory.

- **Supervize:**

Administrátor má možnost kontrolovat obrázky schválené supervizorem a rozhodovat o konečném schválení anotace.

3.4. Datasety

Pod pojmem "dataset" lze chápat soubor dat, která mají shodné nebo podobné vlastnosti vhodné pro trénink algoritmů strojového učení. Tato data obvykle představují rozsáhlé soubory, zejména obrázků, které obsahují opakující se vzory nebo struktury. Příklady takových datasetů mohou zahrnovat různé typy krajiny, budovy ve městech, přírodní prvky jako stromy nebo terénní útvary jako hory a řeky.

Pro úspěšné tréninkové algoritmy strojového učení je nezbytné, aby dataset obsahoval různé vzory a varianty objektů, které mají být klasifikovány. Například, pokud algoritmus má být naučen rozpoznávat budovy ve městě New York, měl by dataset obsahovat správně označené obrázky výškových budov, které jsou charakteristické pro tento konkrétní kontext. Naopak, pokud by algoritmus byl trénován pouze na těchto budovách, neměl by schopnost rozpoznat a klasifikovat jiné objekty, jako jsou pole v nizozemské krajině.

Je také důležité zdůraznit, že vstupní data mohou být velmi rozsáhlá, dosahující desítek gigabajtů. Z tohoto důvodu je důležité zajistit, aby data nebyla duplikována a bylo je možné efektivně organizovat a spravovat, což je klíčovým prvkem pro úspěšné tréninkové procesy strojového učení.

Vytvoření datasetu

Pro přidávání datasetů do aplikace je nejprve potřeba otevřít formulář pro vytvoření nového datasetu, viz obrázek 3.6 na následující straně.

Při vytváření datasetu je třeba vyplnit následující atributy:

- **Jméno:**

Uživatelský popis datasetu. Měl by být výstižný a stručný, aby uživatelé snadno identifikovali obsah datasetu.

- **Alias:**

Je standardizované technické jméno, používané pro export dat. Měl by být zvolen anglický název bez mezer nebo s podtržítky.

- **Typ datového souboru:**

Specifikuje typ souboru v datasetu, pro tento projekt je definován typ souboru *obrázek*. Počítá se s rozšířením o typ *3d obrázek*.

The screenshot shows a web-based form for creating a dataset. At the top, there's a header "Vytvořit dataset". Below it, there are several input fields:

- jmeno:** Les
- alias:** les
- typ datového souboru:** picture
- ikona:** A dropdown menu with "VYBRAT IKONU" highlighted, followed by a preview icon of a tree.
- Popis:** A rich text editor toolbar with icons for Normal, Bold, Italic, Underline, etc.
- Snímky lesních dříst:** A placeholder text area.

Below these is a section titled "Atributy" containing attribute definitions:

- Lat: not exists, latitude
- Lon: not exists, longitude
- Location: not exists
- Resolution [cm]: not exists

At the bottom right of the form is a blue "VYTVOŘIT" button.

Obrázek 3.6.: Formulář pro vytvoření datasetu

- **Ikona:**

Slouží pro lepší vizuální rozpoznání datasetu uživatelem. Ikona je vybírána prostřednictvím vyskakovacího okna s výběrem desítek ikon z knihovny Font Awesome².

- **Popis:**

Poskytuje dodatečné informace o datasetu, například jeho účel nebo zdroj.

- **Atributy:**

Nastavení hodnot definovaných atributů. Detailněji jsou popsány v následující sekci.

Atributy datasetu a obrázků

Slouží k definování specifických informací o datasetu jako jsou rozlišení obrázků, zeměpisné souřadnice a další. Hodnoty těchto atributů lze nastavit pro celý dataset, přičemž jsou automaticky zahrnutы в exportovaných anotacích ve formátu JSON. Zajímavým aspektem je možnost individuální úpravy těchto atributů u každého jednotlivého obrázku v datasetu.

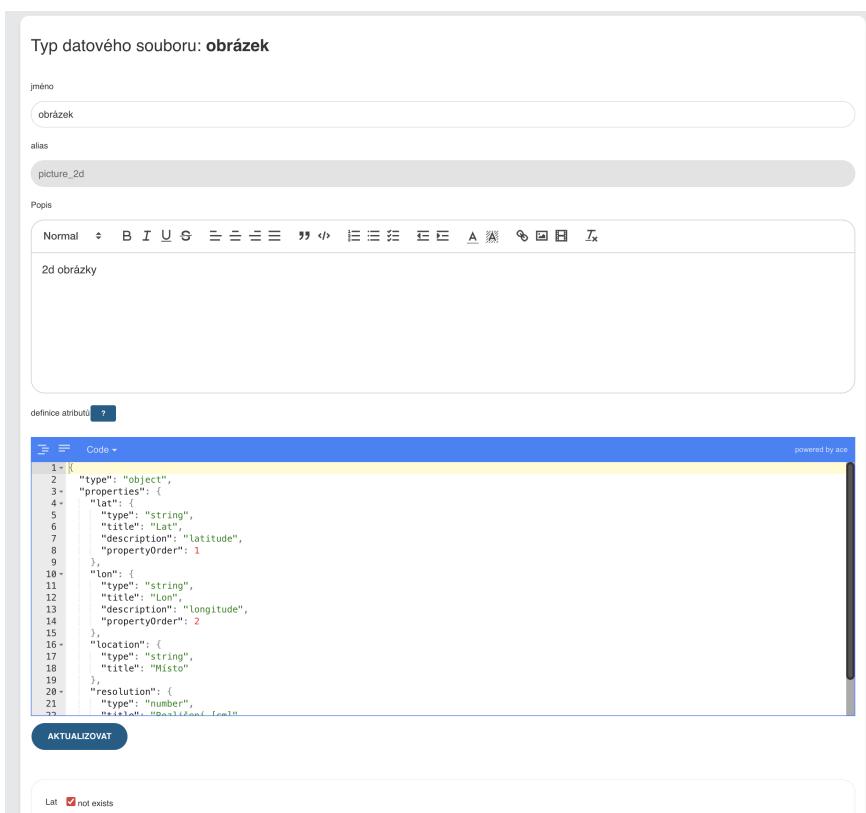
Tato flexibilita je zásadní, neboť některé atributy, jako jsou zeměpisné souřadnice, se mohou lišit obrázek od obrázku. Naopak rozlišení snímků by mělo být konzistentní napříč celým datasetem.

²Font Awesome je populární knihovna ikon, která poskytuje širokou škálu vektorových ikon a sociálních log. Tyto ikony lze snadno integrovat do webových a mobilních aplikací. Více informací naleznete na oficiálních stránkách: <https://fontawesome.com/>

Administrátor má možnost tyto atributy definovat a upravovat prostřednictvím následujícího formuláře, jak je ilustrováno na obrázku 3.7.

Pro potřeby definování specifických atributů odpovídajících požadavkům datasetu a aktuálním potřebám týmu pracujícího na projektech strojového učení bylo rozhodnuto využít knihovnu pro dynamické generování formulářů. Tato knihovna umožňuje definování a vykreslování formulářů na základě struktury definované ve formátu JSON.

Atributy a jejich specifikace jsou vytvářeny pomocí JSON formátu, což usnadňuje generování interaktivních formulářů s příslušnými atributy. Pro tyto účely byla vybrána knihovna `vue-form-generator`, která je dostupná na platformě npmJS. Tato knihovna poskytuje flexibilní a intuitivní nástroje pro tvorbu formulářů v aplikacích Vue.js, což umožňuje efektivní implementaci bez nutnosti dodatečných změn kódu. Pro více informací a dokumentaci knihovny `vue-form-generator` navštivte <https://www.npmjs.com/package/vue-form-generator>.



Obrázek 3.7.: Editace atributů datasetu

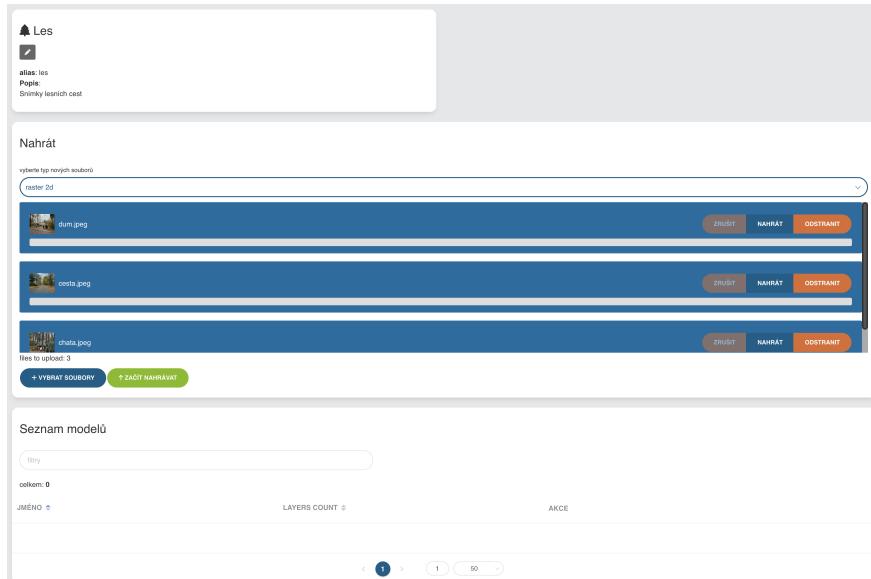
Nahrávání obrázků

Po úspěšném vytvoření datasetu je dalším krokem nahrání obrázků. Nahrávání probíhá následovně:

1. Nahrání obrázku:

Po otevření datasetu lze přidat obrázky prostřednictvím jednoduché uživatelské komponenty, do kterých je možné nahradit soubory přímo z disku počítače. Během nahrávání může

uživatel sledovat průběh nahrávání. V případě, že se administrátor pokusí nahrát soubor v nekompatibilním formátu, je mu zobrazena chybová hláška. Způsob nahrávání souborů je vidět na obrázku 3.8.



Obrázek 3.8.: Nahrávání obrázků

2. Prohlížení a editace atributů

Po úspěšném nahrání jsou soubory zobrazeny v tabulce pod kartou s nahráváním. Uživatel má možnost prohlížet nahrané soubory a editovat jejich atributy. Kliknutím na detail souboru se otevře modální okno. Například, pokud jsou nastaveny atributy zeměpisné šířky a výšky, zobrazí se v tomto okně mapa s bodem nastavených souřadnic.

Význam atributů

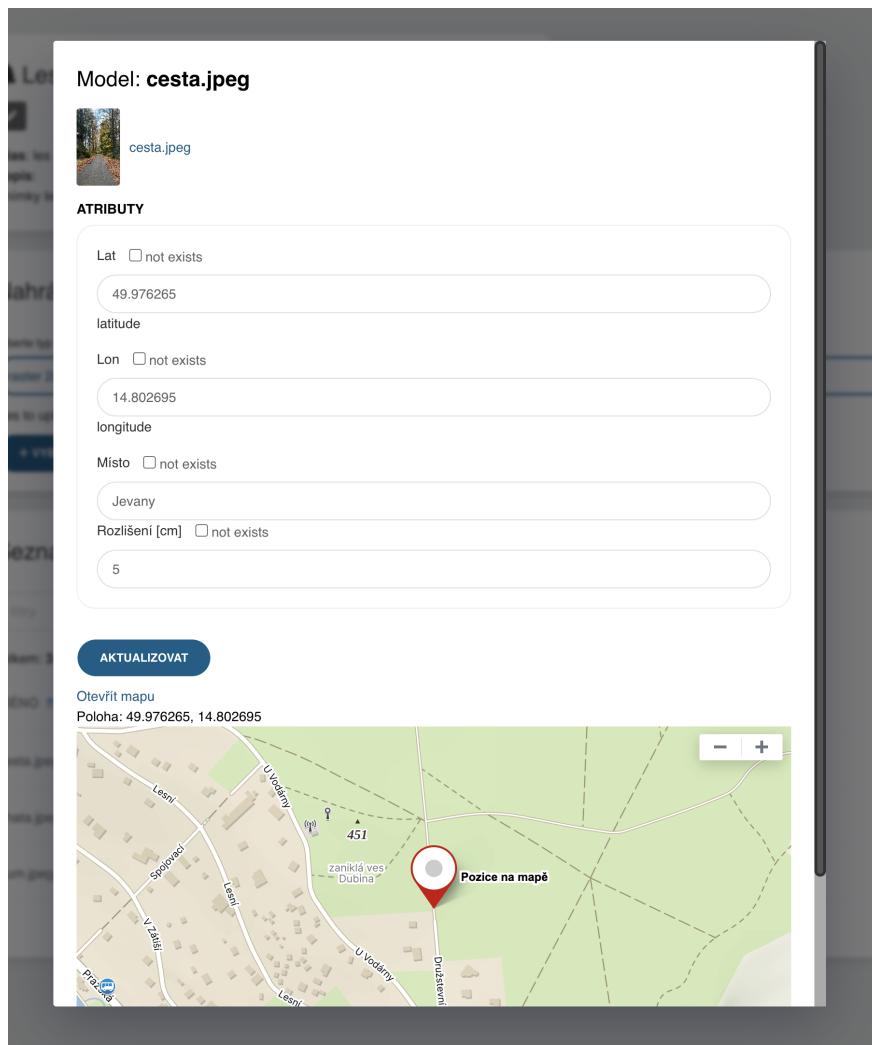
Atributy jsou klíčové pro kontrolu a zjednodušení použití, neboť anotátoři mohou místo pořízení fotografie snadno zobrazit v Google Mapách či jiných mapových službách, což napomáhá přesnéjšímu označování a určování tříd. Detailní náhled funkcionality je na ukázce 3.9 na následující straně.

3.5. Projekt

V rámci aplikace je klíčovým prvkem projekt. Projektem rozumíme soubor datasetů, které jsou označovány stejnou sadou klasifikačních tříd. Projekty slouží k organizaci a správě obrázků z různých datových sad, které mají být zpracovány a anotovány.

Důležitými činnostmi v rámci projektu jsou:

- přidávání obrázků do projektu,



Obrázek 3.9.: Náhled obrázku

- přiřazování anotátorům k editaci,
- správa klasifikačních tříd,
- nastavení priority jednotlivých datasetů z pohledu důležitosti označování,
- vyřazování anotací administrátorem,
- schvalování anotací administrátorem,
- kontrola anotací supervizorem,
- export dat.

V následující části budou tyto části podrobněji popsány.

Práce s projekty

Projekt lze vytvořit pomocí tří jednoduchých kroků, neboli průvodce (wizard).

V prvním kroku se o projektu vyplní základní informace, jako je *jméno* projektu a jeho *alias*, který slouží k exportům dat. Použití aliasu je vhodné zejména k exportu dat a pro jednodušší procházení

exportované anotace v příkazové řádce bez mezer.

Dále se vybírá typ projektu. Typy projektů a jejich rozdíly jsou následující:

1. Obecná segmentace

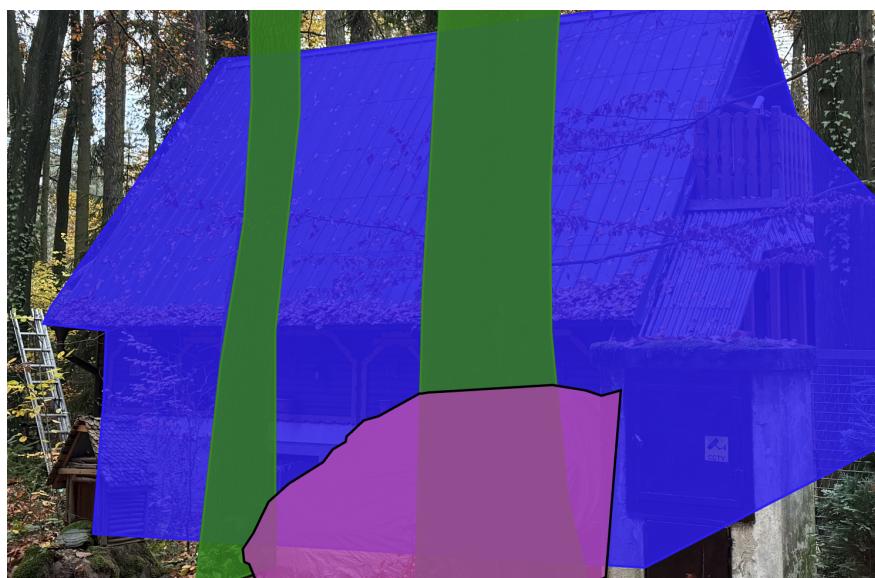
V obecné segmentaci obrázků jsou polygony dané klasifikační třídy uložené ve stejné vrstvě. Jednotlivým klasifikačním třídám lze nastavovat jejich pořadí (z-index).

2. Obrázková segmentace instancí

V projektu typu "obrázková segmentace instancí" má každý polygon definované pořadí nezávisle na klasifikační třídě. Jednotlivé polygony tedy mají definované pořadí vůči všem ostatním polygonům nezávisle na pořadí klasifikačních tříd.

Rozdíl mezi jednotlivými typy projektů je následující:

V obrázkové segmentaci je například klasifikační třída *strom* (zelená barva) ve druhém pořadí. Klasifikační třída *vozidlo* je v pořadí prvním, tedy před stromy. V tomto případě nelze rozlišit, jestli je dané auto na snímku před, nebo za stromem. Tento přístup má výhodu při trénování algoritmů, které mají za úkol pouze kategorizovat objekty (viz obrázek 3.10).

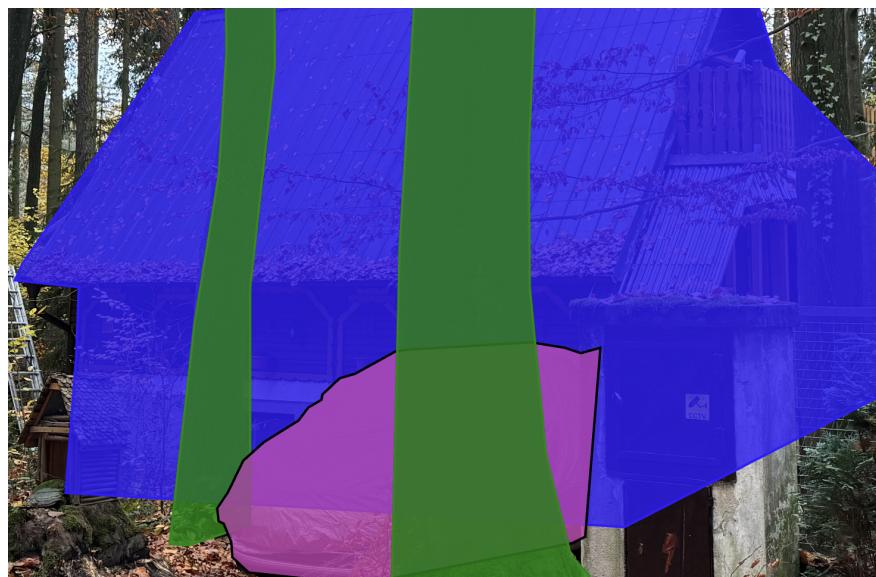


Obrázek 3.10.: Obecná segmentace

V případě obrázkové segmentace instancí kdy nejsou všechny polygony stejné třídy ve stejné vrstvě je možné určit pořadí jednotlivých objektů nezávisle na pořadí vrstvy. Rozdíl je patrný z uspořádání vrstev, kde pravý strom na obrázku 3.11 na následující straně je před označenou vrstvou vozidlo.

Příklad rozdelení projektů

Seskupení klasifikací do jednotlivých projektů je klíčové z důvodu možnosti specializace a přizpůsobení anotačního procesu konkrétním potřebám a cílovým využitím. Následují příklady projektů, které ilustrují tuto potřebu:



Obrázek 3.11.: Obrázková segmentace instancí

- **Letecké snímky:**

V projektu zaměřeném na letecké snímky jsou anotovány obrázky pořízené z výšky, například letadlem nebo dronem. Tyto snímky mají vysoké rozlišení, často v řádu centimetrů na pixel. Tento projekt je vhodný pro označování různých objektů, jako jsou budovy, střechy, silnice, stromy a zemědělská půda. Data z tohoto projektu lze využít pro automatickou kategorizaci objektů pro katastrální úřady, identifikaci stromů v oblasti nebo další aplikace, které vyžadují detailní informace z leteckých snímků.

- **Okna**

Klasifikace oken využívá uliční snímky. Uliční snímky jsou obvykle pořizovány pomocí vozidel vybavených sadou kamer, přesným GPS systémem a technologií Lidar. Klasifikační třídy zahrnují okna, rámy a dveře. Tento projekt umožňuje trénovat algoritmus na rozpoznávání a kategorizaci okenních prvků na budovách. Příkladem využití může být počítání okenních ploch na různých budovách, nebo zaměření určitých typů oken ve městském prostředí.

Přidávání obrázků do projektu

Do projektu je možné přidávat obrázky z různých datasetů. Každý obrázek může být v projektu nanejvýš jednou. Pro přidání obrázku do projektu existují dva přístupy:

- **Přidání neoznačeného obrázku:**

Neoznačený obrázek lze přidat rovnou z datasetu. Tento přístup je vhodný pro přidání nových obrázků, které ještě nebyly anotovány nebo není nastavené mapování tříd mezi projekty.

- **Přesun již označeného obrázku:**

V případě, že jsou již v projektu A označené obrázky, které se mají přidat do projektu B, může být vhodnější nastavit mezi projekty mapování tříd a anotace převést. Například, pokud v projektu A jsou třídy *listnatý strom*, *keř* a *jehličnatý strom*, a v projektu B je třída nazvaná pouze *vegetace*. Předpokládejme, že je nastavené mapování tříd, takže třídy *listnatý strom* a *jehličnatý strom* se mapují na třídu *vegetace*, zatímco třída *keř* nemá nastavené žádné mapování.

Při přesunu obrázků do projektu B dochází k následujícím změnám v klasifikačních třídách:

- Obrázky s třídami *listnatý strom* a *jehličnatý strom* jsou převedeny na třídu *vegetace* v rámci projektu B.
- Obrázky s třídou *keř*, která nemá nastavené žádné mapování, se po přesunu do projektu B ztratí.

Klasifikační třídy

Klasifikační třídy jsou v rámci projektu nezbytné pro rozpoznávání a kategorizaci objektů a prvků zobrazených na obrázcích. Tyto třídy umožňují uživateli systematicky a přesně identifikovat a označovat objekty, jako například střechy, okna, koleje, stromy, silnice a mnoho dalších.

V projektu je možné klasifikační třídy nadefinovat a seřadit podle vrstev, což umožňuje lepší vizuální reprezentaci vztahů mezi objekty. Například stromy mohou být nad silnicemi nebo střechy nad fasádou budovy. Třídy mohou mít dva typy - čáry nebo polygony, které slouží k označování obrázků.

Atributy klasifikačních tříd, jako jsou alias, jméno, typ třídy, barva ohraničení, šířka čáry, barva vykreslování a popis, jsou zásadní pro správné fungování tříd v rámci webové aplikace a některé z nich nutné k exportu. Atributy klasifikačních tříd jsou Jejich význam je následující:

- **Alias:**

Alias unikátní název v rámci projektu pro klasifikační třídu. Alias může být zkratka, nebo jiný název, který systému usnadní práci s třídami. V rámci projektu musí být unikátní.

- **Jméno:**

Jméno je označení klasifikační třídy, které slouží k její jednoznačné identifikaci v rámci projektu.

- **Typ třídy:**

Typ třídy určuje, zda se jedná o čáru nebo polygon. Tento atribut je důležitý pro správné zobrazení a export tříd v rámci projektu. Příkladem může být označování oken. Pro označení plochy okna se použije třída typu polygon. Aby bylo možné jednoduše označit i rámy daného okna, tak se pro třídu *rám* použije typ třídy čára.

- **Barva ohraničení:**

Barva ohraničení určuje, jakým způsobem bude zobrazeno ohraničení dané třídy na obrázku. Tato barva je důležitá pro vizuální identifikaci třídy uživatelem, ale nemá vliv na další zpracování.

- **Šířka čáry:**

Šířka čáry určuje tloušťku ohraničení klasifikační třídy. Tento atribut pomáhá uživateli lépe rozlišit mezi jednotlivými třídami na obrázku. Stejně jako barva ohraničení nemá vliv na další zpracování.

- **Popis:**

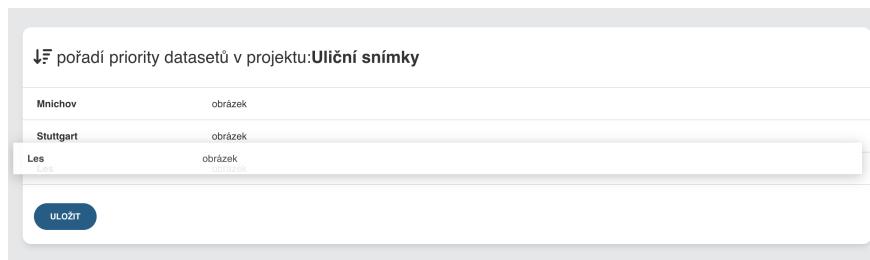
Popis klasifikační třídy poskytuje další informace o třídě a jejím účelu. Tento atribut může být užitečný pro uživatele, kteří nejsou obeznámeni s daným projektem nebo pro zdokumentování specifických informací o třídě.

- **Barva výplně:**

Nastavení barvy výplně se zobrazí pouze pro třídy typu polygon. Určuje barvu, kterou bude mít vnitřní prostor polygonu. Tato barva musí obsahovat transparentnost, což umožňuje vidět prvky nacházející se pod polygonem.

Prioritizace datasetů v rámci projektu

Efektivní řízení a organizace datasetů jsou klíčovými aspekty jakéhokoli datově orientovaného projektu. Zásadním prvkem je nastavení priorit těchto datových sad. Proces nastavování priorit je vyřešen pomocí drag and drop komponenty, ve které je zobrazen seznam všech datasetů v projektu, viz 3.12. Priorita se nastavuje od nejvyšší po nejnižší, přičemž dataset s nejvyšší prioritou je na vrcholu seznamu.



Obrázek 3.12.: Nastavení priority datasetů v projektu

Administrátoři mohou každou datovou sadu řádně a jednoduše zařadit podle důležitosti. Prioritizace má zásadní význam pro automatické přiřazování obrázků anotátorům. Systém s dobře nastavenými prioritami může efektivně a logicky rozdělovat obrázky k anotaci. Tento postup optimalizuje využití času a zdrojů, což přispívá k účinnosti celého projektu.

3.6. 2D Editor

2D editor představuje klíčový a nepostradatelný prvek v rámci aplikace, jelikož umožňuje provádět úkoly spojené s anotací obrázků. Jak již bylo zmíněno, tento editor byl implementován s využitím knihovny Paper.js. Jeho struktura je rozvržena do tří hlavních částí, které jsou navrženy s ohledem na uživatelskou přívětivost a efektivitu práce, kterou mimo jiné zrychlují klávesové zkratky.

Postup anotace

Pro každý projekt jsou vybrány reprezentativní datové sady a klasifikační třídy. Anotátoři začínají anotovat snímky, které nebyly dříve označeny, nebo které byly předem klasifikovány. Postupně anotují každou klasifikační třídu zvlášť. Každá klasifikační třída má své definované pořadí. Anotace vrstev probíhá směrem od objektů nacházející se v pozadí až po objekty nacházející se v popředí. Každý objekt musí být označen s dostatečnou přesností, aby byla zajištěna kvalita anotací.

Navigační panel

Navigační panel poskytuje přístup k editačním nástrojům pro manipulaci s polygony a čárami ve snímcích. Dále zahrnuje možnost kroku zpět a vpřed při práci s anotacemi. V pravé části navigačního panelu jsou následující funkce:

- **Uložit anotaci:**

Anotátoři mohou uložit svůj postup přímo na server pro uchování dat. Změny v anotaci se z důvodu menší pravděpodobnosti ztráty dat uchovávají v paměti webového prohlížeče.

- **Zobrazit návod projektu:**

Po kliknutí na ikonu *i* se zobrazí ve vyskakovacím okně návod k projektu. Návod obsahuje instrukce pro správnou anotaci dat.

- **Zobrazit návod editoru:**

Umožňuje rychlý přístup k návodu editoru. V této návodi jsou popsány klávesové zkratky k editačním nástrojům.

Pohled přiřazeného anotátora k obrázku na navigační panel, viz 3.13. Anotátor nemá přístup k panelu supervize z důvodu řízení právem – nemůže přiřadit anotaci jinému uživateli nebo ji schvalovat.



Obrázek 3.13.: Navigační panel - anotátor

Pro supervizory jsou k dispozici další funkce:

- **Časomíra:**

Supervizoři mohou sledovat čas strávený v editoru. Tato informace je pro anotátory skryta, z důvodu předcházení možnému nastavování času.

- **Workflow management:**

Supervizoři mohou provádět akce související s řízením pracovního postupu, například přiřazovat obrázky anotátorům nebo schvalovat anotace.

- **Časová osa:**

Supervizoři mohou sledovat postup anotace, jednotlivé odevzdání a srovnávat odevzdané verze. Mohou se jednoduše vrátit k předchozím verzím odevzdání.

Pro zjednodušení práce supervizora, který také může editovat obrázky pouze pokud jsou mu přiřazené, slouží rozlišení navigačního panelu barvou. Pokud má supervizor obrázek přiřazený, tak je zobrazen standardním způsobem, viz (3.14).



Obrázek 3.14.: Navigační panel - supervizor

V opačném případě je tento panel podbarvený červeně (viz 3.15). Po kliknutí na tlačítko "uložit anotaci" se mu zobrazí upozornění o tom, že nemůže ukládat postup.



Obrázek 3.15.: Navigační panel - supervizor

Tento jednoduchý přístup zajistí bezpečný postup při anotaci, díky kterému se vyloučí možnost, aby více uživatelů mohlo najednou editovat stejný obrázek.

Editační nástroje

V následující části jsou popsány editační nástroje nezbytné pro provádění anotací. Každou z těchto akcí je možné spustit buď kliknutím na tlačítko v hlavním panelu nebo použitím klávesové zkratky.

- **Kreslení**

Tento nástroj je navržen pro vytváření nových polygonů nebo čar. Je to základní nástroj, který slouží k označování tříd. Rozhodnutí, zda vytvářet čáru nebo polygon, závisí na aktuálně vybrané klasifikační vrstvě. Pro tento nástroj byla přidělena klávesová zkratka F a také se aktivuje jako hlavní nástroj po kliknutí levého tlačítka myši na editor. Po prvním levém kliknutí myší se do obrázku přidá nový bod, za kterým se čára "natahuje" při pohybu

myši. Dalším kliknutím levého tlačítka myši se vytváří nový bod v tvořeném polygonu. Tato sekvence pokračuje až do ukončení kliknutím pravým tlačítkem myši. Funkcionalita tohoto nástroje je založena na rozšíření objektu **Path**.

- **Vyříznutí polygonu**

Nástroj pro vyříznutí díry do polygonu je nezbytný z několika důvodů. Prvním a klíčovým faktorem je zlepšení procesu označování a anotace dat. V situacích, kdy je třeba anotovat objekty s komplexními strukturami, jako jsou mostní konstrukce, může polygonální obrys překrývat různé části objektu. Bez možnosti vytvářet díry v polygonu by anotátoři museli polygon rozdělit na několik menších polygonů, což by mohlo značně zkomplikovat proces a vedlo by k nepřehlednému anotačnímu datu.

Druhým aspektem je zachování přesnosti anotací. Vytváření díry do polygonu umožňuje přesně vymezit oblast, která nepatří do objektu a tím dosáhnout vysoké kvality a přesnosti anotace.

- **Rozdelení polygonu**

Rozdelení polygonu je další nezbytný nástroj, zejména v případě anotování rozsáhlejších struktur a při opravách. Příkladem pro použití toho nástroje může být situace, kdy anotátor označuje stromy a mylně označí jedním polygonem 2 stromy vedle sebe. Díky tomuto nástroji je jednoduše možné tyto objekty oddělit, bez nutnosti smazat a znova označit dvě nové oblasti. Také se může hodit v případě označení více různých tříd jedním polygonem.

- **Přesunutí**

Nástroj přesunutí je nezbytný pro projekty, ve kterých se označují instance. Umožňuje měnit pořadí polygonů, neboli měnit jejich z-index vůči ostatním. Z důvodu přehlednosti, zejména při velkém počtu polygonů, je nástroj navržen tak, aby umožňoval přesun polygonů pouze mezi těmi, které překrývá nebo se s nimi protíná.

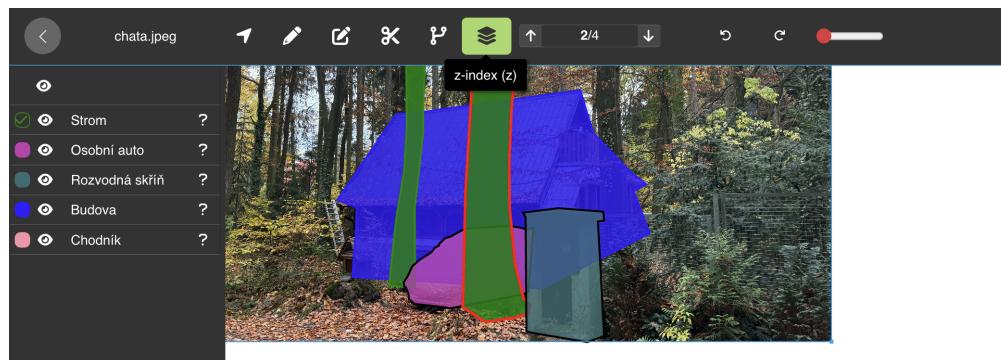
Po vybrání tohoto nástroje a kliknutím levým tlačítkem myši na polygon se polygon zvýrazní červeným okrajem. V hlavním panelu editoru se zobrazí jeho pořadí relativně k polygonům, které překrývá nebo s nimiž se protíná. Na obrázku 3.16 na následující straně je vidět, že vybraný polygon je v pořadí druhý ze čtyř. Pro posunutí polygonu nahoru lze použít klávesovou zkratku *Q* a pro posun dolů *A*.

- **Změna vrstvy polygonu**

Nástroj pro změnu vrstvy polygonu ve 2D editoru je užitečný pro opravu možných chyb v anotaci. Umožňuje změnit vrstvu polygonu bez potřeby jeho mazání a znovuvytváření. S tímto nástrojem lze snadno přesunout polygon z jedné vrstvy, například *Listnatý strom*, do jiné vrstvy, jako je *Keř*.

- **Odevzdání**

Anotátoři jsou povinni svůj pracovní postup pravidelně ukládat na server. Během anotačního procesu jsou data nejprve ukládána do lokálního úložiště prohlížeče. Z bezpečnostních



Obrázek 3.16.: Nástroj přesunutí

důvodů je však po určité odpracované době nezbytné tyto údaje přenést na server. Tento krok se provádí kliknutím na tlačítko pro odevzdání, což vyvolá zobrazení modálního okna (viz Obrázek 3.17). V tomto okně je automaticky předvyplněn doba práce zaznamenaná aplikací, kterou může anotátor upravit podle potřeby. Na server se následně ukládá jak původní, tak modifikovaný časový údaj práce. Kromě toho má anotátor možnost připojit poznámku k ukládaným datům, poskytující další informace o aktuálním stavu práce, jako například specifika anotací střech.



Obrázek 3.17.: Odevzdání na server

3.7. Datový model

Tato sekce popisuje hlavní části datového modelu. Vzhledem k praktičnosti byly všechny databázové entity pojmenovány anglicky.

Správa uživatelů

Pro správu uživatelů bylo navrženo 5 základních tabulek, viz 3.18.



Obrázek 3.18.: Správa uživatelů

Hlavní tabulky a jejich účel jsou následovné:

- **user (uživatel):**

Tabulka 'user' obsahuje základní informace o uživatelích, včetně unikátního identifikátoru typu int, jména, příjmení, emailu a telefonního čísla. Pro další kontaktní údaje, například uživatelské jméno na platformě RocketChat³, se využívá formát JSON kvůli flexibilitě.

- **auth_method (autorizační metoda):**

Číselník 'auth_method' slouží pro různé metody autorizace, jako jsou účty na Microsoft Azure⁴ nebo Google. Sloupec "config" je určen pro konfiguraci konkrétní autorizační metody.

- **user_auth_method (uživatel_autorizační metoda):**

Tabulka 'user_auth_method' propojuje uživatele a autorizační metody. Vazba typu m:n umožňuje více účtů pro jednoho uživatele. Sloupec "account_id" obsahuje identifikátor účtu a "flag_allowed" určuje, zda je daná metoda pro uživatele povolena či zakázána.

- **role:**

V tabulce 'role' je definován seznam rolí v systému. Primárním klíčem je alias role, který usnadňuje výběr rolí.

³RocketChat je open-source webová chatovací platforma, která nabízí funkce jako jsou skupinový chat, přímé zprávy, a další komunikační nástroje. Více informací lze nalézt v oficiální dokumentaci: <https://docs.rocket.chat/>

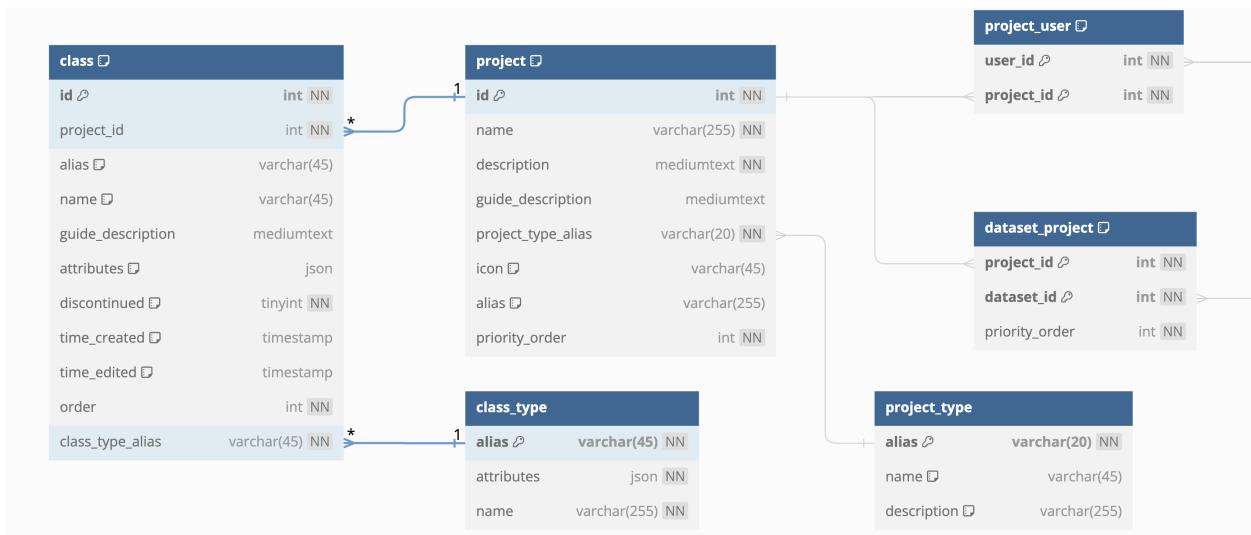
⁴Microsoft Azure je cloudová platforma od společnosti Microsoft, která nabízí širokou škálu cloudových služeb. Více informací naleznete v oficiální dokumentaci: <https://docs.microsoft.com/en-us/azure/>

- **user_role (uživatel_role):**

Tabulka 'user_role' umožňuje přiřazení více rolí jednomu uživateli.

Projekt a klasifikační třídy

Uchování informací o projektech a klasifikačních třídách je realizováno prostřednictvím databázových tabulek, jak je ilustrováno na 3.19:



Obrázek 3.19.: Projekt a klasifikační třídy

- **project (projekt):**

Tabulka 'project' používá jako primární klíč číselné ID typu INT, které je automaticky inkrementováno a tudíž je unikátní. Ukládá název projektu a jeho alias pro export. Pokud alias není nastaven, používá se pro export název projektu. Sloupec 'description' obsahuje popis pro administrátora, zatímco 'guide_description' je určen pro anotátory a zobrazuje se v editoru. Priorita projektu je řešena sloupcem 'priority_order', kde vyšší hodnota značí větší prioritu.

- **project_type (typ projektu):**

Tabulka 'project_type' obsahuje číselník typů projektů.

- **class (třída):**

Klasifikační třídy jsou ukládány v tabulce 'class'. Význam jednotlivých atributů byl popsán dříve.

- **class_type (typ třídy):**

'class_type' je číselník typů tříd. Sloupec 'attributes' definuje specifické hodnoty pro nastavení formuláře, například požadavek na barvu výplně v závislosti na typu třídy.

- **project_user (projekt_uživatel):**

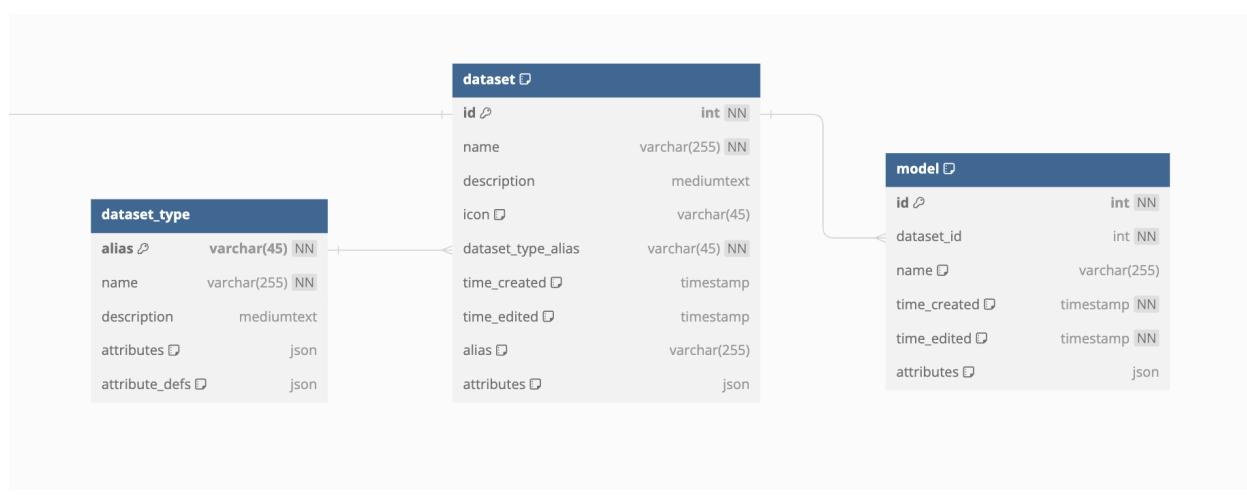
Práva na projekty pro supervizory a anotátory jsou vyřešena tabulkou 'project_user'. Uživatelé mají přístup pouze k projektům ke kterým existuje vazba v této tabulce.

- **dataset_project (dataset_projekt):**

Prioritizace datasetů v rámci projektu se řídí pomocí 'dataset_project'. Při přidání obrázku z dosud nepřiřazeného datasetu do projektu se vytvoří záznam s nejnižší prioritou.

Datasetsy

Uložení datasetů je řešeno následujícím způsobem, vizuální přehled databázových entit naleznete na 3.20:



Obrázek 3.20.: Datasety

- **dataset:**

Datasety jsou ukládány do tabulky 'dataset'. Jako primární klíč je použito generované ID. Pro každý dataset se ukládají informace o jeho názvu a popisu. Alias datasetu funguje podobně jako u projektů. Atributy nastavitelné administrátorem jsou uloženy ve sloupci 'attributes'.

- **dataset_type (typ datasetu):**

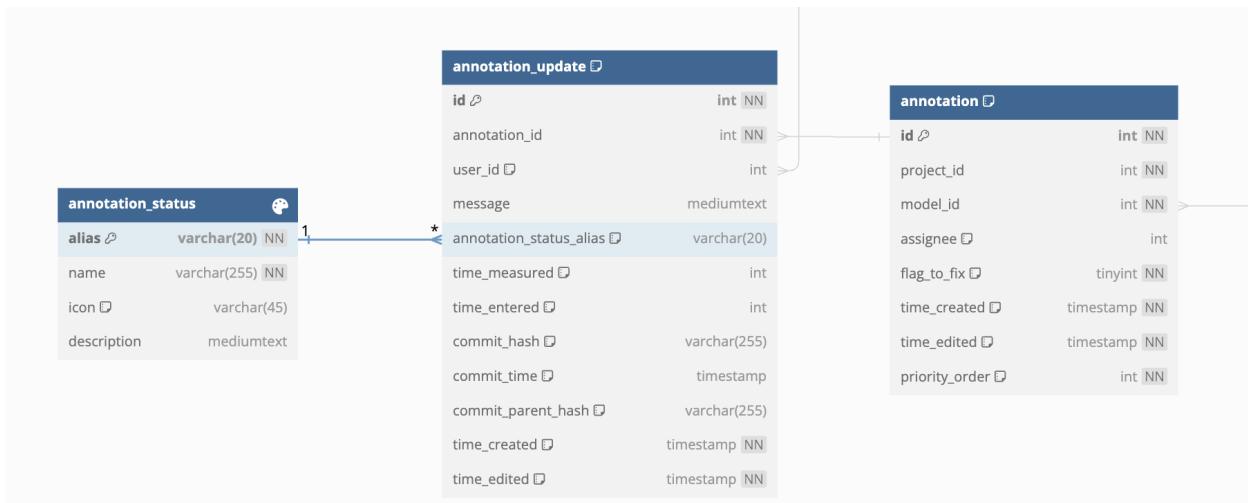
Tabulka 'dataset_type' obsahuje číselník typů datasetů. Na rozdíl od ostatních číselníků v aplikaci zahrnuje nastavení formuláře pro atributy, které jsou uvedeny ve sloupci 'attribute_defs' (definice atributů).

- **model:**

Obrázky nahrávané do aplikace jsou ukládány v tabulce 'model'. Tento název byl zvolen pro obecnost, na rozdíl od možného názvu 'picture'. Umožňuje to přidání nových datových typů bez nutnosti měnit databázový model. Konkrétní nastavení atributů, jako jsou například zeměpisné souřadnice, je uloženo ve sloupci 'attributes'.

Anotace

Uložení postupu anotace a jejích nastavení je realizováno pomocí několika tabulek, jak je znázorňeno na 3.21. Struktura tabulek je následující:



Obrázek 3.21.: Anotace

- **annotation (anotace)**: Po přiřazení nového obrázku do projektu se vytvoří záznam v tabulce 'annotation'. Tabulka je spojena s projektem, modelem a uživatelem. Propojení s uživatelem je skrze sloupec 'assignee', který označuje přiřazeného uživatele, který může ukládat postup. Příznak 'flag_to_fix' indikuje potřebu opravy anotace. Priorita anotace je určena číselnou hodnotou v sloupci priority.
- **annotation_update (aktualizace anotace)**: Tabulka 'annotation_update' zachycuje postup anotace, včetně přiřazení do projektu a postupného ukládání. Odkaz na commit⁵ v lokálním gitlab repozitáři je uložen ve sloupci 'commit_hash' spolu s časem commitu a informací o předchozím commitu.
- **annotation_status (status anotace)**: Tabulka 'annotation_status' uchovává stavy anotací pro workflow management. Stavy zahrnují přiřazení anotace, její uložení, odevzdání a další, přičemž každý stav má svou ikonu pro lepší přehlednost.

⁵V Gitu, 'commit' označuje akci uložení sady změn v souborech do lokálního repozitáře. Každý commit má jedinečný identifikátor (hash) a obsahuje informace o změnách, autora změn, a časové razítce. Commit slouží jako zaznamenaný 'snímek' stavu projektu v určitém čase.

4. Závěr

Cílem této bakalářské práce bylo navrhnout a implementovat webovou aplikaci pro anotování obrázků leteckých a uličních snímků. Aplikace je nyní plně funkční a aktivně využívaná firmou Melown Technologies SE. Díky této aplikaci bylo označeno více než 1000 snímků, které potvrzují její významný přínos pro anotační procesy ve firmě.

Je třeba zdůraznit, že úspěšný vývoj a implementace aplikace byl významně ovlivněn agilním přístupem k vývoji softwaru a schopností flexibilně reagovat na měnící se potřeby a požadavky. Výsledná aplikace nejenže zvyšuje efektivitu anotačního procesu, ale také přispívá k lepšímu pochopení a správě obrazových dat, což je klíčové pro úspěšné využití v oblasti strojového učení.

Byl navržen workflow management, který je úzce propojen s uživatelskými rolemi v aplikaci. Tento přístup přispěl k výraznému zjednodušení procesů a umožnil lepší rozdělení odpovědnosti a víceúrovňovou kontrolu anotací. Díky aplikaci odpadla potřeba pracovníků se znalostí GIS systémů. Vyvinutý 2D editor je jednodušší na použití, a tedy umožňuje jeho využívání i uživateli bez znalostí GIS.

Časově náročnou částí práce byla analýza požadavků a návrh nastavení správného workflow managementu. Tuto práci jsem vytvářel jako jediný vývojář, co znamenalo, že jsem prováděl veškerou analýzu procesů a datového návrhu včetně implementace. Vycházel jsem ze svých předchozích zkušeností a následně návrh konzultoval v rámci firmy s mým nadřízeným, který mi poskytl cennou zpětnou vazbu. Tento proces nejenže podpořil hlubší porozumění firemních potřeb a procesů, ale také mi umožnil zlepšit komunikační dovednosti a schopnost prezentovat technické koncepty.

Jako hlavní jazyk pro tuto aplikaci byl zvolen Javascript z důvodu jeho jednotnosti s ostatními aplikacemi ve firemním prostředí. Tato volba byla podpořena i flexibilitou a rozšířeností Javascriptu, která zjednodušuje integraci a budoucí rozvoj aplikace. Možnost použití jazyka PHP jsem proto v práci nezmiňoval, neboť neodpovídala technologickému přístupu firmy a zaměření na vytvoření jednotné, integrované a dobré spravovatelné aplikace.

V současné době probíhá rozšiřování aplikace o modul pro anotování 3D dat ze skenerů LiDAR, což představuje další krok k evoluci a rozšiřování funkčnosti systému. Během vývoje došlo k řadě úprav a vylepšení, zejména v oblasti 2D editoru, což významně přispělo ke zlepšení pracovních postupů a zvýšení uživatelské přívětivosti.

Pro další vývoj aplikace se nabízí možnost automatizace procesů, především v oblasti přiřazování obrázků anotátorům. Tento krok by mohl dále zvýšit efektivitu a snížit manuální úsilí a náklady

firmy potřebné ke správě a distribuci dat. Dále by bylo vhodné se zaměřit na rozšíření analytických schopností aplikace. To může zahrnovat pokročilé funkce pro sledování a hodnocení kvality anotací. Rovněž by bylo užitečné provést další výzkum v oblasti uživatelského rozhraní a zkušeností, aby byla zajištěna intuitivita a výkonnost aplikace při rostoucí komplexitě a rozsahu funkcí.

Seznam použitých zkrátek

| | |
|----------------------|--|
| 2D | dvoudimenzionální prostor |
| 3D | trojdimenzionální prostor |
| ajax | Asynchronní JavaScript a XML |
| API | Rozhraní pro programování aplikací |
| CNN | Konvoluční neuronová síť |
| CSV | Hodnoty oddělené čárkami |
| DAC | Diskreční přístupová kontrola |
| DOM | Dokumentový objektový model |
| fetch | Fetch API |
| GPS | Globální polohový systém |
| HTML5 | Jazyk pro značkování hypertextu, verze 5 |
| HTTPS | Hypertext transfer protocol secure |
| Lidar | Detekce a určování vzdálenosti pomocí světla |
| MAC | Mandatorní řízení přístupu |
| MIT (licence) | Licence Massachusetského technologického institutu |
| MVP | Minimálně životašchopný produkt |
| MySQL | Jazyk Structured Query Language |
| npmJS | Správce balíčků Node.js pro JavaScript |
| RBAC | Řízení přístupu založené na rolích |
| SEO | Optimalizace pro vyhledávače |
| SPA | Aplikace na jedné stránce |
| UML | unifikovaný modelovací jazyk |
| URL | Jednotný lokátor zdrojů |
| UX | Uživatelská zkušenost |

Seznam obrázků

| | |
|--|----|
| 2.1. scrum | 18 |
| 3.1. Diagram komponent | 27 |
| 3.2. Analýza procesu anotace obrázků | 28 |
| 3.3. Zobrazení modalu s polohou | 29 |
| 3.4. Use case diagram role účetní | 31 |
| 3.5. Tabulka s přehledem odpracovaných hodin | 31 |
| 3.6. Formulář pro vytvoření datasetu | 34 |
| 3.7. Editace atributů datasetu | 35 |
| 3.8. Nahrávání obrázků | 36 |
| 3.9. Náhled obrázku | 37 |
| 3.10. Obecná segmentace | 38 |
| 3.11. Obrázková segmentace instancí | 39 |
| 3.12. Nastavení priority datasetů v projektu | 41 |
| 3.13. Navigační panel - anotátor | 42 |
| 3.14. Navigační panel - supervizor | 43 |
| 3.15. Navigační panel - supervizor | 43 |
| 3.16. Nástroj přesunutí | 45 |
| 3.17. Odevzdání na server | 45 |
| 3.18. Správa uživatelů | 46 |
| 3.19. Projekt a klasifikační třídy | 47 |
| 3.20. Datasety | 48 |
| 3.21. Anotace | 49 |

Seznam použitých zdrojů

1. BISHOP, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006. ISBN 978-0-387-31073-2.
2. BURKOV, A. *The Hundred-Page Machine Learning Book, Quebec City, Canada, 2019*, ISBN 978-1999579517. Taylor & Francis, 2019. ISBN 978-1999579517.
3. EL NAQA, Issam; MURPHY, Martin J. *Machine Learning in Radiation Oncology*. What is Machine Learning? Springer, 2015. ISBN 978-3-319-18304-6. Dostupné z DOI: [10.1007/978-3-319-18305-3_1](https://doi.org/10.1007/978-3-319-18305-3_1).
4. LIU, Z.; LUO, P.; QIU, S.; WANG, X.; TANG, X. Deep learning for generic object detection: A survey. *International Journal of Computer Vision*. 2020, **128**(2), 261–318.
5. KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. 2012, s. 1097–1105.
6. LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*. 2015, **521**(7553), 436–444.
7. ZHANG, L.; LIN, L.; LIANG, X.; HE, K.; SUN, J. Is faster R-CNN doing well for pedestrian detection? In: *European conference on computer vision*. 2016, s. 443–457.
8. AALST, W. van der; HEE, K.M. van. *Workflow Management: Models, Methods, and Systems*. MIT Press, 2004. Cooperative information systems. ISBN 978-0-262-72046-5.
9. MUEHLEN, M.Z. *Workflow-based Process Controlling: Foundation, Design, and Application of Workflow-driven Process Information Systems*. Logos, 2004. Advances in information systems and management science. ISBN 978-3-8325-0388-8.
10. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *Ergonomics of human-system interaction*. International Organization for Standardization, 2010. Dostupné také z: <https://www.iso.org/standard/77520.html>. ISO 9241-210.
11. EXPERIENCE, World Leaders in Research-Based User. *The Definition of User Experience (UX)* [Nielsen Norman Group] [online]. [visited on 2023-11-21]. Available from: <https://www.nngroup.com/articles/definition-user-experience/>.
12. FERRAIOLI, David; KUHN, Richard. Role-Based Access Controls. In: *Proceedings of the 15th National Computer Security Conference*. National Institute of Standards a Technology, 1992, s. 554–563.

13. *Manifest Agilního vývoje software* [online]. [cit. 2023-11-12]. Dostupné z: <https://agilemanifesto.org/iso/cs/manifesto.html>.
14. MIJACOBS. *Plánování efektivních úloh pomocí DevOps - Azure DevOps* [online]. 2023-08-29. [cit. 2023-10-29]. Dostupné z: <https://learn.microsoft.com/cs-cz/devops/plan/planning-efficient-workloads-with-devops>.
15. *Node.js* [online]. [cit. 2023-03-16]. Dostupné z: <https://nodejs.org/en>.
16. *Vue.js* [online]. [cit. 2023-05-06]. Dostupné z: <https://vuejs.org/guide/introduction.html>.
17. *SPA (Single-page application) - MDN Web Docs Glossary: Definitions of Web-related terms* / MDN [online]. 2023-08-28. [visited on 2023-10-29]. Available from: <https://developer.mozilla.org/en-US/docs/Glossary/SPA>.
18. *Paper.js – About* [online]. [cit. 2023-10-29]. Dostupné z: <http://paperjs.org/about/>.
19. CHACON, Scott; STRAUB, Ben. *Pro Git*. 2. vyd. Apress, 2014. ISBN 978-1484200773.
20. MIJACOBS. *Co je Git? - Azure DevOps* [online]. 2023. [cit. 2023-10-29]. Dostupné z: <https://learn.microsoft.com/cs-cz/devops/develop/git/what-is-git>.
21. MIJACOBS. *Co je Git? - Azure DevOps* [online]. [cit. 2023-10-29]. Dostupné z: <https://learn.microsoft.com/cs-cz/devops/develop/git/what-is-git>.
22. *MySQL :: Developer Zone* [online]. [cit. 2023-11-22]. Dostupné z: <https://dev.mysql.com/>.
23. BUTLER, H.; DALY, M.; DOYLE, A.; GILLIES, Sean; SCHaub, T.; HAGEN, Stefan. *The GeoJSON Format* [online]. 2016-08. [cit. 2023-11-22]. Request for Comments, RFC 7946. Internet Engineering Task Force. Dostupné z DOI: [10.17487/RFC7946](https://doi.org/10.17487/RFC7946).

A. Externí přílohy

Externí přílohy této bakalářské práce jsou umístěny na adrese:
https://github.com/jpalenik/thesis_ki_ujep.

Na úložišti GitHub jsou uloženy tyto externí přílohy:

- **zdrojové kódy bakalářské práce**
- **vybrané zdrojové kódy aplikace**
- **zdrojové kódy UML diagramů**

Základní struktura úložiště:

| | |
|-----------------------|---------------------------------------|
| ki-thesis.pdf | text práce v PDF |
| thesis_ki_ujep | zdrojový kód práce v $\text{\LaTeX}u$ |
| diagrams | zdrojové kódy UML diagramů |
| source_codes | vybrané zdrojové kódy aplikace |