UNIVERZITA KOMENSKÉHO V BRATISLAVE

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Webová mapa chránených oblastí Slovenskej republiky

Bakalárska práca

2024

Roman Božik

UNIVERZITA KOMENSKÉHO V BRATISLAVE

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Webová mapa chránených oblastí Slovenskej republiky

Bakalárska práca

Študijný program: Aplikovaná informatika

Študijný odbor: Informatika

Pracovisko (katedra/ústav): Katedra didaktiky matematiky, fyziky a informatiky

Školiteľ záverečnej práce: Ing. František Gyárfáš, CSc.

Konzultant:

Bratislava, 2024

Roman Božik

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Roman Božik
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Webová mapa chránených oblastí Slovenskej republiky
Web map of protected areas of the Slovak Republic

Anotácia: Cieľom práce bude vytvoriť interaktívnu mapu chránených oblastí SR ako webovú aplikáciu. Mapa umožní zobrazovať rôzne vrstvy chránených oblastí, území európskeho významu, či vrstvy stupňov ochrany. Aplikácia poskytne niekoľko úrovní používateľov od admina, zadávateľov vrstiev ochrany, registrovaných používateľov s prístupom k dátam, až po návštevníkov. Umožní pridávať geograficky lokalizované informácie o daných oblastiach, možnosti výletov, fotografie, diskusie ako aj postihy v prípade porušenia pravidiel v danej oblasti. Webová aplikácia bude obsahovať API, z ktorého bude možné dáta stiahnuť. Bude vyvíjaná ako webová aplikácia s použitím nástrojov: Python (Django), HTML5, CSS, JavaScript (jQuery), PHP, PostgreSQL.

Vedúci: Ing. František Gyarfaš, CSc.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: doc. RNDr. Tatiana Jajcayová, PhD.

Dátum zadania: 30.10.2022

Dátum schválenia: 04.11.2022 doc. RNDr. Damas Gruska, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Čestné vyhlásenie

Čestne prehlasujem, že som túto bakalársku prácu vypracoval samostatne s použitím citovaných a na internete dostupných zdrojov.

V Bratislave dňa 01.06.2024 .....

Roman Božik

Poďakovanie

Ďakujem môjmu školiteľovi Ing. Františkovi Gyarfašovi, CSc. za čas strávený vedením bakalárskej práce, trpezlivosť a hlavne ochotu nie len viesť ale aj poskytovať konštruktívnu spätnú väzbu. Veľká vďaka patrí aj daňovým poplatníkom, ktorí aj v čase rekordného zadlženia, aktuálneho rizika durácie a hroziaceho kreditného rizika boli ochotný znášať ťarchu môjho štúdia a umožnili mi dotiahnuť moju bakalársku prácu do úspešného konca bez nutnosti riešiť finančnú záťaž. Taktiež by som chcel poďakovať mojej rodine za neustálu podporu a čas strávený nad testovaním použiteľnosti praktickej časti bakalárskej práce.

Abstrakt

Cieľom bakalárskej práce je vyprodukovať webovú aplikáciu, ktorá umožní užívateľom prehliadať interaktívnu mapu s chránenými oblasťami Slovenskej republiky. V rámci aplikácie je možné získavať informácie o chránených oblastiach, diskutovať o nich, ako aj tvoriť svoje vlastné oblasti a zdieľať ich s inými užívateľmi webovej aplikácie. Aplikácia obsahuje základné funkcie potrebné pre sociálnu interakciu na platforme, ako je napríklad systém notifikácií, manažovanie priateľov či užívateľské profily s možnosťou osobitného kontaktu. Aplikácia je naprogramovaná v jazyku Python a JavaScript a najviac použité technológie sú Django, Django Unicorn, Leaflet, Boostrap. GIS dáta o chránených oblastiach boli poskytnuté na účely použitia v bakalárskej práci Štátnou ochranou prírody Slovenskej republiky.

Kľúčové slová: python, django, mapa, chránené oblasti, gis

Abstract

The aim of the bachelor thesis is to produce a web application that will allow users to browse an interactive map with protected areas of the Slovak Republic. Within the application it is possible to get information about protected areas, discuss them, as well as create your own areas and share them with other users of the web application. The application includes basic functions necessary for social interaction on the platform, such as a notification system, friend management or user profiles with the possibility of a special contact. The application is programmed in Python and JavaScript and the most used frameworks are Django, Django Unicorn, Leaflet, Boostrap.

GIS data on protected areas were provided for use in the bachelor thesis by the State Nature Conservancy of the Slovak Republic.

Keywords: python, django, map, protected areas, gis

Obsah

Zoznam skratiek

OPM	OpenStreetMap
OTM	OpenToolMap
API	Application Programming Interface

Úvod

V posledných dekádach sa čím ďalej tým viac dbá na ochranu prírody, biodiverzity a krajiny. Nároky na štát ako aj jeho kompetencie sa neustále navyšujú a tomuto trendu sa prispôsobujú aj ústredné orgány štátnej správy a legislatíva. Jedným z negatívnych efektov navyšovania kompetencií štátnych orgánov a zmohutňovania legislatívnych predpisov sú nepriehľadné a intuícii odporujúce pravidlá a to môže viesť k rezignácií dlhodobo sledovať aktuálne predpisy z prvej ruky. Zároveň navýšením globálnej efektivity práce a tým sprevádzanému navýšeniu kúpnej sily obyvateľstva v posledných dvoch dekádach

dochádza k pribúdaniu vstupu obyvateľstva do chránených oblastí na motorových vozidlách za účelom rekreačnej jazdy či kempovania v karavanoch.

Motivácia k tvorbe bakalárskej práce prirodzene vznikla z vyššie uvedenej problematiky a ponúka užívateľovi jednoduchú, intuitívne pochopiteľnú možnosť overenia či oblasť do ktorej má práve namierené spadá do chráneného územia a či je možné tam danú aktivitu legálne vykonať. Dáta o chránených oblastiach sú čerstvo získané zo Štátnej ochrany prírody Slovenskej republiky a preto ponúka veľmi kvalitné dáta. Surové dáta sú voľne prístupné verejnosti na požiadanie orgánu Štátnej ochrany prírody Slovenskej republiky.
Aplikácia taktiež rieši problém centralizovaných internetových skupín, ktoré zgrupujú veľké množstvo členov s rovnakou záľubou, v rámci ktorých členovia s túžbou po lokálnom stretnutí majú problém vyhľadať reálnych užívateľov z blízkeho okolia a to tak, že každá oblasť môže mať priradené diskusné vlákno v rámci ktorého sa užívatelia môžu bližšie dohodnúť na konkrétnej sociálnej interakcií.

Bakalárska práca je rozdelená na štyri kapitoly. Prvá kapitola obsahuje špecifikáciu aplikácie ,analýzu súčasného stavu, teda porovnanie už existujúcich konkurentov aplikácie a detailne popisuje použité technologické prostriedky, druhá návrhu aplikácie a tretia sa zaoberá konkrétnou implementáciou webovej aplikácie. Štvrtá kapitola sa bude venovať popisu interakcie reálnych užívateľov s aplikáciou.

1 Špecifikácia aplikácie a analýza súčasného stavu

Táto kapitola sa delí na dve časti: Prvá sa zameriava na vymedzenie funkcionálnych cieľov našej aplikácie, zatiaľ čo druhá poskytuje analytický pohľad na existujúce riešenia v tejto doméne.

1.1 Špecifikácia aplikácie

V rámci špecifikácie aplikácie je hlavným cieľom vytvorenie komplexnej a intuitívnej platformy, ktorá poskytne užívateľom možnosť interaktívneho prehliadania chránených oblastí Slovenskej Republiky prostredníctvom detailnej mapy. Aplikácia bude zahrňovať podrobné informácie o chránených oblastiach, ako sú názvy, stupne ochrany a relevantné legislatívne údaje. Umožní tiež sociálnu interakciu medzi užívateľmi, vrátane možnosti

komentovať a interagovať s mapou a jej obsahom napríklad za pomoci zdieľania užívateľom vytvorených oblastí s inými užívateľmi. Aplikácia by mala umožniť takto vytvorené oblasti aj v budúcnosti upraviť.

Aplikácia by mala mať implementované aj funkcie vyhľadávania oblastí pomocou GPS koordinátov alebo adresy, ako aj filtráciu chránených oblastí podľa stupňu ochrany a významu. Aplikácia by tiež mala podporovať vytváranie a správu užívateľských účtov, poskytujúc každému užívateľovi možnosť pridávať a spravovať vlastné informácie, ako aj funkciu na odosielanie správ medzi užívateľmi.

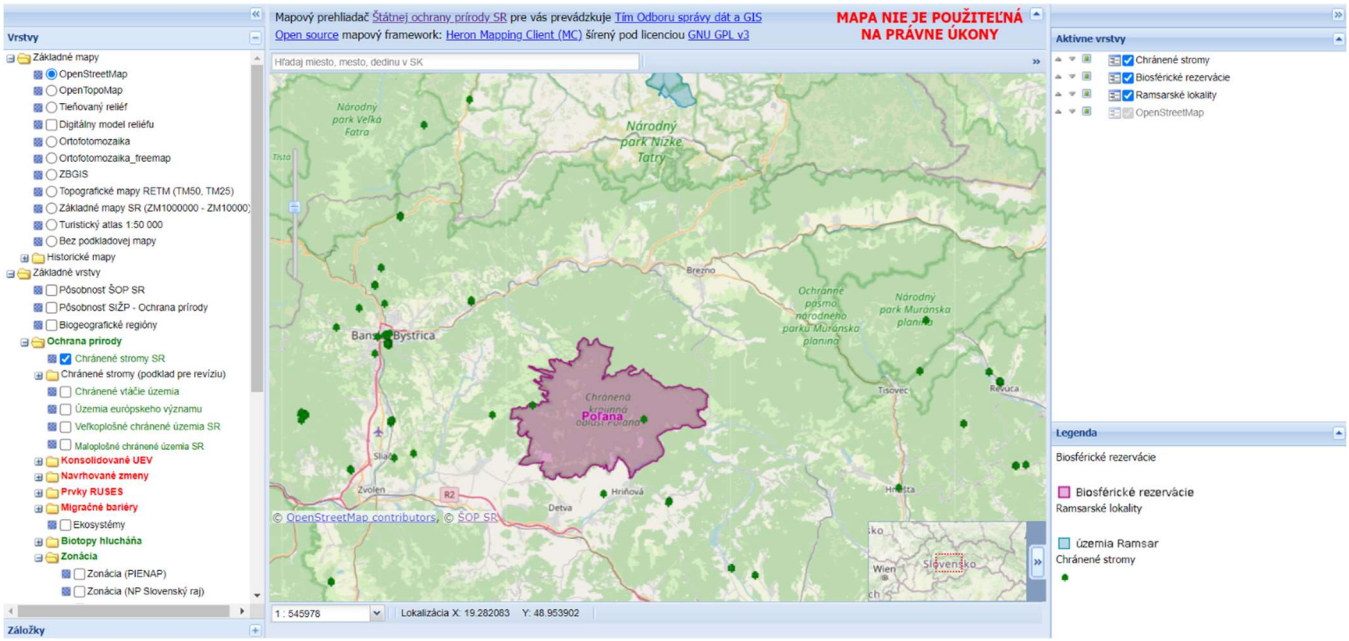
1.2 Analýza súčasného stavu

V analýze dnešným trhom ponúkaných aplikácií sa budem orientovať na ich negatívne stránky s cieľom vylepšenia mnou vyvíjanej aplikácie. Hlavnými kritériami kvality bude intuitívnosť, informačná aktuálnosť a dôraz na podporu a uľahčenie sociálnej interakcie medzi užívateľmi. Výsledky z prieskumu trhovej konkurencie slúžia aj ako usmernenie pri skúmaní vhodných technológií pre tvorbu aplikácie a z toho vyplývajúcej možnosti výberu solídneho technologického základu.

1.2.1 Mapový prehliadač Štátnej ochrany prírody SR

Jedná sa o webovú aplikáciu priamo od Štátnej ochrany prírody Slovenskej republiky dostupnú priamo na stránke https://maps.sopsr.sk/

Webová aplikácia umožňuje užívateľovi vyhľadať a zobraziť veľké množstvo rôznych typov vrstiev ako napríklad chránené stromy, chránené vtáčie územia, migračné bariéry, biotopy hlucháňa ako aj vrstvy od iných inštitúcií, napríklad od Občianskeho združenia PRALES, Národného lesníckeho centra, či Úradu geodézie kartografie a katastra SR. Aplikácia umožňuje aj širokú paletu výberu podkladových máp ako napríklad OSM, OTM, Orto-foto mozaika, ZBGIS, RETM a dokonca aj historické vojenské mapy z rôznych období.



Obrázok 1: Ukážka mapového prehliadača Štátnej ochrany prírody SR

Silné stránky:

- Vývojový tím má informácie o oblastiach, ale aj legislatívnych zmenách z prvej ruky, teda budú vždy aktuálne a hlavne dôveryhodné.
- Aplikácia obsahuje legendu. Obsah v legende vysvetľuje význam vrstiev alebo značiek na mape. Informácie v legende sú z väčšiny prípadov okamžite zrozumiteľné a jednoducho rozlíšiteľné.
- Aplikácia má ihneď viditeľnú záložku s aktívnymi vrstvami, čo v prípade veľkého množstva zvolených typov vrstiev príde vhod.

Slabé stránky:

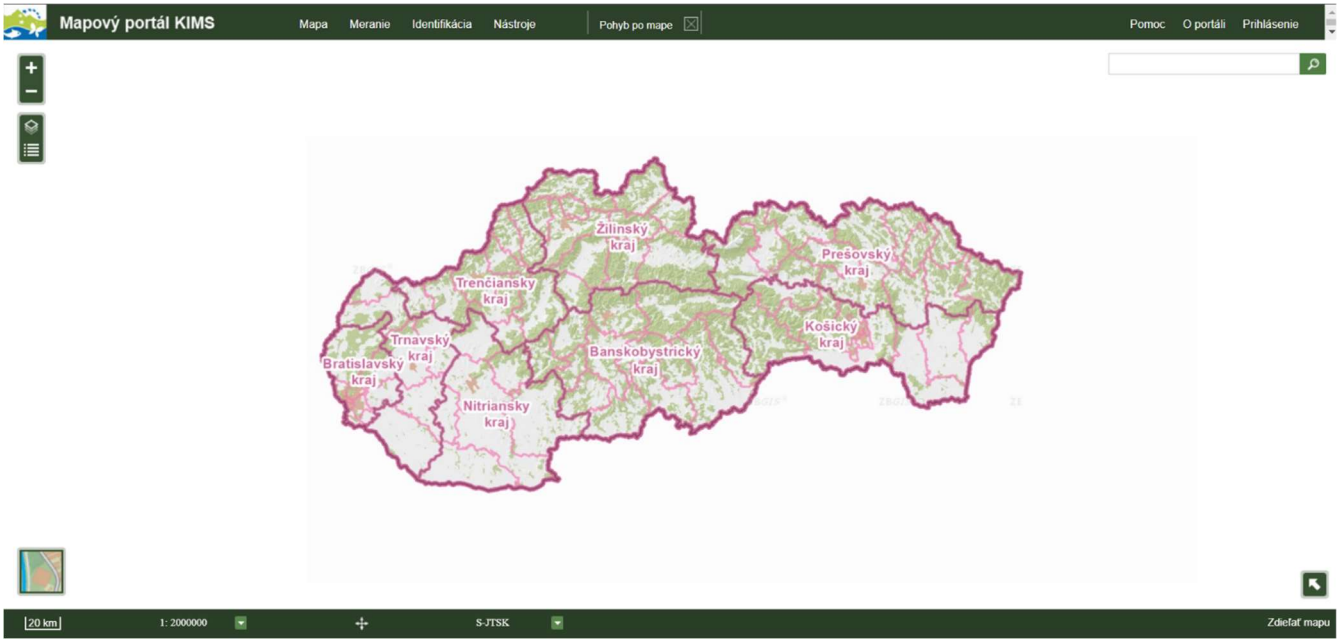
- Nízka miera intuitívnosti nástrojov. Krivka učenia je v tomto prípade dosť dlhá.
- Zlá programátorská kvalita. Rôzne nástroje, ako napríklad meranie plochy, tlačidlá na priblíženie a oddialenie alebo vytvorenie záložky fungujú len sporadicky, alebo vôbec. Pri zvolení a prepínaní medzi niektorými vrstvami a mapovými podkladmi, sa mi podarilo mapu doviesť do nepoužiteľného stavu.
- Miera informovanosti obyvateľstva tiež nie je na požadovanej úrovni. Mapa síce obsahuje spomínané vrstvy, ale zistenie napríklad do akého pásma ochrany spadá daná oblasť je veľmi zložité. Mapa nám ponúka iba externý odkaz k danej oblasti, ktorý ale nie vždy touto informáciou disponuje.

- Nefunkčnosť na niektorých zariadeniach. Mapu vôbec nebolo možné spustiť na dvoch odlišných mobilných telefónoch. Zobrazilo sa iba okno s legendou.
- Nemožnosť tvorby vlastných vrstiev na súkromné použitie
- Aplikácia neumožňuje akúkoľvek sociálnu interakciu a to ani kontaktovanie administrátora.

1.2.2 Mapový portál KIMS

Taktiež webová aplikácia priamo od Štátnej ochrany prírody Slovenskej republiky. *„Mapový portál ako súčasť Verejného portálu KIMS (komplexný informačný a monitorovací systém), slúži na prezentovanie geografických údajov v správe Štátnej Ochrany Prírody (ŠOP) verejnosti.“* (1)
Aplikácia dostupná priamo na stránke https://webgis.biomonitoring.sk/

Na základe možností vo výbere vrstiev je jasná orientácia mapy na chránené druhy a biotopy. V rámci mapy je možné nastaviť až 3 súradnicové systémy a to S-JTSK, WGS84 a ETRS89-LAEA. Mapa umožňuje okrem štandardného priblíženia a oddialenia aj možnosť nastavenia predvolenej mierky z rozmedzia 1: 2 000 000 až po 1: 100. Taktiež je možné manipulovať s priehľadnosťou konkrétnych skupín vrstiev. Podkladové mapy sú tu ZBGIS a RETM.



Obrázok 2: Ukážka mapového portálu KIMS

Silné stránky:

- Jednoduchosť a ľahkosť pracovného prostredia.
- Zrozumiteľná legenda
- Možnosť stiahnuť vizuál mapy vo formáte PDF, JPEG a PNG. Vo formáte PDF sa na druhej strane nachádza aj legenda.
- Vývojový tím priamo z Štátnej ochrany prírody Slovenskej republiky, čo môže mať za následok aktuálnosť vo vrstvách a legislatíve.
- Rýchle vyhľadávanie konkrétnej oblasti a celková plynulosť mapy.

Slabé stránky:

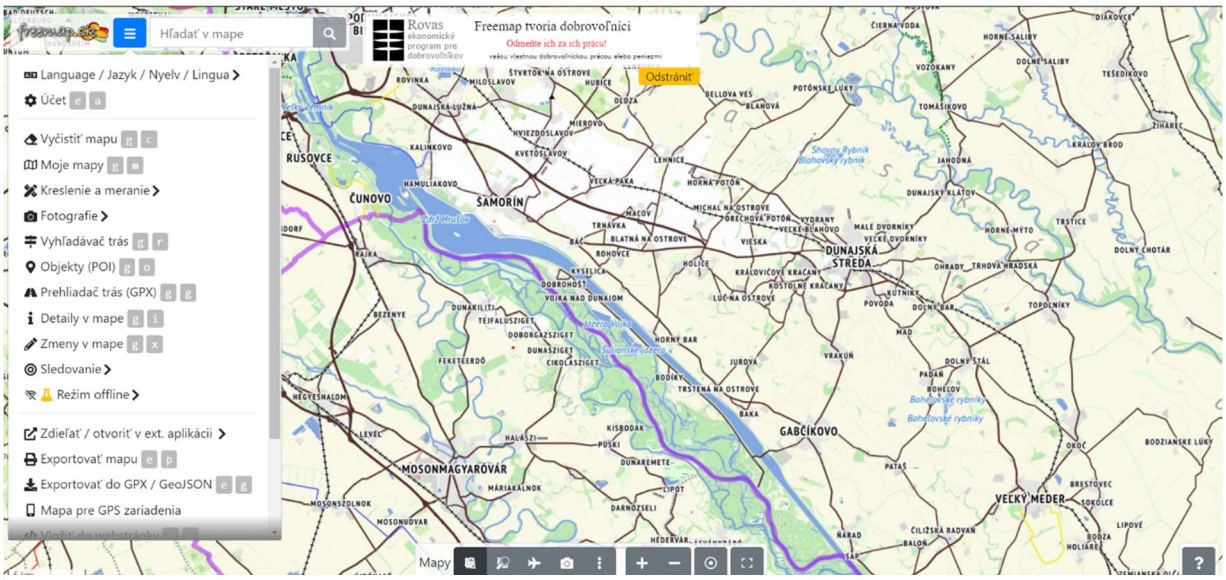
- Aplikácia nemá skoro žiadnu responzivitu. Na mobilných zariadeniach sa síce spustí a mapa sa rozlíšeniu trochu prispôsobí, kontrolné prvky sú ale veľmi ťažko ovládateľné.
- Je tu možnosť pridať do mapy vlastnú externú vrstvu, ale iba za pomoci zadania konkrétneho WMTS servera na ktorom sa dané vrstvy nachádzajú.

- Informácie o konkrétnych oblastiach v čase skúšania nebolo možné získať, aplikácia jednoducho zamrzla a to aj na viacerých zariadeniach.

1.2.3 Freemap.sk

Webová aplikácia freemap.sk nie je spravovaná priamo žiadnym štátnym orgánom, ale občianskym združením Freemap Slovakia. Aplikácia nie je primárne zameraná na chránené oblasti ale na turistické, bežecké a cyklistické chodníky a všeobecné mapovanie terénu. Aplikácia je dostupná na webovom sídle https://www.freemap.sk/

Ako hlavná podkladová mapa je používaná OSM. Aplikácia ale obsahuje aj podkladovú mapu tvorenú z lietadlových snímok, ale aj podkladovú mapu cyklotrás a turistických chodníkov. Aplikácia má široké spektrum výberu vrstiev od airsoftového areálu až po zvonice. Umožňuje aj vyhľadávanie chránených oblastí a chránených stromov. Aplikácia umožňuje tvorbu vlastných objektov priamo na mape a dovoľuje objektu priradiť fotografiu, ktoré následne môžu iný užívatelia komentovať alebo hodnotiť



Obrázok 3: Ukážka Freemap.sk

Silné stránky:

- Aplikácia má radu veľmi užitočných funkcionalít, ako napríklad zobrazenie aktuálnej pozície pomocou GPS, špeciálny režim off-line umožňujúci uložiť mapu do pamäte, prihlasovanie pomocou sociálnych sietí ako napríklad facebook, požiadavka na zmenu obsahu, vyhľadávač trás ,exportovanie vrstiev do formátu GeoJSON a mnoho ďalšieho.

- Legenda je tu veľmi prepracovaná. Obsahuje podrobnú informáciu o každej značke nachádzajúcej sa v mape.
- Vysoká miera informovanosti. Podrobnosti obsahujú meno vo viacerých jazykoch, typ chránenej oblasti, ako aj stupeň ochrany.
- Výborná responzivita. Aplikácia sa výborne ovláda aj na mobilných zariadeniach rôzneho rozlíšenia displeja.

Slabé stránky:

- Aplikácia umožňuje iba veľmi obmedzenú sociálnu interakciu. Objekty síce je možné zdieľať cez externé odkazy na iné sociálne siete, v rámci platformy ale zdieľanie vlastného objektu s inými používateľmi možný nie je. Rovnako platforma umožňuje iba diskusiu pod konkrétnou fotografiou, nie na celom území danej oblasti.
- Niektoré chránené oblasti sú zobrazené iba značkou a nie sú jasné ich hranice.
- Aplikácia neinformuje o zakázaných aktivitách v oblastiach, iba o stupni ochrany.

1.3 Použité technológie

V tejto kapitole sa zameriame na detailný prehľad technologických nástrojov a metodík, ktoré boli vybrané pre vývoj našej full-stack (2) aplikácie. V kontexte rýchleho vývoja IT oblasti (3) a kľúčovej úlohy bezpečnosti v moderných organizáciách je dôležité zvoliť správne technologické riešenia, ktoré zabezpečia efektivitu, bezpečnosť a udržateľnosť projektu. V tejto časti predstavíme základné technológie, na ktorých je naša aplikácia postavená, vrátane výhod a obmedzení, ktoré sú s ich použitím spojené.

V nadväznosti na analýzu súčasného stavu, ktorá bola podrobne rozpracovaná v predchádzajúcej kapitole, tu konkretizujeme funkcionálne a kvalitatívne požiadavky na vývoj aplikácie. Proces výberu technológií nebude izolovaným rozhodnutím; bude priamo ovplyvnený požiadavkami a cieľmi uvedených v špecifikácií aplikácie. Dôkladná selekcia a evaluácia použitých technológií preto predstavujú kľúčový krok k zabezpečeniu úspešnej realizácie a budúcej udržateľnosti aplikácie.

Výber programovacieho jazyka

Aplikáciu som sa rozhodol písať v programovacom jazyku Python (4). Dôvodom uprednostnenia tohto jazyka pred ostatnými je široký výber knižníc vhodných na riešenie problematiky bakalárskej práce a rozsiahla komunitná základňa, keďže Python je jedným z najobľúbenejších programovacích jazykov súčasnosti (5). Aplikácia bude určená pre širokú verejnosť a preto je vhodné, aby k nej bolo možné pristupovať pomocou štandardných internetových prehliadačov. Tie ale k tomuto dátumu nedokážu priamo spracovávať kód napísaný v programovacom jazyku Python a preto je nutné do výberu zahrnúť aj JavaScript, ktorý je natívne podporovaný všetkými globálne používanými prehliadačmi.

Django

Django je komplexný framework s vynikajúcou dokumentáciou vyvinutý pre rýchly vývoj webových aplikácií v jazyku Python. Jeho architektúra podporuje model-view-template (6) štruktúru, ktorá umožňuje oddelenie logiky aplikácie od užívateľského rozhrania. Django automaticky spravuje databázové operácie vďaka integrovanému ORM systému, čo umožňuje pracovať s databázami na určitej úrovni abstrakcie. Ďalšou silnou stránkou je bezpečnosť, kde ponúka vysokú úroven bezpečnosti voči rôznym typom hrozieb. Veľkou výhodou je taktiež aj integrácia s veľkým množstvom kompatibilných knižníc, ktoré rozširujú funkčnosť frameworku v rôznych oblastiach, ako napríklad aj práca s geografickými dátami.

Leaflet

Leaflet je JavaScriptová knižnica s verejne prístupným zdrojovým kódom, ktorá umožňuje vývojárom vytvoriť podkladovú mapu a vďaka API vytvárať interaktívne vrstvy, upravovať ich, zjednocovať do skupín a mnoho ďalšieho. Leaflet primárne používa OPM podkladovú mapu. Leaflet je ako knižnica dobre zdokumentovaná a chod knižnice zabezpečuje veľké množstvo dobrovoľných vývojárov. Keďže Leaflet je komunitný projekt s otvoreným kódom tak sa vývojári rozhodli vytvoriť API aj pre modifikáciu tejto knižnice, čo prináša mnoho ďalších výhod, napríklad možnosť tvorby vyskakovacích okien, alebo iný prídavný modul s funkcionalitou ktorú pôvodný autori ani neplánujú pridať.

Folium

Folium je python knižnica s otvoreným kódom ktorá umožňuje tvorbu interaktívnych máp s použitím JavaScriptovej knižnice Leaflet. Folium umožňuje prácu s Leaflet API v prostredí pythonu a zároveň dodáva vlastné riešenia pre podporu viacerých geopriestorových dátových formátov. Ďalšie modifikácie nad rámec knižnice Leaflet je napríklad balík ikoniek, lodné značky, mini mapa, zobrazenie dňa a noci podľa času v danej krajine a podobne.

Na trhu existujú aj iné knižnice, ktoré vytvárajú mapy, prípadne dokážu priamo spolupracovať s Leaflet. Najpoužívanejšími alternatívami sú mplleaflet a basemap. Knižnica mplleaflet má veľkú výhodu v tom, že na prácu s objektami v mape používa matematickú knižnicu matplotlib (7), vďaka ktorej je možné jednoducho a hlavne rýchlo vykresliť objekty do Leaflet mapy. Knižnica mplleaflet síce ponúka možnosť jednoducho a rýchlo pracovať s Leaflet ale nebola zvolená prevažne kvôli nedostatočnej starostlivosti od tvorcov, keďže posledná aktualizácia bola naposledy pred 6 rokmi a už dnes jej chýbajú moderné API volania pre Leaflet. Knižnica basemap taktiež na prácu s objektami používa matematickú knižnicu matplotlib (7) a je aj autormi stále aktualizovaná, bohužiaľ ale natívne neumožňuje priamu prácu s knižnicou Leaflet a preto rovnako nebola zvolená.

Branca

Branca je python knižnica s otvoreným kódom určená pre generovanie zložitých HTML a JavaScript stránok, špecificky orientovaná na oblasť vizualizácie dát. Generovanie stránok je založené na Jinja2 (8) šablónovacom systéme čo umožňujúce tvorbu dynamických obsahov. Je vhodná pre projekty, kde je potrebné prezentovať dáta interaktívnym a vizuálne príťažlivým spôsobom a z toho dôvodu ju považujem ako vhodnú na vizualizáciu doplnkov k objektom v mape.

GeoPandas

GeoPandas je python knižnica s otvoreným kódom, ktorá je primárne zameraná na prácu s priestorovými dátovými typmi. Knižnica zahrňuje širokú paletu funkcií ako napríklad zmena z jedného dátového typu do druhého, rôzne výpočty oblastí, dĺžok a priestorových spojení, selekcie založené na geografických atribútoch a podobne. Knižnica umožňuje dáta aj vizualizovať, čo značne uľahčuje programátorskú prácu pri tvorbe interaktívnych máp. Knižnica bola vybraná hlavne kvôli najširšej podpore dátových typov na trhu.

Django Unicorn

Django Unicorn je python knižnica s otvoreným kódom priamo určená pre projekty bežiace na django frameworku. Knižnica ponúka nevšedný prístup k tvorbe webových aplikácií s dynamickým obsahom. V kontraste s tradičným formovaním aplikácie, kde aplikácia je rozdelená na frontend (časť viditeľná užívateľom) a backend (funkcie na pozadí) tak knižnica implementuje funkcie reaktívne priamo s využitím django šablón, čo značne znižuje potrebu ďalších javascriptových knižníc a značne urýchľuje a zjednodušuje vývoj.

Knižnica funguje na koncepte komponentov, teda vzájomne nezávislých súčastí, ktoré sa dajú opakovane použiť v rôznych častiach aplikácie. Tento prístup umožňuje vývojárom vytvárať modulárny a hlavne ľahko udržiavateľný kód. Komponenty sú navrhnuté tak, aby boli samostatné, s vlastnými funkciami a štýlmi, vďaka čomu je ich možné izolovane testovať a vylepšovať bez rizika ovplyvnenia ostatných častí aplikácie. Táto architektúra zároveň podporuje opätovné použitie kódu, čím sa znižuje duplicita kódu v celom projekte. Výsledkom je rýchlejší vývoj, lepšia testovateľnosť a vyššia kvalita konečnej aplikácie.

Na trhu je viac knižníc s obdobne fungujúcou filozofiou, najpoužívanejšie sú Django-components a ReactPy. Django-components ponúka, rovnako ako Django Unicorn, tvorbu obsahu pomocou znovu použiteľných a na sebe nezávislých komponentov. Django Unicorn ide v tomto trochu ďalej a na vrch toho ponúka dodatočné funkcionality, ako napríklad lazy loading (objekt sa načíta až v momente kedy je skutočne potrebný), špeciálnu validáciu formulárov, extra integrácia s Javascriptom vďaka ktorej je možné pristupovať z prostredia pythonu k premenným a funkciám z Javascriptu a mnoho ďalšieho.

ReactPy prináša paradigmu ReactJs (9) do prostredia Pythonu umožňujúc vytvárať komponenty s použitím Pythonu bez použitia JavaScriptu. ReactPy nebol zvolený kvôli nedostatočnej podpore django frameworku a obtiažneho procesu pri vytváraní komponentov. Django Unicorn je knižnica, ktorá nie lenže dobre komunikuje, ale aj pasuje do django ekosystému a preto bola zvolená.

Geocoder

Knižnica Geocoder je napísaná v pythone a umožňuje prevádzať informácie o konkrétnej oblasti na súradnice. Pod informáciami sa myslia rôzne slová, ktoré reprezentujú oblasť

ako napríklad poštová adresa, lokálny názov (napríklad názov firmy) alebo aj miestna prezývka. Táto knižnica sa hodí do vyhľadávania konkrétnych miest v mape.

PostgreSQL

PostgreSQL je objektovo relačný databázový systém s otvoreným kódom, ktorý je unikátny vďaka svojej robustnosti, rýchlosti, jednoduchosti a hlavne podpore rôznych aj nevšedných dátových typov. Databázový systém má veľkú komunitnú podporu, vďaka ktorej je možné doinštalovať do systému modifikácie pridávajúce aj také dátové typy, ktoré natívne databázový systém neponúka. Veľká výhoda PostgreSQL je aj dobrá spolupráca s frameworkom django. Na prácu s databázou, kde je potrebný manuálny zásah v rámci tvorenia aplikácie budeme používať softvér PgAdmin (10).

PostGIS

Postgis je softvér s otvoreným zdrojovým kódom, ktorého úloha je rozšíriť PostgreSQL o geografické dátové typy a umožňuje efektívnu prácu s nimi.

Bootstrap

Bootstrap je otvorená kolekcia CSS a Javascript komponentov, ktoré slúžia vývojárom pre vývoj responzívnych a vizuálne atraktívnych webových stránok s čo najvyššou priehľadnosťou kódu a časovou efektivitou. Bootstrap obsahuje preddefinované štýly a funkcie pre najpoužívanejšie html elementy, ako napríklad tlačidlá, navigačné lišty, zoznamy, formuláre a podobne. Bootstrap je komunitou veľmi používaný a prehľadne zdokumentovaný.

Summernote

V rámci aplikácie nastane situácia, kedy užívateľ bude vpisovať text. Keďže obyčajný text je príliš nudný, bude vhodné, aby sa text dal zvýrazňovať, podčiarkovať, vkladať doň emotikony, obrázky a podobne. Presne toto umožňuje Javascriptová knižnica s otvoreným kódom s pracovným menom Summernote. Knižnica generuje WYSIWYG (What You See Is What You Get) editor, ktorý umožňuje užívateľom jednoducho vytvárať a upravovať obsah priamo v prehliadači.

HTMX

HTMX je JavaScriptová knižnica s otvoreným kódom, ktorej úlohou je rozšíriť značkovací jazyk HTML o nové atribúty, ako napríklad zmena štýlu elementu na základe serverovej odpovede. HTMX komunikuje so serverom pomocou AJAX požiadaviek, alebo WebSocket pripojenia.

JQuery

JQuery je kompaktná JavaScriptová knižnica, ktorej úlohou je zefektívniť interakciu medzi HTML elementmi a JavaScriptom. Súčasťou sú aj AJAX metódy, ktoré prídu vhod prevažne kvôli komunikácií s časťou napísanou v jazyku Python.

JQuery UI

JQuery UI je nadstavba pre JavaScriptovú knižnicu JQuery, ktorej úlohou je zjednodušiť vývojárom implementáciu dynamických užívateľských prostredí. Je navrhnutá tak, aby fungovala efektívne bez výrazného spomalenia načítania webovej stránky.

Výhody a nevýhody

Použitím tejto súpravy technológií sme schopný zrealizovať webovú aplikáciu s atraktívnym užívateľským prostredím fungujúcim v reálnom čase. Hlavnými prednosťami je pohodlná tvorba komplexných interaktívnych užívateľských prostredí, častá komunikácia so serverom zaručujúca obsah v reálnom čase a celková prehľadnosť kódu kvôli deleniu do komponentov. Z pohľadu čistoty, efektívnosti a bezpečnosti softvéru je táto možnosť vhodná, ale rozdelenie aplikácie na časť pre klienta a server má aj svoje slabé stránky. Je nutné simultánne pracovať na dvoch oddelených projektoch, čo zvyšuje množstvo práce, pridáva komplikácie pri riešení problémov a pridáva potrebu znalostí veľkého množstva rôznych knižníc.

2 Návrh aplikácie

V tejto kapitole bude popísaný návrh aplikácie, jej funkcionality s podrobným popisom, štruktúra aplikácie, návrh databázového modelu a taktiež aj špecifikácia požiadaviek.

2.1 Architektúra aplikácie: Klient-server

Najpoužívanejšou a pre túto aplikáciu asi najvýhodnejšou architektúrou je architektúra klient – server. Architektúra funguje tak, že softvér klienta (webový prehliadač) inicializuje komunikáciu so serverom, ktorý následne požiadavku spracuje a odošle odpoveď klientovi. Táto architektúra je vhodná aj preto, že výpočtová práca je bremenom servera a teda klient potrebuje disponovať výpočtovým výkonom len na spracovanie výsledku, nie na jeho generovanie. Netreba opomenúť aj to, že zdrojové kódy sú uložené na strane servera a teda pre klienta nedostupné čo znižuje riziko cieleného vyhľadávania bezpečnostných dier v kóde.

2.2 Funkcionálne požiadavky

Funkcionálne požiadavky softvéru sú jasne definované a merateľné vlastnosti, ktoré softvér musí splniť aby softvér napĺňal požadované ciele.

Webová aplikácia musí spĺňať:

- **Autorizácia a autentifikácia užívateľa**
- **Vykreslenie podkladovej mapy**
- **Uloženie pozícií objektov a informácií o nich do databázy**
- **Možnosť roztriediť objekty do podskupín a tie do skupín**
- **Vykreslenie objektov z databázy na podkladovú mapu**
- **Interaktívne prvky (priblíženie, oddialenie, režim celej obrazovky, zobrazenie aktuálnej polohy)**
- **Možnosti sociálnej interakcie (pridanie priateľa, zdieľanie objektu, diskusia, zhromaždenie užívateľov do skupín)**
- **Vyhľadávanie pozície na mape na základe poštovej adresy**

2.3 Kvalitatívne požiadavky

Kvalitatívne alebo nefunkcionálne požiadavky sú také požiadavky, ktoré opisujú vlastnosti softvéru a neopisujú priamo špecifickú funkcionalitu softvéru.

Webová aplikácia musí spĺňať:

- **Responzivita podkladovej mapy (veľkosť sa musí prispôsobiť rozlíšeniu obrazovky užívateľa)**
- **Intiutívnosť**
- **Aktualizácie by mali byť distribuované bezproblémovo a bez ovplyvnenia dostupnosti mapy**

2.4 Rozdelenie chránených oblastí

Chránené oblasti na mape by mohli byť vykreslené len ako jednotlivé separované oblasti. Takýto prístup by mal ale veľké množstvo problémov (napríklad nemožnosť efektívneho filtrovania) a preto je potrebné dané oblasti nejako zjednotiť. Toto rozdelenie už za nás je viac menej legislatívne vyriešené. Každá oblasť patrí do nejakej väčšej kategórie, napríklad „národné prírodné pamiatky“. Takéto delenie len na skupiny by mohlo byť dostačujúce, ale v prípade väčšieho množstva na prvý pohľad podobných kategórií trochu otravné. Preto mi príde vhod objekty vkladať do podskupín a tie následne do skupín, napríklad: OP NPP Brestovská jaskyňa -> Národné prírodné pamiatky -> Pamiatky.

2.5 Užívateľské role

Užívatelia v rámci aplikácie budú mať povolené alebo obmedzené svoje pole pôsobnosti na základe svojej užívateľskej role. Užívateľské role by mali byť na pochopenie čo najjednoduchšie a užívateľsky najprívetivejšie. Problémom tohto prístupu ale je, že jasne určuje hranice každému užívateľovi čo môže byť veľmi frustrujúce. Z tohto dôvodu som sa rozhodol tento prístup okoreniť o individuálne prístupy. Napríklad konkrétny užívateľ bude môcť danú mapu upravovať a iný zas len prehliadať, ďalší ju neuvidí vôbec. Takýto príklad by sa s jasne stanovenými pravidlami realizoval len veľmi ťažko. Aplikácia bude vo svojom jadre poznať tri užívateľské role a to neprihlásený užívateľ, prihlásený užívateľ a administrátor.

Oprávnenia neprihláseného užívateľa:

- Voľné prehliadanie podkladovej mapy a globálne dostupných vrstiev
- Registrácia, prihlásenie

Oprávnenia prihláseného užívateľa:

- Obsahuje všetky oprávnenia neprihláseného užívateľa

- Odhlásenie
- Tvorba vlastných vrstiev
- Úprava vlastných vrstiev
- Zdieľanie vlastných vrstiev s priateľmi
- Mazanie vlastných vrstiev a obnovenie z koša
- Účasť v diskusií ku konkrétnej vrstve
- Vyhľadanie užívateľského profilu
- Pridanie priateľa
- Odstránenie priateľa
- Blokovanie užívateľa
- Vyhľadávanie a pridanie sa do užívateľských skupín
- Tvorba a spravovanie vlastnej skupiny
- Posielanie súkromných správ iným užívateľom

Oprávnenia administrátora:

- Obsahuje všetky oprávnenia prihláseného užívateľa
- Neobmedzený prístup do administrácie
- Úprava všetkých globálnych vrstiev, skupín a podskupín
- Pridávanie globálnych vrstiev, skupín a podskupín
- Nahrávanie a sťahovanie globálnych vrstiev pomocou JSON formátu.

2.6 Schéma aplikácie

Aplikácia je založená na MTV štruktúre vychádzajúcej z frameworku Django, teda aplikácia bude rozdelená na 3 nezávislé časti.

Model

Model je akýsi reprezentant databázovej dátovej štruktúry, ktorého úloha je mapovať databázové záznamy na programátorské objekty a naopak. Model obsahuje vlastné atribúty, ktoré sa prekladajú ako stĺpce v tabuľke.

View

View prijíma prichádzajúce požiadavky od klienta a následne ich programátorsky spracuje a pošle do šablóny (Template) na ďalšie spracovanie.

Template

Úlohou šablóny je spracovať informácie získané z view a následne ich správne reprezentovať vo výstupe. V našom prípade výstup je HTML kód, ktorý ale treba generovať dynamicky. Presne k tomuto účelu šablóny slúžia. Šablóny všeobecne používajú svoj vlastný jazyk so špecifickou syntaxou pre jasné oddelenie od HTML.

2.7 Návrh databázy

Konkrétne vyhotovenie databázovej štruktúry necháme na django framework a našou úlohou bude iba navrhnúť reprezentantov v rámci Modelov. Reprezentanti v modeloch budú nasledovný:

Objekty – Slúži na uchovanie geografických dát o konkrétnej oblasti, jej meno, CSS štýl, popis v HTML formáte a bude si držať informáciu o stupni ochrany.

Viditeľnosť – Úlohou tohto modelu bude uchovávať si informáciu o viditeľnosti, teda prístupové práva užívateľa. Každý iný model, ktorého vlastnosti sa menia podľa prístupových oprávnení bude reláciu s týmto modelom.

Skupiny – Uchováva meno, reláciu na viditeľnosť, správcu a diskusiu.

Podskupiny - Obsahuje meno, reláciu na viditeľnosť, správcu, skupinu.

Diskusia – Definuje užívateľskú diskusiu a drží informácie o správcovi, aktívnym odberateľoch notifikácií a aktivite.

Diskusný príspevok - Obsahuje reláciu na autora a diskusiu ku ktorej príspevok patrí, časovú známku, samotný text v html formáte a karmu príspevku.

Profil – Každý registrovaný užívateľ bude mať reláciu 1 k 1 s užívateľským profilom, ktorý bude udržiavať informácie o užívateľovi ako base64 hash (11) užívateľskej ikonky, užívateľský popis, trvalé bydlisko, dátum narodenia, dátum registrácie, nastavenia mapy, odkazy na sociálne siete ako Instagram, YouTube, Facebook, LinkedIn či osobnú webstránku.

Notifikácie – Eviduje relácie s prijímateľom a odosielateľom, správu v HTML formáte, časovú známku a informáciu, či užívateľ notifikáciu prečítal.

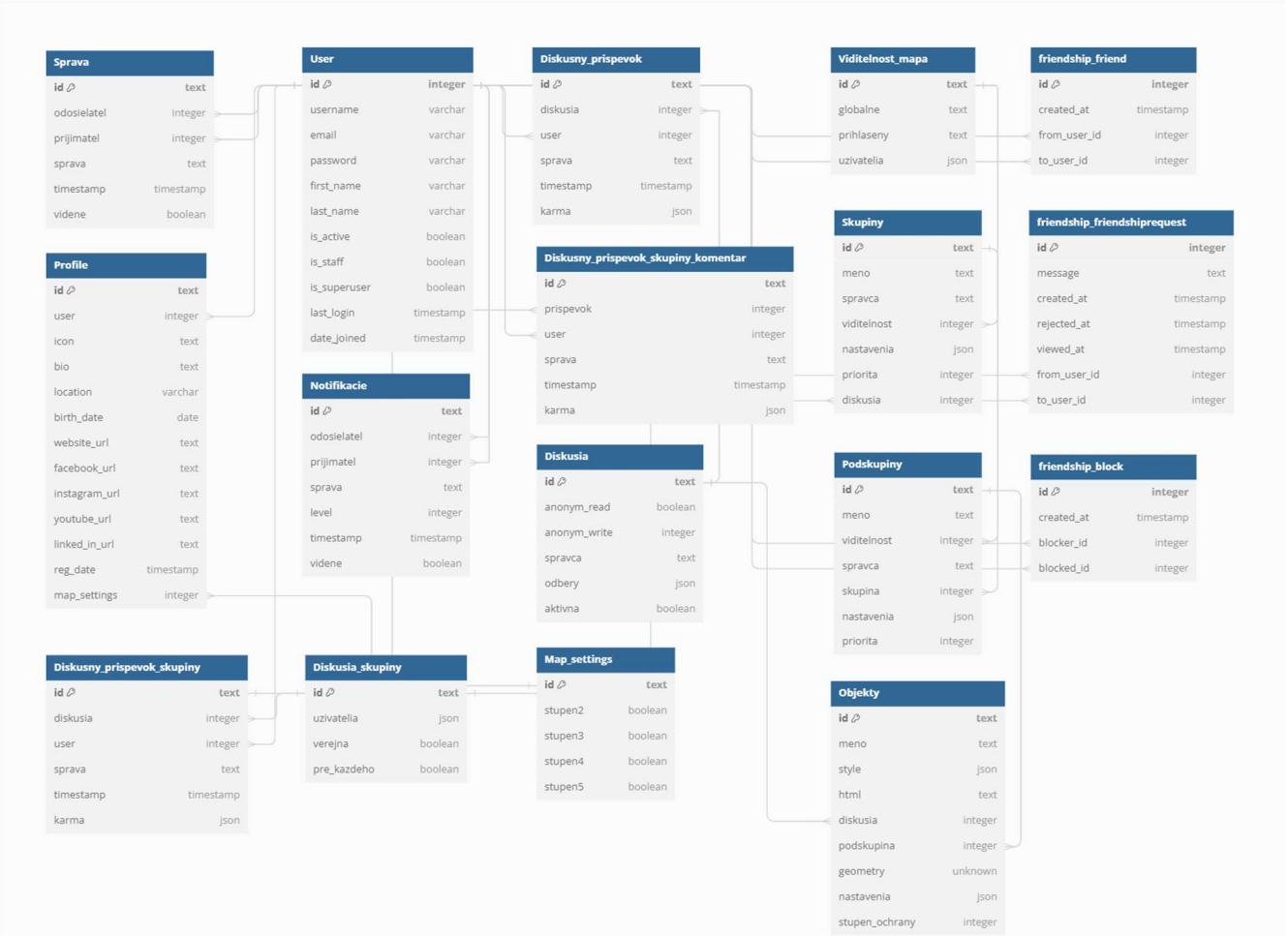
Sprava - Definuje relácie s odosielateľom a prijímateľom, správu v HTML formáte, časovú známku a informáciu, či užívateľ správu prečítal.

Friendshiprequest – Drží informáciu o žiadosti o priateľstvo.

Friend – Drží informáciu o aktívnom priateľstve medzi dvomi užívateľmi.

Block – Drží informáciu o blokovaní užívateľa A užívateľom B

Pre zobrazenie štruktúry databázy použijeme entitno-relačný diagram, ktorý slúži na vizualizovanie databázových tabuliek, atribútov a vzájomných prepojení.



Obrázok 4: Entitno-relačný diagram databázy

2.8 Návrh komponentov

Aplikácia by mala byť tvorená s navzájom nezávislých komponentov. Komponent sa myslí nejaká časť webstránky, napríklad navigácia. Môže sa ale stať, že niektoré komponenty sú zdieľané naprieč stránkami (ako napríklad navigácia) a teda ich nie je možné jasne priradiť ku konkrétnej webovej podstránke. V kontraste, niektoré podstránky sú funkcionálne natoľko jednoduché, že delenie na komponenty by programátorskú časť iba skomplikovalo. Táto podkapitola teda bude obsahovať návrh komponentov v mixe s podstránkami webovej aplikácie.

Prihlásenie

Táto podstránka slúži čisto iba na prihlásenie užívateľa. Mala by obsahovať pole pre prihlasovacie meno, pole pre prihlasovacie heslo, tlačidlo pre vykonanie prihlásenia a odkaz na podstránku pre registráciu neregistrovaných užívateľov. Podstránka by mala správnosť mena a hesla nielen validovať, ale o nesprávnosti aj informovať. Po úspešnom prihlásení by užívateľ mal byť presmerovaný na podstránku z ktorej prišiel.

Registrácia

Registrácia by mala obsahovať pole pre užívateľské meno, heslo, opakovať heslo, email, adresu trvalého bydliska a dátum narodenia. Podstránka by mala registráciu validovať a v prípade nesúladu s pravidlami o tom užívateľa informovať. Po registrácií by užívateľ mal byť presmerovaný na hlavnú stránku.

Notifikačná roletka

Tento komponent by mal nosiť všetky užívateľom neprečítané notifikácie. Ak nastane udalosť o ktorej by užívateľ mal byť informovaný, tak v rámci notifikačného systému by o tom mal byť informovaný v reálnom čase.

Navigácia

Súčasťou podstránok by mala byť navigácia. Navigácia bude slúžiť ako súbor odkazov / tlačidiel na ďalšie podstránky.

Mapa

Pre aplikáciu asi najpodstatnejší komponent. Komponent by okrem podkladovej mapy mal obsahovať súbor tlačidiel pre prechod do režimu na celú obrazovku, priblíženie alebo oddialenie, pre prihlásených užívateľov možnosť dodatočných nastavení, vyhľadávacie tlačidlo a zoznam skupín s možnosťou filtrácie. Komponent ako celok by mal byť plne responzívny, teda sa prispôsobiť voľnému priestoru na stránke tak, aby sa mapa zobrazovala v celej veľkosti bez zrezania.

Diskusný príspevok

Komponent by mal vykresliť diskusný príspevok, tak aby obsahoval profilovú fotografiu autora, text, čas uverejnenia príspevku a karma systém.

Diskusia

Tento komponent bude združovať všetky užívateľmi vpísané príspevky a umožňuje v rámci diskusie ich aj vygenerovať a zoradiť buď chronologicky alebo podľa karma bodov. Tento komponent by mal umožňovať komunikáciu s notifikačným systémom tak, aby v prípade, že si užívateľ praje dostávať upozornenia z danej diskusie, tak ich aj dostane.

Kôš

V rámci košu by mal mať užívateľ možnosť vrátiť naspäť svoje zmazané objekty alebo ich odstrániť natrvalo. Kôš by mal obsahovať názov objektu ako aj jeho podskupinu a skupinu pre jasnú identifikáciu.

Priatelia

Komponent by mal vykresliť všetkých spriatelených užívateľov. V rámci komponentu by mal užívateľ mať možnosť pozrieť si s užívateľom zdieľané vrstvy, prezrieť si jeho profil, zablokovať priateľa alebo zrušiť priateľstvo.

Žiadosti o priateľstvo

Komponent vypíše všetky aktívne žiadosti o priateľstvo od iných užívateľov, pričom užívateľ bude môcť žiadosť prijať alebo odmietnuť. Rovnako by v rámci komponentu mal existovať zoznam užívateľom odoslaných žiadostí s možnosťou stiahnutia žiadosti.

Hľadanie užívateľa

Tento komponent by mal obsahovať vyhľadávacie pole, kde po zadaní čo i len časti mena užívateľa, by sa mali nájsť všetky zhody s možnosťou odoslania žiadosti o priateľstvo a prezretia si užívateľského profilu.

Skupiny

Obsahuje zoznam skupín v ktorých je užívateľ pridaný. Zoznam by mal obsahovať názov skupiny, prechod na ňu, popis, počet členov, počet príspevkov.

Tvorba skupiny

Mal by zabezpečovať možnosť užívateľovi vytvoriť vlastnú skupinu. Komponent by mal obsahovať textové pole pre názov a popis a taktiež aj možnosť voliť, či je skupina verejne prístupná na vyhľadanie a či sa do nej môže verejne každý pridať.

Vyhľadávanie skupín

Komponent by mal obsahovať vyhľadávacie pole, ktoré hľadá zhodu s názvami verejne dostupných skupín. V prípade zhody vráti užívateľovi zoznam zhodných skupín s možnosťou ich prezretia a v prípade verejnej prístupnosti aj pridania sa do nich.

Administrácia – vrstvy

V rámci tohto komponentu budú oprávnené osoby môcť upraviť oprávnenia alebo upraviť konkrétne vrstvy v skupinách alebo podskupinách. Administrátori budú môcť pridať skupiny a podskupiny a odstrániť tie skupiny alebo podskupiny, ktoré neobsahujú žiadne vrstvy.

Užívateľský profil

Tento komponent bude obsahovať súhrn informácií o užívateľovi ako meno, profilová fotografia, popis, počet priateľov a počet vytvorených objektov. Ostatný užívatelia budú môcť cez profil napísať správu dotyčnému užívateľovi.

3 Implementácia aplikácie

V tejto kapitole budú popísané konkrétne kroky nutné pre úspešnú implementáciu aplikácie na základe návrhu z predchádzajúcej kapitoly ale aj rozobraté riešenia výziev, ktoré prirodzene vznikli počas vývojového procesu alebo z podstaty projektu.

Nadobudnutie dát o chránených oblastiach

Základnou nutnou podmienkou pre správne fungovanie aplikácie, ktorá pracuje s chránenými oblasťami je prístup k dátam, kde sa aká oblasť nachádza. Táto informácia nie je verejne dostupná, respektíve surové dáta sa mi nepodarilo štandardnými metódami zohnať a preto bolo nutné kontaktovať zodpovednú osobu zo Štátnej ochrany prírody Slovenskej republiky. Dáta mi vo veľmi krátkom čase na účely použitia k bakalárskej práci boli poskytnuté a pravidelne nebol problém ich opakovane vyžiadať pre zaručenie aktuálnosti údajov v mape.

3.1 Implementačné prostredie

Každý vývoj softvéru so sebou nesie nejaké implementačné prostredie. Podľa návrhu budeme používať programovací jazyk Python s Django frameworkom. Verziu Pythonu nie je pre túto aplikáciu dôvod riešiť inak ako použitím najnovšej práve dostupnej verzie (

3.12.2). S Django frameworkom to už také jasné nebude. Aktuálne máme na výber najnovšiu verziu (5.0.4 – latest) a verziu s predĺženou podporou (4.2.11 – long-term support). Intuitívne je prirodzené siahať rovno po najnovšej verzií, ale Django ponúka 2 rôzne typy verzií a to latest – teda najnovšia verzia poskytujúca najnovšie funkcie a long-term support čo je verzia, ktorá obsahuje časom overené funkcionality a jej podpora a udržiavanie je omnoho dlhšia. Rozhodol som sa zvoliť LTS verziu vzhľadom k tomu, že latest verzia nie je ešte na trhu dosť dlho a nie je komunitou preverená, čo môže viesť k nečakaným vlastnostiam alebo priamej nekompatibilite s existujúcimi knižnicami. Taktiež funkcionality, ktoré najnovšia verzia obsahuje nie sú pre potreby aplikácie nijako potrebné.

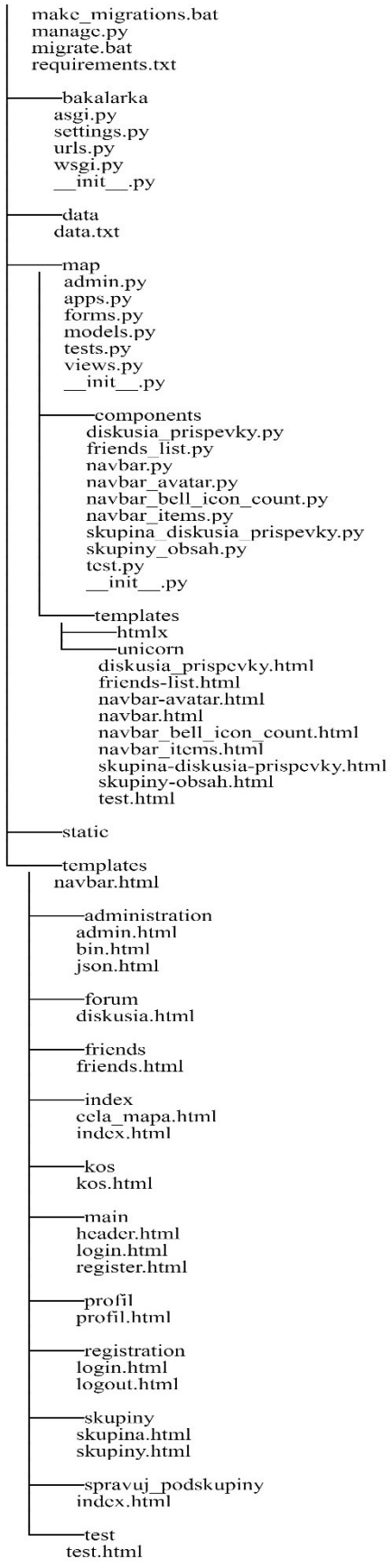
Vývojové prostredie

Vývojové prostredie je softvér, ktorý umožňuje vývojárovi produkovať, testovať a ladiť programátorský kód. Na trhu existuje veľké množstvo vývojových prostredí s rôznymi pomocnými funkciami. Hlavným kritériom je cena a pre vývoj aplikácie budem uprednostňovať také vývojové prostredia, ktoré sú buď bezplatné, alebo bezplatné pre študentov. Ďalším hlavným kritériom je jeho popularita a uznávanosť v programátorskej komunite, keďže každé vývojové prostredie má svoju krivku učenia a bez komunitnej pomoci môže byť problém v prostredí efektívne pracovať. Vybral som si vývojové prostredie PyCharm Professional, keďže okrem jednoduchého rozbehnutia projektu vo virtuálnom prostredí, dopĺňania kódu a automatizovaného testovania ponúka aj podporu pre Django framework, nástroje pre prácu priamo s databázami a vývojové prostredie nie je nutne viazané len na jazyk python ale dokáže pracovať aj s JavaScriptom, HTML, CSS a mnoho ďalšími čo v prípade vývoja tejto aplikácie príde vhod a preto bolo toto vývojové prostredie vybrané.

Štruktúra projektu

V Django frameworku je aplikácia súčasťou nejakého projektu, kde projekt môže obsahovať viacero aplikácií. Ja som sa z dôvodu čitateľnosti a jednoduchej orientácií v štruktúre rozhodol v rámci jedného projektu mať jednu aplikáciu a až tú následne deliť do komponentov.

Výsledná štruktúra projektu bude vyzerať nasledovne:



Obrázok 5: Štruktúra projektu

V zložke bakalarka sú uložené projektové súbory. Najzaujímavejšie sú settings.py a urls.py.
V settings.py sa nachádzajú nastavenia projektu, ako napríklad jazyk webstránky, inštalované aplikácie, prihlasovacie údaje do databázy a podobne. Súbor ulrs.py obsahuje vzory, ako má aplikácia priradiť URL adresu konkrétnej funkcií, ktorá vykonáva zobrazenie. Štruktúra urls.py bude vyzerať nasledovne:

```
from django.contrib import admin

from django.urls import path, include

from map import views as map_views

urlpatterns = [

    path('login',map_views.login_request),

    path('',map_views.index,name='index'),

    path('diskusia',map_views.forum,name='forum'),

    path('skupiny',map_views.skupiny_request,name='skupiny'),

     ...
]
```

Adresár data obsahuje surové dáta poskytnuté ŠOPSR.
Adresár map obsahuje všetky súbory aplikácie. Najzaujímavejšie sú forms.py, views.py a models.py. Adresár taktiež obsahuje podadresár s komponentami aplikácie.

Súbor models.py obsahuje implementáciu Django modelov. Názorná ukážka jedného z modelov:

```
class Sprava(models.Model):
    odosielatel = models.ForeignKey(User, on_delete=models.CASCADE,related_name='odosielatel_sprava')
    prijimatel = models.ForeignKey(User, on_delete=models.CASCADE,related_name='prijimatel_sprava')
    sprava = models.TextField(blank=True, null=False,default="")
    timestamp = models.DateTimeField(default=timezone.now,null=False)
    videne = models.BooleanField(default=False)
```

Obrázok 6: Ukážka models.py

Súbor views.py obsahuje zobrazenia respektíve funkcie, ktoré spracovávajú užívateľské požiadavky.

```
def login_request(request):
    errors = {}
    if request.method == "POST":
        form = AuthenticationForm(request,data=request.POST)
        if form.is_valid():
            user = form.get_user()
            login(request, user)
            if "next" in request.GET:
                return redirect(request.GET.get('next'))
            return redirect('/')
        errors = form.errors
    form = AuthenticationForm()
    return render(request=request, template name="main/login.html", context={"register form": form,"errors": errors})
```

Obrázok 7: Náhľad do views.py

Nasledujúci kód definuje funkciu generujúcu odpoveď na požiadavku užívateľa, ktorej úlohou je overiť formulár užívateľa pokúšajúceho sa o prihlásenie. V prípade úspešného overenia validity formuláru funkcia užívateľa prihlási, inak mu budú v odpovedi poslané chybné hlášky.

Súbor forms.py obsahuje Django formuláre. Formuláre predstavujú spôsob, ako užívateľ môže upraviť dáta v rámci aplikácie. Každý formulár je naviazaný na Django model od ktorého sa odvíjajú aj údaje s ktorými formulár pracuje a ďalej validuje.
Názorná ukážka formuláru z forms.py:

```
class NewUserForm(UserCreationForm):
    email = forms.EmailField(required=True)
    location = forms.CharField(required=True, min_length=1, max_length=75, label="Adresa")
    date_of_birth = forms.DateField(required=True, label="Dátum narodenia")
    class Meta:
        model = User
        fields = ("username", "email", "password1", "password2")

    def clean_date_of_birth(self):
        dob = self.cleaned_data['date_of_birth']
        today = date.today()
        if (dob.year + 18, dob.month, dob.day) > (today.year, today.month, today.day):
            raise forms.ValidationError('Registrácia nie je povolená osobám mladším ako 18 rokov.')
        return dob

    def save(self, commit=True):
        user = super(NewUserForm, self).save(commit=False)
        user.email = self.cleaned_data['email']
        dob = self.cleaned_data['date_of_birth']
        loc = self.cleaned_data['location']
        if commit:
            user.save()
        profil = Profile.objects.create(user=user,map_settings=Map_settings.objects.create())
        profil.birth_date = dob
        profil.location = loc
        profil.save()
        return user
```

Obrázok 8: Ukážka forms.py

Adresár templates obsahuje všetky Django šablóny. V rámci týchto šablón sa premenné z views.py aplikujú pomocou {{ premenná }}, alebo v prípade komplexnejšiej logiky {% príkaz %}.
Názorná ukážka šablóny:

```
{% for notification,profil in notifications_unread reversed %}
                <li>
                    {% if profil %}
                    <div class="col-md-3 col-sm-3 col-xs-3"><div class="notify-img"></div></div>
                    {% endif %}
                    <div class="col-md-9 col-sm-9 col-xs-9" style="margin-left: 60px">
                        <a href="">{{ notification.odosielatel }}</a>
                        <p style="margin-bottom: auto;">{{ notification.sprava|safe }}</p>
                        <a u:click="videne({{ notification.id }})" onclick="Unicorn.call('navbar_bell_icon_count', 'dekrement');" >

                    <p class="time" >
                        {{ notification.timestamp }}</p>
                    </div>

                </li>
                <hr style="margin-top: 0">


                {% endfor %}
```

Obrázok 9: Ukážka šablóny

3.2 Bezpečnosť

Každá verejne dostupná webová aplikácia by sa mala snažiť chrániť dáta umiestnené v danej lokalite, ako aj celkovú integritu aplikácie. Počet útokov je v posledných rokoch na vzostupe (12) (13) a preto treba o to viac dbať na bezpečnosť.

Šifrovanie komunikácie

Úplne základný bezpečnostný prvok aplikácie je šifrovanie komunikácie medzi serverom a klientom. Pre zabezpečenie šifrovania použijeme protokol HTTPS. Pre tento protokol si budeme musieť zaobstarať SSL certifikát od nejakej certifikačnej autority. Na trhu existuje veľké množstvo certifikačných autorít, ponúkajúcich okrem produkcie SSL certifikátu rôzne doplnkové služby, ale v tomto prípade budeme hľadieť len na certifikát a preto som zvolil Let's Encrypt, ktorá vydáva certifikáty bezplatne. Okrem šifrovania si klient môže byť istý, že pristupuje na danú webovú lokalitu bez podvrhnutia obsahu treťou stranou na rovnakej sieti.

Cross Site Scripting (XSS)

XSS útok umožňuje útočníkovi v prípade nedostatočného zabezpečenia webovej aplikácie vložiť ostatným klientom rôznu do aplikácie pôvodne nepatriacu časť kódu. Riešením, ako sa tomuto typu útoku vyhneme, je použitie takzvaného escape tagu, ktorý django framework ponúka.

Cross Site Request Forgery

CSRF útok pri ktorom útočník napríklad podstrčí užívateľovi infikovanú webovú linku vďaka ktorej získa oprávnenia daného užívateľa na ktorého útočí a môže na danej lokalite

robiť akcie v jeho mene. Aby sme zabránili takémuto typu útoku, tak pri každom formulári v ktorom sa používa http požiadavka potrebujeme použiť csrf_token Django tag.

SQL Injenction

SQL Injenction je typ útoku, pri ktorom útočník zneužíva vlastnosti jazyka SQL a do formulárov vkladá škodlivý SQL kód, vďaka ktorému môže získať neoprávnené dáta, alebo obísť autentifikačné mechanizmy. Aby sme sa vyhli takémuto typu útoku, budeme sa v implementácií vyhýbať priamemu styku SQL dopytov s kódom a budeme iba používať Django ORM, ktoré tento typ útoku rieši za nás.

3.3 Nasadenie aplikácie na server

Aby aplikácia bola verejne prístupná pomocou URL adresy, je potrebné aplikáciu nasadiť na produkčný server s verejne prístupnou IPv4 adresou a zabezpečiť prepojenie s doménou. Cieľom tejto podkapitoly je popísať proces, akým sa zabezpečí sprístupnenie projektu pomocou domény mapujeme.sk .

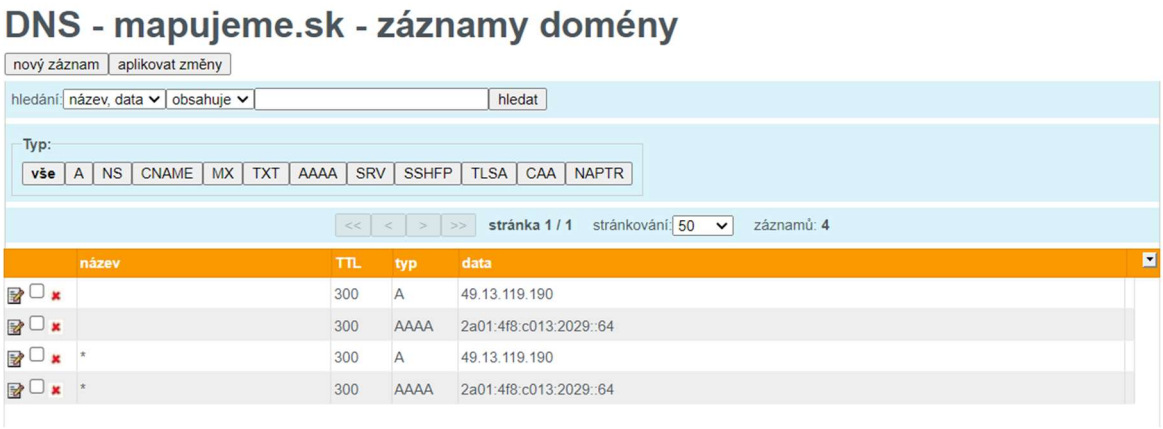
Produkčný server

Pre produkčný server som vybral systém Ubuntu verzie 22.04 keďže táto verzia bude podporovaná až do roku 2027. Server musí hlavne byť stabilne prístupný väčšinu času a musí byť verejne prístupný administrátorom pre bezpečné spravovanie cez internet bez nutnosti fyzického prístupu. To sa dá zabezpečiť pomocou sieťového protokolu SSH.

Nasadenie aplikácie

Pre úspešné nasadenie aplikácie na produkčný server je potrebné splniť zopár podmienok. Ako prvé nainštalujeme Python a všetky potrebné balíčky. Následne nainštalujeme PostgreSQL bez dodatočných úprav. Ďalším krokom je nainštalovanie Apache2 webového servera a mod_WSGI modifikácie, vďaka ktorej bude Apache2 webserver hosťovať našu webovú aplikáciu. Aby bola zaručená funkčnosť CSRF tokenu, potrebujeme zabezpečiť šifrovanú komunikáciu medzi klientom a serverom, teda spojazdniť HTTPS protokol. Pre spojazdnenie HTTPS protokolu potrebujeme nie len SSL certifikát ale aj jeho pravidelnú obnovu. Túto službu obnovy ako aj samotné generovanie SSL certifikátu vie zabezpečiť aplikácia certbot. Na generáciu SSL certifikátu musíme doméne zadať IPv4 a IPv6 adresu

servera a k tomuto účelu slúžia DNS záznamy typu A a AAAA.



Obrázok 10: mapujeme.sk - DNS záznamy

Aj na strane servera musíme nastaviť komunikáciu s doménou. Pre nastavenie hostiteľov sa v Ubuntu nachádza súbor s cestou /etc/hosts, ktorý nastavíme následovne:

```
127.0.0.1 localhost
127.0.0.1 mapujeme.sk
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Obrázok 11/etc/hosts

Ako posledný krok musíme vytvoriť konfiguračný súbor pre webový server Apache2 s inštrukciami ako presne má hosťovať webovú aplikáciu. Tento konfiguračný súbor nazveme mapa.conf a vložíme do priečinku /etc/apache2/sites-available .
Konfiguračný súbor vyzerá nasledovne:

```
<VirtualHost *:80>
    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
</VirtualHost>

<VirtualHost *:443>
        ServerName mapujeme.sk

        ServerAdmin webmaster@mapujeme.sk

    WSGIDaemonProcess myproject python-path=/home/mapa-chranenych-oblasti-sr
        WSGIProcessGroup myproject
    WSGIScriptAlias / /home/mapa-chranenych-oblasti-sr/bakalarka/wsgi.py

        <Directory /home/mapa-chranenych-oblasti-sr/bakalarka>
                <Files wsgi.py>
                        Require all granted
                </Files>
        </Directory>


        Alias /static /home/mapa-chranenych-oblasti-sr/static
    <Directory /home/mapa-chranenych-oblasti-sr/static>
        Require all granted
    </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined


    Include /etc/letsencrypt/options-ssl-apache.conf

    SSLCertificateFile /etc/letsencrypt/live/mapujeme.sk/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/mapujeme.sk/privkey.pem


</VirtualHost>
```

Obrázok 12: Konfiguračný súbor pre Apache2

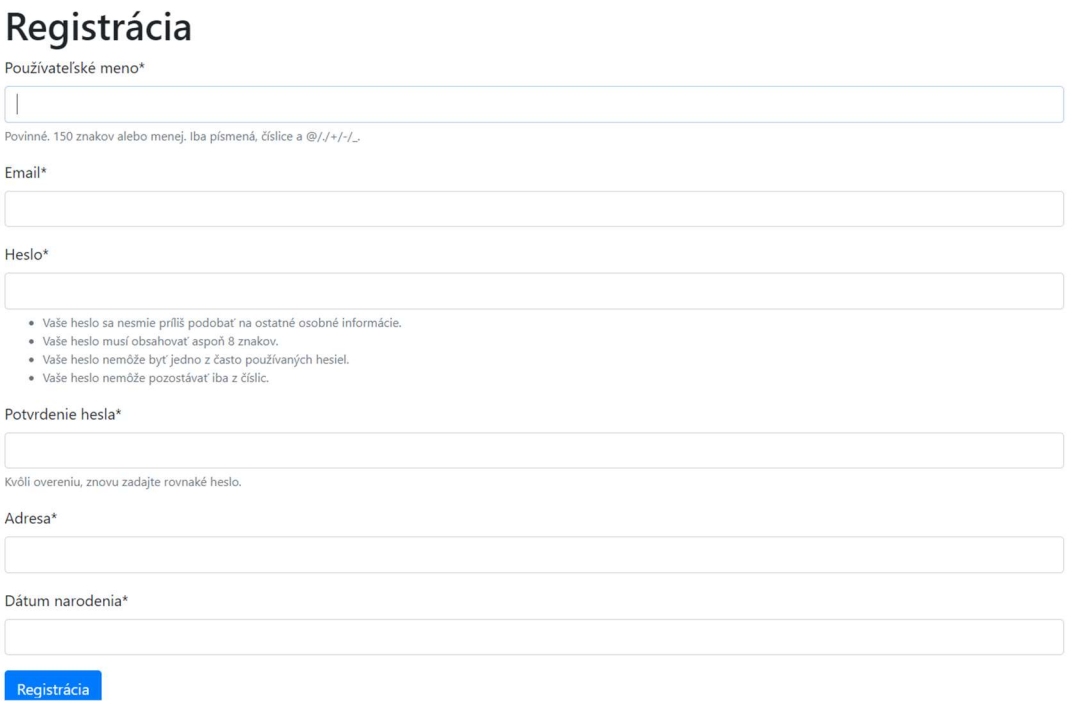
Tag virtualhost určuje na ktorom porte má aplikácia počúvať a v jeho vnútri sa nachádzajú konkrétne vlastnosti. V prvom tagu server počúva na porte 80, čo je štandardný HTTP port. Jediná práca, ktorú má tento tag na starosti je, že má presmerovať všetku HTTP komunikáciu na HTTPS. V druhom tagu server počúva na porte 443, čo je vyhradený port pre HTTPS komunikáciu. Značka ServerName definuje doménový názov, ktorý má webový server spracovávať. Značka WSGIDaemonProcess nesie cestu k Django projektu. Nasledujú povoľovacie značky pre umožnenie prístupu webovému serveru k Django projektu a definovanie kam sa majú ukladať prípadné chybové hlášky. Ako posledné je nutné definovať umiestnenie SSL certifikátov, ktoré sú generované aplikáciou certbot.

3.4 Užívateľské prostredie

V tejto podkapitole sa zameriame na praktickú realizáciu tých častí aplikácie, s ktorými môže užívateľ interagovať a sú nosné pre maximálny užívateľský zážitok. Prostredníctvom konkrétnych príkladov predstavíme dizajn a funkčnosť jednotlivých komponentov a objasníme aj logiku, ktorá stojí za ich fungovaním.

Registrácia

Aby používateľ mohol využiť všetky možnosti aplikácie v plnej kvalite, je potrebné aby sa zaregistroval. V rámci registrácie sa požaduje iba minimum informácií ako používateľské meno, heslo, dátum narodenia a korešpondenčná adresa.



Obrázok 13: Registrácia

Po úspešnej registrácií je užívateľ presmerovaný na hlavnú stránku. V prípade nesprávneho užívateľského vstupu (napríklad už existujúce užívateľské meno) je užívateľ o tejto skutočnosti upozornený a registrácia neprebehne.

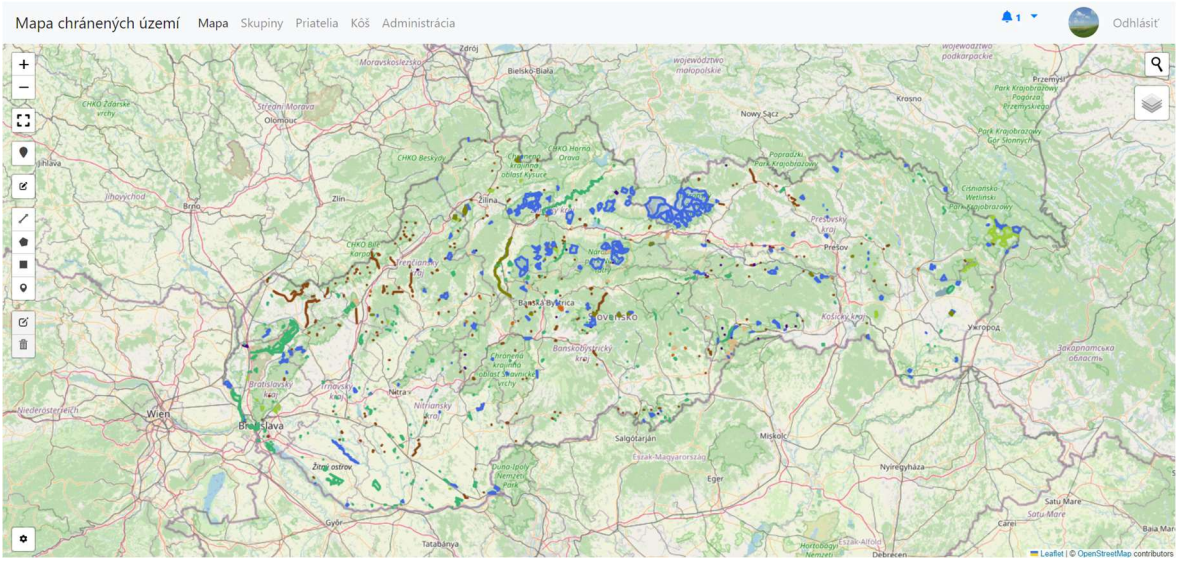
Prihlásenie

Pomocou prihlasovacieho formulára sa môže prihlásiť každý zaregistrovaný užívateľ. Každý komponent, ktorý pre svoje fungovanie vyžaduje prihlásenie namiesto svojho načítania presmeruje užívateľa na prihlasovací formulár.

Obrázok 14: Prihlásenie

Mapa

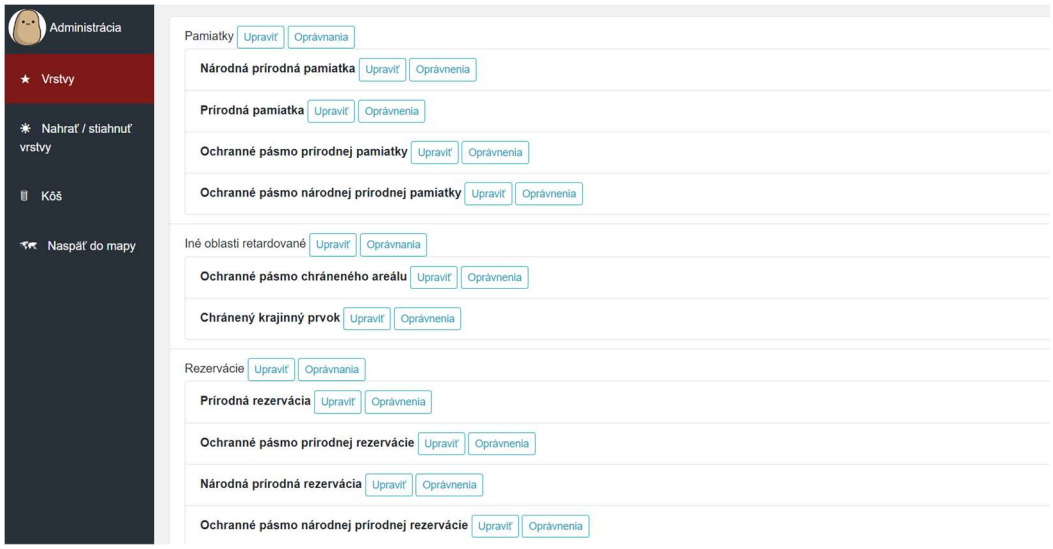
Úvodnou a hlavnou stránkou aplikácie je samotná mapa. Mapa obsahuje štandardné tlačidlá na obsluhu, tlačidlá na tvorbu objektov a režim úpravy. V režime úpravy je možné hromadne upravovať práve na mape zobrazené užívateľom vytvorené objekty.



Obrázok 15: Úvodná stránka

Administrácia

Administrácia slúži na spravovanie systémových vrstiev. V rámci administrácie je možné pridať skupiny, podskupiny, upraviť vrstvy v nich, zmeniť oprávnenia (napríklad, aby určitú skupinu videli iba prihlásený užívatelia) alebo ich vymazať. Administrátor môže poveriť kohokoľvek oprávnením úpravy systémovej skupiny alebo podskupiny a tým mu sprístupniť administráciu s možnosťou úpravy danej skupiny alebo podskupiny.



Obrázok 16: Aministrácia

4 Test použiteľnosti

V rámci testu použiteľnosti som oslovil 5 rôznych ľudí rôzneho veku a povolania pre testovanie webovej aplikácie. Každý oslovený používateľ dostal sadu inštrukcií, ktoré má vykonať a každý užívateľ mi na konci uviedol či požiadavky splnil, ako náročné boli, aká bola intuitívnosť, či nastali nejaké chyby a aká bola celková spokojnosť. Inštrukcie boli nasledovné:

- Registrácia
- V mape nechať zobrazené iba rezervácie
- Nájsť Horšiansku dolinu
- Priblížiť / oddialiť mapu tak, aby boli viditeľné Levice
- Pokryť Levice vlastným štvorcom
- Štvorec uložiť a ľubovoľne pomenovať
- Požiadať užívateľa „admin“ o priateľstvo
- Odhlásiť sa
- Prihlásiť sa ako užívateľ „admin“ (prihlasovacie údaje boli poskytnuté)
- Prijať žiadosť o priateľstvo
- Prejsť do administrácie
- Zmeniť viditeľnosť všetkých skupín (okrem „Rezervácie“) tak, aby ich verejnosť, ale ani prihlásený užívatelia nemohli vidieť

- Na začiatku vytvorenému užívateľovi udeliť možnosť úpravy a zobrazenia pre podskupinu „Národná prírodná rezervácia“
- Odhlásiť sa a prihlásiť sa do pôvodného účtu
- S užívateľom „admin“ zdieľať na začiatku vytvorený štvorec
- Prejsť do administrácie
- Nájsť NPR Choč a prejsť do režimu úpravy a zmeniť tvar, farbu, orámovanie objektu
- Odísť z režimu (uložiť)
- Odhlásiť sa a prihlásiť ako admin
- Overiť, či je predtým vytvorený štvorec viditeľný
- Vypnúť zobrazenie 5. stupňa ochrany a sledovať rozdiel
- Vytvoriť ľubovoľný vlastný objekt, následne ho zmazať a potom obnoviť z koša
- Vytvoriť novú skupinu (užívateľskú) tak, aby bola verejne dostupná a pre každého
- Vytvoriť objekt v rámci skupiny
- Odhlásiť sa a prihlásiť sa do pôvodného účtu
- Pridať sa do vyššie vytvorenej užívateľskej skupiny
- Pridať do skupiny vlastný príspevok

Výsledky testu použiteľnosti dopadli pozitívne. Všetkým zúčastneným sa podarilo bez problémov vykonať všetky uvedené inštrukcie. Každý zúčastnený aplikáciu hodnotil pozitívne a v rámci možností aj intuitívne. Z testu použiteľnosti vyplynuli aj určité drobné nedokonalosti:

1. dvom respondentom by prišlo pohodlnejšie spúšťať akcie pomocou pravého tlačidla, nie ľavého.
2. jeden respondent uviedol, že nastavenia mapy sú osamotené a nie každý si ich môže okamžite všimnúť.
3. jeden respondent uviedol, že by bolo lepšie, ak by všetky ovládacie prvky mapy boli na jednej strane.

Každý bod som zobral do úvahy. Prvý bod nemôžem splniť, pretože sa jedná o webovú aplikáciu a pravé tlačidlo má vyhradený prehliadač. Je síce technicky možné, aby kontextové menu prehliadača bolo ignorované, nie je ale isté či takéto riešenie by fungovalo vo všetkých prehliadačoch a či by sa časom nestalo nefunkčným. Druhý a tretí bod mi prídu vysoko subjektívne a mne osobne usporiadanie tlačidiel príde vhodné.

5 Záver

Cieľom bakalárskej práce bolo navrhnúť a vytvoriť interaktívnu mapu s chránenými oblasťami Slovenskej republiky, pričom užívatelia majú možnosť vytvárať vlastné objekty, ktoré môžu následne zdieľať s ostatnými užívateľmi, organizovať sa do skupín a podobne čo aplikáciu neidentifikuje len ako informačný zdroj, ale aj ako platformu pre komunitnú spoluprácu a diskusiu.

V rámci úvodu a prvej kapitoly som predstavil dôvody, ktoré ma viedli k vytvoreniu webovej aplikácie, ktorej detailný návrh architektúry a funkcií je uvedený v druhej kapitole bol vytvorený aj na základe analýzy konkurencie.

V tretej kapitole som popísal technickú realizáciu implementácie webovej aplikácie ako aj rôzne aspekty praktického vývoja softvéru, ako napríklad bezpečnosť, vývojové prostredie, štruktúra projektu alebo nasadenie na produkčný server.

Štvrtá kapitola je venovaná testovaniu aplikácie s reálnymi užívateľmi a analýze ich interakcie s aplikáciou. V tejto časti sa nachádza jednotná séria pokynov, ktoré má užívateľ splniť a následne svoju celkovú skúsenosť zhodnotiť v rámci spätnej väzby. Celkový výsledok testovania bol pozitívny a jeho výsledok indikuje spokojnosť u 100 percent testovacej vzorky užívateľov.

Výstupom bakalárskej práce je webová aplikácia, ktorá poskytuje podrobné informácie o jednotlivých chránených oblastiach vrátane ich stupňov ochrany splňujúca všetky navrhnuté nároky ohľadom užívateľskej interakcie, pričom aplikáciu je možné bez najmenších problémov obohatiť o dodatočné funkcionality.

Zoznam použitej literatúry

1. **Štátna ochrana prírody SR.** Užívateľská príručka pre prácu s verejným mapovým portálom KIMS. [Online] [Dátum: 2. 2 2024.] https://www.biomonitoring.sk/CMS/FileDownload/Detail/99.

2. **MongoDB, Inc.** Full Stack Development Explained. [Online] [Dátum: 10. 03 2024.] https://www.mongodb.com/languages/full-stack-development.

3. **European Central Bank.** The digital economy and the euro area. [Online] [Dátum: 10. 03 2024.] https://www.ecb.europa.eu/pub/economic-bulletin/articles/2021/html/ecb.ebart202008_03~da0f5f792a.en.html.

4. **Python Software Foundation.** Python documentation. [Online] [Dátum: 10. 03 2024.] https://docs.python.org/3/.

5. **TIOBE Software BV.** TIOBE Index for March 2024. [Online] [Dátum: 10. 03 2024.] https://www.tiobe.com/tiobe-index/.

6. **JavaTpoint.** Django MVT. [Online] [Dátum: 10. 03 2024.] https://www.javatpoint.com/django-mvt.

7. **Matplotlib Project.** Matplotlib: Visualization with Python. [Online] [Dátum: 22. 03 2024.] https://matplotlib.org/.

8. **Pallets Project.** Jinja Documentation. [Online] [Dátum: 22. 03 2024.] https://jinja.palletsprojects.com/en/3.1.x/.

9. **Meta Platforms, Inc.** React - Getting Started. [Online] [Dátum: 22. 03 2024.] https://legacy.reactjs.org/docs/getting-started.html.

10. **PostgreSQL Community Association.** pgAdmin. [Online] [Dátum: 22. 03 2024.] https://www.pgadmin.org/.

11. **Red Hat, Inc.** Base64 encoding: What sysadmins need to know. [Online] [Dátum: 28. 03 2024.] https://www.redhat.com/sysadmin/base64-encoding.

12. **BOARD OF GOVERNORS of the FEDERAL RESERVE SYSTEM.** Cyberattacks and Financial Stability: Evidence from a Natural Experiment. [Online] [Dátum: 10. 04 2024.] https://www.federalreserve.gov/econres/feds/cyberattacks-and-financial-stability-evidence-from-a-natural-experiment.htm.

13. **Statista Inc.** Annual number of data compromises and individuals impacted in the United States from 2005 to 2023. [Online] [Dátum: 10. 04 2024.] https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed/.