

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-5212-103097

Peter Plevko

Bojové pole

Priebežná správa o riešení projektu BP1

Vedúci práce: Ing. Bindas Bystrik

December 2021

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-5212-103097

Peter Plevko
Bojové pole

Priebežná správa o riešení projektu BP1

Študijný program: Informatika

Študijný odbor: Informatika

Miesto vypracovania: Ústav počítačového inžinierstva a aplikovanej informatiky

Vedúci práce: Ing. Bindas Bystrik

December 2021



ZADANIE BAKALÁRSKEJ PRÁCE

Študent: **Peter Plevko**
ID študenta: 103097
Študijný program: informatika
Študijný odbor: informatika
Vedúci práce: Ing. Bystrík Bindas
Vedúci pracoviska: Ing. Katarína Jelemenská, PhD.

Názov práce: **Bojové pole**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

Veľká vojna – prvá svetová vojna, bola – bola pohromou aj z hľadiska obetí na životoch z radov obyvateľstva. V multinárodnej armáde “Cisára pána” sa však osudy Slovákov stratili. Stratili sa aj pamiatky na Veľkú vojnu na území dnešného Slovenska, ich bitky, cintoríny. Nedávno uplynulo 100 rokov od ukončenia týchto jatiek... A pamätníci už asi nie sú... Ale ešte sú ich synovia (aj keď minimum), vnuci, pravnuci... A oni môžu poskytnúť svedectvo! Cieľom práce je vytvoriť prostredie, ktoré by zaujalo svojim obsahom, vizuálnou stránkou, “karanténou” pre zadávanie informácií. Malo by ukázať priebeh vojenských operácií počas Veľkovojen na území Slovenska, mapovať vojenské cintoríny 1. Svetovej vojny, vedieť zadávať overený príbeh k pochovanému vojakovi, mapovať multimediálne pamiatky. Analyzujte možnosti zadávania, zberu a overovania informácií pre dáta s historickým pôvodom, zamerajte sa na možnosti podporných tvrdení z existujúcich zdrojov dát, dokumentov, overených databáz, čiže analyzujte aj možnosti verifikácie zo známych zdrojov. Vyriešte priradenie bojiska, cintorínu ku konkrétnemu vojakovi, rote, zoskupeniu... Navrhnete riešenie, ktoré bude schopné na základe analytických vstupov prezentovať výsledky zistení, ako portál pre prezentáciu informácií so stupňom overenia. Vyriešte proces schvaľovania obsahu. Navrhnuté riešenie implementujte ako online verziu, ktorá bude mať možnosť pridávania informácií po overení. Samotné riešenie overte pomocou testu.

Rozsah práce: 40

Termín odovzdania bakalárskej práce: 16. 05. 2022
Dátum schválenia zadania bakalárskej práce: 23. 11. 2021
Zadanie bakalárskej práce schválil: doc. Ing. Valentino Vranič, PhD. – garant študijného programu

Čestne vyhlasujem, že som túto prácu vypracoval(a) samostatne, na základe konzultácií a s použitím uvedenej literatúry.

V Bratislave, December 2021

Peter Plevko

Peter Plevko

Pod'akovanie

Moje pod'akovanie patrí vedúcemu mojej práce Ing. Bystríkovi Bindasovi za jeho ochotu a užitočné rady ktoré mi poskytol pri písaní bakalárskej práce.

Anotácia

Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Študijný program: Informatika
Autor: Peter Plevko
Bakalárska práca: Bojové pole
Vedúci bakalárskej práce: Ing. Bindas Bystrík
December 2021

Práca sa zaoberá tvorbou web stránky. Účelom tejto web stránky je vytvoriť prostredie ktoré by zaujalo svojim obsahom, vizuálnou stránkou, pôjde o takzvanú „karanténu“ pre zadávanie informácií. Web stránka by mala ukázať priebeh vojenských operácii počas veľko vojen na území Slovenska, ďalej by mala vedieť mapovať vojenské cintoríny 1. svetovej vojny, zadávať overený príbeh k pochovanému vojakovi, mapovať multimediálne pamiatky a v neposlednom rade pridávať boje ktoré sa odohrali na území Slovenskej Republiky. Teoretická časť – Budú v nej stručne popísane najzákladnejšie jazyky web stránok akými sú HTML, CSS a javascript ale popíšem aj tie menej známe, samozrejme nebude chýbať popísanie tých najpoužívanejších frameworkov a frameworkov ktoré použijem. Bude aj ukázané ktoré jazyky na tvorbu webovej stránky som si vybral a prečo. Nebude chýbať aj porovnanie všetkých technológií ich výhod a samozrejme aj ich nevýhod. Praktická časť - V praktickej časti vytvorím funkčnú web stránku zaoberajúcu sa tematikou prvej svetovej vojny ktorá dovoľuje užívateľom pridávať pramene.

Annotation

Slovak University of Technology Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Degree Course: Informatics

Author: Peter Plevko

Bachelor Thesis: Battle field

Supervisor: Ing. Bindas Bystrík

December 2021

This work is focused on creating a web page. The aim of this website is to create a environment which would be interesting because of its contents, visual side, it will be a safe a place for storing information. It should show the course of war operations during the first world war which happened on Slovak soil, it should also be able to map cemeteries, it should be able to add verified story to the deceased soldier, map multimedial memories and lastly it should be able to add battle that happened on Slovak soil. Teoretical part: This part will contain a little bit information about the most basic website languages HTML, CSS, Javascript and the most used frameworks and of course frameworks that i will use. I will show which languages i choose and why. Another important aspect will be comparison between technologies and listing its pros and cons. Practical part – In this part my job is to create working website about first world war which enables the users to add information about this war.

Obsah

1 Úvod	1
2 Analýza problému	5
2.1 Základný prehľad historických súvislostí Veľkej vojny	5
2.1.1 Prvá svetová vojna	5
2.1.2 Arcivojvoda Franz Ferdinand	5
2.1.3 Kaiser Wilhelm 2	6
2.1.4 Odkaz prvej svetovej vojny	6
2.1.5 Mapa Európy	7
2.1.6 Vojna na území Slovenska	8
2.2 Technológie webovej aplikácie	10
2.2.1 Webová stránka	10
2.2.2 Open source a framework	11
2.2.3 Rozdelenie programovacích jazykov	12
2.2.4 HyperText Markup Language	13
2.2.5 Cascading Style Sheets	13
2.2.6 JavaScript	14
2.2.7 Hypertext Preprocessor	15
2.2.8 Databázy	16
2.2.9 Docker	17
2.3 Zabezpečenie aplikácie	18
2.3.1 Protokol HTTP	18
2.3.2 Šifrovanie dát	20

2.3.3	Autentifikácia a Autorizácia	20
2.3.4	Typy používateľov	21
2.4	Útoky na webové aplikácie	23
2.4.1	SQL injection	23
2.5	Architektúra aplikácií	26
2.5.1	Trojvrstvová aplikácia	26
2.6	Search engine optimization	29
3	Návrh riešenia	33
3.1	Požiadavky webovej prezentácie	33
3.2	Špecifikácia webovej aplikácie	34
3.3	Cieľová skupina	34
3.4	Technológie	35
3.5	Povolenia	38
3.6	Používateľské rozhranie	38
3.6.1	Hlavná stránka	39
3.6.2	Zmena mapy - pod stránka	40
3.6.3	Slovensko - pod stránka	41
3.6.4	Pridanie zdroja - pod stránka	42
3.6.5	Schválenie zdroja - pod stránka	43
3.6.6	Armáda - pod stránka	45
3.6.7	Diskusné fórum - pod stránka	46
3.6.8	Konkrétna otázka - pod stránka	47
3.7	Model databázy	49
3.8	Diagramy	50
3.8.1	Pridanie prameňa	50
3.8.2	Prihlásenie používateľa	51

A Harmonogram práce v zimnom semestri	58
A.1 Zhodnotenie	58

Zoznam skratiek

XML	Extensible Markup Language
RDBMS	Relational Database Management System
SQL	Structured Query Language
NoSQL	Non Relational Database
Back end	Data Access Layer
Front end	Presentation Layer
API	Application Programming Interface
PHP	Hypertext Preprocessor
GUI	Graphical User Interface
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
DNS	Domain Name system
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
JS	Javascript
JSON	JavaScript Object Notation
DOM	Document Object Model
URL	Uniform Resource Locator
OOP	Object-Oriented Programming
IP	Internet Protocol
CMS	Content Management System
OS	Operating System
TCP	transmission control protocol
TLS	Transport Layer Security

SSL	Secure Sockets Layer
SMS	Short Message Service
DDOS	Denial Of Service Attack
WAF	Web Application Firewall
AJAX	Asynchronous JavaScript and XML
MVC	Model view controller
SERP	Search Engine Results Pages
CTR	Clickthrough rate
SEO	Search engine optimization
SSL	Secure Sockets Layer
Sass	Syntactically awesome style sheets
JRE	Java Runtime Environment
DQL	Data query language
DDL	Data definition language
DCL	Data control language
DML	Data manipulation language

1 Úvod

Na počiatku moderných dejín stojí jeden z najkrvavejších konfliktov v dejinách – Prvá svetová vojna. Odštartovala tým jedno z najvražednejších období, ktorého následky si nesieme v pamätiach až dodnes. Štyri roky trvajúci konflikt znamenal vznik nových štátov, ale taktiež viedol k nebývalej skaze nie len na bojisku, ale aj v životoch jednotlivcov a spoločnosti. Slovenskí vojaci boli z ťažkých podmienok v ktorých žili pripravení na boj, a preto ich vojenské velenie posielalo do najťažších frontových úsekov.

Územie dnešného Slovenska bolo poznačené aj mnohými bojmi. Vojnový konflikt v Karpatoch zasiahol územie okresov Bardejov, Svidník, Stropkov, Medzilaborce, Humenné a okres Snina. Toto pohraničné pásmo historického Uhorska bolo miestom stretu cárskej armády a rakúsko-uhorskej cisárskej armády. Mesto Svidník si v roku 1911 zvolil František Ferdinand d'Este za sídlo generálneho štábu.

Ako môžeme teda vidieť Slovensko zohrávalo neopomenuteľnú rolu v tejto Veľkej vojne, a preto je dôležité verejnosti priblížiť deje a udalosti, ktoré sa mohli odohrávať v blízkosti ich rodných miest alebo do nich mohli byť zapojení aj ich najbližší príbuzní.

Stránky o prvej svetovej vojne na Slovensku môžeme samozrejme nájsť. Bohužiaľ, častokrát neprechádzajú revíziami a sú v nevyhovujúcom stave, častokrát nefunkčné. Tieto stránky sú graficky zastaralé, neočaria nás svojou vizuálnou stránkou ale ani užívateľským prostredím. V minulosti však nebol kladený taký dôraz na formu ako to je dnes.

Otázka, prečo sa zaoberať tvorbou webových stránok o prvej svetovej vojne, je určite na mieste. Dôvodov je hneď niekoľko, ale ten najzásadnejší je dostupnosť širšej verejnosti ale hlavne mladším ročníkom. Samozrejme kniha ako médium má svojich zástancov, ale webová stránka s pestrými vizualizáciami dokáže pritiahnuť pozornosť ľudí, ktorí v histórii nevidia svoj záujem. Ďalšou výhodou je taktiež dostupnosť aj na cestách, napríklad v mobilnom telefóne. Pri dlhších cestách si rozmyslíme, či so sebou brať záťaž vo forme knihy, alebo ďalšieho zariadenia ako e-book.

V dnešnej dobe sa stáva publikovanie informácií na webových stránkach stále žiadanejšie a to hlavne tam kde vzniká potreba častých aktualizácií a zásahov do textu. To dáva vznik CMS systémom. Ktoré umožňujú užívateľom webových stránok aby si webový obsah spracovali sami. Táto vlastnosť úpravy informácií je veľmi dôležitá v dejepise a bližšie v prvej svetovej vojne.

Samozrejme v dnešnej dobe máme mnoho potvrdených faktov a zásadných informácií o tom, ako ktorý boj prebiehal, ale častokrát sú aj objavené nové fakty alebo dôležité miesta, ktoré je vhodné pridať alebo poopraviť. V takejto chvíli jediné čo môžeme s knihou spraviť je upraviť jej obsah a znova ju vydať. Avšak v našej web stránke by táto situácia bola oveľa ľahšia. Pridanie textu, zmenenie kódu, znovu načítanie stránky a máme hotovo. 1

2 Analýza problému

2.1 Základný prehľad historických súvislostí Veľkej vojny

2.1.1 Prvá svetová vojna

Prvá svetová vojna tiež známa ako veľká vojna začala v roku 1914 po atentáte na arcivojvodu Franza Ferdinanda z rakúska. Jeho smrť spustila vojnu v Európe ktorá trvala do roku 1918. Počas tohto konfliktu krajiny akými sú Nemecko, Rakúsko-Uhorsko, Bulharsko tiež známe ako Ústredne veľmoci. Bojovali proti Veľkej Británii, Francúzku, Rusku, Taliansku, Rumunsku, Japonsku a Spojeným Štátom Americkým, tiež známym ako štáty Dohody. Vďaka novým vojenským technológiám a hororom zákopovej vojny. Prvá svetová vojna videla skazu a deštrukciu doposiaľ nevydaných rozmerov. Kým skončila vojna a štáty dohody zvíťazili zomrelo viac ako 16 miliónov ľudí. Vojna nehľadela nato či ste vojak alebo civilista, zomierali aj vojaci aj civilisti a to hlavne muži.

2.1.2 Arcivojvoda Franz Ferdinand

Napätie narastalo v celej Európe avšak najviac v regióne Balkánu v juhovýchodnej Európe ešte predtým než začala prvá svetová vojna. Existovalo veľmi veľa aliancií Európskych mocností akými sú napríklad Otomanska ríša, Rusko, ale politická nestabilita na Balkáne najmä v Bosne, Srbsku a Hercegovine ohrozovala tieto dohody.

Iskra ktorá odštartovala prvú svetovú vojnu vznikla v Sarajeve v Bosne keď zavraždili Arcivojvodu Franza Ferdinanda. Nástupník Rakúsko-Uhorskej ríše bol zastrelený, spolu so svojou ženou Sophie, srbským nacionalistom Gavrilom Principom 28. júna 1914. Princip a ďalší nacionalisti sa snažili o ukončenie nadvlády Rakúsko-Uhorska nad Bosnou a Hercegovinou. Vražda Franza Ferdinanda naštartovala reťaz udalostí. Rakúsko-Uhorsko ako veľá krajín Európy obviňovalo srbský parlament z útoku a chceli použiť tento incident na finálne vyriešenie Srbského nacionalizmu.

2.1.3 Kaiser Wilhelm 2

Pretože Rusko podporovalo Srbsko, Rakúsko-Uhorsko počkalo s deklaráciou vojny dokiaľ im Nemecko vládca Kaiser Wilhelm 2 nepotvrdil, že Nemecko sa pridá na ich stranu. Velitelia Rakúsko-Uhorska sa báli Ruskej intervencie a takisto ich spojencov Francúzska a veľkej Británie. Dňa 5. júla ich Kaiser Wilhelm uistil, že v prípade vojny ich Nemecko podporí. Následne duálna monarchia Rakúsko-Uhorska poslala Srbsku ultimátum s tak tvrdými podmienkami, že bolo skoro nemožné ich splniť.

2.1.4 Odkaz prvej svetovej vojny

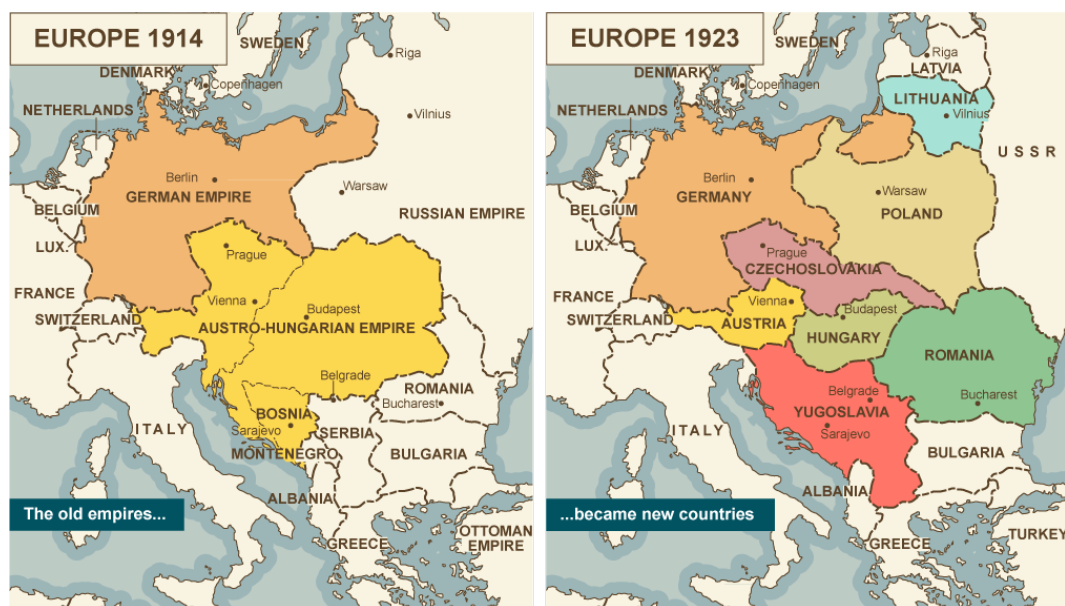
Keďže v prvej svetovej vojne zomierali hlavne muži tak do robôt museli nastúpiť vo veľkom množstve ženy. Prvá svetová vojna taktiež pomohla rozšíriť jednu z naj-smrteľnejších pandémie tiež známu ako Španielska chrípka ktorá zabila okolo 35 miliónov ľudí.

Prvá svetová vojna sa tiež označuje ako prvá moderná vojna. Veľa technológií asociovaných s vojenským konfliktom ako guľomety, tanky, vzdušný boj a rádiová

komunikácia boli použité vo veľkej miere počas prvej svetovej vojny. Efekty ktoré spôsobili chemické zbrane ako horčičný plyn a fosgén ľuďom počas prvej svetovej vojny spôsobili nechť verejnosti k ich ďalšiemu používaniu. Ženevská konvencia podpísaná v roku 1925 zakázala používanie chemických zbraní v boji a platí dodnes. [2]

2.1.5 Mapa Európy

Ríše padajú a mapy sa menia. Na obrázku číslo 1 môžeme vidieť ako sa zmenila mapa Európy zásluhou prvej svetovej vojny. Najdôležitejšou zmenou pre nás ako Slovákov bolo samozrejme vytvorenie samostatného Česko-Slovenska. Na mape môžeme taktiež vidieť zánik Rakúsko-Uhorska, Ruskej ríše, Nemeckej ríše, avšak aj vznik Juhoslávie. Podobne môžeme vidieť obrovskú zmenu mapy Európy za veľmi krátky čas.



Obr. 1: Mapa Európy zachytávajúca vznik nových štátov [3]

2.1.6 Vojna na území Slovenska

Slovenskí politici vyhlásili po vypuknutí vojny politiku pasivity, vojaci však museli narukovať a bojovať za cisára. Najväčšie počty obetí boli zaznamenané na východnom fronte v Haliči a na južnom fronte v Taliansku na Piave kde zahynulo takmer 70 000 Slovákov.

Územiu Slovenska sa front 1. svetovej vojny z väčšej časti vyhol. Jedine východ nášho územia obsadili ruské vojská. Medzi obsadené dediny patrili napríklad Bardejov, Zborov, Svidník. Slovenskí vojaci boli skromní a odvážni preto ich vojenské velenie posielalo do najťažších bojov. Často zažívali zimu, hlad a samozrejme aj šikanovanie zo strany dôstojníkov.

Veľmi veľa obyvateľov s vojnou nesúhlasilo. Nanešťastie ľudia boli za svoje protestné výroky trestaní, neraz sa dostali aj do väzenia. Každý protivojnový protest sa prísne karhal. Tak ako vo väčšine Európy keďže všetci muži boli na vojne ženy

pracovali nielen na poliach ale aj v továrňach. Počas vojny brali notári úplatky za sľub, že mužov dostanú na nejakú tu chvíľu domov a preč z frontu. Kvôli tomuto vznikla vrstva vojnových zbohatlíkov. Vojna spôsobila úpadok dedín a naopak rozkvet miest. Dôvod tohto trendu je jednoduchý, mestá mali vytvorené podmienky pre priemyselnú výrobu a bohatli z obchodu pre armádu.

Medzičasom vznikali v zahraničí prvé plány na osamostatnenie Slovákov spod Rakúsko-Uhorska. Dominantnou sa stala myšlienka spojenia slovenského národa s českým. 30. Mája 1918 bola podpísaná Pittsburská dohoda - dohoda slovenských a českých organizácií o autonómnom postavení Slovenska v novovzniknutom štáte. Náš zahraničný odboj mal centrum vo Francúzku. Hlavnými predstaviteľmi tohto odboja boli T. G. Masaryk a M.R. Štefánik. [4]

2.2 Technológie webovej aplikácie

Webová aplikácia sa nemusí inštalovať na zariadenie užívateľa. Je dostupná na ktoromkoľvek zariadení pomocou webového prehliadača. Aplikácia používa klient-server štruktúru, to znamená že je spustená na strane serveru oproti bežným aplikáciám ktoré bežia na OS. Webový prehliadač sa stará o zobrazenie webovej aplikácie. Toto je oveľa lepšie ako aplikácie ktoré bežia na OS ktoré pre svoju funkčnosť potrebujú vlastný program. Tento program musí byť nainštalovaný na každom zariadení na ktorom chce užívateľ túto aplikáciu používať.

Nevýhodou webových aplikácií avšak môže byť to, že je potrebné byť pripojený na internet pre prácu na nich. Webové aplikácie tvoria viacero webových stránok vo formáte HTML. Ktoré sú podporované bežnými prehliadačmi, najpopulárnejším je Google Chrome. Komunikácia prebieha pomocou protokolu HTTP.

Tento protokol je určený na výmenu HTML dokumentov. Dôležité je pred samotným vývojom aplikácie si túto aplikáciu rozvrhnúť. Urobiť si nejaké mockupy stránky, ako by asi mala vyzerať po vizuálnej stránke. Zamyslieť sa nad tým aké technológie budeme používať nad logikou aplikácie a počas celého vývoja tejto aplikácie tieto veci dodržiavať a nemeniť.

2.2.1 Webová stránka

Webová stránka je skupina celosvetovo dostupných pospájaných stránok pod jedným doménovým menom. Môže byť vyvíjaná jedným človekom alebo organizáciou. Webová stránka je hostovaná na jednom alebo viacerých webových serveroch. Je

dostupná pomocou internetu. Pripojíme sa na ňu pomocou IP adresy. Máme dva typy stránok a tými sú následovne.

1. Statická stránka obsahuje fixný počet stránok. Tento druh stránok je vytvorený z HTML a CSS pomocou jednoduchého textového editora ako je napríklad Notepad. Napríklad stránka firmy, nejakej inštitúcie a podobne. Statické stránky zobrazia tie isté informácie nezávisiac na tom kto ich navštívi. Statické stránky nemusia byť len jednoduchý text. Môžu obsahovať aj pekný dizajn alebo videá. Ale ako som spomínal už vyššie každý užívateľ dostane to isté video vždy keď sa pripojí na stránku. Jediným spôsobom ako zmeniť to čo uvidí užívateľ je zmenením zdrojového kódu. Tento typ stránky používame najčastejšie pre zobrazenie stránky o firme.
2. Dynamická stránka vie všetko to čo aj statická stránka avšak hlavným rozdielom je to, že po prihlásení sa na stránku každý používateľ vidí iné informácie. A na zmenu toho čo užívateľ vidí nie je potrebné meniť zdrojový kód. Na rozdiel od statickej stránky sa táto stránka používa keď potrebujeme z používateľom nejako komunikovať získavať od neho dáta a zase mu nejaké dáta posilať a ukazovať a tieto dáta sa s každým používateľom menia.



2.2.2 Open source a framework

Open source je to typ softvéru, ktorý ľudia môžu upravovať a zdieľať, pretože daný softvér je verejne dostupný. Open source je vyvíjaný decentralizovane a spoločnou cestou. Spolieha sa na komunitnú tvorbu. Veľmi často je to lacnejšie flexibilnejšie pretože tento software je vyvíjaný komunitou ľudí a nie jedným autorom alebo firmou. Výborným príkladom open source je Linux. Je to najväčší open source

projekt na svete. [6]

Framework, ako programátor nepotrebujeme vždy začínať od začiatku keď máme nástroje vytvorené na pomoc s projektami. Frameworky sú softvéry, ktoré sú vyvíjané a používané vývojármi na tvorbu aplikácií. Sú stavané, testované a optimalizované viacerými skúsenými programátormi, vďaka čomu sú efektívne a vo väčšej miere bezchybné. Vďaka frameworkom sa môžeme sústrediť na funkcionálnu na vysokom levely. Toto môžeme urobiť pretože o funkcionálnu na nízkom levely sa stará framework samotný. Väčšina dnešných frameworkov stavá na MVC architektúre. [7]

2.2.3 Rozdelenie programovacích jazykov

Kompilované jazyky: Sú to programovacie jazyky ktoré sú kompilované a nie interpretované. Program napísané v takomto jazyku keď je raz skompilovaný je prepísaný do inštrukcií nášho stroja do takzvaného machine codu ktorý je pre nás ľudí nečitateľný.

Tieto jazyky sú veľmi rýchle ale pri každej zmene je nutné program rekompilovať čo je veľmi prácne a pomalé. Medzi kompilované jazyky patria napríklad Go, C++, C#, Java, COBOL a tak ďalej. Interpretované jazyky: Sú to programovacie jazyky ktoré sú interpretované bez toho aby sa program skompiloval do inštrukcií stroja. Sú to programovacie jazyky kde inštrukcie nie sú priamo vykonané našim strojom, ale sú čítané a vykonávané nejakým iným programom.

Tieto jazyky sú pomalšie ich úzkym hrdlom je komunikácia s databázou. Pri zmene zdrojového kódu aplikácie stačí iba obnoviť stránku v prehliadači. O všetku kompiláciu sa stará webový server. Medzi tieto jazyky patria JavaScript, Perl, Python, BASIC a tak ďalej. [8]

2.2.4 HyperText Markup Language

Je to základný stavebný block webu. Definuje význam a štruktúru webového obsahu. Skladá sa z párových a nepárových značiek. Hypertext slúži na spojenie webových stránok medzi sebou, buď v jednej kostre alebo medzi web stránkami. Linky sú základom webu. Pridávaním obsahu na internet a linkovaním stránok iných ľudí sa stávame aktívnym účastníkom World Wide Webu.

HTML element sa odlišuje od textu v dokumente pomocou značiek ktoré sa skladajú z názvu elementu a sú obklopené `<` a `>`. Názov elementu v ohraničení je nezávislý na veľkosti písma to znamená že `DIV` je to isté ako `div`.

Deklarácia dokumentu v HTML5 vyzerá nasledovne. Je potrebné mať značku `<!DOCTYPE html>` Koreňovým elementom je značka `<html></html>` ktorá obsahuje celý dokument. Do hlavičky `<head></head>` sa umiestňuje názov stránky, odkaz na CSS štýly ktoré budeme používať a iné. [\[9\]](#)

2.2.5 Cascading Style Sheets

Je to jazyk, ktorý nám dovoľuje špecifikovať ako sú dokumenty prezentované užívateľom, ako sú štylované, dizajnované a podobne. To znamená že môžeme napríklad zmeniť farbu a veľkosť textu. Môžeme tvoriť tabuľky, navigačné menu a ak sme veľmi zdatný dokonca aj animácie. Tento jazyk bol vytvorený v roku 1996 ako súčasť značkovacieho jazyka HTML4. Dajú sa však s ním upravovať aj stránky ktoré nevznikli za použitia HTML ale za použitia iných značkovacích jazykov ako napríklad XML alebo XHTML. Grafickú úpravu docielime zadefinovaním vlastností určitých elementov, ktoré vložíme do súboru s koncovkou `.css` a v `html` súbore ho prepojíme.

CSS preprocessor je skriptovací jazyk ktorý rozširuje CSS a dovoľuje vývojárom

písať kód v jednom jazyku a skompilovať ho do CSS. Najpopulárnejším takýmto preprocessorom je Sass. Sass rozširuje CSS o premenne, vnhiezené pravidla, a mnoho ďalšieho. [10]

2.2.6 JavaScript

JavaScript je skriptovací programovací jazyk, ktorý nám dovoľuje implementovať komplexné vlastnosti na webových stránkach vždy, keď pre nás web stránka robí niečo iné než zobrazuje statické informácie. Napríklad zobrazuje interaktívne mapy, po prihlásení sa nám zobrazia naše údaje môžeme si byť istý že bol použitý JS. Patrí medzi tri najhlavnejšie štandardné webové technológie ďalšími dvomi sú HTML a CSS opísané vyššie.

Tento interpretovaný programovací jazyk nemá jednotnú podobu. Najčastejšie sa používa iba na skripty. Ale existuje veľmi veľa jeho modifikácií a programov ktoré používajú jeho syntax a jadro. Takýmito modifikáciami sú napríklad Node.js, ktorý slúži na operácie na strane servera alebo knižnice React a framework Angular, ktoré bežia na strane klienta. [11]

1. Typescript je nadstavba jazyka JS pridáva do JS možnosť špecifikovať pri každej premennej jej typ. Je open source vyvíjaný Microsoftom a vznikol v roku 2012. Je taký užitočný že ho firma Google integrovala do svojho frameworku Angular. Nevýhodou Typescriptu je avšak to že potrebuje transpiler.
2. jQuery bol vytvorený v roku 2006. Je to knižnica jazyka JS ktorá sa zaoberá interakciou medzi JS a HTML. Zjednodušuje manipuláciu s DOM, obsluhu udalostí a animácií. Dôvodom vzniku bola potreba zjednodušenia používania JS na stránkach. Jedná sa o open source software. Používa ho 73% najpopulárnejších web stránok. Z tohto dôvodu to je aj momentálne

najpoužívanejšia javascript knižnica na svete. Medzi firmy ktoré používajú JQuery sú napríklad Google, Microsoft, IBM. [12]

3. Node.js je prostredie, ktoré nám dovoľuje spúšťať JavaScript kód mimo webového prehliadača. Je postavený na Chrome V8 JS engine. Node nám slúži na tvorbu serverovej časti webových aplikácií podobne ako jazyk PHP. Vďaka tomuto prostrediu môžeme používať JS na strane klienta aj na strane servera čo je veľkou výhodou. [13]
4. Nest.js, je to node.js framework pre vývoj škálovateľných aplikácií na strane servera pomocou jazyka Typescript. Štruktúra kódu je veľmi podobná Angularu. Ma vlastný command line. Jedná sa o ideálny backend framework pre vývojárov, ktorý už pracovali s Angularom. [14]
5. Angular, je to open-source framework napísaný v TypeScript-e. Vyvíjaný je firmou Google. Bol vydaný v roku 2016. Obsahuje knižnice, ktoré pokrývajú veľmi veľkú škálu funkcií. Používa sa ako frontend u MEAN stacku. [15]
6. JS Minifier robí stránky menšie a rýchlejšie na načítanie tým, že zmenší JS a CSS kód. Minifier odstráni medzery, komentáre, skombinuje viacero súborov do jedného a nakoniec optimalizuje a skráti premenné. [16]

2.2.7 Hypertext Preprocessor

Je to open-source skriptovací jazyk určený predovšetkým na tvorbu webových aplikácií či už to statických alebo dynamických. V dnešnej dobe je to najčastejšie používaný skriptovací jazyk. Vďaka zato tomu že je nezávislý na platforme a vďaka vysokej miere kompatibility s rôznymi databázovými systémami. PHP kód je možné priamo vkladať do HTML. Kód tohto programovacieho jazyka sa môže spracovávať iba na webovom servere, a to pomocou PHP interpretera. Klientovi

stačí webový prehliadač. [17]

2.2.8 Databázy

Databáza je organizovaná kolekcia štruktúrovaných informácií alebo štruktúrovaných dát. Väčšinou je uchovávaná elektronicky v počítačových systémoch. Databáza býva kontrolovaná DBMS. Data a DBMS spolu s aplikáciami, ktoré sú s nimi spojené nazývame Databázový systém alebo skrátene databáza. Väčšina databáz používa SQL pre zapisovanie a vyhľadávanie dát. SQL je programovací jazyk určený na prácu s databázami. SQL sa skladá zo štyroch pod jazykov a týmito jazykmi sú.

1. DQL: slúži na extrahovanie dát z databázy
2. DDL: slúži na vytváranie tabuliek a definovanie typov atribútov v tabuľke
3. DCL: slúži na nastavenie prístupu a autorizácii
4. DML: slúži na modifikáciu dát uložených v databáze

Máme dva typy databáz relačné a nerelačné.

1. Relačné

- (a) MySQL, je to momentálne najpoužívanější a najrozšírejší databáza.

Dôvodov enormnej popularity tejto databázy je viacero. Medzi hlavne patrí to, že sa jedna o open-source, je relačná a veľmi ľahko sa používa. Jedna sa o jednu z najrýchlejších databáz. Na správu obsahu databázy sa používa nástroj Adminer. [18]

- (b) PostgreSQL je to silná open-source relačná databáza. Táto databáza si získala silnú reputáciu pre svoju architektúru, spoľahlivosť a pre svoje

venovanie sa open source komunite. Táto databáza beží na všetkých operačných systémoch. [19]

2. Nerelačné

- (a) MongoDB, Je to systém pre správu databáz ktorý používa na prácu s dátami štruktúrovaný dotazovací jazyk SQL. Jedna sa o najpokročilejšiu dokumentovo orientovanú databázu. Bola vytvorená v roku 2007. Dáta sú uložené v JSON dokumentoch. Vývojári túto databázu používajú veľmi radi a to z viacerých dôvodov. Zabezpečuje nám vysokú spoľahlivosť, znovu použiteľnosť a škálovateľnosť. [20]

2.2.9 Docker

Je to open source kontainerizačná platforma. Dovoľuje vývojárom zabaliť aplikácie do kontajnerov. Kontajner je štandardizovaný komponent ktorý kombinuje source kód aplikácie s knižnicami operačného systému a s potrebnými závislosťami aby sme to mohli spustiť v každom prostredí. So stále zväčšujúcou sa popularitou cloud systémov sa stáva stále populárnejším aj Docker. [21]

2.3 Zabezpečenie aplikácie

2.3.1 Protokol HTTP

Je protokol pre získavanie dát akými sú napríklad HTML dokumenty. Je to základ každej výmeny informácií na internete a celosvetovej sieti World Wide Web. Jedná sa o klient-server protokol. Požiadavky protokolu sú poslane serveru prijímateľom to znamená webovým prehliadačom do ktorého používateľ zadal URL. Táto URL sa vyhodnotí pomocou DNS záznamu. Následne sa naviaže spojenie cez TCP protokol na server. Potom je dokument odoslaný a prijatý prijímateľom. Klient a server komunikujú pomocou výmeny správ.

Správa poslaná webovým prehliadačom sa nazýva požiadavka a správa poslaná serverom sa nazýva odpoveď. Tento protokol bol nadizajnovaný v roku 1990. Od svojho vzniku prešiel mnohými zmenami, najdôležitejšou zmenou bolo zavedenie šifrovanej verzie HTTP, takzvaný HTTPS. Tento je oveľa bezpečnejší. Je to protokol aplikačnej vrstvy ktorý sa posiela cez TCP. Vďaka svojej všestrannosti sa používa nielen na posielanie HTML dokumentov ale aj obrázkov a videí.

HTTPS používa šifrovací protokol na zašifrovanie komunikácie. Ak používame tento protokol vieme, že dáta poslané z webového serveru neboli zachytené alebo pozmenené treťou stranou. Je dôležité používať tento protokol vždy, keď potrebujeme ochranu a súkromie. Príkladom vhodného použitia je internet banking. Tento protokol sa nazýva TLS, avšak môžeme ho poznať aj pod názvom SSL.

Protokol použije na zašifrovanie dát infraštruktúru verejného asymetrického kľúča. Takáto komunikácia používa dva typy kľúčov na komunikáciu. Prvým kľúčom je

kľúč, ktorý vlastní majiteľ webovej stránky. K tomuto kľúču nemá nikto prístup len webový server, ktorý tento kľúč používa na odšifrovanie informácií, ktoré boli zašifrované verejným kľúčom. Druhým kľúčom je takzvaný verejný kľúč. Tento kľúč je dostupný všetkým, ktorý chcú nejakým spôsobom komunikovať so serverom a tak ako správy zašifrované verejným kľúčom môžu byť ošifrované len súkromným kľúčom.

Dotazovacie metódy. Protokol HTTP definuje niekoľko metód ktoré môžeme vykonávať nad objektom. Návštevník zadá dáta prostredníctvom formulárov ktoré sú následne uvedenými metódami spracované na strane serveru týmito metódami sú.

1. Get: Slúži na zaslanie požiadavky na server a nemala by robiť nič iné ako získať data od servera. Get metóda je základom všetkých webových stránok.
2. Head: Rovnako ako get ale bez tela odpovede servera.
3. Post: Používa sa na posielanie dát, serveru ktoré zväčša spôsobujú zmeny na serveri. Najčastejšie sa používa na vytvorenie nových dát na serveri.
4. Put: Používa sa na pozmenenie už existujúcich dát na servery.
5. Patch: Taktiež sa používa na pozmenenie dát ale patch sa používa ak chceme zmeniť len časť nejakého zdroja napríklad iba názov. Ak chceme zmeniť všetky atribúty použijeme put.
6. Delete: Slúži na zmazanie dát z databázy.
7. Connect: Mení request na TCP/IP tunnel zvyčajne sa používa na založenie bezpečného SSL šifrovaného tunelu na server.
8. Options: Vráti HTTP metódy aké server podporuje na zadanej URL. Používa sa na skontrolovanie funkčnosti servera.

9. Trace: Používa sa na vrátenie správy klientovi aby sa zistilo aké a či nastali nejaké zmeny keď requesty vybavovali servery medzi klientom a koncovým serverom.

[22, 23]

2.3.2 Šifrovanie dát

Šifrovanie je premena dát z čitateľnej formy na zakódovaný formát. Takéto zakódované dáta sami o sebe nemajú žiadnu výpovednú hodnotu. Informácie z nich vieme zistiť až po ich dekodovaní. Medzi najstaršie typy šifier patri napríklad vigenérova šifra alebo cézarova šifra. Šifrovanie dát je základným blokom počítačovej bezpečnosti. Je to najjednoduchšia a najdôležitejšia časť ochrany systému.

Šifrovanie dát sa používa či už to firmami alebo jedincami na ochranu komunikácie medzi prehliadačom a serverom. Funguje to nasledovne keď informáciu zdieľame na internete táto informácia putuje po veľmi veľa zariadeniach na celom svete. keď táto informácia cestuje po internete je šanca že môže byť ukradnutá hackermi. Na ochranu našej informácie sa používa šifrovanie.

Funguje to takto, šifrovanie používa kryptografický kľúč a matematické hodnoty na ktorých sa odosielateľ a prijímateľ dohodli. Čím zložitejší je kryptografický kľúč tým je komunikácia bezpečnejšia. [24]

2.3.3 Autentifikácia a Autorizácia

Autentifikácia je neoddeliteľná súčasť prihlásenia užívateľa do webovej aplikácie. Znamená to overenie, či užívateľ ktorý sa prihlasuje je ten za koho sa vydáva. Sú tri typy autentizácie

1. Jedno-faktorová autentizácia: Autentizácia pomocou znalostí: meno a heslo, túto autentizáciu budem používať ja. Pomocou predmetu: karta, Pomocou používateľa: sietnica,
2. Dvoj-faktorová autentizácia: sú potrebné dve od seba nezávislé veci najčastejším prípadom je meno, heslo a kód z SMS.
3. Troj-faktorová autentizácia: Jedna sa o tri od seba nezávislé veci. Meno heslo, SMS kód, sken sietnice.

Autorizácia je proces udelenia práv užívateľovi na prístup k informáciám. V aplikáciách sa často stretávame s tým, že každý typ používateľa má prístup k iným funkciám. V mojom návrhu budem implementovať tento model. 25

2.3.4 Typy používateľov

Aplikácia bude poznať 5 typov používateľov. Najčastejším používateľom bude takzvaný bežný používateľ a najmenej bežným developer. Užívateľov budem deliť na registrovaných a neregistrovaných. Iba registrovaný užívateľ bude môcť na stránku pridávať obsah. Je to z toho dôvodu aby som mohol kontrolovať kto čo pridáva a ak jeden používateľ viacej krát zadá nejakú hlúposť môžem mu zakázať navštevovať stránku. Nevýhodou registrácie je možné odradenie niektorých užívateľov od pridávania informácií.

1. Návštevník: Používateľ s najmenšími právomocami. Právomocou tohto užívateľa je len prezeranie si web stránky. Ak chce pridať zdroj musí sa registrovať.
2. Bežný používateľ: Tento typ používateľa bude mať rovnaké právomoci ako návštevník, avšak bude môcť aj zadávať informácie o vojakoch, bojoch, cin-torínoch a všetkých pamiatkach, ktoré sú pre nás zaujímavé. Avšak tieto in-

formácie ktoré zadá sa neobjavia ihneď na našej web stránke pretože najprv potrebujú byť schválené vyššou autoritou - historikom.

3. Historik: Tento typ používateľa bude mať rovnaké právomoci ako bežný používateľ, nemusí byť informaticky zdatný. Informácie ktoré zadajú užívatelia nemusia vždy byť presné, pravdivé alebo k veci. V tejto chvíli nastupuje na scénu historik ktorý pozrie čo užívateľ zadá a zisťuje či sú tieto informácie pravdivé ak áno pridá ich ak nie nepridá ich. Ak náhodou sa nájde užívateľ ktorý veľmi často zadáva falošne informácie má tento historik právo tohto používateľa nahlásiť. Po nahlásení používateľa je to na adminovi či tomuto používateľovi zablokuje vstup.
4. Admin: Tento typ používateľa bude mať rovnaké právomoci ako Historik. Môže robiť všetko okrem menenia zdrojového kódu a to z dôvodu, že by si mohol nastavovať vyššie právomoci a mohol by zosadiť z trónu mňa ako developera a v najhoršom prípade mi ukradnúť stránku a vydávať ju za svoju. Ako admin vidíme ktorých používateľov koľko historikov nahlásilo a máme samozrejme aj právo dať im ban to znamená zakázať im prístup na našu stránku. Stará sa o plynulý chod stránky stará sa o prípadne vzniknuté chyby. Admin má samozrejme aj prístup k databáze, kde môže ručne opravovať prípadne vzniknuté chyby, takže admin by mal byť človek programátorsky zdatný.
5. Developer Tento typ používateľa má najvyššie právomoci a bude len jeden, developer som ja. Mám právo meniť zdrojové kódy.

2.4 Útoky na webové aplikácie

Je veľmi dôležité chrániť našu stránku pred útokmi hackerov či už to tých ktorý sa snažia našu stránku zhodiť alebo vymazať údaje z našej databázy alebo ukradnúť informácie o používateľovi.

2.4.1 SQL injection

Tento útok smeruje na webovú stránku a konkrétnejšie na databázu. Pri vykonaní útočník vloží kus kódu ktorí odhalí skryté dáta a užívateľské vstupy. Keďže hlavným miestom útoku je to čo používateľ zadá. Najlepší spôsob je kontrolovať vstup a sledovať či sa tam nenachádzajú prvky útoku. Ako ochrana slúži kontrola vstupu, parametrizované queriny. Na obrázkoch 2, 3, 4 si takýto SQL injekcion ukážeme.

Na obrázku číslo 2 môžeme vidieť ako ma vyzeráť správne zadaný vstup.

Query:

```
SELECT * FROM Users
```

```
WHERE Name = 'meno' AND Pass = 'heslo'
```



A user authentication form with two input fields. The first field is labeled 'meno' (name) and contains the text 'meno'. The second field is labeled 'heslo' (password) and contains the text 'heslo'. To the right of the password field is an eye icon. Below the fields is a blue button with the text 'Prihlásiť sa' (Log in). Below the button is a link that says 'Zabudli ste heslo?' (Forgot your password?).

Obr. 2: autentizácia užívateľa

Injekciu nášho formulára môžeme vidieť na obrázku číslo 3. Toto sql nám vráti všetky riadky z tabuľky používateľov. Pretože or '=' je vždy pravda.

Query:

```
SELECT * FROM Users
```

```
WHERE Name = " or '=' AND Pass = " or '='
```



A user authentication form similar to the one in Figure 2, but with SQL injection payloads. The 'meno' field contains the text '" or ""=''. The 'heslo' field also contains the text '" or ""=''. The rest of the form, including the 'Prihlásiť sa' button and the 'Zabudli ste heslo?' link, is identical to Figure 2.

Obr. 3: Autentizácia užívateľa a injekcia

Tato querina nám vráti všetky riadky z tabuľky users a následne zmaže tabuľku používateľov. Môžeme to vidieť na obrázku 4.

```
SELECT * FROM Users
```

```
WHERE Name = " or '=' AND Pass = " or '=';
```

```
DROP TABLE Users
```



The image shows a login interface with two input fields. The first field contains the text " or \"\"=\". The second field contains the text " or \"\"=\"; DROP TABLE users. Below the fields is a blue button labeled "Prihlásiť sa". Underneath the button is a link that says "Zabudli ste heslo?".

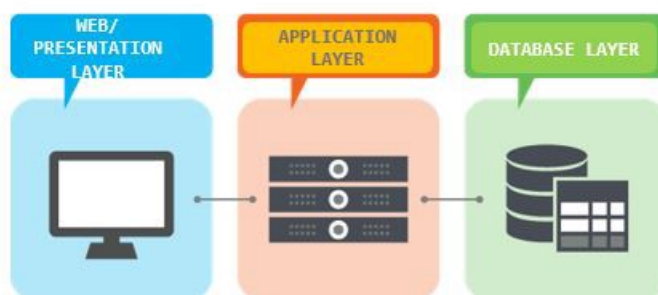
Obr. 4: Autentizacia užívateľa a zmazanie tabuľky

2.5 Architektúra aplikácií

2.5.1 Trojvrstvová aplikácia

Je to jeden z typov architektúry informačných systémov, túto architektúru môžeme vidieť na obrázku číslo 5. Rozdeľuje nám aplikáciu na časti, jednou z týchto častí je to čo vidí a používa užívateľ (prezenčná vrstva) a to čo sa odohráva na pozadí na strane servera (aplikačná a dátová vrstva).

Let us walk through a three tier architecture :



Obr. 5: Trojvrstvová architektúra [27]

1. Prezenčná vrstva, je to vrstva na ktorej sa nachádza užívateľské rozhranie toto je najčastejšie webová stránka napríklad e-shop kde používateľ hádže veci do nákupného košíka, pridá spôsob platby alebo si vytvorí účet. Ďalej sa tu nachádza komunikačná vrstva aplikácie kde používateľ komunikuje s aplikáciou. Jej hlavný cieľ je zobraziť informácie a zbierať informácie od používateľa. Tato najvyššia úroveň vrstvy môže bežať na webovom prehliadači, ako počítačová aplikácia alebo ako GUI. Webová prezentačná vrstva sa väč-

šinou vyvíja v jazykoch HTML, CSS a JS. Obsah môže byť statický alebo dynamický. Počítačové aplikácie môžu byť napísané vo veľmi veľkom rozsahu jazykov, záleží na platforme.

2. Aplikačná vrstva, túto vrstvu môžeme tiež poznať pod názvom logická vrstva alebo ako strednú vrstvu, táto vrstva je srdcom aplikácie. V tejto vrstve sa zozbierane informácie z prezenčnej vrstvy spracujú použitím nejakej logiky alebo nejakým použitím konkrétnych pravidiel. Budem pokračovať ďalej s príkladom na e-shope. Toto je vrstva ktorá vyhľadá v databáze a vráti či daný produkt je dostupný, alebo pridá informácie do užívateľovho profilu. Táto vrstva je najčastejšie vyvíjaná v jazykoch Python, Ruby alebo PHP a beží na frameworkoch ako napríklad Django, Rails, Symphony alebo ASP.NET. Tato vrstva komunikuje s dátovou vrstvou použitím API volaní.
3. Dátová vrstva, táto vrstva sa často nazýva aj ako databázová vrstva, vrstva prístupu k dátam alebo back-end, v tejto vrstve sú informácie spracované aplikáciou a následne sú uložené a manažované. Na výber máme z dvoch možností. Prvou možnosťou sú RDBMS servery na manažovanie dát akými sú napríklad PostgreSQL, MySQL, MariaDB, Oracle, DB2, Informix alebo Microsoft SQL Server. Druhou možnosťou sú NoSQL servery na manažovanie dát akými sú napríklad Cassandra, CouchDB alebo MongoDB. V trojvrstvovej aplikácii všetka komunikácia prechádza cez aplikačnú vrstvu. Prezentačná vrstva a dátová vrstva nemôžu komunikovať priamo medzi sebou.

Benefity trojvrstvovej architektúry. Najväčšou výhodou tejto architektúry je logické a fyzické rozdelenie funkcionality. Každá vrstva môže bežať na rôznych operačných systémoch a serverových platformách napríklad webový server, aplikačný server, databázový server na tom ktorý najlepšie spĺňa funkcionálne požiadavky. Každá vrstva beží na aspoň jednom špecializovanom servery. Takže každá vrstva

môže byť upravovaná a optimalizovaná bez toho aby to zasiahlo ostatné vrstvy. Medzi ďalšie výhody patria napríklad tieto:

1. Rýchlejší vývoj: Pretože každá vrstva môže byť vyvíjaná naraz rôznymi tímami a firma môže aplikáciu dostať na trh rýchlejšie a programátori môžu použiť najnovšie a najlepšie jazyky a nástroje pre každú vrstvu.
2. Vylepšená rozširiteľnosť: Každá vrstva môže byť škálovaná donekonečna osamostatnene od ostatných.
3. Zlepšená spoľahlivosť: Výpadok v jednej vrstve je menej náchylný nato, že zasiahne dostupnosť alebo výkon ostatných vrstiev.
4. Vylepšená bezpečnosť: Prezenčná vrstva a dátová vrstva nemôžu komunikovať spolu priamo, dobre nadizajnovaný aplikačná vrstva môže fungovať ako vnútorný firewall, ktorý chráni pred SQL injekciami a ďalšími možnými útokmi zo strany hackerov.

2.6 Search engine optimization

Ak chceme aby naša stránka bola úspešná je potrebné aby sa zobrazovala v Google search ako prvá, na toto nám slúži search engine optimization. Keď užívateľ zadá určité kľúčové slova tak sa mu zobrazí naša stránka najvyššie, to znamená nad našou konkurenciou. SEO ide o získavanie premávky z hľadaných výsledkov v search engine. Na základe keywords sú web stránky ohodnotené čím relevantnejšia pre používateľa tým vyššie hodnotenie. Top veci na ktoré sa chceme sústrediť.

1. Dobrá kvalita obsahu, Dlhší obsah sa umiestňuje vyššie. Dobrý zvykom je napísať krátke predstavenie článku napríklad. Najkrvavejšia bitka v histórii sveta. Snažiť sa písať len o keyworde v mojom príklade budem písať len o vojne. Používať relevantné obrázky so správne nastaveným alt textom. Dôležité je nezabúdať na mobilne telefóny, tablety a inú elektroniku a prispôbiť im CSS aby to vyzeralo dobre nielen na počítači ale aj na mobilnom telefóne a ostatných zariadeniach. Na mobile vždy preferujeme krátke vety oproti dlhším. Posledným bodom je aktualizovať náš obsah aby bol stále relevantný.
2. Rýchlosť stránky, definuje nám ako rýchlo sa zobrazí obsah stránky na displeji. Čím vyššia rýchlosť stránky tým je naša stránka lepšie hodnotená. Môžeme toho dosiahnuť pomocou zlepšenia odozvy servera. Napríklad zvýšením pamäte alebo úpravy databázových querin. Používať jednoduché dizajny stránok a nepoužívať príliš veľké obrázky. V neposlednom rade optimalizácia kódu ale môžeme aj použiť programy na Minifikáciu. Minifikácia spočíva v odstránení medzier a komentárov, mena premenných sa nahradia napríklad nasledovne. Premenná totoJePremenna sa zmení na a.

3. URL a title tag, je dôležité mať v názve stránky to o čom je v mojom prípade je napríklad užitočné mať stránku pomenovanú PrvaSvetovaVojna.sk alebo podobne. Krátke URL sa umiestňujú lepšie ako dlhšie.
4. SSL certifikáty, SSL alebo tiež známy ako aj TLS je protokol na šifrovanie internetovej premávky a overovanie serverovej identity. Každá stránka s HTTPS webovou adresou používa SSL. Stránky ktoré používajú HTTPS sa umiestňujú lepšie ako tie ktoré ho nepoužívajú.
5. Architektúra stránky, čím lepšia štruktúra stránky tým lepšie umiestnenie. Dobru štruktúru vieme dosiahnuť nasledovne. Použijeme hierarchickú štruktúru. Použijeme sitemap. Sitemap je súbor v ktorom poskytneme informácie o stránkach videách a iných súboroch našej stránky a popíšeme vzťahy medzi nimi. Search engine ako Google prečíta tento súbor a prechádza našu stránku efektívnejšie. Sitemap povie Googlu ktoré stránky sú podlá nás najdôležitejšie.
6. Backlinks, najdôležitejším bodom je pravé tento. Jedna sa o link z iných web stránok na našu web stránku. Tieto backlinky nám zvyšujú premávku na nasej stránke. Čím viac ich máme tým viac ľudí sa dozvie o našej web stránke.

Maslowa hierarchia potrieb. Je to psychologická teória ktorá pokladá za dôležitejšie základné ľudské potreby akými sú napríklad vzduch, voda a fyzické bezpečie, pričom za menej dôležité pokladá pokročilejšie potreby akými sú napríklad priateľstvo, sebavedomie. Teória je taká, že nemôžeme dosiahnuť pokročilejších potrieb bez toho aby sme najprv mali tie základne. Nezáleží či máme lásku ak nemáme jedlo. Na obrázku 6 môžeme vidieť, že niečo podobné vytvoril Rand Fishkin. Rozdelil SEO do 7 častí a zoradil ich do pyramídy. To znamená síce môžeme mať rýchlo načítavajúcu sa stránku avšak ak náš obsah stránky neodpovedá tomu čo

uživatel hledá je to zbytočné. [29]



Obr. 6: Mozlov hierarchy [30]

3 Návrh riešenia

3.1 Požiadavky webovej prezentácie

1. Použiteľnosť, webová stránka je určená pre prezentáciu vojen a bojov počas prvej svetovej vojny na Slovensku
2. Informovanosť, webová stránka obsahuje všetky informácie ktoré nás môžu zaujímať týkajúce sa tejto tematiky.
3. Dostupnosť, webová aplikácia bude k dispozícii na localhoste. Dostupná bude len keď bude spustená. Stránka je dostupná pre každého návštevníka na prezzeranie, ak však človek chce na túto stránku niečo pridávať musí sa zaregistrovať. Veľkou výhodou je naša znalosť o tom ktorý človek čo pridal a v prípade ak človek pridáva nesprávne informácie môžeme mu zakázať vstup na stránku. Nevýhodou je, že človeka môže registrácia odradiť od pridania informácii na stránku.
4. Zrozumiteľnosť, k stránke samozrejme bude poskytnutý aj jednoduchý manuál ako sa tato stránka ma používať. Pre tých menej technicky zdatných.
5. Jednoduchá ovládateľnosť, rýchle a jednoduché ovládanie na stránke. Vidíme a vieme sa dostať všade kde chceme.

3.2 Špecifikácia webovej aplikácie

Cieľom tohto projektu je realizovať webovú aplikáciu ktorá bude určená pre každého človeka ktorý sa zaujíma o prvú svetovú vojnu alebo má nejaké informácie o ktoré by sa rád podelil z obdobia prvej svetovej vojny. Používateľom je umožnené pristupovať na stránku z akéhokoľvek zariadenia ktoré má pripojenie k internetu. Samostatná aplikácia sa skladá z niekoľkých webových stránok ktoré sú rozdelené podľa požadovanej kategórie. Mam viaceré druhy užívateľov a každý užívateľ ma samozrejme rozdielne práva.

3.3 Cieľová skupina

Cieľovou skupinou tejto web stránky budú najmä ľudia zaujímajúci sa o históriu konkrétnejšie o históriu prvej svetovej vojny a o jej priebeh na Slovensku. Keďže záujem o históriu môže mať človek v každom veku môžeme predpokladať, že osoby ktoré navštívia moju stránku môžu mať od 18 do 70 rokov. Z toho nám vyplýva potreba vytvoriť prehľadné a intuitívne stránky ktoré budú ľahko ovládateľné v akomkoľvek veku. Vďaka tejto jednoduchosti sa budú chcieť užívatelia vracieť na našu stránku a nemajú potrebu ísť na inú stránku. Dôležité je mať aj intuitívne menu so všetkými hlavnými kategóriami. Na našu stránku bude chodiť veľká skupina používateľov na rôznych zariadeniach. Preto treba počítať s tým že na našu stránku sa budú pripájať nielen počítače ale aj iné zariadenia. Týmto zariadeniami môžu byť mobily, čítačky, tablety a pod. Z tohto dôvodu vzniká potreba vytvorenia responzívnej stránky aby Každé zariadenie malo stránku prispôbenú.

Každoročne stúpa počet zobrazení web stránok z telefónov. Momentálne sa nachádzame na číslach 68% návštev stránok prišlo z mobilov a 28% prišlo z počítačov a 3% prišli z tabletou. Z týchto čísel vyplýva, že je veľmi dôležité mať responzívnu

stránku pre mobily pretože tvoria viac ako polovicu našich návštevníkov avšak netreba zabudnúť ani na počítače pretože síce tvoria menšiu časť zobrazení avšak celkový čas strávený na stránkach je vyšší u počítačov ako u mobilov. Z týchto čísel môžeme vyvodiť nasledujúci záver našim hlavným cieľom je zabezpečiť peknú responsívnu stránku pre každý typ mobilu a pre každý typ počítača. Tablety a čítačky niesu až také dôležité. [31]

3.4 Technológie

Čo sa týká vyberú jazyka pre tvorbu web stránok na vyber máme veľmi veľa jazykov a frameworkov medzi tie najhlavnejšie patria.

1. Front-end

- (a) Jazyk JS a knižnica React
- (b) Jazyk JS a framework Angular
- (c) Jazyk PHP framework Laravel

2. Back-end

- (a) NodeJS a framework NestJS
- (b) Jazyk PHP framework Laravel
- (c) NodeJS nejedna sa ani o jazyk ani o framework jedna sa o JRE

3. Databázy

- (a) MongoDB
- (b) MySQL
- (c) PostgreSQL

Pre vývoj mojej web stránky som si vybral nasledujúce technológie. Jedna sa o takzvaný MEAN (solution stack).

1. MEAN je akronym pre MongoDB, Express.js, Angular, a Node.js. MongoDB použijem ako databázu. Sem by sme samozrejme očakávali Express.js avšak ja trošku tento MEAN pozmením a namiesto Express.js použijem NestJS. NestJS sa mi bude starať o routing. Angular sa mi bude starať o front-end. Node.js bude môj back-end. Dôvodom výberu týchto jazykov je viacero. Teraz popíšem ich výhody. [32]

- (a) JS patri medzi najlepšie a najpoužívanejšie jazyky webu. [33]
- (b) Píšem front-end aj back-end v jednom jazyku a tým jazykom je JS.
- (c) Jedná sa o jeden z najpoužívanejších frameworkov JS.
- (d) Mean stack je veľmi flexibilný.
- (e) AngularJs a NodeJs používajú JSON formát na ukladanie a posielanie dat, dokonca aj MongoDB používa JSON formát na ukladanie dat.
- (f) MongoDB je postavené pre cloud ktorý sa stava stále populárnejší.
- (g) Angular ma veľmi veľkú a aktívnu komunitu. To znamená že keď mám nejaký problém je veľmi veľká šanca, že takýto problém už niekto mal a bude mi vedieť poradiť.

Samozrejme ako každá vec na svete má aj MEAN stack nevýhody. Teraz tieto nevýhody popíšem

- (a) Pri veľkom nápore môže nastať strata údajov zapísaných v MongoDB
- (b) Neodporúča sa pre vývoj obrovských aplikácií.

2. Keďže používam viacero frameworkov, používam databázu. Spracovávam

front-end aj back-end vytvára to komplexitu na zjednodušenie tejto complexity použijem docker.

3. Pre zrýchlenie načítania mojej stránky a pre zníženie jej veľkosti použijem Minifier. Samozrejme minifier použijem len keď to uznám za vhodné a čas načítania mojej stránky bude príliš dlhý.

3.5 Povolenia

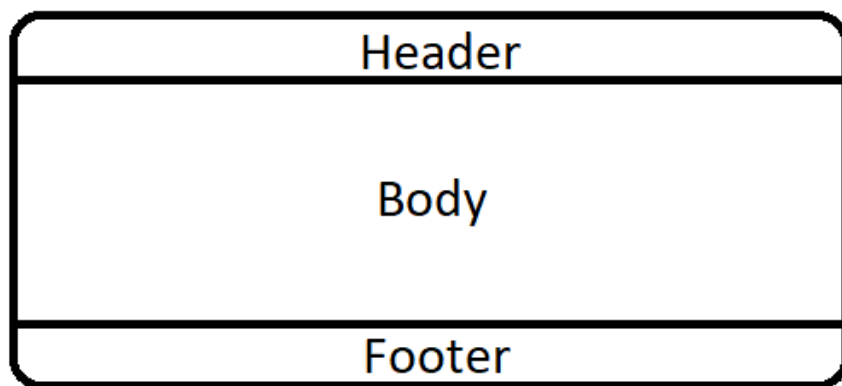
Na obrázku číslo 7 môžeme vidieť typy používateľov a právomoci každého jedného z nich.

	Prezerat' stránku	Pridávať pramene	Schvaľovať zdroje	Nahlásiť používateľa	Zablokovať užívateľa	Prístup k databáze	Prístup k zdrojovému kódu
Návštevník	👍	👎	👎	👎	👎	👎	👎
Používateľ	👍	👍	👎	👎	👎	👎	👎
Historik	👍	👍	👍	👎	👎	👎	👎
Admin	👍	👍	👍	👍	👍	👍	👎
Developer	👍	👍	👍	👍	👍	👍	👍

Obr. 7: Rozloženie stránky

3.6 Používateľské rozhranie

Návrh úplne základného modelu stránky môžeme vidieť na obrázku 8.



Obr. 8: Rozloženie stránky

Rozloženie je prehľadné a užívateľ sa na ňom prirodzene pohybuje od hlavičky dolu cez obsah až ku koncu stránky čiže ku pätičke. Toto rozdelenie má výhodu, pretože navigačné panely sa nachádzajú v hlavičke alebo v pätičke. Výhodou tohto

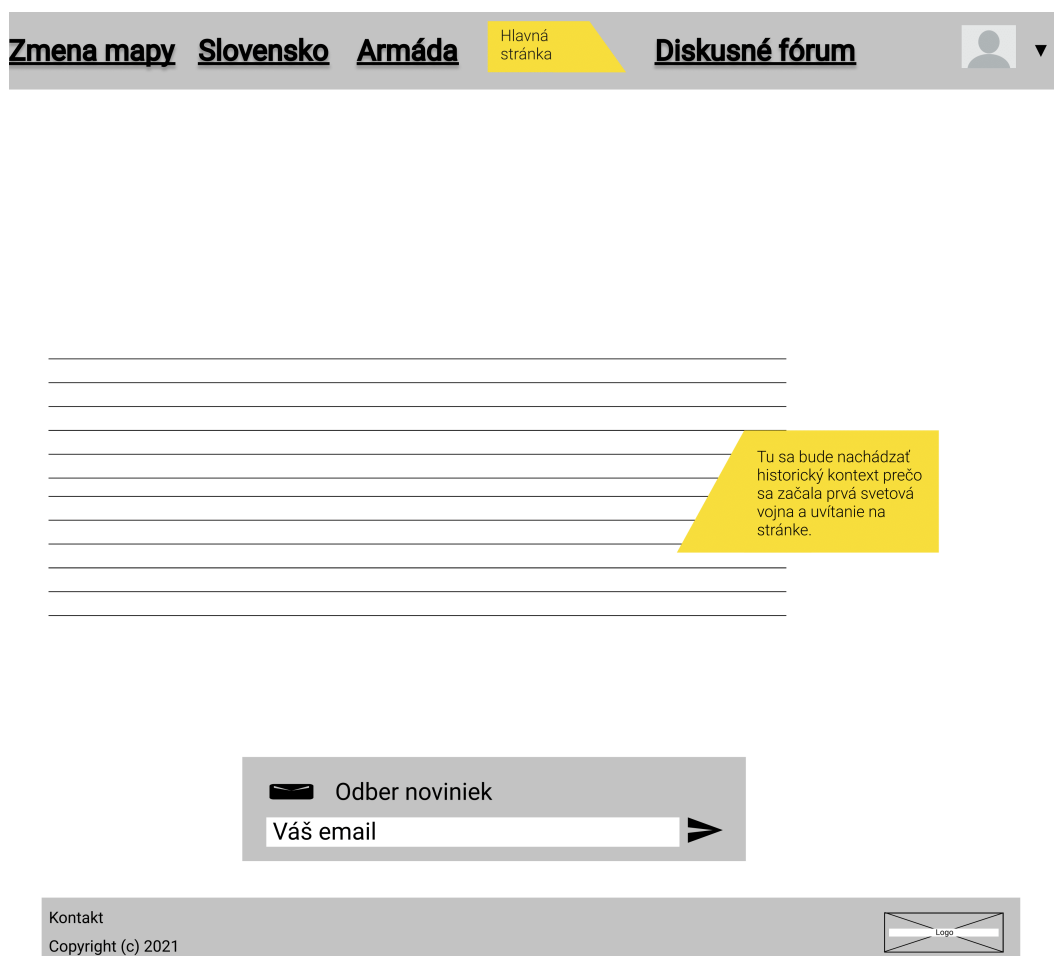
oproti bočnému navigačnému panelu je jednoduchšie vytvorenie responzivity. Ak by sme mali bočné panely mohlo by dochádzať medzi elementami ku kolízii a zničilo by to používateľsky zážitok. Máme zadefinovaný základný koncept na ktorom budú fungovať všetky stránky. V nasledujúcej časti ukážem na wireframeoch návrh stránok.

3.6.1 Hlavná stránka

Jedna sa o stránku ktorú uvidí používateľ ako prvú keď si otvorí našu web stránku. Úlohou tejto stránky je zaujať a poskytnúť informácie aby užívateľ vedel čomu sa naša stránka venuje.

Stránka pozostáva s headeru v ktorom je menu a linky na ďalšie pod stránky. Vpravo sa nachádza buď text prihlásiť alebo už konto prihláseného používateľa. Toto konto je ďalej rozkliknuteľné a zobrazia sa nám ďalšie informácie o používateľovi a samozrejme sa tam bude nachádzať aj tlačidlo pre odhlásenie. Vo footery sa nachádza kontakt copyright a logo. Keďže sme si vysvetlili čo sa nachádza v headery a footery už to znova nebudeme opakovať pretože pre každú stránku sa tam nachádza vždy to iste.

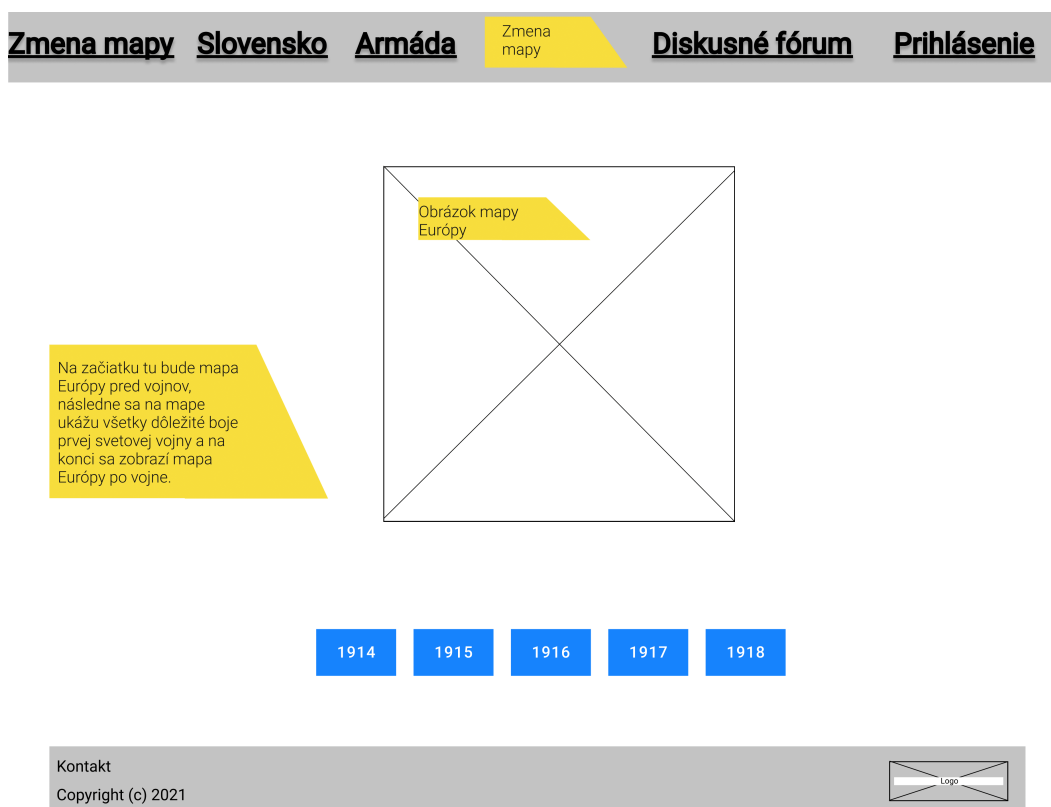
Teraz poďme na to najdôležitejšie a tým je čo sa bude na tejto stránke nachádzať. Bude sa tu nachádzať historicky kontext prečo sa začala prvá svetová vojna a uvítanie na našej web stránke. Ako som už vyššie spomínal ide o to upútať to znamená nezahltiť užívateľa príliš veľa informáciami. Dole môžeme vidieť aj Odber noviniek kde po zadaní emailu budú užívateľovi chodiť maily o novej funkcionalite našej stránky a o novo pridaných informáciách.



Obr. 9: Hlavná stránka

3.6.2 Zmena mapy - pod stránka

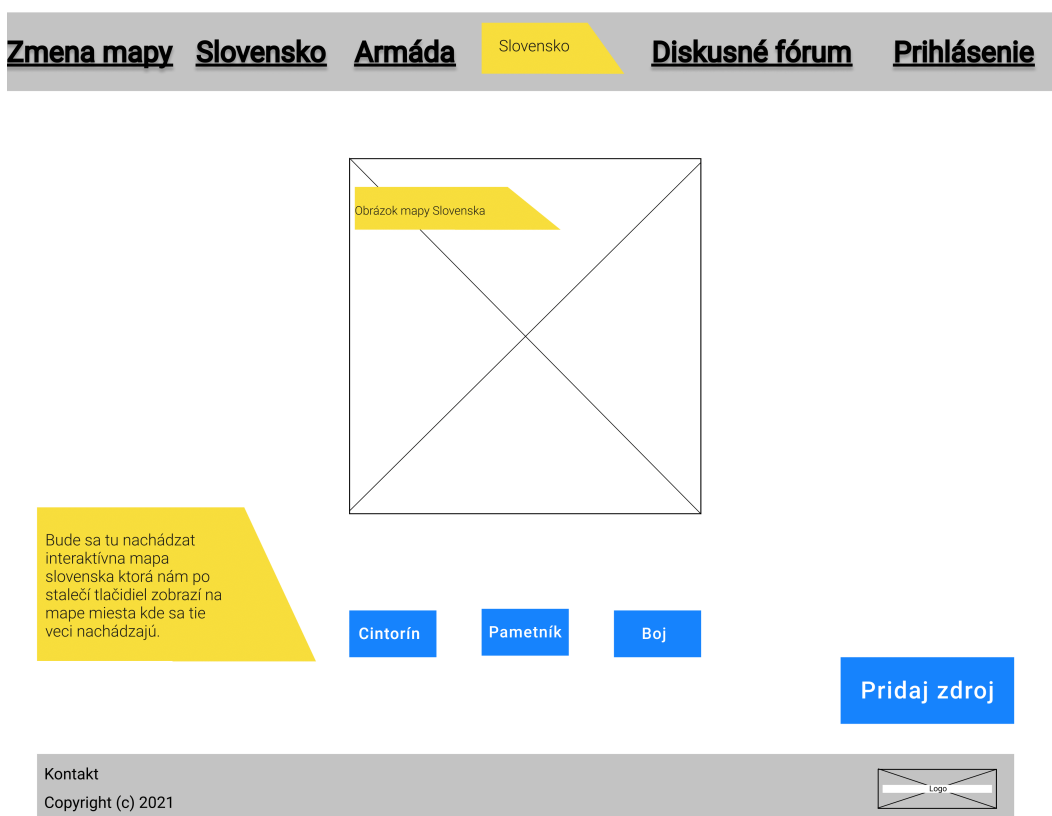
Na tejto pod stránke sa bude nachádzať mapa Európy pred vojnou následne sa ukážu všetky dôležité boje prvej svetovej vojny a na konci sa zobrazí mapa Európy po vojne. Zmena mapy bude ukázaná animáciou kde uvidíme ako sa posúvali fronty a nakoniec uvidíme aj novo vzniknuté štáty. Tieto bitky si bude môcť užívateľ pozrieť aj pomocou tlačidiel dole kde po kliknutí napríklad na rok 1914 sa mu ukážu všetky bitky v tomto roku.



Obr. 10: Zmena mapy

3.6.3 Slovensko - pod stránka

Na tejto pod stránke sa bude nachádzať interaktívna mapa Slovenskej republiky kde sa po stlačení tlačidla cintorín na mape ukážu všetky cintoríny z obdobia prvej svetovej vojny na Slovensku. Rovnako to bude fungovať aj pre tlačidla pamätník a tlačidlo boj. Chýbať tu nebude ani tlačidlo pridaj zdroj ktoré presmeruje užívateľa na ďalšiu pod stránku kde vyplní údaje a pridá nejaký zdroj, ako táto stránka pridania zdroja bude vyzerat' si ukážeme zachvíľu. Na tejto pod stránke sa bude samozrejme aj nachádzať legenda ktorá nám bude ukazovať aký znak znamená aký prameň prvej svetovej vojny.



Obr. 11: Slovensko

3.6.4 Pridanie zdroja - pod stránka

Na nasledujúcej stránke môžeme vidieť ako bude vyzeráť pridanie každého jedného zdroja to znamená pridanie cintorínu, pamätníka, boja. Hore v sliding window si užívateľ vyberie o aký typ zdroja ide následne sa mu zobrazí príslušný formulár na vyplnenie informácií. Informácie ktorými užívateľ nedisponuje môže nechať samozrejme aj prázdne. Na tomto obrázku vidíme ako vyzerá každý jeden formulár pretože som sa rozhodol to na ukážku dať všetko do jedného obrázka a preto to momentálne môže pôsobiť ako veľmi veľa informácií naraz pre užívateľa avšak užívateľovi sa zobrazí vždy iba jeden formulár takže zahltený nebude. Po pridaní prameňa je potrebné aby to skontroloval historik.

The image shows a web application interface for 'Slovensko'. At the top, there is a navigation bar with links: 'Zmena mapy', 'Slovensko', 'Armáda', 'Pridaj zdroj', 'Diskusné fórum', and 'Prihlásenie'. Below the navigation bar, there is a dropdown menu for 'Zdroj' with options: 'Zdroj', 'Cintorin', 'Pametnik', and 'Boj'. The main content area is divided into several sections:

- Pametnik**: A form with fields for 'Meno', 'GPS', 'Ulica', 'PSČ', 'Mesto', 'Dátum dokončenia výstavby', and 'Info'. A 'Pridat fotky' button is at the bottom.
- Cintorin**: A form with fields for 'Meno', 'GPS', 'Ulica', 'PSČ', 'Mesto', and 'Info'. There are 'Pridat fotky' and 'Pridat vojaka' buttons at the bottom.
- Boj**: A form with fields for 'Začiatok', 'Dátum', 'Koniec', 'Dátum', 'GPS', 'Mesto', 'Výsledok', 'Krajina', 'Vodca', 'Veliteľ', 'Sila', 'Straty', and 'Info'. There are 'Pridat jednotku' and 'Pridat fotky' buttons at the bottom.
- Jednotka**: A form with fields for 'Meno', 'Typ', 'Veľkosť', 'Komandéri', and 'Info'. A 'Pridat fotky' button is at the bottom.
- Vojak**: A form with fields for 'Celé meno', 'Narodenie', 'Úmrtie', 'Vek', 'Regiment', 'Boje', and 'Info'. A 'Pridat fotky' button is at the bottom.

There is a 'VS' label between the 'Boj' and 'Jednotka' forms. At the bottom of the page, there is a 'Pridaj zdroj' button and a footer with 'Kontakt' and 'Copyright (c) 2021'.

Obr. 12: Slovensko

3.6.5 Schválenie zdroja - pod stránka

Na obrázku číslo 13 môžeme vidieť ako vyzerá schválenie zdroja. Presnejšie ide o schválenie zdroja pamätníka, na obrázku sa budú nachádzať dve tlačidlá. Zelené pre schválenie zdroja a červené pre zamietnutie zdroja. Údaje akými sú Meno, GPS, Ulica a podobne budú vyplnené z databázy informáciami o konkrétnom zdroji ktorý bol poslaný na schválenie. Tieto informácie budú vložené do input

fieldov to znamená, že historik ich bude môcť meniť. Obrázky bude môcť užívateľ zmazať po kliknutí na červené x v pravom hornom rohu.

Zmena mapy Slovensko Armáda Schválenie zdroja Diskusné fórum Prihlásenie

Typ zdroja

Meno

GPS

Ulica

PSČ

Mesto

Dátum dokončenia výstavby

Info

Obrázky

Schváliť Zamietnuť

Kontakt Copyright (c) 2021 Logo

Obr. 13: Schválenie zdroja

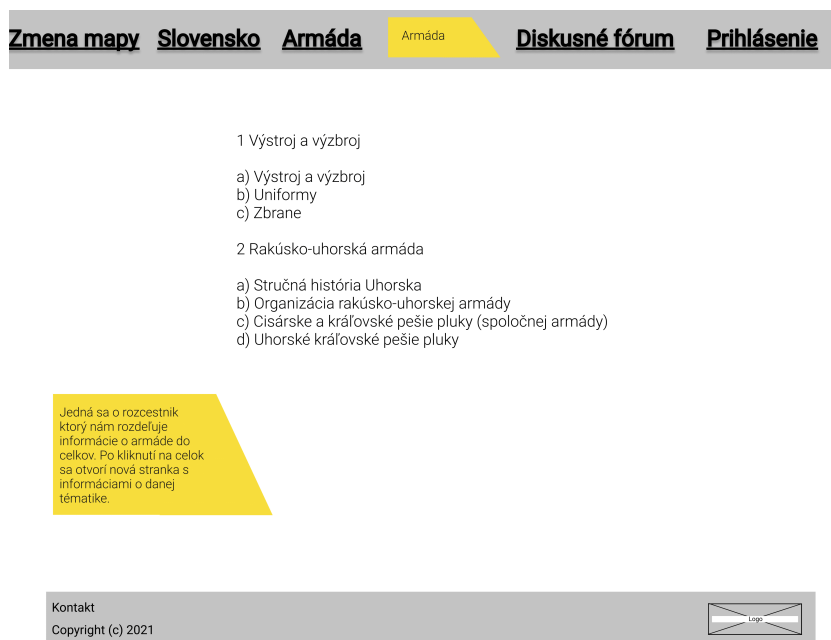
Ak by sa náš historik rozhodol, že tento prameň je správny a rozhodol by sa ho pridať a stlačil by tlačidlo schváliť. Vytiahli by sa všetky údaje a poslal by sa nasledovný JSON. Tento JSON môžeme vidieť na obrázku číslo 14. V tomto konkrétnom prípade sa jedná o typ zdroja pamätník. Údaje sú len ilustračné, nie sú pravdivé.

```
{  
  name: 'Slavín',  
  GPSCoordinates: '48.1583893,17.0642846',  
  street: 'Ilkovičova 2',  
  postalCode: '811 01',  
  city: 'Bratislava',  
  constructionDate: 1980,  
  info: 'jedná sa o pamätník z roku 1970',  
  photos: 'path/images/image',  
  state: 'approved',  
},
```

Obr. 14: JSON

3.6.6 Armáda - pod stránka

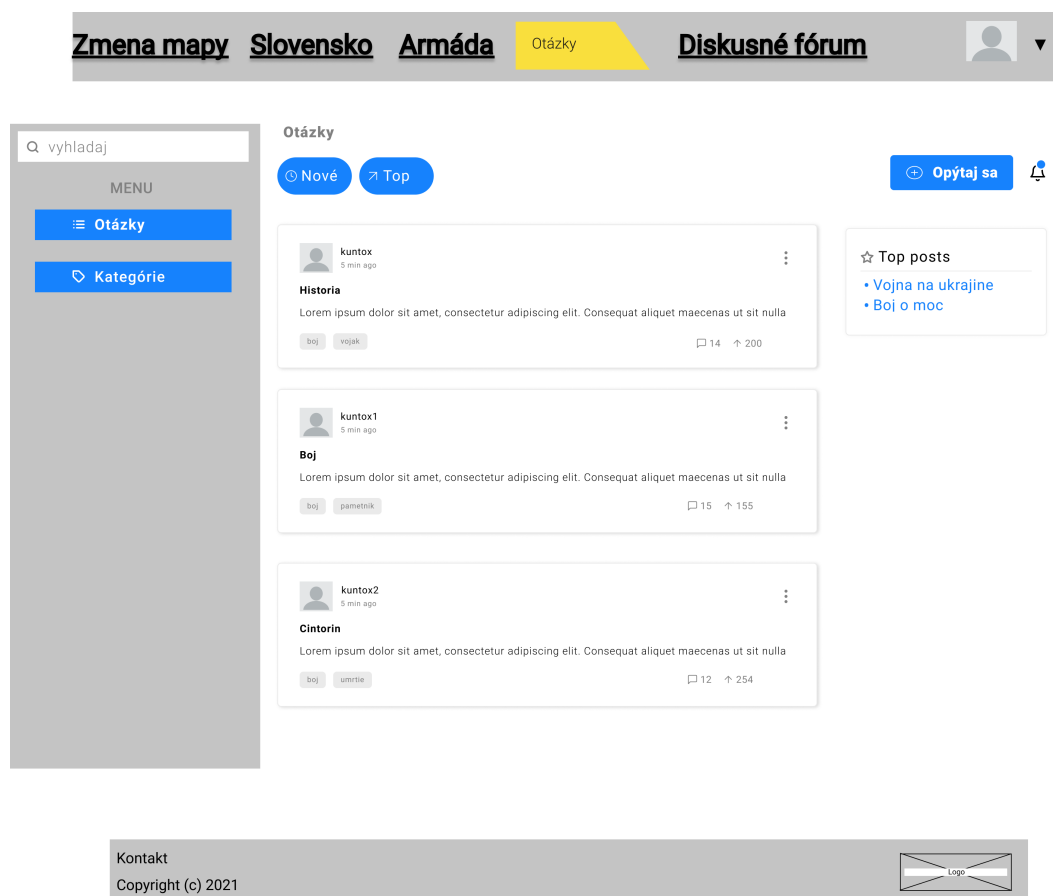
Tu sa bude nachádzať rozcestník. Tento rozcestník bude rozdelený na dve sekcie. Prvou sekciou bude výstroj a výzbroj, po kliknutí na text sa presunieme na ďalšiu pod stránku. Napríklad keď klikneme na uniformy presmeruje nás to a dozvieme sa informácie aké uniformy sa nosili počas prvej svetovej vojny. Druhou sekciou bude Rakúsko-Uhorska armáda. Znova po kliknutí na text budeme presmerovaný. Z názvu presmerovania vieme určiť čo sa tam bude nachádzať. Úlohou tejto stránky je užívateľa dobre nasmerovať k informácii ktorú práve hľadá.



Obr. 15: Armáda

3.6.7 Diskusné fórum - pod stránka

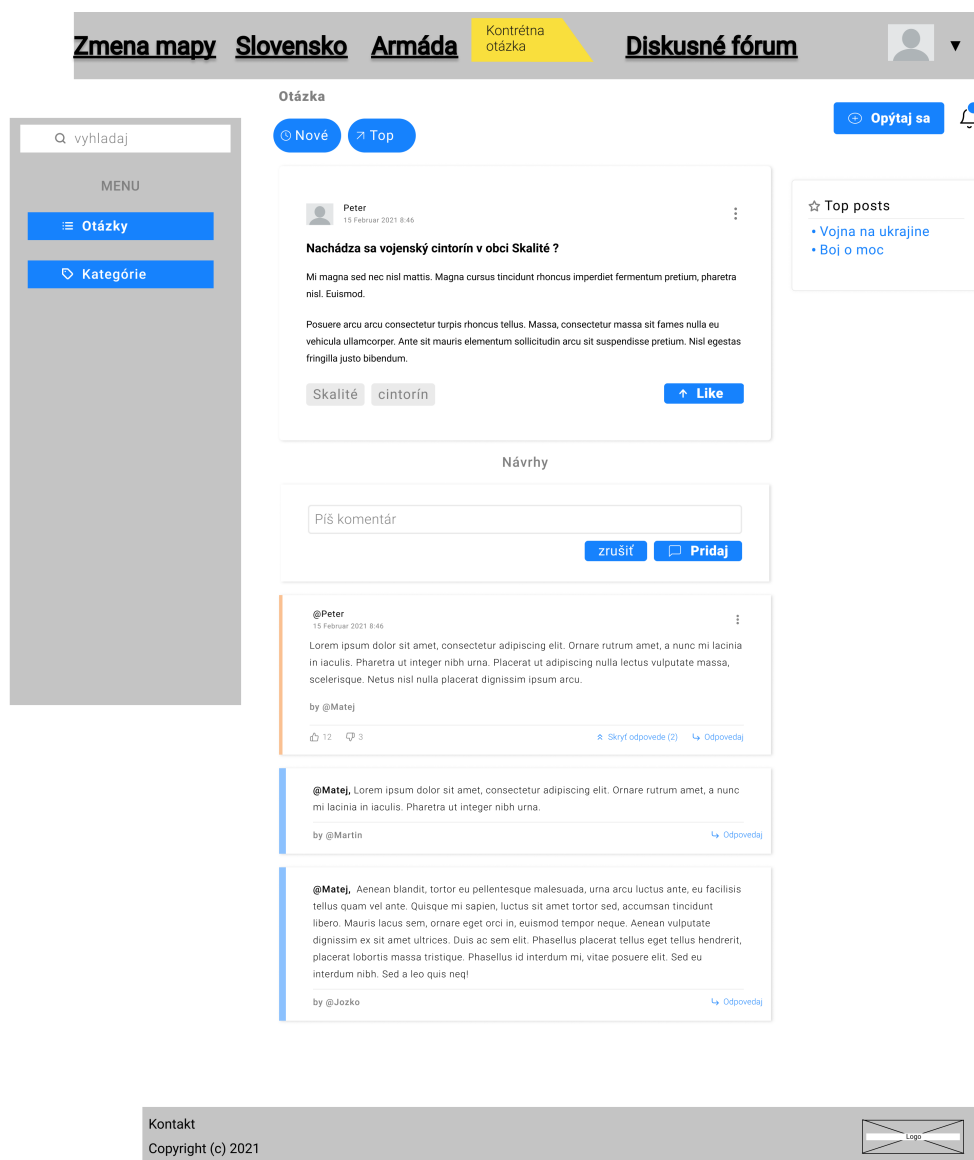
Nachádzame sa na pod stránke diskusného fóra. Vľavo môžeme vidieť vyhľadávanie do ktorého môžeme zadať kľúčové slová podľa ktorých budeme v databáze hľadať otázku. Môžeme vidieť aj tlačidlo otázky ktoré nám zobrazí všetky otázky. Tlačidlo kategórie nám dovolí zobrazit' otázky podľa kategórii. V strede môžeme vidieť, že sa dajú otázky zoradiť podľa dátumu pridania nato slúži tlačidlo nové. Samozrejme vieme zoradiť otázky aj podľa počtu lajkov nato nám slúži tlačidlo top. Ďalej môžeme vidieť ako vlastne tieto otázky vyzerajú. Napravo môžeme vidieť tlačidlo opýtaj sa ktoré slúži na pridanie otázky a pod týmto tlačidlom môžeme vidieť najaktívnejšie otázky.



Obr. 16: Diskusné fórum

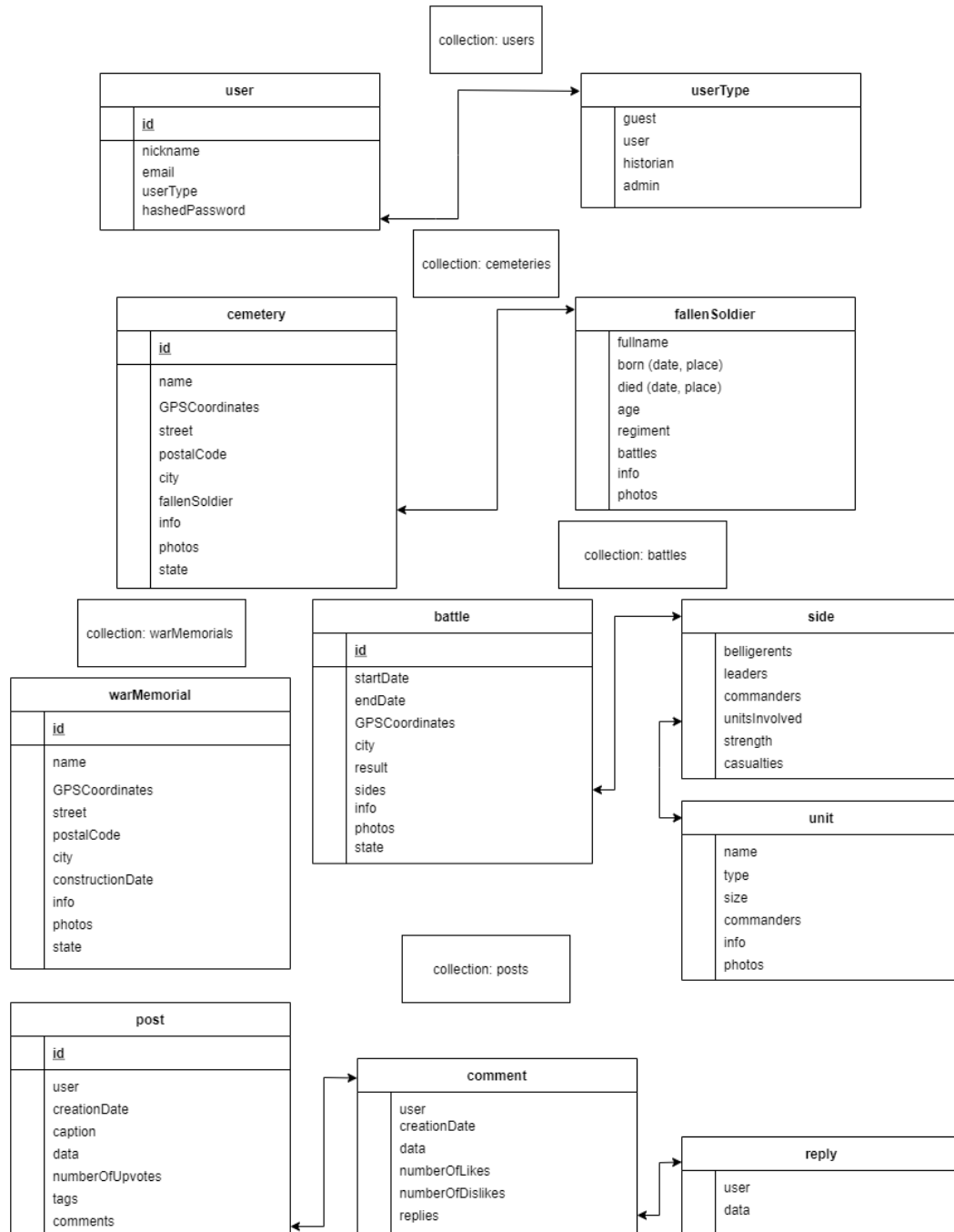
3.6.8 Konkrétna otázka - pod stránka

Na tomto obrázku môžeme vidieť ako bude vyzeráť keď si používateľ rozklikne otázku. Uvidíme otázku jej text a samozrejme jej kategórie. Tejto otázke môžeme dať like. Samozrejme vidíme aj komentáre k tejto otázke. Ku takýmto komentárom máme možnosť vidieť aj odpovede od ostatných používateľov. Tieto odpovede si môžeme aj skryť alebo môžeme tiež prispieť svojim komentárom. Každý komentár obsahuje na koho reagujeme, text komentára a meno používateľa ktorý ho napísal.



Obr. 17: Konkrétne otázka

3.7 Model databázy



Obr. 18: Model databázy

Na obrázku číslo 18 môžeme vidieť model mojej databázy. Používam mongoDB. Meno mojej databázy bude "bakalárskaPráca". V tejto databáze sa bude nachádzať viacero kolekcií. Každá takáto kolekcia bude obsahovať dokument. Teraz si to ukážeme na konkrétnom príklade. Kolekcia users obsahuje dokument user. V ktorom sa nachádzajú

field nickname(string) a value 'PeterPlevko'

field email(string) a value 'pplevko@gmail.com'

field userType(string) a value 'admin'

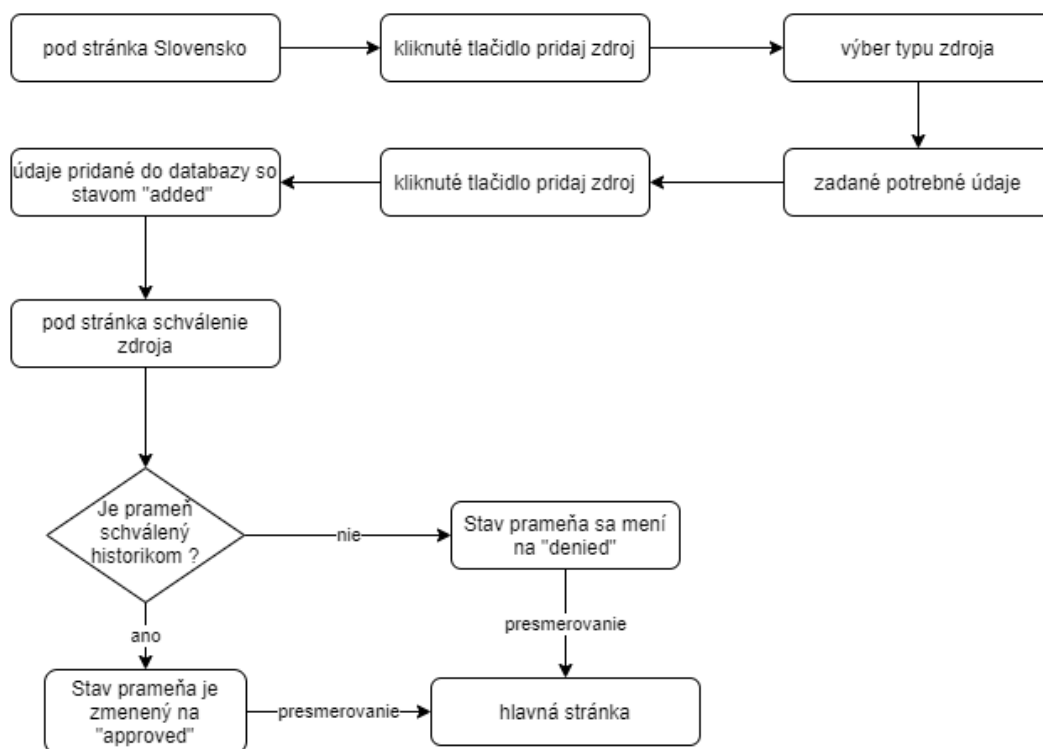
field hashedPassword(string) a value '*****'

V mojej databáze sa bude nachádzať 5 takýchto kolekcii. Každá kolekcia bude obsahovať viacero dokumentov.

3.8 Diagramy

3.8.1 Pridanie prameňa

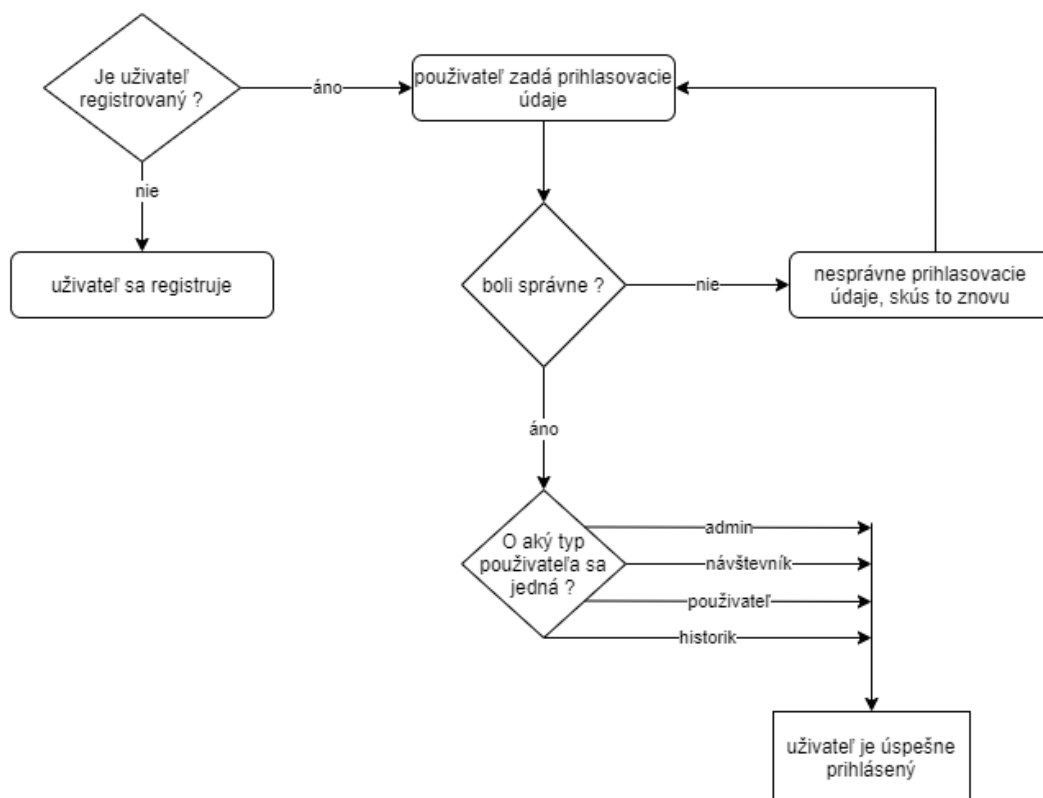
Na obrázku číslo 19 môžeme vidieť ako vyzerá pridanie prameňa. Cyklus je ukázaný od zadania prameňa užívateľom až po jeho schválenie alebo neschválenie historikom.



Obr. 19: Diagram pridania prameňa

3.8.2 Prihlásenie používateľa

Na obrázku číslo 20 môžeme vidieť ako vyzerá prihlásenie užívateľa. Cyklus je ukázaný od registrácie užívateľa, cez kontrolu prihlasovacích údajov a kontrolu typu užívateľa až po prihlásenie.



Obr. 20: Diagram Prihlásenie používateľa

Literatúra

- [1] Juraj Červenka. *Prvá Svetová Vojna*. Jan. 2021. URL: <https://beliana.sav.sk/heslo/prva-svetova-vojna>.
- [2] Michael Howard. *The first world war*. Oxford University Press, USA, 2003.
- [3] popumaps. *Map of europe 1914*. Feb. 2013. URL: <https://popumaps.blogspot.com/2013/02/map-of-europe-1914.html?m=0>.
- [4] M Mark Stolarik. „The role of American Slovaks in the creation of Czechoslovakia, 1914-1918“. Diz. pr. University of Ottawa (Canada), 1967.
- [5] Mehdi Jazayeri. „Some Trends in Web Application Development“. In: *Future of Software Engineering (FOSE '07)*. 2007, s. 199–213. DOI: [10.1109/FOSE.2007.26](https://doi.org/10.1109/FOSE.2007.26).
- [6] Bruce Perens et al. „The open source definition“. In: *Open sources: voices from the open source revolution 1* (1999), s. 171–188.
- [7] *What is frameworks? [definition] types of frameworks*. URL: <https://hackr.io/blog/what-is-frameworks>.
- [8] Ampomah Ernest Kwame, Ezekiel Mensah Martey a Abilimi Gilbert Chris. „Qualitative assessment of compiled, interpreted and hybrid programming languages“. In: *Communications 7.7* (2017), s. 8–13.
- [9] Chuck Musciano a Bill Kennedy. *HTML & XHTML: The Definitive Guide: The Definitive Guide*. Ö'Reilly Media, Inc.", 2002.
- [10] Eric A Meyer. *CSS: The Definitive Guide: The Definitive Guide*. Ö'Reilly Media, Inc.", 2006.
- [11] David Flanagan. *JavaScript: the definitive guide*. 2013.
- [12] Jonathan Chaffer. *Learning jQuery*. Packt Publishing Ltd, 2013.

- [13] Stefan Tilkov a Steve Vinoski. „Node. js: Using JavaScript to build high-performance network programs“. In: *IEEE Internet Computing* 14.6 (2010), s. 80–83.
- [14] *Documentation: Nestjs - a progressive node.js framework*. Sept. 2017. URL: <https://docs.nestjs.com/>.
- [15] Elar Saks. „JavaScript Frameworks: Angular vs React vs Vue.“ In: (2019).
- [16] Minifier. *Minify JS and CSS Online, or include the minifier in your project for on-the-fly compression*. Máj 2009. URL: <https://www.minifier.org/>.
- [17] Rasmus Lerdorf et al. *Programming Php*. Ö'Reilly Media, Inc.", 2002.
- [18] Paul DuBois. *MySQL*. Pearson Education, 2008.
- [19] Behandelt PostgreSQL. „PostgreSQL“. In: *Web resource: http://www. PostgreSQL. org/about* (1996).
- [20] Kristina Chodorow. *MongoDB: the definitive guide: powerful and scalable data storage*. Ö'Reilly Media, Inc.", 2013.
- [21] Charles Anderson. „Docker [software engineering]“. In: *Ieee Software* 32.3 (2015), s. 102–c3.
- [22] David Gourley et al. *HTTP: the definitive guide*. Ö'Reilly Media, Inc.", 2002.
- [23] *HTTP request methods - http: MDN*. URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>.
- [24] Kaspersky. *What is Data Encryption?* Aug. 2021. URL: <https://www.kaspersky.com/resource-center/definitions/encryption>.
- [25] Audun Jøsang. „A Consistent Definition of Authorization“. In: sept. 2017, s. 134–144. ISBN: 978-3-319-68062-0. DOI: [10.1007/978-3-319-68063-7_9](https://doi.org/10.1007/978-3-319-68063-7_9).
- [26] Justin Clarke. *SQL injection attacks and defense*. Elsevier, 2009.
- [27] Surabhi Pandey. *Three tier architecture : The beginning*. Feb. 2018. URL: <https://medium.com/coffeetechandme/three-tier-architecture-the-beginning-2d2f6063fa1e>.

- [28] Eduardo B Fernandez et al. „The secure three-tier architecture pattern“. In: *2008 International Conference on Complex, Intelligent and Software Intensive Systems*. IEEE. 2008, s. 555–560.
- [29] Harold Davis. *Search engine optimization*. Ö'Reilly Media, Inc.", 2006.
- [30] *Beginner's Guide to SEO [search engine optimization]*. URL: <https://moz.com/beginners-guide-to-seo>.
- [31] Eric Enge. *Mobile vs. desktop usage in 2020*. Mar. 2021. URL: <https://www.perficient.com/insights/research-hub/mobile-vs-desktop-usage>.
- [32] N Nirgudkar a P Singh. „The MEAN stack“. In: *International Research Journal of Engineering and Technology* (2017), s. 2395–56.
- [33] Paul Javin. *Top 5 programming languages for web development in 2021*. Jan. 2021. URL: <https://medium.com/javarevisited/top-5-programming-languages-for-web-development-in-2021-f6fd4f564eb6>.

A Harmonogram práce v zimnom semestri

1.týždeň	Štúdium problému
2.týždeň	Zber literatúry
3.týždeň	Analýza prvej svetovej vojny
4.týždeň	Analýza technológií webovej aplikácie
5.týždeň	Analýza JavaScriptu
6.týždeň	Analýza zabezpečenia aplikácie
7.týždeň	Analýza search enginu
8.týždeň	Tvorba návrhu riešenia
9.týždeň	Výber technológií
10.týždeň	Modelovanie databázy
11.týždeň	Tvorba používateľského rozhrania
12.týždeň	Upravovanie finálneho dokumentu

A.1 Zhodnotenie

Prvá časť bakalárskeho projektu sa odohrávala v zimnom semestri. Prvých 7 týždňov som mne zadanú tému analyzoval a študoval. Nasledujúce týždne som tvoril návrh riešenia. Navrhol som si ako bude moja web stránka vyzeráť pomocou programu figma. Pomocou programu draw.io som navrhol model databázy a diagramy. V poslednom týždni som si prácu ešte raz celú prečítal a urobil finálne úpravy. Počas práce na projekte som sa každý týždeň zúčastňoval na konzultáciách so školiteľom. Prešli sme si čo som napísal a diskutovali o nedostatkoch, nakoniec som spomenul čo mam v pláne stihnúť do ďalšej konzultácie.