



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Vícejazyčný informační dynamický web

Bakalářská práce

Studijní program: B2646 – Informační technologie
Studijní obor: 1802R007 – Informační technologie
Autor práce: **Matěj Beran**
Vedoucí práce: Mgr. Kamil Nešetřil, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

Multilingual information dynamic web

Bachelor thesis

Study programme: B2646 – Information Technology
Study branch: 1802R007 – Information Technology

Author: **Matěj Beran**
Supervisor: Mgr. Kamil Nešetřil, Ph.D.



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Jméno a příjmení: **Matěj Beran**
Název práce: **Vícejazyčný informační dynamický web**
Zadávací katedra: **Ústav mechatroniky a technické informatiky**
Vedoucí práce: **Mgr. Kamil Nešetřil Ph.D.**
Rozsah práce: **30–40 stran**
Konzultant:

Zásady pro vypracování:

1. Navrhněte a implementujte webovou aplikaci umožňující správu a zpřístupnění jazykových verzí externích informačních zdrojů (odkazy na informační obsah na jiných vícejazyčných webech).
2. Aplikace ať umožňuje správu, zpracování a validaci zdrojů, poloautomatizovanou lokalizaci do velkého množství jazyků a rozhodovací logiku na základě zeměpisné polohy, jazyka prohlížeče a typu zařízení.
3. Napište testy. Aplikaci otestujte, nasadte a monitorujte.

Seznam odborné literatury:

- [1] KROENKE, David a David J. AUER, 2015. Databáze. Přel. Jakub GONER. Brno: Computer Press. ISBN 978-80-251-4352-0.
- [2] LAVIN, Peter, 2009. PHP - objektově orientované: koncepty, techniky a kód. Přel. Michal POSPÍŠEK. Praha: Grada. ISBN 978-80-247-2137-8.
- [3] ISO 639-3. SIL International 2017 [online]. Dostupné z: <https://iso639-3.sil.org>
- [4] LOCKHART, Josh, 2015. Modern PHP: New Features and Good Practices. 1 edition. Sebastopol, CA: O'Reilly Media. ISBN 978-1-4919-0501-2.

V Liberci dne

.....
Mgr. Kamil Nešetřil Ph.D.

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že texty tištěné verze práce a elektronické verze práce vložené do IS STAG se shodují.

11. 4. 2019

Matěj Beran

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce Mgr. Kamilu Nešetřilovi, Ph.D. za odborné vedení, cenné rady a věcné připomínky při konzultaci a vypracování této práce.

Děkuji své rodině a mým přátelům za psychickou podporu, čas a prostor, který mi během vytváření této bakalářské práce poskytli.

Abstrakt

Tato práce se zabývá tvorbou webové aplikace, která zpřístupňuje jejím návštěvníkům externí informační zdroje umístěné na jiných webových stránkách v různých jazykových mutacích. Aplikace umí tyto zdroje návštěvníkovi navrhnout v závislosti na jeho geolokaci, jazyku prohlížeče a typu zařízení. Aplikace je naprogramována v jazyce PHP za použití frameworku Symfony a standardních nástrojů pro vývoj webových aplikací. Po přihlášení s právy administrátora umožňuje spravovat zdroje a jejich URL, jazyk, státy a uživatele. Aplikace umí zdroje automaticky validovat a generovat.

Klíčová slova

PHP, Symfony, Doctrine, web, vícejazyčnost

Abstract

This thesis deals with creating a web application, that provides its visitors with external information resources located on other websites in different language versions. The application can suggest these resources to the visitor depending on its geolocation, browser language and device type. The application is programmed in PHP using the Symfony framework and standard web application development tools. Once logged in with administrator rights, it allows you to manage external resources and their URLs, languages, states and users. The external resources can be automatically validated and generated.

Keywords

PHP, Symfony, Doctrine, web, multilingual

Obsah

Prohlášení.....	4
Abstrakt.....	6
Klíčová slova.....	6
Abstract.....	6
Keywords.....	6
Seznam ilustrací.....	9
Seznam zkratk.....	10
Úvod.....	11
1 Standardy jazykových a státních kódů.....	12
1.1 ISO 639.....	12
1.1.1 ISO 639-1.....	12
1.1.2 ISO 639-2.....	12
1.1.3 ISO 639-3.....	12
1.2 ISO 3166.....	13
1.2.1 ISO 3166-1.....	13
2 Návrh řešení.....	14
2.1 Požadavky na aplikaci.....	14
2.2 Datový model.....	14
2.3 Návrh databáze.....	14
2.4 Lokalizace.....	15
2.5 Zdroje.....	16
2.6 Administrace.....	17
2.7 Výběr technologie.....	17
2.7.1 Symfony.....	17
2.7.2 Doctrine 2.....	18
3 Implementace.....	19
3.1 Vývojové nástroje.....	19
3.1.1 XAMPP Stack.....	19
3.1.2 Git.....	19
3.2 Frontend.....	19
3.2.1 Automatický výběr jazyka.....	19
3.2.2 GeoIP 2.....	20
3.3 Zobrazování zdrojů.....	21
3.3.1 Haversinův vzorec.....	22
3.4 Administrace.....	22

3.4.1	Validování zdrojů	23
3.4.2	Poloautomatizovaná lokalizace zdrojů	23
3.4.3	Plánovač úloh.....	24
4	Testování.....	25
4.1	Import dat.....	25
4.2	PHPUnit.....	25
4.2.1	Jednotkové testování	25
4.2.2	Funkční testování.....	26
4.3	Statická analýza	26
5	Nasazení	27
5.1	FTP.....	27
5.2	SSH.....	27
6	Závěr	29
6.1	Další možnosti rozvoje.....	30
	Použitá literatura.....	31
	Příloha I – Datový model	32
	Příloha II – Logika zobrazování zdrojů	33
	Příloha III – obsah přiloženého DVD.....	34

Seznam ilustrací

Obrázek 2.1: Konceptuální návrh databáze	15
Obrázek 2.2: Lokalizace	16
Obrázek 3.1: Ukázka manuálního výběru jazyka dle státu.....	20
Obrázek 3.2: Ukázka zobrazení zdrojů.....	21
Obrázek 3.3: Ukázka administrace jazyků.....	22
Obrázek 3.4: Ukázka generování URL zdroje	23

Seznam zkratek

API	Application Programming Interface
CMS	Content Management System
FTP	File Transfer Protocol
ISP	Internet Service Provider
ORM	Object-Relational Mapping
SSH	Secure Shell
URL	Uniform Resource Locator
VPN	Virtual Private Network

Úvod

Hlavním cílem této práce bylo navrhnout a implementovat webové řešení, které bude zpřístupňovat jeho návštěvníkům externí informační zdroje umístěné na jiných webových stránkách v různých jazykových mutacích. Tyto zdroje bude zobrazovat v závislosti na jazyce, geolokaci a typu zařízení. Web je podporován administrací, která takový web umožňuje spravovat.

Požadavky vycházejí z existujícího portálu `nextinfo.cz`, který je implementován jako statický web. Spravuje ho jeden člověk, který je schopen editovat HTML. Vyvíjený portál má strukturovaný charakter, a tedy je u něj možné generovat obsah automaticky, proto se zde nabízí využití automatizovaného řešení.

Tato práce navazuje na předchozí zpracování v bakalářském projektu, který byl zpracován jako prototypní řešení.

Jako alternativní řešení byly zvažovány existující služby pro generování webových stránek a CMS jako je například Wordpress, Joomla nebo Drupal. Zmíněné řešení ale obsahují hodně nepotřebných knihoven a pro zobrazování zdrojů by se musel buď vytvořit nový modul v rámci pravidel CMS nebo by se musel upravit nějaký z existujících modulů (například Blog). Nakonec se tedy přistoupilo k vytvoření nové aplikace, která umožňuje se lépe přizpůsobit specifickým požadavkům bez nutnosti obcházet existující kód výše zmíněných řešení.

1 Standardy jazykových a státních kódů

Protože webové řešení vyžaduje lokalizaci, bylo nutné provést rešerši standardů, které tyto kódy popisují.

1.1 ISO 639

ISO 639 [1] je jedním z mezinárodních standardů, který udává seznam kódů pro názvy jazyků. Standard se skládá z několika částí. Některé jsou zatím nedokončené.

Tento standard je široce používán především lingvisty a etnology pro kategorizování mluvených jazyků v rámci regionů.

1.1.1 ISO 639-1

ISO 639-1 [2] je první část mezinárodního standardu ISO 639 definujícího kódy jazyků. Každý jazyk je zde popsán dvouznakovým kódem. Norma nezahrnuje strojové jazyky (např. programovací). Ovšem obsahuje některé umělé jazyky jako je esperanto nebo interlingua. Norma zahrnuje 184 dvouznakových kódů. Například čeština je identifikována kódem *cs*.

1.1.2 ISO 639-2

ISO 639-2 je druhou částí mezinárodního standardu ISO 639. Standard byl publikován roku 1998, ačkoliv pracovat se na něm začalo již v roce 1989. Momentálně tato norma obsahuje 547 světových jazyků. Například pro češtinu je zde kód *cze/ces*.

1.1.2.1 B a T kódy

Některé jazyky jsou popsány dvěma kódy. Bibliografický kód (ISO 639-2/B) je odvozený z anglického názvu jazyka. Naopak terminologický kód (ISO 639-2/T) je odvozen z názvu jazyka v něm samém. Například pro němčinu kód *ger* je z anglického German, naopak *deu* je z německého deutsch.

1.1.3 ISO 639-3

ISO 639-3 [3] byla publikována 5. února 2007. Hlavním úkolem je rozšířit standard ISO 639-2 s cílem pokrýt všechny přírodní jazyky. K roku 2019 norma obsahuje 7 863 záznamů.

Poskytuje ucelený výčet jazyků v maximálním měřítku. Standard obsahuje co možná nejúplnější výčet jazyků živých i mrtvých, starověkých i umělých, světových i menšinových, psaných i nepsaných. Nezahrnuje ovšem rekonstruované jazyky jako je protoindoevropština.

1.2 ISO 3166

ISO 3166 [4] je jeden ze standardů, který definuje kódy států a závislých oblastí. Byl vydán v roce 1974 a skládá se ze tří částí: ISO 3166-1, ISO 3166-2 a ISO 3166-3.

Pro rozsah práce stačí kódy ISO 3166-1, ostatní normy poté pouze doplňují normu první o informace, které nejsou relevantní. ISO 3166-2 doplňuje kódy jednotlivých částí uvnitř státu. A ISO 3166-3 přidává kódy států, které od vydání ISO 3166 zanikly nebo se z nich staly státy jiné.

1.2.1 ISO 3166-1

První část z normy ISO 3166 definuje tři kódy pro každý stát. A to numerický, dvoupísmenný a třípísmenný. Jedná se pouze o základní výčet, obsahuje 249 záznamů. Například pro Českou republiku obsahuje záznam *203/CZE/CZ/Česko*.

2 Návrh řešení

2.1 Požadavky na aplikaci

Dle zadání se má jednat o webovou aplikaci, která zpřístupní externí jazykové zdroje v mnoha jazykových verzích. Hlavním cílem je pak navrhnout rozhodovací logiku pro zobrazení zdrojů na základě jazyka prohlížeče, zeměpisné polohy a typu zařízení.

Aplikace má disponovat administračním rozhraním, které administrátorovi umožní tuto aplikaci spravovat. Má umožnit správu zdrojů, jejich validaci a poloautomatizovanou lokalizaci. Na závěr je pak potřeba aplikaci otestovat, nasadit a monitorovat.

2.2 Datový model

Pro aplikaci je nutné zobrazovat a spravovat zdroje v mnoha jazycích. Proto byl navržen datový model, který je přiložen (viz Příloha I).

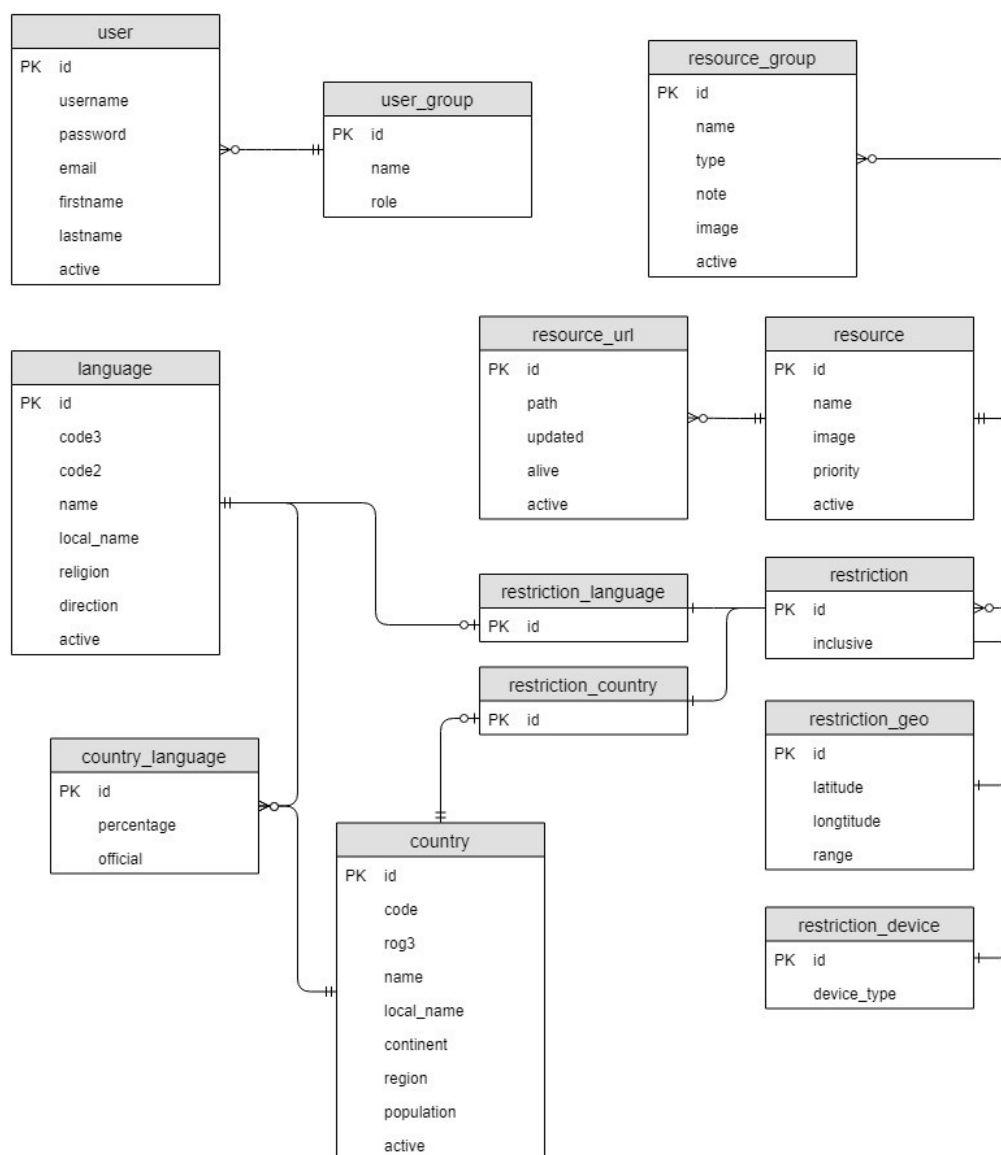
Prvním bodem je třída uživatele a uživatelské skupiny, které umožní realizovat víceuživatelské administrační rozhraní.

Dále třídy pro jazyky a státy. Protože web a zdroje mají být lokalizované, byly navrženy třídy *Language* a *Country*, které mezi sebou mají vztah M:N realizovaný přes třídu *CountryLanguage*. Tato třída uchovává pouze informace, zda je jazyk v daném státě oficiální a kolik procent populace jím mluví.

Poslední, ale nejdůležitější je pak návrh datového modelu pro samotné zdroje. V návrhu je možné vidět, že zdroje se organizují do jednotlivých skupin pro přehlednost. Následně zdroje obsahují jednotlivé URL, které jsou závislé na jazyce. Zároveň agregují omezení zdroje, které se dle požadavků dělí na omezení jazyka, státu, geolokace a typu zařízení.

2.3 Návrh databáze

Konceptuální návrh databáze (viz obrázek 2.1) vychází z datového modelu. Jelikož je databáze implementována formou ORM, kde jsou středem aplikace entity, konceptuální model nebyl nutný, ale pomohl v jejich modelování.



Obrázek 2.1: Konceptuální návrh databáze

2.4 Lokalizace

Jako lokalizace je myšleno překládání uživatelského rozhraní do algoritmem nebo uživatelem zvoleného jazyka. Jak je uvedeno výše (viz kapitola 1), existuje mnoho standardů pro jazykové kódy. Pro lokalizaci byl zvolen standard ISO 639-1 z několika důvodů.

Prvním důvodem byla praktičnost, protože by bylo velmi náročné web překládat do cca 7 000 jazyků, pokud by se zvolil jeden ze standardů ISO 639-2 nebo ISO 639-3. Druhým důvodem jsou již definovaná locale v samotném jazyku PHP, který se řídí unixovou formou zápisu. Jak funguje lokalizace pro návštěvníka webu je znázorněno níže (viz obrázek 2.2).



Obrázek 2.2: Lokalizace

2.5 Zdroje

Zdrojem je myšlena sada odkazů, které směřují na externí zdroj v různých jazycích. Tyto zdroje se zobrazují uživateli, který web navštíví na základě několika parametrů: zvolený jazyk, stát, zeměpisná poloha a typ zařízení.

Zobrazování zdrojů uživateli poté funguje tak, že se zkontrolují všechna omezení pro daný zdroj. Pokud alespoň jedno z omezení nabylo platnosti, pak se zdroj nezobrazí.

Zároveň se zde ještě řeší exkluzivita omezení. Například pokud bude nastaveno jazykové omezení pro jazyk čeština a bude označeno jako exkluzivní, pak se zdroj bude zobrazovat uživatelům s jakýmkoliv jiným jazykem. Celá logika je v příloze II.

2.6 Administrace

Administrační rozhraní má poskytovat administrátorům schopnost spravovat uživatele, zdroje, státy, jazyky a skupiny zdrojů. V závislosti na uživatelské skupině bude mít uživatel různé úrovně oprávnění. Rozhraní má dále umožňovat poloautomatizované generování URL zdroje (viz kapitola 3.4.2), které značně zjednoduší vytváření nových zdrojů lokalizovaných do mnoha jazyků.

2.7 Výběr technologie

Webové řešení je možné realizovat pomocí mnoha programovacích jazyků. Například PHP, Ruby nebo JavaScript. Každý z jazyků poskytuje řadu frameworků, které urychlují proces vývoje. Pro JavaScript je to například Express.js, pro PHP pak například Symfony, CakePHP nebo Laravel.

Pro realizaci aplikace byl zvolen programovací jazyk PHP ve verzi 7.2 v kombinaci s webovým aplikačním frameworkem Symfony a MySQL databází realizovanou formou ORM. Žádná z technologií nemá výrazný benefit v rychlosti nebo ve vyhovování požadavkům aplikace. Hlavním důvodem výběru technologie byla tedy již existující znalost autora. Dalším aspektem byla široká komunita a kvalitní dokumentace frameworku Symfony.

2.7.1 Symfony

Symfony [5] je framework nebo sada nástrojů pro vývoj webových aplikací. Symfony vychází z návrhového vzoru MVC a je inspirováno řadou jiných webových aplikačních frameworků jako je například Ruby on Rails, Django nebo Spring.

Framework Symfony byl jako open-source poprvé ve své verzi 1.0 vydán roku 2005 pod licencí MIT. Od té doby prošel framework mnoha změnami a nyní se nachází ve verzi 4.2. Vývoj Symfony je dále podporován širokou komunitou a sponzorován francouzskou firmou Sensio Labs.

Symfony také čerpá z mnoha open-source projektů, které implementuje jako součást frameworku. Jedná se například o Doctrine, Twig, PHPUnit nebo Swift Mailer. Zároveň Symfony poskytuje své knihovny volně k běžnému použití, bez jádra Symfony.

2.7.2 Doctrine 2

Pro databázi byla vybrána databázová vrstva Doctrine 2. Funguje na principu Objektově Relačního Mapování (ORM), tedy tvoří vrstvu mezi aplikací a databází, kde jednotlivé entity jsou reprezentovány formou objektů. Další velkou výhodou je již snadná integrace s frameworkem Symfony.

3 Implementace

3.1 Vývojové nástroje

Pro realizaci aplikace bylo využito standardních nástrojů pro vývoj webových aplikací. Mezi tyto nástroje se řadí Composer (správce knihoven a balíků pro jazyk PHP), npm (správce knihoven a balíku pro javascript) a Gulp. Gulp je frontendový nástroj pro kompilaci a minifikaci CSS/JS souborů.

3.1.1 XAMPP Stack

XAMPP je software, který se skládá z několika nejčastěji používaných programů pro chod webového serveru. Je dostupný zdarma pro Windows, Linux i macOS.

Jedná se o zkratku: je multiplatformní (X), obsahuje Apache (A), MariaDB (M), PHP (P) a Perl (P). V této práci zastoupil roli testovacího serveru. K tomuto serveru bylo následně přistupováno pomocí adresy *localhost* (IP adresa 127.0.0.1). Verze použitého programu byla 7.2.11.

3.1.2 Git

Celý systém je verzován pomocí technologie Git a je umístěn na službě Bitbucket. V době návrhu byla služba zvolena, protože umožňovala zdarma vytvářet privátní repozitáře. Z dalších služeb tuto možnost poskytoval pouze GitLab, v současné době ji poskytuje i GitHub.

3.2 Frontend

Frontend popisuje prostředí pro návštěvníka webu. Je zde zahrnuta lokalizace, zobrazování zdrojů nebo manuální výběr jazyka na základě státu, potažmo přímo jazyka.

Aby prostředí dobře vypadalo, bylo využito již existující šablony, která tyto vlastnosti do jisté míry pokrývá. Pro bezplatné použití šablony byl do patičky webu umístěn odkaz na jejího autora.

3.2.1 Automatický výběr jazyka

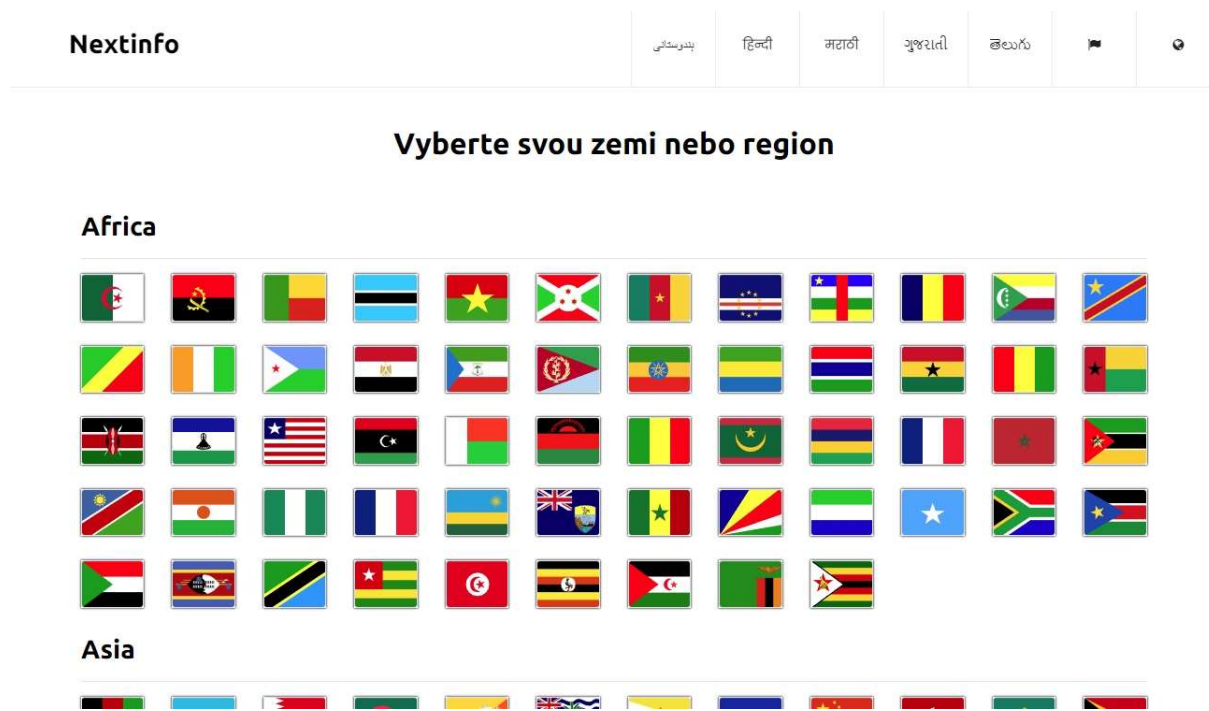
Při první návštěvě webu uživatelem je důležité mu co nejpřesněji nastavit jeho jazyk. To je možné provést několika způsoby. Je možné ho zjistit na základě jazyka prohlížeče nebo

na základě polohy a pokud není možné ani jeden z údajů ověřit, pak se nastaví výchozí jazyk aplikace – tedy český jazyk.

Jazyk prohlížeče je možné najít v http hlavičce *HTTP_ACCEPT_LANGUAGE*. Tento údaj poskytuje koncové zařízení uživatele a jednotlivé jazyky udává v prioritách. Z tohoto textového řetězce je pak pomocí regulárních výrazů možné přechíst jazyk s nejvyšší prioritou a ten nastavit.

Pokud nebylo možné uživateli nastavit jazyk podle prohlížeče, tak je nastaven podle polohy. Pomocí knihovny *geoip2* je možné na základě IP adresy identifikovat polohu uživatele na úrovni státu. Toto řešení není odolné například vůči VPN, díky které může uživatel svou reálnou IP adresu skrýt.

Samozřejmě pokud uživatel není spokojen s vybraným jazykem, nebo chce zjistit, jak web vypadá v jiném jazyce, je možné jazyk vybrat manuálně buď podle vlajky státu (viz obrázek 3.1) nebo přímo podle názvu jazyka.



Obrázek 3.1: Ukázka manuálního výběru jazyka dle státu

3.2.2 GeoIP 2

Pro aplikaci je nutná schopnost co nejpřesněji určit polohu uživatele, který web navštíví. Z toho důvodu byl do aplikace doinstalován balíček *geoip2*. Jedná se o veřejnou databázi

IP adres s co největším množstvím relevantních informací. Například právě stát, město, ISP nebo zeměpisné informace jako zeměpisná délka a šířka.

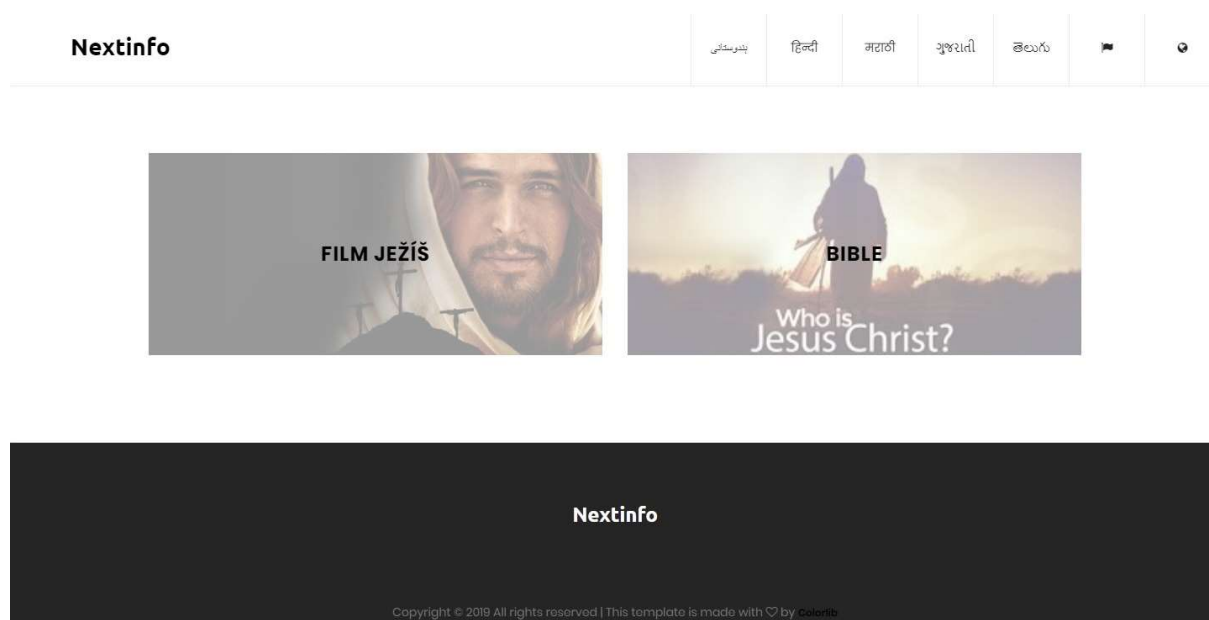
Tento balík je volně dostupný jako knihovna pro integraci s PHP pod Apache licenci 2.0. Zároveň je to následník předchozího GeoIP rozšíření přímo do Apache, které tuto funkcionalitu zajišťovalo.

3.3 Zobrazování zdrojů

Po automatickém nastavení jazyka následuje zobrazení zdrojů. Zdroje je možné omezit na jazyk, stát, typ zařízení nebo zeměpisnou polohu. Jediným kritériem, na které je možné se spolehnout je jazyk. A to z důvodu, že stránka vždy musí být lokalizována.

Zjišťování státu, typu zařízení nebo zeměpisné polohy může selhat. Selhat může třeba proto, že uživatel nepovolí prohlížeči vyčíst informace o poloze nebo je taky možné, že to v první řadě prohlížeč vůbec nepodporuje.

Podobné to je i v případě zjišťování státu na základě IP adresy. IP adresa se nemusí nacházet v databázi. Stejně tak i typ zařízení nemusí být zjištěn dostatečně přesně pro dobré rozpoznání. Hlavně u starších prohlížečů. Příklad zobrazených zdrojů je níže (viz obrázek 3.2).



Obrázek 3.2: Ukázka zobrazení zdrojů

3.3.1 Haversinův vzorec

Při zobrazování zdrojů je možné definovat omezení přesné pozice. Pro toto omezení bylo nutné, aby aplikace byla schopná spočítat vzdálenost mezi dvěma body na zeměkouli. Pro tento případ existuje Haversinův vzorec [6][7] zobrazen níže, který vychází z trigonometrie.

$$d = 2 \cdot r \cdot \arcsin \left(\sqrt{\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cdot \cos(\varphi_2) \cdot \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

V tomto vzorci je φ zeměpisná šířka, λ zeměpisná délka a r je poloměr Země, tedy 6 371 km. Zároveň vzorec není úplně přesný, protože počítá vzdálenost tzv. vzdušnou čarou, nerespektuje charakter terénu a Země není dokonalá koule.

3.4 Administrace

Administrační část aplikace poskytuje správci efektivní a intuitivní nástroje pro správu aplikace. Může spravovat zdroje, jazyky (viz obrázek 3.3), státy a uživatele. Zároveň poskytuje víceuživatelské prostředí, kde je možné každému uživateli nastavit jinou úroveň oprávnění.

Kód jazyka (639-2)	Kód jazyka (639-1)	Název jazyka	Název jazyka v něm samém	Směr textu	Náboženství	Aktivní	
zul	zu	Zulu	isiZulu	Zleva doprava	Christianity	<input checked="" type="checkbox"/>	<input type="button" value="✎"/> <input type="button" value="✖"/>
chi	zh	Chinese		Zleva doprava		<input checked="" type="checkbox"/>	<input type="button" value="✎"/> <input type="button" value="✖"/>
zha	za	Zhuang; Chuang	Vahcuengh	Zleva doprava		<input checked="" type="checkbox"/>	<input type="button" value="✎"/> <input type="button" value="✖"/>
yor	yo	Yoruba	Yorùbá	Zleva doprava	Christianity	<input checked="" type="checkbox"/>	<input type="button" value="✎"/> <input type="button" value="✖"/>
yid	yi	Yiddish	ייִדיש	Zleva doprava		<input checked="" type="checkbox"/>	<input type="button" value="✎"/> <input type="button" value="✖"/>
xho	xh	Xhosa	isiXhosa	Zleva doprava	Christianity	<input checked="" type="checkbox"/>	<input type="button" value="✎"/> <input type="button" value="✖"/>
wol	wo	Wolof	Wolof	Zleva doprava	Islam	<input checked="" type="checkbox"/>	<input type="button" value="✎"/> <input type="button" value="✖"/>

Obrázek 3.3: Ukázka administrace jazyků

3.4.1 Validování zdrojů

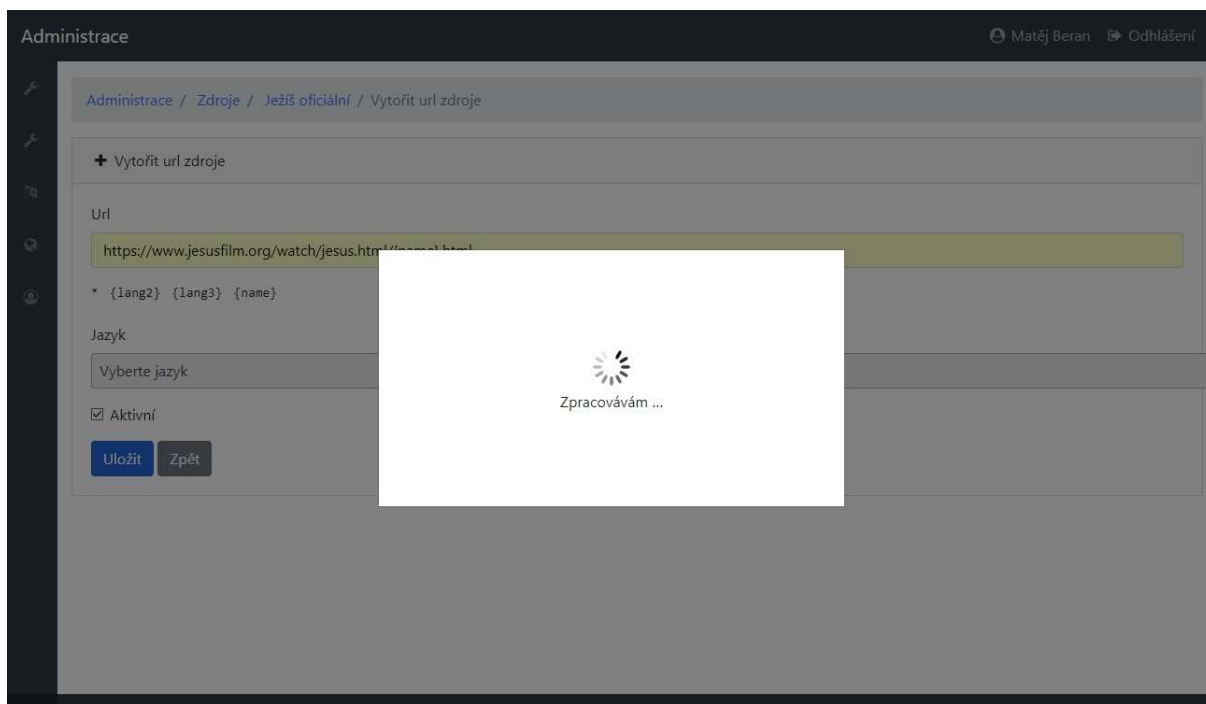
Součástí správy zdrojů je i jejich validace. Validace se děje na dvou místech. Při samotném vytváření zdrojů a poté v pravidelném intervalu. Z datového modelu (viz Příloha I) je zřejmé, že zdroj se skládá z několika URL, kde každá URL je v jiném jazyce.

Validace potom probíhá na tom principu, že se pošle GET požadavek na danou URL. Pokud požadavek vrátí kód 200, považuje se za validní. Zároveň byla omezena doba čekání na odpověď ze 30 s na 5 s z důvodu časové náročnosti.

3.4.2 Poloautomatizovaná lokalizace zdrojů

Protože každá URL zdroje je lokalizována, tak ruční zadávání by trvalo velmi dlouho a bylo by nepraktické. Z toho důvodu byly navrženy tzv. „divoké karty“. To je speciální řetězec, který se při cyklickém zpracování nahrazuje jazykovým kódem nebo anglickým názvem jazyka.

Při generování velkého množství URL nastává problém s časovým omezením na straně serveru. Obvykle to je 30 s. Tento problém byl vyřešen asynchronním zpracováním (viz obrázek 3.4) v blocích po 20. Nevýhoda je dlouhé zpracování, i v řádech desítek minut.



Obrázek 3.4: Ukázka generování URL zdroje

3.4.3 Plánovač úloh

Může se stát, že URL časem přestane existovat. Proto byl implementován plánovač úloh, který každých 60 minut spouští skript, který validuje celkem 20 zdrojů. Validuje jich právě 20, protože opět nastává stejný problém s délkou vykonávání skriptu. U každého URL se tedy vede datum poslední validace a při spuštění skriptu se všechny URL seřadí podle tohoto data vzestupně a vezme se prvních 20 výsledků.

4 Testování

Automatizované testování softwaru je jedna z moderních metodik, jak udržet zdravý kód, tedy efektivní a dobře čitelný s co nejmenším počtem chyb. Protože i když můžete mít existující kód plně otestovaný a funkční, tak každý nový řádek kódu může tuto funkčnost narušit. A poté je velmi náročné znovu testovat kód, který jsme testovali předtím.

Samozřejmě není možné pokrýt všechny stavy těmito testy, ale je dobré je implementovat minimálně pro triviální úlohy, které se často opakují. Zbytek už pak možné otestovat manuálně.

4.1 Import dat

Pro importování základních dat bylo využito dat ze služby *joshuaproject.net*. Služba poskytuje API, kde jsou data o státech, jazycích a národnostních skupinách. Pro používání služby je nutné si pouze nechat vygenerovat API klíč, který služba na žádost vygeneruje. Při žádosti je potřeba uvést způsob užití.

Import je realizován PHP skriptem, který komunikuje s API. Komunikace se děje pomocí GET požadavků na URL poskytnuté službou. Data jsou službou poskytována ve formátu JSON, který je jednoduchý na zpracování.

Importování těchto dat je důležité z hlediska testování, protože to umožní realizovat testy nad reálnou strukturou i velikostí dat.

4.2 PHPUnit

PHPUnit je nezávislá knihovna napsaná v jazyce PHP, která poskytuje robustní framework pro testování. Symfony tuto knihovnu integruje přes návrhový vzor *Bridge*, který vlastně pouze přizpůsobuje původní implementaci pro prostředí Symfony. Před použitím bylo tedy nutné tuto komponentu do aplikace doinstalovat.

Všechny testy jsou soustředěny ve složce *tests* a kopírují hlavní adresářovou strukturu, kde jsou zdrojové kódy aplikace (*src*). Pro následné spuštění testů je pak potřeba spustit příkaz *php bin/phpunit*.

4.2.1 Jednotkové testování

Jednotkové testy jsou určeny k testování implementace daného systému. Jednotkou je pak myšlen jeden samostatně testovatelný blok v rámci aplikace. Z pohledu procedurálního

programování pak jednotka může představovat metodu, proceduru nebo proměnnou. Naopak v objektově orientovaném programování jednotka představuje obvykle třídu nebo konkrétní metodu.

4.2.2 Funkční testování

Funkční testy se zaměřují na to, jak má aplikace fungovat v jednotlivých bodech používání aplikace. Zákazník nebo zadavatel aplikace definuje způsob, jakým bude aplikace používána. Tyto požadavky jsou obvykle pak zpracovány do jednotlivých způsobů použití (Use-Case), ze kterých právě tyto testy vycházejí. Z pohledu knihovny PHPUnit se tyto testy nijak neliší.

V případě této aplikace testy slouží primárně pro testování administračního rozhraní aplikace a v druhé řadě pro základní otestování stránek frontendu. Pro toto testování je vhodné doinstalovat další dvě komponenty.

První komponentou je BrowserKit, která simuluje chování webového prohlížeče. To umožňuje odesílat požadavky nebo programově odesílat formuláře. Praktický důsledek je možná simulace chování uživatele.

Druhou komponentou je pak CssSelector poskytující css selektory v jazyce PHP. Díky tomu je pak možné kontrolovat přítomnost různých HTML značek na testovaných případech.

4.3 Statická analýza

Statická analýza kódu se soustředí na hledání chyb v kódu, aniž by se daný kód spouštěl. Pro tuto analýzu byla využita knihovna PHPStan, která pro spuštění vyžaduje kód, který je napsaný ve verzi PHP 7.1 a vyšší, typovaný a objektově orientovaný.

Pro analýzu je možné definovat několik úrovní striktnosti, kde úroveň 0 je velmi základní a úroveň 7 naopak velice striktní. V rámci implementace aplikace vždy bylo cíleno na to, aby kód splňoval co nejvyšší úroveň. Pro bezproblémový chod aplikace postačila úroveň 6.

5 Nasazení

Pro nasazení aplikace byl poskytnut hosting na serverech služby WEDOS. Tento hosting disponuje PHP 7.3 a MySQL databází. Vše tedy koresponduje s požadavky aplikace.

5.1 FTP

Jedním ze způsobů, jak nasadit výslednou aplikaci je prostřednictvím FTP (FTPS) protokolu. Jedná se pouze o přenos souborů mezi dvěma stroji. Jeho výhodou je pak jednoduchost použití.

Tento způsob nasazování není efektivní, protože má delší odezvu. Zároveň v případě aktualizace aplikace není vždy možné nahradit jen několik souborů se změnou, pak je nutné nahrát celý obsah projektu znova. To přináší i delší odezvu pro programátora, protože není možné zjistit chybu, dokud se nenakopíruje celý obsah projektu na server.

Ve výsledku byl zvolen tento způsob nasazení, protože poskytnutý hosting níže uvedený způsob nepodporuje.

5.2 SSH

Další možností je použít vzdálený přístup přes SSH. Tento způsob dává větší flexibilitu a lépe automatizuje celý proces nasazování aplikace. Obzvlášť, pokud je aplikace verzována nástrojem jako je například Git.

Protože se při přístupu na server jedná pouze o vykonání několika za sebou jdoucích příkazů, je jednoduché proces automatizovat. Stačí implementovat skript v Shellu, který ve správném pořadí za sebou provede jednotlivé příkazy. Pro zvýšení robustnosti je pak možné kontrolovat přítomnost nástrojů, které skript ke svému vykonání potřebuje (například Git nebo Composer).

Následná aktualizace je rovněž snadná, protože stačí si stáhnout novou verzi aplikace z repozitáře (Git) a poté projít velmi podobný proces jako při samotném nasazování. Zde je též na místě implementovat skript, který provede aktualizaci automaticky.

Nevýhodou tohoto způsobu nasazování je pak právě použití nástrojů jako je Git. Ne každý umí tyto nástroje ovládat, nebo nemá technologické zázemí pro jejich použití nebo dokonce ani neví o jejich existenci. V existujících projektech je pak velmi náročné na tyto nástroje přecházet.

6 Závěr

Na základě požadavků bylo vyvinuto webové řešení, které návštěvníkům zpřístupní externí informační zdroje v závislosti na jejich zvoleném jazyce, geolokaci a typu zařízení. Jazyk si uživatel může zvolit sám buď podle názvu jazyka nebo podle státu.

Při implementaci rozhodovací logiky se jako první problém projevila knihovna `geopip2`. V předchozích verzích se nejednalo o PHP knihovnu, ale o rozšíření do Apache. Tato změna nastala v průběhu vypracování práce, takže bylo nutné kompletně změnit přístup pro zjišťování informací na základě IP adresy.

Dále bylo implementováno administrační rozhraní, které umožňuje správcům tuto aplikaci spravovat. V administraci je možné spravovat zdroje a jejich URL, jazyky a státy. Uživateli s maximálním oprávněním umožňuje správu ostatních uživatelů. Hned po přihlášení je uživatel přesměrován na stránku s přehledem zdrojů, kde vidí, kolik jich je validních. V případě, že některé validní nejsou, tak jsou detailně pod sebou vypsané i s odkazem na daný zdroj pro manuální inspekci problému. Po inspekci má správce možnost zdroj označit jako validní nebo jej deaktivovat.

Validitu těchto zdrojů zajišťuje plánovač úloh, který každých 60 minut validuje 20 zdrojů. Validace jako taková byla při mé práci jedním z problémů, které bylo nutné vyřešit. Problémem je objem dat, které server musí zpracovat za 30 sekund. Validace jako taková funguje na jednoduchém principu, a to že se zkontroluje kód odpovědi požadavku na URL. Pokud vrací kód 200, tak je vše v pořádku, ale zkontrolovat jednu URL může trvat až 30 sekund. V implementovaném kódu to bylo omezeno na 5 sekund, ale i to je stále hodně. Tento problém byl tedy vyřešen tím, že každá URL zdroje má čas poslední validace, podle kterého se vybírá 20 zdrojů pro validaci.

Při implementování aplikace se také osvědčil výběr frameworku. Díky výborné dokumentaci s příklady použití nebylo nutné se pozastavovat nad trivialitami nebo naopak framework obcházet.

Řešení je demonstrováno na křesťanských zdrojích (film Ježíš, Bible), protože mají velkou výhodu – a to že jsou dostupné ve velkém množství jazyků.

6.1 Další možnosti rozvoje

Dle zadání bylo vytvořeno webové řešení. V případě implementace například mobilní aplikace, která by mohla rozšiřovat funkčnost webu, by mohlo být implementováno obecné API, které by sloužilo pouze pro zprostředkování dat pro jinou aplikaci.

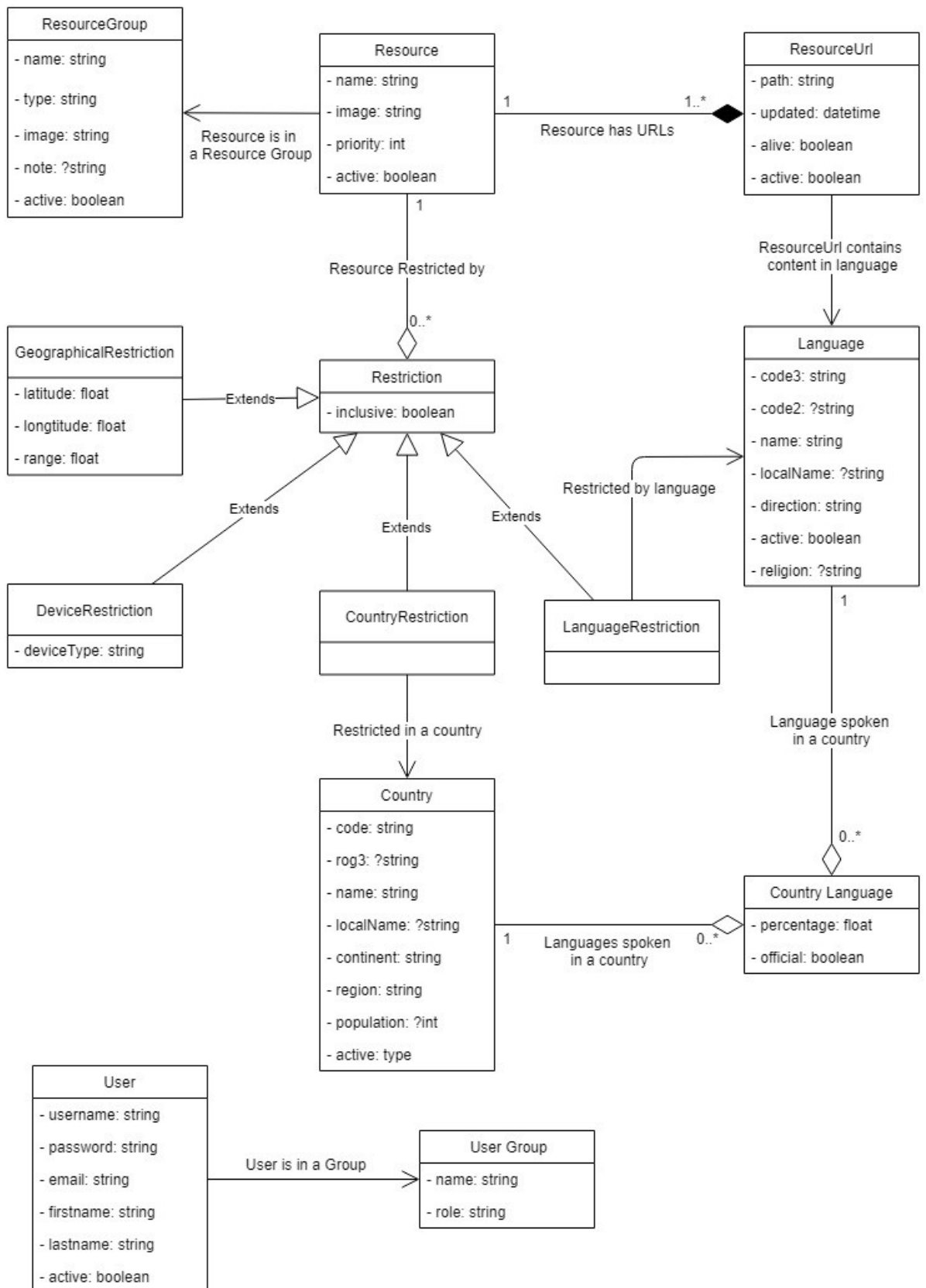
V případě například implementace blogu by pak bylo možné abstrahovat logiku pro zobrazování zdrojů. Poté by bylo možné jednotlivé příspěvky na blogu zobrazovat stejně jako zdroje v závislosti na geolokaci, jazyce a typu zařízení.

Zároveň aplikace obsahuje určité aspekty CMS, takže by bylo možné například do-programovat modul pro články a modul pro správu jednotlivých stránek aplikace.

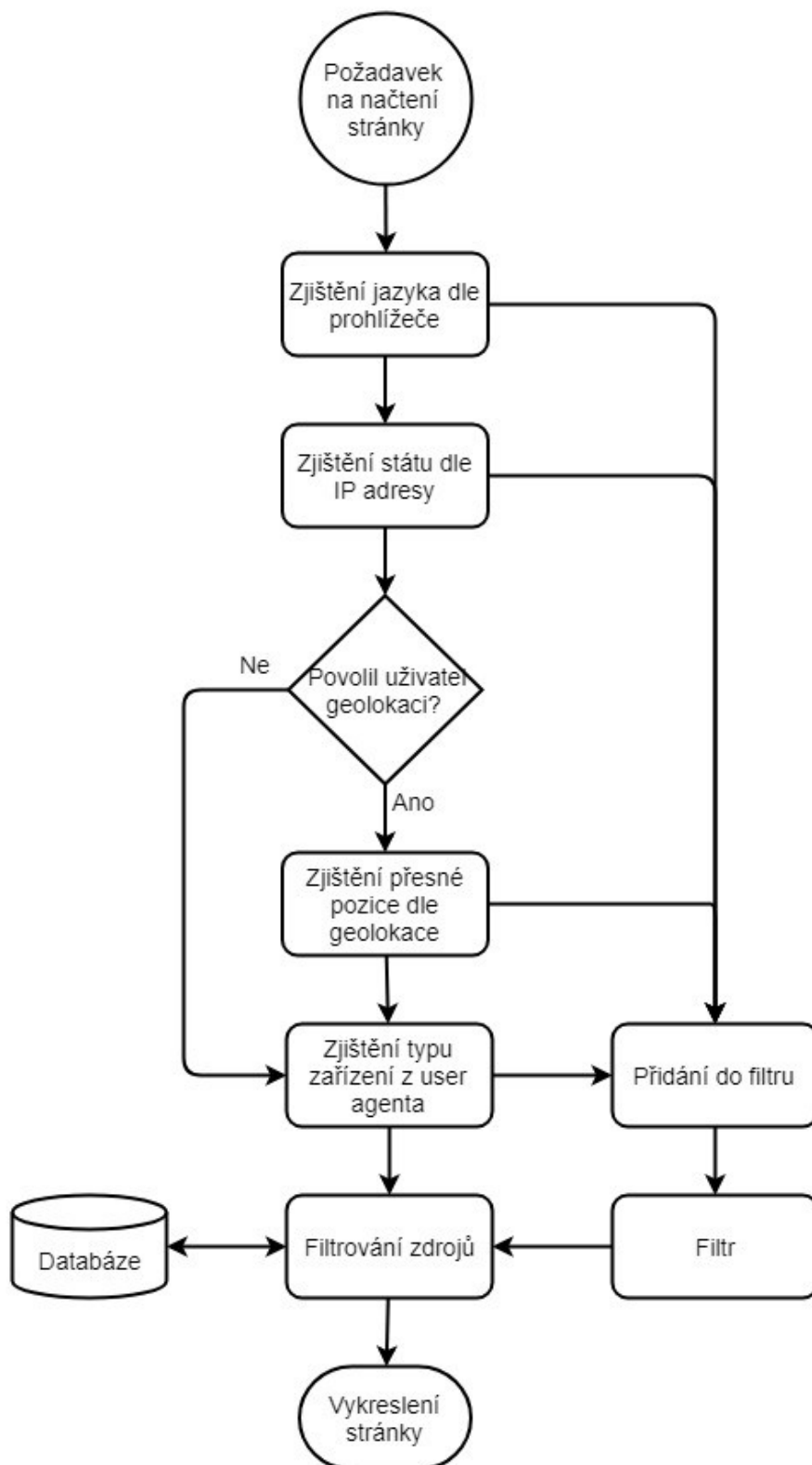
Použitá literatura

- [1] Kódy pro názvy jazyků: ČSN ISO 639 (01 0182). Praha: Český normalizační institut, 2000. ČSN ISO, 639-2 (01 0182).
- [2] ISO 639-1 – Wikipedie. [online]. [cit. 2019-04-27]. Dostupné z: https://cs.wikipedia.org/wiki/ISO_639-1
- [3] ISO 639-3. SIL International 2017 [online]. [cit. 2019-04-27]. Dostupné z: <https://iso639-3.sil.org>
- [4] ISO 3166 Country Codes. [online]. [cit. 2019-04-27]. Copyright © All Rights Reserved [cit. 21.04.2019]. Dostupné z: <https://www.iso.org/iso-3166-country-codes.html>
- [5] About Symfony Project. *Symfony, High Performance PHP Framework for Web Development* [online]. [cit. 2019-04-27]. Dostupné z: <https://symfony.com/about>
- [6] Haversine formula to find distance between two points on a sphere - GeeksforGeeks. GeeksforGeeks | A computer science portal for geeks [online]. [cit. 2019-04-27]. Dostupné z: <https://www.geeksforgeeks.org/haversine-formula-to-find-distance-between-two-points-on-a-sphere/>
- [7] Haversine formula - Wikipedia. [online]. [cit. 2019-04-27]. Dostupné z: https://en.wikipedia.org/wiki/Haversine_formula

Příloha I – Datový model



Příloha II – Logika zobrazování zdrojů



Příloha III – obsah přiloženého DVD

- Text bakalářské práce
 - Matěj Beran – Bakalářské práce.pdf
 - Matěj Beran – Zadání bakalářské práce.pdf
- Zdrojové kódy
 - Aplikace (napsáno v programu PhpStorm 2019.1.1)