

WAP - Internetové aplikace

Webový frontend využívající dostupná otevřená data

Covid bez bariér: covidbezbarier.cz

1 Úvod

Řada odporníků se shoduje, že klíčový faktor pro zvládnutí pandemie Covid-19 je dostatečné proočkování populace. Je proto důležité, aby bylo očkování dostupné pro všechny, kteří o něj mají zájem. Pro lidi s omezenou možností pohybu však může být nalezení vhodného očkovacího místa ztíženo tím, že ne všechna tato místa mají bezbariérový přístup.

Tento projekt se zaměřuje na specifickou skupinu obyvatel, kteří k možnosti otestovat se či přijmout vakcínu vyžadují bezbariérový přístup k místu testování či aplikace vakcíny.

2 Návrh

Fáze návrhu se skládala ze dvou částí. První byla nalezení vhodných, veřejně dostupných dat, které by bylo možné v aplikaci vizualizovat. Druhou částí pak byl návrh samotného uživatelského rozhraní pro aplikaci.

2.1 Uživatelské rozhraní

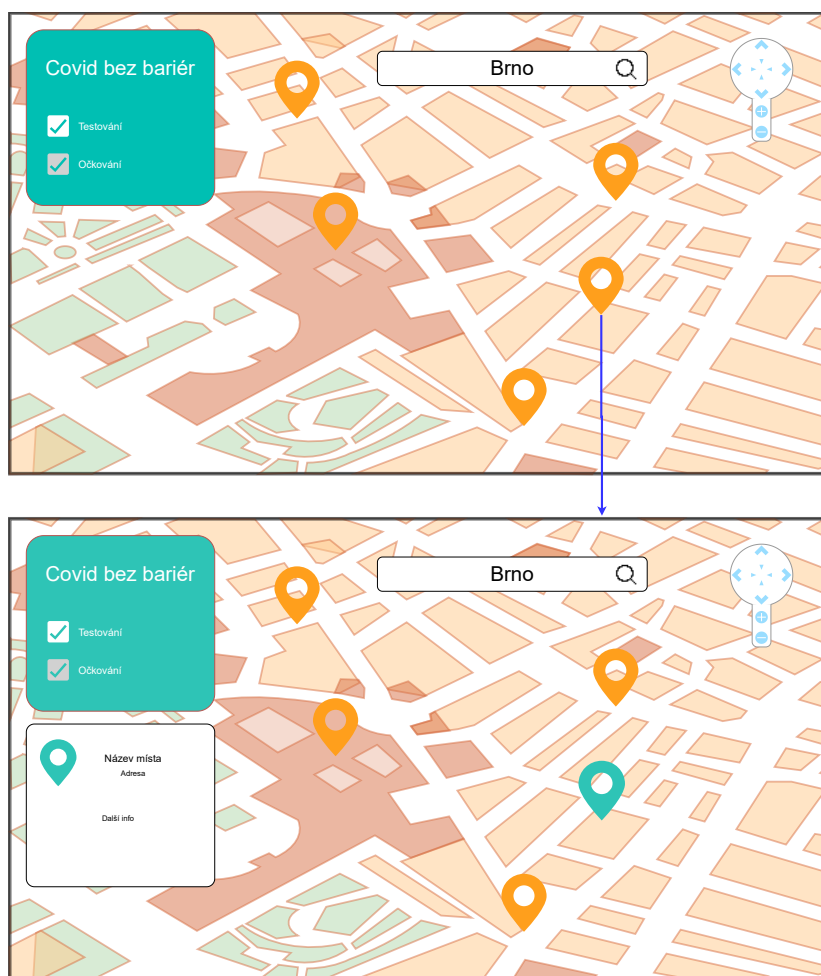
Během návrhu uživatelského rozhraní jsme se zaměřili především na jednoduchost, přehlednost a responzivitu. Návrhy uživatelského jsou zobrazené na obrázcích 1 a 2.

Dominantním prvkem uživatelského rozhraní je mapa, na které se zobrazují dostupná očkovací místa. Navigování na mapě je dostupné třemi možnými způsoby. Prvním z nich je klasické manuální tažení. Takhle si uživatel může například najít vlastní cestu, či sám zhodnotit, která z dostupných očkovacích míst mu vyhovují nejvíce. Druhou možností je využití vyhledávacího pole na horním okraji mapy. Pomocí tohoto vyhledávacího pole je možné nalézt konkrétní adresu přičemž se na mapě zobrazí všechna vyhovující očkovací místa v okolí dané adresy. Posledním způsobem je možnost využití lokalizačních služeb, k automatickému zjištění polohy uživatele a zobrazení vyhovujících očkovacích míst v jeho okolí.

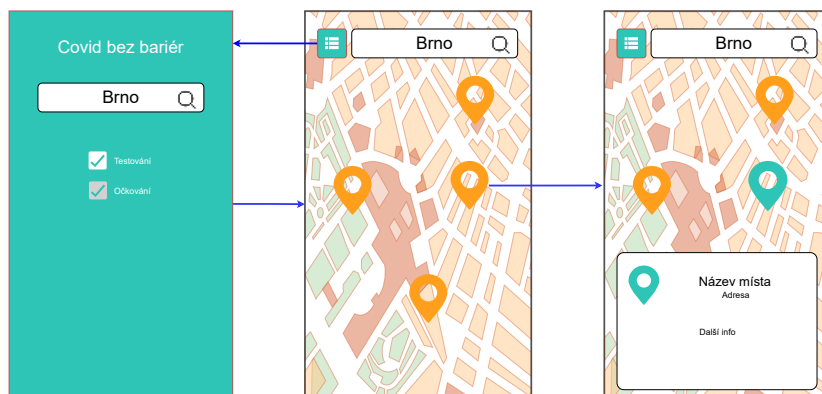
Do návrhu rozhraní samozřejmě vstoupilo také to, že velká většina uživatelů používá k prohlížení webových stránek a aplikací své mobilní zařízení. Ačkoliv mapa jako dominantní prvek rozhraní zůstává prakticky nezměněná, uzpůsobili jsme ostatní komponenty rozhraní tak, aby se aplikace pohodlně používala jak v prohlížeči na běžném počítači (rozhraní vyobrazeno na obrázku 1), tak na mobilním zařízení (rozhraní vyobrazeno na obrázku 2).

Prvním takto přizpůsobujícím se prvkem rozhraní je panel s nastavením zobrazování očkovacích míst. Ta obsahuje ovládací prvky, který je možné filtrovat zobrazené výsledky. V případě zobrazení na počítači je tento panel viditelná v rohu mapy. V případě zobrazení na mobilním zařízení, kde není k dispozici tolik zobrazovacího prostoru, se oddíl chová jako samostatný panel, který se zobrazí po zmáčknutí tlačítka vedle vyhledávacího pole.

V neposlední řadě se na uživatelského rozhraní nachází karta obsahující detaily konkrétního očkovacího místa. Tato karta standardně viditelná není, dokud není vybráné nějaké očkovací místo. Jakmile uživatel na mapě místo vybere, karta s detailem daného místa se v případě rozhraní na počítači zobrazí pod kartou s nastavením vyhledávání. V případě mobilního zařízení se tato karta zobrazí na spodní straně obrazovky. Na samotném oddílu se zobrazuje především



Obrázek 1: Návrh uživatelského rozhraní pro web



Obrázek 2: Návrh uživatelského rozhraní pro mobilní zařízení

název vybraného očkovacího místa, jeho adresa a další informace, které je možné získat z dostupných dat.

2.2 Dostupná data

Ministerstvo zdravotnictví České republiky poskytuje na portálu, dedikovaném probíhající pandemii [1], kvalitní veřejně dostupná data o všem, týkajícím se pandemie. Tato data jsou poskytována ve formátech CSV a JSON.

Pro naši aplikaci jsou významná data o očkovacích a testovacích místech – terminologie se rámci veřejně dostupných dat neshoduje a používá se v kontextu odběrového místa i pojmenování „testovací“; pro účely tohoto projektu vyjadřují tyto pojmy identickou službu a jsou vzájemně zaměnitelné. Vzhledem k tomu, že vyvíjíme webovou aplikaci, rozhodli jsme se použít data ve formátu JSON – ty mají následující strukturu:

```
{
  ockovaci_misto_id: string
  ockovaci_misto_nazev: string
  okres_nuts_kod: string
  operacni_status: boolean
  ockovaci_misto_adresa: string
  latitude: string
  longitude: string
  nrpzs_kod: integer
  minimalni_kapacita: integer
  bezbarierovy_pristup: boolean
}
```

Ukázka 1: Struktura informací dostupných pro očkovací místa – důležitý pro kontext tohoto projektu je především atribut `bezbarierovy_pristup`.

```

{
  odberove_misto_id: string
  odberove_misto_nazev: string
  okres_nuts_kod: string
  operacni_status: boolean
  odberove_misto_adresa: string
  latitude: string
  longitude: string
  testovaci_kapacita: integer
  nasofaryngealni_odber: boolean
  orofaryngealni_odber: boolean
  antigenni_odber: boolean
  drive_in: boolean
}

```

Ukázka 2: Struktura informací dostupných pro testovací místa – důležitý pro kontext tohoto projektu je především atribut `drive_in`.

Jak je ve schématech vidět, informace o tom, zda je očkovací místo bezbariérově přístupné je v datech obsažená – u vakcinačních míst v ukázce 1 se jedná o atribut `bezbarierovy_pristup`. Pro odběrová místa (ukázka 2) je situace poněkud složitější, neobsahují totiž explicitní příznak bezbariérovosti konkrétního místa – pro použití aplikací tedy postačil atribut `drive_in`. Mimo to obsahují data další informace o daném místě a v poslední řadě také jeho souřadnice, podle kterých je možné dané místo zobrazit na mapě.

3 Implementace

Projekt je implementačně postaven jako aplikace běžící v prohlížeči s naprostou většinou funkcionalit implementovaných přímo v prohlížeči, tedy bez signifikantního přispění serverových technologií. Jako hlavní platforma pro realizaci byl zvolen nástroj Nuxt.js, který integruje techniku SSR¹ společně s frameworkem Vue.js.

Pro převážnou část projektu byl zvolen jako implementační jazyk TypeScript (především pro jeho typovou bezpečnost) – definice stavu aplikace i funkcionalit komponent tedy obsahuje typové definice.

3.1 Nuxt.js

Jako hlavní nástroj pro implementaci projektu byl zvolen projekt Nuxt.js, který realizuje SSR pro Vue.js a zároveň stanovuje standardní adresářovou strukturu pro projekty – tedy cíleně definuje strukturu pro routování stránek, asynchronní načítání komponent, umístění

¹SSR, Server-side rendering je technika používaná frameworky v rodině jazyků Javascript, která umožňuje vykreslení výsledného HTML kódu již na straně serveru (v komponentách, kde je to možné) pro optimalizaci vykreslování až na klientské straně – před-renderovaný HTML kód je poté hydratován a jsou na něj dodatečně navázány obsluhy událostí.

dodatečných definic např. kaskádových stylů či definice vnitřního stavu aplikace. Volba na tento nástroj padla na základě zkušeností autorů, jeho podpory na platformě Vercel² a (relativně) efektivní výsledné webové stránky z pohledu datové velikosti a náročnosti na čas prohlížeče.

3.1.1 Stránky aplikace

Vzhledem k rozsahu aplikace definují celý projekt dvě hlavní stránky – těmi jsou `pages/about.vue` a `pages/index.vue`. První jmenovaná je prakticky statická stránka informující uživatele o vývojářích projektu, zdrojích datech a odkazovaných externích službách – důležitým prvkem jsou pak data poslední aktualizace zdrojových dat a samotná verze aplikace. Hlavní domovská stránka projektu v souboru `pages/index.vue` poté definuje či deleguje veškerou funkcionalitu projektu.

3.1.2 Mapová komponenta

Pro implementaci komponenty pro zobrazení jednotlivých míst na mapovém podkladu byla zvolena známá knihovna Leaflet, specificky v balíčku uzpůsobeném pro integraci do Nuxt.js (který je nadstavbou balíčku integrující tuto knihovnu interaktivních map do komponent frameworku Vue.js.). Pro integraci mapy do hlavní stránky projektu byla abstrahována vlastní komponenta `components/Map.vue` – ta zajišťuje následující důležité funkce:

- definice hranic pro zobrazení mapy a základního pohledu včetně přiblížení a jeho středu – zajištěno přes veřejné rozhraní komponenty `<l-map />`
- definice serveru pro získání mapových podkladů pomocí komponenty `<l-tile-layer />` – využity veřejně dostupné podklady ze serveru `osm.org`
- vložení hranice okresů komponentou `<l-geo-json />` – během kompilace projektu dojde k načtení datového souboru `nuts.json`, který obsahuje geografická data o okresech v České republice, které jsou následně vykresleny do výsledné mapy
- shluky značek pro testovací a vakcinační místa je zajištěn pomocí komponenty `<l-marker-cluster />`, která prakticky bez konfigurace shlukuje větší množství značek do popisů s jejich počtem – což snižuje náročnost výkonu v oblasti prvního vykreslení stránky a taktéž zpřehledňuje mapu pro uživatele
- značky pro jednotlivá místa zajišťují instance komponent `<l-marker />` – pro každé místo existuje jedna instance této komponenty s geografickou pozicí odpovídající reálnému umístění – na tyto komponenty je též zaregistrována obsluha zobrazení detailu konkrétního místa při kliknutí na ně

²Vercel je volně dostupná služba určená pro nasazování statických aplikací bez potřeby konkrétní implementace serveru – podporuje širokou škálu frameworků od čistě statických stránek, přes standardní generátory stránek typu Jekyll či Hugo, až k SSR, tedy frameworkům typu Next.js či Nuxt.js. Poskytuje taktéž integraci do platformy Github a nasazování jednotlivých instancí aplikace může být přímo svázáno s konkrétní Git větví.

- vlastní komponenta `<PlaceDetail />` zajišťuje zobrazení okna s detailem konkrétního místa – to zahrnuje název, adresu místa (s odkazem na vyhledání ve službě Google Maps), informační značky o poskytovaných službách, případné upozornění o aktuální ne-operabilitě místa či odkaz na web <https://ockovani.opendatalab.cz> poskytující data statistiky o očkování na konkrétních místech.

3.1.3 Layout aplikace

Vzhledem k automatickému routování frameworkem, je v souboru `layouts/default.vue` uložena výchozí komponenta pro renderování stránky – ta obsahuje základní definici struktury s nadpisem aplikace a komponentou pro vykreslení obsahu konkrétních stránek.

3.1.4 Vnitřní stav aplikace

Pro uchování stavu aplikace na jednom místě je použit framework Vuex – ten rozděluje definici stavu na samotný objekt stavu, synchronní mutace, které jej mění, a potenciálně asynchronní akce, které zapouzdřují větší logické celky jejichž výsledkem mohou být volání mutací.

Pro účely projektu byl stav rozdělen do následujících Vuex modulů (v adresáři `store/`):

- `places.ts` – obsahově nejdůležitější modul zahrnuje načtená očkovací a testovací místa, stejně tak jako stav jejich poslední modifikace a filtrů, a také vybrané místo do detailu. Dvě důležité akce poté vystavuje jako své rozhraní, `loadTestingPlaces`, resp. `loadVaccinationPlaces` zajišťují asynchronní načtení dat ze zdroje a následné jejich uložení do stavu. Dvojice mutací `set*PlacesResponse` filtrují odpověď z API na výhradně bezbariérová místa a následně z objektů obecného typu konstruují objekty typu `TestingPlace`, resp. `VaccinationPlace`.
- `layout.ts` – definuje jedinou datovou proměnnou a tou je příznak, zda-li je hlavička aplikace rozbalená (s odpovídající mutací pro její nastavení)
- `map.ts` – pro účely centrování mapy na pozici uživatele obsahuje tento modul proměnnou pro konkrétní lokaci, stejně tak s mutací tuto lokaci nastavující

3.1.5 Statické zdroje

Adresář `static/` obsahuje statické zdroje určené pro výstupní stránku – tyto zdroje nejsou zpracovávány ani optimalizovány nástrojem Webpack³. Patří mezi ně například obrázky pro Open Graph metadata⁴, ikona stránek či soubor `robots.txt`.

³Webpack je nástroj pro sestavování balíčků primárně založených na jazyku Javascript – běží v pozadí platformy Nuxt.js a zajišťuje transpilaci Typescript kódu, zpracování kaskádových stylů z kódu pro preprocesory, optimalizaci obrázků a vytvoření hromadného balíčku pro zavedení do prohlížeče.

⁴Open Graph metadata zajišťují distribuci strojově čitelných dat mezi webovými stránkami – prakticky se jedná o HTML tagy v hlavičce definující jaký titulek či obrázek se má použít při zobrazení náhledu na cílenou stránku.

3.1.6 Grafické a další zdroje pro aplikaci

Aplikace používá sadu SVG ikon, jejichž zdrojové soubory jsou uloženy v adresáři [assets/](#) – ten spadá do gesce nástroje Webpack, jednotlivé ikony jsou tak optimalizovány z pohledu jejich datové velikosti. Taktéž tento adresář obsahuje definici hranic okresů v souboru [nuts.json](#), konkrétně ve standardizovaném formátu GeoJSON, který následně umí přímo zpracovat mapová komponenta.

3.1.7 Internacionalizace

Aplikační rozhraní je s pomocí knihovny [nuxt-i18n](#) přeloženo do dvou jazyků, a to konkrétně to angličtiny a češtiny. Jazyk je touto knihovnou při startu aplikace automaticky vybrán dle preferovaného jazyka prohlížeče, mezi jazyky lze ovšem přepínat dynamicky. Překladařové řetězce jsou u každé komponenty uvedeny v bloku `<i18n />` jakožto JSON dokument, překlad do případného dalšího jazyka je tedy otázkou definice samotných přeložených řetězců.

3.2 Tailwind CSS

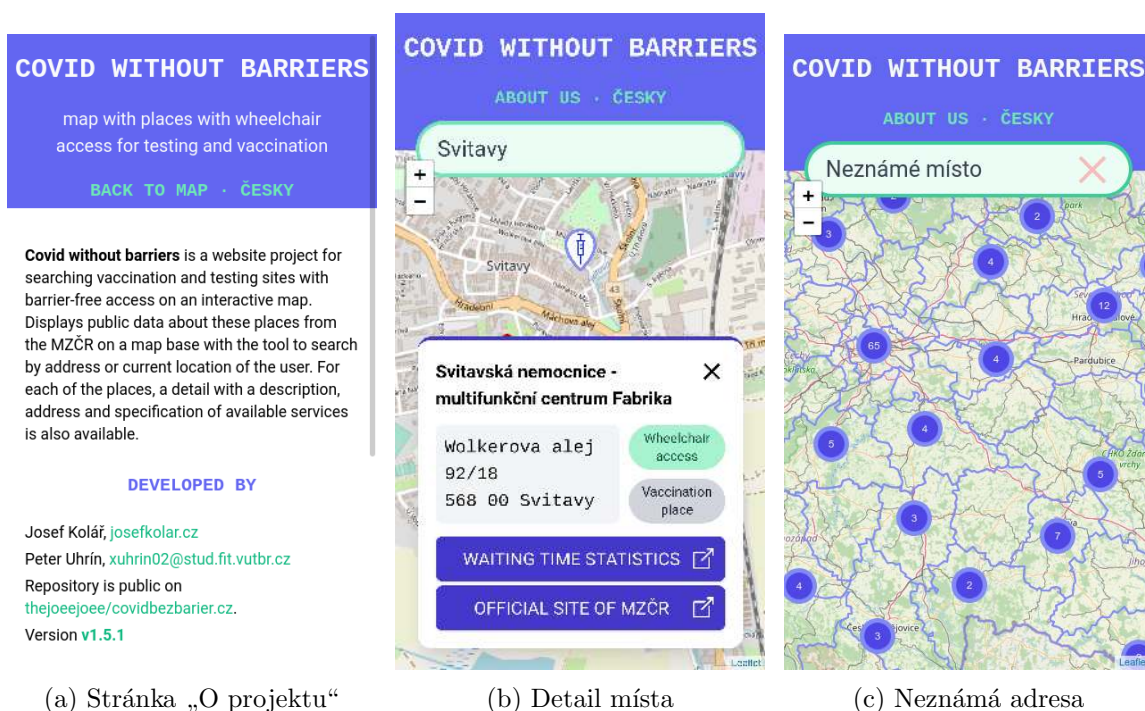
Tailwind CSS je zástupcem CSS frameworků postavených na utilitních CSS třídách – kompletní rozložení prvků, jejich typografie i vizuál aplikace je definovaný explicitně přímo v HTML elementech pomocí tříd z tohoto frameworku. Ten je použit ve výchozí konfiguraci, barevné škály jsou tedy taktéž výchozí, stejně tak jako odsazení a tzv. breakpointy pro responzivní zobrazení. Výhodou kódování pomocí explicitních tříd pro každou CSS vlastnost je možnost optimalizace výsledných kaskádových stylů zpracovávaných prohlížeči – obsahují pouze ta pravidla a třídy, které reálně aplikace používá. Další výhodou je přímo svázaný HTML kód s kaskádovými styly, struktura i vzhled jsou tak definovány společně na jednom místě.

4 Závěr

V rámci projektu se podařilo na otevřených dat MZČR o testovacích a očkovacích místech proti Covid-19 vyvinout aplikaci interaktivně zobrazující bezbariérová místa na mapovém podkladu. Na mapě lze vyhledávat dle adresy či dle lokace uživatele, taktéž je možné zobrazit pouze určitý typ míst – na detailu každého z míst jsou pak zobrazeny bližší informace.

Služba Vercel zajišťuje aplikaci hostování, pro analýzu návštěvnosti a chování uživatelů byl do aplikace nasazen nástroj Google Tag Manager. Pro přístupnost od vyhledávačů a dalších služeb jsou na stránkách dostupné [og:](#) a [tw:](#) tagy s titulkem, popisem a náhledovým obrázkem webu. Výsledný vzhled aplikace lze shlédnout na obrázcích [3](#), [4](#), [5](#) a [6](#).

Aplikace je v době odevzdání projektu nasazena na adrese <https://covidbezbarier.cz/> ve verzi *v1.5.1* a autoři předpokládají snahu ji rozšířit do odpovídající komunity, která využije služby vyhledávání bezbariérových míst.



(a) Stránka „O projektu“

(b) Detail místa

(c) Neznámá adresa

Obrázek 3: Mobilní zobrazení aplikace na 414x716 px odpovídající zobrazení na zařízení *Apple iPhone 6S Plus*.



Obrázek 4: Úvodní stránka při vstupu uživatele informuje o funkci aplikace a zároveň nabízí možnost přímo vyhledávat dle adresy či lokalizovat dle jeho geografické pozice. Ve chvíli, kdy se tak stane, je horní panel upraven na velikost pouze pro titulek a ovládací prvky.



Obrázek 5: Filtrování míst – v aplikaci lze dočasně skrýt jeden z typů zobrazovaných míst.



Obrázek 6: Vyhledávání dle adresy – jednou z možností pro specifikaci místa na mapě je vyhledávání dle adresy, ta je následně lokalizována geograficky přes server <https://nominatim.openstreetmap.org/> a uživateli je zobrazeno odpovídající místo. Další možností je geografická lokalizace uživatele s pomocí prohlížeče a jeho Geolocation API.

Reference

- [1] COVID-19 v ČR: Otevřené datové sady a sady ke stažení: Onemocnění Aktuálně MZČR.
URL <https://onemocneni-aktualne.mzcr.cz/api/v2/covid-19#>
- [2] Leaflet: API reference v1.7.1.
URL <https://leafletjs.com/reference-1.7.1.html>
- [3] MDN Web Docs: Accessibility.
URL <https://developer.mozilla.org/en-US/docs/Learn/Accessibility>
- [4] MDN Web Docs: Geolocation API.
URL https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API
- [5] Nuxt.js: Docs v2.15.2.
URL <https://nuxtjs.org/docs/2.x/get-started/installation>
- [6] The Open Graph protocol.
URL <https://ogp.me/>
- [7] Tailwind CSS: Docs v2.1.0.
URL <https://tailwindcss.com/docs>
- [8] Vercel Documentation.
URL <https://vercel.com/docs>