



Zadání diplomové práce

Název:	Webová aplikace pro interaktivní procházky s využitím QR kódů
Student:	Bc. Štěpán Štrba
Vedoucí:	Ing. Marek Suchánek
Studijní program:	Informatika
Obor / specializace:	Webové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2023/2024

Pokyny pro vypracování

QR kódy se těší velké oblibě pro snadné sdílení odkazů, platebních příkazů, údajů pro připojení k WiFi síti a mnoho dalšího. Současně mnoho organizací a zájmových skupin (muzea, tábory, města, turistické spolky atd.) vytváří informačně-interaktivní procházky často pouze s využitím informačních tabulí, papírů s úkoly a podobně. Takovou procházkou může být průvodce muzeem, ZOO, hradem či parkem, bojovka na táboře s úkoly, vědomostní kvíz po městě či například sdílení fotografií z neobvyklých míst.

- Seznamte se s principy fungování QR kódů a jejich využití v praxi.
- Analyzujte doménu interaktivních procházek a současných přístupů pro realizace. Zohledněte procházky venku (outdoor) a uvnitř budov (indoor). Dále se zaměřte na možnosti gamifikace pro tuto doménu.
- Proveďte stručnou rešerši existujících řešení umožňující snadné plánování interaktivních procházek, případně aplikací využívající QR kódy v podobném duchu.
- Sestavte funkční a nefunkční požadavky na aplikaci dle analýzy a rešerše. Webová aplikace umožní uživatelům vytvářet interaktivní procházky s využitím QR kódů a bude podporovat alespoň tři různé druhy interaktivních prvků pro QR kód.
- Vytvořte návrh architektury a UI webové aplikace. Při návrhu počítejte s tím, že aplikaci musí být možné snadno obohatit o další druhy interaktivních prvků.
- Implementujte aplikaci dle návrhu za použití vybraných technologií.
- Řádně otestujte a zdokumentujte implementaci dle běžné praxe pro zvolené technologie.
- Zhodnoťte vytvořenou aplikaci a navrhněte její další možný rozvoj.



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Diplomová práce

Webová aplikace pro interaktivní procházký s využitím QR kódů

Bc. Štěpán Štrba

Katedra softwarového inženýrství
Vedoucí práce: Ing. Marek Suchánek

4. května 2023

Poděkování

Tímto bych rád poděkoval vedoucímu práce Ing. Markovi Suchánkovi za vedení, pomoc a cenné rady při zpracování této práce. Dále bych chtěl poděkovat svým přátelům a rodině, pro jejich podporu při studiu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu) licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 4. května 2023

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2023 Štěpán Štrba. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Štrba, Štěpán. *Webová aplikace pro interaktivní procházky s využitím QR kódů*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2023.

Abstrakt

Diplomová práce se zabývá návrhem a implementací webové aplikace k tvorbě interaktivních procházek s využitím QR kódů.

Čtenář je nejdříve seznámen s principy fungování QR kódů, současnou realizací interaktivních procházek a možnostmi jejich gamifikace. Dále jsou sestaveny požadavky na aplikaci, které jsou rozřazeny do kategorií podle metody MoSCoW. Podle požadavků je vytvořen návrh architektury a UI aplikace a dle návrhu je aplikace implementována a následně otestována.

Webová aplikace je implementována pomocí Javascriptového frameworku Nuxt.js, PHP frameworku Laravel a několika balíčků.

Klíčová slova interaktivní procházky, Laravel, Nuxt, webová aplikace, QR kódy

Abstract

The thesis deals with the design and implementation of a web application to create interactive walks using QR codes.

The reader is first introduced to the principles of QR codes, the current implementation of interactive walks and the possibilities of their gamification. Next, the application requirements and their classification according to the MoSCoW method are presented. According to the requirements, the architecture and UI design of the application is created and according to the design the application is implemented and then tested.

The web application is implemented using the Javascript framework Nuxt.js, PHP framework Laravel and several packages.

Keywords interactive walks, Laravel, Nuxt, web application, QR codes

Obsah

Úvod	1
1 Cíl práce	3
2 QR kód	5
3 Analýza	7
3.1 Současná realizace interaktivních procházek	7
3.2 Možnosti gamifikace interaktivních procházek	7
3.3 Existující řešení	8
3.3.1 QR Treasure Hunt Generator!	8
3.3.2 Locatify	8
3.3.3 Smart Guide	9
3.3.4 Actionbound	10
3.3.5 Shrnutí	10
3.4 Požadavky	11
3.4.1 Funkční požadavky	11
3.4.2 Nefunkční požadavky	14
3.4.3 MoSCoW	15
4 Návrh	19
4.1 Výběr technologií	19
4.1.1 Frontend	19
4.1.2 Backend	20
4.2 Návrh databáze	20
4.3 Uživatelské rozhraní	21
4.3.1 Admin	22
4.3.2 Klient	27
5 Implementace	31

5.1	Backend	31
5.1.1	Struktura zdrojového kódu	31
5.1.2	Použité dodatečné balíčky	31
5.1.3	Implementované funkční požadavky	32
5.2	Frontend	39
5.2.1	Struktura zdrojového kódu	39
5.2.2	Použité dodatečné balíčky	40
5.2.3	Implementované funkční požadavky	40
6	Testování a nasazení	47
6.1	Uživatelské testování	47
6.2	Admin	48
6.2.1	Scénář	48
6.2.2	Úkoly	48
6.2.3	Otázky	50
6.2.4	Shrnutí	51
6.3	Klient	51
6.3.1	Scénář	51
6.3.2	Úkoly	51
6.3.3	Otázky	51
6.3.4	Shrnutí	52
6.4	Nasazení	52
7	Prototyp	53
7.1	Admin	53
7.2	Klient	56
	Závěr	61
	Literatura	63
	A Seznam použitých zkratek	65
	B Obsah příloženého CD	67

Seznam obrázků

2.1	Struktura QR kódu	5
3.1	Vytváření úkolů v Locatify	9
3.2	Vytvořená otázka v Actionbound	10
3.3	Mobilní aplikace Actionbound	11
4.1	Návrh databáze	21
4.2	Návrh organizace procházek	22
4.3	Návrh editoru procházky	23
4.4	Návrh přidání překladu	23
4.5	Návrh statistik procházky	24
4.6	Návrh akcí v editoru procházky	25
4.7	Návrh organizace interaktivních prvků	25
4.8	Návrh editoru interaktivních prvků	26
4.9	Návrh čtečky QR kódů	27
4.10	Návrh zobrazení otázky s možnostmi True/False	28
4.11	Návrh zobrazení otázky s omezením času	29
4.12	Návrh přehledu a detailu procházek	30
4.13	Návrh profilu	30
5.1	Zjednodušená struktura backend části aplikace	32
5.2	Struktura uložení souborů na Amazon S3	38
5.3	Struktura frontend části aplikace	39
7.1	Organizace procházek	53
7.2	Editor procházek	54
7.3	Organizace interaktivních prvků	55
7.4	Zobrazení interaktivních prvků	55
7.5	Přehled a detail procházek	56
7.6	Profil	57
7.7	Statistiky procházky	57

7.8	Akce v editoru procházky	58
7.9	Editor interaktivních prvků	59

Seznam ukázek kódu

1	Příklad využití balíčku <code>Sanctum</code> pro zamezení přístupu k API endpointu.	34
2	Autorizace uživatele při duplikaci procházky	34
3	Validace dat při ukládání procházky	35
4	Import jazyků do seznamu dostupných jazyků aplikace	36
5	Oprava funkce <code>isEmptyTranslatableAttribute</code> v balíčku <code>Astrolonomic/laravel-translatable</code>	37
6	Zjednodušené použití balíčku <code>Google/cloud-translate</code>	38
7	Zjednodušená funkce pro smazání souborů při mazání procházky	39
8	Registrace uživatele	41
9	Funkce k zobrazení notifikací	41
10	Úprava chování záložky při použití komponenty <code>Tabs</code>	42
11	Přesměrování z krátké URL pro prvky na správnou URL	43
12	Část funkce pro kontrolu obsahu načteného QR kódu	44

Úvod

QR kódy byly vytvořeny pro použití na štítky k automobilovým dílům, ale v dnešní době se velmi často objevují v běžném životě. Využívají se například ke sdílení odkazů, údajů k bankovním převodům či údajů pro připojení k WiFi.

Současně se vytváří stále více interaktivních procházek k seznámení návštěvníka s určitým místem. Takové procházky většinou obsahují otázky a úkoly, které má návštěvník splnit. Úkoly a otázky se většinou tisknou na papír nebo se vytváří informační tabule.

V této diplomové práci se budu věnovat vytvoření webové aplikace pro tvorbu interaktivních procházek s využitím QR kódů. Interaktivní procházky vytvořené v této aplikaci by měly umožňovat větší interaktivitu a zábavu pro návštěvníky a zároveň zjednodušit jejich tvorbu.

V první kapitole stanovím cíle, které budu chtít při plnění diplomové práce splnit. Ve druhé kapitole seznámím čtenáře s principy fungování QR kódů a jejich využití v praxi. V následující kapitole provedu analýzu současných realizací interaktivních procházek a možností jejich gamifikace. Dále provedu rešerši již existujících webových aplikací pro tvorbu procházek. Z analýzy současných realizací, možností gamifikace a již existujících webových aplikací sestavím funkční a nefunkční požadavky na aplikaci. Funkčním požadavkům určím priority jejich rozřazením do čtyř kategorií dle metody MoSCoW.

Ve čtvrté kapitole představím návrh aplikace včetně popisu zvolených technologií, návrhu databáze a návrhu uživatelského rozhraní. V páté kapitole popíši implementaci prototypu aplikace a v následující kapitole představím testování a nasazení aplikace. V sedmé kapitole popíšu implementovaný prototyp aplikace a následně v závěru zhodnotím vytvořený prototyp a navrhnou další možný rozvoj.

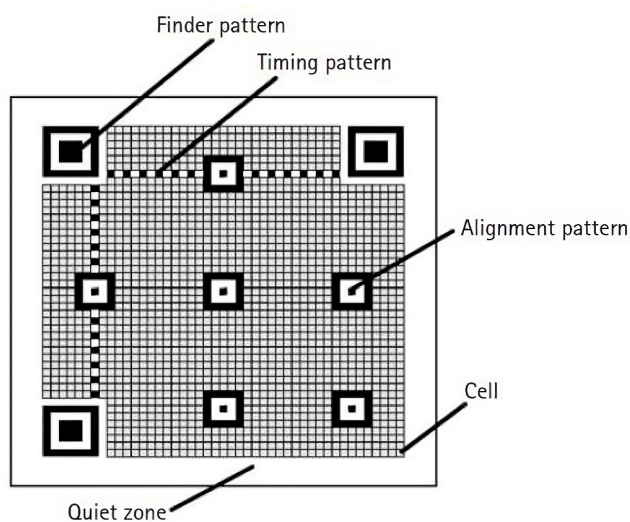
Cíl práce

V této diplomové práci je hlavním cílem vytvoření webové aplikace k tvorbě interaktivních procházek s využitím QR kódů a podporou alespoň tří různých druhů interaktivních prvků pro QR kód. Prvním dílčím cílem je seznámení s principy fungování QR kódů a jejich využitím v praxi. Následující dílčí cíl je analýza současných přístupů k realizaci interaktivních procházek a analýza možností jejich gamifikace. Třetím dílčím cílem je provedení stručné rešerše již existujících webových aplikací pro tvorbu procházek. Čtvrtým dílčím cílem je sestavení funkčních a nefunkčních požadavků dle analýzy a rešerše. Z těchto požadavků je dále nutné vytvořit návrh UI webové aplikace a její architektury, která bude lehce rozšiřitelná o další druhy interaktivních prvků. Posledními dílčími cíli je implementace aplikace a její následné otestování.

QR kód

QR kód [1] byl vyvinut v roce 1994 společností Denso pro použití na štítky k automobilovým dílům, ale v dnešní době se využívá v mnoha jiných odvětvích.

QR kód se skládá z černých a bílých buněk a obsahuje pět hlavních částí. Tři ohraničené černé čtverce v rozích se nazývají finder pattern a využívají se k získání pozice, velikosti a úhlu QR kódu. Finder patterns jsou spojeny střídavě jdoucími černými a bílými buňkami. Ty jsou nazývány timing patterns a využívají se k určení souřadnic každé buňky v QR kódu. Dále se timing patterns používají k opravě zkreslení QR kódu, například při jeho nalepení na zakřivený povrch nebo čtení QR kódu ze strany. K opravě zkreslení se také využívají alignment patterns, které jsou zobrazeny jako ohraničené černé



Obrázek 2.1: Struktura QR kódu [1]

2. QR KÓD

buňky.

Mezi všemi vzory je data area pro uložení zakódovaných dat. Data area také obsahuje Reedovy–Solomonovy kódy, díky kterým lze QR kód stále přečíst, i když je až 30 % jeho plochy poškozené. Aby byl QR kód lépe čitelný, tak musí být ohraničen takzvanou quiet zone, což je bílé místo o šířce nejméně čtyř buněk.

QR kód je mezinárodně standardizovaný, a jelikož je jeho patent ve veřejném vlastnictví, tak má momentálně velmi rozšířené využití v běžném životě. Mezi využití patří například QR platby ke snadnému zaslání údajů pro bankovní převod. QR kódy se také využívají ve zdravotnictví k ukládání informací o pacientovi nebo například k identifikaci zkumavek. Mezi další využití patří i tisk na vstupenky a jízdenky pro jejich rychlé naskenování.

Analýza

Cílem této kapitoly je určení funkčních a nefunkčních požadavků a rozřazení funkčních požadavků podle metody MoSCoW. Ke správnému určení těchto požadavků zanalyzuji současnou realizaci interaktivních procházek, ve které se zaměřím na možnosti gamifikace a dále zanalyzuji již existující řešení.

3.1 Současná realizace interaktivních procházek

Interaktivní procházky se stále více využívají k seznámení návštěvníka s určitým místem. Takovým místem může být například ZOO, muzeum, hrad, město či park. Interaktivní procházka může obsahovat nějaké otázky, na které musí uživatel odpovědět, a úkoly, které musí splnit.

Nejčastěji se takové úkoly a otázky tisknou na papír a na něj uživatel píše své odpovědi. Tyto odpovědi se poté musí donést zadavateli a ten je musí manuálně zkontrolovat. Taková kontrola může být časově náročná.

Další možností realizace interaktivních procházek jsou informační tabule. Tyto tabule mohou obsahovat otázky a úkoly, ale většinou se již nepočítá s odevzdáním odpovědí. Texty na nich mohou být přeložené, ale jejich překlad do nového jazyka je náročnější, jelikož se poté tabule musí znovu vyrobit. Informační tabule se často používají v přírodě a dají se najít pod názvem naučné stezky.

V současné době se objevují způsoby, jak k interaktivním procházkám využít digitální technologie. Jejich využití zjednoduší kontrolu odpovědí a případné přidání nového překladu. Využití digitálních technologií také umožní další možnosti gamifikace.

3.2 Možnosti gamifikace interaktivních procházek

Cílem gamifikace je zvýšení motivace a zábavy uživatelů pomocí využití prvků herního designu v neherním kontextu [2]. Mezi tyto prvky se řadí například

získávání bodů, omezení času či počtu pokusů a zobrazení tabulky výsledků. V gamifikaci lze také využít interaktivní prvky, jako jsou například QR kódy, a přidání možnosti získání nějaké ceny za určitý počet bodů.

Při realizaci interaktivních procházek vytisknutím na papír či vytvořením informačních tabulí se takové herní prvky často využívají. Za splnění úkolů či správné zodpovězení otázek mohou uživatelé získat body. Při plnění úkolů se může omezit čas či počet pokusů. Uživatelé si následně mohou porovnávat počet získaných bodů. Za určitý počet získaných bodů mohou uživatelé na konci procházky získat nějakou cenu.

V případě využití digitálních technologií může být implementace herních prvků více pohlující. Uživatelé nemusí čekat na opravení otázek či úkolů, jelikož se opraví automaticky po jejich odeslání. Počet aktuálně získaných bodů mohou tedy sledovat během plnění procházky.

Při použití digitálních technologií se také otevírají další možnosti gamifikace. Uživatelé mohou například za správné splnění úkolů kromě bodů získávat i virtuální předměty nebo jim lze zobrazit, kolik bodů je ještě potřeba k získání nějaké ceny. Na konci procházky je možné porovnat získané body v tabulce výsledků, tím se ulehčí zjišťování, kdo získal více bodů.

3.3 Existující řešení

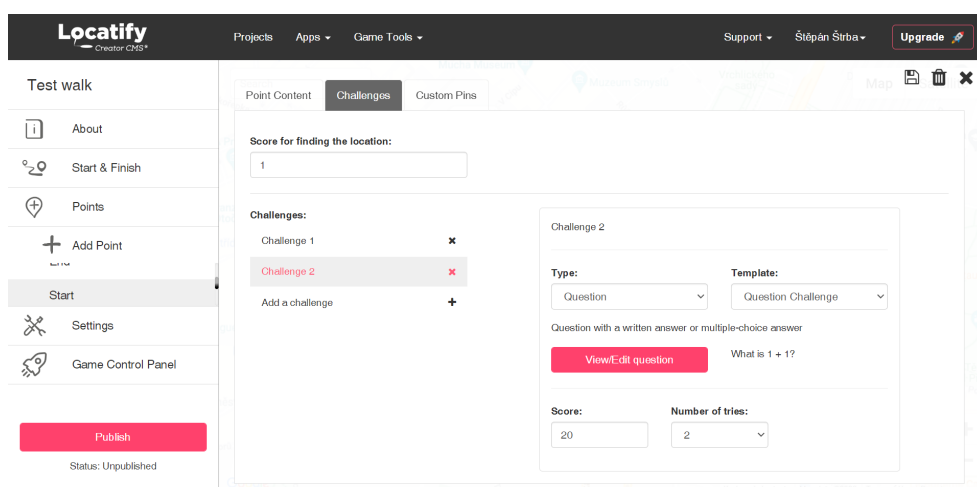
Při analýze existujících řešení pro snadné plánování interaktivních procházek jsem se chtěl zaměřit pouze na řešení využívající QR kódy, ale jelikož takových není mnoho, tak jsem do analýzy zahrnul i řešení, které je nevyužívají. Zvolená řešení jsem našel pomocí online vyhledávačů a zvolil na základě popularity, nabízených vlastností a využití QR kódů.

3.3.1 QR Treasure Hunt Generator!

QR Treasure Hunt Generator! [3] je velmi jednoduchá webová aplikace pro vytváření QR kódů s otázkami. Otázky jsou zapsané v QR kódu jako text, aplikace je tedy nijak nestyluje ani neumožňuje odesílání odpovědí. Uživatel si musí odpovědi sám někde zapisovat a tyto odpovědi poté donést nebo poslat zadavateli. Aplikace umožňuje editaci otázek a přegenerování QR kódů. Vygenerované QR kódy jsou připravené k tisknutí na webové stránce.

3.3.2 Locatify

V pokročilejší webové aplikaci Locatify [4] je možné vytvořit indoor i outdoor procházky. Při vytváření obou typů procházek je možné přidat různé způsoby interakcí, mezi nimi je jednoduchá otázka s odpovědí, pexeso, puzzle či nahrání fotky. Puzzle či pexeso lze vytvořit s vlastním obrázkem. Tyto interakce lze umístit na mapu, určit jejich pořadí, přeložit jejich texty či omezit počet pokusů a potřebný čas k jejich dokončení.



Obrázek 3.1: Vytváření úkolů v Locatify [4]

V případě indoor procházky je možné interakce umístit na plánek patra budovy. Vytvořenou procházku lze sdílet přes QR kód nebo pomocí odkazu na Facebook, LinkedIn a email.

Aplikace využívá gamifikaci pomocí skórování interakcí a vytvoření tabulky s výsledky. Locatify nabízí i placený účet, ve kterém je možné skórování hráčů podle využitého času nebo vytváření placených a soukromých procházek.

Pro zapnutí procházky a její průchod si uživatel musí stáhnout aplikaci na telefon. V této aplikaci je seznam veřejných procházek včetně jejich zobrazení na mapě. V seznamu se dá vyhledávat podle názvu. Po zapnutí procházky lze vytvořit soukromou hru pro kamarády.

3.3.3 Smart Guide

Webová aplikace Smart Guide [5] je určená pouze k vytváření neinteraktivních venkovních procházek. Aplikace již nabízí seznam zajímavých míst k navštívení s vyplněným popisem, gps lokací a vygenerovanou zvukovou stopou. Místa je také možné vytvářet a lze k nim přidat obrázek či vygenerovat zvukovou stopu z popisu.

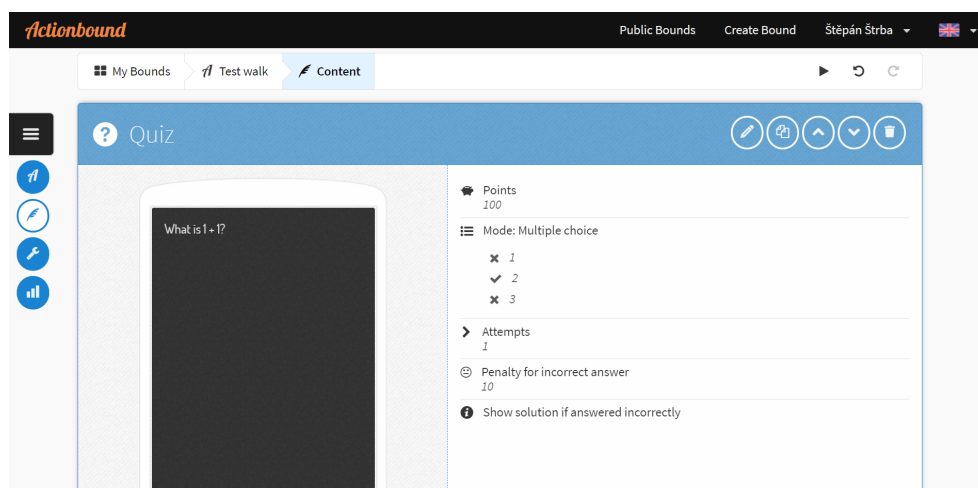
Procházky se vytváří v editoru s mapou, kde se vybírají místa a aplikace automaticky generuje pěší cestu mezi nimi. Procházky a místa lze přeložit do dalších jazyků manuálně nebo automaticky přes strojový překlad DeepL.

Mobilní aplikace nabízí oproti Locatify možnost zvolení preferovaného jazyka či uložení procházky mezi oblíbené. Po zapnutí procházky je také na mapě zobrazena cesta po místech procházky.

3.3.4 Actionbound

Ve webové aplikaci Actionbound [6] lze vytvářet procházky bez specifikování, zda jsou indoor nebo outdoor. Aplikace podporuje zobrazení interakcí na mapě, ale neumožňuje nahrání plánu budovy pro indoor procházky.

Mezi způsoby interakcí patří otázka s psanou odpovědí, odhadem čísla a seřazení seznamu. Další interakce jsou například nalezení místa podle GPS, nalezení a naskenování QR kódu či vyplnění ankety. Interakce se dají rozdělit na úseky, tyto úseky se mohou procházet postupně nebo podle volby hráče. Hráč si je může volit v seznamu nebo podle načtení QR kódu. Tímto způsobem se dá přiřadit QR kód k jednotlivým interakcím. Vygenerované QR kódy se dají stáhnout v jednom PDF souboru.



Obrázek 3.2: Vytvořená otázka v Actionbound [6]

Interakce se mohou také zobrazit uživateli dynamicky, například podle počtu bodů, s určitou šancí, jestli hráč používá mobil se systémem android nebo iOS atp. U interakcí se může omezit čas a počet pokusů nebo přidat penalizace za špatnou odpověď. Aplikace nabízí i placené účty, ve kterých je možné přidat překlad procházky či nechat procházku soukromou.

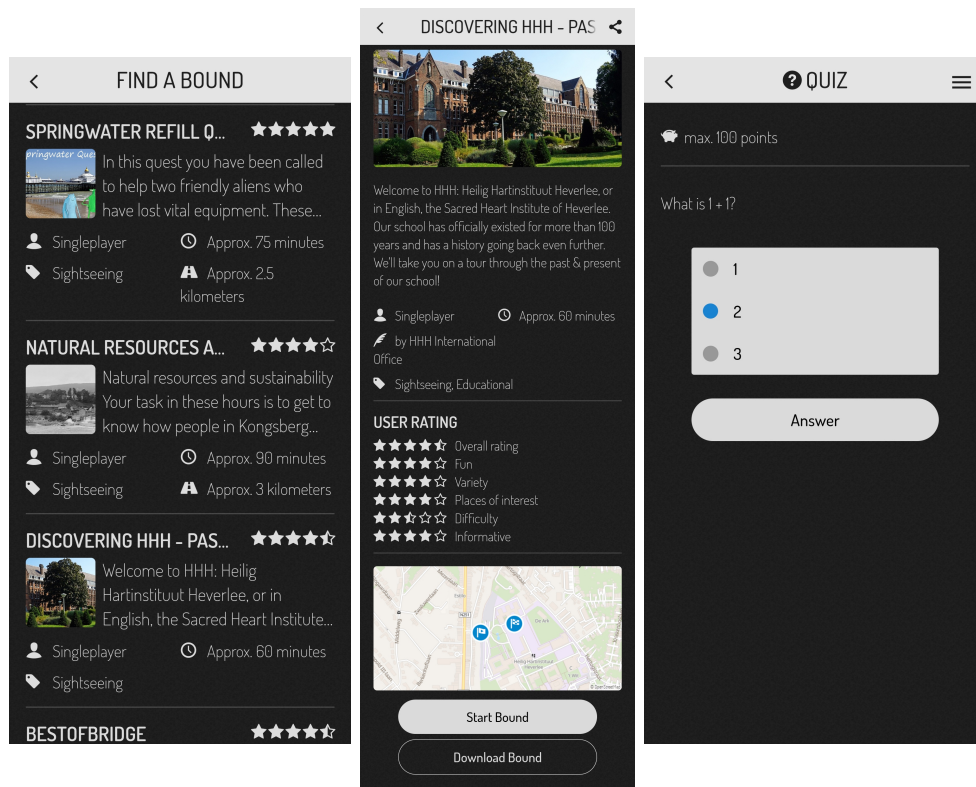
V mobilní aplikaci se oproti ostatním analyzovaným aplikacím dají skenovat QR kódy a přihlásit se do účtu.

3.3.5 Shrnutí

Aplikace Action Bound je jediné z analyzovaných řešení, které podporuje vytváření interaktivních procházek s využitím QR kódů a má alespoň tři různé druhy interaktivních prvků pro QR kód. Pro procházení vytvořených prochá-

zek je ale nutné stažení jejich aplikace na telefon, není to tedy pouze webová aplikace. QR kódy je také nutné manuálně přidávat k interaktivním prvkům.

Většina analyzovaných aplikací je rozdělena na dvě části. Webový editor procházek, který je uživatelským rozhraním a ovládáním zaměřen na uživatele počítačů, a mobilní aplikaci pro procházení vytvořených procházek.



Obrázek 3.3: Seznam procházek, detail procházky a plnění interaktivního prvku v mobilní aplikaci Actionbound

3.4 Požadavky

Z analýzy současné realizace interaktivních procházek a již existujících řešení jsem sestavil následující funkční a nefunkční požadavky. Tyto požadavky jsou důležité pro plánování rozsahu projektu. Funkční požadavky jsem následně rozdělil do čtyř prioritních kategorií podle metody MoSCoW.

3.4.1 Funkční požadavky

Funkční požadavky jsou specifikace funkcí a vlastností, které by měla aplikace splňovat.

3. ANALÝZA

Aplikace bude rozdělena na dvě části. Admin část bude sloužit k organizaci vytvořených procházek a klientská část bude sloužit k procházení vytvořených procházek.

F1 Admin – V admin části aplikace se budou procházky vytvářet a editovat. Současně se zde budou vytvářet interaktivní prvky, sledovat statistiky procházek a bude zde možnost procházku nasdílet či vytisknout vygenerované QR kódy. Pro přístup do aplikace bude nutná registrace.

F1.1 Registrace

F1.2 Přihlášení

F1.3 Organizace procházek – Ve vytvořených procházkách bude možné vyhledávat podle jejich názvu. Procházky bude možné vytvářet, duplikovat, mazat a exportovat či importovat.

F1.3.1 Přehled vytvořených procházek

F1.3.2 Hledání ve vytvořených procházkách

F1.3.3 Přidání procházky

F1.3.4 Duplikování procházky

F1.3.5 Smazání procházky

F1.3.6 Export procházky

F1.3.7 Import procházky

F1.4 Editor procházek – V editoru procházky bude možné vyplnit základní informace a přidat jejich překlad manuálně nebo jej nechat automaticky vygenerovat z již napsaných informací v jiném jazyce. Dále bude mimo jiné možné nastavení viditelnosti procházky, zda bude nutné procházení vytvořených interaktivních prvků v určitém pořadí, nastavení úvodního obrázku či plánku a přidání zvukové stopy či její vygenerování ze základních informací.

F1.4.1 Nastavení základních informací (název, popis, očekávaná časová délka)

F1.4.2 Přidání překladu základních informací

F1.4.3 Vygenerování překladu základních informací

F1.4.4 Nahrání zvukové stopy

F1.4.5 Vygenerování zvukové stopy ze základních informací

F1.4.6 Změna viditelnosti

F1.4.7 Nastavení úvodního obrázku

F1.4.8 Nahrání plánku

F1.4.9 Nastavení lokace procházky na mapě

F1.4.10 Přípravení QR kódů pro tisk

F1.4.11 Zobrazení statistik

- F1.4.12 Povinné procházení prvků v určeném pořadí**
- F1.4.13 Organizace interaktivních prvků** – V organizaci interaktivních prvků bude možné přidat nový interaktivní prvek k procházce či ho smazat, změnit pořadí nebo duplikovat.
 - F1.4.13.1 Přehled interaktivních prvků v procházce**
 - F1.4.13.2 Změna pořadí interaktivních prvků**
 - F1.4.13.3 Přidání interaktivního prvku**
 - F1.4.13.4 Duplikování interaktivního prvku**
 - F1.4.13.5 Odebrání interaktivního prvku**
- F1.4.14 Historie změn**
- F1.4.15 Sdílení vytvořené procházky pomocí odkazu**
- F1.4.16 Sdílení vytvořené procházky na facebook, email**
- F1.5 Editor interaktivních prvků** – V editoru interaktivních prvků bude možné vyplnění základních informací a jejich překlad či přidání automatického překladu z již vyplněných informací v jiném jazyce. Dále bude možné nastavení typu, počtu bodů, omezení počtu pokusů nebo potřebného času.
 - F1.5.1 Nastavení základních informací (názevu a popisu)**
 - F1.5.2 Přidání překladu základních informací**
 - F1.5.3 Vygenerování překladu základních informací**
 - F1.5.4 Nahrání zvukové stopy**
 - F1.5.5 Vygenerování zvukové stopy ze základních informací**
 - F1.5.6 Umístění na mapu**
 - F1.5.7 Nastavení typu**
 - F1.5.8 Nastavení počtu bodů**
 - F1.5.9 Omezení počtu pokusů**
 - F1.5.10 Omezení potřebného času**
 - F1.5.11 Přidání penalizace za překročení času**
 - F1.5.12 Změna viditelnosti**
 - F1.5.13 Nastavení dynamického zobrazení**
- F1.6 Typy interaktivních prvků** – V aplikaci bude možné vytvořit následující typy interaktivních prvků.
 - F1.6.1 Otázka s vyplněním textu**
 - F1.6.2 Otázka s vyplněním čísla**
 - F1.6.3 Otázka se seřazením odpovědí**
 - F1.6.4 Otázka s možnostmi True/False**
 - F1.6.5 Informace**
 - F1.6.6 Načtení správného QR kódu z několika QR kódů**

F2 Klient – V klientské části se budou zobrazovat naskenované interaktivní prvky a informace o vytvořených procházkách.

F2.1 Čtečka QR kódů

F2.2 Zobrazení načteného interaktivního prvku

F2.3 Zobrazení základních informací o načtené procházce

F2.4 Zobrazení plánu načtené procházky

F2.5 Zobrazení mapy načtené procházky

F2.6 Přehled procházek – Procházky bude možné zobrazit v seznamu nebo na mapě. V seznamu procházek bude možné vyhledávat.

F2.6.1 Seznam procházek

F2.6.2 Vyhledávání procházek

F2.6.3 Mapa procházek

F2.7 Profil – Pro přístup do aplikace bude nutná registrace. Uživatel si díky tomu bude moci ukládat procházky do oblíbených procházek a nastavit preferovaný jazyk procházek.

F2.7.1 Registrace

F2.7.2 Přihlášení

F2.7.3 Seznam splněných procházek

F2.7.4 Seznam zahájených procházek

F2.7.5 Seznam oblíbených procházek

F2.7.6 Nastavení profilu

F2.7.6.1 Preferovaný jazyk

3.4.2 Nefunkční požadavky

Nefunkční požadavky specifikují technické vlastnosti aplikace.

N1 Webová aplikace – Implementace aplikace bude ve formě webové aplikace.

N2 Responzivita – Admin část aplikace bude mířit primárně na uživatele počítačů, ale bude přístupná i z mobilních telefonů. Klientská část aplikace bude mířit pouze na uživatele mobilních telefonů.

N3 Lokalizace – Aplikace bude pouze v anglickém jazyce, procházky bude možné vytvářet i v jiných jazycích.

N4 Indikace stavu – Uživatel bude mít vždy přehled o aktuálním stavu aplikace, v případě dlouhé operace se zobrazí animace načítání.

N5 Rozšiřitelnost – Aplikace bude lehce rozšiřitelná o další typy interaktivních prvků.

3.4.3 MoSCoW

MoSCoW [7] je metoda pro prioritizaci požadavků. Priority se požadavkům přidělují jejich rozdělením do čtyř kategorií podle významnosti:

- 1 **Must have** – požadavky v této kategorii mají největší prioritu. Bez těchto požadavků je projekt nesplněný.

F1 Admin

F1.1 Registrace

F1.2 Přihlášení

F1.3 Organizace procházek

F1.3.1 Přehled vytvořených procházek

F1.3.3 Přidání procházky

F1.3.5 Smazání procházky

F1.4 Editor procházek

F1.4.1 Nastavení základních informací (název, popis, očekávaná časová délka)

F1.4.6 Změna viditelnosti

F1.4.10 Připravení QR kódů pro tisk

F1.4.13 Organizace interaktivních prvků

F1.4.13.1 Přehled interaktivních prvků v procházce

F1.4.13.3 Přidání interaktivního prvku

F1.4.13.5 Odebrání interaktivního prvku

F1.4.15 Sdílení vytvořené procházky pomocí odkazu

F1.5 Editor interaktivních prvků

F1.5.1 Nastavení základních informací (názvu a popisu)

F1.5.7 Nastavení typu

F1.5.8 Nastavení počtu bodů

F1.6 Typy interaktivních prvků

F1.6.1 Otázka s vyplněním textu

F1.6.4 Otázka s možnostmi True/False

F1.6.5 Informace

F2 Klient

F2.1 Čtečka QR kódů

F2.2 Zobrazení načteného interaktivního prvku

F2.3 Zobrazení základních informací o načtené procházce

F2.6 Přehled procházek

F2.6.1 Seznam procházek

2 Should have – požadavky v této kategorii jsou důležité a přidávají projektu významnou hodnotu

F1 Admin

F1.3 Organizace procházek

F1.3.2 Hledání ve vytvořených procházkách

F1.4 Editor procházek

F1.4.2 Přidání překladu základních informací

F1.4.7 Nastavení úvodního obrázku

F1.4.8 Nahrání plánu

F1.4.13 Organizace interaktivních prvků

F1.4.13.2 Změna pořadí interaktivních prvků

F1.5 Editor interaktivních prvků

F1.5.2 Přidání překladu základních informací

F1.5.9 Omezení počtu pokusů

F1.5.10 Omezení potřebného času

F1.5.12 Změna viditelnosti

F1.6 Typy interaktivních prvků

F1.6.2 Otázka s vyplněním čísla

F1.6.3 Otázka se seřazením odpovědí

F2 Klient

F2.4 Zobrazení plánu načtené procházky

F2.6 Přehled procházek

F2.6.2 Vyhledávání procházek

F2.7 Profil

F2.7.1 Registrace

F2.7.2 Přihlášení

F2.7.3 Seznam splněných procházek

F2.7.4 Seznam zahájených procházek

F2.7.6 Nastavení profilu

F2.7.6.1 Preferovaný jazyk

3 Could have – požadavky v této kategorii nejsou důležité, a tedy ani nutné ke splnění, ale jsou příjemným dodatkem do projektu.

F1 Admin

F1.3 Organizace procházek

F1.3.4 Duplikování procházky

F1.4 Editor procházek

F1.4.3 Vygenerování překladu základních informací

F1.4.4 Nahrání zvukové stopy

F1.4.5 Vygenerování zvukové stopy ze základních informací

F1.4.9 Nastavení lokace procházky na mapě

F1.4.12 Povinné procházení prvků v určeném pořadí

F1.4.11 Zobrazení statistik

F1.4.16 Sdílení vytvořené procházky na facebook, email

F1.4.13 Organizace interaktivních prvků

F1.4.13.4 Duplikování interaktivního prvku

F1.5 Editor interaktivních prvků

F1.5.3 Vygenerování překladu základních informací

F1.5.4 Nahrání zvukové stopy

F1.5.5 Vygenerování zvukové stopy ze základních informací

F1.5.6 Umístění na mapu

F1.5.11 Přidání penalizace za překročení času

F1.6 Typy interaktivních prvků

F1.6.6 Načtení správného QR kódu z několika QR kódů

F2 Klient

F2.5 Zobrazení mapy načtené procházky

F2.6 Přehled procházek

F2.6.3 Mapa procházek

F2.7 Profil

F2.7.5 Seznam oblíbených procházek

4 **Won't have** – požadavky v této kategorii nebudou realizovány v aktuálním projektu, ale mohou být implementovány v dalších verzích projektu.

F1 Admin

F1.3 Organizace procházek

F1.3.6 Export procházky

F1.3.7 Import procházky

F1.4 Editor procházek

3. ANALÝZA

F1.4.14 Historie změn

F1.5 Editor interaktivních prvků

F1.5.13 Nastavení dynamického zobrazení

Návrh

V této kapitole se budu z velké části věnovat výběru technologií, které použiji k implementaci aplikace, a návrhu uživatelského rozhraní. Představím také databázový model.

4.1 Výběr technologií

Výběr správných technologií je nedílnou součástí návrhu webové aplikace. Podle návrhového vzoru Split Stack Development [8] jsem se rozhodl rozdělit vývoj frontendu a backendu na dvě části. Díky tomu mohu pro frontend a backend zvolit rozdílné technologie.

4.1.1 Frontend

V kontextu webových aplikací je frontend část aplikace, kterou uživatel vidí a interaguje s ní. Na frontendu se většinou používá kombinace technologií HTML, CSS a JavaScript [9]. Vzhledem k těmto technologiím se k tvorbě frontend částí aplikací nejčastěji využívají JavaScriptové frameworky. Pro implementaci frontendu diplomové práce jsem se rozhodl využít framework Nuxt.

4.1.1.1 Nuxt

Nuxt [10] je framework založený na Vue.js a vytvořený s cílem ulehčení a zefektivnění vytváření Vue.js aplikací.

Jedna z jeho funkcí, určená k lehčímu vývoji, je například automatické vytváření URL k souborům uložených ve složce pages. Pro všechny takto vytvořené URL Nuxt automaticky dělí kód aplikace a při vykreslení aplikace posílá uživateli pouze ty části, které potřebuje. Tím zmenšuje velikost posílaných dat a zvyšuje rychlost načtení aplikace.

Mezi další výhody Nuxtu patří využití modulární architektury, která umožňuje snadné rozšíření pomocí modulů. Tyto moduly vytváří velká a aktivní komunita.

Nuxt dále podporuje Server Side Rendering a generování statických stránek, což je ideální pro webové stránky, jejichž obsah se nemění příliš často.

4.1.2 Backend

Backend je oproti frontendu část aplikace, kterou uživatel nevidí. Tato část aplikace zpracovává požadavky z frontendu a stará se o data v databázi, správu nahrávaných souborů, autentizaci, autorizaci a o další funkce potřebné k fungování webových aplikací. K implementaci backendu jsem se rozhodl využít PHP framework Laravel.

4.1.2.1 Laravel

Laravel je open source webový framework vytvořený v roce 2011. Dá se využít například jako full stack framework nebo API backend. Pro využití jako API backend je vhodný díky jeho implementaci autentizace či validace dat.

Pro práci s daty z databáze Laravel využívá Eloquent ORM. Díky ORM lze lehce pracovat s řádky z databáze jako s objekty, které mají vlastnosti a vztahy k ostatním objektům. Získávání dat je prováděno pomocí vlastního tvůrce SQL dotazů, který má intuitivní syntaxi a umožňuje efektivní práci s databází. Laravel podporuje práci s pěti databázemi, mezi kterými je například MySQL, PostgreSQL či SQLite.

Routing se využívá k mapování HTTP požadavků na specifické akce. V Laravelu je tento proces velmi flexibilní a intuitivní. Při definici URL adresy je velmi jednoduché specifikovat požadovanou HTTP metodu nebo určit proměnné parametry v URL s jejich následnou validací. Mapování HTTP požadavků lze také rozdělit do skupin podle různých vlastností, například podle subdomény či prefixu. Díky tomu je jejich organizace a správa jednodušší.

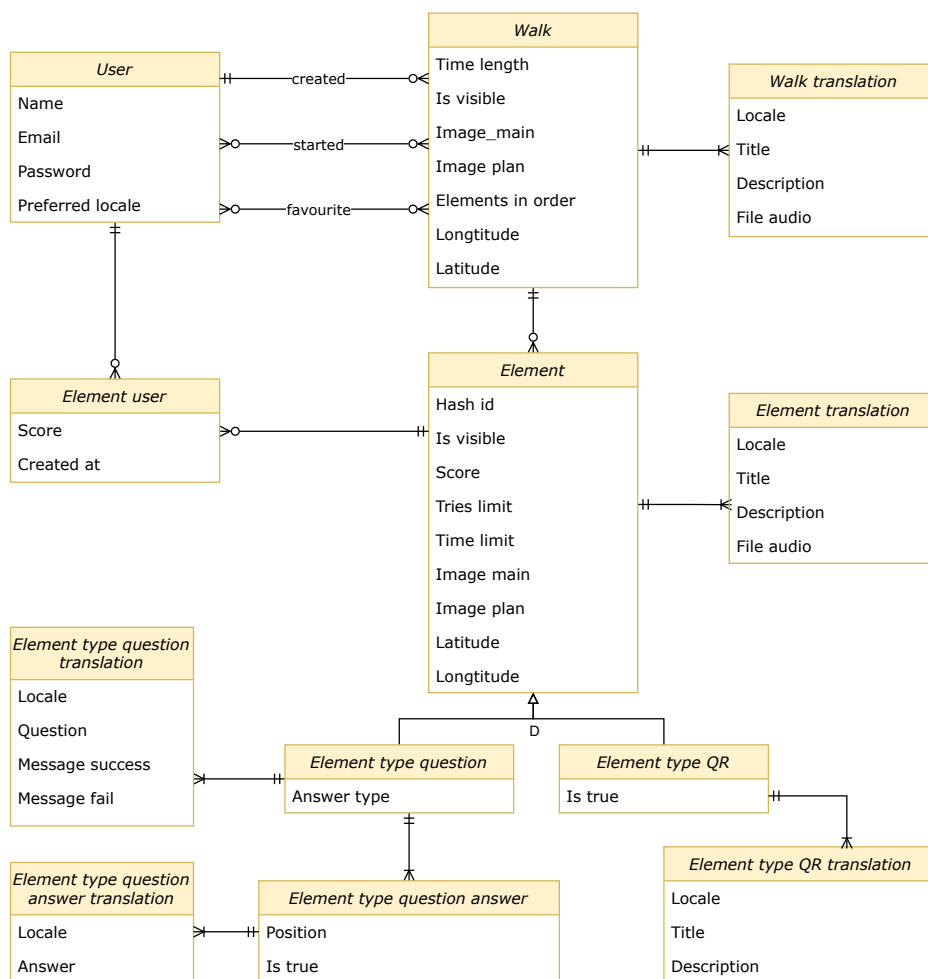
Mezi další funkčnosti Laravelu patří jednoduché zvolení způsobu uložení souborů či session dat.

4.2 Návrh databáze

Dobrý návrh databáze umožňuje minimalizovat problémy, které by mohly vzniknout při její implementaci či správě. Proto jsem se rozhodl udělat návrhový diagram podle entitně vztahového modelu. Takový diagram se nazývá entitně vztahový diagram a využívá se k popisu a návrhu databázových systémů se zaměřením na popis struktury a vztahů entit.

Pro tento diagram existuje mnoho notací. Já jsem zvolil notaci Crow's foot. Tato notace znázorňuje kardinalitu pomocí svislých čar, prázdných kružnic

a diagonálních čar. Při návrhu databáze jsem se snažil dodržet, aby aplikace byla snadno rozšiřitelná o další druhy interaktivních prvků.



Obrázek 4.1: Návrh databáze

4.3 Uživatelské rozhraní

Cílem uživatelského rozhraní je umožnit uživateli snadnou interakci s aplikací. Díky uživatelskému rozhraní může aplikace přijímat vstupy od uživatele a na jejich základě provádět nějaké akce. Mezi tyto akce patří zobrazení dat, změna nastavení aplikace či uložení dat. Uživatelské rozhraní se skládá z různých typů vizuálních prvků, například z tlačítek a formulářů.

Návrh uživatelského rozhraní obsahuje prvky pro všechny funkční požadavky z kategorií MoSCoW metody must have, should have, could have, a i ně-

4. NÁVRH

které z kategorie won't have. Aplikace je rozdělena na dvě části, admin a klient. Vytvořený návrh splňuje všechny nefunkční požadavky.

Při návrhu jsem se inspiroval u existujících řešení, která jsem analyzoval v kapitole 3.3. Dále jsem se zaměřil na minimalismus a snadno srozumitelný design. Toho jsem docílil použitím intuitivních vizuálních prvků, které jsou běžně používané i v jiných aplikacích. Díky tomu se bude v aplikaci uživatel snadno a rychle orientovat.

4.3.1 Admin

Tato část aplikace obsahuje přihlášení, registraci, organizaci a editor procházek a interaktivních prvků. Designem je zaměřená na uživatele počítačů.

4.3.1.1 Přihlášení a registrace

Aplikace by měla obsahovat přihlášení a registraci, proto jsem navrhl jednoduché formuláře. Uživatel bude při registraci muset vyplnit své uživatelské jméno, email a heslo a na přihlášení bude používat email a heslo.



Obrázek 4.2: Návrh organizace procházek

4.3.1.2 Organizace procházek

V organizaci procházek má být přehled procházek s možností vyhledávání. Procházky by měly jít vytvářet, duplikovat, mazat, exportovat a importovat.

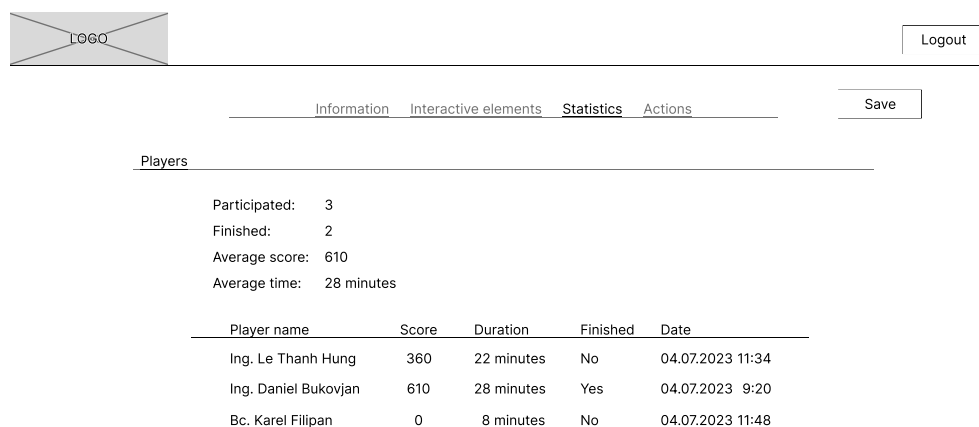
Pro přehled procházek jsem vytvořil jednoduchou tabulku se jménem a id procházky (viz obrázek 4.2). Dále jsou u procházky akce, které lze s procházkou provádět. Nad touto tabulkou je textové pole pro vyhledávání mezi procházkami a tlačítka pro vytvoření či importování procházky. V návrhu jsem také přidal stránkování tabulky.

Při kliknutí na akci duplikace, exportu či smazání se uživateli zobrazí dialog s otázkou, zda chce akci opravdu provést. Tím zamezím neúmyslné smazání procházky či zapnutí operace, která by mohla trvat dlouho.

Obrázek 4.3: Návrh editoru procházky

Obrázek 4.4: Návrh přidání překladu

4. NÁVRH



Participated: 3
Finished: 2
Average score: 610
Average time: 28 minutes

Player name	Score	Duration	Finished	Date
Ing. Le Thanh Hung	360	22 minutes	No	04.07.2023 11:34
Ing. Daniel Bukovjan	610	28 minutes	Yes	04.07.2023 9:20
Bc. Karel Filipan	0	8 minutes	No	04.07.2023 11:48

Obrázek 4.5: Návrh statistik procházky

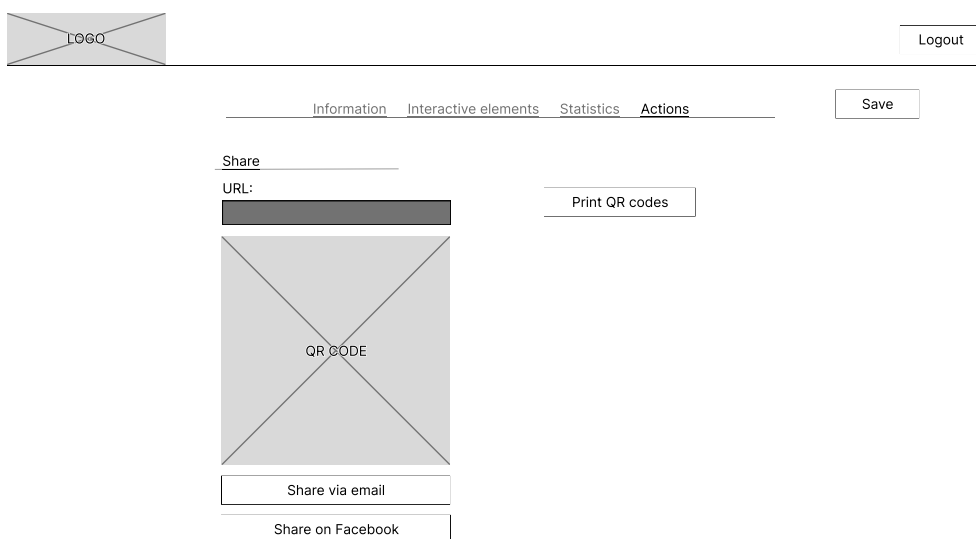
4.3.1.3 Editor procházek

Editor procházky jsem pro větší přehlednost rozdělil na 4 části. V první části je editace vlastností procházky (viz obrázek 4.3). Zde lze nastavit viditelnost procházky či zda je povinné procházení prvků v určeném pořadí. Lze také nastavit lokaci procházky, nahrát hlavní obrázek a obrázek plánku. Ohledně textu je možné nastavit název a popis procházky. Název a popis je možné manuálně či automaticky přeložit (viz obrázek 4.4). K procházce lze také nahrát zvukovou stopu či ji vygenerovat z popisu.

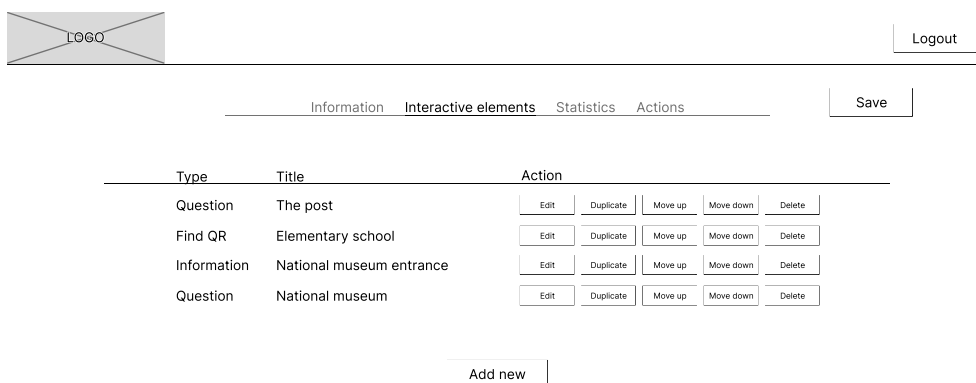
Další částí je organizace interaktivních prvků a jejich editor, popisu této části se budu věnovat později.

Následující částí je zobrazení statistik o procházce (viz obrázek 4.5). V té je zobrazený počet hráčů, kteří se procházky zúčastnili, a počet hráčů, kteří procházku dokončili. Je zde také zobrazeno průměrné skóre a průměrně strávený čas hráčů, kteří procházku dokončili. Ve statistikách je také tabulka všech hráčů, kteří procházku zahájili. V té je vidět jejich aktuální skóre, strávený čas, zda již procházku dokončili a datum, kdy ji zahájili.

V poslední části je možné vytvořenou procházku sdílet pomocí odkazu či QR kódu. Procházku lze také sdílet na facebook či pomocí emailu. V této části je také možné vytisknout QR kódy procházky a všech jejích interaktivních prvků (viz obrázek 4.6).



Obrázek 4.6: Návrh akcí v editoru procházky



Obrázek 4.7: Návrh organizace interaktivních prvků

4.3.1.4 Organizace interaktivních prvků

Organizace interaktivních prvků obsahuje podobné akce jako organizace procházek. U interaktivních prvků je nutné mít možnost jejich přidání, duplikace a odebrání. Oproti organizaci procházek navíc obsahuje možnost změny jejich pořadí.

V této části jsem pro přehled interaktivních prvků navrhl velice podobnou tabulku jako pro přehled vytvořených procházek. V tabulce se zobrazuje název a typ interaktivních prvků a akce, které s nimi lze provádět (viz obrázek 4.7).

4. NÁVRH

Prvky oproti procházkám nelze exportovat ani importovat, ale lze měnit jejich pořadí. Tato tabulka také neobsahuje stránkování a tlačítko pro přidání nového interaktivního prvku jsem umístil pod tabulku. Díky přemístění tlačítka pro vytvoření bude uživatel očekávat nově vytvořený prvek na konci tabulky. Stejně jako v organizaci procházek zde zobrazím dialog pro potvrzení akce při kliknutí na duplikaci či smazání.

LOGO Logout

Add new interactive element

Visibility: Public

Location in GPS coordinates:

Points for completing: Yes

Limit time in seconds: Yes

Limit number of tries: Yes

Penalise if time over x seconds: Yes

Type

Choose type: ▾

English Czech +

Name:

Description:

Audio:

0 1 Add QR code +

Is correct: Yes

Name:

Message after scanning:

Obrázek 4.8: Návrh editoru interaktivních prvků

4.3.1.5 Editor interaktivních prvků

Editor interaktivních prvků je možné otevřít po kliknutí na vytvoření či editaci prvku v organizaci interaktivních prvků. Editor se zobrazuje v modálním okně (viz obrázek 4.8) a po jeho zavření se uživatel vrátí do organizace prvků.

S cílem zlepšení přehlednosti jsem rozdělil editor interaktivních prvků na tři sekce. V první sekci se nastavuje viditelnost, lokace, počet bodů, penalizace

za překročení určitého času a omezení času či pokusů.

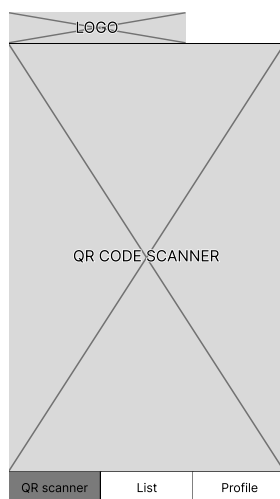
Ve druhé sekci se určuje typ interaktivního prvku. Prvek může být typu informace, otázka nebo načtení správného QR kódu z několika. U otázky se dále určuje typ odpovědi mezi textem, číslem, True/False odpověďmi či seřazením odpovědí.

Ve třetí sekci se nastavují textové informace o prvku, jako je název a popis. Stejně jako u editoru procházek je možné přidat manuální či automatický překlad textových informací a nahrát či vygenerovat zvukovou stopu.

Vzhledem k typu prvku se mění vlastnosti, které je možné upravit. U informace nelze nastavit počet pokusů, penalizaci za překročení času či omezení času. U otázek je nutné vyplnit otázku a zprávu po úspěšném vyplnění. K otázkám je také nutné vyplnit odpovědi.

U odpovědí se vždy musí vyplnit text nebo číslo, dále se ale nastavují různé vlastnosti vzhledem k typu odpovědi. K otázce s textovou odpovědí lze přidat jakékoliv textové odpovědi, k otázce s číselnou odpovědí lze přidat pouze číselné odpovědi. U otázky se seřazením možností je možné nastavení správného seřazení odpovědí a u otázky s možnostmi True/False je nutné nastavení jejich správnosti.

V případě prvku typu načtení správného QR kódu z několika je možné přidání QR kódů. U těchto kódů je nutné nastavit jejich název, správnost a text po naskenování.



Obrázek 4.9: Návrh čtečky QR kódů

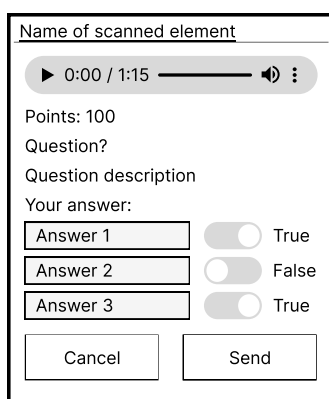
4.3.2 Klient

Klientská část aplikace je zaměřená na uživatele mobilních zařízení. Obsahuje přihlášení, registraci, čtečku QR kódů, zobrazení načteného interaktivního prvku, seznam procházek, detail procházky a zvolení preferovaného jazyka.

Pro snadnější orientaci v aplikaci jsem klientskou část rozdělil na tři části. Mezi těmito částmi se dá navigovat pomocí tlačítek na spodní části aplikace. Tlačítka jsou zvýrazněná vzhledem k aktuální zobrazené části aplikace a jejich pozice se při používání aplikace nemění. Při dlouhém obsahu je umožněno jeho posouvání a tlačítka k navigaci stále zůstávají na spodní části aplikace.

4.3.2.1 Přihlášení a registrace

Přihlášení a registrace používá stejné formuláře jako v admin části aplikace, jen jsou přizpůsobené pro mobilní zařízení.



The image shows a mobile application interface for a question. At the top, there is a header "Name of scanned element". Below it is a progress bar showing "0:00 / 1:15". The question details include "Points: 100", "Question?", and "Question description". The user's answer is shown in three rows: "Answer 1" with a "True" toggle, "Answer 2" with a "False" toggle, and "Answer 3" with a "True" toggle. At the bottom, there are "Cancel" and "Send" buttons.

Obrázek 4.10: Návrh zobrazení otázky s možnostmi True/False

4.3.2.2 Čtečka QR kódů

Čtečka QR kódů vyplňuje celou část obsahu (viz obrázek 4.9). Po načtení QR kódu procházky je uživatel přesměrován na detail načtené procházky. Při načtení QR kódu interaktivního prvku jsou informace o tomto prvku zobrazeny v modálním okně. Vzhledem k typu a vyplněným vlastnostem prvku jsou zobrazeny různé informace.

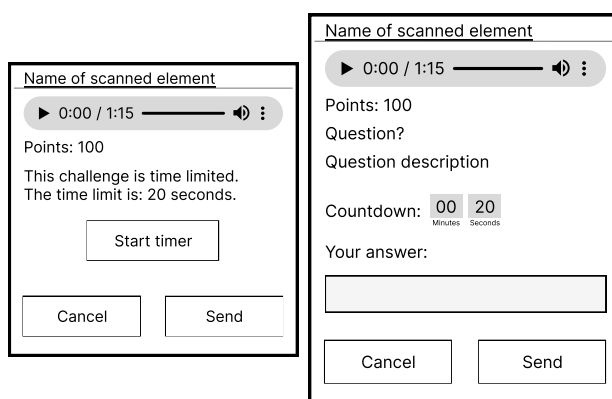
Po načtení informačního prvku je zobrazen název, počet bodů, popis a zvuková stopa. Při načtení otázky je navíc zobrazen text otázky a možnost na ni odpovědět. U otázky s vyplněním textu či čísla je zobrazen textový či číselný vstup. U otázky se seřazením odpovědí je možné měnit pozici odpovědí a u otázky s určením True/False je možné určit jejich správnost (viz obrázek 4.10).

V případě omezení potřebného času se nezobrazí text otázky, popis prvku, zvuková stopa ani případné odpovědi. Místo toho je v okně zobrazen časový limit a tlačítko pro zapnutí časovače. Po zapnutí časovače se schované informace opět zobrazí (viz obrázek 4.11).

Po odeslání odpovědi se otevře jiné modální okno, ve kterém je zobrazen počet získaných bodů. Vzhledem ke správnosti odeslané odpovědi se zde také

zobrazí text. Pokud uživateli zbývají pokusy, tak se může vrátit k vyplnění odpovědi.

Interaktivní prvek typu načtení správného QR kódu je rozdělen na dvě části. Při načtení prvku se zobrazí stejné informace jako při načtení informačního prvku. Při načtení QR kódu se navíc zobrazí tlačítko pro potvrzení naskenování. Po jeho kliknutí se zobrazí stejné modální okno, jako při odeslání odpovědi k otázce.



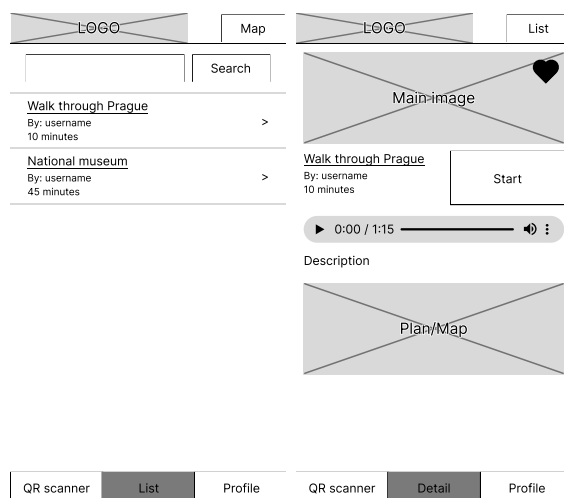
Obrázek 4.11: Návrh zobrazení otázky s omezením času

4.3.2.3 Přehled a detail procházek

V této části aplikace je zobrazen přehled a detail procházek (viz obrázek 4.12). Přehled procházek lze zobrazit v seznamu či na mapě. Při jejich zobrazení v seznamu je možné procházky vyhledávat podle názvu. Z informací o procházce je v seznamu zobrazen pouze název, autor a časová náročnost procházky. Po kliknutí na procházku se uživatel dostane na detail procházky.

V detailu procházky jsou zobrazeny všechny základní informace o procházce, jako je název a popis. Dále je zde zmíněn autor, časová náročnost a zvuková stopa. Nad základními informacemi je zobrazen hlavní obrázek procházky společně s možností přidání procházky do seznamu oblíbených procházek. Vedle názvu je tlačítko pro zahájení procházky. Po kliknutí na něj se na jeho místě zobrazí aktuální skóre a strávený čas. Pod popisem procházky je zobrazen plánek či lokace na mapě.

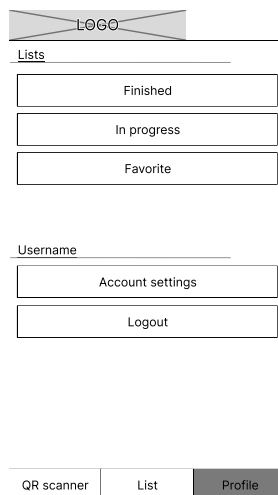
4. NÁVRH



Obrázek 4.12: Návrh přehledu a detailu procházek

4.3.2.4 Profil

Profil obsahuje tlačítka pro zobrazení seznamů splněných, zahájených a oblíbených procházek. Dále obsahuje tlačítko k odhlášení a tlačítko k zobrazení nastavení účtu, ve kterém se nastavuje preferovaný jazyk.



Obrázek 4.13: Návrh profilu

Implementace

V této kapitole popíši implementaci funkčních požadavků na frontendu i backendu včetně struktury kódu a seznamu použitých dodatečných balíčků. Při implementaci jsem vycházel z návrhu databáze a uživatelského rozhraní, které byly představeny v kapitole 4 Návrh.

5.1 Backend

Backend aplikace jsem implementoval ve frameworku Laravel. Základ aplikace jsem vytvořil pomocí příkazu `composer create-project laravel/laravel example-app`. Tento základ jsem poté rozšiřoval pro účely implementace funkčních požadavků.

5.1.1 Struktura zdrojového kódu

Základ aplikace poskytuje výchozí strukturu projektu, která je určena jako výchozí bod pro malé i velké aplikace. Laravel sice tuto strukturu prakticky nijak nevynucuje, ale i tak je dobré ji dodržovat. Dodržením jsem zaručil dobrou orientaci ve zdrojovém kódu. Zjednodušenou strukturu zdrojového kódu backend části aplikace popisují v obrázku 5.1. V této struktuře zmiňuji pouze soubory a složky, které jsem upravoval nebo jsou něčím zajímavé.

5.1.2 Použité dodatečné balíčky

Při implementaci jsem využil následující dodatečné balíčky. Díky těmto balíčků jsem nemusel těžce implementovat nové funkčnosti. Využitím balíčků jsem také zaručil bezpečnější kód, jako je tomu například při balíčcích `Sanctum` a `Fortify`, které se starají o autentizaci.

Astronomic/laravel-translatable¹ – Balíček pro jednoduché vytváření multijazyčných modelů databázových objektů.

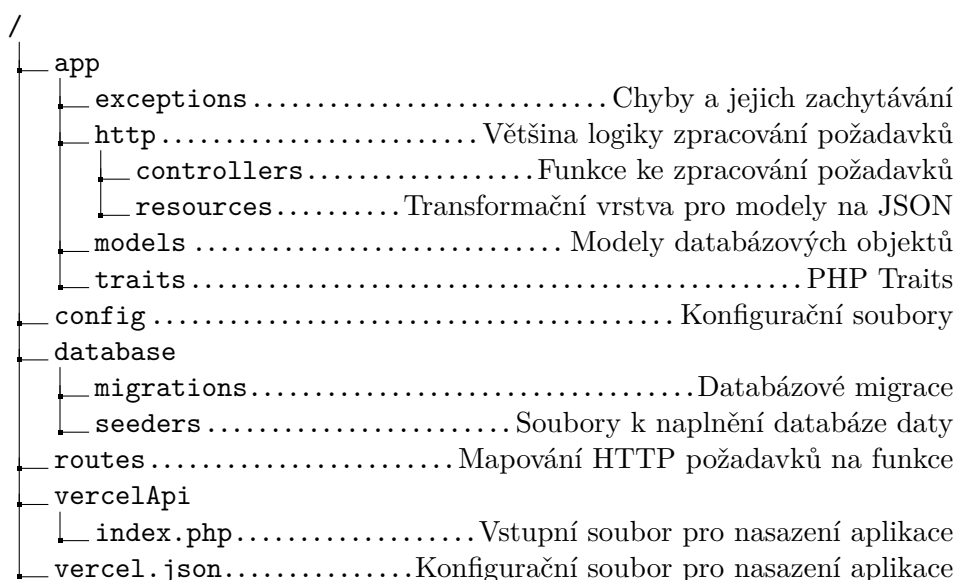
Google/cloud-text-to-speech² – PHP klient od společnosti Google pro používání jejich Text-to-Speech API.

Google/cloud-translate³ – PHP klient od společnosti Google pro používání jejich API na překlad textu.

Laravel/fortify⁴ – Balíček pro správu uživatelů při autentizaci.

Laravel/sanctum⁵ – Balíček pro autentizaci uživatelů z frontendu.

League/flysystem-aws-s3-v3⁶ – Balíček pro připojení k souborovému systému Amazon S3.



Obrázek 5.1: Zjednodušená struktura backend části aplikace

5.1.3 Implementované funkční požadavky

Při implementaci backend části aplikace jsem implementoval většinu funkčních požadavků, v této kapitole popisují pouze ty zajímavější.

5.1.3.1 Řízení přístupu uživatelů

Aplikace je rozdělena na dvě části, admin a klient. V admin části aplikace mohou uživatelé upravovat a vytvářet své procházky, v klientské části mají uživatelé informace o svých zahájených, dokončených a oblíbených procházkách. V obou případech je nutné řídit přístup k těmto datům. Řízení přístupu uživatelů se skládá ze čtyř částí.

Při přístupu k datům musí uživatel nejdříve sdělit svoji identitu. Této identitě nelze automaticky věřit, a proto ji ověříme pomocí autentizace.

Autentizace je proces verifikace uživatelské identity. Při autentizaci se údaje poskytnuté uživatelem zkontrolují s údaji o uživateli uloženými v databázi. Pokud jsou údaje stejné, tak je uživateli povolen přístup do aplikace. Metody verifikace identity se dají rozdělit do následujících tří kategorií:

Co uživatel zná – Nejčastěji používaný typ autentizace. Využívá se tajné informace, kterou uživatel zná. Většinou jméno a heslo nebo pin.

Co uživatel má – K autentizaci se využívá něco, co uživatel fyzicky vlastní. Příkladem je třeba chytrá karta, kreditní karta nebo i občanský průkaz.

Čím uživatel je – Použití biometrických senzorů ke kontrole specifických vlastností, které uživatel má a nelze je jednoduše zfalšovat. Například otisky prstu či skenování sítnice.

K implementaci autentizace jsem se rozhodl pro metodu verifikace identity co uživatel má, jelikož je nejčastěji používaná a jednoduchá k implementaci. Pro implementaci jsem na backendu využil Laravel balíčky **Fortify** a **Sanctum**.

Fortify má již připravené API endpointy a controllery pro registraci, přihlášení a mnoho dalšího. Při instalaci byl také vytvořen model pro uživatele a migrace pro vytvoření tabulky v databázi. Po instalaci a provedení migrace jsem prakticky nemusel nic nastavovat a nainstalované API endpointy jsem mohl rovnou začít používat z frontendu.

Sanctum se stará o autentizaci uživatelů ze Single Page Aplikací. K autentizaci využívá vestavěné autentizační služby v Laravelu založené na cookies. Dále poskytuje ochranu před CSRF či XSS útoky. Po instalaci balíčku **Sanctum** jsem musel nastavit několik proměnných prostředí. Pro správné nastavení ochrany proti CSRF útokům jsem musel nastavit domény, které mohou provádět stateful požadavky a povolenou doménu k ukládání session cookie. Dále jsem nastavil ukládání session informací do databáze. Balíček **Sanctum** jsem poté využíval i k zamezení přístupu neautorizovaných uživatelů k několika API endpointům.

Po autentizaci uživatele se přechází k autorizaci. Při autorizaci se rozhoduje, zda má uživatel oprávnění přístupu k požadovaným informacím nebo prováděným akcím.

5. IMPLEMENTACE

```
1 Route::group(  
2     [  
3         'prefix' => 'user',  
4         'middleware' => 'auth:sanctum'  
5     ],  
6     function () {  
7         Route::post(  
8             '/updateLocale',  
9             [UserController::class, 'updateLocale']  
10        );  
11    }  
12 );
```

Ukázka kódu 1: Příklad využití balíčku Sanctum pro zamezení přístupu k API endpointu.

V admin části mojí aplikace mají k informacím a akcím ohledně procházek a interaktivních prvků mít přístup pouze uživatelé, kteří tyto procházky a interaktivní prvky vytvořili. Podobně tomu je i v klientské části aplikace, kde mají mít uživatelé přístup k seznamům pouze svých zahájených, splněných či oblíbených procházek. Dále mají přístup k získaným bodům pouze u procházek, které zahájili a mohou nastavovat pouze svůj preferovaný jazyk. Proto je při většině HTTP požadavků kontrolováno, zda uživatel má přístup k žádaným datům a akcím (viz ukázka kódu 2).

```
1 public function duplicate(Request $request, int $id = NULL)  
2 {  
3     $duplicateWalk = Walk::find($id);  
4  
5     if (empty($duplicateWalk)) {  
6         abort(404);  
7     }  
8  
9     if ($duplicateWalk->owner_id !== $request->user()->id) {  
10        abort(403);  
11    }  
12    :  
13 }
```

Ukázka kódu 2: Autorizace uživatele při duplikaci procházky

Poslední částí řízení přístupu uživatelů je sledování a zaznamenávání aktivit uživatelů. Všechny aktivity uživatele v aplikaci by se měly zaznamenávat pro kontrolu, zda při používání aplikace nedošlo k neoprávněnému přístupu k informacím nebo provedení neoprávněné akce [11, 12]. Tuto část jsem se rozhodl neimplementovat, jelikož samotná implementace by byla náročná a zabrala by mnoho času, který jsem takto mohl využít k implementaci funkčních požadavků.

5.1.3.2 Validace

Pro validaci příchozích dat obsahuje Laravel validační metodu s mnoha již implementovanými validačními pravidly. Pomocí této validační metody jsem validoval prakticky všechna příchozí data. Z již implementovaných pravidel jsem využil například `nullable` a `required` pro určení, zda jsou atributy povinné nebo ne.

Dále jsem často využíval pravidla `min` a `max`, ty validují různé věci vzhledem k ostatním použitým validačním pravidlům. Při použití s validačním pravidlem `mimes` je kontrolována velikost souboru, při použití s `integer` hodnota čísla a s `string` délka textového řetězce. Validační chyby jsou automaticky odchyťovány Laravelem a vráceny v odpovědi.

```

1  $request->validate([
2      'elements_in_order' => 'nullable|boolean',
3      'is_visible' => 'nullable|boolean',
4      'is_with_plan' => 'nullable|boolean',
5      'image_main' => 'nullable|string|max:255',
6      'image_plan' => 'nullable|string|max:255',
7      'time_length' => 'nullable|numeric|min:0',
8      'translations' => 'array|min:1',
9      'translations.*.title' => 'required|string|min:1|max:255',
10     'translations.*.description' => 'required|string|min:1',
11     'translations.*.file_audio' => 'nullable|string|max:255'
12 ]);

```

Ukázka kódu 3: Validace dat při ukládání procházky

5.1.3.3 Překladatelné procházky a prvky

Základní informace procházek a interaktivních prvků mají být překladatelné. Pro splnění tohoto požadavku jsem v databázi vytvořil tabulky s překlady. K lehčí správě překladů jsem využil balíček `Astronomic/laravel-translatable`. Pro použití tohoto balíčku v kódu jsem k modelům přidal `trait Translatable` a specifikoval překladatelné vlastnosti.

5. IMPLEMENTACE

Z veřejně dostupných seznamů používaných jazyků jsem vytvořil vlastní seznam jazyků s jejich kódem, názvem v angličtině a názvem v daném jazyce. Tento seznam jsem pomocí seederu `LanguageSeeder` přidal do databáze. Při zapnutí aplikace jsou kódy jazyků přidány do seznamu dostupných jazyků pomocí provideru `AppServiceProvider`. Díky vyplněnému seznamu dostupných jazyků je možné při požadavku na informace nastavit jazyk aplikace na preferovaný jazyk uživatele a poté vrátit vyplněné informace v tomto jazyce.

```
1 public function register(): void
2 {
3     $this->app->booted(function () {
4         foreach($this->getAvailableLocales() as $locale) {
5             app(Locales::class)->add($locale);
6         }
7     });
8 }
9 public function getAvailableLocales()
10 {
11     return DB::table('languages')
12         ->whereNotIn('code', ['en'])
13         ->orderBy('code')
14         ->pluck('code');
15 }
```

Ukázka kódu 4: Import jazyků do seznamu dostupných jazyků aplikace

V případě, že informace nejsou vyplněny v určitém jazyce poskytuje použitý balíček možnost zapnutí automatického získání překladu z jiného jazyku. Jazyk může být jasně specifikovaný nebo balíček projde seznam všech dostupných jazyků aplikace seshora dolů a použije první, ve kterém je informace přeložena. Já jsem zvolil druhou možnost. Proto jsem chtěl, aby jazyk angličtina byl v seznamu dostupných jazyků jako první. Specifikoval jsem ho tedy v konfiguraci a poté jsem při zapnutí aplikace do seznamu přidával pouze ostatní jazyky. V případě chybějícího překladu tedy balíček vrátí jiný překlad, který je vyplněn.

Do překladů jsem také ukládal odpovědi na otázky. Tyto odpovědi mohou být i číselné, například 0. V takovém případě si balíček myslel, že překlad není vyplněn, jelikož ke kontrole vyplnění překladu využíval PHP funkci `empty`. Pro opravu jsem vytvořil vlastní trait, který používá trait `Translatable` a opravuje funkci `isEmptyTranslatableAttribute`, ve které dochází ke kontrole vyplnění překladu (viz ukázka kódu 5).

```

1 trait MyTranslatableTrait
2 {
3     use Translatable {
4         isEmptyTranslatableAttribute as protected
5         ↪ myIsEmptyTranslatableAttribute;
6     }
7     protected function isEmptyTranslatableAttribute($key,
8         ↪ $value)
9     {
10         return is_null($value) ||
11             $value === '' ||
12             (is_string($value) && strlen(trim($value)) === 0);
13     }
14 }

```

Ukázka kódu 5: Oprava funkce isEmptyTranslatableAttribute v balíčku `Astros-nomic/laravel-translatable`

5.1.3.4 Automatický překlad procházek

K automatickému překladu základních informací procházky jsem využil balíček `Google/cloud-translate`, který umožňuje jednoduché napojení k API pro generování překladů od společnosti Google. Při používání tohoto balíčku jsem narazil na problém se specifikací autentizačních údajů.

Google ve své dokumentaci o autentizaci popisuje pouze možnost autentizace pomocí informací uložených v JSON souboru. V čase implementace této funkčnosti jsem již počítal s budoucím nasazením pomocí služby Vercel⁷, která pouze stahuje soubory z git repozitáře. Tento soubor s autentizačními informacemi bych tedy musel nahrát do svého repozitáře a poté by k těmto informacím měli přístup všichni uživatelé, kteří mají přístup k mému repozitáři. Proto jsem se rozhodl tento JSON specifikovat v proměnných prostředí a autentizační informace specifikovat až při vytváření třídy pro překlad (viz ukázka kódu 6).

5.1.3.5 Automatické vygenerování zvukové stopy

Pro vygenerování zvukové stopy ze základních informací o procházce či prvku jsem použil balíček `Google/text-to-speech`. Tento balíček umožňuje napojení ke Google API, ve které je možné generování zvukových stop z textu. Použití je velmi jednoduché, stačí nastavit text, jazyk textu a formát zvukového souboru. Problémy se specifikací autentizačních údajů jsem vyřešil již při implementaci automatického překladu informací.

⁷<https://vercel.app/>

```

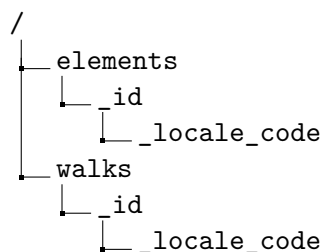
1  $translate = new TranslateClient([
2      "keyFile" => json_decode(
3          env('GOOGLE_APPLICATION_CREDENTIALS'),
4          true
5      )
6  ]);
7  $translatedTitle = $translate->translate($title, [
8      'source' => $localeSrc,
9      'target' => $localeTarget
10 ])[ 'text' ];

```

Ukázka kódu 6: Zjednodušené použití balíčku Google/cloud-translate

5.1.3.6 Nahrávání obrázků a zvukových stop

Laravel obsahuje podporu pro správu souborů na různých souborových systémech. Soubory lze ukládat na lokální disk, FTP server, SFTP server či na Amazon S3. Při implementaci jsem již počítal s nasazením projektu pomocí služby Vercel. Tato služba při každém nasazení přemaže starší nasazené soubory, a proto jsem nemohl použít lokální disk. Pro správu souborů jsem se rozhodl využít Amazon S3. Oproti FTP serveru nabízí lehké škálování, vysokou dostupnost a odolnost proti chybám.



Obrázek 5.2: Struktura uložení souborů na Amazon S3

Pro použití souborového systému Amazon S3 jsem nainstaloval balíček `league/flysystem-aws-s3-v3` a nastavil několik proměnných prostředí. Mezi nimi je nastavení autentizačních informací pro přístup k Amazon S3 a také nastavení výchozího disku na S3. Díky nastavení výchozího disku jsem nemusel při každé akci se soubory specifikovat disk.

Pro snadnější správu souborů jsem vytvořil strukturu souborů, která je zobrazena v obrázku 5.2. Díky této struktuře je jednoduché mazat již nevyužité soubory, například při mazání interaktivního prvku lze smazat celá jeho složka a tím se smažou všechny soubory, které se interaktivního prvku týkají (viz ukázka kódu 7). Při duplikaci je také možné jen zkopírovat složku procházky nebo prvku.

```

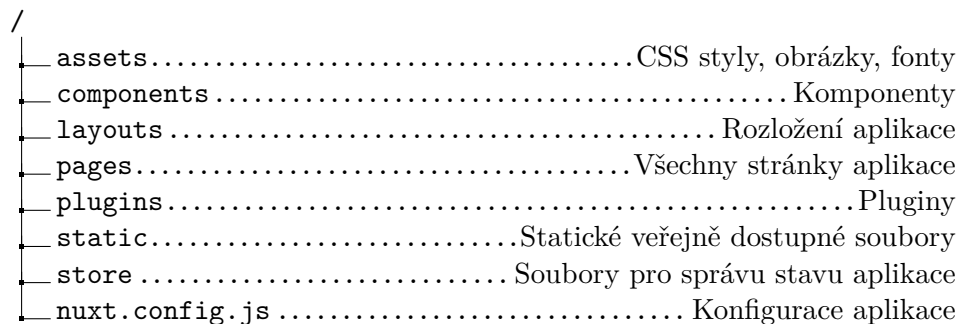
1 public static function boot ()
2 {
3     parent::boot();
4     self::deleting(function (Walk $walk) {
5         Storage::deleteDirectory('walks/'.$walk->id);
6     });
7 }

```

Ukázka kódu 7: Zjednodušená funkce pro smazání souborů při mazání procházky

5.2 Frontend

Frontend aplikace jsem implementoval pomocí frameworku Nuxt. Základní aplikaci jsem vytvořil pomocí příkazu `npm init nuxt-app <project-name>`. Během vytváření základní aplikace se systém ptal na zvolení několika věcí. Mezi nimi byl například název projektu, správce balíčků či framework pro usnadnění tvorby uživatelského rozhraní. Po vytvoření byla základní aplikace připravena pro spuštění a rozšiřování o funkčnosti.



Obrázek 5.3: Struktura frontend části aplikace

5.2.1 Struktura zdrojového kódu

Vytvořená základní aplikace obsahuje jednoduchou strukturu projektu. Stejně jako Laravel ani Nuxt tuto strukturu prakticky nevynucuje, ale pro lepší orientaci v kódu jsem ji dodržel. V obrázku 5.3 jsem popsal strukturu zdrojového kódu frontend části aplikace. V této struktuře jsem zmínil jen soubory a složky, které jsem upravoval.

5.2.2 Použité dodatečné balíčky

Pro zjednodušení práce jsem při implementaci použil následující dodatečné balíčky.

Nuxt-buefy⁸ – Implementace knihovny s responzivními komponentami Buefy.

Nuxt-clipboard⁹ – Balíček pro implementaci funkce kopírování.

Nuxtjs/auth-next¹⁰ – Balíček pro přidání podpory autentizace.

Nuxtjs/device¹¹ – Balíček pro detekci typu zařízení, který uživatel používá pro zobrazení stránky.

Qrcode.vue¹² – Komponenta pro generování QR kódů.

Vue-html-to-paper¹³ – Balíček pro přípravu HTML kódu k tisku.

Vue-qrcode-reader¹⁴ – Balíček s komponenty pro detekování a dekodování QR kódů.

Vue-social-sharing¹⁵ – Komponenta pro sdílení odkazů na sociální síť.

5.2.3 Implementované funkční požadavky

Stejně jako u backendu jsem i na frontendu implementoval většinu funkčních požadavků. V této kapitole popisuji ty zajímavější z nich. Při implementaci uživatelského rozhraní jsem převážně používal již vytvořené komponenty z knihovny Buefy.

5.2.3.1 Registrace a přihlášení

Pro usnadnění implementace autentizace jsem použil balíček **Nuxtjs/auth-next**. Po jeho instalaci jsem v konfiguraci nastavil stránky, na které se má uživatel přeměrovat po odhlášení nebo úspěšné autentizaci a URL na backend. Pro přihlášení a registraci uživatelů jsem vytvořil responzivní stránky s formuláři k vyplnění autentizačních údajů.

Při přihlášení volám funkci `login` s vyplněným emailem a heslem a pro odhlášení funkci `logout`. Pro registraci uživatele musím nejdříve nastavit CORS cookie, poslat požadavek na backend na registraci a poté uživatele přihlásit (viz ukázka kódu 8).

Vzhledem k zařízení uživatele ho po úspěšné registraci či přihlášení přeměruji do klientské nebo admin části aplikace. K určení zařízení uživatele jsem použil balíček **Nuxtjs/device**.


```
1  try {
2    await this.$axios.get('/sanctum/csrf-cookie')
3    await this.$axios.post('/register', this.form)
4    await this.$auth.login({
5      data: this.form
6    })
7    this.$router.push({
8      path: this.$device.isMobileOrTablet
9        ? '/QR-scanner'
10       : '/admin', query: this.$route.query
11    })
12 } catch (err) {
13   : //show error
14 }
```

Ukázka kódu 8: Registrace uživatele

```
1  showNotification (message, type) {
2    this.$buefy.notification.open({
3      message,
4      type,
5      queue: false,
6      duration: 3500
7    })
8  },
9  showNotificationDanger (message) {
10   this.showNotification(message, 'is-danger')
11 },
12 showNotificationSuccess (message) {
13   this.showNotification(message, 'is-success')
14 }
```

Ukázka kódu 9: Funkce k zobrazení notifikací

5.2.3.2 Zobrazení notifikací

Pro zobrazení notifikací v aplikaci jsem využil Buefy komponentu Notification. Kvůli častému použití jsem napsal funkce pro zobrazení notifikací `showNotification`, `showNotificationDanger` a `showNotificationSuccess` (viz ukázka kódu 9). Tyto funkce jsem implementoval jako plugin ve složce `plugins` a ten poté importoval v konfiguračním souboru `nuxt.config.js`.

Díky takovému způsobu implementace jsou tyto funkce dostupné kdekoliv v kódu bez jakéhokoliv dalšího importování.

```
1 <b-tab-item disabled value="addTrans">
2   <template #header>
3     <div
4       class="tab_add_translation"
5       @click.stop
6       @click="isAddTranslationModalActive = true"
7     >
8       Add translation
9       <b-icon
10        icon="plus-circle-outline"
11        size="is-small"
12      />
13     </div>
14   </template>
15 </b-tab-item>
```

Ukázka kódu 10: Úprava chování záložky při použití komponenty Tabs

5.2.3.3 Admin: Překlad procházky a interaktivního prvku

Pro zobrazení překladů procházek a interaktivních prvků jako záložek v admin části aplikace jsem použil komponentu Tabs. Při použití jsem změnil chování jedné záložky po kliknutí, aby místo otevření obsahu záložky bylo otevřeno modální okno s přidáním překladu (viz ukázka kódu 10).

V modálním okně pro přidání překladu k procházce je zobrazen seznam dostupných jazyků. Ten před načtením aplikace získám z backend API endpointu a uložím ho do stavu aplikace. Při jeho zobrazení z něj odebírám jazyky, které již byly pro překlad procházky využity. V modálním okně pro přidání překladu k interaktivnímu prvku je zobrazen seznam jazyků, které byly použity pro překlad procházky bez jazyků, které byly použity pro překlad interaktivního prvku.

Při přidání automatického překladu zasílám text zdroje překladu na backend a získaný text poté vyplňuji do nově přidaného překladu. Pro lehké použití stejného modálního okna při překladu procházek i interaktivních prvků jsem z tohoto modálního okna vytvořil komponentu, kterou poté využívám v editoru procházek a interaktivních prvků.

5.2.3.4 Admin: Přípravení QR kódů pro tisk

Vytvořené QR kódy k procházce a k interaktivním prvkům je potřeba připravit k jejich tisku. Proto jsem vytvořil komponentu se seznamem QR kódů, které

generují pomocí balíčku `Qrcode.vue`.

Při generování QR kódů specifikuji úroveň opravy chyb na Q. Díky tomu bude QR kód možné dekódovat, i když bude 25 % jeho plochy nečitelné. Při této úrovni opravy chyb je ale v QR kódu uloženo mnohem více dat, buňky QR kódu jsou tedy menší a tím může být QR kód těžší k načtení. Toto jsem se snažil kompenzovat vytvořením co nejkratší URL pro prvky. Tuto URL používám pouze k přesměrování uživatele na správné zobrazení naskenovaného prvku (viz ukázka kódu 11).

Pro splnění minimální šířky quiet zone zobrazuji QR kódy v seznamu s odsazeným ohraničením, který je určen pro budoucí ulehčení střihání. Dále je pomocí balíčku `vue-html-to-paper` tato komponenta se seznamem QR kódů připravena pro tisk. Při tisku se na jednu A4 stránku vejde devět QR kódů.

```

1 router: {
2   extendRoutes(routes, resolve) {
3     routes.push({
4       path: '/e/:hash',
5       redirect: (route) => {
6         return '/QR-scanner?hash_id=' +
7           ↪ route.params.hash
8       }
9     })
10  }
  }

```

Ukázka kódu 11: Přesměrování z krátké URL pro prvky na správnou URL

5.2.3.5 Admin: Sdílení procházky

Pro sdílení procházky pomocí odkazu zobrazuji tento odkaz v textovém poli. V tomto poli nelze odkaz upravit, ale po kliknutí na něj ho pomocí balíčku `nuxt-clipboard` zkopíruji. Pro implementaci sdílení na Facebook či email jsem použil balíček `vue-social-sharing`. Při sdílení na email jsem specifikoval text, předmět a odkaz na procházku.

5.2.3.6 Klient: Navigační tlačítka

V návrhu uživatelského rozhraní klientské části aplikace jsem pro navigaci v aplikaci vytvořil tři navigační tlačítka na spodní části aplikace. Jejich text, odkaz a zvýraznění se mění vzhledem k právě otevřené stránce a stavu aplikace.

Informace o těchto tlačítkách jsou uloženy ve stavu aplikace společně s ID otevřené části aplikace. Tlačítka jsou zvýrazněna, pokud se jejich ID rovná s ID otevřené části aplikace. Díky uložení informací do stavu aplikace je možné

navigační tlačítka vytvořit v rozvržení aplikace a pouze měnit jejich vlastnosti místo vytváření nových tlačítek pro každou stránku aplikace.

```
1  onDecode (decodedString) {
2      const foundWalk = decodedString.match(
3          new RegExp(
4              '(' + this.$config.APP_URL + ')(/walk/\\d+)'
5          )
6      )
7      if (foundWalk && foundWalk.length === 3) {
8          this.$router.push(foundWalk[2])
9          return
10     }
11
12     const foundElement = decodedString.match(
13         new RegExp(
14             '(' + this.$config.APP_URL + '/e/)(.{8})(?=$)'
15         )
16     )
17     if (foundElement && foundElement.length === 3) {
18         this.loadElement(foundElement[2])
19     }
20 }
```

Ukázka kódu 12: Část funkce pro kontrolu obsahu načteného QR kódu

5.2.3.7 Klient: Čtečka QR kódů

Pro implementaci kamery s detekováním a dekodováním QR kódů jsem využil komponentu z balíčku `Vue-qrcode-reader`. U této komponenty lze určit použití přední či zadní kamery, zda a případně jak se má detekovaný QR kód zvýraznit a akci po dekodování QR kódu.

Určení použité kamery jsem nechal na možnosti auto, ve které se balíček pokusí použít zadní kameru, a pokud není dostupná, tak přední kameru. Zvýraznění detekovaného QR kódu jsem nastavil na zobrazení červeného obrysu. Po dekodování QR kódu pomocí regulárních výrazů kontroluji, jestli je obsahem QR kódu odkaz na procházku nebo na interaktivní prvek. Při detekci odkazu na procházku uživatele přesměruji na detail procházky, při detekci odkazu na interaktivní prvek kameru zastavím a zobrazím modální okno s jeho informacemi.

Při používání komponenty z balíčku `Vue-qrcode-reader` jsem narazil na dva problémy. Balíček nejdříve špatně fungoval na zařízeních iPhone. Zobrazená kamera byla mnohdy otočená o devadesát stupňů nebo byla divně

natažená. V dnešní době mají mobilní zařízení několik zadních kamer, proto balíček obsahuje seznam názvů kamer, které nemá používat. V tomto seznamu chyběly přeložené názvy iPhone kamer v češtině. Udělal jsem tedy pull-request na přidání těchto názvů a tím jsem problém opravil.

Dále jsem zjistil, že komponenta přestane detekovat QR kódy, pokud kamera zůstane otevřená více jak 80 vteřin v prohlížeči Chrome na zařízeních s operačním systémem Android. Tento problém také řešili ostatní uživatelé balíčku, ale zatím nebyla vydána oprava. Rozhodl jsem se tedy každých přibližně 80 vteřin kameru v této komponentě vypnout a zapnout, pokud uživatel přistupuje k aplikaci pomocí mobilního prohlížeče Chrome. Pro kontrolu prohlížeče využívám již zmíněný balíček `Nuxtjs/device`.

Při implementaci regulárních výrazů pro kontrolu obsahu QR kódu jsem narazil na problém s použitím vlastností `lookahead` a `lookbehind`. Tyto vlastnosti regulárních výrazů bohužel nejsou implementovány v mobilním prohlížeči Safari. Regulární výrazy jsem tedy přepsal, aby tato vlastnost nebyla použita.

Testování a nasazení

Implementovaná aplikace byla řádně otestována pomocí uživatelského testování. Ohledně uživatelského testování popisují vytvořené scénáře, otázky a získané informace. Aplikaci jsem poté nasadil pomocí služby Vercel.

6.1 Uživatelské testování

Uživatelské testování je proces pro ověření použitelnosti a kvality produktu z pohledu uživatelů, kteří produktu zatím nebyli vystaveni. Při testování se může přijít na mnoho nedostatků a chyb, na které by se jinak mohlo přijít až po vydání produktu. Mezi tyto nedostatky může patřit například neintuitivní prostředí, špatně označená tlačítka či chybějící vlastnosti.

Pro správné testování je dobré připravit scénář, ve kterém se specifikují důvody použití a očekávaný cíl, aby se uživatel lépe vžil do situace používání aplikace. Během testování je také nutné uvést úkoly, které musí uživatel splnit. Úkoly by neměly být moc specifické, aby uživatel používal aplikaci spíše podle jeho intuice a nejednal pouze podle zadaných kroků. Na konec testování se musí připravit dotazník pro získání zpětné vazby od uživatele. V tomto dotazníku může být například otázka na spokojenost s designem aplikace nebo na obtížnost zadaných úkolů [13].

Při testování se lze rozhodnout mezi různými typy testování. Testování lze provést osobně či na dálku. Při testování osobně je nutné připravit místnost pro uživatele a jeho používání aplikace nejlépe sledovat a zapisovat si poznámky. Při testování na dálku bude uživatel testovat aplikaci sám a jeho používání uvidíme pouze z nahrané obrazovky [14].

Uživatele je také nutné před testováním seznámit s cíli testování a přesvědčit ho, že pokud bude mít problém s používáním aplikace, tak je problém v designu aplikace, a ne v jeho schopnostech.

Pro své uživatelské testování jsem vybral testování na dálku. Při osobním testování by mohl být uživatel více nervózní a mohl by mít sklony se hned

doptávat, místo přemýšlení o využití aplikace. Hlavním cílem mého testování je objevení chyb a zjištění, zda je aplikace intuitivní.

Uživatelského testování se zúčastnilo osm uživatelů. Uživatelé tuto aplikaci před testováním neznali, byli tedy vhodnými uživateli pro zjištění míry intuitivnosti aplikace.

6.2 Admin

V aplikaci lze vytvořit procházky pro různé použití. Pro testování jsem vymyslel scénář, ve kterém má uživatel vytvořit jednoduchou procházku pro muzeum. Uživatelské testování admin části aplikace jsem vytvořil ve službě Loop11 [15].

6.2.1 Scénář

Jste zaměstnancem malého muzea, do kterého momentálně nechodí mnoho lidí i přesto, že obsahuje přesně 52 knížek o prokrastinaci. Přišel za Vámi šéf s nápadem vytvoření interaktivních prvků pro lepší zabavení návštěvníků. Šéf Vám sepsal informace v angličtině o muzeu a dvou interaktivních prvcích, které by chtěl v muzeu mít. Informace o muzeu je nutné mít v českém a anglickém jazyce. Vytvoření procházky a interaktivních prvků máte zadané v následujících třech krocích.

6.2.2 Úkoly

6.2.2.1 Vytvoření procházky se základními informacemi

Vytvořte v aplikaci soukromou procházku, u které nezáleží na pořadí procházení prvků. V informacích vyplňte následující:

- Název procházky: "Museum of procrastination"
- Popis: "This is a very simple walk through our museum to help you be a little more productive in your life."

Dále přidejte český překlad a zvukovou stopu, ve které bude přečtený popis. Očekávaná délka procházky je 10 minut.

6.2.2.2 Vytvoření interaktivních prvků

K vytvořené procházce vytvořte dva veřejné interaktivní prvky podle následujícího zadání. První prvek má být u expozice s knihovnou, ve které jsou knihy o prokrastinaci.

Chcete se dotázat návštěvníků na přesný počet knížek o prokrastinaci v této knihovně, správná odpověď je 52. Interaktivní prvek by měl být časově limitován na 1 minutu, aby návštěvníci měli pořádnou výzvu. Počet bodů za

správné splnění prvku má být 50 a počet pokusů 2. Informace v angličtině jsou následující:

- Název prvku: "Procrastination books"
- Otázka: "How many books are in the bookcase in front of you?"
- Popis: "There are at least 5 rows of books in the bookcase, you should count fast!"
- Zpráva po správném zodpovězení: "That is correct!"
- Zpráva po špatném zodpovězení: "That's not the correct answer, but at least you tried!"

Druhý prvek má být u exponátu, který obsahuje počítač na stole a postel. V popisu má být popsána imaginární situace, ve které návštěvník musí psát diplomovou práci. Naskenováním správného QR kódu si má vybrat, zda začne psát diplomovou práci nebo si raději na chvíli lehne.

Počet bodů nastavte na 100 a počet pokusů na jeden. Informace v angličtině jsou následující:

- Název prvku: "Procrastination test"
- Popis: "Imagine you are a student that has to finish writing his master's thesis because the deadline is this midnight. In front of you are two options, will you scan the QR code on the table and start working or will you scan the QR code on the bed and lay down for a bit?"

QR kód na posteli nemá být správnou volbou a má mít následující informace:

- Název: "Bed"
- Zpráva po naskenování: "Maybe you thought that a little break will help? Sadly you fell asleep and overslept the deadline."

QR kód na stole má být správnou volbou a má mít následující informace:

- Název: "Table"
- Zpráva po naskenování: "Yes! You sat down, finished the last few edits and submitted the thesis just in time."

6.2.2.3 Sdílení na Facebook a vytisknutí QR kódů

Nasdílejte vytvořenou procházku na Facebook a nalezněte možnost vytisknutí QR kódů, ale netiskněte je.

6.2.3 Otázky

Po dokončení všech úkolů byly uživatelům zobrazeny následující otázky.

6.2.3.1 Ohodnoťte prosím míru intuitivnosti webové aplikace.

V této otázce měli uživatelé zvolit jednu z následujících možností:

- Velmi intuitivní
- Intuitivní
- Neintuitivní
- Velmi neintuitivní

Většina uživatelů zvolila možnost intuitivní, pouze jeden uživatel zvolil možnost neintuitivní. Uživatelé své zkušenosti s aplikací blíže specifikovali při zodpovídání dalších otázek.

6.2.3.2 Co se Vám na aplikaci líbilo?

Zde uživatelé zmínili možnost automatického vygenerování zvuku z popisu procházky či interaktivního prvku a automatické generování překladů. Dále se jim líbil počet možností u volení typu interaktivního prvku, přehledné zobrazení odpovědí u vytváření otázek, možnost sdílení na Facebook, přímočarost aplikace a fakt, že je to moderní Single Page Aplikace.

6.2.3.3 Co byste na aplikaci vylepšili? Narazili jste při vykonávání úkolů na nějaký problém?

V odpovědích na tuto otázku zmiňovali uživatelé většinou problémy s uživatelskou přívětivostí aplikace. Při testování se narazilo i na pár chyb ohledně funkčnosti aplikace.

Z problémů ohledně uživatelské přívětivosti aplikace zmiňovali uživatelé například chybějící zvýraznění špatně vyplněných informací při editaci procházky či interaktivního prvku nebo chybějící proklik z názvu procházky či interaktivního prvku v přehledu na jejich editaci. Dále bylo doporučeno přidání tlačítka **Uložit a zavřít** k editaci interaktivního prvku a u tlačítka **Uložit** změna jeho funkčnosti, aby se editace po jeho kliknutí nezavřela.

Ohledně funkčnosti aplikace byly problémy se špatným zobrazením vygenerovaného zvuku a s nedopsaným textem notifikace ohledně správného vygenerování překladu.

6.2.4 Shrnutí

Uživatelé převážně hodnotili aplikaci jako intuitivní. Při testování se našlo pár problémů s uživatelskou přívětivostí a s funkčností aplikace. Většinu chyb s intuitivností aplikace jsem opravil, často ještě před tím, než aplikaci stihli otestovat další uživatelé. Z nalezených chyb ohledně funkčnosti aplikace jsem opravil všechny. Uživatelé si navíc zkusili například nahrání obrázku nebo sdílení přes email.

6.3 Klient

Pro správné uživatelské testování klientské části aplikace musí uživatel skenovat QR kódy. Proto jsem se rozhodl nevyužít služby Loop11, ale vytvořil jsem dokument, do kterého jsem QR kódy vložil. Uživatel si tento dokument otevřel na počítači a QR kódy skenoval z telefonu. Pro uživatelské testování jsem vymyslel následující krátký scénář s jednoduchými úkoly a otázkami.

6.3.1 Scénář

Jste návštěvníkem malého muzea o prokrastinaci. Muzeum obsahuje interaktivní procházku a vy jste se rozhodli jí účastnit.

6.3.2 Úkoly

6.3.2.1 Zahájení procházky

U vstupu do muzea o prokrastinaci je následující QR kód pro zahájení procházky. Naskenujte ho mobilním zařízením a zahajte procházku. Procházku si poté uložte do oblíbených a v aplikaci si nastavte preferovaný jazyk na český.

6.3.2.2 Naskenování a vyplnění otázky

U knihovny plné knih o prokrastinaci je následující QR kód. Naskenujte ho a splňte interaktivní prvek.

6.3.2.3 Naskenování správného QR kódu

Přišli jste k expozici, ve které je postel a vedle ní stůl s počítačem. V expozici jsou následující tři QR kódy. První QR kód je u informací k expozici. Druhý QR kód je na posteli a třetí na stole u počítače.

6.3.3 Otázky

Po dokončení úkolů byly uživatelům položeny stejné otázky, jako při dokončení úkolů při testování admin části aplikace.

6.3.3.1 Ohodnoťte prosím míru intuitivnosti aplikace.

Uživatelé měli znovu možnost zvolit ze čtyř možností: Velmi intuitivní, intuitivní, neintuitivní, velmi neintuitivní. Tuto část aplikace hodnotili uživatelé jako velmi intuitivní, pár uživatelů ji ohodnotilo jako intuitivní.

6.3.3.2 Co se Vám na aplikaci líbilo?

Na tuto otázku neměli uživatelé moc specifické odpovědi. Líbilo se jim, že si mohou vytvořené procházky lehce projít pomocí využití mobilního zařízení. Jako kladnou stránku aplikace zmínili například implementaci kamery pro skenování QR kódů přímo v aplikaci.

6.3.3.3 Co byste na aplikaci vylepšili? Narazili jste při vykonávání úkolů na nějaký problém?

I při testování klientské části aplikace se přišlo na pár drobných chyb ohledně uživatelské přívětivosti. Nejvíce uživatelé zmiňovali nutnost registrace přes email a heslo. Uvítali by možnost přihlášení například pomocí Google či Facebook účtu. Mezi další drobné chyby patřilo například přetékání pár textů na určitých mobilních zařízeních nebo nepřesné pojmenování části aplikace se seznamem procházek. Dále bylo doporučeno zobrazení nějakého textu při prázdném seznamu procházek.

6.3.4 Shrnutí

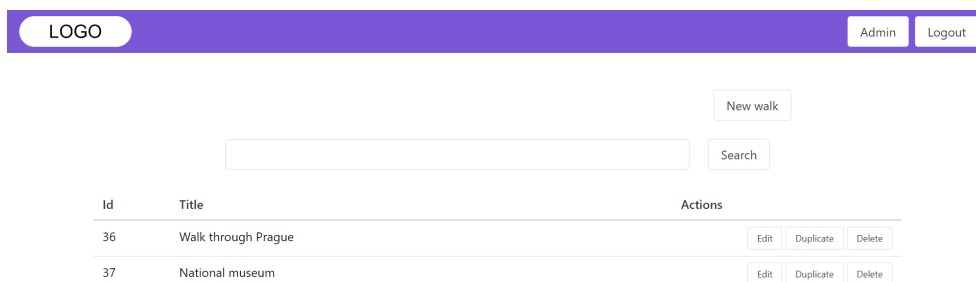
Klientská část aplikace se uživatelům velmi líbila. Všichni ji hodnotili jako intuitivní, většina dokonce jako velmi intuitivní. Všechny chyby s uživatelskou přívětivostí jsem opravil, kromě přidání přihlášení pomocí Google či Facebook účtu.

6.4 Nasazení

Pro nasazení aplikace jsem použil webovou službu Vercel. Tato služba nabízí bezplatné nasazení aplikací po připojení k GitHub repozitáři s kódem aplikace. Pro správné nasazení backendu jsem musel vytvořit konfigurační soubor. V tomto souboru jsem specifikoval vývojový jazyk aplikace, proměnné prostředí a cestu ke vstupnímu souboru. Pro nasazení frontendu stačilo připojení GitHub repozitáře ke službě Vercel. V nastavení projektů ve službě Vercel jsem poté nastavil nasazení na správné domény a určité proměnné prostředí.

Prototyp

V této kapitole představuji vzniklý prototyp aplikace včetně zmínění všech implementovaných funkčních požadavků. Aplikace je dle návrhu rozdělena na dvě části. Admin část je zaměřena na uživatele počítačů a klientská část je zaměřena na uživatele mobilních telefonů.



Obrázek 7.1: Organizace procházek

7.1 Admin

Po přihlášení či registraci na počítači je uživatel přesměrován do admin části aplikace. V této části má přehled vytvořených procházek s možností jejich přidání, editace, duplikace a smazání. Přehled podporuje stránkování a dají se něm vyhledávat procházky podle názvu (viz obrázek 7.1).

Editace procházky je rozdělená na čtyři části. V první části je možné vyplnit základní informace (název, popis, očekávaná časová délka) a přidat jejich

7. PROTOTYP

překlad manuálně, či automaticky z jiného vyplněného překladu. K procházce lze také nahrát zvukovou stopu nebo ji vygenerovat z popisu procházky. Dále lze nahrát úvodní obrázek a po zapnutí plánku i plánek procházky. U procházky lze také nastavit viditelnost a zda je nutné obsažené interaktivní prvky procházet v určeném pořadí (viz obrázek 7.2).

The screenshot displays the 'Editor procházek' (Walkthrough Editor) interface. At the top, there is a purple navigation bar containing a 'LOGO' placeholder, 'Admin', and 'Logout' buttons. Below this, there are 'Save and close' and 'Save' buttons. The main content area is organized into tabs: 'Information' (selected), 'Interactive elements', 'Statistics', and 'Actions'. The 'Information' tab is active and contains several settings sections: 'Visibility' with a 'Public' toggle; 'Elements have to be completed in order' with a 'Yes' toggle; 'Expected time in minutes' with a text input field set to '45'; language selection for 'Czech' and 'English'; 'Title' with a text input field containing 'Walk through Prague'; 'Description' with a text area containing 'Experience Prague like never before with an interactive walk-through tour. Explore historic streets, interact with iconic landmarks and learn about the city's culture.'; 'Audio' section with a playback control showing '0:00 / 0:09' and a 'Generate audio from description' button; and 'Main image' section with an upload area and a 'Plan image' toggle set to 'No'. The interface also features a top navigation bar with 'LOGO', 'Admin', and 'Logout' buttons, and a top right area with 'Save and close' and 'Save' buttons. A tabbed interface at the top shows 'Information' as the active tab, with other tabs for 'Interactive elements', 'Statistics', and 'Actions'.

Obrázek 7.2: Editor procházek

V další části je přehled vytvořených interaktivních prvků s možností jejich přidání, editace, duplikace a smazání (viz obrázek 7.3). V tomto přehledu lze také měnit jejich pořadí, pokud je v procházce nastaveno povinné procházení prvků v určeném pořadí. V editaci či přidání interaktivního prvku lze nastavit viditelnost prvku, počet bodů po správném splnění a omezení počtu pokusů a potřebného času. Dále lze nastavit typ prvku mezi informací, otázkou s vyplněním textu, otázkou s vyplněním čísla, otázkou se seřazením odpovědí, otázkou s možnostmi True/False a načtením správného QR kódu z několika. U prvku je možné nastavit základní informace a manuálně přidat jejich překlad. K prvku je také možné nahrát zvukovou stopu či ji vygenerovat z popisu.

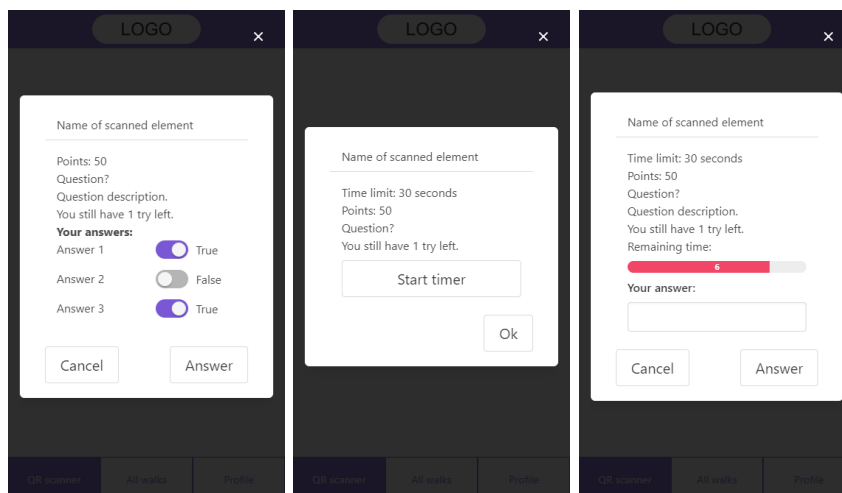
Podle vybraného typu prvku lze dále nastavit text otázky, text odpovědi či text zpráv po správném či špatném splnění prvku (viz obrázek 7.9).

Následujícími částmi je zobrazení statistik (viz obrázek 7.7) a zobrazení možných akcí u procházky (viz obrázek 7.8). V akcích je možné procházku sdílet pomocí odkazu, Facebooku či emailu. V této části je také možné vytisknout QR kódy procházky a vytvořených interaktivních prvků.

Id	Title	Type	Visibility	Actions
116	The post	Question with text	Public	Edit ↑ ↓ Duplicate Delete
117	Elementary school	Scan correct QR code	Public	Edit ↑ ↓ Duplicate Delete
118	National museum entrance	Information	Public	Edit ↑ ↓ Duplicate Delete
119	National museum	Question with True or False	Public	Edit ↑ ↓ Duplicate Delete

Buttons: Save and close, Save, Add new, Admin, Logout

Obrázek 7.3: Organizace interaktivních prvků

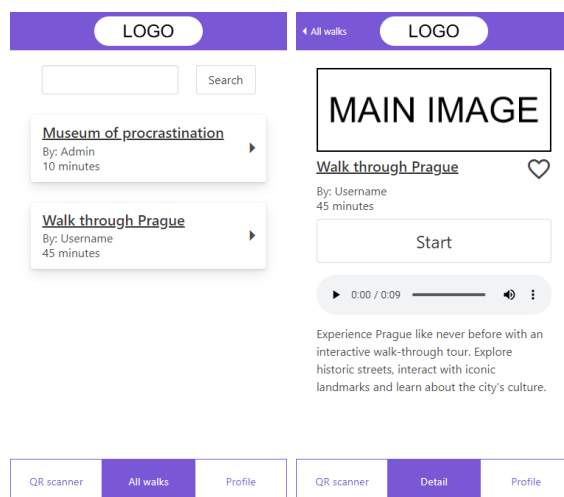


Obrázek 7.4: Zobrazení interaktivních prvků

7.2 Klient

Po přihlášení či registraci na mobilním zařízení je uživatel přesměrován do klientské části aplikace. Tato část je dále rozdělena do tří hlavních částí.

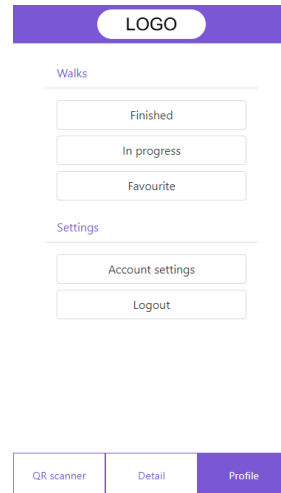
V první části může uživatel skenovat QR kódy. Po naskenování QR kódu procházky je uživatel přesměrován do druhé části aplikace. Po naskenování QR kódu interaktivního prvku je uživateli tento prvek zobrazen (viz obrázek 7.4).



Obrázek 7.5: Přehled a detail procházek

V následující části je uživateli zobrazen seznam procházek a jejich detail (viz obrázek 7.5). V seznamu procházek je možné vyhledávat podle názvu. V detailu procházky jsou zobrazeny základní informace o procházce, včetně nahrané či vygenerované zvukové stopy. Dále je zde zobrazen úvodní obrázek, očekávaná délka procházky, autor procházky a obrázek plánu. V detailu je možné procházku zahájit nebo ji uložit do oblíbených procházek.

Poslední částí je zobrazení profilu (viz obrázek 7.6). Odtud je možné se dostat na seznamy dokončených, zahájených či oblíbených procházek. Dále je zde možnost odhlášení nebo proklik na změnu preferovaného jazyka.



Obrázek 7.6: Profil

The screenshot shows a walk statistics page with a purple header containing a 'LOGO' button, an 'Admin' button, and a 'Logout' button. Below the header, there are two buttons: 'Save and close' and 'Save'. The main content area has four tabs: 'Information', 'Interactive elements', 'Statistics', and 'Actions', with 'Statistics' selected. The statistics section shows the following data:

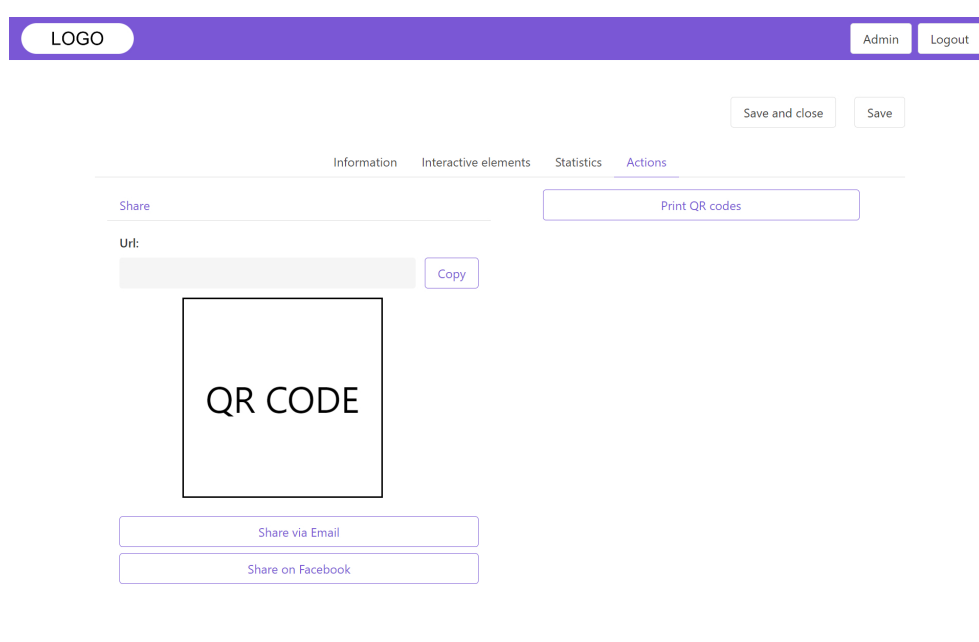
In favourite lists:	5
Participated:	8
Finished:	7
Average score for finished:	100
Average time for finished:	00:01:52

Below the statistics is a table with the following columns: Player name, Score, Duration, Finished, and Date and time ↑.

Player name	Score	Duration	Finished	Date and time ↑
Tomáš P	100	00:02:01	Yes	22.4.2023, 02:10:33
Hung	0	--	No	22.4.2023, 03:08:50
Martin	150	00:01:54	Yes	22.4.2023, 14:34:46
Tomáš H	150	00:01:09	Yes	24.4.2023, 18:56:32
Filip	0	00:01:54	Yes	25.4.2023, 15:09:04
Narek	50	00:01:38	Yes	26.4.2023, 18:36:35
Tomáš D	150	00:01:52	Yes	26.4.2023, 15:24:05
Mike Rotch	100	00:02:34	Yes	29.4.2023, 02:39:53

Obrázek 7.7: Statistika procházky

7. PROTOTYP



Obrázek 7.8: Akce v editoru procházky

The image shows a web-based editor for creating interactive elements. The interface is titled "Add new interactive element" and is set against a dark background. It features several sections for configuration:

- Visibility:** A toggle switch is set to "Public".
- Points for completing:** A toggle switch is set to "Yes" with a text input field containing the value "50".
- Number of tries:** A toggle switch is set to "Limited" with a text input field containing the value "1".
- Type:** A dropdown menu is set to "Scan correct QR code".
- Language:** The language is set to "English" with an "Add translation" link.
- Title:** A text input field is present.
- Description:** A larger text area for entering a description.
- Audio:** A pink message box states "You need to save the element to add audio!".
- Is correct:** A toggle switch is set to "yes".
- Title (secondary):** A text input field for a secondary title.
- Message after scanning:** A text area for a message to be displayed after the element is scanned.

At the bottom right, there are three buttons: "Cancel", "Save and close", and "Save".

Obrázek 7.9: Editor interaktivních prvků

Závěr

V této diplomové práci jsem vytvořil webovou aplikaci pro tvorbu interaktivních procházek s využitím QR kódů a podporou alespoň tří druhů interaktivních prvků pro QR kódy.

Při vytváření webové aplikace jsem postupoval podle cílů v kapitole 1. Hlavním cílem bylo vytvoření webové aplikace pro tvorbu interaktivních procházek. Mezi dílčími cíli bylo seznámení s principy fungování QR kódů, analýza současných přístupů k realizaci interaktivních procházek a analýza možností jejich gamifikace. Dalším dílčím cílem bylo provedení rešerše již existujících řešení a následné vytvoření návrhu UI webové aplikace a její architektury, která bude lehce rozšiřitelná o další druhy interaktivních prvků. Posledními dílčími cíli byla samotná implementace webové aplikace a její testování.

Nejdříve jsem se seznámil s principy fungování QR kódů a jejich využitím. Dále jsem zanalyzoval současné přístupy k realizaci interaktivních procházek a možností jejich gamifikace. Poté jsem provedl stručnou rešerši webových aplikací pro tvorbu procházek. Z analýzy a stručné rešerše jsem sestavil funkční a nefunkční požadavky a funkční požadavky následně rozdělil do čtyř kategorií podle metody MoSCoW. Podle sestavených požadavků jsem vytvořil návrh UI a architektury aplikace. Webovou aplikaci jsem implementoval a otestoval pomocí uživatelského testování. Implementované požadavky jsem shrnul v kapitole 7 Prototyp.

Požadavky ze zadání diplomové práce byly úspěšně splněny. Dále byly implementovány všechny funkční požadavky z kategorií must have a should have a většina z kategorie could have metody MoSCoW. Aplikace by šla rozšířit o mnoho dalších funkcností, například o funkční požadavky v kategorii won't have a nesplněné funkční požadavky z kategorie could have metody MoSCoW.

Funkční požadavky, o které by webová aplikace mohla být rozšířena, jsou následující:

F1 Admin

F1.3 Organizace procházek

F1.3.6 Export procházky

F1.3.7 Import procházky

F1.4 Editor procházek

F1.4.9 Nastavení lokace procházky na mapě

F1.4.14 Historie změn

F1.5 Editor interaktivních prvků

F1.5.3 Vygenerování překladu základních informací

F1.5.6 Umístění na mapu

F1.5.11 Přidání penalizace za překročení času

F1.5.13 Nastavení dynamického zobrazení

F2 Klient

F2.5 Zobrazení mapy načtené procházky

F2.6 Přehled procházek

F2.6.3 Mapa procházek

Literatura

- [1] Soon, T. J.: QR code. *Synthesis journal*, ročník 2008, 2008: s. 59–78.
- [2] Groh, F.: Gamification: State of the art definition and utilization. *Institute of Media Informatics Ulm University*, ročník 39, 2012: str. 31.
- [3] Tarr, R.: . Dostupné z: <https://www.classtools.net/QR/index.php>
- [4] Locatify: Locatify. Dostupné z: <https://locatify.com>
- [5] Guide, S.: Smart Guide. Dostupné z: <https://www.smart-guide.org/>
- [6] Actionbound: Actionbound. Dostupné z: <https://en.actionbound.com/>
- [7] Kuhn, J.: Decrypting the MoSCoW Analysis. *The workable, practical guide to Do IT Yourself*, ročník 5, 2009.
- [8] Dunkley, W.: 2016. Dostupné z: https://medium.com/@Future_Friendly/split-stack-development-a-model-for-modern-applications-d7b9abb47bd5
- [9] Abdullah, H. M.; Zeki, A. M.: Frontend and backend web technologies in social networking sites: Facebook as an example. In *2014 3rd international conference on advanced computer science applications and technologies*, IEEE, 2014, s. 85–89.
- [10] Nuxt: Nuxt. Dostupné z: <https://nuxtjs.org/>
- [11] Pahwa, M.: Identification, Authentication, Authorization, and Accountability. Dostupné z: <https://www.mayurpahwa.com/2018/06/identification-authentication.html>
- [12] Lal, N. A.; Prasad, S.; Farik, M.: A review of authentication methods. *vol*, ročník 5, 2016: s. 246–249.

LITERATURA

- [13] Wingify: Usability Testing. Dostupné z: <https://vwo.com/usability-testing>
- [14] Digital, N.: The Ultimate Guide to Remote Usability Testing. Dostupné z: <https://medium.muz.li/the-ultimate-guide-to-remote-usability-testing-734faad239a5>
- [15] Loop11: Online User Testing Made Easy | Loop11. Dostupné z: <https://www.loop11.com/>
- [16] Otwell, T.: Laravel. Dostupné z: <https://laravel.com/>

Seznam použitých zkratk

API Application Programming Interface.

CORS Cross-Origin Resource Sharing.

CSRF Cross-Site Request Forgery.

CSS Cascading Style Sheets.

FTP File Transfer Protocol.

HTML HyperText Markup Language.

HTTP HyperText Transfer Protocol.

JSON JavaScript Object Notation.

ORM Object-Relational Mapping.

PDF Portable Document Format.

QR Quick Response.

SFTP SSH File Transfer Protocol.

SQL Structured Query Language.

UI User Interface.

URL Uniform Resource Locator.

WiFi Wireless Fidelity.

XSS Cross-Site Scripting.

Obsah přiloženého CD

readme.txt	stručný popis obsahu CD
images	
├ design.....	návrh uživatelského rozhraní
└ prototype.....	výsledná podoba prototypu aplikace
src	
├ thesis.....	zdrojová forma práce ve formátu \LaTeX
text	text práce
└ DP_Štrba_Štěpán_2023.pdf	text práce ve formátu PDF