

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Mobilní cestovní deník

Místo této strany bude
zadání práce.

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 21. července 2020

Milan Kohoutek

Poděkování

Tímto bych chtěl poděkovat vedoucímu mé práce Ing. Ladislavu Pešíčkovi za věcné připomínky a čas, který strávil vedením mé práce.

Abstract

The aim of this thesis is to create a mobile travel journal for the iOS platform. The main motivation is to force users to visit important historical monuments in the Czech Republic. Users will be able to easily monitor their progress in the number of visited monuments in the application. The analytical part of the thesis deals with the research of competing applications and the development of mobile applications. The practical part contains the design of the whole system and a description of its implementation. The conclusion includes suggestions for further extensions of the application and describes the testing of the implemented system.

Abstrakt

Cílem této práce je vytvoření aplikace mobilního cestovního deníku pro platformu iOS. Účelem aplikace je motivovat uživatele k navštívení významných historických památek České republiky. Uživatelé svůj pokrok v počtu navštívených památek budou moci v aplikaci jednoduše sledovat. Analytická část práce se zabývá prozkoumáním trhu konkurenčních aplikací a způsoby vývoje mobilních aplikací. Praktická část obsahuje návrh celého systému a popis jeho implementace. Závěr zahrnuje návrhy dalších rozšíření aplikace a je v něm popsáno testování implementovaného systému.

Obsah

1	Úvod	10
2	Konkurenční aplikace	11
2.1	Místopisy	11
2.2	Turistické známky	12
2.3	Wander book – Turistický deník	13
2.4	Klíč k památkám	14
2.5	Hrady a zámky, Maxův průvodce	15
2.6	Shrnutí	16
3	Vývoj aplikací pro operační systém iOS	17
3.1	Přístupy k vývoji mobilních aplikací	17
3.1.1	Nativní aplikace	17
3.1.2	Progresivní webová aplikace	17
3.1.3	Hybridní aplikace	18
3.2	Technologie pro vývoj iOS aplikací	18
3.2.1	Swift	18
3.2.2	Xamarin	19
3.2.3	React Native	19
3.3	Shrnutí	20
4	Návrh systému	21
4.1	Funkce systému	21
4.1.1	Přehled památek	21
4.1.2	Hodnocení památek	21
4.1.3	Seznam navštívených památek	22
4.1.4	Uživatelské účty	22
4.1.5	Načítání obrázků	22
4.1.6	Komunikace klienta se serverem	22
4.2	Wireframing	23
4.2.1	Navigace mezi obrazovkami	25
4.3	Obrazovky aplikace	26
4.3.1	Načítací obrazovka	26
4.3.2	Přihlašovací obrazovka	26
4.3.3	Obrazovka registrace	27
4.3.4	Obrazovka pro obnovení hesla	27

4.3.5	Obrazovka se seznamem památek	28
4.3.6	Obrazovka s mapou	29
4.3.7	Obrazovka s profilem	30
4.4	Modální okna	31
4.4.1	Filtrace a řazení památek	31
4.4.2	Informace o památce	31
4.4.3	Hodnocení památky	32
4.4.4	Informace o aplikaci	32
4.4.5	Žebříček turistů	33
4.5	Ukládání dat v systému	34
4.5.1	Informace o uživateli	34
4.5.2	Informace o památkách	35
4.5.3	Obrázky	36
4.6	Autentizace uživatele	36
5	Implementace systému	38
5.1	Zvolené technologie	38
5.2	Struktura projektu	38
5.3	Mobilní aplikace	39
5.3.1	Struktura aplikace	39
5.3.2	Zvolené knihovny	40
5.3.3	Funkce aplikace	41
5.3.4	Mapový podklad	46
5.3.5	Oprávnění aplikace	46
5.3.6	Offline zápis návštěvy památky	46
5.4	Webový server	47
5.4.1	Struktura serveru	47
5.4.2	Webová aplikace	47
5.5	Zabezpečení	48
5.5.1	Zabezpečení mobilní aplikace	48
5.5.2	Zabezpečení webové aplikace	48
5.5.3	Zabezpečení databáze	49
5.5.4	Zabezpečení komunikace	49
5.6	Komunikace mezi klientem a serverem	50
5.6.1	Validace autentizačního tokenu	50
5.6.2	Přihlášení přes účet v aplikaci	51
5.6.3	Přihlášení přes Facebook	52
5.6.4	Registrace uživatele	53
5.6.5	Zaslání kódu pro obnovu hesla	53
5.6.6	Změna hesla	54

5.6.7	Získání informací o památkách klientem	55
5.6.8	Navštívení památky	56
5.6.9	Získání profilových dat	56
6	Testování	58
6.1	Výkonnostní testování	58
6.2	Testovací scénáře	58
6.2.1	Registrace	58
6.2.2	Obnovení hesla	59
6.2.3	Přihlášení přes účet v aplikaci	60
6.2.4	Přihlášení přes Facebook	60
6.2.5	Zobrazení seznamu památek	62
6.2.6	Zobrazení detailu památky	64
6.2.7	Ohodnocení památky	66
6.2.8	Zobrazení mapy	68
6.2.9	Zobrazení profilu	70
7	Možná rozšíření systému	72
7.1	Rozšíření mimo Českou republiku	72
7.2	Rozšíření typů památek	72
7.3	Autentizace přes další sociální síť	72
7.4	Webová aplikace	72
7.5	Převod aplikace na platformu Android	73
7.6	Publikace aplikace	73
7.7	Stahování dat památek ze serveru	73
8	Závěr	74
	Seznam zkratk	75
	Literatura	76
	Přílohy	79
A	Instalační příručka projektu	79
A.1	Mobilní aplikace	79
A.2	Webový server	81
B	Uživatelská příručka	82
B.1	Přihlášení přes účet v aplikaci	82
B.2	Přihlášení přes účet na Facebooku	82
B.3	Registrace	83
B.4	Obnovení hesla	84

B.5	Seznam památek	85
B.6	Mapa s památkami	87
B.7	Informace o památce	89
B.8	Profil	91
C	Obsah přiloženého CD	93

1 Úvod

Cílem této práce je vytvoření mobilní aplikace, která by měla podpořit aktivní činnost nadšenců pro cestování po České republice a umožnit jim lépe poznat historické památky. Aplikace bude obsahovat přehled turisticky atraktivních míst a zároveň zaznamenává již navštívené památky. Turisticky atraktivní místa v aplikaci ve verzi 1.0 jsou pouze hrady a zámky.

V teoretickém rozboru jsou analyzovány některé konkurenční aplikace s podobným zaměřením. Mezi porovnávanými funkcemi jsou volba mapového rozhraní a jeho optimalizace, přehled navštívených míst a umožnění zpětné vazby k navštívené památce. Následuje analýza vývoje aplikací pro operační systém iOS. Je zde řešena volba přístupu k vývoji a výběr technologie k vývoji mobilní aplikace.

Další kapitola se zabývá návrhem celého systému. Jsou zde popsány hlavní funkcionality systému, specifikace navigace obrazovek, návrh ukládání dat v aplikaci a proces autentizace uživatele.

V kapitole implementace systému jsou kromě popisu jednotlivých funkcionalit popsány zvolené technologie, struktura systému a použité knihovny. Zmíněné je i zabezpečení systému a komunikace mobilní aplikace s webovým serverem.

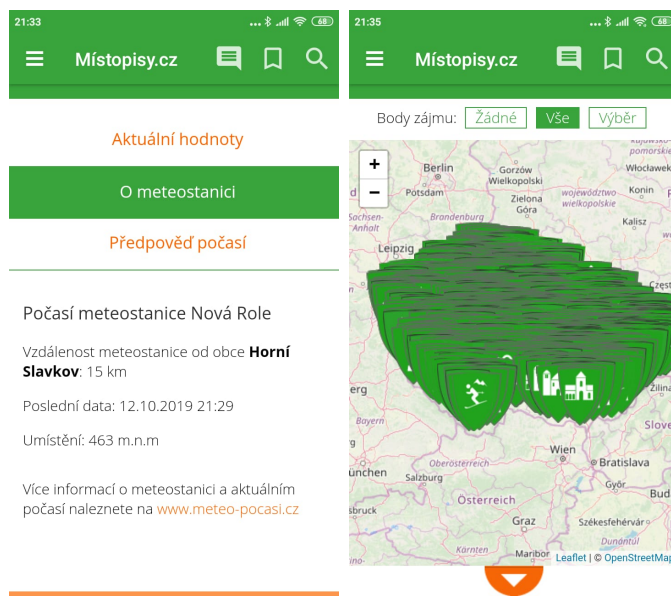
V poslední části práce je popsáno výkonnostní testování a testování pomocí uživatelských scénářů. Také jsou zde náměty na možná rozšíření systému.

2 Konkurenční aplikace

Důležitým krokem je prozkoumat konkurenční aplikace na trhu a porovnat tak náš námět aplikace s ostatními. Bude tak možné se inspirovat o dodatečné funkcionality. Všechny níže zmíněné aplikace mají alespoň 10 000 stažení a jsou zdarma. Vzhledem k možnému pozdějšímu rozšíření aplikace i na platformu Android, budou analyzovány i konkurenční aplikace pro tuto platformu.

2.1 Místopisy

Aplikace Místopisy přináší ucelené informace o všech městech a obcích v České republice. Dále informuje o kulturních a turistických zajímavostech a možnostech sportovního využití. Dále je zde možné vyhledat různé přírodní zajímavosti, případně i ubytování a dopravní spojení. Jednotlivé zápisy jsou rozděleny do kategorií: informace o obci, historie obce, současnost obce, památky a turistické zajímavosti, ubytování a restaurace, sportovní možnosti, okolí a příroda, doprava a statistiky.



(a) Meteostanice

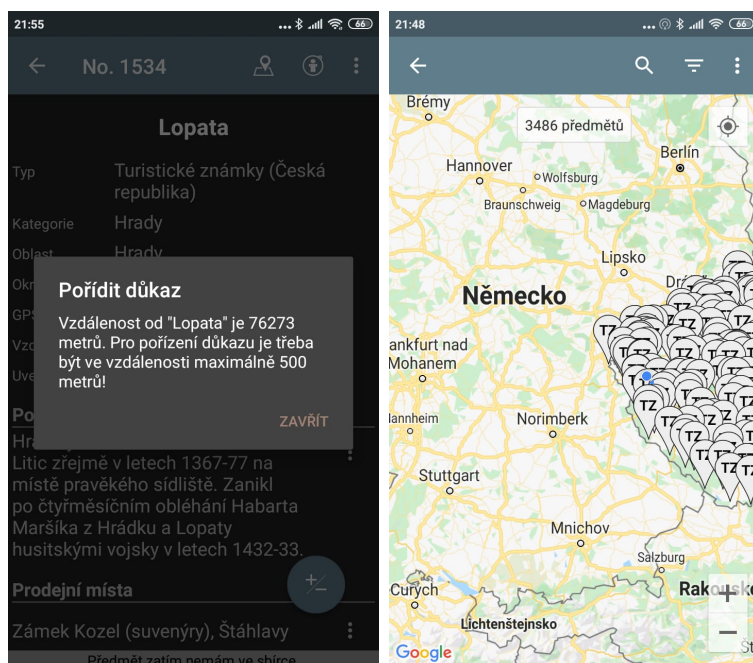
(b) Mapa zájmových bodů

Obrázek 2.1: Aplikace Místopisy

Zajímavým doplňkem aplikace je předpověď počasí a výčet přilehlých me-teostanic pro konkrétní místa viz obrázek 2.1a. Mapa při oddálení vypadá poměrně nepřehledně vzhledem ke stovkám viditelných značek viz obrázek 2.1b. Aplikace je dostupná pouze pro platformu Android.

2.2 Turistické známky

Aplikace Turistické známky pomůže uživateli získat turistické známky¹, ná-lepky a další sběratelské předměty. Uživatel zde také může najít hodnocení daných sběratelských předmětů, případně je sám ohodnotit. Sběrka uživa-tele je vedena v databázi aplikace. Tato databáze je synchronizována mezi webovou a mobilní aplikací [1].



(a) Zaznamenání návštěvy (b) Mapa zájmových bodů

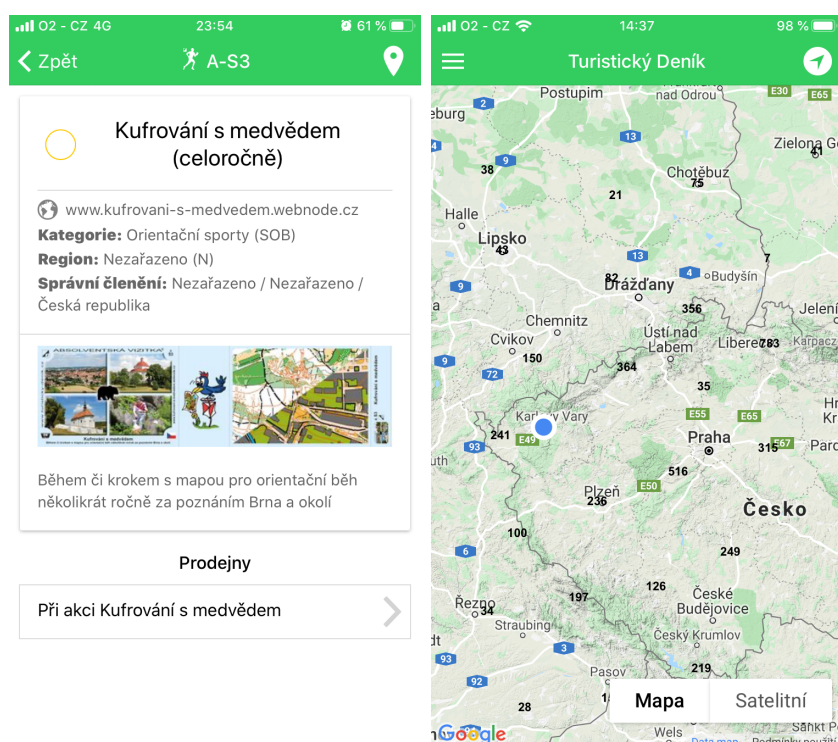
Obrázek 2.2: Aplikace Turistické známky

Pro zaznamenání navštíveného zájmového bodu v aplikaci, od tohoto bodu je zjišťována aktuální vzdálenost polohy uživatele. Pokud se pohybuje ve vzdálenosti menší než 500 metrů od zmíněného bodu, aplikace nám ho do-volá přidat do seznamu navštívených viz obrázek 2.2a. Aplikace je dostupná pouze pro platformu Android.

¹Turistická známka je sběratelským předmětem, jež slouží jako upomínka navštíve-ného turisticky atraktivního místa. Jedná se o kulatý dřevěný předmět. Na našem území je momentálně distribuováno 2 601 turistických známek [1].

2.3 Wander book – Turistický deník

Aplikace Wander book – Turistický deník vyhledává zájmové body v blízkosti polohy uživatele. Po registraci uživatel získá svůj vlastní cestovatelský profil, ve kterém si může zaznamenat všechna navštívená místa. Velkou část aplikace tvoří systém vizitek, jejichž získávání je zpoplatněno. Cena každé vizitky je 15 Kč. Tyto vizitky reprezentují turistické úspěchy uživatelů. Dělí se na čtyři kategorie: turistika, gastronomie, kultura a sport. Zajímavou funkcí je možnost vytvoření vlastní vizitky jakékoliv kategorie, které se dají získat pouze od tvůrce této vizitky. Je zde možnost zakoupení fyzického Turistického deníku (Wander book).



(a) Turistická vizitka

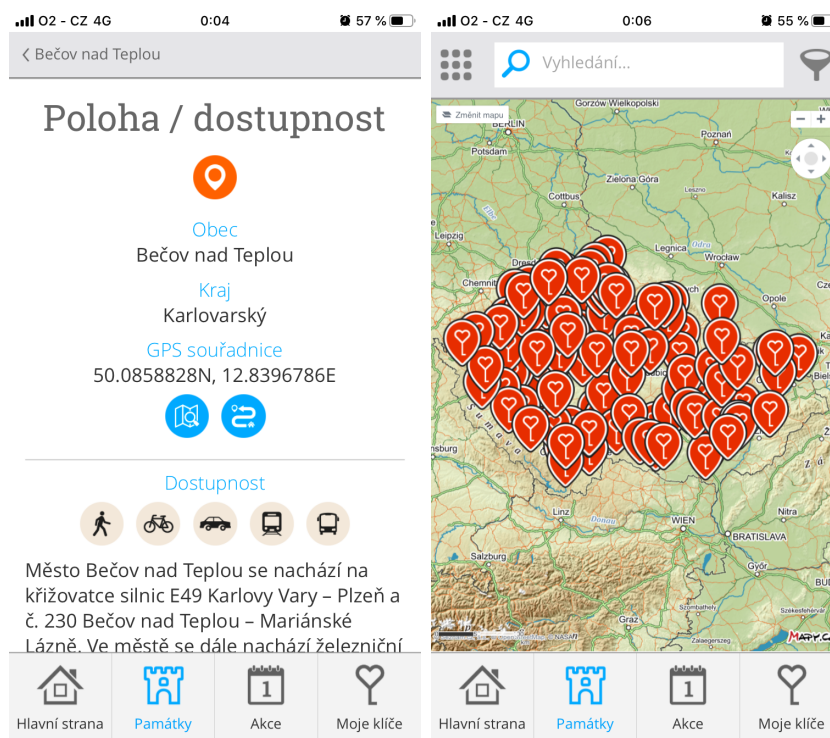
(b) Mapa zájmových bodů

Obrázek 2.3: Aplikace Wander book – turistický deník

Aplikace obsahuje seznam a přehlednou mapu zájmových bodů viz obrázek 2.3b. Po otevření konkrétní značky na mapě uživatel získá informace o prodejnách turistických vizitek, jejich náhled a odkazy na webové stránky s doplňujícími informacemi o získání vizitky viz obrázek 2.3a. Tato aplikace je rozšířena i do zahraničí. Uživatel má možnost si zakoupit vizitky například i v Německu, Rakousku, Polsku, Slovensku či Maďarsku. Aplikace je dostupná pro platformy iOS a Android.

2.4 Klíč k památkám

Klíč k památkám obsahuje informace o významných památkách České republiky. Například o jejich historii, zajímavostech a podrobnosti o kulturních akcích, které se na památkách pořádají. Dále jsou v aplikaci fotogalerie, informace o dostupnosti, službách, prohlídkových okruzích včetně vstupného a otevírací doby. V rámci aplikace má uživatel k dispozici všechny důležité informace pro naplánování výletu i bez přístupu k internetu.



(a) Turistická vizitka

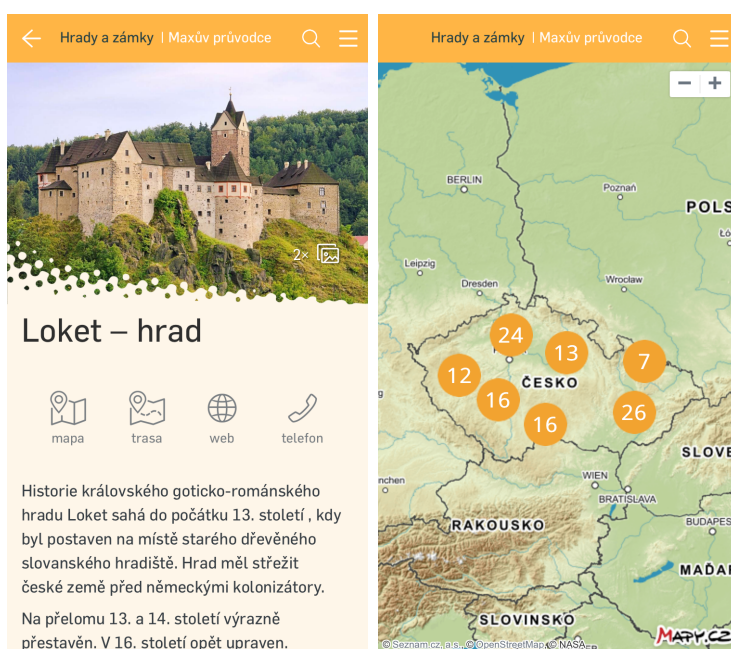
(b) Mapa památek

Obrázek 2.4: Aplikace Klíč k památkám

Za touto aplikací stojí Ministerstvo kultury České republiky, Národní památkový ústav a Asociace majitelů hradů a zámků. Aplikace nabízí věrnostní program, který umožňuje po navštívení pěti památek zahrnutých v aplikaci, získat do šesté vstup zdarma. Na pokladně při zakoupení vstupenky ukážete svůj QR klíč v mobilní aplikaci. Kód bude načten a vstup do památky bude v aplikaci zaznamenán. Aplikace je dostupná pro platformy iOS a Android.

2.5 Hradý a zámky, Maxův průvodce

Aplikace Hradý a zámky, Maxův průvodce je velmi uživatelsky přívětivá. Jednotlivé památky si uživatel může zobrazit jako seznam nebo mapu s vyznačenými body. Řazení je možné dle abecedy nebo vzdálenosti od aktuální polohy. Každý záznam obsahuje popis objektu, kontaktní informace, přednastavenou GPS v mapě pro nalezení trasy a odkaz na webové stránky památky viz obrázek 2.5a. Aplikace obsahuje kvízy a poznávací testy k ověření znalostí. Aplikace zahrnuje všechny hradý a zámky České republiky, avšak základní verze nemá přístup ke všem památkám. Uživatel si tak může dokoupit zbylé památky formou nákupu v aplikaci za 49 Kč pro iOS a 29,99 Kč pro Android.



(a) Informace o památce (b) Mapa hradů a zámků

Obrázek 2.5: Aplikace Hradý a zámky, Maxův průvodce

Při prvním spuštění aplikace je uživatel tázán, zdali chce stáhnout fotografie památek pro využívání aplikace bez připojení k internetu. Aplikace je založena na mapovém podkladu od společnosti Seznam, nicméně navigace ke konkrétní památce je řešena přes přesměrování na Google maps. Nechybí také velmi důležitý prvek pro přehlednost mapy, kterým je značení hradů a zámků a jejich chytré shlukování při oddálení viz obrázek 2.5b. Po výběru značky na mapě se zobrazí okno s názvem památky, jeho obrázkem a vzdáleností od pozice uživatele v kilometrech. Aplikace je dostupná pro platformy iOS a Android.

2.6 Shrnutí

Z analýzy vybraných konkurenčních aplikací vyplývá, že nejbližší k tématu práce má aplikace Hrady a zámky, Maxův průvodce. Tato aplikace je ale spíše informativního charakteru.

Mezi zajímavé funkcionality konkurenčních aplikací bych jistě zařadil informace o meteostanicích a předpovědi počasí pro zájmové body z aplikace Místopisy. Stejnomená aplikace také obsahovala posuvník v kilometrech, kterým se upravoval seznam zájmových bodů (měst, obcí) od aktuální polohy. Aplikace Turistické známky obsahovala polohové potvrzení, kde pro odškrtnutí navštívené památky uživatel musel být do 500 metrů od zájmového bodu. Tato funkcionality bude implementována i v této práci. Designově byla inspirativní aplikace Hrady a zámky, Maxův průvodce, která je velmi uživatelsky přívětivá.

Název aplikace	Operační systém	Mapový podklad
Místopisy	Android	OpenStreet
Turistické známky	Android	Google
Turistický deník - Wander book	iOS	Google
Klíč k památkám	Android, iOS	Seznam
Hrady a zámky, Maxův průvodce	Android, iOS	Seznam

Tabulka 2.1: Přehled konkurenčních mobilních aplikací

V tabulce 2.1 si můžeme všimnout použití rozdílných mapových rozhraní mezi aplikacemi. I přesto, že jsou použita různá mapová rozhraní, každá mapa obsahuje vlastní značky, které po výběru otevřou okénko s dodatečnými informacemi o zájmovém bodu.

Velmi důležitým faktorem přehlednosti aplikace bude chytré shlukování zájmových bodů (značek) na mapě. Příklad jak vypadají mapy se značkami, které se po oddálení neshlukují, můžete vidět u aplikací Místopisy (viz obrázek 2.1b), Turistické známky (viz obrázek 2.2b) a Klíč k památkám viz obrázek 2.4b. Výsledkem je velmi nepřehledný jednolitý shluk značek po celém území.

3 Vývoj aplikací pro operační systém iOS

3.1 Přístupy k vývoji mobilních aplikací

Třemi základními přístupy k vývoji mobilních aplikací jsou: **nativní aplikace**, **progresivní webové aplikace** a **hybridní aplikace**. Správná volba typu přístupu k vývoji je velmi důležitá. Závisí na našich požadavcích a má velký vliv na rychlost a nákladnost vývoje.

3.1.1 Nativní aplikace

Nativní aplikace je vytvořena pouze pro jednu platformu. Aplikace je například dostupná pouze pro iOS. Pro další platformy je nutné vytvořit jinou aplikaci, i přes shodnou funkcionalitu. Nativní mobilní aplikace je vyvíjena dle pokynů pro uživatelské prostředí zvoleného operačního systému [14]. Vzhledem ke stabilním pravidlům je uživatelská interakce intuitivní a aplikace si zachovává konzistentní vzhled. Koncový uživatel se tak ve stejném prostředí rychleji naučí navigovat a používat mobilní aplikaci.

Rozvoj na další platformy je pro standardní nativní aplikace problematický, protože různé platformy nesdílí jednotné vývojové prostředí a programovací jazyk. Pro menší společnosti by multiplatformní aplikace při nativním vývoji byla velkou zátěží, protože každá platforma obvykle vyžaduje vlastní tým pro vývoj. Aplikace se i při totožné funkcionalitě musí znovu programovat a je i znovu testována a udržována.

Hlavní výhodou nativních aplikací je jejich přímý přístup ke konkrétnímu aplikačnímu programovému rozhraní, dále jen API, což umožňuje úzké spojení s platformou. Výsledkem je to, že odezva daných aplikací je zpravidla rychlejší.

3.1.2 Progresivní webová aplikace

Progresivní webová aplikace funguje obvykle na všech platformách jako přizpůsobení webové stránky [2]. Využívá výhod nových funkcí moderních webových prohlížečů, jako jsou service workers a webové aplikační manifesty. Service workers poskytují programovatelnou síťovou proxy ve webovém prohlížeči k obsluze webových (HTTP) požadavků. Service worker se nachází mezi

sítí a zařízením k doplnění obsahu. Service workers používají efektivní cacheovací mechanismus a umožňují používat aplikaci i v případě odpojení od sítě. Webový aplikační manifest je JSON manifest pro centralizovanou definici metadat vývojářem dané aplikace. Manifest umožňuje „nainstalovat“ webovou aplikaci na domovskou obrazovku zvolením možnosti „Přidat na plochu“ při otevření webu [3]. Omezením je nižší možnost spolupráce s hardwarem mobilních zařízení. Mezi známé progresivní webové aplikace patří například AliExpress, Twitter, Forbes nebo Pinterest [4].

Vývoj je založen na běžných webových technologiích jako je HTML5, CSS, JQuery a JavaScript. Výkon progresivní webové aplikace závisí na faktorech, které nejsou plně pod kontrolou vývojáře. Například na zvoleném webovém prohlížeči nebo zdali má uživatel kvalitní internetové připojení.

3.1.3 Hybridní aplikace

Hybridní aplikace je kombinací webových a nativních technologií [5]. Specializované nástroje umožňují vývoj pomocí specifických webových technologií a následnou konverzi na nativní aplikace pro jednotlivé platformy.

Náklady na vývoj hybridní aplikace jsou srovnatelné s náklady na vývoj nativní aplikace, avšak dosáhne se multiplatformního výsledku. Vývoj hybridních aplikací je ve výsledku rychlejší a levnější než vývoj nativních aplikací.

3.2 Technologie pro vývoj iOS aplikací

Aktuálně existují různé technologie pro vývoj mobilních aplikací pro zařízení s operačním systémem iOS. V následujících podkapitolách budou zmíněny dva hybridní a jeden nativní přístup k vývoji mobilních aplikací.

3.2.1 Swift

Programovací jazyk Swift slouží k vytvoření nativní mobilní aplikace pro platformu iOS. Pro použití této technologie je třeba počítač Macintosh s nainstalovaným iOS SDK. Swift vychází z programovacího jazyka Objective-C. Swift byl představen na konferenci WWDC 2014 a byl vyvinut společností Apple pro operační systémy iOS, macOS, watchOS a tvOS [6]. Hlavním vývojovým prostředím pro Swift je Xcode, které je srovnatelné s Android Studiem, jako k nejvíce používanému a podporovanému IDE pro konkrétní platformu.

3.2.2 Xamarin

Xamarin je hybridní vývojový framework aktuálně vlastněný společností Microsoft [7]. Na trhu se poprvé objevil v roce 2011. Používá programovací jazyk C#, který je možné poté kompilovat do nativních aplikací různých platforem. Klíčovou komponentou pro sestavování aplikací pro více platforem je schopnost sdílet kód napříč různými projekty pro konkrétní platformy. To zaručují přenosné knihovny tříd [8]. Mezi oblíbené knihovny patří například SQLite, Json.NET nebo Reactive UI. Xamarin Component Store poskytuje přístup ke komponentám, které mohou být přidány přímo do aplikace z IDE. Mezi tyto komponenty patří ovládací prvky, API webových služeb a další. Komponenty také umožňují integraci populárních backendových řešení, jako je například Salesforce. Výhodou je například propojení s nativními knihovnami, výkonnost velmi blízká nativní, snadná údržba kódu a založení na .NET frameworku. Mezi nevýhody patří například velikost aplikace, kdy nejprimitivnější vytvořená aplikace „Hello World“ má velikost 16 MB. Mezi nejpopulárnější mobilní aplikace v Xamarinu patří Firefox, FOX sports, UPS, WordPress a další [9].

3.2.3 React Native

React Native je populární JavaScriptový framework [10]. Používá se pro vývoj aplikací pro Android, iOS, UWP a webové stránky. Je kombinací hybridního a nativního vývoje. Využívá kvalit obou přístupů. Vývoj je velmi rychlý, neboť aplikace je skládána pomocí předpřipravených komponent, které jsou také samostatně optimalizovány na cílové platformy. To zaručuje rychlost a spolehlivost spojovanou s nativním vývojem. Tento přístup neumožňuje práci s tolika komponenty jako nativní vývoj.

Z hybridního vývoje je zachována také jednoduchá portabilita. Za vývojem React Native stojí společnost Facebook, která ho představila v roce 2015 na F8 – Facebook Developer Conference. Pomocí tohoto frameworku bylo již vytvořeno mnoho významných aplikací, jako třeba mobilní aplikace Facebook, Instagram nebo Skype. Kód psaný v JavaScriptu je následně přeložen do nativních programovacích jazyků, například Java pro Android a Objective-C pro iOS.

Další důležitou vlastností, kterou React Native přebírá z původního frameworku React, je koncept manipulace s DOM. React Native usnadňuje věci tím, že místo renderování celé stránky eviduje změny na DOMu stránky, které jsou uloženy v paměti a poté využívá minimum výkonu na vykreslení pouze těchto změn [11].

Změny v DOM by měly být vázány na změnu dat. React Native toto řeší zavedením životního cyklu jednotlivých komponent, který je pevně definovaný. Hlavní charakteristikou životního cyklu je nové vykreslení komponent na základě změny dat nebo-li stavů, které ho definují.

Výhodou je například rychlejší a levnější vývoj, vysoký výkon, velká podpora komunity nebo možnost jednoduchého propojení s jinými technologiemi. Dále obsahuje funkci Hot Reloading, jež udržuje běh aplikace a zároveň umožňuje nahrávat nové verze zdrojových souborů, které jsme za běhu aplikace upravili [12]. Toto přináší velký vývojářský komfort oproti nativnímu vývoji, kde je nutné zdrojové soubory aplikace vždy znovu kompilovat. Mezi nevýhody můžeme zařadit větší paměťovou náročnost oproti nativní aplikaci.

3.3 Shrnutí

Vzhledem k zachování možného rozšíření pro ostatní platformy není nativní přístup vhodný. Nutnost vlastnit počítač Macintosh s procesorem Intel a nainstalovaným iOS SDK pro vývoj nativní aplikace pro platformu iOS je další výraznou komplikací nativního přístupu.

Z výše zmíněných důvodů tak rozhodování spadá mezi technologie Xamarin a React Native. React Native je oproti ostatním technologiím na trhu celkem nový a od jeho vydání popularita rychle stoupá. Díky komunitě podporující React Native lze stáhnout nespočet komunitou vytvořených komponent a obohatit tak aplikaci o zajímavé funkcionality.

Aplikace bude vyvíjena v jazyce JavaScript za použití frameworku React Native.

4 Návrh systému

V této kapitole jsou popsány funkce systému, specifikace obrazovek, navigace mezi obrazovkami, návrh databázového modelu, ukládání uživatelských dat a udržování dat památek. Dále zde bude navržen průběh autentizace uživatele. Název výsledné aplikace bude Motrjo.

4.1 Funkce systému

Mezi hlavní funkce systému mobilního cestovního deníku patří přehled památek, hodnocení památek a seznam navštívených památek. Další řešené problematiky jsou správa uživatelských účtů, načítání obrázků a komunikace se serverem.

4.1.1 Přehled památek

Uživatel si bude moci zobrazit významné památky v seznamu či na mapě. Seznam bude možné filtrovat a řadit dle vybraných parametrů. Uživateli bude umožněno seznam filtrovat podle konkrétního typu památky nebo podle stavu navštívení památky. Řadit bude možné dle abecedy, hodnocení či vzdálenosti od aktuální pozice uživatele.

Nedílnou součástí bude i mapový podklad s interaktivními značkami památek. Důležitým prvkem mapy bude chytré shlukování značek a jejich zbarvení. Spektrum barev bude červeno-žluto-zelené, kde světle zelená barva shluku památek bude znázorňovat navštívení všech památek v dané oblasti.

Výběrem památky v seznamu, či na mapě, se uživateli zobrazí informační okno památky. Toto informační okno bude obsahovat základní popis, náhledový obrázek, vzdálenost uživatele od objektu, přesměrování pro navigaci, celkové hodnocení, čas kdy uživatel památku navštívil a komentáře uživatelů.

4.1.2 Hodnocení památek

Uživateli bude umožněno své návštěvy ukládat a zároveň památku ohodnotit. Při zaznamenání návštěvy památky bude uživatel vyzván k ohodnocení počtem hvězdiček a zároveň bude mít možnost k hodnocení přidat komentář. Podmínkou k zaznamenání návštěvy bude maximálně 500 metrová vzdálenost od bodu zájmu. Aplikace tedy bude vyžadovat povolení polohového

oprávnění a bude aktualizovat polohu uživatele každou minutu.

Záznamy bude možné uložit i bez přístupu k internetu. Na server se však nahrají až při opětovném připojení k síti.

Pokud by chtěl uživatel své hodnocení nebo komentář změnit, bude tak moci učinit do 24 hodin od posledního zaznamenání návštěvy konkrétní památky. Opětovným odesláním záznamu návštěvy s opraveným hodnocením a komentářem je uživateli přepsán poslední záznam této památky. Navzdory této funkci bude uživateli umožněno zaznamenat více návštěv stejné památky. Podmínkou ovšem bude, že stejnou památku navštíví ve více než 24 hodinovém intervalu od posledního záznamu navštívení této památky.

4.1.3 Seznam navštívených památek

Důležitým prvkem cestovního deníku je přehled všech navštívených památek. Uživatel tedy bude mít možnost si ve svém profilu zobrazit veškeré navštívené památky seřazené dle data navštívení.

4.1.4 Uživatelské účty

Uživatel bude mít možnost vytvoření vlastního účtu pomocí registrace v aplikaci, nebo přihlášením přes Facebook. Účet vytvořený registrací v aplikaci nelze propojit s Facebookovým účtem.

V případě, že uživatel zapomene přihlašovací údaje, bude mu umožněna obnova hesla. Obnova hesla bude probíhat zadáním e-mailové adresy existujícího účtu, na který bude následně odeslán náhodně vygenerovaný ověřovací kód. Tento kód poté zadá v aplikaci zároveň s novým heslem a požadavek o změnu hesla odešle.

4.1.5 Načítání obrázků

Vzhledem k počtu památek a případnému budoucímu rozšíření aplikace i mimo Českou republiku, je vhodné ukládat náhledové obrázky památek mimo aplikaci. Veškeré obrázkové podklady památek budou uloženy na webovém serveru. Aplikace bude pro jejich zobrazení využívat internetové připojení. Výsledná aplikace bude méně paměťově náročná, na úkor zobrazení obrázků památek v offline režimu.

4.1.6 Komunikace klienta se serverem

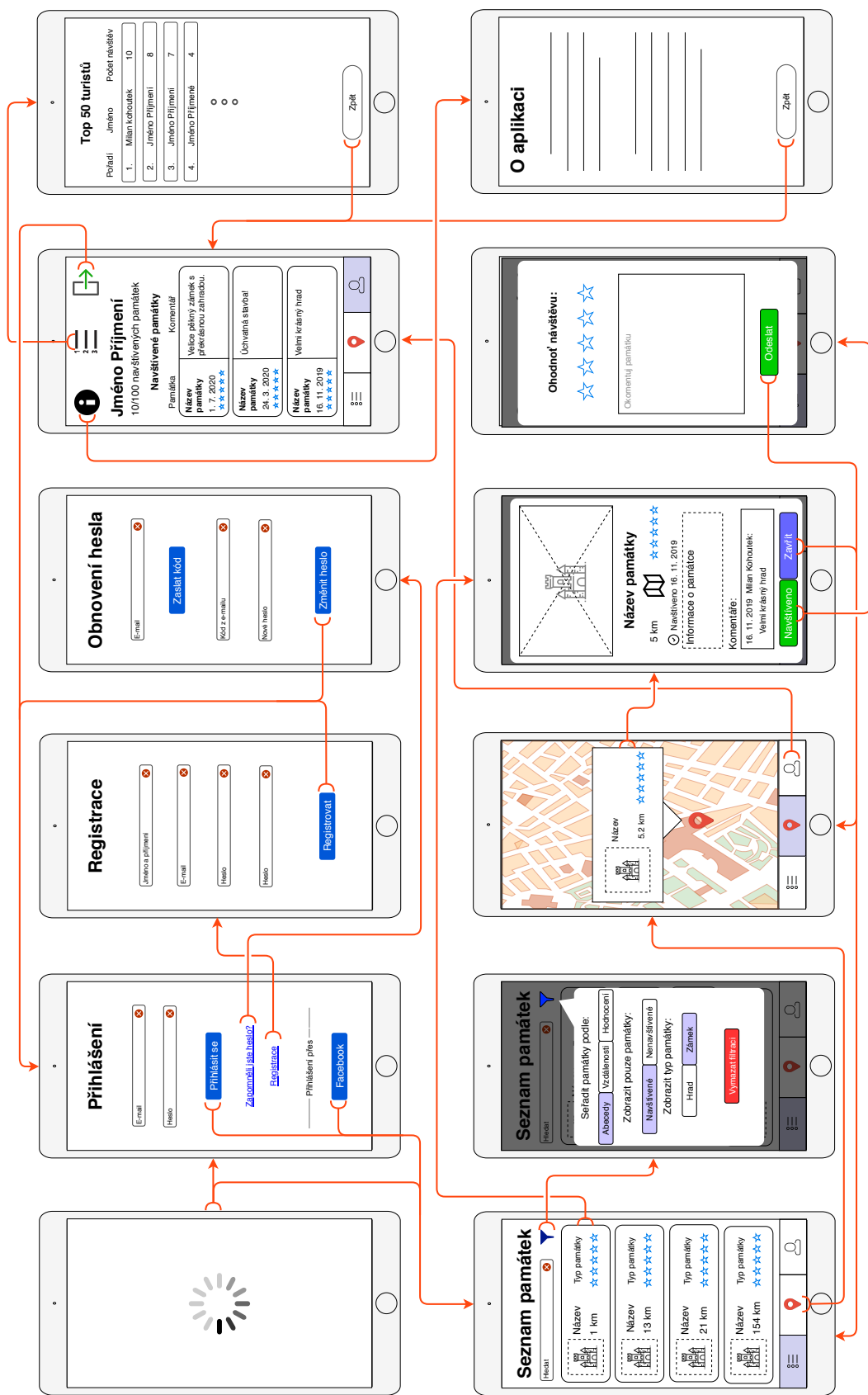
Uživatelská data se budou ukládat na webový server. Tato data obsahují přihlašovací údaje a záznamy navštívených památek. Klient bude odesílat

HTTPS požadavky na webový server, který požadavky obslouží a vrátí klientovi výsledky HTTPS odpovědí. Komunikace bude sloužit pro získání hodnocení a komentářů u památek všech uživatelů, uložení záznamu navštívení, přihlášení, registrace a změna hesla uživatele.

Hesla budou posílána a ukládána v hash podobě (hashovací funkce převádí vstupní data do speciálního řetězce neboli hashe). Komunikace bude zabezpečena protokolem HTTPS. Více o zabezpečení systému v podkapitole 5.5.

4.2 Wireframing

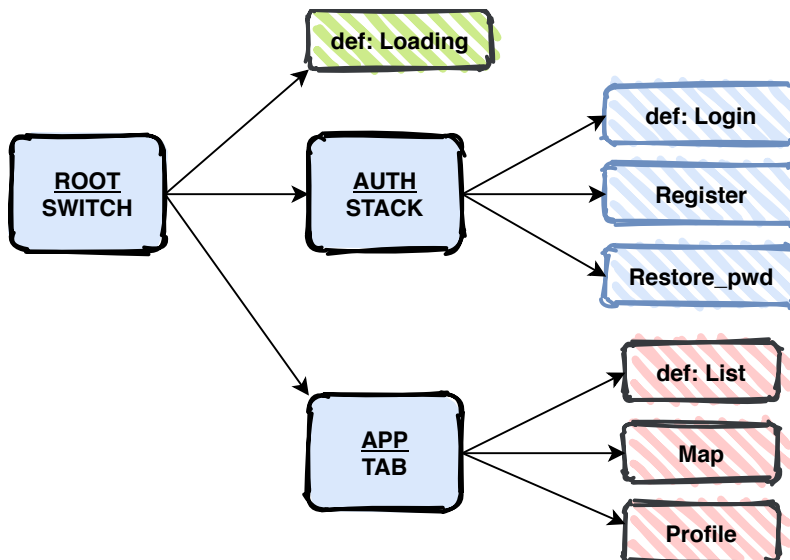
Wireframing, nebo-li specifikace obrazovek, se bude skládat ze sedmi obrazovek a pěti modálních dialogových oken. Tento návrh je znázorněn na obrázku 4.1.



Obrázek 4.1: Wireframing - specifikace navigace obrazovek

4.2.1 Navigace mezi obrazovkami

Mobilní aplikace bude obsahovat tři základní navigace mezi sedmi obrazovkami viz obrázek 4.2.



Obrázek 4.2: Rozvržení navigace mezi obrazovkami

Kořenový navigátor [13] tzv. switch navigator, bude sloužit pro přepínání mezi obrazovkami. Switch navigator udržuje záznam pouze aktivní obrazovky. Nebude umožňovat návrat na předchozí obrazovku. Výchozí hodnota aktivní obrazovky po spuštění aplikace bude obrazovka s načítáním viz podkapitola 4.3.1.

V případě neúspěšné autentizace bude uživatel z obrazovky s načítáním přeměrován na výchozí obrazovku tzv. stack navigator, kterou bude obrazovka s přihlášením viz podkapitola 4.3.2. Uživateli bude umožněno se z této obrazovky přepnout na obrazovku s registrací (viz podkapitola 4.3.3), nebo na obrazovku pro obnovení hesla viz podkapitola 4.3.4. Stack navigator udržuje záznam aktivních obrazovek v přesném pořadí jejich navštívení. Zpětná navigace mezi obrazovkami bude řešena intuitivním gestem – tahem z levé strany obrazovky k pravé straně.

Po úspěšné autentizaci bude uživatel přeměrován na výchozí obrazovku se seznamem památek viz podkapitola 4.3.5. V dolní části obrazovky bude umístěn tzv. tab navigator, který bude nabízet tři možnosti. První možností bude navigace na obrazovku se seznamem památek viz podkapitola 4.3.5. Druhou možností bude navigace na obrazovku s mapou viz podkapitola 4.3.6. Třetí možností bude navigace na obrazovku s profilem viz podkapitola 4.3.7.

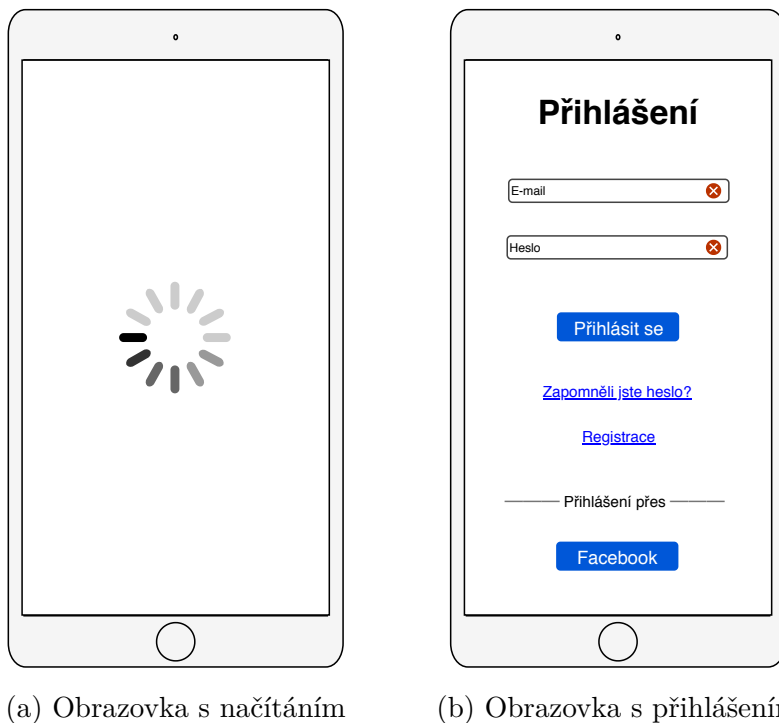
4.3 Obrazovky aplikace

4.3.1 Načítací obrazovka

Po spuštění aplikace bude zobrazena načítací obrazovka viz obrázek 4.3a. Tato obrazovka nebude interaktivní. Bude sloužit pro zaplnění prodlevy při rozhodování, zdali má aplikace v globálním úložišti data, která jsou potřebná pro přístup bez manuálního přihlášení. Podrobněji o procesu autentizace uživatele v podkapitole 4.6. Pokud neproběhne automatické přihlášení, bude uživatel navigován na přihlašovací obrazovku viz podkapitola 4.3.2. Pokud automatické přihlášení proběhne, bude navigován na obrazovku se seznamem památek viz podkapitola 4.3.5.

4.3.2 Přihlašovací obrazovka

Na této obrazovce (viz obrázek 4.3b) se bude uživatel moci přihlásit přes účet vytvořený v aplikaci, nebo přes sociální síť Facebook. Dále zde bude možnost přechodu na obrazovku registrace (viz podkapitola 4.3.3) a možnost přechodu na obrazovku obnovení hesla viz podkapitola 4.3.4. Pokud se uživatel úspěšně přihlásí, bude navigován na obrazovku se seznamem památek viz podkapitola 4.3.5.



Obrázek 4.3: Specifikace obrazovek

4.3.3 Obrazovka registrace

Na obrazovce s registrací po uživateli budeme požadovat jméno, e-mailovou adresu a heslo viz obrázek 4.4a. Po úspěšné registraci bude uživatel navigován zpět na přihlašovací obrazovku viz podkapitola 4.3.2.

4.3.4 Obrazovka pro obnovení hesla

Pro obnovení hesla se bude odesílat vygenerovaný kód na zadanou e-mailovou adresu uživatele. Tento kód bude požadován v aplikaci spolu s novým heslem viz obrázek 4.4b. Po úspěšné obnově hesla bude uživatel navigován zpět na přihlašovací obrazovku viz podkapitola 4.3.2.

The image shows two mobile application screens. Screen (a) is titled 'Registrace' and contains four text input fields: 'Jméno a příjmení', 'E-mail', 'Heslo', and another 'Heslo' field. Each field has a red 'X' icon on the right. Below the fields is a blue button labeled 'Registrovat'. Screen (b) is titled 'Obnovení hesla' and contains three text input fields: 'E-mail', 'Kód z e-mailu', and 'Nové heslo', each with a red 'X' icon. Below the fields is a blue button labeled 'Změnit heslo'. Both screens have a white background and a light gray border, with a small circle at the top center and a larger circle at the bottom center, representing a mobile device interface.

(a) Obrazovka s registrací

(b) Obrazovka s obnovením hesla

Obrázek 4.4: Specifikace obrazovek

4.3.5 Obrazovka se seznamem památek

Tato obrazovka (viz obrázek 4.5) bude sloužit jako úvodní obrazovka po autentizaci uživatele. Bude obsahovat základní seznam památek. Seznam se bude skládat z dlaždic, které budou obsahovat základní informace o památce – náhledový obrázek, typ a název památky, vzdálenost uživatele od objektu a celkové hodnocení uživatelů.

Bude zde možnost seznam filtrovat pomocí textového vstupu.

Pokud uživatel zvolí ikonu filtrace, bude zobrazeno modální okno viz podkapitola 4.4.1. V případě, že uživatel zvolí dlaždici s konkrétní památkou, bude zobrazeno modální okno viz podkapitola 4.4.2.

V dolní části obrazovky bude umístěn tab navigátor viz podkapitola 4.2.1.



Obrázek 4.5: Obrazovka se seznamem památek

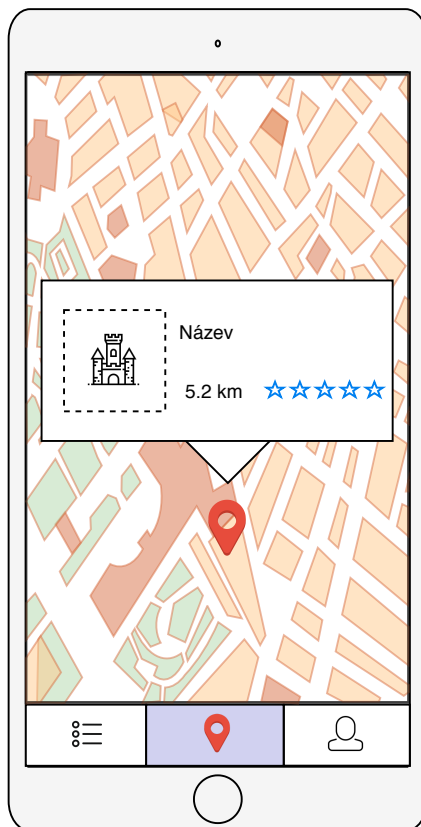
4.3.6 Obrazovka s mapou

Obrazovka s mapou (viz obrázek 4.6) bude obsahovat značky, které budou znázorňovat pozice památek. Při dostatečném oddálení mapy se budou značky památek shlukovat. Tyto shluky budou různě zbarvené dle počtu navštívených památek v dané oblasti.

Po zvolení značky památky bude zobrazen informační box s náhledovým obrázkem, názvem, celkovým hodnocením památky a se vzdáleností uživatele od objektu.

Pokud uživatel zvolí informační box památky, bude zobrazeno modální okno s popisem památky viz podkapitola 4.4.2.

V dolní části obrazovky bude umístěn tab navigátor viz podkapitola 4.2.1.



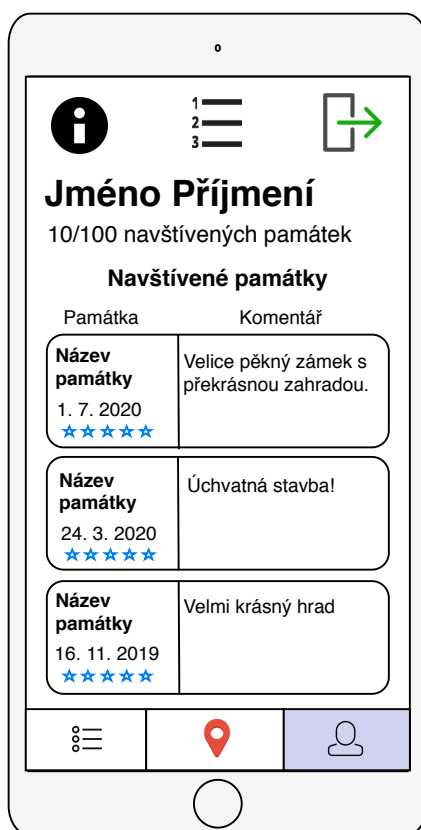
Obrázek 4.6: Obrazovka s mapou

4.3.7 Obrazovka s profilem

Tato obrazovka (viz obrázek 4.7) bude sloužit jako přehled zaznamenaných dat konkrétního uživatele. Především zde bude seznam všech navštívených památek hierarchicky seřazených dle posledního záznamu návštěvy. Jednotlivé dlaždice seznamu budou obsahovat název, datum navštívení, hodnocení a komentář. Dále zde bude uvedeno celé jméno uživatele a počet již navštívených památek.

V horní části obrazovky bude ikona informace, žebříčku a odhlášení se. Po zvolení ikony informace, bude uživateli zobrazeno modální okno s informacemi o aplikaci viz podkapitola 4.4.4. Pokud uživatel zvolí ikonu žebříčku, bude zobrazeno modální okno – žebříček turistů viz podkapitola 4.4.5. Pokud uživatel zvolí ikonu odhlášení se, bude přesměrován na přihlašovací obrazovku viz podkapitola 4.3.2.

V dolní části obrazovky bude umístěn tab navigátor viz podkapitola 4.2.1.



Obrázek 4.7: Obrazovka s profilovými údaji

4.4 Modální okna

4.4.1 Filtrace a řazení památek

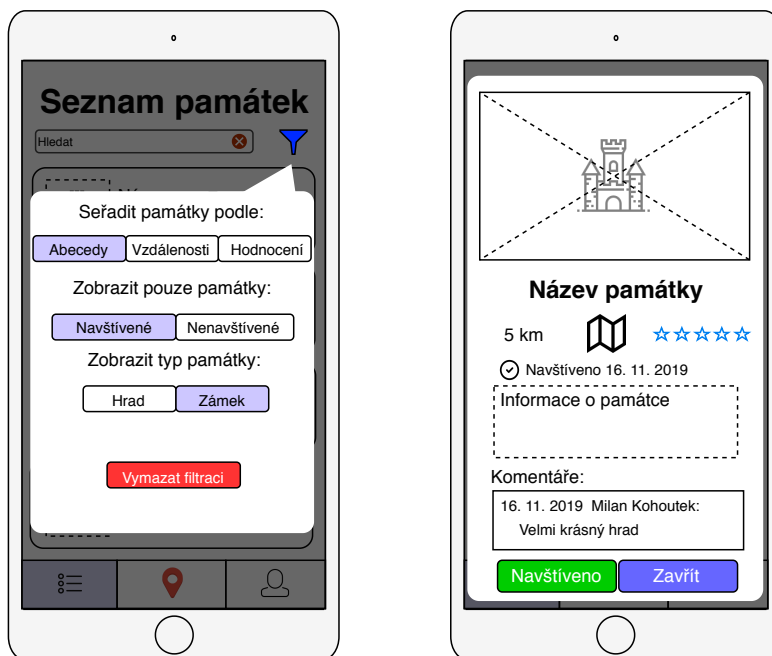
Modální okno s filtrací a řazením (viz obrázek 4.8a) bude obsahovat možnosti seřazení seznamu památek dle abecedy, vzdálenosti od objektu nebo hodnocení. Dále zde bude možnost filtrace seznamu památek dle navštívených či nenavštívených nebo typu památky.

Kliknutím na obrazovku mimo modální okno se toto okno zavře a uživatel bude zpět na obrazovce se seznamem památek viz podkapitola 4.3.5.

4.4.2 Informace o památce

Modální okno s informacemi o památce (viz obrázek 4.8b) bude obsahovat náhledový obrázek, název, celkové hodnocení, vzdálenost od objektu, popis památky, sekce pro komentáře a ukazatel, zdali uživatel vybranou památku navštívil spolu s datem jeho návštěvy.

Pokud uživatel zvolí náhledový obrázek památky, zobrazí se galerie obrázků památky. Pokud uživatel zvolí tlačítko „Navštíveno“, bude zobrazeno modální okno s hodnocením památky viz podkapitola 4.4.3. Pokud uživatel zvolí tlačítko „Zavřít“, bude toto modální okno uzavřeno.



(a) Obrazovka s filtrací a řazením (b) Informace o památce

Obrázek 4.8: Specifikace obrazovek

4.4.3 Hodnocení památky

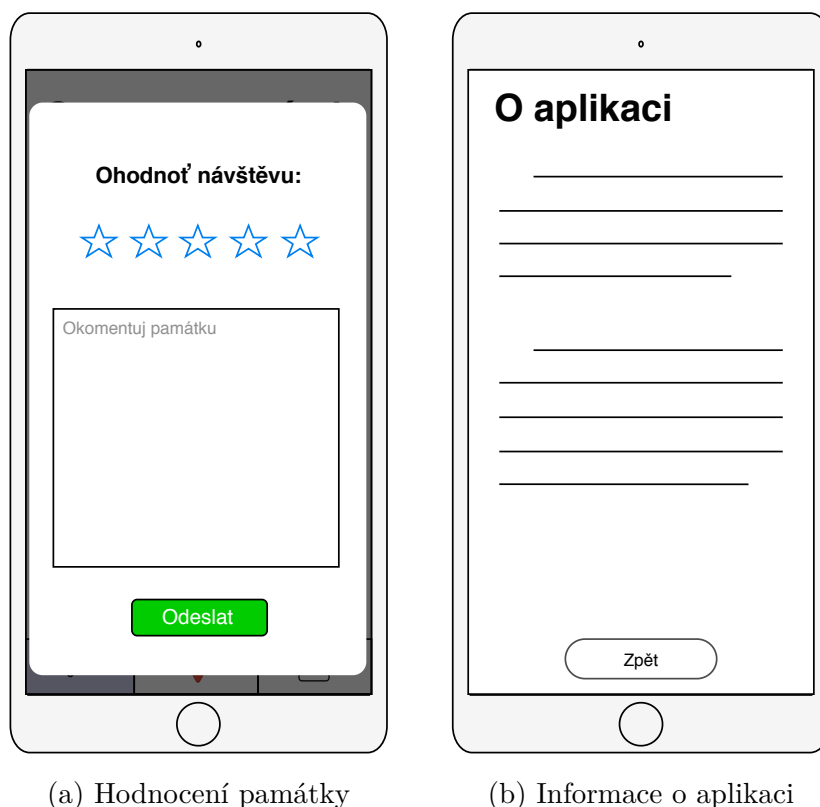
Modální okno s hodnocením památky (viz obrázek 4.9a) bude sloužit pro ohodnocení památky uživatelem. Pro úspěšné zaznamenání návštěvy bude uživatel vyzván k udělení počtu hvězdiček a zároveň bude mít možnost k hodnocení přidat komentář.

Kliknutím na obrazovku mimo modální okno, se toto okno zavře a uživatel bude zpět na modálním oknu – informace o památce viz podkapitola 4.4.2. Pokud uživatel zvolí tlačítko „Odeslat“, proběhne zaznamenání návštěvy a toto modální okno bude uzavřeno.

4.4.4 Informace o aplikaci

Modální okno s informacemi o aplikaci (viz obrázek 4.9b) bude obsahovat text, vysvětlující veškeré možnosti používání mobilní aplikace a dodatečné informace.

Po zvolení tlačítka „Zpět“, se uživatel dostane zpět na obrazovku s profilem viz podkapitola 4.3.7.

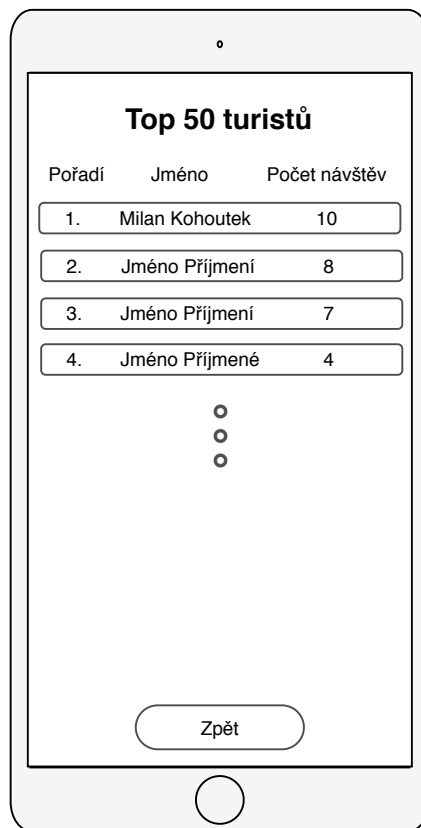


Obrázek 4.9: Specifikace obrazovek

4.4.5 Žebříček turistů

Modální okno s žebříčkem turistů (viz obrázek 4.10) bude obsahovat hierarchický seznam padesáti nejlepších uživatelů v počtu nejvíce navštívených památek. Tento seznam bude sloužit k porovnávání se s ostatními uživateli. Uživatel bude moci vidět svou pozici i přesto, že nebude dosahovat výsledků poslední příčky tohoto žebříčku.

Po zvolení tlačítka „Zpět“, se uživatel dostane zpět na obrazovku s profilem viz podkapitola 4.3.7.

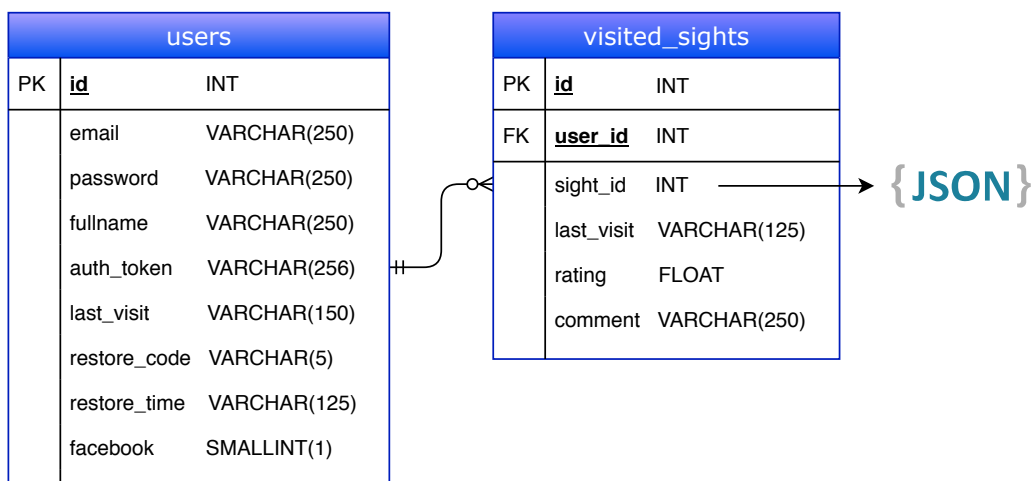


Obrázek 4.10: Žebříček turistů

4.5 Ukládání dat v systému

4.5.1 Informace o uživateli

V databázi na webovém serveru budou uloženy přihlašovací údaje a záznamy o navštívených památkách. Návrh databázového modelu lze vidět viz obrázek 4.11.



Obrázek 4.11: Databázový model

Tabulka *users* bude obsahovat následující informace o uživateli:

- *id* – unikátní identifikátor uživatele,
- *email* – e-mail uživatele,
- *password* – hash hesla uživatele,
- *fullname* – celé jméno uživatele,
- *auth_token* – autentizační token uživatele,
- *last_visit* – posledně zaznamenaný čas úspěšné autentizace uživatele,
- *restore_code* – vygenerovaný kód pro obnovu hesla uživatele,
- *restore_time* – čas vygenerování kódu pro obnovu hesla uživatele,
- *facebook* – booleovskou hodnotu, pro autentizaci přes Facebook.

V případě přihlášení uživatele přes sociální síť Facebook, bude v tabulce *users* hodnota u sloupce *facebook* nastavena na **1**. Zároveň bude sloupec *password* obsahovat hodnotu **null**.

Tabulka *visited_sights* bude obsahovat záznamy uživatelem navštívených památek. Tabulka bude obsahovat tyto sloupce:

- *id* – unikátní identifikátor záznamu návštěvy,
- *user_id* – cizí klíč odkazující na tabulku *users*,
- *sight_id* – unikátní identifikátor památky,
- *last_visit* – čas záznamu návštěvy,
- *rating* – hodnocení počtem hvězdiček (1 až 5),
- *comment* – komentář k navštívené památce.

Unikátní identifikátor památky *sight_id* není označen jako cizí klíč, protože data o památkách jsou uložena v dokumentu JSON v mobilní aplikaci nikoliv v databázi na webovém serveru viz podkapitola 4.5.2.

4.5.2 Informace o památkách

Data o památkách budou uchována v datovém formátu JSON. Tato data budou uložena v mobilní aplikaci. Aplikace nebude načítat informace o památkách ze serveru z důvodu zachování funkčnosti aplikace bez internetového připojení. JSON soubor bude obsahovat pole objektů v tomto formátu:

- *id* – unikátní identifikátor památky,
- *type* – typ památky, hodnota **0** pro hrad, hodnota **1** pro zámek,
- *name* – název památky,
- *description* – popis památky,
- *latitude* – zeměpisná šířka památky,
- *longitude* – zeměpisná délka památky,
- *visited* – **true** pro navštíveno, **false** pro nenavštíveno,
- *last_visit* – pole časů záznamů návštěv památky,
- *rating* – celkové hodnocení památky (1 až 5, 5 nejlepší),
- *rating_count* – počet uživatelů, jež památku ohodnotili,
- *comments* – pole s komentáři uživatelů,
- *images* – pole s odkazy na obrázky památek.

4.5.3 Obrázky

Obrázky památek budou, na rozdíl od ostatních dat památek, uloženy na webovém serveru. Tímto se docílí nižší paměťové náročnosti mobilní aplikace. Nevýhodou bude nemožnost zobrazení obrázků památek při offline režimu.

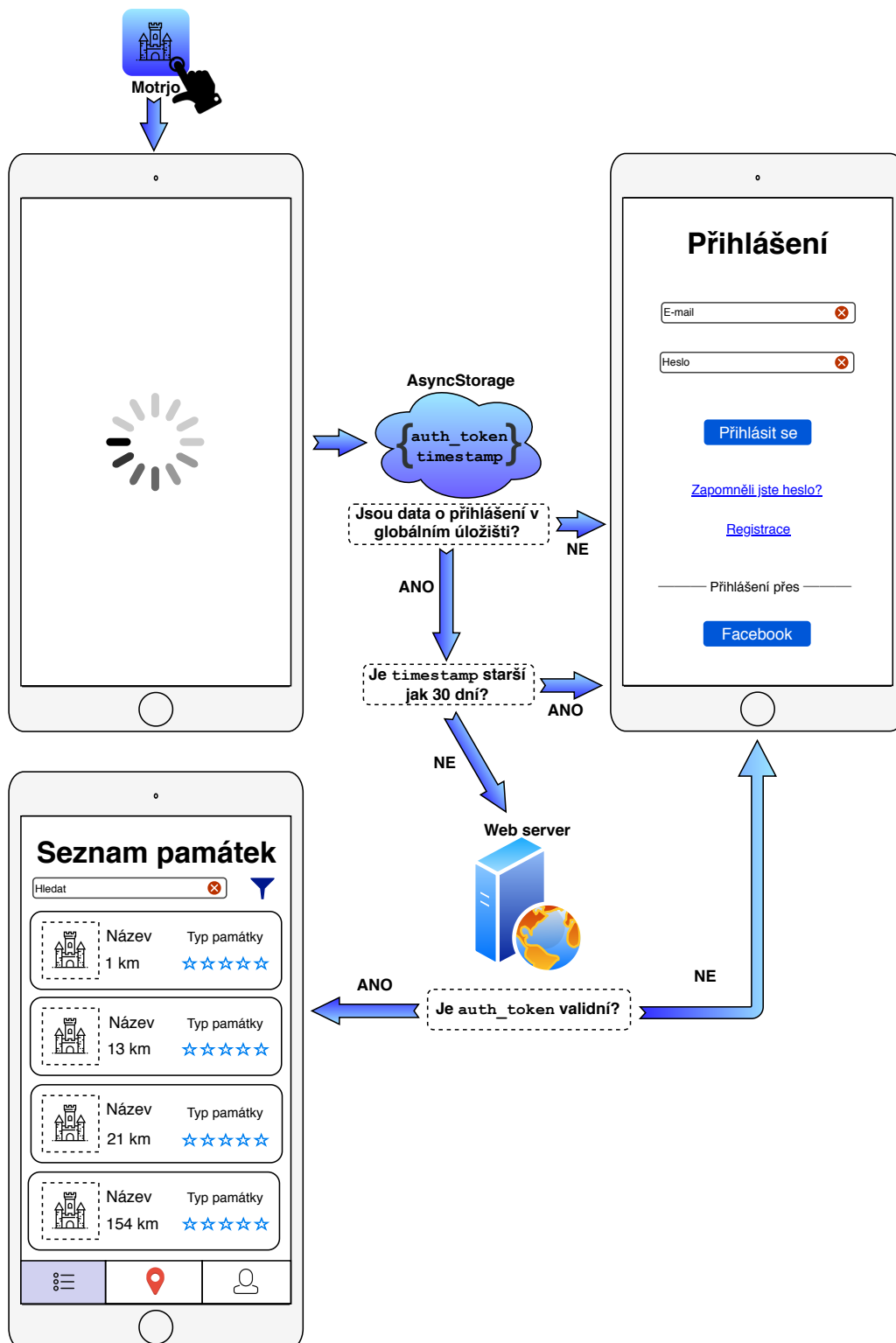
4.6 Autentizace uživatele

Autentizace uživatele (viz obrázek 4.12) bude probíhat při zobrazení obrazovky s načítáním viz podkapitola 4.3.1 [15].

Nejprve se bude kontrolovat, zdali má aplikace uložená autentizační data v perzistentním globálním úložišti tzv. AsyncStorage viz podkapitola 5.3.2. Aplikace si bude do úložiště AsyncStorage ukládat hodnotu *auth_token* (autentizační token) a *timestamp* (datum a čas poslední úspěšné autentizace). Pokud bude úložiště AsyncStorage prázdné, uživatel bude přeměrován na obrazovku s přihlášením viz podkapitola 4.3.2.

Pokud bude v AsyncStorage záznam již existovat, provede se kontrola hodnoty u proměnné *timestamp*. V případě uplynutí více jak 30 dní od poslední úspěšné autentizace bude uživatel přeměrován na obrazovku s přihlášením viz podkapitola 4.3.2.

Jestliže neuplynulo více jak 30 dní od poslední úspěšné autentizace, odešle se autentizační token přes HTTPS požadavek na webový server k validaci. Pokud bude autentizační token validní, server klientovi přes HTTPS odpověď předá pokyn k přeměrování uživatele na obrazovku se seznamem památek viz podkapitola 4.3.5. V případě nevalidního autentizačního tokenu server klientovi přes HTTPS odpověď předá pokyn k přeměrování uživatele na obrazovku s přihlášením viz podkapitola 4.3.2.



Obrázek 4.12: Autentizační proces

5 Implementace systému

5.1 Zvolené technologie

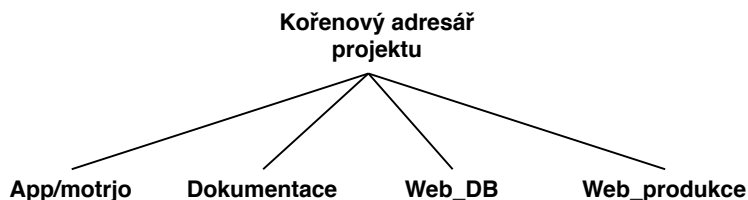
Mobilní aplikace byla programována v jazyce JavaScript za použití frameworku *React native v0.60.22*. Dále byl využit *Node.JS v13.9*, který obsahuje tzv. package system *npm v5.14.4*. Aplikace byla laděna na reálném zařízení iPhone 6s v aplikaci *Expo Client v2.16*. Tato aplikace je zdarma dostupná v Google Play i v App Store.

Webový server k datu 16. července 2020 běží pod serverem Linux u webhostingové společnosti <https://wedos.com/>. Serverová databáze byla zvolena *MariaDB v10.1.37*. Webová aplikace je veřejně dostupná pod doménou <https://kohin.eu/> a byla naprogramována skriptovacím jazykem *PHP v7.1*. Design webové aplikace byl programován v jazyce JavaScript.

Projekt byl zálohován ve službě *GitLab*. Použité vývojové prostředí bylo *Visual Studio Code v1.47*. Databáze byla spravována v *PhpMyAdmin v3.5.8.2*.

5.2 Struktura projektu

Obsah adresářové struktury projektu viz obrázek 5.1:



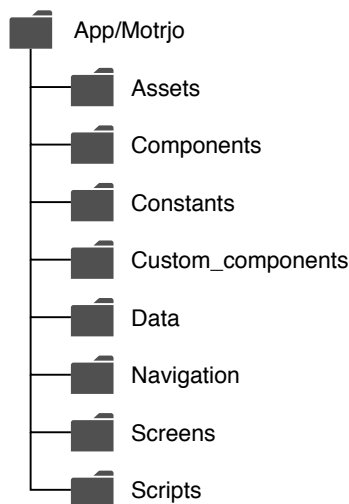
Obrázek 5.1: Adresářová struktura projektu

- **App/motrjo** – zdrojové soubory mobilní aplikace,
- **Dokumentace** – dokumentace bakalářské práce,
- **Web_DB** – SQL soubory pro vytvoření databází,
- **Web_produkce** – zdrojové soubory webové aplikace.

5.3 Mobilní aplikace

5.3.1 Struktura aplikace

Adresář `App/Motrjo` obsahuje zdrojové soubory mobilní aplikace a dodatečné podadresáře viz obrázek 5.2.



Obrázek 5.2: Adresářová struktura mobilní aplikace

Obsah podadresářů:

- `Assets` – obrázky a ikony aplikace,
- `Components` – zdrojové soubory vytvořených komponent,
- `Constants` – konstanty,
- `Custom_components` – modifikované zdrojové soubory komunitou vytvořených komponent,
- `Data` – data památek v souboru JSON,
- `Navigation` – zdrojový soubor sloužící pro definování navigace mezi obrazovkami,
- `Screens` – zdrojové soubory obrazovek aplikace,
- `Scripts` – skript pro přepis zdrojových souborů komunitou vytvořených komponent ze složky `Custom_components`.

5.3.2 Zvolené knihovny

Jednotlivé knihovny se do projektu instalovaly přes package system *npm*. Příkaz pro nainstalování knihovny je: `npm install <název knihovny>`. V následující části jsou popsány knihovny s důležitou funkcionalitou v aplikaci.

React Navigation

Navigaci mezi obrazovkami (viz podkapitola 4.2.1) obsluhují celkem tři knihovny:

- `react-navigation` – *Switch navigator*
- `react-navigation-stack` – *Stack navigator*
- `react-navigation-tabs` – *Tab navigator*

Tyto navigátory jsou taktéž popsány v podkapitole 4.2.1.

AsyncStorage

Knihovna `AsyncStorage` umožní vytvořit pro naši aplikaci úložiště ve fyzickém zařízení, čímž zajistíme perzistenci dat [16]. V tomto úložišti bude pole následujících objektů:

- `sights` – pole dat památek viz podkapitola 4.5.2,
- `visited_sights` – pole záznamů návštěv památek viz podkapitola 5.3.6,
- `auth` – autentizační údaje uživatele viz podkapitola 4.6,
- `list_refresh` – booleovská hodnota pro vynucení stažení aktuálních dat pro seznam památek,
- `map_refresh` – booleovská hodnota pro vynucení stažení aktuálních dat pro mapu,
- `profile_refresh` – booleovská hodnota pro vynucení stažení aktuálních profilových dat ze serveru,

Při spuštění aplikace jsou `auth` data z úložiště vytažena pro následný autentizační proces uživatele viz podkapitola 4.6.

Map Clustering

`React-native-map-clustering` je komunitou modifikovaná knihovna pro zobrazování mapového rozhraní s přidanou funkcionalitou shlukování značek na mapě – *clustering*.

Geolib

Knihovna `Geolib` obsahuje funkci `getDistance`, která vypočte vzdálenost mezi dvěma body v metrech viz ukázka kódu 5.1.

```
1 import { getDistance } from 'geolib';
2 getDistance(userPosition, sightPosition);
```

Kód 5.1: Výpočet vzdálenosti od bodu zájmu

Expo-crypto

Knihovna `expo-crypto` vygeneruje hash ze zadaného řetězce viz ukázka kódu 5.2. V aplikaci je využita šifrovací funkce *SHA-512*. Více o zabezpečení v podkapitole 5.5.1.

```
1 import * as Crypto from 'expo-crypto';
2 const encryptedPassword = await Crypto.digestStringAsync(
3   Crypto.CryptoDigestAlgorithm.SHA512, (password)
4 );
```

Kód 5.2: Vytvoření hashe hesla

Expo-facebook

Knihovna `expo-facebook` poskytuje integraci Facebooku, která je využita pro autentizaci uživatele. Pomocí Graph API Facebooku [17] lze vyžádat například seznam přátel uživatele, jež jsou také registrovaní v aplikaci Motrjo. Tato funkce je však povolena až po ověření aplikace Facebookem.

5.3.3 Funkce aplikace

Načítání

Při spuštění aplikace je zobrazena obrazovka s načítáním. V pozadí aplikace bude probíhat načítání obrázků a autentizace uživatele.

Nejprve jsou načteny veškeré ikonky a pozadí aplikace z lokálního úložiště. Poté je staženo prvních 10 náhledových obrázků památek z webového serveru. Tímto je docíleno plynulejšího chodu aplikace.

Dále zde probíhá autentizace uživatele viz podkapitola 4.6.

Přihlášení přes vytvořený účet v aplikaci

Pro přihlášení přes vytvořený účet v aplikaci jsou potřeba dva údaje. E-mail a heslo zvolené při registraci. Po zadání přihlašovacích údajů dochází k jejich validaci. E-mail je kontrolován regulárním výrazem, zdali se vůbec jedná o e-mailovou adresu. U hesla je kontrolována povolená délka, která byla stanovena na rozmezí mezi 4 až 250 znaky.

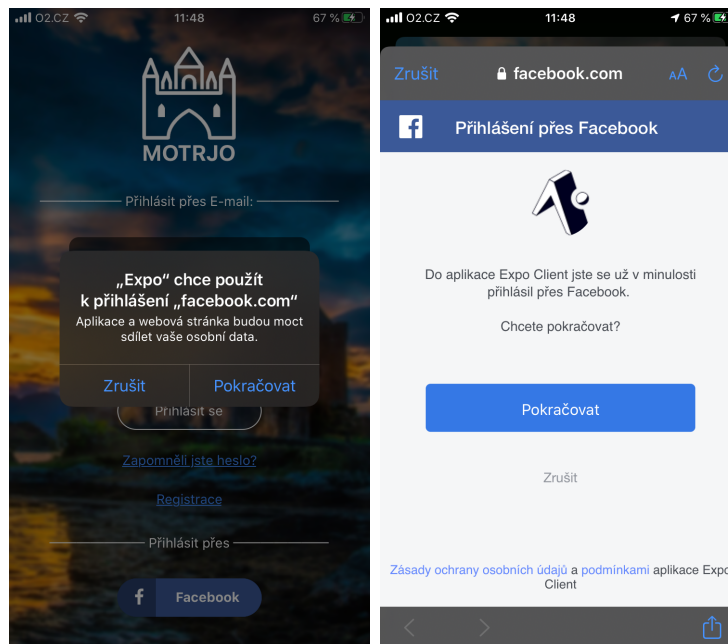
Pokud validace údajů proběhne v pořádku, vytvoří se hash ze zadaného hesla pomocí knihovny `expo-crypto` viz podkapitola 5.3.2. Dále se na webový server odesílá zadaný e-mail spolu s hashem hesla pro jeho validaci. Na webovém serveru se kontroluje shoda hashe hesla s hashem hesla uloženém v databázi. Pokud budou shodné, vygeneruje se autentizační token a aktuální datum s časem viz ukázka kódu 5.3. Tyto údaje se uloží do databáze. Poté jsou klientovi HTTPS odpovědí odeslána autentizační data a pokyn pro přesměrování uživatele na úvodní obrazovku aplikace.

```
1 $auth_token = bin2hex(openssl_random_pseudo_bytes(64));
2 $last_visit = strval((int) (microtime(true) * 1000));
```

Kód 5.3: Vytvoření autentizačního tokenu s aktuálním časem

Přihlášení přes Facebook

Uživatel je vyzván k potvrzení oprávnění přesměrování z aplikace na stránky Facebook viz obrázek 5.3a. Zde je uživatel ušetřen zadáváním přihlašovacích údajů, za předpokladu přihlášeného Facebookového účtu ve fyzickém zařízení. Součástí prvního přihlašování se přes Facebookový účet je udělení oprávnění aplikaci Motrjo. Při následném přihlašování uživatel již potvrzuje přihlášení viz obrázek 5.3b.



(a) Potvrzení přesměrování na jinou aplikaci (b) Potvrzení přihlášení

Obrázek 5.3: Autentizace přes Facebook

Při autentizaci přes facebookový účet jsou využity následující funkce z knihovny `expo-facebook` viz podkapitola 5.3.2.

Prvně je předáván unikátní identifikátor aplikace vygenerovaný Facebookem, což umožňuje přístup k údajům uživatele viz ukázka kódu 5.4.

```
1 await Facebook.initializeAsync(APP_ID);
```

Kód 5.4: Identifikace aplikace Motrjo u Facebooku

Po úspěšné autentizaci uživatele u Facebooku aplikace získá autentizační token uživatele spolu s výsledkem autentizačního procesu viz ukázka kódu 5.5. Pokud výsledek autentizačního procesu uložený do proměnné `type` bude obsahovat řetězec „`success`“, implikuje to úspěšné přihlášení uživatele.

```
1 const {
2   type,
3   token
4 } = await Facebook.logInWithReadPermissionsAsync({
5   permissions: ['public_profile', 'email'],
6 });
```

Kód 5.5: Získání výsledku autentizace uživatele

V neposlední řadě je třeba získat údaje z profilu uživatele. Toho je docíleno pomocí Graph API Facebooku, kterým získáme jméno uživatele a e-mail. Jako parametr pro získání uživatelských údajů je třeba doplnit autentizační token, který byl získán v minulém kroku viz ukázka kódu 5.6.

```
1 if (type === 'success') {  
2   const response = await fetch('https://graph.facebook.com/me  
   ?fields=name,email&access_token=${token}');  
3 }
```

Kód 5.6: Získání uživatelských údajů

Získané údaje uživatele se dále přeposílají k uložení na webový server. Na webovém serveru je kontrolována existence účtu s těmito údaji. Pokud takový účet ještě není zaznamenán v databázi, bude založen nový účet. V případě existujícího účtu je u takového účtu přepsán autentizační token s časem posledního přihlášení na nově získané údaje. Následně je uživateli povolena HTTPS odpověď navigace na obrazovku se seznamem památek.

Registrace

Pro registraci nového uživatele je třeba vyplnit uživatelské jméno, e-mail a heslo viz podkapitola 4.3.3.

Po zvolení tlačítka „Registrovat“ je spuštěn proces validace vstupních údajů. Je kontrolován správný formát e-mailové adresy a shodnost zadaných hesel. Aplikace vytvoří hash zadaného hesla. Uživatelské jméno, e-mailová adresa a hash hesla jsou odeslány na webový server.

Na webovém serveru proběhne kontrola existence zadané e-mailové adresy. Při nalezené shodě je poslána odpověď klientovi, s informací o existenci zadaného účtu. Pokud takový účet neexistuje, dojde k jeho vytvoření. Klient bude vyrozuměn o úspěšném průběhu registrace HTTPS odpovědí a bude přeměrován na obrazovku s přihlášením.

Změna hesla

Na obrazovce pro obnovení hesla (viz podkapitola 4.3.4) je uživateli umožněno vyplněním e-mailové adresy a zvolením tlačítka „Zaslat kód na e-mail“ odeslání žádosti o obnovu hesla. Prvně proběhne kontrola existence účtu se zadanou e-mailovou adresou. Následně probíhá kontrola hodnoty sloupce *restore_time*. Pokud je hodnota nenulová, bude obsahovat čas posledního odeslání žádosti o obnovu hesla. Pro opětovné vygenerování kódu pro obnovu hesla musí uplynout alespoň 10 minut.

Kód pro obnovu hesla je pětimístný a je složen z písmen a čísel. Takto vytvořený kód je zároveň s aktuálním časem uložen do databáze viz ukázka kódu 5.7. Vytvořený kód je odeslán na přiřazenou e-mailovou adresu přes e-mail vytvořený u webhostingové společnosti.

```
1 $restore_code = substr(bin2hex(openssl_random_pseudo_bytes
    (64)), 0, 5);
2 $restore_time = strval((int) (microtime(true) * 1000));
```

Kód 5.7: Získání dat uživatele ze sociální sítě Facebook

Na e-mail uživatele přijde zpráva ve tvaru: „Kód pro změnu hesla: 49A02“ a výzva o zadání kódu v mobilní aplikaci. Společně s tímto kódem v aplikaci také zadá dvakrát nové heslo. Po zvolení tlačítka „Změnit heslo“ aplikace odešle všechny zapsané údaje na server spolu s vygenerovaným hashem nového hesla. Na serveru se nejprve zkontroluje čas vytvoření kódu, protože kód bude platný pouze po dobu dvou hodin od jeho vytvoření. Pokud kód souhlasí s kódem v databázi a je stále platný, provede se přepis hashe hesla v databázi. Klient dále bude uvědoměn o úspěšné či neúspěšné změně hesla.

Ohodnocení památky

Pro uložení záznamu o navštívení památky je od uživatele vyžadováno ohodnocení navštívené památky počtem hvězdiček. Minimální hodnocení je 1 hvězdička a maximální počet je 5. Toto ohodnocení je nutnou podmínkou pro odeslání záznamu návštěvy na server.

Nemusí však památku ohodnotit jen počtem hvězdiček, ale může i přidat komentář. V komentáři jsou povoleny i emotikony. Tento komentář bude viditelný pro všechny uživatele.

Po výběru tlačítka „Odeslat“ jsou na server odeslány údaje o uživateli, jeho hodnocení, unikátní identifikátor památky a čas návštěvy. Poté je kontrolováno, zdali již není v databázi záznam o návštěvě této památky. Pokud takový záznam v databázi existuje, je porovnán čas poslední návštěvy s časem aktuálního záznamu o navštívené památce. Jestliže je rozdíl mezi časy menší než 24 hodin, uživateli se pouze přepíše hodnocení u záznamu poslední návštěvy památky v databázi. Tímto způsobem je uživateli umožněno si například opravit překlep v komentáři. Pokud však uplynulo více jak 24 hodin od poslední návštěvy stejné památky, bude vytvořen nový záznam návštěvy. Starší záznamy uživatel již nemůže opravovat.

V offline režimu nedojde při uložení ke ztrátě tohoto záznamu. Záznam je uložen do perzistentního globálního úložiště. Z tohoto úložiště jsou zá-

znamy při přechodu do online režimu nahrány na webový server viz podkapitola 5.3.6.

Přehled památek

Uživatel má možnost zobrazení přehledu památek jako seznam nebo jako mapu se značkami památek.

Od původního návrhu obrazovek aplikace bylo přidáno stránkování seznamu. Také byla změněna pozice informačního boxu při zvolení značky na mapě a to z důvodu optimalizace aplikace.

5.3.4 Mapový podklad

V aplikaci je použit mapový podklad *Apple maps*. Mapový podklad lze jednoduše změnit například na *Google maps* předáním providera v komponentě *MapView* viz ukázka kódu 5.8. Mapový podklad *Apple maps* byl zvolen pro jeho velmi plynulý chod.

```
1 import { PROVIDER_GOOGLE } from 'react-native-maps';
2 <MapView
3   ...otherOptions
4   provider={PROVIDER_GOOGLE}
5 />
```

Kód 5.8: Změna mapového podkladu na mapy Google

5.3.5 Oprávnění aplikace

Při prvotním spuštění aplikace je vyžadováno lokalizační oprávnění. Pokud je udělení oprávnění odmítnuto nebude mít aplikace přístup k poloze zařízení. V takovém případě nebude zobrazena poloha uživatele na mapě, nebude zobrazena vzdálenost uživatele od památek a uživateli nebude umožněno zaznamenávat návštěvy památek.

5.3.6 Offline zápis návštěvy památky

Při uložení záznamu o navštívené památce v offline režimu je tento záznam uložen do perzistentního globálního úložiště `visited_sights`.

Po úspěšné autentizaci uživatele je zkontrolováno úložiště `visited_sights`, zdali neobsahuje záznamy o navštívených památkách. Pokud úložiště `visi-`

`ted_sights` obsahuje při této kontrole nějaké záznamy a zároveň je zajištěno internetové připojení, jsou tyto záznamy odeslány na webový server k uložení.

5.4 Webový server

5.4.1 Struktura serveru

Adresář `Web_produkce` obsahuje zdrojové kódy webové aplikace a dodatečné podadresáře viz obrázek 5.4.



Obrázek 5.4: Adresářová struktura webové aplikace

Obsah podadresářů:

- `App_requests` – Zdrojové soubory, které zpracovávají HTTPS požadavky od klienta.
- `Images` – Zde jsou uloženy všechny obrázky památek.

5.4.2 Webová aplikace

Součástí systému je i webová aplikace. V současné verzi 1.0 slouží pouze pro správu komentářů u památek a jako přehled statistických údajů mobilní aplikace. K těmto údajům se však dostane pouze administrátor a to přes přihlášení. Po přihlášení se administrátor dostane na stránku, kde uvidí statistické údaje a tabulku pro správu komentářů. Přihlašovací údaje administrátora jsou v databázi, která nijak nesouvisí s databází, ve které jsou uloženy informace o uživatelích mobilní aplikace. Webovou aplikaci bude vhodné rozšířit o další funkcionalitu viz podkapitola 7.4.

Správa komentářů je umožněna v tabulce, která obsahuje sloupce:

- `id` – unikátní identifikátor komentáře,
- `user_id` – unikátní identifikátor uživatele, jenž komentář napsal,
- `sight_id` – unikátní identifikátor památky,

- `date` – datum zaznamenání komentáře,
- `comment` – komentář uživatele a
- `action` – ve verzi 1.0 zahrnuje pouze možnost smazat komentář. Smazáním komentáře se nesmaže záznam navštívení památky uživatelem, ale pouze text komentáře. Pokud uživatel ohodnotí památku a nenapíše žádný komentář, tento záznam v tabulce vůbec nebude zobrazen.

Kromě tabulky jsou zde také statistické údaje aplikace. Je zde celkový počet registrovaných uživatelů, počet registrovaných uživatelů přes aplikaci, počet registrovaných uživatelů přes Facebook, počet uživatelů přihlášených za posledních 14 dní, počet všech uživatelských hodnocení památek a celkový počet komentářů.

5.5 Zabezpečení

V mobilní i webové aplikaci je možnost přihlašování se přes přihlašovací údaje. Vzhledem k rozšíření práce i o webovou aplikaci, celý systém obsahuje dvě databáze. V databázi pro mobilní aplikaci se ukládají veškerá uživatelská data. Databáze pro webovou aplikaci obsahuje pouze přihlašovací údaje administrátorů. Hesla jsou v databázi uložena jako hash řetězce.

5.5.1 Zabezpečení mobilní aplikace

Při pokusu o přihlášení vytvoří mobilní aplikace hash zadaného hesla, který je odeslán HTTPS žádostí webovému serveru k následné validaci. Validace probíhá kontrolou hashe zadaného hesla s hashem hesla uloženého v databázi pro daného uživatele. Hash zadaného hesla je vytvořen pomocí funkce *SHA-512* [18]. Tato funkce spadá pod skupinu *SHA-2*, která patří mezi bezpečné hashovací funkce [19]. Hashovací funkce je v aplikaci řešena pomocí knihovny *Expo-crypto* viz podkapitola 5.3.2.

5.5.2 Zabezpečení webové aplikace

Při pokusu o přihlášení vytvoří webová aplikace hash zadaného hesla, nyní však pomocí PHP funkce `password_hash` [20]. Tato funkce generuje výstupní hash pomocí *Bcryptu* a *solení*¹. Shoda zadaného hesla s hashem hesla

¹Solení hesel je proces vkládání náhodného řetězce do originálního vstupu, čímž je zamezeno tomu, že útočník nebude moci použít předpočítanou tabulku hesel a hashů

uvedeným v databázi, se ověřuje zavoláním specializované funkce `password_verify` [21].

Výhoda Bcryptu je jeho vysoká výpočetní náročnost a automatické solení. Díky této náročnosti (například 100 ms) je pro útočníka velmi drahé testovat mnoho hesel [22].

5.5.3 Zabezpečení databáze

Webový server, který spravuje příchozí HTTPS žádosti se k databázi připojuje přes uživatelský účet s omezenými právy. Tento účet může pracovat s daty, ale nesmí měnit strukturu tabulek.

Dále je ošetřen SQL injection pomocí funkce `bind_param` [23], která svazuje proměnné se jmennými parametry. Jmenné parametry lze nahradit otázníky viz ukázka kódu 5.9 [24].

```
1 $stmt = $conn->prepare("SELECT id FROM users WHERE email = ?
    AND auth_token = ?");
2 $stmt->bind_param('ss', $email, $auth_token);
3 $stmt->execute();
```

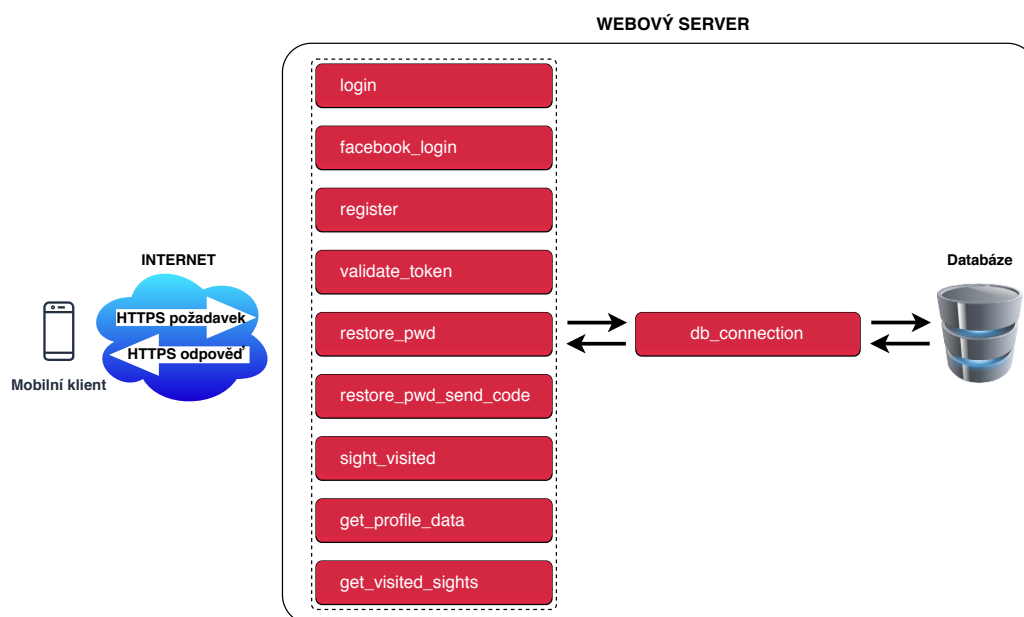
Kód 5.9: Ukázka kódu ošetření SQL Injection

5.5.4 Zabezpečení komunikace

Pro zabránění úniku informací při komunikaci mezi klientem a serverem, je použit protokol HTTPS, který využívá protokol HTTP spolu s protokolem SSL nebo TLS [25].

5.6 Komunikace mezi klientem a serverem

Mobilní aplikace komunikuje se serverem pomocí HTTPS požadavků, které server následně zpracuje a poté odešle zpracovaný požadavek zpět HTTPS odpovědí klientovi. Na server může přijít 9 různých HTTPS požadavků viz obrázek 5.5.



Obrázek 5.5: Komunikace mezi mobilním klientem a webovým serverem

Aplikace si s webovým serverem přeposílá informace v datovém formátu JSON. Požadavky z mobilní aplikace jsou posílány metodou *POST* na adresu serveru s odkazem na zdrojový soubor konkrétního požadavku. Mobilní aplikace odesílá požadavky následujícím způsobem:

```
https://kohin.eu/app_requests/<název požadavku>.php
```

5.6.1 Validace autentizačního tokenu

Zatím co je aktivní obrazovka s načítáním viz obrázek 4.3a, v pozadí aplikace probíhá autentizační proces uživatele. Pokud se jedná o opětovné přihlášení uživatele, je kontrolován čas posledního přihlášení, který je reprezentován hodnotou *timestamp*. Jestliže není tento záznam starší než 30 dní, bude odeslán požadavek na webový server pro validaci uživatelských údajů. Na server se odesílá zpráva v následující podobě:

```
{
  "email": "e-mail",
  "auth_token": "autentizační token"
}
```

Server zkontroluje předané hodnoty a ověří shodnost e-mailové adresy a autentizačního tokenu s hodnotami v databázi. Server může vrátit 4 rozdílné výsledky požadavku:

- **success** – Validací proces byl úspěšný.
- **acc_not_found** – E-mailová adresa s autentizačním tokenem neodpovídá žádnému záznamu v databázi.
- **db_error** – Selhalo připojení k databázi.
- **wrong_inputs** – Požadavek přišel s prázdnými hodnotami u e-mailové adresy nebo autentizačního tokenu.

5.6.2 Přihlášení přes účet v aplikaci

Uživatel v aplikaci vyplní přihlašovací údaje a odešle žádost o přihlášení. Před odesláním požadavku je vygenerován hash hesla, který bude přeposlán webovému serveru společně s e-mailovou adresou uživatele. JSON formát odesílaného požadavku vypadá takto:

```
{
  "email": "e-mail",
  "password": "hash hesla"
}
```

Na serveru proběhne kontrola shodnosti přijatých přihlašovacích údajů s přihlašovacími údaji v databázi. Při nalezené shodě je vygenerován autentizační token a aktuální čas. Tyto dva údaje jsou uloženy do databáze. Klient od serveru může obdržet 5 různých odpovědí:

- **success** – Přihlášení uživatele proběhlo úspěšně. V tomto případě jsou odeslány dodatečné uživatelské údaje:
 - **fullname** – celé jméno uživatele,
 - **auth_token** – vygenerovaný autentizační token,

– `last_visit` – čas úspěšného přihlášení.

- `wrong_pwd` – Hesla se neshodují.
- `acc_not_found` – E-mailová adresa neodpovídá žádnému záznamu v databázi.
- `db_error` – Selhalo připojení k databázi.
- `wrong_inputs` – Požadavek přišel s prázdnými hodnotami.

5.6.3 Přihlášení přes Facebook

Výhoda autentizace přes sociální síť Facebook tkví v absenci nutnosti vytvářet a posílat hash hesla na webový server. Proces autentizace se namísto v aplikaci dělá na straně Facebooku, od kterého je získáván pouze výsledek tohoto procesu. Pokud byl úspěšný, společně se zprávou je obdrženo autentizační token. Token je použit jako parametr při odeslání HTTPS požadavku Graph API Facebooku, který vrátí e-mailovou adresu a jméno uživatele. Uživatelské údaje se poté odesílají na webový server v následující podobě:

```
{
  "email": "e-mail",
  "fullname": "celé jméno",
  "auth_token": "autentizační token"
}
```

Po přijetí těchto dat proběhne kontrola existence účtu se stejnou e-mailovou adresou a s hodnotou sloupce *facebook* nastavenou na *1*. Pokud nenastane taková shoda, přihlašující se uživatel se přihlašuje v aplikaci poprvé, a proto mu bude vytvořen nový uživatelský účet. Pokud je však účet v databázi se shodnými údaji a hodnotou sloupce *facebook* nastavenou na *1*, pouze se aktualizuje hodnota posledního přihlášení a získaný autentizační token od Facebooku. Server může odpovědět třemi různými způsoby:

- `success` – Přihlášení uživatele proběhlo úspěšně. V tomto případě je odeslán údaj s časem přihlášení:
 - `last_visit` – čas úspěšného přihlášení.
- `db_error` – Selhalo připojení k databázi.
- `wrong_inputs` – Požadavek přišel s prázdnými hodnotami.

5.6.4 Registrace uživatele

Uživatel vyplní registrační údaje a tlačítkem „Registrovat“ vytvoří požadavek pro registraci nového uživatele. Prvním krokem tohoto procesu je vytvoření hashe hesla. Na webový server jsou odeslány následující údaje:

```
{
  "fullname": "celé jméno",
  "email": "e-mail",
  "password": "hash hesla"
}
```

Po přijetí žádosti o registraci proběhne na serveru kontrola existence účtu se stejnou e-mailovou adresou. Pokud v databázi neexistuje účet s touto e-mailovou adresou, bude vytvořen nový uživatelský účet. Pokud v databázi existuje účet s touto e-mailovou adresou, proběhne kontrola hodnoty sloupce *facebook*. Hodnota *1* značí existující účet, který je ale založen přes Facebook. V tomto případě bude vytvořen nový uživatelský účet. Hodnota *0* značí existenci uživatelského účtu vytvořeného v aplikaci. V takovém případě nebude účet založen.

Verze aplikace 1.0 umožňuje použití stejné e-mailové adresy pro přihlášení se přes Facebook a zároveň pro registraci účtu vytvořeného v aplikaci. Server může vrátit čtyři různé odpovědi:

- **success** – Nový účet byl úspěšně vytvořen.
- **account_exists** – Účet se stejnou e-mailovou adresou byl již přes aplikaci zaregistrován.
- **db_error** – Selhalo připojení k databázi.
- **wrong_inputs** – Požadavek přišel s prázdnými hodnotami.

5.6.5 Zaslání kódu pro obnovu hesla

Na obrazovce pro obnovu hesla je nejprve třeba zadat e-mailovou adresu. Zvolením tlačítka *Zaslat kód na e-mail* je odeslán požadavek na webový server. Tento požadavek vypadá takto:

```
{
  "email": "e-mail"
}
```

Na serveru je nejprve v databázi kontrolována hodnota *restore_time*. Tato hodnota reprezentuje čas posledního vygenerování kódu pro obnovu hesla. Pokud je čas uplynutý od posledního vygenerování menší než 10 minut, je vygenerování nového kódu zamítnuto. Pokud je čas uplynulší od posledního vygenerování větší než 10 minut, bude vygenerován kód pro obnovu hesla. Tento kód je pětimístný a skládá se z písmen a číslic. Následně je odeslán na zadanou e-mailovou adresu. Server může vrátit celkem 6 různých odpovědí:

- `success` – Kód byl úspěšně odeslán na zadanou e-mailovou adresu.
- `acc_not_found` – Účet nenalezen.
- `spam_timeout` – Kód byl za posledních 10 minut již vygenerován.
- `mail_error` – E-mail s kódem se nepodařilo odeslat.
- `db_error` – Selhalo připojení k databázi.
- `wrong_inputs` – Požadavek přišel s prázdnými hodnotami.

5.6.6 Změna hesla

Po obdržení kódu pro obnovu hesla uživatel tento kód spolu s novým heslem zadá do aplikace. Po zvolení tlačítka „Změnit heslo“ je vygenerován hash ze zadaného hesla. Následně je odeslán požadavek na server v této podobě:

```
{
  "email": "e-mail",
  "code": "kód z e-mailu",
  "password": "hash hesla"
}
```

Po přijetí tohoto požadavku je kontrolována existence účtu vytvořeného v aplikaci. Pokud takový účet existuje, zkontroluje se hodnota sloupce *restore_time*² a *restore_code*³. Kód je platný po dobu dvou hodin od jeho vygenerování. Shodnost zadaného kódu s kódem v databázi umožní změnu hesla. Server může vrátit celkem šest různých odpovědí:

²Čas vygenerování kódu pro obnovu hesla

³Kód pro obnovu hesla

- `success` – Heslo bylo úspěšně změněno.
- `acc_not_found` – Účet nenalezen.
- `wrong_code` – Zadaný kód pro obnovu hesla nebyl správný.
- `code_timeout` – Platnost kódu vypršela.
- `db_error` – Selhalo připojení k databázi.
- `wrong_inputs` – Požadavek přišel s prázdnými hodnotami.

5.6.7 Získání informací o památkách klientem

Po úspěšné autentizaci je uživatel přesměrován na obrazovku se seznamem památek a je odeslán požadavek na server pro získání aktuálních dat památek. Data v tomto požadavku vypadají takto:

```
{
  "email": "e-mail",
  "auth_token": "auth_token"
}
```

Server zkontroluje shodu předaných přihlašovacích údajů uživatele. Pokud kontrola proběhne úspěšně, je z databáze načteno pole uživatelem navštívených památek, pole s hodnocením památek a pole všech komentářů. Server může vrátit celkem čtyři různé odpovědi:

- `success` – Získání dat z databáze bylo úspěšné. Zároveň se přeposílají následující data:
 - `users_visits` – pole obsahující všechny památky, které uživatel již navštívil,
 - `ratings` – pole s průměrným hodnocením památek a počtem uživatelů kteří památku ohodnotili,
 - `comments` – pole památek společně s komentářem, časem kdy byl zapsán a jménem autora.
- `acc_not_found` – Účet nenalezen.
- `db_error` – Selhalo připojení k databázi.
- `wrong_inputs` – Požadavek přišel s prázdnými hodnotami.

Na straně klienta je jednotný seznam památek doplněn o tyto získané hodnoty.

5.6.8 Navštívení památky

Pro uložení návštěvy památky je uživatel vyzván k ohodnocení své návštěvy počtem hvězdiček. Také má možnost přidat k hodnocení komentář. Zvolením tlačítka *Odeslat* je odeslán požadavek o zaznamenání návštěvy památky na server. Tento požadavek obsahuje tato data:

```
{
  "email": "e-mail",
  "auth_token": "autentizační token",
  "sight_id": "unikátní identifikátor památky"
  "last_visit": "čas navštívení památky",
  "rating": "hodnocení",
  "comment": "komentář"
}
```

Na serveru je prvně kontrolována existence účtu. Dále existence záznamu o návštěvě stejné památky. Pokud takový záznam existuje, porovnává se čas tohoto záznamu s aktuálním časem. Pokud je rozdíl mezi těmito časy menší než 24 hodin, přepíše hodnocení, komentář a čas tohoto záznamu údaji z přijatého požadavku. Pokud však existuje záznam se stářím větším než 24 hodin, vytvoří se v databázi úplně nový záznam o navštívení této památky. Stejně tak se vytvoří nový záznam o navštívení památce pokud uživatel u této památky nebude mít zatím žádný záznam. Server může vrátit čtyři různé odpovědi:

- `success` – Záznam návštěvy památky proběhl úspěšně.
- `acc_not_found` – Účet nenalezen.
- `db_error` – Selhalo připojení k databázi.
- `wrong_inputs` – Požadavek přišel s prázdnými hodnotami.

5.6.9 Získání profilových dat

Při inicializaci obrazovky s profilovými daty, případně při vynuceném obnovení dat, je na server odeslán požadavek pro získání informací o navštívených památkách uživatele a pro získání žebříčku padesáti uživatelů co navštívilo nejvíce památek. Požadavek obsahuje tyto údaje:


```
{
  "email": "e-mail",
  "auth_token": "autentizační token"
}
```

Server zkontroluje shodu předaných přihlašovacích údajů uživatele. Pokud kontrola proběhne úspěšně, SQL dotazem se získá seznam 50 uživatelů seřazených sestupně dle počtu navštívených památek. Dále jsou SQL dotazem získány veškeré záznamy návštěv uživatele. Server může vrátit čtyři různé odpovědi:

- **success** – Získání dat z databáze bylo úspěšné. Zároveň se přeposílají následující data:
 - **top_fifty** – pole uživatelů spolu s jejich počtem navštívených památek,
 - **visits** – pole záznamů návštěv uživatele.
- **acc_not_found** – Účet nenalezen.
- **db_error** – Selhalo připojení k databázi.
- **wrong_inputs** – Požadavek přišel s prázdnými hodnotami.

6 Testování

Při vývoji aplikace bylo provedeno výkonnostní testování. Výsledná aplikace byla podrobena testování pomocí scénářů.

6.1 Výkonnostní testování

Při vývoji bylo provedeno výkonnostní testování. Především se testoval výkon při zobrazení většího počtu památek, nebo většího počtu záznamů navštívených památek.

U seznamu s památkami (viz obrázek 6.3a) docházelo k pomalejšímu renderování dlaždic památek. Tato chyba byla opravena zavedením stránkování a zobrazením pouze několika dlaždic památek na stránku.

Na obrazovce s mapou (viz obrázek 6.7a) docházelo po oddálení či přiblížení k zamrznutí aplikace na několik sekund. Mapa obsahovala informační box, jež byl součástí výchozí knihovny pro mapový rozhraní. Tento informační box se zobrazoval po zvolení značky památky. Informační boxy byly při oddalování a přibližování opětovně načítány, což bylo hlavním důvodem zmíněného zamrznutí. Z tohoto důvodu byla implementována vlastní komponenta nahrazující předchozí informační box. Přibližování a oddalování mapy se tak výrazně zrychlilo. Zamrznutí se vytratilo úplně.

Posledním pozorovaným objektem byl seznam posledně navštívených památek viz obrázek 6.8a. Pro verzi 1.0 s maximálním počtem památek 140 bylo procházení seznamu plynulé. Při rozšíření aplikace o výrazně větší počet zájmových bodů by bylo vhodné seznam optimalizovat.

6.2 Testovací scénáře

V této kapitole jsou popsány testovací scénáře výsledné mobilní aplikace. Scénáře byly testovány na mobilním zařízení *iPhone 6s* a *iPhone X* se systémem *iOS v13.5.1*.

6.2.1 Registrace

Popis scénáře

Před prvotním spuštěním aplikace zkontrolujeme připojení k internetu. Spustíme aplikaci a v případě upozornění, že aplikace využívá polohová zařízení

uživatele, potvrdíme vyskakovací dialogové okno. Po načtení aplikace (viz obrázek 6.1a) se zobrazí obrazovka s možností přihlášení se viz obrázek 6.1b. Na obrazovku s registrací nového účtu (viz obrázek 6.2a) se přesměrujeme zvolením modrého odkazu „Registrace“.

Registrační formulář vyplníme uživatelským jménem, e-mailovou adresou, heslem a potvrzením hesla. Pokud žádná kolonka nebude prázdná, e-mailová adresa bude ve správném formátu a hesla budou shodná, zvolením tlačítka „Registrovat“ se odešle žádost o registraci. V případě úspěšné registrace budeme přesměrováni na obrazovku s přihlášením a je nám zobrazeno dialogové okno s oznámením „Účet úspěšně vytvořen“.

Z obrazovky s přihlášením se odkážeme zpět na obrazovku s registrací a vyplníme všechny údaje se stejnou e-mailovou adresou zadanou v předchozím kroku. Po zvolení tlačítka „Registrovat“ se objeví chybová hláška „Uživatel s tímto e-mailem již existuje!“. Následně smažeme e-mailovou adresu a zkusíme zvolit tlačítka „Registrovat“. Objeví se chybová hláška „E-mail je ve špatném formátu!“. Zadáme novou e-mailovou adresu a zkusíme zapsat rozdílná hesla. Zvolením tlačítka „Registrovat“ se objeví chybová hláška „Hesla se neshodují!“. Dále zkusíme zapsat shodná hesla o délce například dvou znaků. Zvolením tlačítka „Registrovat“ se objeví chybová hláška „Hesla musí být v rozsahu 4 až 250 znaků!“. Přetažením z levé strany obrazovky na stranu pravou se dostaneme zpět na přihlašovací obrazovku.

Výsledek

Aplikace reagovala přesně podle popsaného scénáře. V případě odpojení od internetové sítě byla po zvolení tlačítka „Registrovat“ vypsána hláška „Zkontrolujte připojení k síti!“.

6.2.2 Obnovení hesla

Popis scénáře

Na obrazovce s přihlášením (viz obrázek 6.1b) zvolíme odkaz s názvem „Zapomněli jste heslo?“. Zobrazí se obrazovka pro obnovení hesla viz obrázek 6.2b.

Do políčka s e-mailovou adresou zkusíme zapsat e-mailovou adresu v neplatném formátu a zvolíme tlačítka „Zaslat kód na e-mail“. Objeví se chybová hláška „E-mail je ve špatném formátu!“. Dále zapíšeme e-mailovou adresu k neexistujícímu účtu a zvolíme tlačítka pro zaslání kódu na e-mailovou adresu. Zobrazí se chybová hláška „Účet neexistuje“.

Poté napíšeme e-mail existujícího účtu v aplikaci a zvolíme tlačítko „Zaslat kód na e-mail“. Zobrazí se dialogové okno s informací „Kód byl zaslán na váš e-mail“. Zvolením tlačítka „Ok“ dialogové okno zmizí. Nyní je třeba zkontrolovat příchozí e-maily na zvolené e-mailové adrese. Příchozí e-mail s předmětem *Mobilní cestovní deník - Obnovení hesla* bude obsahovat pětimístný kód pro obnovu hesla. Tento kód opíšeme do kolonky v aplikaci s názvem „Kód z e-mailu“. Následně zapíšeme shodná hesla a zvolíme tlačítko „Změnit kód“. Budeme přesměrováni na obrazovku s přihlášením a zobrazí se dialogové okno „Heslo úspěšně změněno.“.

Výsledek

Aplikace reagovala přesně podle scénáře. Po vyzkoušení opětovného odeslání kódu na zvolený e-mail se zobrazila chybová hláška „Kód byl již nedávno zaslán – opakujte akci později“. Kolonky pro hesla jsou stejně robustní jako u registrace viz podkapitola 6.2.1. Zapsáním jiného kódu pro obnovu hesla se objevila hláška „Neplatný kód!“.

6.2.3 Přihlášení přes účet v aplikaci

Popis scénáře

Jedním ze dvou způsobů přihlášení se je zadání e-mailové adresy a hesla k účtu, který byl vytvořen přes registrační formulář v aplikaci.

Do kolonek zapíšeme e-mailovou adresu a špatné heslo. Zvolením tlačítka „Přihlásit se“ se zobrazí chybová hláška „Zadali jste špatné heslo!“. Pokud zapíšeme e-mailovou adresu neexistujícího účtu, objeví se chybová hláška „Účet neexistuje!“. Dále zapíšeme e-mailovou adresu s heslem účtu, který jsme v předchozím kroku vytvořili. Zvolením tlačítka „Přihlásit se“ se provede přesměrování na obrazovku se seznamem památek viz obrázek 6.3a.

Výsledek

Aplikace reaguje přesně podle scénáře. Bez internetového připojení se objeví hláška „Zkontrolujte připojení k síti!“.

6.2.4 Přihlášení přes Facebook

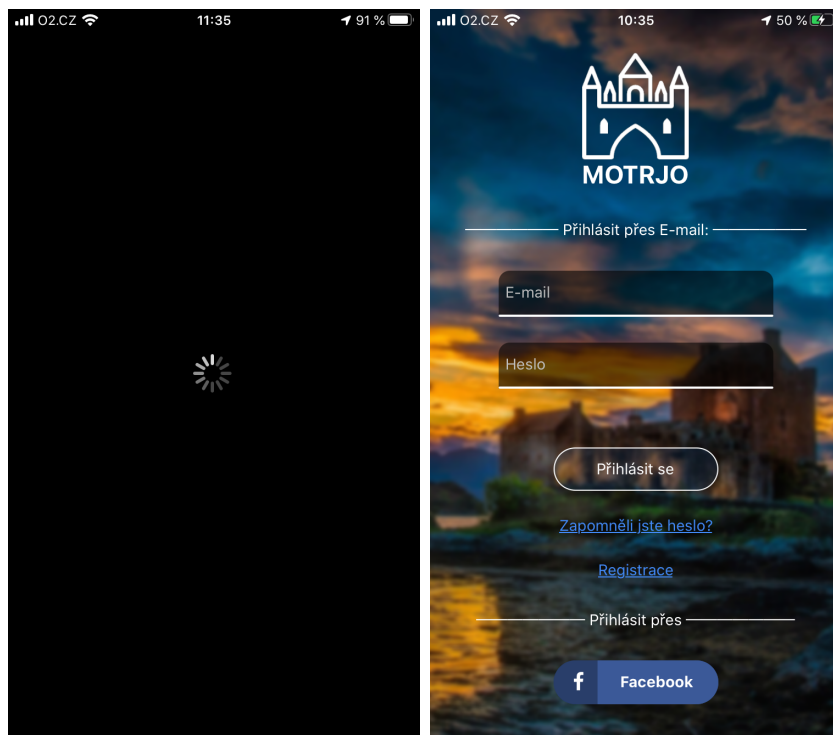
Popis scénáře

Druhým způsobem je přihlášení přes externí autentizaci u sociální sítě Facebook.

Na přihlašovací obrazovce viz obrázek 6.1b zvolíme tlačítko „Facebook“. Vyskočí dialogové okno, kde potvrdíme otevření jiné aplikace. Následně se zobrazí webová stránka Facebooku kde se přihlásíme zadáním přihlašovacích údajů k Facebookovému účtu, případně potvrdíme přímé přihlášení zvolením tlačítka „Pokračovat“. Po úspěšné autentizaci u Facebooku budeme přesměrování zpět do aplikace, kde se zobrazí obrazovka se seznamem památek viz obrázek 6.3a.

Výsledek

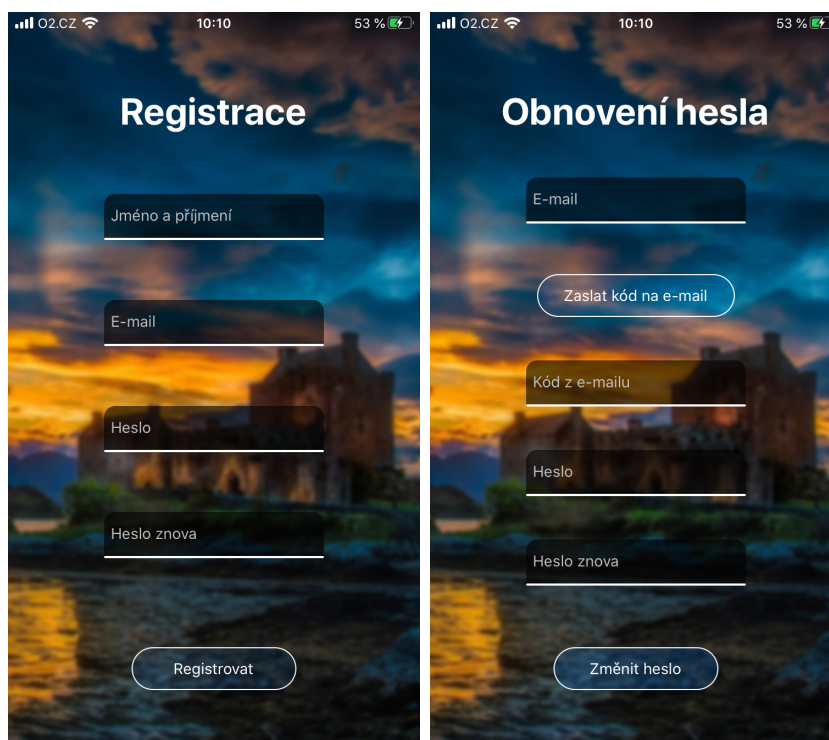
Aplikace reagovala přesně dle scénáře.



(a) Obrazovka s načítáním

(b) Obrazovka s přihlášením

Obrázek 6.1: Obrazovky aplikace testovacích scénářů



(a) Obrazovka registrace uživatele (b) Obrazovka s obnovením hesla

Obrázek 6.2: Obrazovky aplikace testovacích scénářů

6.2.5 Zobrazení seznamu památek

Popis scénáře

Obrazovka se seznamem památek (viz obrázek 6.3a) je úvodní obrazovkou aplikace po přihlášení.

Seznam můžeme posouvat a listovat jím. Tlačítka „Další“ a „Zpět“ můžeme listovat seznamem památek. Při přepínání mezi stránkami je vždy jedna památka v seznamu duplicitní – poslední památka v listu bude na prvním místě na další stránce.

Zvolením ikony filtrace (viz obrázek 6.3b) se zobrazí možnosti řazení seznamu. Zvolíme zobrazení památek, které jsme již navštívili. Kliknutím mimo modální okno filtrace, toto okno zavřeme. Pokud nemáme zaznamenané žádné návštěvy, zobrazí se chybová hláška „Nebyly nalezeny žádné odpovídající památky“. Znovu zvolením stejné možnosti filtrace se výběr zruší. Z možností filtrace v jednom řádku, lze vybrat pouze jednu z nich. Zvolením tlačítka „Vymazat filtraci“ se zruší výběr všech možností a modální okno

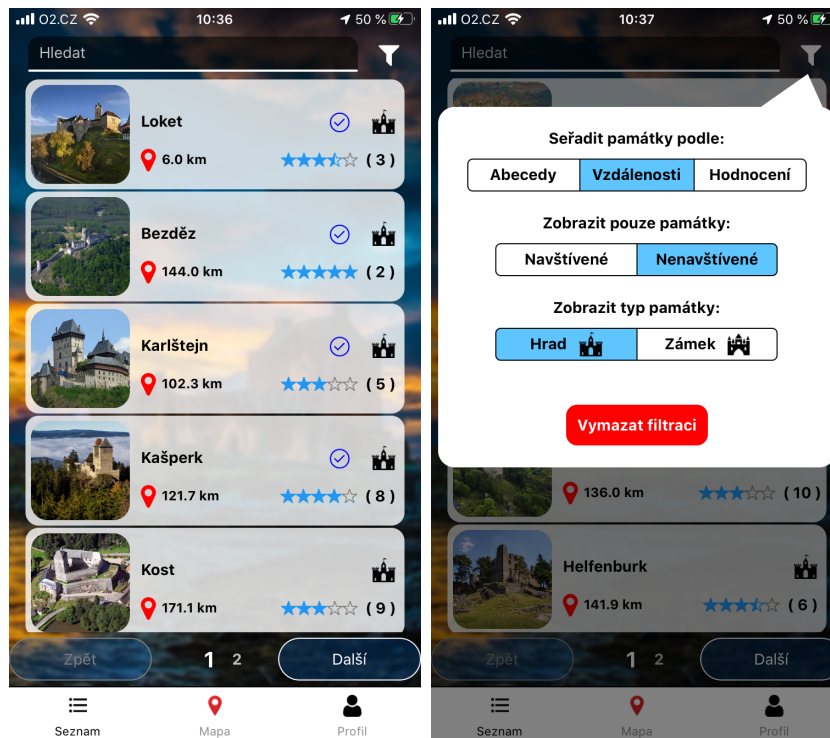
se zavře.

V seznamu můžeme i vyhledávat. Do vstupního políčka pro text s nadpisem „Hledat“ zadáme požadovaný řetězec. Dostaneme odpovídající výsledky viz obrázek 6.4a. V případě, že zadáme řetězec, který neodpovídá žádným výsledkům, zobrazí se hláška „Nebyly nalezeny žádné odpovídající památky“.

Po zvolení dlaždice konkrétní památky se ze spodní části obrazovky vysune informační okno s detaily památky viz podkapitola 6.2.6. Zvolením tlačítka „Zpět“ se toto informační okno zavře.

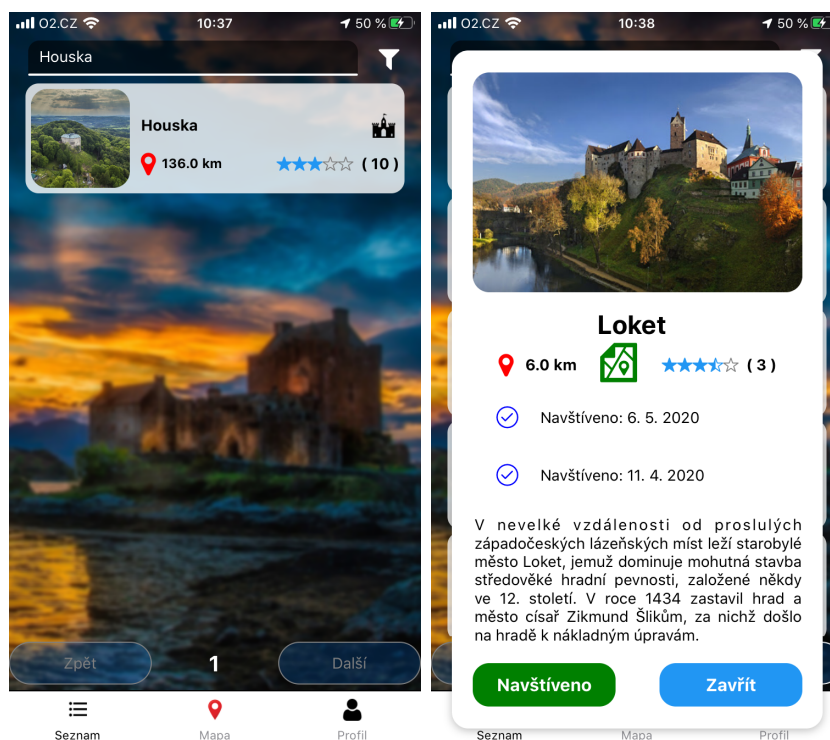
Výsledek

Aplikace reaguje přesně dle popsaného scénáře. Filtrace, řazení a stránkování seznamu probíhá korektně.



(a) Obrazovka se seznamem památek (b) Obrazovka s možností filtrace a řazení seznamu

Obrázek 6.3: Obrazovky aplikace testovacích scénářů



(a) Obrazovka se seznamem památek a aktivním vyhledáváním o památce (b) Obrazovka s informacemi o památce

Obrázek 6.4: Obrazovky aplikace testovacích scénářů

6.2.6 Zobrazení detailu památky

Popis scénáře

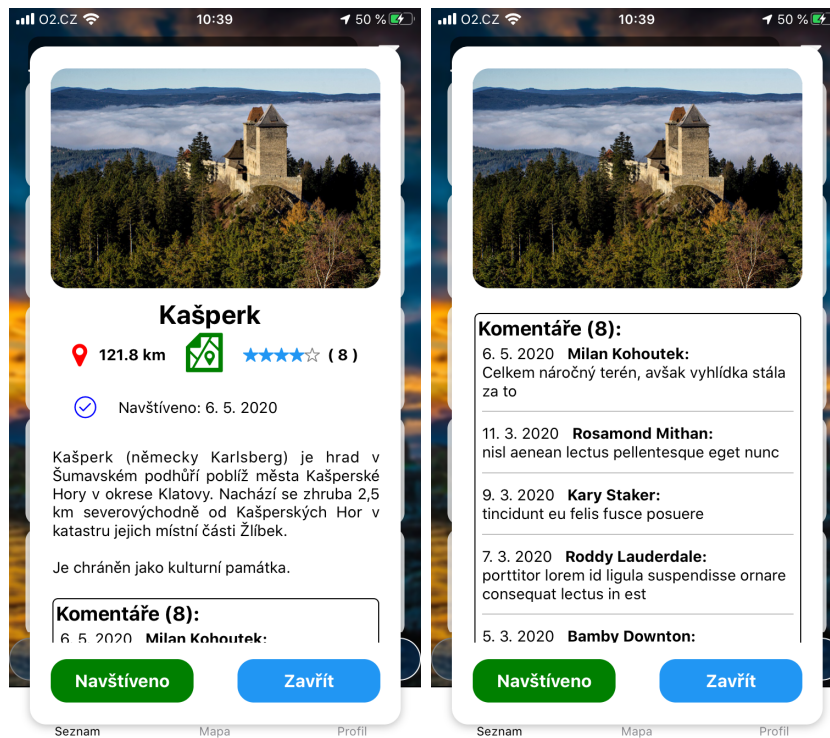
Detailní popis památek lze zobrazit vybráním dlaždice v seznamu památek či zvolením interaktivního informačního boxu u památky na obrazovce s mapou.

Informační modální okno obsahuje vzdálenost uživatele od památky, celkové hodnocení a popis památky viz obrázek 6.5a. Pokud památka obsahuje nějaký komentář, bude zobrazen pod popisem památky viz obrázek 6.5b.

Zvolením ikony s mapou se otevře navigace na mapách od společnosti Google. Vybráním počáteční pozice uživatele se zobrazí trasa od vybrané pozice k památce viz obrázek 6.5c.

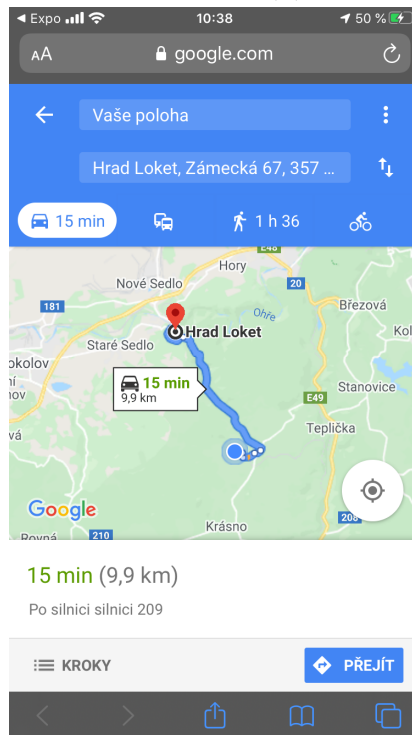
Výsledek

Aplikace reagovala dle popsaného scénáře. Vzdálenost od památky byla správná a pozice se aktualizuje každou minutu.



(a) Informace o památce

(b) Zobrazení komentářů



(c) Navigace k památce

Obrázek 6.5: Obrazovky aplikace testovacích scénářů

6.2.7 Ohodnocení památky

Popis scénáře

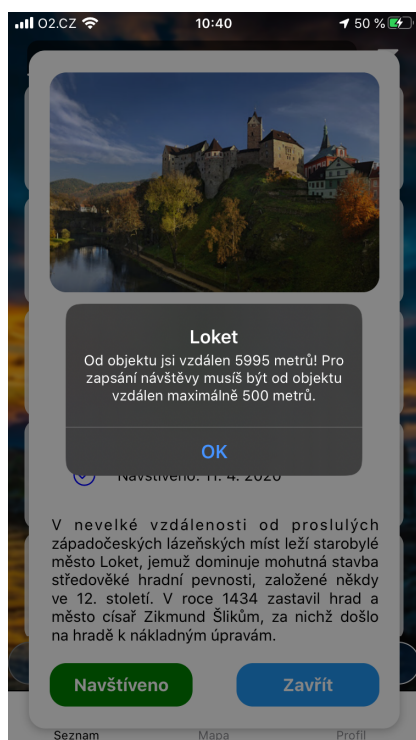
Pokud na modálním okně s informacemi o památce zvolíme tlačítko „Navštíveno“, objeví se modální okno s ohodnocením a okomentováním památky viz obrázek 6.6b. Případně se může objevit chybová hláška, která upozorní na přílišnou vzdálenost od památky viz obrázek 6.6a.

V tomto okně napíšeme komentář, neoznačíme žádnou hvězdičku a zvolíme tlačítko „Odeslat“. Zobrazí se dialogové okno s oznámením, že jsme nevybrali počet hvězdiček viz obrázek 6.6c.

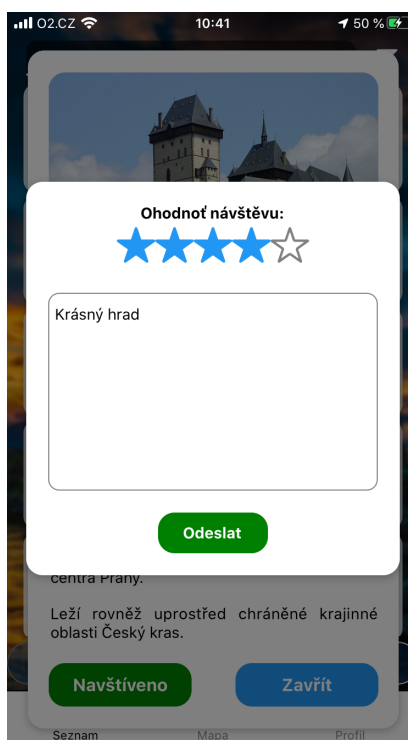
Vybráním počtu hvězdiček, zapsáním komentáře a zvolením tlačítka „Odeslat“ se provede uložení památky. Zobrazí se dialogové okno s oznámením úspěšného ohodnocení památky a uložení záznamu na server viz obrázek 6.6d. Aplikace nás přeměruje na výchozí obrazovku, kde jsme zvolili zobrazení detailnějších informací o památce.

Výsledek

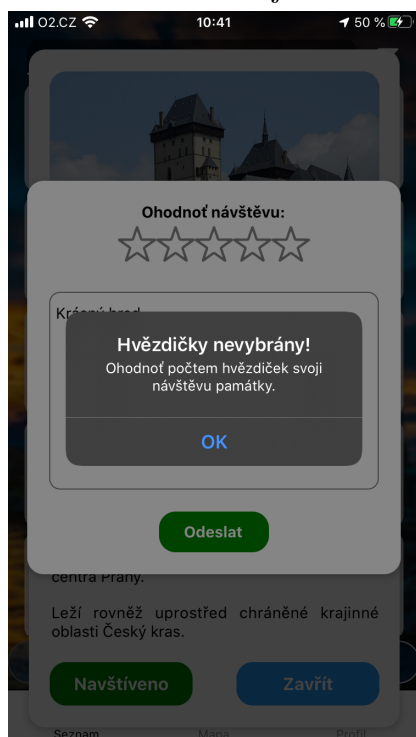
Popis scénáře je shodný se všemi událostmi, které nastaly. Dále bylo vyzkoušeno odeslání záznamu o navštívení památky bez internetového připojení. Zobrazilo se dialogové okno s oznámením chyby při zápisu dat. V tomto případě se památka uložila při znovunabytí internetového připojení. Opětovným ohodnocením stejné památky se minulé hodnocení pouze přepsalo.



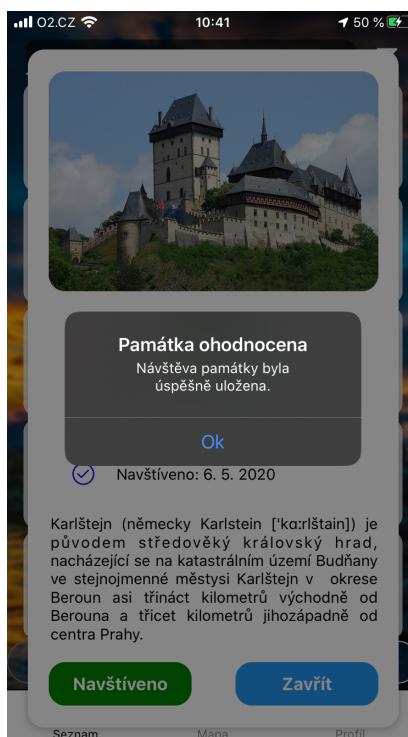
(a) Chybová hláška se vzdáleností uživatele od objektu



(b) Ohodnocení památky



(c) Chybová hláška – hvězdičky nevybrány



(d) Dialogové okno o úspěšném navštívení památky

Obrázek 6.6: Obrazovky aplikace testovacích scénářů

6.2.8 Zobrazení mapy

Popis scénáře

Zvolením ikony s nadpisem „Mapa“ v dolní části menu se přesměrujeme na obrazovku s profilovými daty viz obrázek 6.7a z obrazovky se seznamem památek, nebo z obrazovky s profilem.

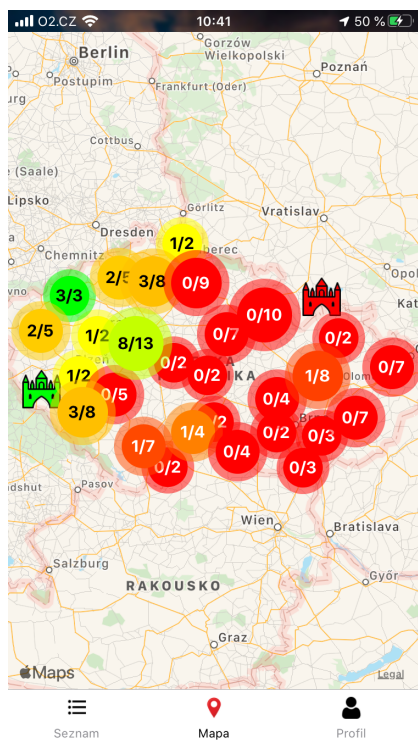
Přibližováním a oddalováním mapy se shlukují značky památek. Ve verzi aplikace 1.0 jsou pouze dva různé typy památky – hrad a zámek viz obrázek 6.7b. Značky různých typů památek jsou odlišné. Barva značek památek je zelená nebo červená. Zelená barva značí navštívenou památku. Červená barva značí nenavštívenou památku. Shluky značek jsou ve spektru barev červeno-žluto-zelené.

Zvolením značky památky vyjede ze spodní strany informační box a dojde k přiblížení na zvolenou památku na mapě viz obrázek 6.7c.

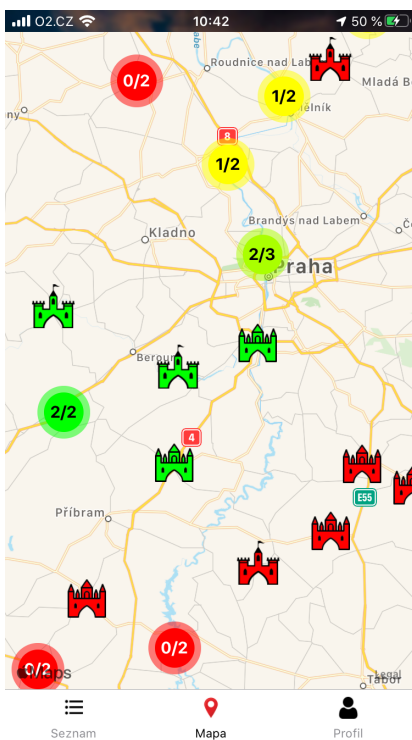
Po zvolení informačního boxu se objeví detailní popis památky (viz obrázek 6.7d), který byl zmíněn v podkapitole 6.2.6.

Výsledek

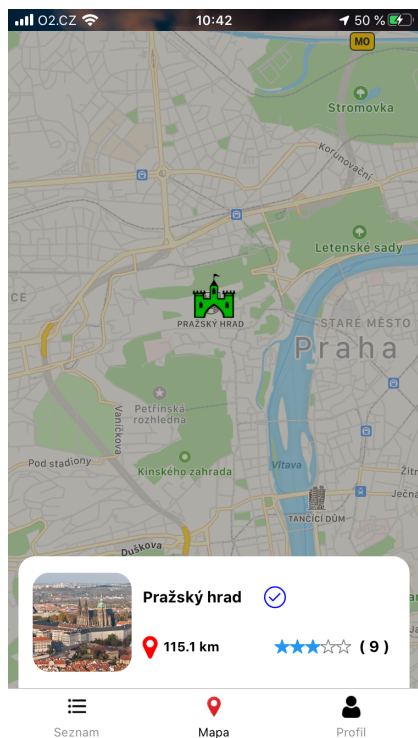
Aplikace reagovala přesně dle popsaného scénáře. Shlukování značek probíhalo plynule.



(a) Obrazovka s mapou



(b) Zobrazení značek památek a shluků



(c) Informační box památky



(d) Modální okno – Informace o památce

Obrázek 6.7: Obrazovky aplikace testovacích scénářů

6.2.9 Zobrazení profilu

Popis scénáře

Zvolením ikony postavičky s popisem „Profil“ v dolní části menu se přesměrujeme na obrazovku s profilovými údaji viz obrázek 6.8a.

V profilu uvidíme chronologicky seřazený seznam navštívených památek dle data poslední návštěvy. Mělo by zde být vidět naše hodnocení památky a také komentář.

Dále je zde počet již navštívených památek a kolik jich ještě zbývá k navštívení.

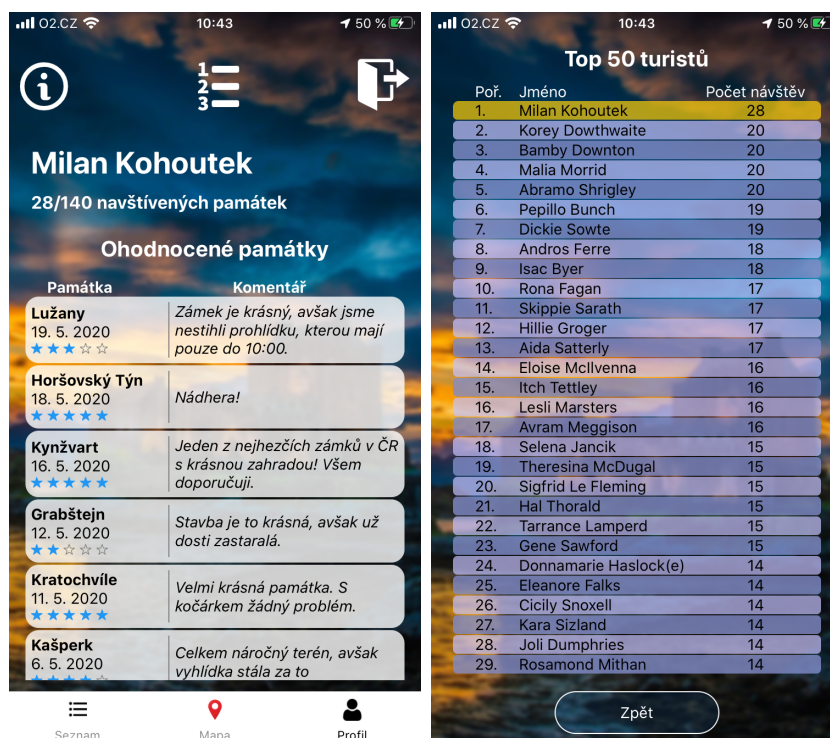
Zvolením levé ikony v horní části obrazovky se zobrazí modální okno s informacemi o aplikaci viz obrázek 6.8c. Zvolením tlačítka „Zpět“ se toto okno zavře.

Zvolením prostřední ikony v horní části obrazovky se zobrazí žebříček nejlepších padesáti uživatelů dle počtu uskutečněných návštěv. Naši pozici bychom vždy měli vidět v seznamu. Zvolením tlačítka „Zpět“ se toto okno zavře.

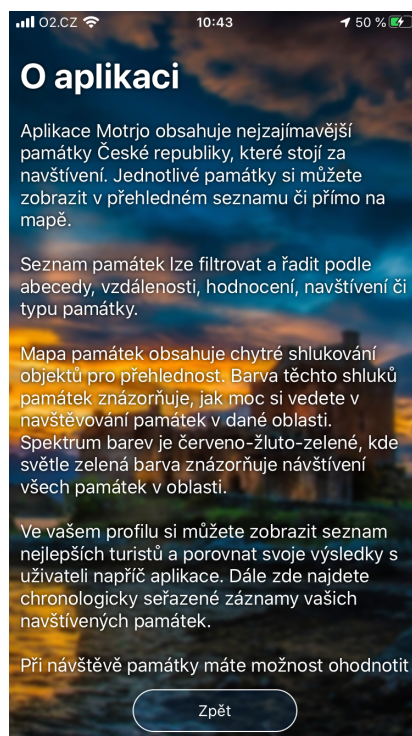
Poslední ikona v horní části obrazovky reprezentuje odhlášení se z účtu. Aplikace nás přesměruje na obrazovku s přihlášením.

Výsledek

Aplikace reagovala stejně jako ve scénáři. Navštívené památky, jejich ohodnocení a komentáře byly zobrazeny korektně. Také souhlasil počet již navštívených památek s počtem návštěv v žebříčku nejlepších turistů aplikace.



(a) Obrazovka s profilem uživatele (b) Modální okno – žebříček turistů



(c) Modální okno – o aplikaci

Obrázek 6.8: Obrazovky aplikace testovacích scénářů

7 Možná rozšíření systému

7.1 Rozšíření mimo Českou republiku

Aplikace je ve verzi 1.0 pouze v češtině a památky jsou lokalizovány pouze v České republice. Dalším rozšířením aplikace tak může být rozšíření do jiných států. Toto zahrnuje také rozšíření jazykové podpory.

7.2 Rozšíření typů památek

Celkový systém umožňuje jednoduchou implementaci dalších typů objektů zájmu. Verze 1.0 obsahuje 140 vybraných hradů a zámků, avšak není to úplný seznam hradů a zámků na území České republiky. Mezi další body zájmu mohou být zahrnuty například i rozhledny, kostely, zříceniny hradů nebo zámků.

7.3 Autentizace přes další sociální sítě

V současné verzi 1.0 je implementován pouze jeden externí autentizační postup, a to přes sociální síť Facebook. Dalšími možnostmi externí autentizace mohou být například přes Twitter nebo Gmail.

7.4 Webová aplikace

Celková funkcionální mobilní aplikace může být implementována i pro webové rozhraní. Framework React Native podporuje také vývoj webových aplikací, tudíž většina použitých komponent by měla fungovat i pro webovou aplikaci.

Také je zde možnost rozšíření administračního panelu viz podkapitola 5.4.2. Vhodné je například rozšíření statistických údajů systému a implementace správy uživatelů.

7.5 Převod aplikace na platformu Android

Framework React Native podporuje multiplatformní vývoj, díky čemuž by distribuce pro platformu Android nebyla komplikovaná. Aplikace je však primárně vyvíjena pro platformu iOS, což může mít za následek problémy s kompatibilitou některých komponent.

7.6 Publikace aplikace

Současnou verzi aplikace 1.0 je možné publikovat v App Store. Příkazem `expo build:ios` lze vygenerovat soubor s koncovkou IPA, který je následně nahrán do App Store ke kontrole. Podmínkou publikace je nutné vyplnění potřebných údajů v App Store jako je například název, popis aplikace, verze, kategorie, klíčová slova a profil vývojáře. Dále jsou vyžadovány údaje k testovacímu účtu v aplikaci pro následnou kontrolu. Vedení vývojářského účtu na App Store podléhá zpeněžení.

7.7 Stahování dat památek ze serveru

V případě úspěšné publikace mobilní aplikace a častého přidávání nových hradů a zámků bude jednodušší ukládat data památek na webový server. Přidání nové památky tak nebude vyžadovat vydání nové verze aplikace.

8 Závěr

V této práci byla prozkoumána nabídka aplikací s tematikou turistiky zájmových bodů. Byla navržena a vytvořena mobilní aplikace pro platformu iOS. Tato aplikace se ujímá role mobilního cestovního deníku a je vhodným motivátorem pro navštěvování turisticky atraktivních míst.

V analytické části práce byly rozebrány možné přístupy a možnosti volby technologií pro vývoj mobilních aplikací pro operační systém iOS. S využitím poznatků z teoretického rozboru byl výsledný systém následně navržen. Bylo navrženo kompletní grafické uživatelské rozhraní aplikace, stejně tak jako veškeré funkcionality systému. Implementace proběhla shodně s návrhem přičemž odchylek bylo velmi málo. Změny v původním návrhu byly provedeny za účelem optimalizace výkonu aplikace. Důležitým prvkem systému bylo vyřešení komunikace mobilního klienta s webovým serverem. Byl zde kladen důraz na zabezpečení systému, vhodnou autentizaci uživatele a sdílení dat mezi uživateli. Výrazným rozšířením oproti návrhu bylo vytvoření webové aplikace, která slouží pro zobrazení statistických údajů a administrátorskou správu.

Hybridní přístup k vývoji mobilní aplikace za pomoci frameworku React Native byl velmi intuitivní a nemohu jej než doporučit.

Výsledná aplikace „Motrjo“ nabízí přehled turisticky atraktivních míst, které přehledně vizualizuje v mapovém rozhraní. Uživatel má možnost své návštěvy těchto míst zaznamenávat v aplikaci a udržovat si tak formu cestovního deníku. Aplikace má v sobě jisté prvky sociální sítě, umožňuje uživatelům například přidávat hodnocení a komentáře.

Seznam zkratek

API – Application Programming Interface

CSS – Cascading Style Sheets

DOM – Document Object Model

HTML – Hypertext Markup Language

HTTP(S) – Hypertext Transfer Protocol (Secure)

IDE – Integrated Development Environment

IPA – iOS App Store Package

JSON – JavaScript Object Notation

OS – Operating System

PHP – Hypertext Preprocessor

PWA – Progressive Web Apps

SQL – Structured Query Language

SSL – Secure Sockets Layer

TLS – Transport Layer Security

UWP – Universal Windows Platform

WWDC – Worldwide Developers Conference

Literatura

- [1] *Oficiální web Turistické známky* [online]. [turisticke-znamky.cz](http://www.turisticke-znamky.cz) [cit. 2019/11/19] Dostupné z: <http://www.turisticke-znamky.cz/>
- [2] DEAN ALAN HUME *Progressive Web Apps*. Prosinec 2017. ISBN 9781617294587
- [3] JOSEF GATTERMAYER *Proč a jak psát progresivní webové aplikace* [online]. ackee.cz, 11. ledna 2017 [cit. 2019/11/19]. Dostupné z: <https://www.ackee.cz/blog/proc-a-jak-psat-progresivni-webove-aplikace/>
- [4] DAVID ORAGUI *Příklady progresivních webových aplikací* [online]. themanifest.com, 10. dubna 2018 [cit. 2019/11/20]. Dostupné z: <https://themanifest.com/app-development/11-examples-progressive-web-apps>
- [5] MAHESH PANHALE *Beginning Hybrid Mobile Application Development*. Apress Leden 2016. ISBN 9781484213148
- [6] *Oficiální stránky Swift* [online]. [cit. 2020/01/10]. Dostupné z: <https://swift.org/>
- [7] *Hybridní vývoj s Xamarinem* [online]. [cit. 2020/06/21]. Dostupné z: <https://docs.microsoft.com/en-us/xamarin/cross-platform/>
- [8] *Přenosné knihovny tříd* [online]. [cit. 2020/06/24]. Dostupné z: <https://docs.microsoft.com/cs-cz/xamarin/cross-platform/app-fundamentals/pcl?tabs=windows>
- [9] *Xamarin Customer Showcase* [online]. [cit. 2019/11/26]. Dostupné z: <https://dotnet.microsoft.com/apps/xamarin/customers>
- [10] ADAM BODUCH *React and React Native*. Packt Publishing Březen 2017. ISBN 9781786465658
- [11] SHAUN EDUO *What is React Native? How does it work?* [online]. [cit. 2020/01/24]. Dostupné z: <https://medium.com/@eduoshaun/what-is-react-native-how-does-itwork-e745ca1fae7b>

- [12] MARTÍN BIGIO *Introducing Hot Reloading* [online]. [cit. 2020/01/24]. Dostupné z: <https://facebook.github.io/react-native/blog/2016/03/24/introducing-hot-reloading.html>
- [13] *React Navigation* [online]. reactnavigation.org [cit. 2020/06/16]. Dostupné z: <https://reactnavigation.org/>
- [14] *Pokyny pro uživatelské rozhraní* [online]. [cit. 2019/12/10]. Pro iOS dostupné z: <https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/>. Pro Android dostupné z: <https://developer.android.com/design>
- [15] ROSS BULAT *React Native: User Authentication Flow Explained* [online]. medium.com, prosinec 2019 [cit. 2020/04/04]. Dostupné z: <https://medium.com/@rossbulat/react-native-user-authentication-flow-explained-d988905ba106>
- [16] SAJAL DULAL *React Native Async Storage (persist) Patterns for beginners* [online]. medium.com, březen 2019 [cit. 2020/06/14]. Dostupné z: <https://medium.com/wesionario-team/react-native-persist-values-pattern-for-beginners-c74aa1a037bf>
- [17] *Facebook Graph API* [online]. developers.facebook.com [cit. 2020/06/10]. Dostupné z: <https://developers.facebook.com/docs/graph-api/>
- [18] *Descriptions of SHA-256, SHA-384, and SHA-512* [online]. [cit. 2020/04/03]. Dostupné z: <http://www.iwar.org.uk/comsec/resources/cipher/sha256-384-512.pdf>
- [19] DAN ARIAS *Hashing* [online]. [cit. 2020/04/03]. Dostupné z: <https://auth0.com/blog/hashing-passwords-one-way-road-to-security/>
- [20] *PHP manuál funkce password_hash* [online]. [cit. 2020/06/15]. Dostupné z: <https://www.php.net/manual/en/function.password-hash.php>
- [21] *PHP manuál funkce password_verify* [online]. [cit. 2020/06/15]. Dostupné z: <https://www.php.net/manual/en/function.password-verify.php>
- [22] JAN BARÁŠEK *Hashování řetězců a hesel* [online]. php.baraja.cz, 2020 [cit. 2020/04/09] Dostupné z: <https://php.baraja.cz/hashovani>

- [23] *PHP manuál funkce bind_param* [online]. [cit. 2020/06/15]. Dostupné z: <https://www.php.net/manual/en/mysqli-stmt.bind-param.php>
- [24] ARITRA MUKHERJEE *Prevent SQL Injection in PHP, Web security* [online]. geekyminds.co.in, 2019 [cit. 2020/04/08] Dostupné z: <https://geekyminds.co.in/prevent-sql-injection-with-parameter-binding/>
- [25] *What is SSL, TLS and HTTPS?* digicert.com [online]. [cit. 2020/07/13] Dostupné z: <https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https>

Přílohy

A Instalační příručka projektu

A.1 Mobilní aplikace

Předpoklady

Pro zprovoznění projektu je nutné stažení *Node.JS* s minimální verzí *10 (LTS)* obsahující package systém *npm*. Dále je potřeba mít nainstalovanou mobilní aplikaci *Expo Client*.

Instalace

Obsah projektu z CD zkopírujeme do úložiště v počítači. V příkazové řádce se přesměrujeme do adresáře zdrojových souborů mobilní aplikace *App/motrjo*. Pokud v adresáři *App/motrjo* není podadresář *node_modules*, provedeme v příkazové řádce následující příkaz:

```
user:~/App/motrjo$ npm install
```

Tímto příkazem se nainstalují potřebné knihovny do nově vytvořeného adresáře *node_modules*. Po stažení knihoven je možné projekt Motrjo spustit příkazem:

```
user:~/App/motrjo$ npm start
```

Po tomto příkazu je do příkazové řádky zaznamenán průběh přepisování komponent aplikace viz obrázek 8.1.

```
kohin@kohin:~/bp_mobile_travel_journal/App/motrjo$ npm start
> @ start /home/kohin/bp_mobile_travel_journal/App/motrjo
> ./scripts/start.sh && expo start
--- MOTRJO ---
--- Re-writing library in node_modules - react-native-map-clustering
--- 1/4 completed
--- 2/4 completed
--- 3/4 completed
--- Re-writing library in node_modules - rn-sliding-up-panel
--- 4/4 completed
--- Customized libraries were successfully copied.. Starting Project..
```

Obrázek 8.1: Kopírování komponent a spuštění Expo

Při úspěšném spuštění projektu je zobrazen QR kód s odkazem do mobilní aplikace *Expo Client* viz obrázek 8.2.



Obrázek 8.2: Indikace úspěšně spuštěného projektu Expo

Pro spuštění aplikace na mobilním zařízení je třeba načíst vygenerovaný QR kód. Na mobilním zařízení je následně zobrazeno modální okno s přesměrováním do aplikace *Expo Client*. Po odsouhlasení přesměrování je zobrazena načítací obrazovka *Expo Client* a aplikace Motrjo je sestavována. Po úspěšném sestavení je zobrazena úvodní obrazovka aplikace Motrjo.

Dodatečná konfigurace

Úspěšná konfigurace komunikace mezi mobilní aplikací a webovým serverem vyžaduje změnu adresy pro zasílání HTTPS požadavků. Tato změna je realizována upravením hodnoty *WEB_URL* v souboru *Constants.js* v podadresáři *App/motrjo/constants*.

Pro zprovoznění přihlašování se přes sociální síť Facebook je zapotřebí aplikaci registrovat na stránkách <https://developers.facebook.com/> a zde povolit funkci *Facebook Login*. Facebook vygeneruje unikátní identifikátor aplikace, který je nutné vložit do proměnné *APP_ID* v souboru *Constants.js* v podadresáři *App/motrjo/constants*.

A.2 Webový server

Předpoklady

Pro zprovoznění serverové části bude zapotřebí Apache HTTP server s minimální verzí 2.4. Dále bude zapotřebí podpora skriptovacího jazyka *PHP* s minimální verzí 7.1, relační databáze *MariaDB* s minimální verzí 10.1.37 a program pro správu databází například *SQLyog*.

Instalace

Zdrojové soubory z adresáře *Web_produkce* je třeba zkopírovat do kořenového adresáře webového serveru. Tento adresář je typicky nazván *www* nebo *htdocs*.

Dále je nutné vytvořit dvě databáze. První databáze bude sloužit pro uchování dat mobilní aplikace. Do této databáze nahrajeme z adresáře *Web_DB* SQL soubor *create_mobile_app_database.sql*. Druhá databáze bude sloužit pro přihlašování do webové aplikace. Do této databáze nahrajeme z adresáře *Web_DB* SQL soubor *create_web_app_database.sql*

Dodatečná konfigurace

Pro úspěšnou konfiguraci webového serveru je zapotřebí změnit přihlašovací údaje k databázím. Přihlašovací údaje k databázi mobilní aplikace je nutné změnit ve zdrojovém souboru *db_connection.php*. Tento soubor je v adresáři *Web_produkce/app_requests*. Přihlašovací údaje k databázi webové aplikace je možné změnit ve zdrojových souborech *login.php* a *session.php*. Tyto soubory jsou v adresáři *Web_produkce*.

V adresáři *Web_produkce/images* je nutné extrahovat obrázky ze souborů *hrady.zip* a *zamky.zip*.

Přihlašovací údaje k účtu do webové aplikace jsou uloženy v souboru *README.md* v kořenovém adresáři CD.

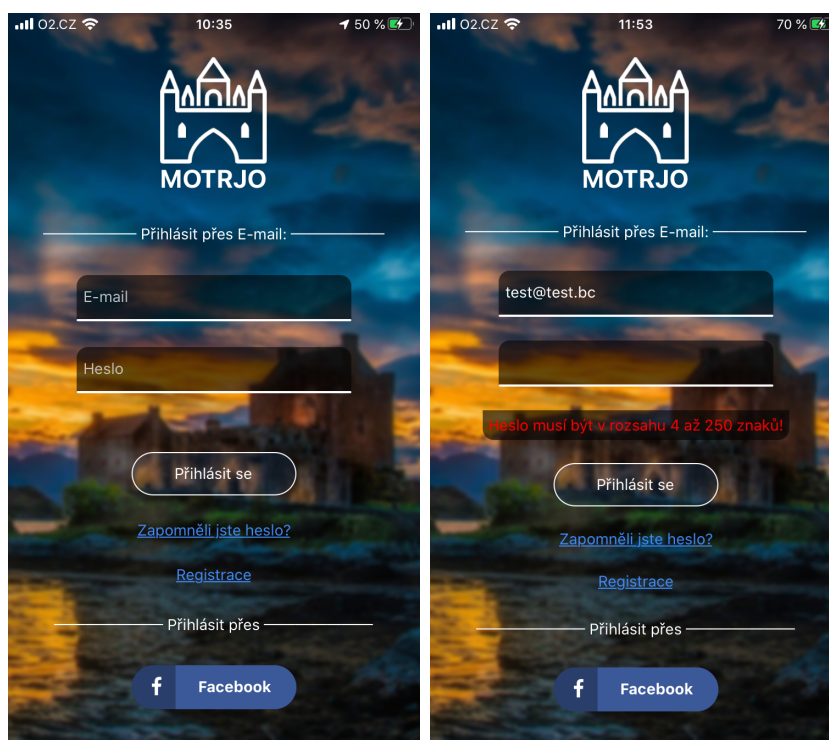
B Uživatelská příručka

B.1 Přihlášení přes účet v aplikaci

Po spuštění aplikace je zobrazena obrazovka s možností přihlášení se k uživatelskému účtu viz obrázek 8.3a.

Pokud účet nemáme vytvořený, je nutné ho založit. Zvolením modrého odkazu *Registrace* je zobrazena obrazovka s registrací viz podkapitola B.3.

K vytvořenému účtu se lze přihlásit zadáním e-mailové adresy, hesla a zvolením tlačítka *Přihlásit se*. V případě neúspěšného přihlášení je zobrazena červená chybová hláška s popisem problému viz obrázek 8.3b. Po úspěšném přihlášení je zobrazena obrazovka se seznamem památek viz podkapitola B.5.



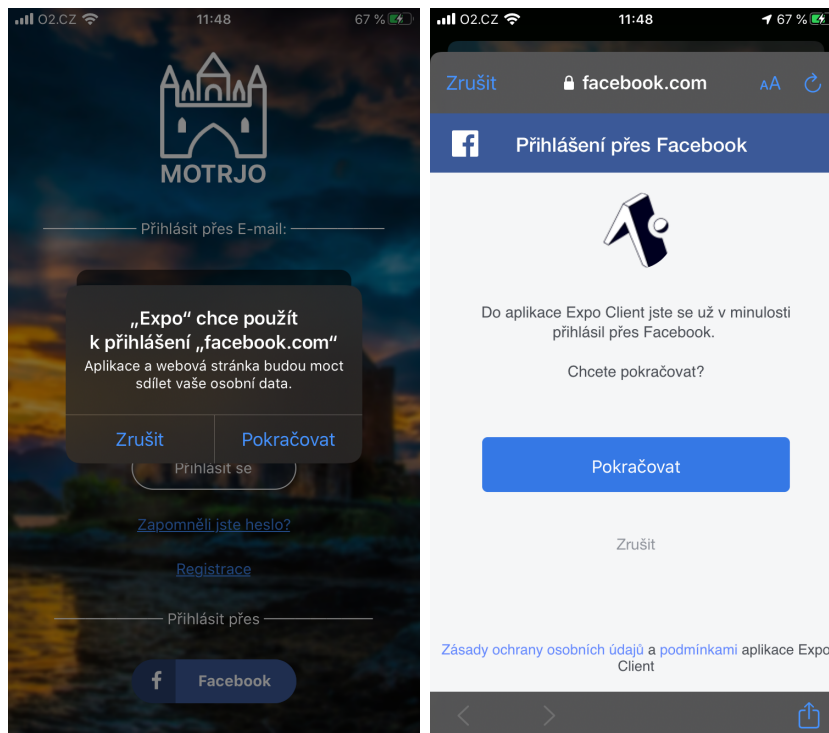
(a) Obrazovka s přihlášením (b) Špatně zadané přihlašovací údaje

Obrázek 8.3: Obrazovky s přihlášením

B.2 Přihlášení přes účet na Facebooku

Druhým způsobem přihlášení se je přes účet na sociální síti Facebook. Zvolením modrého tlačítka *Facebook* na obrazovce s přihlášením (viz obrázek

8.3a) je zobrazena žádost o přesměrování do jiné aplikace viz obrázek 8.4a. Zde je ušetřeno zadávání přihlašovacích údajů, za předpokladu přihlášeného Facebookového účtu ve fyzickém zařízení. Součástí prvního přihlašování se přes Facebookový účet je udělení oprávnění aplikaci Motrjo. Při následném přihlašování uživatel již potvrzuje přihlášení viz obrázek 8.4b. Po úspěšném přihlášení je zobrazena obrazovka se seznamem památek viz podkapitola B.5.

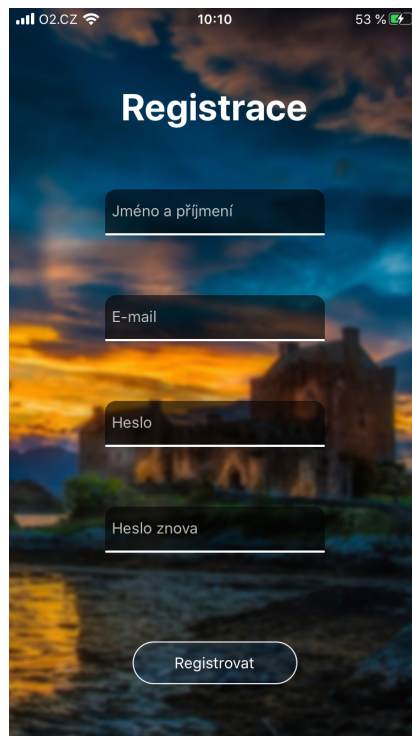


(a) Potvrzení přesměrování na jinou aplikaci (b) Potvrzení přihlášení

Obrázek 8.4: Přihlášení přes účet na Facebooku

B.3 Registrace

Na obrazovku s registrací se dostaneme zvolením tlačítka *Registrace* viz obrázek 8.3a. Pro úspěšné vytvoření nového účtu je potřeba zadat své jméno či přezdívkou, e-mailovou adresu, dvakrát stejné heslo a zvolit tlačítko *Registrovat* viz obrázek 8.5. V případě neúspěšného založení účtu se zobrazí červená hláška s popisem chyby. Po úspěšné registraci je zobrazena obrazovka s přihlášením (viz podkapitola B.1) a dialogové okno s oznámením o úspěšně založeném účtu.



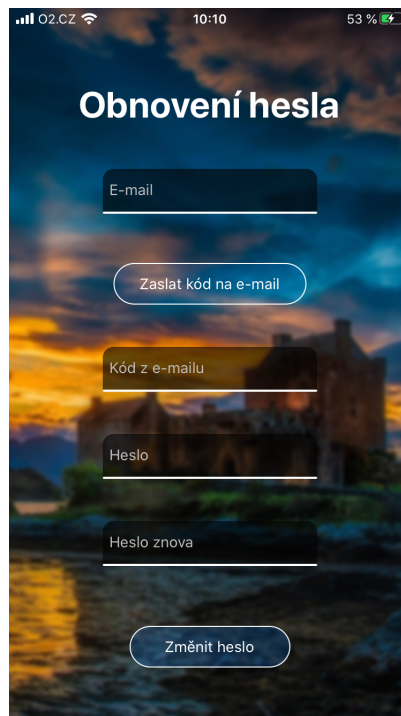
Obrázek 8.5: Obrazovka registrace

B.4 Obnovení hesla

V případě zapomenutého hesla je možnost heslo ke svému účtu změnit. Po zvolení modrého odkazu *Zapomněli jste heslo?* na obrazovce s přihlášením (viz obrázek 8.3a) je zobrazena obrazovka s možností obnovy hesla viz obrázek 8.6.

Prvním krokem je napsání e-mailové adresy do kolonky pro e-mail a zvolit tlačítko *Zaslat kód na e-mail*. Na zadanou adresu bude odeslán kód, který bude nutný pro další krok. V případě neúspěšného odeslání kódu na e-mail se zobrazí červená chybová hláška s popisem chyby. Druhým krokem je zkopírování kódu z e-mailové schránky a jeho zadání do kolonky „Kód z e-mailu“. Dále je nutné zapsat dvakrát stejné heslo. Zkontrolujeme všechny vyplněné kolonky na obrazovce a zvolíme tlačítko *Změnit heslo*.

Při úspěšné změně hesla dojde k přesměrování na obrazovku s přihlášením (viz podkapitola B.1) a zobrazí se dialogové okno s oznámením o úspěšné změně hesla.



Obrázek 8.6: Obrazovka s obnovením hesla

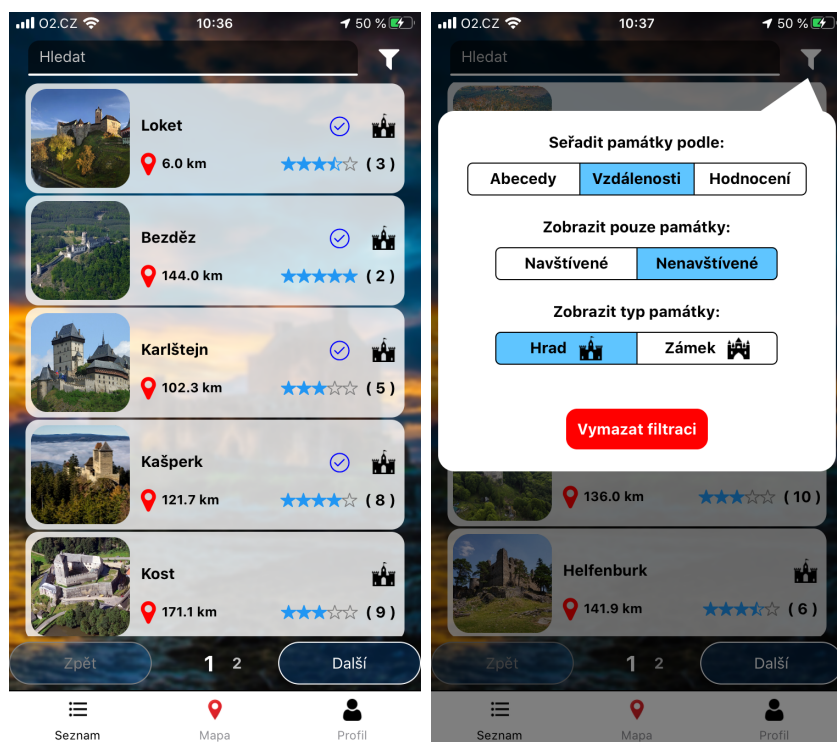
B.5 Seznam památek

Po úspěšném přihlášení je zobrazena obrazovka se seznamem památek viz obrázek 8.7a. Seznam památek se skládá z dlaždic se základními informacemi o památce. Každá dlaždice obsahuje název, typ a hodnocení památky. Dále je zobrazena vzdálenost památky od naší polohy a ukazatel navštívení v podobě ikony fajfky. Seznam je posuvný nahoru a dolů. Tlačítka *Zpět* a *Další* lze seznamem listovat. Dlaždice jsou interaktivní. Po zvolení jednotlivé dlaždice je zobrazeno modální okno s informacemi o památce viz podkapitola B.7.

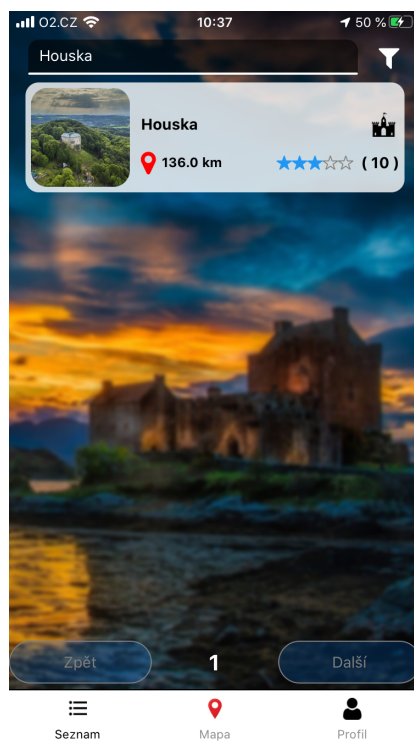
Seznam lze různě řadit a filtrovat. Zvolením ikony filtrace je zobrazeno modální okno viz obrázek 8.7b. Zvolením možností lze seznam seřadit například podle abecedy, vzdálenosti či hodnocení. Dále je možné si zobrazit pouze navštívené či nenavštívené památky nebo typ objektu hrad či zámek. Veškeré zvolené možnosti řazení a filtrace seznamu je možné vypnout zvolením tlačítka *Vymazat filtraci*.

V seznamu je možné vyhledávat zadáním názvu objektu do kolonky „Hledat“ viz obrázek 8.7c.

Z obrazovky se seznamem památek je možnost přesměřovat se na obrazovku s mapou (viz podkapitola B.6) nebo na obrazovku s profilem viz podkapitola B.8.



(a) Úvodní obrazovka se seznamem památek (b) Filtrace a řazení památek



(c) Vyhledávání v seznamu památek

Obrázek 8.7: Obrazovky se seznamem památek

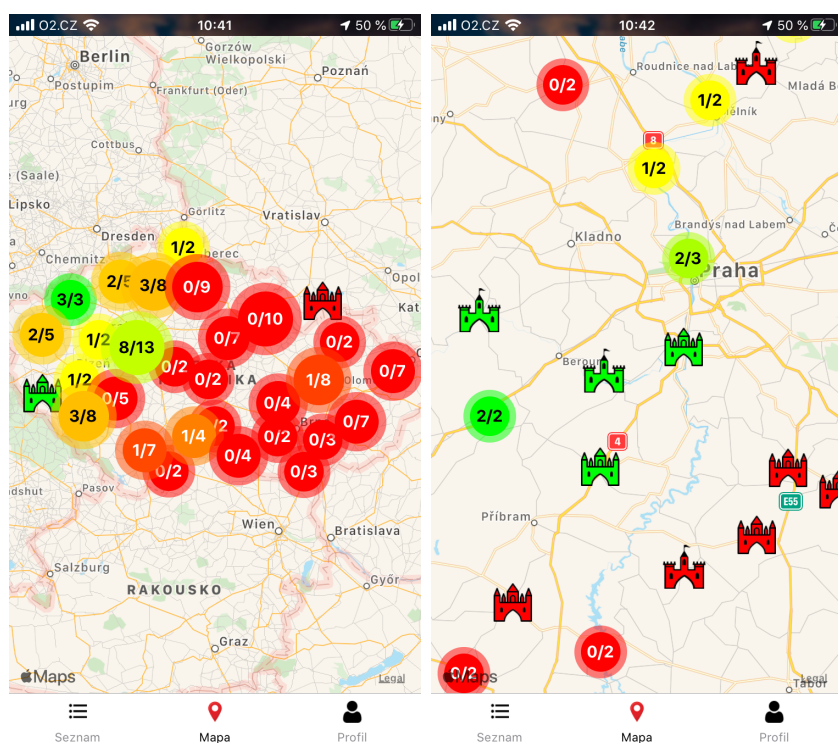
B.6 Mapa s památkami

Na obrazovku s mapou (viz obrázek 8.8a) lze přesměrovat z obrazovky se seznamem památek, nebo z obrazovky s profilem zvolením prostřední ikony ve spodní části obrazovky.

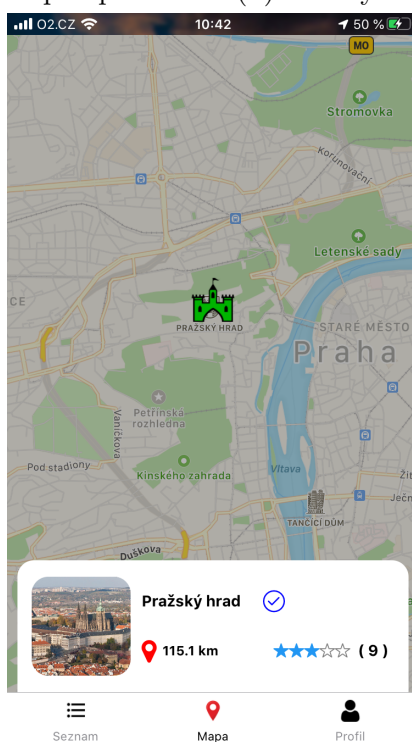
Mapa lze různě přibližovat, oddalovat a otáčet. Ve verzi 1.0 jsou celkem dva druhy značek, které znázorňují typ objektu. Značku hradu poznáme podle ochozů zatímco značka zámku má věžičky viz obrázek 8.8b. Barva značky může být buď červená nebo zelená. Pokud je značka zelená, památka byla navštívena. Pokud je značka červená, památku jsme ještě nenavštívili. Kulaté shluky značek jsou ve spektru barev červeno-žluto-zelené, kde světle zelená barva shluku znamená navštívení všech památek v dané oblasti.

Po zvolení značky památky je zobrazen interaktivní informační box viz obrázek 8.8c. Informační box obsahuje název a hodnocení památky. Dále obsahuje vzdálenost od naší polohy a ukazatel navštívení v podobě ikony fajfky. Po zvolení informačního boxu je zobrazeno modální okno s informacemi o památce viz podkapitola B.7.

Z obrazovky s mapou je možnost přesměrování se na obrazovku se seznamem památek (viz podkapitola B.5) nebo na obrazovku s profilem viz podkapitola B.8.



(a) Obrazovka s mapou památek (b) Značky a shluky památek



(c) Informační box památky

Obrázek 8.8: Obrazovky s mapou památek

B.7 Informace o památce

Zvolením dlaždice v seznamu památek (viz podkapitola B.5) nebo zvolením informačního boxu na obrazovce s mapou (viz podkapitola B.6) bude zobrazeno modální okno s informacemi o památce viz obrázek 8.9a.

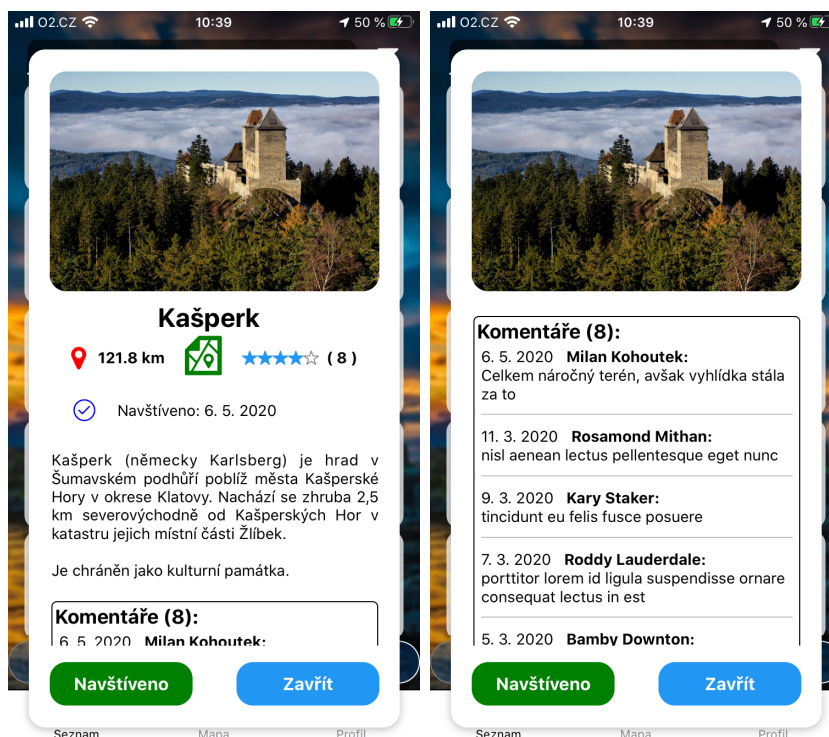
Modální okno obsahuje název, hodnocení a základní popis památky. Dále je zde vzdálenost památky od naší polohy, v případě navštívení ukazatel s časem návštěvy a přesměrování na stránky Google maps s možností navigace k památce viz obrázek 8.9c. Pokud památka obsahuje nějaké komentáře od uživatelů, budou zobrazeny pod základním popisem památky viz obrázek 8.9b. Zvolením náhledového obrázku památky je obrázek zvětšen přes celou obrazovku. Pokud má památka více obrázků, přetahováním do stran jimi lze listovat.

Po zvolení tlačítka *Navštíveno* je zobrazeno modální okno s hodnocením památky viz obrázek 8.9d. Nutnou podmínkou odeslání záznamu o navštívené památce je zvolení počtu hvězdiček. Navíc je možnost k hodnocení připsat komentář. Zvolením tlačítka *Odeslat* a v případě úspěšného uložení záznamu je zavřeno modální okno s ohodnocení památky i modální okno s informacemi o památce a zobrazí se dialogové okno s oznámením úspěšného uložení návštěvy.

Pokud chceme opravit naše hodnocení památky, musíme tak učinit znovu zaznamenáním návštěvy památky do 24 hodin od odeslání záznamu opravovaného hodnocení.

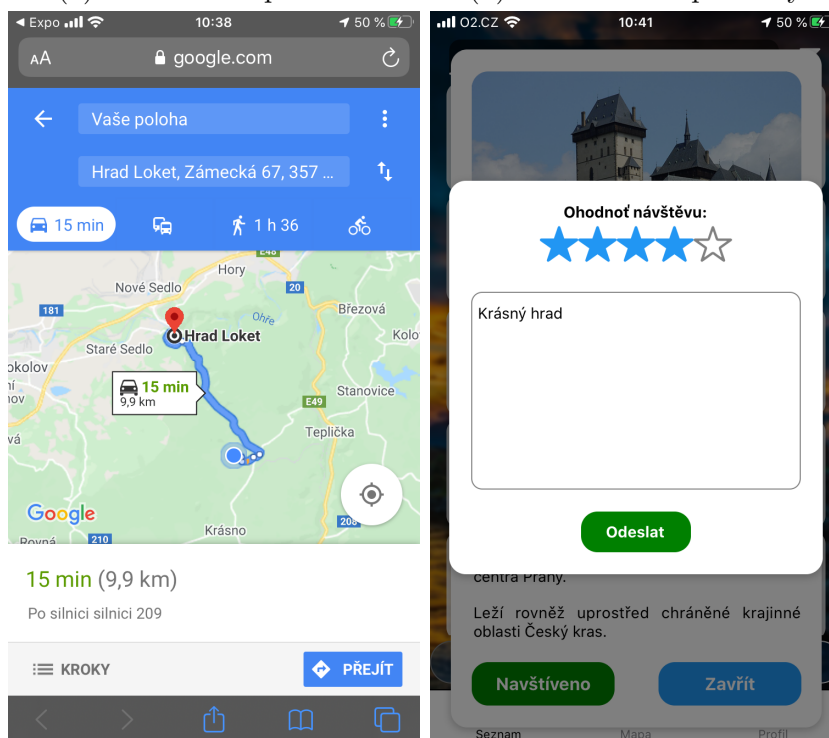
Navštívit památku lze vícekrát. Pro opakované uložení navštívení památky je nutné památku ohodnotit po uplynutí alespoň 24 hodin od posledního záznamu navštívení.

Zvolením tlačítka *Zavřít* se modální okno s informacemi o památce zavře.



(a) Informace o památce

(b) Komentáře u památky



(c) Navigace k památce

(d) Ohodnocení památky

Obrázek 8.9: Obrazovky s informacemi o památce

B.8 Profil

Na obrazovku s profilem (viz obrázek 8.10a) se lze přesměrovat z obrazovky se seznamem památek, nebo z obrazovky s mapou zvolením pravé ikony ve spodní části obrazovky.

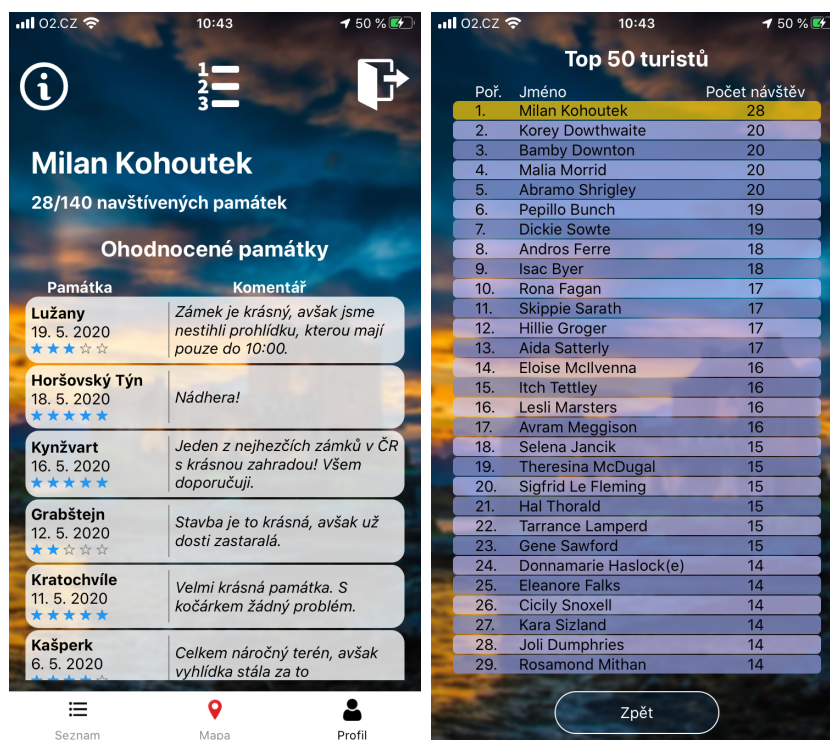
Je zde zobrazeno jméno zadané při registraci účtu a počet navštívených památek z celkového počtu.

Většinu obrazovky zabírá seznam navštívených památek. Jednotlivé dlaždice seznamu obsahují název památky, čas návštěvy, naše hodnocení památky a také komentář. Tento seznam je chronologicky seřazen dle data navštívení.

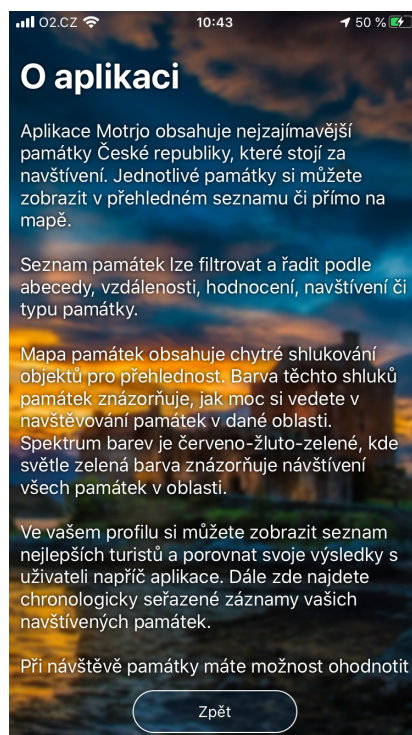
Zvolením pravé ikony v horní části obrazovky se zobrazí modální okno s informacemi o aplikaci viz obrázek 8.10c. Zde je popsáno základní používání aplikace, vysvětlení funkcionalit a také adresa pro zkontaktování správce aplikace. Zvolením tlačítka *Zpět* se modální okno zavře.

Zvolením prostřední ikony v horní části obrazovky se zobrazí žebříček padesáti uživatelů, seřazených dle počtu uskutečněných návštěv viz obrázek 8.10b. Zvolením tlačítka *Zpět* se modální okno zavře.

Z obrazovky s profilem je možnost přesměrování se na obrazovku s mapou (viz podkapitola B.6) nebo na obrazovku se seznamem památek viz podkapitola B.5.



(a) Obrazovka s profilovými údaji (b) Modální okno s žebříčkem uživatelů



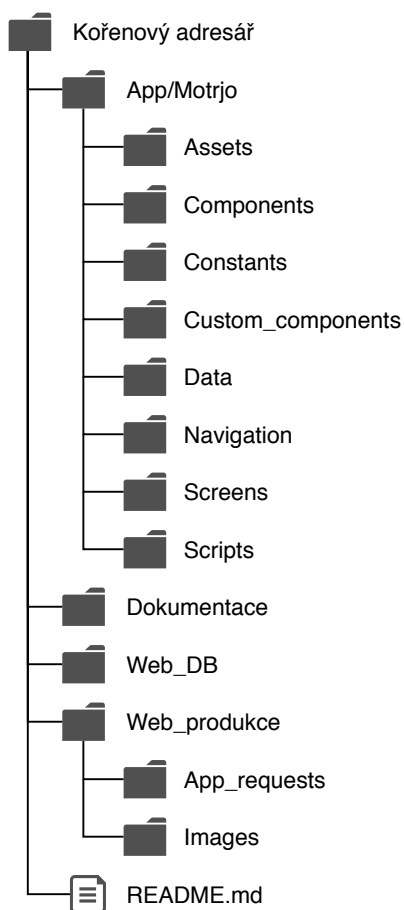
(c) Modální okno s informací o aplikaci

Obrázek 8.10: Obrazovka s profilem

C Obsah přiloženého CD

Adresářová struktura (viz obrázek 8.11) přiloženého CD obsahuje tyto podadresáře:

- App/motrjo – zdrojové soubory mobilní aplikace,
- Dokumentace – dokumentace bakalářské práce,
- Web_DB – SQL soubory pro vytvoření databází,
- Web_produkce – zdrojové soubory webové aplikace.
- README.md – textový soubor s adresářovou strukturou a s přihlašovacími údaji k webové aplikaci.



Obrázek 8.11: Adresářová struktura projektu