

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Offline webové aplikace
Bakalářská práce

Autor: Michal Koleč
Studijní obor: Aplikovaná informatika

Vedoucí práce: Mgr. Daniela Ponce, Ph. D.

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Polomi dne 29.4.2022

Michal Koleš

Poděkování:

Děkuji vedoucí bakalářské práce Mgr. Daniele Ponce, Ph. D. za metodické vedení práce a její rady, které přispěly k jejímu zdárnému dokončení. Také bych chtěl poděkovat svojí rodině, která mě podržela i ve chvílích, kdy šlo psaní pomalu.

Anotace

Bakalářská práce se zabývá offline webovými aplikacemi. Konkrétně aplikací odečtů, kterou autor dostal za úkol vytvořit. Hlavní cíl práce je analyzovat již existující aplikaci odečtů, která má nějaké nedostatky. Tyto nedostatky práce analyzuje a na jejich základě je navržena aplikace, která je schopná pracovat jak v online, tak v offline režimu. Poté je popsáno, jak je aplikace naprogramována, jaké technologie pro to byly využity. Je také popsáno, jakým způsobem jsou jednotlivé části aplikace implementovány, aby spolu mohly spolupracovat. V poslední části je aplikace testována v různých stavech. Práce ukazuje celý postup vytvoření offline webové aplikace. I když je tento postup konkrétní pro výše zmíněnou aplikaci měl by se dát použít i pro jiné aplikace podobného typu. Jsou také vyčleněny vhodné technologie pro takovou to aplikaci. Práce by měla vytvořit přehled o těchto technologiích a zjednodušit proces vybírání potřebných technologiích při tvoření dalších aplikací podobného typu. Tyto technologie a postupy byly otestovány v poslední části ze které vyplývá, že fungují a jsou vhodné pro tyto typy aplikací.

Annotation

Title: Offline web applications

The Bachelor thesis covers offline web applications. Specifically, application for readings. Which author got as a task to create. Main goal of the thesis is to analyze already existing application, which has some shortcomings. The thesis analysis these shortcomings and on their foundation a new app is designed, which can work both in online and offline mode. Then the way is described in which the app is programmed and which technologies were used to do so. It is also described in which way are individual parts of the app implemented. So, they could work together. In the last the app is tested in different states. The thesis shows the entire process of creating the offline web app. Even though the process is specific for the above-mentioned app. It should be sufficient to be used for other similar apps.

Technologies that are fitting for this app are also set aside. The thesis should create overview of these technologies and should make the process of choosing technologies for creating another application of similar type easier. These technologies and processes were tested in the last part, from which it is clear that they work and are fitting for these types of applications.

Obsah

1	Úvod.....	1
2	Cíl práce.....	3
3	Metodika zpracování.....	4
3.1	výzkumné otázky.....	4
3.2	Metodiky.....	4
4	Kapitola - Vlastní text práce	6
4.1	Analýza stávající aplikace a vytyčení jejích problémů.....	6
4.1.1	Popis aplikace	6
4.1.2	Nedostatky stávající aplikace.....	7
4.1.3	Výčet funkcí a požadavků pro Offline aplikaci	9
4.1.4	Soupis požadavků na Offline aplikaci	9
4.2	Návrh offline aplikace.....	10
4.2.1	Nalezené technologie pro použití v offline aplikaci	10
4.2.2	Vybrané technologie pro offline aplikaci.....	11
4.2.3	Datová struktura	14
4.2.4	Logické rozvržení aplikace	16
4.3	Programové řešení.....	18
4.3.1	Uživatelské rozhraní.....	23
4.4	Implementace	29
4.5	Testování aplikace v různých stavech.....	31
4.5.1	Simulace skutečné pochůzky se zapnutými daty.	31
4.5.2	Simulace pochůzky s vypnutými daty.	34
4.5.3	Simulace opakované pochůzky	36

4.6	Zabezpečení dat	38
5	Shrnutí výsledků.....	40
6	Závěry a doporučení	41
7	Seznam použité literatury.....	42
7.1	Tištěné zdroje	42
7.2	Internetové zdroje	42

Seznam obrázků

Obrázek 1 – Digram datové struktury	15
Obrázek 3 - Uložení údajů do local storage	18
Obrázek 2 - Ověření uživatele v offline režimu	19
Obrázek 5 - Práce s IndexedDB uložení číselníku důvodů odečtu.....	20
Obrázek 4 - Práce s IndexedDB založení a upgrade DB	20
Obrázek 6 - Service worker uložení do cache	22
Obrázek 7 – Hlavní obrazovka dávek.....	23
Obrázek 8 – Ikonka připojení k internetu	23
Obrázek 9 – Měřicí místa akce	24
Obrázek 10 – Měřicí místa navigační menu.....	24
Obrázek 11 – Synchronizace navigační menu	25
Obrázek 12 – Nastavení navigační menu.....	25
Obrázek 13 – Měřicí místa.....	26
Obrázek 14 – Měřicí místa filtry.....	27
Obrázek 15 – Měřicí místo stav	27
Obrázek 16 – Měřicí místo stav	27
Obrázek 17 – Formulář odečtu.....	28
Obrázek 18 – Nastavení odchylky	29
Obrázek 19 – Identifikace online	31
Obrázek 20 – Výběr měřicího místa	32
Obrázek 21 – Měřicí místo akce	32
Obrázek 22 – Vyplnění formulář odečtu.....	33
Obrázek 23 – Záznam v databázi z online zadání.....	33
Obrázek 24 – Počet zadaných měřicích míst.....	34
Obrázek 25 – Identifikace offline.....	34
Obrázek 26 – Vyplnění formulář odečtu z v režimu offline.....	35
Obrázek 27 – Záznam v indexedDB	36
Obrázek 28 - Nesynchronizováno	37
Obrázek 29 - Synchronizováno.....	37
Obrázek 30 – Záznam v databázi z offline zadání	37

Obrázek 31 – Formulář s hláškou o odečtu mimo přepokládaný stav.....38

Seznam tabulek

Tabulka 1 - Funkce odečtů 9

Tabulka 2 – Technologie offline web aplikace.....11

1 Úvod

Bakalářská práce se zabývá offline webovými aplikacemi. Konkrétně pak aplikací pro odečítání průměrného průtoku vody skrz odběrná místa.

Důvodem výběru tohoto tématu je autorova předešlá zkušenost s online a offline webovými aplikacemi. Problémová situace se naskytla při používání stávající aplikace odečtů. Ta při standardním přístupu k internetu funguje velmi dobře, jakmile ale dojde ke ztrátě připojení, aplikace se stane nefunkční a nepoužitelnou. V dnešní době již nebývá pokrytí signálem problém, ale vzhledem k reálnému umístění některých Měřících zařízení (MZ) na místech hůře dostupných a odstíněných (sklepy, šachty, suterény,...), naráželi odečítači na nemožnost použít aplikaci často. Ztratili tím přístup k informacím o Odběrném místě, údaje museli zapsat ručně někam do svých poznámek bez možnosti jejich přímého naskenování a vše doplňovat zpětně až ve chvíli, kdy opět měli přístup k internetu.

Vzhledem k výše uvedeným situacím dostal autor za úkol sestavit takovou aplikaci, která bude schopná nedostatky vyřešit. Stávající aplikace je online aplikace, která je zasazená do informačního systému, který slouží pro řízení chodu firmy. Není tedy možné extrahovat aplikaci do samostatné podoby a je nutné pracovat s již existujícím systémem. Ten pro potřeby této aplikace nabízí uživatelský systém se kterým aplikace pracuje, ale nebude dále popisován.

Z důvodu nutnosti zasazení do systému je třeba, aby byla aplikace webová. Desktopová aplikace tedy nepřipadá v úvahu. Kromě toho se předpokládá hlavní využití na mobilních zařízeních. Pro aplikaci s takovými požadavky je offline aplikace vhodná volba.

Proto se práce soustředí na takovou offline aplikaci, která bude splňovat požadavky zadavatele. Vychází z již existující aplikace a jde hlavně o vytvoření její funkčnosti v offline režimu.

I když je práce zaměřená na konkrétní aplikaci a případ použití, postup a technologie použité pro dosažení požadovaného výsledku se dají s malou modifikací využít i pro tvorbu dalších podobných aplikací.

2 Cíl práce

Autor si klade za cíl analyzovat existující online webovou aplikaci poskytující řešení ke sběru odečtů měřících zařízení (MZ) prostřednictvím pochůzek a vstupu dat s přímým online ukládáním na podnikový server.

Současně musí identifikovat problémy aplikace spočívající především v přístupu k podnikovému serveru přes nestabilní internet, případně úplný výpadek a nedostupnost internetu uvnitř budov a lokací MZ.

Cílem je také navrhnout a vyvinout aplikaci k řešení výše zmíněných problémů a následně vyhodnotit funkčnosti offlinové části této aplikace. Soupis jednotlivých problémů bude tvořit a specifikovat požadavky pro sestavení offline web aplikace, která bude nezávislá na přístupu k internetu v době pochůzky.

3 Metodika zpracování

3.1 Výzkumné otázky

Jaké jsou potřebné technologie pro fungování offline web aplikace?

Jak se tyto technologie implementují?

Poskytuje offline aplikace dostačující zabezpečení dat?

Bude takto sestavená aplikace skutečně fungovat v reálných situacích?

3.2 Metodiky

V první části práce je uveden všeobecný popis webové aplikace (technologie) a dále pak specializace na konkrétní aplikaci Odečty MZ a její funkce.

Vyzkoušením a úvahou nad funkcemi aplikace při vypnutém internetu vznikne soupis problémů, které je nutné vyřešit, aby aplikace byla schopná pracovat offline.

Autor chce docílit analýzou jednotlivých požadavků a jejich porovnáním s dostupnými webovými technologiemi, že je možné provozovat takovou aplikaci úplně offline v tom čase, který požaduje provozovatel aplikace, tj. v době pochůzek a sběru dat.

Autor předpokládá, že bude možné všechny požadavky pokrýt technologiemi dostupnými v čase vzniku této práce.

Cílem je najít pro každou požadovanou offline aplikační funkci (požadavek) webovou technologii zajišťující funkčnost na základě vlastních zkušeností nebo různých dostupných webových zdrojů.

Tyto nalezené technologie popsat, porovnat a srovnat podle vhodnosti použití k danému požadavku. V případě možnosti použití více technologií pro jeden požadavek, bude kritériem pro výběr v první řadě jednoduchost implementace, a dále pak velikost kódu a složitost kódu, doplňujícími kritérii může být náročnost

předpokládané údržby a jak případné změny této technologie mohou ovlivnit budoucí offline aplikaci a její údržbu.

Dalším krokem je vytvoření aplikace. Na základě zkušeností autora s tímto procesem jsou vytyčeny postupy a potřeby pro navržení a sestavení aplikace. Tyto postupy jsou pak dále popsány a měli by se dát využít pro sestavení dalších podobných aplikací.

Zda je aplikace funkční, a opravdu splňuje vytyčené požadavky, se ověří při testování se skutečnými daty, která otestují jednotlivé funkce aplikace a ověří jejich funkčnost při simulovaném provozu.

4 Kapitola - Vlastní text práce

4.1 Analýza stávající aplikace a vytyčení jejích problémů

4.1.1 Popis aplikace

Jedná se o komplexní webovou modulární aplikaci MVC. Hlavními znaky aplikace jsou ...

- Jeden přístupový bod
- Autentifikace – uživatel/heslo – Tabulka uživatelů a zakódovaných hesel
- Autorizace – přiřazení rolí pro MVC
 - o Tabulka rolí
 - o Tabulka user – role
 - o Tabulka role – res_akce
 - o Každá stránka je zabezpečena pomocí resource a akce, kterou musí mít uživatel přiřazenu
- PHP – generuje stránky HTML, JS, CSS [1]
 - o Stránky jsou většinou připravovány dopředu
 - o Na základě šablonování připraví celou stránku
- Klient – zobrazení stránek PC, Mobil
 - o formulářová výběrová pole nezávislá – příprava dopředu
 - o formulářová pole závislá – příprava Ajaxem [3] přes JS
- GUI
 - o TOP Menu (Moduly – Controlery)
 - o Menu akcí stránek (Funkce)
- Modul Odečty MZ
 - o Online aplikace

Aplikace slouží pro evidenci pochůzek pracovníků (odečítačů) provozovatele společnosti sbírajících data z měřících zařízení (vodoměrů). Pochůzka je definována jako soubor odběrných míst s instalovanými MZ, která je potřeba navštívit a ručně odečíst a zkontrolovat. Provedení odečtu probíhalo dříve do zápisových archů, ale dnes již probíhá do on-line aplikace ovšem s problémy.

Jde o jednoduchý aplikační modul skládající se ze čtyř základních částí. První část je obsluha plnění údajů o pochůzkách importem ze zákaznického systému. Systém eviduje číslo pochůzky, datum, počet MZ, počet vyřízených MZ, počet nezjištěných MZ, počet nevyřešených MZ. Dále pak obsahuje informace o MZ jako je adresa, GPS, odběratel, kontakt, předchozí spotřeba, datum posledního stavu, poslední stav a pořadí.

Druhá část pracuje s pochůzkami. V této části vedoucí organizuje práci pracovníkům přidělováním pochůzek. Po dokončení pochůzky exportuje data zpět do zákaznického modulu ke zpracování fakturace.

Třetí část se zabývá provedením pochůzky. Uživatel (odečítač) má možnost otevřít pochůzku a vyplnit odečet pro konkrétní vybrané MZ. Vyplňuje tyto údaje: datum odečtu, odečtený stav MZ a důvod odečtu (př. fakturace a kontrolní odečet). Nebo označí MZ jako nezjištěné (př. nezastižen, nefunkční MZ) a datum. Další možností je speciální režim pochůzka. V režimu pochůzka se opakuje zadávací obrazovka pro jednotlivá MZ v pochůzce v předem stanovené pořadí. Pracovník pak jen doplňuje údaje odečtu podobně jako v jednotlivém režimu.

Při uložení odečtu jsou data validována na serveru na odchylky od předpokládaného stavu. Ty musí uživatel vyřešit poznámkou a potvrzením před uložením. Uživatel má ve třetí části data možnost pouze číst nebo i zapisovat.

Čtvrtá část je konfigurační. Jsou zde ukládány parametry validace odečtů. Jedná se o formulář s uložením několika číselných hodnot.

4.1.2 Nedostatky stávající aplikace

Hlavní nedostatek aplikace spočívá v neumožnění uživateli zadávat data ve chvíli, kdy nemá přístup k internetu, což je u odběrových stanic, které se nachází v terénu, běžné. To v praxi znamená, že odečítači si musí odběr zaznamenat někde u sebe a teprve až když se znovu připojí k internetu, ho můžou zadat do aplikace. To vytváří zbytečné chyby, komplikace a velkou časovou náročnost. Dalším podstatným nedostatkem stávající aplikace je omezený přehled dat pochůzky – jsou zde všechny údaje, ale zobrazení na mobilním zařízení a filtrování není moc praktické a

přehledné. Zobrazení jen neodečtených a nezkontrolovaných MZ nebo jen přeskočených (příklad z důvodu nezastižen) je obtížné.

Odečítač ne vždy provádí odečítání podle stanového pořadí pochůzky. Potřebuje dynamicky vyhledávat podle aktuální potřeby průchodu dávkou.

Základním požadavkem na pochůzku je zobrazení dle pořadí trasy. Dalšími požadavky jsou vyhledání konkrétního čísla popisného pro případ, kdy odečítač začne odečtem někde uprostřed trasy, když se nachází v dané oblasti mimo svoji obvyklou pochůzku. Dále pak vyhledání podle čísla parcelního nebo dalších údajů v poznámce v menším množství filtrů. Stávající aplikace má filtr ke každému sloupci. Protože jsou sloupce roztahané a je jich hodně, zadání filtru trvá velmi dlouho.

Dalším požadovaným výběrem je filtrování ještě nezadaných stavů v kombinaci dalšího filtrování. A posledním požadavkem, který aktuální aplikace neřeší, je zobrazení zadaných, ale neodečtených stavů, pro usnadnění přehledu opakovaných návštěv.

4.1.3 Výčet funkcí a požadavků pro Offline aplikaci

Funkce modulu Odečty	Web Offline
Přihlášení uživatele Autentifikace	Prvotní přihlášení online a pak uložení přihlašovacích údajů na klientovi
Moje pochůzky	Synchronizace dat z moje pochůzky Online a uložení na klientovi
Pochůzka – seznam MZ	Synchronizace dat z moje pochůzky Online a uložení na klientovi
Pochůzka – odečet MZ	Synchronizace dat z moje pochůzky Online a uložení na klientovi
Konfigurace validace	Prvotní načtení konfigurace online a pak možnost uložení a změny na klientovi
Validace odečtu před uložení	Načtení konfigurace a validace se zadaným stavem na klientovi bez odeslání dat
Uložení dat odečtu	Uložení dat na klientovi do pochůzky, synchronizace zpět na server
Generování stránek a zobrazení	Načtení stránky z Cache a její spuštění, kód již musí být kompletní celé stránky včetně všech podstránek a formulářů. Případné generování stránky na klientovi s přístupem k uloženým datům.

Tabulka 1 - Funkce odečtů

4.1.4 Soupis požadavků na Offline aplikaci

Zdrojem pro tento soupis je předchozí odstavec. Nejedná se výlučně o požadavky na offline aplikaci, ale uvedené požadavky je potřebné řešit jinak v offline režimu práce.

Běžné

- Zajištění spuštění aplikačního kódu bez připojení k serveru
- Autentifikace pro danou session
- Navigace a generování aplikačního kódu
- Čtení lokálních dat
- Vstup, validace a ukládání lokálních dat
- Synchronizace dat a jejich validace na serveru

Speciální

- Použití senzorů mobilního zařízení (např. GPS – lokalizace a umístění bodů do mapy pochůzky, Bluetooth – komunikace a ovládání dalších zařízení př. MZ, Kamera – QR kódy načtení a ověření čísla MZ)

4.2 Návrh offline aplikace

Na základě výše uvedených nedostatků a stanovených požadavků je aplikace navržena takto

4.2.1 Nalezené technologie pro použití v offline aplikaci

Klient – webový prohlížeč (běžná součást Javascriptu)

Local/Session Storage – Dříve bylo možné data na klientovi ukládat jenom v cookies, ty měly ale pro některé potřeby nedostatek paměti. Proto Web Storage umožňuje uložit větší množství informací. Data jsou exkluzivně uložena na klientovi. Velikost se liší podle prohlížeče, ale většinou je kapacita okolo 5 MB. Vhodné pro typ dat klíč-hodnota. (převzato a upraveno z [5], 2021)

Cache – Slouží k uložení odpovědí serveru na dotazy klienta. Klient si zde může uložit stránky nebo další objekty jako obrázky, fonty, CSS pro pozdější použití v offline režimu, když nebudou online dostupné.

IndexedDB – Slouží pro ukládání většího množství dat na klientovi, umožňuje ukládat i soubory. Vhodné pro cache data. Data jsou na rozdíl od Local Storage strukturovaná. Používá indexy pro rychlé vyhledávání. (převzato a upraveno z [4], 2020)

Service Worker – Je to speciální typ Web workera. Program také v javascriptu, který běží v pozadí nezávisle na webové stránce. Umožňuje přerušit a zpracovat requesty, řídí chod aplikace v režimu offline. Řídí ukládání stránek do cache. (převzato a upraveno z [6], 2020)

Aplikační server – nová část serveru (backend) zpracovávající synchronizaci dat

Databázový server – Se používá pro ukládání dat na serveru a jejich zpětné načtení. Dá se zálohovat. Nově validuje synchronizovaná data přicházející při synchronizaci z offline aplikace.

4.2.2 Vybrané technologie pro offline aplikaci

Funkce modulu	Web Online	Web Offline	Vybrané technologie
Odečty			
Přihlášení uživatele Autentifikace	Proti tabulce uživatelů	Poprvé online, poté proti uloženým údajům	Local Storage
Moje pochůzky	Generování stránky – data z SQL	Generování stránky – data z IndexedDB	IndexedDB
Pochůzka – seznam MZ	Generování stránky – data z SQL	Generování stránky – data z IndexedDB	IndexedDB
Pochůzka – odečet MZ	Generování stránky – data z SQL	Generování stránky – data z IndexedDB	IndexedDB
Konfigurace validace	Generování stránky – data z SQL	Generování stránky – data z Local Storage	Local Storage
Validace odečtu před uložením	Aplikační server PHP	Klient Javascript	Javascript, Local Storage
Generování stránek a zobrazení	Dynamické PHP, HTML	Předgenerované HTML z Cache, změny a navigace Javascript	Cache, Javascript, Local Storage, IndexedDB
Synchronizace dat	Ne	Ano	Javascript, IndexedDB

Tabulka 2 – Technologie offline web aplikace

Co je IndexedDB

IndexedDB je databáze, která je zabudována do prohlížeče a je mnohem výkonnější než localStorage.

Ukládá téměř jakýkoli druh hodnot podle klíčů, více typů klíčů. Podporuje transakce pro spolehlivost. Podporuje dotazy na klíčový rozsah, indexy. Dokáže ukládat mnohem větší objemy dat než localStorage. Tato síla je u tradičních aplikací klient-server obvykle nadměrná. IndexedDB je určen pro offline aplikace, které mají být kombinovány s ServiceWorkers a dalšími technologiemi. (převzato a upraveno z [7],2021)

IndexedDB je transakční systém obdobně jako jsou SQL databáze. Rozdíl je hlavně v tom, jak ukládají a definují data. SQL používá pevnou strukturu sloupců v tabulce. Webové prohlížeče poskytují k IndexedDB API. IndexedDB Javascript API pracuje s JSON objekty uloženými v databázi pod klíčem.

Pojmy v IndexedDB

Databáze – Toto je nejvyšší úroveň IndexedDB. Obsahuje úložiště objektů, která zase obsahují data, která chcete uchovat. Můžete vytvořit více databází s libovolnými názvy, ale běžně je použita jedna databáze na aplikaci. (převzato a upraveno z [8],2021)

Úložiště objektů – Úložiště objektů je samostatný segment pro ukládání dat. Úložiště objektů si můžete představit jako podobné tabulkám v tradičních relačních databázích. Obvykle existuje jedno úložiště objektů pro každý typ dat (objekt). Pro řešení offline aplikace bude návrh obsahovat seznam těchto objektů například Odečtová dávka, Seznam MM, Odečty MM. (převzato a upraveno z [8],2021)

Index – Index je druh úložiště objektů pro uspořádání dat v jiném úložišti objektů (nazývané úložiště referenčních objektů) podle individuální vlastnosti dat. Index se používá k načítání záznamů v úložišti objektů pomocí této vlastnosti. Například provázání seznamu MM přes vlastnost IDdávky na objekt Odečtová dávka. (převzato a upraveno z [8],2021)

Operace – interakce s databází. Načtení záznamu, uložení záznamu apod.

Transakce – transakce označuje začátek a konec operace nebo skupiny operací, která zajišťuje integritu databáze. Pokud některá z operací v rámci transakce selže, žádná z nich se nepoužije a databáze se vrátí do stavu, ve kterém byla před zahájením transakce. Všechny operace čtení nebo zápisu v IndexedDB musí být součástí transakce. To umožňuje atomické operace čtení-úpravy-zápisu, aniž byste se museli starat o to, že na databázi budou současně působit jiná vlákna. (převzato a upraveno z [8], 2021)

Kurzor – Mechanismus pro iteraci přes více záznamů v databázi obdoba průchodu jako v SQL databázi. (převzato a upraveno z [8], 2021)

Co je ServiceWorker

Service Worker (do češtiny by se dalo přeložit jako Pracovní služba, ale přesnější je Služba na pozadí) umožňuje stránce spouštět kód v pozadí a provádět akce, i když není stránka připojená k serveru (např. je uživatel bez internetu).

Service Worker (dále jen služba) se z vaší stránky nainstaluje do prohlížeče a prohlížeč si zaregistruje, na které události chce služba reagovat a pokud k takové události dojde, spustí příslušný kód služby. (převzato a upraveno z [9],2019)

Další vlastnosti ServiceWorker

Zobrazení v offline režimu

Služba může reagovat na událost fetch, která se vyvolá kdykoliv vaše stránka požádá server o nějaký soubor (HTML, CSS, obrázek, atd.). Služba pak může soubor stáhnout sama a uložit si ho do cache. Pokud pak uživatel přejde do offline režimu a bude chtít znovu navštívit danou stránku, služba může daný soubor vrátit z cache a fungovat tedy i bez internetu. V případě offline režimu není nutné zobrazovat stejný obsah jako online, ale je možná zjednodušená verze stránky. (převzato a upraveno z [9],2019)

Stahování a odesílání dat na pozadí

Sledováním událostí sync a online dokáže služba poznat, kdy uživatel přešel do online režimu a může například odeslat data, která uživatel uložil, když by offline, nebo může naopak stáhnout ze serveru aktualizace dat a uložit je do cache (odkud se pak mohou zobrazit v offline režimu). (převzato a upraveno z [9],2019)

4.2.3 Datová struktura

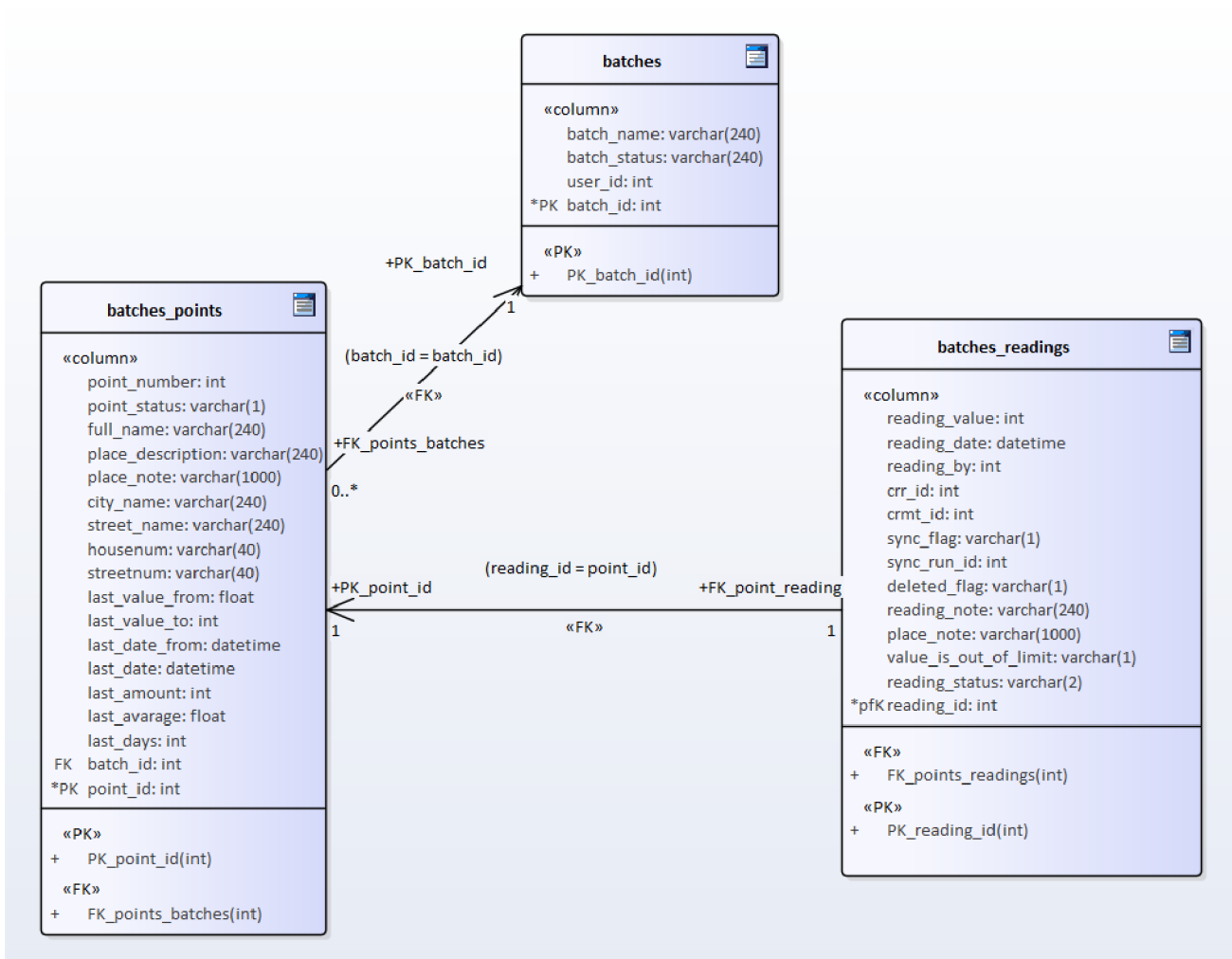
Pro správné fungování aplikace je důležité navrhnout datovou strukturu, která bude přehledná a usnadní práci s daty. Rozdělí jednotlivé body odběru podle místa, kde se nacházejí do jednotlivých pochůzek. Tato struktura se skládá ze tří hlavních tabulek, které dohromady umožňují zaznamenat data pro jednotlivé pochůzky, jejich body a jednotlivá měření.

Tabulka pochůzek obsahuje údaje o konkrétních pochůzkách, jejich názvy a stavy. Na tuto základní tabulku se ostatní tabulky odkazují a napojují.

Tabulka bodů je seznam jednotlivých měřících míst, která se nacházejí v dané pochůzce a odkazují se na ID hlavní tabulky. Na tato ID navazuje pořadové číslo a díky němu je jasné pořadí v pochůzce, což odečítačům podstatně zjednodušuje jejich orientaci. Aby bylo zřejmé, která MM má přiřazena který odečítač, musí existovat o všech odečítačích záznam v tabulce. Ten je pro potřeby této práce vyřešen jednoduše pouze údajem o jméně a příjmení. Při implementaci do skutečného systému by musel existovat odkaz na uživatele systému pro další potřeby a např. kooperaci s dalšími systémy, které odečítači naplánují trasu apod. Dále je v tabulce údaj o tom, jestli je místo otevřené nebo uzavřené. Uzavřené místa se vůbec nezobrazují a měření se na nich neprovádí. Dále je důležité zaznamenat konkrétní místo, kde se takový odběrový bod nachází. Pro tyto účely slouží sloupce Popis místa, Poznámka místa, Název města, Číslo domu a Číslo ulice. Tato dohromady tvoří dostačující záznam o adrese místa i případných upřesňujících poznámkách. I když jednotlivé odečty zaznamenává až třetí tabulka, je dobré udržovat základní souhrnné údaje hodnot, které slouží pro rychlou orientaci a jednodušší práci s údaji.

Konkrétně jde o poslední odečtenou hodnotu, datum posledního odečtu, poslední známou hodnotu spotřeby a průměrnou spotřebu dle posledních dnů.

Na tuto tabulku se pomocí ID bodu navazuje právě třetí tabulka. V té jsou jednotlivé odběry navázány na příslušné řádky, pro které zaznamenávají jednotlivé hodnoty. Konkrétně jsou to hodnota odečtu, datum provedení odečtu. Podstatný je také odkaz na člověka, který odběr provedl. Dále se zaznamenává sloupeček, který ukazuje na to, zda byl řádek synchronizován spolu s odkazem na konkrétní číslo synchronizace. Tabulka také obsahuje doplňující poznámky o konkrétním čtení. Je také potřeba zaznamenat, jestli hodnota nepřekročila limit o to se postará další sloupeček. A jako poslední sloupeček tabulky je status odečtení.



Obrázek 1 – Digram datové struktury

4.2.4 Logické rozvržení aplikace

Dávky (Pochůzky)

Nová offline web aplikace pro sběr dat z měřících zařízení (MZ) vychází vzhledově z původní online firemní webové aplikace. Důraz je zde kladen na jednoduchost ovládání i spuštění aplikace. Nový návrh pro použití na mobilním zařízení má přispět také k lepšímu přehledu a filtrování v rámci pochůzky, to v původní aplikaci není.

Po spuštění aplikace musí tato otestovat stav přihlášení uživatele. Pokud je již přihlášen, zobrazí mu aktuální seznam aktivních pochůzek. Pokud není uživatel přihlášen, vyžádá si jeho přihlášení zobrazením přihlašovacího formuláře. Po odeslání dat uživatel/web přihlašovacím formulářem dojde k autentifikaci uživatele online na firemním serveru nebo online oproti uloženým údajům z offline úložiště. Autorizace dalších funkcí již nebude probíhat. Uživatel získává plný přístup ke všem funkcím a datům již staženým do mobilního zařízení. Po neúspěšném přihlášení zůstane aktivní přihlašovací formulář a vypíše se hláška o neúspěšném přihlášení. Po úspěšném přihlášení dojde k přesměrování na seznam aktivních pochůzek.

Uživatelské rozhraní nabídne uživateli 4 záložky k práci.

Dávky

První záložka jsou dávky. Ta se zobrazí po přihlášení. Seznam dávek bude zobrazen tabulkovým přehledem a bude obsahovat sloupec s údaji ID dávky, název dávky (pochůzky), Stav (pouze) A aktivní, Počet MM – celkový počet MZ v dávce, Zadáno MM – počet již odečtených nebo navštívených míst. Oranžový čtvereček znamená připravená data k synchronizování. Dále pak tlačítko pro výběr dávky.

Měřící Místa

Druhou záložkou v pořadí budou Měřící místa. Na Měřící místa může přepnout uživatel v seznamu dávek tlačítkem vpravo na dané dávce nebo přímo na záložce

Měřicí místa. Ta zobrazí seznam MM poslední vybrané dávky. Pokud ještě nebyla dávka vybraná, přesměruje odkaz na záložce MM zpět na záložku Dávky.

Měřicí místa budou také zobrazena tabulkovým přehledem. U většího počtu MM bude uživatel rolovat posunem dolů a nahoru. Pro zjednodušení vyhledání konkrétních míst budou na začátku stránky záložky umístěna filtrovací políčka a tlačítko pro rychlé přepínání přednastavených filtrů.

Tabulkové zpracování bude odlišné od běžného zobrazení. K jednomu MM se musí vejít více zobrazovaných údajů. Barevně proto budou odlišena jednotlivá MM, předpokládají se 3 řádky pro jedno MM.

Synchronizace

Další záložkou je synchronizace. Oproti prvním dvěma záložkám se stiskem záložky spustí rovnou funkce synchronizace. Synchronizace se spustí pouze v případě, že je zařízení se spuštěnou aplikací online. Synchronizace bude dvojího druhu. Synchronizace směrem na server UP a synchronizace směrem do zařízení doma. Synchronizace ze serveru směrem do zařízení bude spuštěna jen na vyžádání, obvykle na začátku pracovní směny (pochůzky). Při této synchronizaci dojde k aktualizaci číselníků důvodů odečtu a aktivních dávek a jejich měřících míst. Synchronizace směrem ze zařízení na server může být spuštěna také na vyžádání a zároveň je automaticky spouštěna po uložení záznamu odečtu nebo po připojení online po předchozím odpojení.

Nastavení

Poslední záložka Nastavení bude sloužit k zadání konfiguračních hodnot validačních algoritmů zadávaných hodnot z měřících zařízení. Kontrolním údajem bude maximální spotřeba – systém po zadání zkontroluje zadanou hodnotu, spočítá z ní spotřebu a tu porovná s touto konfigurační hodnotou. Pokud je zadaná hodnota vyšší, vyžádá si od uživatele aplikace potvrzení po vložení. Uživatel ověří správnost hodnoty a tu pak definitivně potvrdí nebo ji opraví.

4.3 Programové řešení

Program byl vytvořen jako Jednostránková JS aplikace. Všechny její části (přihlašovací obrazovky, záložky) jsou překreslovány pomocí JS do jedné zobrazované HTML stránky.

Při načítání dat z IndexedDB jsou pomocí JS sestaveny zobrazovací a ovládací prvky a menu, které umožňuje přepínat funkce.

Synchronizace je řešena pomocí AJAX [3] volání serverového REST API. Výměnným formátem jsou JSON data odečtů a dávek.

Přihlášení uživatele – Autentifikace

Při přihlašování musí být uživatel ověřen oproti existujícím údajům. V online aplikaci jsou uživatelské údaje uloženy v databázi a při zadání požadavku je server z databáze přečte a pošle klientovi. Aby se mohl uživatel ověřit offline, musí být tato data k dispozici. To znamená, že se uživatel musí alespoň jednou přihlásit do aplikace online, aby mohlo dojít k prvotnímu stažení dat. Pro taková data je vhodné využít Local Storage, protože se ukládá jen velmi malý objem data. Následně při dalším offline požadavku na přihlášení je možné uživatele vůči těmto datům ověřit. Local Storage splňuje kritérium jednoduchosti implementace a je vhodným k použití, na rozdíl od IndexedDB, které by znamenalo složitější implementaci a více programového kódu. Výhodou Local Storage je synchronní uložení i načtení hodnot oproti asynchronnímu u IndexedDB.

```
if (myObj.status == 'LogIn') {  
    heslo = myObj.password;  
    myObj.password = CryptoJS.MD5(heslo).toString(CryptoJS.enc.Base64);  
    localStorage.setItem("xxmklogon_" + myObj.username + "_" + myObj.server, JSON.stringify(myObj));  
}
```

Obrázek 2 - Uložení údajů do local storage

```

server = document.getElementById("frm-signInForm-onlineweb").value;
username = document.getElementById("frm-signInForm-username").value;
password = document.getElementById("frm-signInForm-password").value;

logonl = JSON.parse(localStorage.getItem("xxmklogon_"+username+"_"+server));

if (CryptoJS.MD5(password).toString(CryptoJS.enc.Base64)==logon.password) {
  xxmk_status = 'LogIn';
  xxmk_userid = logon.userid;
  document.getElementById("loguser").innerHTML=username;
  xxmkGetPage('batches');
}

```

Obrázek 3 - Ověření uživatele v offline režimu

Moje pochůzky, Pochůzka – seznam MZ, Pochůzka – odečet MZ

Pro tyto části aplikace musí být k dispozici data z online verze aplikace. Těchto dat bude mnohem více a jsou zároveň strukturovaná do více úrovní (Pochůzky – MZ k pochůzce – Odečet k MZ). Zobrazení dat je buď jen seznam pochůzek, nebo seznam MZ k jedné pochůzce včetně odečtu k MZ (jeden k MZ), nebo formulář pro zadání odečtu s údaji ze všech úrovní a navíc číselníkem pro výběrová pole formuláře. Local Storage v tomto případě, i když má jednodušší implementaci, by nebyl vhodným nástrojem. Proto je výhodnější využít IndexedDB pro uložení většího množství dat na klientovi a to v požadované struktuře.

ObjectStore představuje základní obdobu objektu ukládaného v IndexedDB. Je obdobou tabulky v SQL. Musí tedy obsahovat primární klíč a ostatní požadované sloupce (zde položky objektu). To, jak celá struktura dat vypadá, je definováno verzí databáze při jejím založení. IndexedDB podporuje také aktualizaci struktury DB. Při otevření DB s novou verzí je původní přepsána novou a jsou zavolány příslušné rutiny pro její update (onupgradeneeded).

Pokud potřebujeme například data filtrovat, jsou k dispozici dvě možnosti. Buď budeme při čtení dat na výstupu některá data vynechávat (programově) nebo použijeme Index součást IndexedDB. Například index pro pochůzku v seznamu MZ. Při transakci čtení dat pak jednoduše dotážeme index místo Objectstoru a získáme tak již předfiltrovaná data.

```

req.onupgradeneeded = function(event) {
  console.log("init onupgradeneeded indexedDB ");
  var db = event.target.result;
  xxmkDB = db;

  db.onerror = function(event) {
    console.log("deletion or creation objects failed");
  };

  var x = db.objectStoreNames;
  console.log("objectStoreNames");
  console.log(x);
  for (j = 0; j < x.length; j++) {
    console.log(x[j]);
    var osr = db.deleteObjectStore(x[j]);
  }

  for (var i in IDBSetting.tables) {
    var OS = db.createObjectStore(IDBSetting.tables[i].tableName, {
      keyPath: IDBSetting.tables[i].keyPath,
      autoIncrement: IDBSetting.tables[i].autoIncrement
    });

    for (var j in IDBSetting.tables[i].index) {
      OS.createIndex(IDBSetting.tables[i].index[j], IDBSetting.tables[i].index[j], {
        unique: IDBSetting.tables[i].unique[j]
      });
    }
  }
}

```

Obrázek 5 - Práce s IndexedDB založení a upgrade DB

```

if (myObj.data.reasons.length!==undefined) {
  var t_RR = xxmkDB.transaction(["mr reasons"], "readwrite");
  t_RR.oncomplete = function(event) {
    console.log('t_RR completed: database modification finished');
  };
  t_RR.onerror = function(event) {
    console.log('t_RR not opened due to error. Duplicate items not allowed.');
```

```

  };

  var objectStore_RR = t_RR.objectStore("mr reasons");
  for (i = 0; i < myObj.data.reasons.length; i++) {
    objectStore_RR.put(myObj.data.reasons[i]);
    console.log(myObj.data.reasons[i].name);
  }
}

```

Obrázek 4 - Práce s IndexedDB uložení číselníku důvodů odečtu

Konfigurace validace

Při prvním použití aplikace stáhneme ze serveru konfigurační hodnoty a uložíme je do validačních pravidel na klientovi. Je vhodné použít Local storage. Opět jde o jednoduchou implementaci a je možné pracovat synchronně. Jedná se o několik konfiguračních hodnot, které se nemění moc často, spíše výjimečně například v odečtu pro danou lokaci apod. Načtení do formuláře pro změnu a uložení změn bude tedy jednoduché při použití Local storage a ne IndexedDB.

Validace odečtu před uložením

Při každém spuštění stránky offline aplikace bude vhodné si v proměnných v JS uložit načtením z Local storage konfigurační parametry. Tyto proměnné je dobré udržovat i při změně v konfiguraci. Validování vstupních dat odečtů podle konfiguračních parametrů, pak může probíhat pouze v javascriptu a nebude dále nutné již provádět načtení hodnot opakovaně z Local storage.

Generování stránek a zobrazení

Při úvodním spuštění aplikace se vygenerují celé stránky a uloží se na klientovi v Cache. Poté při požadavku na stránku v offline režimu se Service worker postará o zobrazení stránky. Přerušuje request a místo typické odpovědi od serveru se místo přesměrování jen překreslí stránka podle celé uložené stránky a doplní se daty uloženými na klientovi. O zobrazení dat se stará část programu v JS, která načte data z IndexedDB a vloží je do určitého místa na předgenerované stránce vytažené z Cache.

```

self.addEventListener('fetch', function(event) {
  console.log(event.request.url);
  event.respondWith(
    caches.match(event.request)
      .then(function(response) {
        // Cache hit - return response
        if (response) {
          return response;
        }

        return fetch(event.request).then(
          function(response) {
            // Check if we received a valid response
            if(!response || response.status !== 200 || response.type !== 'basic') {
              return response;
            }
            var responseToCache = response.clone();
            caches.open(CACHE_NAME)
              .then(function(cache) {
                cache.put(event.request, responseToCache);
              });

            return response;
          }
        );
      })
  );
});

self.addEventListener('activate', function(event) { ...

```

Obrázek 6 - Service worker uložení do cache

Synchronizace dat

Při vytvoření, úpravě a odstranění dat v offline režimu je nezbytné data ukládat na klientovi. Synchronizovat má smysl pouze měněné údaje z IndexedDB. Při následném získání připojení je nutné srovnat stavy na klientovi a v databázi na serveru a poté je uvést do stejného stavu. Vhodným řešením je použití běhu programu na pozadí ve Web workeru, čímž nebude synchronizace nijak omezovat uživatele v zadávání a ovládání aplikace.

Je důležité také vyřešit, jaké změny mají přednost, například změny provedené na serveru i na klientovi pro stejné MZ. Například určíme, že prioritu dat mají změněná data na klientovi.

4.3.1 Uživatelské rozhraní

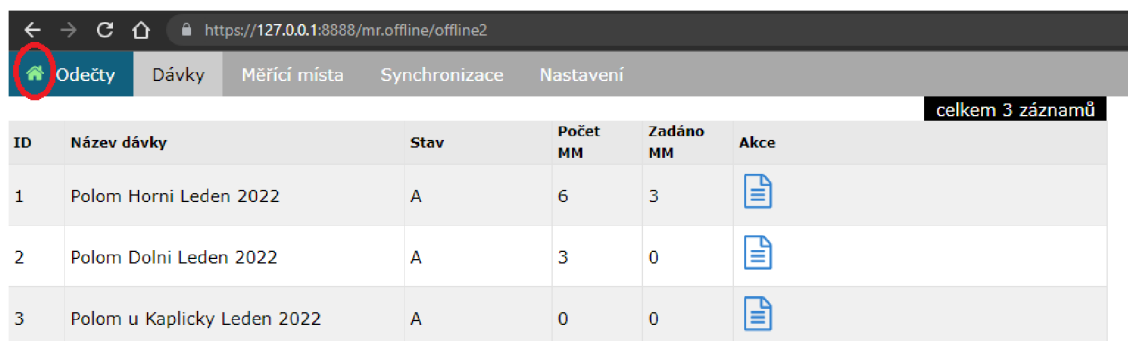


ID	Název dávky	Stav	Počet MM	Zadáno MM	Akce
1	Polom Horni Leden 2022	A	6	3	
2	Polom Dolni Leden 2022	A	3	0	
3	Polom u Kaplicky Leden 2022	A	0	0	

Obrázek 7 – Hlavní obrazovka dávek

Na hlavní obrazovce aplikace lze najít seznam všech dávek, který slouží jako orientační základ. Vidíme tu přehledně celkový počet Měřících míst v jedné dávce a počet již zadaných míst. Nahoře se nachází navigační lišta, která umožňuje přepínání mezi záložkami (obrazovkami).

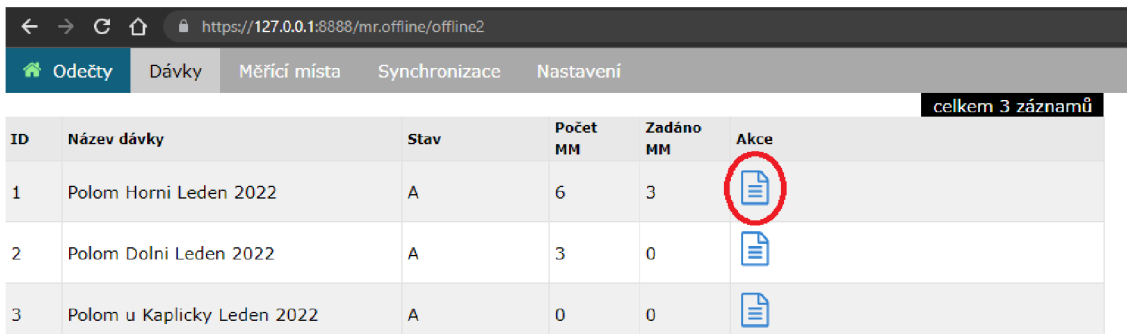
Dále se zde nachází dům, který znázorňuje připojení k serveru. Pokud svítí zeleně je připojení v pořádku a při zadání odečtů se data okamžitě synchronizují. Taktéž synchronizace dávek je možná. Pokud ale svítí červeně je aplikace v offline režimu a synchronizace není možná. Data se ukládají pouze lokálně a synchronizují se až při následném opětovném připojení.






ID	Název dávky	Stav	Počet MM	Zadáno MM	Akce
1	Polom Horni Leden 2022	A	6	3	
2	Polom Dolni Leden 2022	A	3	0	
3	Polom u Kaplicky Leden 2022	A	0	0	

Obrázek 8 – Ikonka připojení k internetu

Na hlavní stránce je dále akce, která nastaví dávku jako vybranou a přesměruje aplikaci na detailní přehled měřících míst.



ID	Název dávky	Stav	Počet MM	Zadáno MM	Akce
1	Polom Horni Leden 2022	A	6	3	
2	Polom Dolni Leden 2022	A	3	0	
3	Polom u Kaplicky Leden 2022	A	0	0	

celkem 3 záznamů

Obrázek 9 – Měřící místa akce

Navigační položka měřící místa přesune uživatele na přehled měřících míst, spadajících pod vybranou dávku. Zde je bude moci i zadávat. Aktivní neboli vybraná dávka se nastaví akcí popsanou výše.



ID	Název dávky	Stav	Počet MM	Zadáno MM	Akce
1	Polom Horni Leden 2022	A	6	3	
2	Polom Dolni Leden 2022	A	3	0	
3	Polom u Kaplicky Leden 2022	A	0	0	

celkem 3 záznamů

Obrázek 10 – Měřící místa navigační menu

Další záložka v navigačním menu je Synchronizace, která aktualizuje data dávek. Z tabulek nahraje jednotlivé dávky, údaje o nich a všechna měřící místa, která pod ně náleží. Jde o předem vytvořenou strukturu, kterou budou odečítači využívat v terénu a zapisovat do ní data.

ID	Název dávky	Stav	Počet MM	Zadáno MM	Akce
1	Polom Horni Leden 2022	A	6	3	
2	Polom Dolni Leden 2022	A	3	0	
3	Polom u Kaplicky Leden 2022	A	0	0	

celkem 3 záznamů

Obrázek 11 – Synchronizace navigační menu







Jako poslední záložka na liště s navigací je Nastavení. Ta přesune uživatele do jednoduché formuláře, kde se dají nastavit proměnné rozsahy.

ID	Název dávky	Stav	Počet MM	Zadáno MM	Akce
1	Polom Horni Leden 2022	A	6	3	
2	Polom Dolni Leden 2022	A	3	0	
3	Polom u Kaplicky Leden 2022	A	0	0	

celkem 3 záznamů

Obrázek 12 – Nastavení navigační menu

Obrazovka měřícího místa zobrazuje jednotlivé řádky pro aktivní dávku. Aktivní dávka se nachází nahoře na obrazovce. Hned pod ní jsou již jednotlivé záznamy měřících míst, které zobrazují údaje o nich jako jsou adresa a Jméno odběratele.

Odečty		Dávky	Měřící místa	Synchronizace	Nastavení	
č.Parcel.	čp	Q	Q			
ID	Název dávky	Stav	Počet MM			
1	Polom Horni Leden 2022	A	6			
Poř.	Spínač	Fólio	Vx	Datum	O.	Akce
4	undefined	undefined	undefined			 
undefined			Prijmeni 4, Jmeno 4			
Polom			mob. 799156894			
čp. 16						
5	undefined	undefined	undefined			 
undefined			Prijmeni 5, Jmeno 5			
Polom			mob. 799156895			
čp. 17						
6	undefined	undefined	undefined			 
undefined			Prijmeni 6, Jmeno 6			
Polom			mob. 799156896			
čp. 37						

Obrázek 13 – Měřící místa

Na horní části obrazovky se nachází filtry pomocí kterých se dá v seznamu měřících míst vyhledávat podle různých kritérií. Je například možné zobrazit jen neodečtená či jen odečtená.



Obrázek 14 – Měřicí místa filtry

Čtverec na obrazovce znázorňuje stav měřicího místa. Červená znamená, že je místo nezadané. Dále může mít barvu oranžovou, která se objevuje jen v offline režimu kdy je sice měřicí místo zadané, ale není synchronizované se serverem. Jakmile dojde k obnovení spojení, stav se aktualizuje a čtverec se změní na zaškrtnutý čtverec zelené barvy. To je stav, kdy je měřicí místo jak zadané tak synchronizované.

Poř.	Spínač	Fólio	Vx	Datum	O.	Akce
1	undefined	undefined	undefined			
undefined			Prijmeni 7, Jmeno 7			
Polom			mob. 799156891			
čp. 33						

Obrázek 15 – Měřicí místo stav

Jednotlivé měřicí místa se dají odečíst stisknutím tlačítka ve sloupci akce. To přenese uživatele na obrazovku odečtu.

Poř.	Spínač	Fólio	Vx	Datum	O.	Akce
1	undefined	undefined	undefined			
undefined			Prijmeni 7, Jmeno 7			
Polom			mob. 799156891			
čp. 33						

Obrázek 16 – Měřicí místo stav

Ve formuláři pro zadání odečtu jsou zobrazeny údaje sloužící k ověření, že uživatel zvolil správné měřicí místo a dávku. Dále má uživatel možnost vybrat z listu hodnot důvod odečtu. Zadá zde konkrétní hodnotu odečtu a případnou poznámku k průběhu odečítání. Formulář obsahuje validaci hodnoty odečtu, který musí být v určitém rozsahu. Ten je určen proměnnými hodnotami v záložce Nastavení. Pokud odečet není v rozsahu, je možné tento krok, po provedené kontrole správnosti, potvrdit žaskrtnutím potvrzovacího políčka, které se v takovém případě nabídne.

Odečty	Dávky	Měřicí místa	Synchronizace	Nastavení
Data				
Dávka	Polom Dolni Leden 2022			
Odběratel	Prijmeni 7, Jmeno 7			
Adresa	Polom, 33			
Umístění	ve sklepe			
Pozn. MM	mob. 799156891			
Vodoměr	undefined			
Datum stavu:	15.06.2021			
Poslední	rozdíl / odečet			
	168	/	18963	
Datum:	31.01.2022 13:50			
Důvod:	Fakturační odečet			
Odečet:				
Poznámka				
Dokončení zadání				
<input type="button" value="Uložit data"/>				
<input type="button" value="Vymazat odečet"/>				

Obrázek 17 – Formulář odečtu

V podslední záložce Nastavení je jednoduchý formulář, kde je možné zadat odchylku od průměrné spotřeby. Tato funce slouží pro upozornění uživatele na výkyvné odečty. Uživatel může také změnit maximální spotřebu.

Odečty Dávky Měřicí místa Synchronizace Nastavení

Nastavení

Max. - odchylka od Ø spotřeby (%)	<input type="text" value="25"/>
Max. + odchylka od Ø spotřeby (%)	<input type="text" value="25"/>
Maximální spotřeba (m3)	<input type="text" value="1000"/>

Dokončení zadání

Obrázek 18 – Nastavení odchylky

4.4 Implementace

Stávající aplikace je zasazena do komplexnějšího systému, který nabízí spoustu podstatných funkcí. Ty jsou pro samostatné fungování této aplikace i její zasazení a využití ve firemním systému nezbytné. Tyto již vyřešené funkce jako jsou GUI - Grafické uživatelské rozhraní, přihlášení pro jednotlivé uživatele atd., nebude práce sice implementovat, ale pro zjednodušení budou v programu využity. Práce se bude soustředit jen na části podstatné pro vytvoření a následné správné fungování aplikace v offline podobě.

Pro fungování všech částí dohromady jako jedné aplikace je potřeba několik komponent.

Instalace databázové vrstvy

Pro účely offline aplikace bylo nutné rozšířit základní datovou strukturu odečtové aplikace o další vrstvu a to API pro synchronizaci dat. Nově bylo nutné zaznamenávat průběh synchronizace a prováděné změny dat. Proto byly přidány příslušné tabulky a view a bylo přidáno logování změn pomocí DB trigrů. K implementaci byly připraveny skripty, které vše do DB nainstalují.

Instalace Webserveru a PHP

Tato vrstva, jak bylo zmíněno výše, je použita pro zpřístupnění offline aplikace klientovi. Vše běží na jednom aplikačním portu včetně odkazu na offline aplikaci či synchronizace. [1]

Implementace Klient

Klientská část aplikace je propojena se serverovou částí kromě přenosu dat také v její instalaci. Pokud se klient připojí do online části, nalezne tam odkaz pro přepnutí do offline aplikace. Po stisknutí odkazu server vygeneruje celou aplikaci včetně všech JS skriptů a pošle ji klientovi. Klientovi se aplikace stáhne a již dříve uvedené technologie zabezpečí, že daný odkaz na offline aplikaci bude funkční i při testovaných stavech nedostupnosti serveru nebo internetu.

K fungování klient potřebuje prohlížeč jako jedinou aplikaci a při první implementaci nebo při následných aktualizacích přístup do online verze. Pro synchronizaci dat potřebuje přístup do API serverové části, která je součástí online verze aplikace.

Implementační postup – Instalace aplikace

Odečtová aplikace je součástí celého balíku aplikací firemního frameworku. Pro testovací implementaci je nutné použít základní Framework a doinstalovat dvě součásti nové offline odečtové aplikace.

První částí je instalace programů pro offline aplikaci. Druhá část, jejíž instalace také probíhá nakopírováním do adresářové struktury aplikace, se zabývá vygenerováním kódu offline aplikace běžící na klientovi a odesláním kódu na klienta.

Pro propojení a snadnou implementaci na klientovi (viz. Dále) bylo nutné umístění offline aplikace vhodně rozdělit na část generovanou, jejímž obsahem je hlavní výkonný skript Aplikace v JS, tak část obslužnou (negerovanou), jejíž součástí je Service worker zajišťující offline provoz programu.

4.5 Testování aplikace v různých stavech

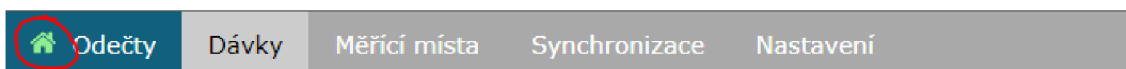
Aplikaci je nezbytné otestovat, aby se ověřilo jak celý proces předávání dat a přepínání mezi režimy funguje. Nestačí jen ověřit nově přidané části, ale jde i o to, jak bude aplikace fungovat jako celek s již zaběhnutými částmi aplikace. Pro tyto účely se použijí data z testovací lokace. Nejdříve se otestuje aplikace online režimu. Zde by se neměly žádné problémy objevit. Jde o neupravovanou část, proto není nutné, aby byl objem testovacích dat příliš velký. Následně se ověří funkčnost offline části. Zde je žádoucí aplikaci prověřit lépe, a tak je zvolený objem dat větší. A jako poslední se otestuje přechod mezi módy aplikace a také různé situace, které by mohly nastat a kde je možné, že by se objevila nějaká nekonzistence.

4.5.1 Simulace skutečné pochůzky se zapnutými daty.

Tento typ testování se zaměří na aplikaci v online režimu. Připojení k internetu bude k dispozici po celou dobu pochůzky. Postupně se vykoná 30 odečtů. Bude se sledovat, zda se zadaná data dostanou do databáze, zda odpovídají realitě. Také půjde o to, zda se aplikace správně synchronizuje a zadaná data se hned zobrazují v seznamu.

Aplikace má již nějaká měřící data zadaná, aby byl simulován skutečný průběh pochůzky. Kdy by její přerušení nemělo způsobit žádné problémy.

Nejprve bude ověřeno, zda aplikace detekuje, že je přístup k internetu. To se ověří podle ikonky, která toto zobrazuje. V levém horním rohu pokud svítí ikonka domu zeleně aplikace si myslí, že je připojená k internetu. Na obrázku je vidět že tomu tak skutečně je.



Obrázek 19 – Identifikace online

Dále se ze seznamu dávek vybere správná dávka v tomto případě „Polom Dolni Duben“.

Odečty Dávky Měřicí místa Synchronizace Nastavení celkem 3 záznamů					
ID	Název dávky	Stav	Počet MM	Zadáno MM	Akce
1	Polom Horni Leden 2022	A	6	3	
2	Polom Dolni Duben 2022	A	3	1	
3	Polom u Kaplicky Duben 2022	A	0	0	

Obrázek 20 – Výběr měřícího místa

Po vybrání se zobrazí seznam měřících míst (MM) pro konkrétní vybranou dávku. Bez nastavování filtrů se zobrazí jen nezadané MM. Kdyby bylo potřeba opravit, nebo pracovat již se zadaným MM lze tak docílit pomocí filtrů. Konkrétně zmáčknutím červeného tlačítka nahoře, které nastaví filtr na všechny MM. Poté se vybere MM, na které odběratel stojí.

Odečty Dávky Měřicí místa Synchronizace Nastavení						
<input type="text" value="č.Parcel."/> <input type="text" value="čp"/>						
ID	Název dávky	Stav	Počet MM			
2	Polom Dolni Duben 2022	A	3			
Poř.	Spínač	Fólio	Vx	Datum	O.	Akce
2	undefined	undefined	undefined			
undefined			Prijmeni 8, Jmeno 8			
Polom			mob. 799156892			
čp. 30						
3	undefined	undefined	undefined			
undefined			Prijmeni 9, Jmeno 9			
Polom			mob. 799156893			
čp. 20						

Obrázek 21 – Měřicí místo akce

Zobrazí se konkrétní formulář pro zadání odečtu. Do formuláře se doplní důvod odečtu, hodnota odečtu a případná poznámka. A kliknutím na tlačítko uložit data se formulář odešle.

Odečty Dávky Měřicí místa Synchronizace Nastavení

Data

Dávka: Polom Dolni Duben 2022

Odběratel: Prijmeni 8, Jmeno 8

Adresa: Polom, 30

Umístění: ve sklepě

Pozn. MM: mob. 799156892

Vodoměr: undefined

Datum stavu: 15.06.2021

Poslední: rozdíl / odečet

141 / 17884

Datum: 28.04.2022 10:51

Důvod: Fakturační odečet

Odečet: 18100

Poznámka:

Dokončení zadání

Uložit data

Vymazat odečet

Obrázek 22 – Vyplnění formulář odečtu

Zadané data by se nyní měla zobrazit v databázi, jak lze vidět na obrázku opravdu tomu tak je.

id	reading_value	reading_date	reading_by	corr_id	crmt_id	sync_flag	sync_run_id	mrs_readings_id	deleted_flag	reading_note	place_note	value_is_out_of_limit	reading_status
1	13000	2022-01-30 16:20:00	NULL	13	NULL	Y	1	NULL	N	NULL	mob. 799156899	N	NULL
2	13020	2022-01-31 10:15:00	NULL	13	NULL	Y	2	NULL	N	NULL	mob. 799156892	N	NULL
3	12400	2022-01-31 10:21:00	NULL	13	NULL	Y	3	NULL	N	NULL	mob. 799156893	N	NULL
7	19200	2022-01-31 16:17:00	NULL	13	NULL	Y	4	NULL	N	NULL	mob. 799156891	N	NULL
8	18100	2022-04-28 10:51:00	NULL	13	NULL	Y	5	NULL	N	NULL	mob. 799156892	N	NULL
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Obrázek 23 – Záznam v databázi z online zadání

Navíc se na hlavní straně aplikace u dávky „Polom Dolni Duben“ zobrazuje počet zadaných MM 2.



ID	Název dávky	Stav	Počet MM	Zadáno MM	Akce
1	Polom Horni Leden 2022	A	6	3	
2	Polom Dolni Duben 2022	A	3	2	
3	Polom u Kaplicky Duben 2022	A	0	0	

celkem 3 záznamů

Obrázek 24 – Počet zadaných měřících míst

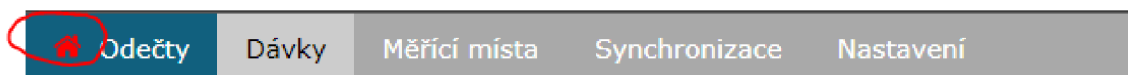
Tento proces se opakuje 30. Mněl by pokaždé proběhnout stejně s jediným rozdílem v zadaných hodnotách.

Testování proběhlo v pořádku. Jak lze vidět z obrázků data jsou správně nahrána v databázi a zobrazují se správně i v aplikaci. I u dalších testů se data správně uložila a vše se správně zobrazilo. Práce v online režimu je konzistentní a spolehlivá.

4.5.2 Simulace pochůzky s vypnutými daty.

V tomto případě se bude při testování simulovat nepokrytí oblasti internetovým připojením. A to tak, že se zařízení, na kterém se provádí odečty, odpojí od internetu vypnutím mobilních dat. Ověří se, zda jsou k dispozici data synchronizovaná před opojením internetu. A dále se budou postupně zadávat data z vykonání 60 odečtů. Zkontroluje se, zda se tato data ukládají a poté zobrazují lokálně na testovacím zařízení.


Opět bude nejprve ověřeno, zda aplikace správně detekuje přístup k internetu v tomto případě by měla ikonka svítit červeně. Jak lze ověřit na obrázku ikonka svítí správně.



Obrázek 25 – Identifikace offline

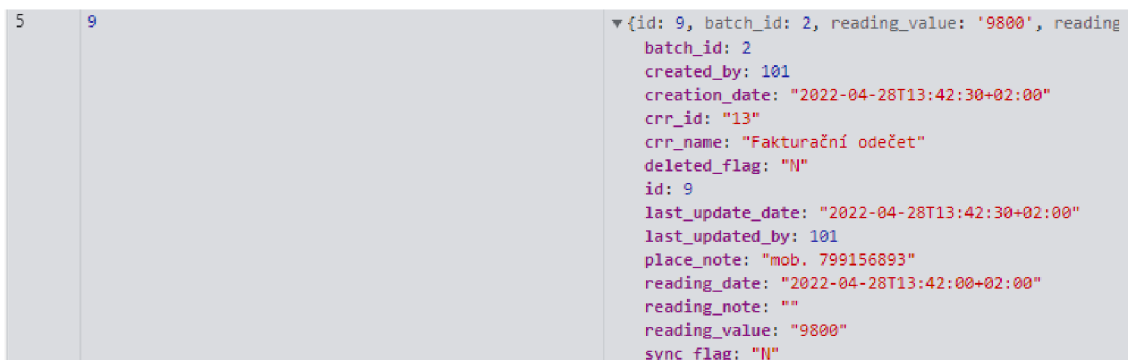
Postup bude velmi podobný jako v online testování s tím rozdílem že se data budou ověřovat v lokálním uložení místo v databázi.

Opětovně se vybere dávka „Polom Dolni Duben 2022“. Následně se zvolí MM a zobrazený formulář se vyplní.

Odečty	Dávky	Měřicí místa	Synchronizace	Nastavení
Data				
Dávka	Polom Dolni Duben 2022			
Odběratel	Prijmeni 9, Jmeno 9			
Adresa	Polom, 20			
Umístění	ve sklepe			
Pozn. MM	mob. 799156893			
Vodoměr	undefined			
Datum stavu:	15.06.2021			
Poslední	rozdíl / odečet			
	31	/	9631	
Datum:	28.04.2022 13:42			
Důvod:	Fakturační odečet ▼			
Odečet:	9800			
Poznámka				
Dokončení zadání				
Uložit data				
Vymazat odečet				

Obrázek 26 – Vyplnění formulář odečtu z v režimu offline

Nyní se ověří, zda jsou data skutečně lokálně uložena. Na obrázku lze vidět záznam v IndexedDB, který je uložený v prohlížeči.



```
{id: 9, batch_id: 2, reading_value: '9800', reading_batch_id: 2, created_by: 101, creation_date: '2022-04-28T13:42:30+02:00', crr_id: '13', crr_name: 'Fakturační odečet', deleted_flag: 'N', id: 9, last_update_date: '2022-04-28T13:42:30+02:00', last_updated_by: 101, place_note: 'mob. 799156893', reading_date: '2022-04-28T13:42:00+02:00', reading_note: '', reading_value: '9800', sync_flag: 'N'}
```

Obrázek 27 – Záznam v indexedDB

Testování znovu proběhlo bez problémů. Data jsou tam kde mají být a aplikace v offline režimu reaguje, jak se očekávalo.



4.5.3 Simulace opakované pochůzky

Posledním typem testování se zaměří na přepínání mezi oběma módy aplikace. Jde hlavně o synchronizaci dat oběma směry. Přenesení dat, která jsou uložena lokálně na testovacím zařízení, do databáze. A také aktualizace dat v lokální podobě na novou verzi dostupnou díky znovu nabití internetového připojení. Dále se také otestuje několik dalších situací, jako je např. upozornění uživatele na odchylky hodnot od průměru.

Začátek testování proběhne v online módu. Jak už bylo ukázáno v minulých testováních, nejprve se ověří stav aplikace, zda je opravdu online. Následně proběhne zadání odečtu a aplikace se synchronizuje. Bude simulováno odpojení od internetu ikonka domu vlevo nahoře by opět měla svítit červeně. Poté se ověří, zda se data zadaná v online režimu promítla i do offline režimu. Po tomto testu se předešlé zadané odečty objevily i po spuštění aplikace offline režimu.



Druhý test proběhne opačně, začne se v offline režimu a zadá se odečet.

Po zadání odečtu vypadá v aplikaci vše jako v online módu kromě upozornění oranžového čtverce na to, že odečet není synchronizovaný.

3	undefined	undefined	undefined	28.04.2022	9800		
				Fakturační odečet	169		
undefined				Prijmeni 9, Jmeno 9			
Polom				mob. 799156893			
čp. 20							

Obrázek 28 - Nesynchronizováno

Bude se simulovat opětovné připojení k internetu. Ikonka internetu by měla svítit zeleně. A ukazatel u konkrétního MM by se měl změnit na zelený.

3	undefined	undefined	undefined	28.04.2022	9800		
				Fakturační odečet	169		
undefined				Prijmeni 9, Jmeno 9			
Polom				mob. 799156893			
čp. 20							

Obrázek 29 - Synchronizováno


Poté se aplikace synchronizuje a ověří se, zda se data dostala až do databáze. Na obrázku je vidět, že se skutečně záznam objevil i v databázi.

id	reading_value	reading_date	reading_by	cr_id	ormt_id	sync_flag	sync_run_id	mrs_readings_id	deleted_flag	reading_note	place_note	value_is_out_of_limit	reading_status	creation_date
1	13000	2022-01-30 16:20:00	XXXX	13	XXXX	Y	1	XXXX	N	XXXX		N	XXXX	2022-01-30 16:21:05
2	13020	2022-01-31 10:15:00	XXXX	13	XXXX	Y	2	XXXX	N	XXXX	mob. 799156892	N	XXXX	2022-01-31 10:20:54
3	12400	2022-01-31 10:21:00	XXXX	13	XXXX	Y	3	XXXX	N	XXXX	mob. 799156893	N	XXXX	2022-01-31 10:22:01
7	19200	2022-01-31 16:17:00	XXXX	13	XXXX	Y	4	XXXX	N	XXXX	mob. 799156891	N	XXXX	2022-01-31 16:17:17
8	18100	2022-04-28 10:51:00	XXXX	13	XXXX	Y	5	XXXX	N	XXXX	mob. 799156892	N	XXXX	2022-04-28 10:51:38
9	9800	2022-04-28 13:42:00	XXXX	13	XXXX	Y	6	XXXX	N	XXXX	mob. 799156893	N	XXXX	2022-04-28 13:42:30

Obrázek 30 – Záznam v databázi z offline zadání

Poslední test ověří fungování kontroly odchylky odečtu a to v offline režimu. Ve formuláři se zadá odečet, který je mimo předpokládané hodnoty. Měla by vyskočit hláška, která uživatele na tuto skutečnost upozorní. Uživatel může tuto hlášku ignorovat a zadat odečet i tak. Varování slouží jen pro upozornění v rozdílech hodnot. Mohlo by to například znamenat únik vody. Hláška se objevila, když byl zadán odečet mimo předpokládaný rozsah a objevilo se i zaškrtnuté pro potvrzení mimo rozsah.

Data

Dávka	<input type="text" value="Polom Dolni Duben 2022"/>
Odběratel	<input type="text" value="Prijmeni 9, Jmeno 9"/>
Adresa	<input type="text" value="Polom, 20"/>
Umístění	<input type="text" value="ve sklepe"/>
Pozn. MM	<input type="text" value="mob. 799156893"/>
Vodoměr	<input type="text" value="undefined"/>
Datum stavu:	<input type="text" value="15.06.2021"/>
Poslední	rozdíl / odečet
	<input type="text" value="31"/> / <input type="text" value="9631"/>
Datum:	<input type="text" value="28.04.2022 13:42"/> 
Důvod:	<input type="text" value="Fakturační odečet"/>
Odečet:	<input type="text" value="9900"/> Hodnota je větší než předpokládaný stav !
Poznámka	<input type="text"/>

Dokončení zadání

Potvrďte odečet mimo rozsah

Obrázek 31 – Formulář s hláškou o odečtu mimo předpokládaný stav

Všechny testy proběhly, jak bylo očekáváno. Aplikace funguje v pořádku. Zvolené technologie a návrh aplikace jsou dostatečnými k fungování aplikace ve vyžadovaných situacích.

4.6 Zabezpečení dat

Na rozdíl od online aplikace musí offline aplikace pracovat s lokální kopií dat. Zabezpečení dat na úrovni online aplikace se tedy soustředuje na oblasti přístupu do aplikace a zabezpečení serveru apod.

Zabezpečení offline aplikace musí řešit bezpečné uložení dat na klientovi a chránit k nim přístup.

Zabezpečit data je třeba na více úrovních. Fyzický přístup k rozhraní – lze vyřešit pracovními postupy. Uživatel by neměl ukládat heslo pro přístup do aplikace. Uživatel by neměl na zařízení vykonávat jiné činnosti než pracovní.

Úroveň přihlašování a práce s aplikací. K offline ověření se používají uložené přihlašovací údaje. Z tohoto důvodu je vhodné kromě ztížení fyzického přístupu k zařízení mít také plán pro případ odcizení zařízení, jeho ztrátu apod. Například v případě ztráty zařízení okamžitá blokáce přístupových údajů na serveru, protože je offline aplikace s online propojena. Případně používáním různých hesel pro přístup online a offline. Mít možnost vzdáleně vymazat data ze zařízení, případně provést úplný reset zařízení.

Fyzická zařízení HW šifrování!

Z pohledu úrovně uložených dat na klientovi, která nejsou šifrována, by bylo vhodné navrhnout způsob jejich šifrování. Dalším možným řešením by bylo pravidelné čištění dat na klientovi po ukončení práce a následná nová synchronizace další den před jejím zahájením.

5 Shrnutí výsledků

Cílem práce bylo analyzovat stávající aplikaci odečtů. A na základě této analýzy sestavit seznam požadavků a technologií, které budou potřebné k sestavení nové aplikace. Která bude schopná pracovat i v oblastech bez přístupu k internetu.

Práce popisuje celý postup vytvoření Offline webové aplikace.

Podařilo se analyzovat nedostatky původní aplikace. Ty potom sloužili jako základ pro výběr technologií potřebných pro správné fungování aplikace. Následně se podařilo navrhnout novou aplikaci, která splňuje vytyčené požadavky.

V průběhu testování se nezjistili žádné problémy. Technologie, které byly vybrány pro aplikaci se jeví jako vhodné. Aplikace je opravdu schopná pracovat i bez přístupu k internetu splňuje tedy stanovené požadavky.

6 Závěry a doporučení

Vytvořená offline aplikace se zaměřuje na konkrétní jednu úlohu. Zajímavým tématem pro další rozvoj aplikace z pohledu funkčnosti a otestování offline vlastností by mohlo být otestování a využití dalších možností fyzického zařízení.

- 1) Fotoaparát a ukládání fotek se zachyceným stavem měřícího zařízení, použití GPS a zobrazení mapy, což by ovlivnilo velikost dat potřebných ukládat offline apod.
- 2) Využití offline aplikace k dalšímu firemnímu účelu jako jsou například různé výkony práce, podklady k práci, dokumenty atd.

V průběhu vývoje hlavní části offline aplikace, tedy zobrazování, načtení a ukládání offline dat, se objevilo několik problémů, se kterými autor nepočítal. Jedním z nejzásadnějších je práce s IndexedDB. Pracuje se s ní jinak a dle autora obtížněji než s běžnými databázovými nástroji. V rámci vývoje aplikace tak bylo obtížné oddělit práci s daty samostatně od zobrazení. Na této problematice je tedy možné dále pracovat a například vytvořit nástroj, který by potřebné části JS kódu generoval automaticky ze zadaných požadavků jako jsou: obsah stránky, grid, sloupec, data a generování by vytvořilo šablonu pro update/create IndexedDB, zobrazení stránky a dotazy do IndexedDB.

7 Seznam použité literatury

7.1 Tištěné zdroje

- [1] LAVIN, P. PHP – objektově orientované: koncepty, techniky a kód. Praha: Grada, 2009. 224 s. ISBN: 978-80-247-2137-8.
- [2] SPAINHOUR, S. a ECKSTEIN, R. Webmaster v kostce. Praha: Computer Press, 1999. 504 s. ISBN: 80-7226-450-8.
- [3] ASLESON, R. a SCHUTTA, N. T. AJAX: Vytváříme vysoce interaktivní webové aplikace. Brno: Komputeru Press, 2006. 269 s. ISBN: 80-251-1285-3.

7.2 Internetové zdroje

- [4] Shoemaker, Craig. in: www.codemag.com [online]. 2020, poslední úpravy 18.6.2020 [cit. 20. 6. 2021]. Dostupné z: <https://www.codemag.com/article/1112051/Build-an-HTML5-Offline-Application-with-Application-Cache-Web-Storage-and-ASP.NET-MVC>
- [5] MDN Web Docs. in: www.developer.mozilla.org [online]. 2021, poslední úpravy 14.6.2021 [cit. 20. 6. 2021]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API
- [6] Gaunt, Matt. in: www.developers.google.com [online]. 2020, poslední úpravy 24.7.2020 [cit. 20. 6. 2021]. Dostupné z: <https://developers.google.com/web/fundamentals/primers/service-workers>
- [7] Javascript info. in: www.javascript.info [online]. 2021, poslední úpravy 12.12.2021 [cit. 29. 4. 2022]. Dostupné z: <https://javascript.info/indexeddb>
- [8] Developers google. in: www.developers.google.com [online]. 2021, poslední úpravy 1.5.2019 [cit. 29. 4. 2022]. Dostupné z: <https://developers.google.com/web/ilt/pwa/working-with-indexeddb>
- [9] Nothrem. in: js.nothrem.cz [online]. 2021, poslední úpravy 18.6.2019 [cit. 29. 4. 2022]. Dostupné z: <http://js.nothrem.cz/service-worker>



Zadání bakalářské práce

Autor:	Michal Koleč
Studium:	I1900118
Studijní program:	B1802 Aplikovaná informatika
Studijní obor:	Aplikovaná informatika
Název bakalářské práce:	Offline webové aplikace
Název bakalářské práce AJ:	Offline web applications
Garantující pracoviště:	Katedra informačních technologií, Fakulta informatiky a managementu
Vedoucí práce:	Mgr. Daniela Ponce, Ph.D.
Datum zadání závěrečné práce:	21.1.2020