



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

SYSTEM VZDÁLENÉHO MONITOROVÁNÍ PŘÍZNAKŮ PARKINSONOVY NEMOCI

REMOTE PARKINSON'S DISEASE MONITORING SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Pavel Podlužanský

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jiří Mekyska, Ph.D.

BRNO 2023

Diplomová práce

magisterský navazující studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Bc. Pavel Podlužanský

ID: 205867

Ročník: 2

Akademický rok: 2022/23

NÁZEV TÉMATU:

Systém vzdáleného monitorování příznaků Parkinsonovy nemoci

POKYNY PRO VYPRACOVÁNÍ:

Cílem této práce je implementovat systém vzdáleného monitorování motorických a nemotorických příznaků Parkinsonovy nemoci na základě předloženého prototypu realizovaného v prostředí Figma. Jednu část systému (obsluhovaná pacienty) bude tvořit mobilní aplikace (podporována systémy iOS i Android), která bude sloužit ke sběru dat. Sběr bude prováděn pomocí různých senzorů mobilního zařízení, např. pomocí mikrofonu nebo akcelerometru. Další data budou sbírána pomocí elektronických dotazníků. Data budou přenášena na server, kde k nim budou přistupovat neurologové pomocí webového rozhraní (data bude možné stahovat/vizualizovat). Zároveň budou moci neurologové pacientům vytvořit časový plán různých vyšetření. Velký důraz bude kladen na zabezpečení celého systému tak, aby neunikla osobní/zdravotní data.

DOPORUČENÁ LITERATURA:

[1] ABBOTT, Devin, Houssein DJIRDEH, Anthony ACCOMAZZO a Sophia SHOEMAKER. Fullstack React Native: Create beautiful mobile apps with JavaScript and React Native. New York: Fullstack.io, 2019. ISBN 978-1728995557.

[2] FLANAGAN, David. JavaScript: the definitive guide: master the world's most-used programming language. Seventh edition. Beijing: O'Reilly, 2020. ISBN 978-1491952023.

Termín zadání: 6.2.2023

Termín odevzdání: 19.5.2023

Vedoucí práce: Ing. Jiří Mekyska, Ph.D.

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Parkinsonova choroba je neliečiteľná choroba, ale s vhodným monitorovaním sa dá výrazne zlepšovať stav pacientov. Práca sa zaoberá vytvorením systému na monitorovanie Parkinsonovej choroby. Takýto systém dokáže neurológom aj pacientom ušetriť drahocenný čas a zároveň môže neurológom umožniť venovať sa väčšiemu počtu pacientov. Systém je implementovaný, zabezpečený a otestovaný podľa návrhu. Systém dokáže spoľahlivo vyhodnocovať výsledky vyšetrení od pacientov a odosielať ich neurológom. Jednotlivé vyšetrenia patria do rôznych kategórií. Pacienti sú tak monitorovaní na základe pohybových, dotazníkových, či rečových vyšetrení.

KĽÚČOVÉ SLOVÁ

Parkinsonova choroba, progresívna webová aplikácia, React, JavaScript

ABSTRACT

Parkinson's disease is an incurable disease, but with appropriate monitoring, the condition of patients can be significantly improved. The thesis deals with the creation of a system for monitoring Parkinson's disease. The system can save both neurologists and patients valuable time, and at the same time, it can allow neurologists to take care of more patients. The system is implemented, secured and tested as designed. The system can reliably evaluate the results of examinations from patients and send them to neurologists. Individual examinations belong to different categories. Patients are monitored based on movement, questionnaire or speech examinations.

KEYWORDS

Parkinson's disease, progressive web application, React, JavaScript

PODLUŽANSKÝ, Pavel. *Systém vzdáleného monitorování příznaků Parkinsonovy nemoci*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2022, 40 s. Diplomová práce. Vedúci práce: Ing. Jiří Mekyska, Ph.D.

Vyhlásenie autora o pôvodnosti diela

Meno a priezvisko autora: Bc. Pavel Podlužanský
VUT ID autora: 205867
Typ práce: Diplomová práca
Akademický rok: 2022/2023
Téma záverečnej práce: Systém vzdáleného monitorování pří-
znaků Parkinsonovy nemoci

Vyhlasujem, že svoju záverečnú prácu som vypracoval samostatne pod vedením vedúcej/cého záverečnej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej záverečnej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno

.....

podpis autora*

*Autor podpisuje iba v tlačenej verzii.

POĎAKOVANIE

Rád by som poďakoval vedúcemu diplomovej práce pánovi Ing. Jiřímu Mekyskovi, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Obsah

1	Úvod	9
2	Systém na monitorovanie priebehu Parkinsonovej choroby	10
2.1	Parkinsonova choroba	10
2.1.1	Príčiny vzniku	10
2.1.2	Symptómy	11
2.2	Existujúce riešenia aplikácii na monitorovanie Parkinsonovej choroby	11
2.2.1	Apkinson	12
2.2.2	StrivePD	12
2.2.3	Limitácie existujúcich riešení	12
2.3	Návrh a koncepty systému	12
2.3.1	Pacient	13
2.3.2	Neurológ	14
2.3.3	Typy vyšetrení	15
2.4	Prípady použitia aplikácie (Use cases)	16
2.4.1	Pacient	16
2.4.2	Neurológ	17
3	Progresívna webová aplikácia (PWA)	18
3.1	Service worker	18
3.2	Web app manifest	20
3.3	Push notifikácie	20
3.4	Dostupnosť bez ohľadu na pripojenie	21
3.5	Odkaz na domovskej obrazovke	21
3.6	Bezpečnosť progresívnych webových aplikácií	21
3.7	React	23
3.8	Docker	23
4	Implementácia	24
4.1	Server a databáza	24
4.2	Pacientová časť aplikácie	24
4.2.1	Prihlasovanie	24
4.2.2	Kalendár	25
4.2.3	Vyšetrenia	26
4.3	Neurológova časť aplikácie	29
4.3.1	Prihlasovanie	29
4.3.2	Menu s pacientami	29

4.3.3	Kalendár	30
4.3.4	Pridávanie nových pacientov a neurológov	31
4.3.5	Pridávanie vyšetrení	32
4.3.6	Pridávanie poznámok	33
4.4	Zabezpečenie aplikácie	33
4.5	Testovanie implementácie	34
4.6	Inštalácia systému	34
5	Legislatívna úprava	36
5.1	Ochrana osobných údajov - zákony	36
5.1.1	Osobný údaj	36
5.1.2	Spracovanie osobných údajov	36
5.1.3	Osobné údaje a spracovanie v aplikácii	37
5.2	Zabezpečený prenos dát	37
5.3	Ikony	37
	Záver	38
	Literatúra	39

1 Úvod

Parkinsonova choroba je neurodegeneratívne ochorenie centrálnej neurovej sústavy. Jedná sa o neliečiteľnú chorobu, ktorej symptómy je však v dnešnej dobe možné pomerne veľmi dobre potláčať. Potláčať túto chorobu je samozrejme možné najmä za pomoci lekárov, ktorí pacientovi stanovia vhodnú liečbu. Častokrát však pacienti nie sú schopní k lekárom dochádzať či už kvôli samotnej chorobe, finančnej situácii, či kvôli iným problémom. Pre pacientov je tak určite priaznivá možnosť, ak by ich lekári vedeli monitorovať na diaľku. Navyše toto riešenie umožní aj lekárom zrýchliť ich prácu a v prípade napríklad mimoriadnych opatrení v nemocniciach, ako napríklad za pandémie Covid-19 by takisto ostal lekár s pacientom v kontakte a vedel by stále sledovať jeho stav.

Takéto riešenie je možné dosiahnuť vďaka súčasným možnostiam informačných systémov či mobilným zariadením. V dnešnej dobe sú stále viac a viac populárne webové a mobilné aplikácie, ktoré dokážu vďaka svojim moderným technológiám častokrát poskytnúť lekárom možnosť, ako monitorovať pacienta. Častým spoločným problémom súčasných riešení je, že funguje väčšinou iba pre jeden operačný systém, a preto sú tieto riešenia nie až tak dostupné pre každého pacienta.

Cieľom práce je implementovať systém na monitorovanie Parkinsonovej choroby, ktorý pomôže pacientom vďaka tomu, že lekár bude môcť neustále sledovať ich stav, zadávať im do systému vyšetrenia a podobne.

V prvej kapitole je popísaná Parkinsonova choroba, jej symptómy, príčiny vzniku a podobne. Ďalej je v kapitole uvedený prehľad existujúcich riešení aplikácii na monitorovanie Parkinsonovej choroby, rovnako ako aj návrh a koncepty systému implementovaného v tejto práci. Druhá kapitola obsahuje informácie o tom, čo to je progresívna webová aplikácia a aj zmienku o frameworku React, ktorý je využitý pri implementácii. Tretia kapitola popisuje implementáciu systému, rovnako, ako aj testovanie a inštaláciu a štvrtá kapitola obsahuje legislatívnu úpravu.

2 Systém na monitorovanie priebehu Parkinsonovej choroby

Cielom diplomovej práce je implementovať aplikáciu podľa vopred stanoveného návrhu. Aplikácia sa zaoberá monitorovaním priebehu Parkinsonovej choroby u pacientov, čo dáva dobrú spätnú väzbu lekárom. Návrh celého systému je možné rozdeliť na 2 samostatné celky. Jedná sa o systém, v ktorom vystupujú pacienti a lekári, konkrétne neurológovia. Systém je navrhnutý tak, aby pacientova časť aplikácie bola schopná získavať dôležité informácie o jeho vopred stanovených vyšetreniach, ktoré mu neurológ do jeho aplikácie priradí. Informácie o priebehu daných vyšetrení potom neurológ uvidí a na základe výsledkov sa vie viac zamerať na pacientovu liečbu, pretože dostáva neustále informácie o tom, ako choroba u pacientov prebieha.

2.1 Parkinsonova choroba

Parkinsonova choroba je postupujúca porucha, ktorá napadá neurový systém. Symptómy sa začínajú objavovať pomaly. Prvým pozorovateľným príznakom môže byť kľudový tremor jednej ruky, ktoré je sotva všimnutelné. Kľudový tremor je bežný, ale Parkinsonova choroba môže spôsobiť aj stuhnutosť alebo spomalenie pohybu. V počiatočných štádiách parkinsonovej choroby môže tvar jedinca vykazovať malý alebo žiadny výraz. Ruky sa pri chôdzi nemusia hojdať. Reč môže byť až nezrozumiteľná. Symptómy parkinsonovej choroby sa zhoršujú časom. Hoci parkinsonovu chorobu nejde liečiť, tak s vhodnými liečebnými postupmi je možné symptómy potláčať [3].

2.1.1 Príčiny vzniku

Najvýraznejšie príznaky a symptómy Parkinsonovej choroby sa vyskytujú, keď sa nervové bunky v bazálnych gangliách, oblasti mozgu, ktorá kontroluje pohyb, poškodia a/alebo odumrú. Normálne tieto nervové bunky alebo neuróny produkujú dôležitú mozgovú chemickú látku známu ako dopamín. Keď neuróny odumierajú alebo sú poškodené, produkujú menej dopamínu, čo spôsobuje pohybové problémy spojené s chorobou. Vedci stále nevedia, čo spôsobuje smrť neurónov [1].

Niektoré prípady Parkinsonovej choroby sa zdajú byť dedičné a v niekoľkých prípadoch možno vysledovať špecifické genetické mutácie. Aj keď sa predpokladá, že genetika zohráva úlohu pri Parkinsonovej chorobe, vo väčšine prípadov sa zdá, že táto choroba sa nevyskytuje v rodinách. Mnohí vedci sa teraz domnievajú, že

Parkinsonova choroba je výsledkom kombinácie genetických a environmentálnych faktorov, ako je vystavenie toxínom [1].

2.1.2 Symptómy

Príznaky a symptómy Parkinsonovej choroby môžu byť u každého človeka iné. Skoré príznaky môžu byť mierne, alebo takmer nepovšimnuteľné. Tieto symptómy obvykle začínajú na jednej strane tela a zvyčajne na tejto konkrétnej strane zostávajú horšie a to aj po tom, čo príznaky začnú ovplyvňovať končatiny na oboch stranách. Príznaky a symptómy Parkinsonovej choroby môžu byť:

- **Chvenie:** Chvenie väčšinou začína v končatine, najčastejšie v ruke, alebo v prstoch.
- **Spomalený pohyb (bradykinéza):** V priebehu času môže Parkinsonova choroba spomaliť pohyb u jedinca, takže jednoduché úlohy začnú byť ťažšie a časovo náročnejšie. Kroky sa môžu skrátiť a môže byť aj ťažké vstať zo stoličky. Pri pokuse o chôdzu môže nastať ťahanie alebo šúchanie nohami.
- **Tuhé svaly (rigidita):** Svalová stuhnutosť sa môže vyskytnúť v ktorejkoľvek časti tela. Stuhnuté svaly môžu byť bolestivé a to následne obmedzuje aj rozsah pohybu.
- **Zhoršené držanie tela a rovnováha:** Časté sú problémy s rovnováhou, teda aj riziko spadnutia, ale aj zhrbený postoj.
- **Strata automatických pohybov:** Schopnosti vykonávať nevedomé pohyby, akými sú napríklad žmurkanie, úsmev či hojdanie rúk pri chôdzi, môže byť ťažšie vykonávať.
- **Zmeny reči a v písaní:** Môžu nastať rozdiely v reči oproti zvyčajným rečovým vzorcom, reč môže byť monotónna a pred rozprávaním môže byť zmenená na tichú, rýchlu, váhavu alebo vulgárnu. S písaním môže byť takisto problém a písmo sa môže zdať malé [3].

2.2 Existujúce riešenia aplikácii na monitorovanie Parkinsonovej choroby

V tejto sekcii sú popísané podobné existujúce riešenia, ktoré majú rovnaký, alebo podobný cieľ monitorovania Parkinsonovej choroby u pacientov. Riešenie existuje samozrejme viac, ako tie čo sú popísané v tejto sekcii. Väčšina však funguje veľmi podobne a obsahuje rovnaké nedostatky, ktoré sa snaží táto práca vylepšiť.

2.2.1 Apkinson

Apkinson je androidová aplikácia, ktorá má za cieľ kontinuálne vyhodnocovať rečové a pohybové symptómy pacientov s Parkinsonovou chorobou a tým poskytuje mechanizmus spätnej väzby o aktuálnom štádiu ochorenia. V aplikácii sú pacienti požiadaní, aby každý deň robili rôzne rečové či pohybové cvičenia, ktoré sú vybrané z banky cvičení. Banka obsahuje 35 cvičení. Aplikácia má jednoduchý dizajn s veľkými tlačítkami, pre jednoduchšie klikanie na displeji.

Rečové cvičenia zahŕňajú predĺženú fonáciu trvalých samohlások, diadochokinetické úlohy, niekoľko viet, ktoré musí pacient prečítať, alebo popis obrázkov, zjavujúcich sa na obrazovke smartfónu.

Na druhej strane, pohybové cvičenia sa zachytávajú pomocou inerciálnych senzorov smartfónu na vyhodnotenie symptómov na horných a dolných končatinách, ako sú okrem iného posturálny tremor, kinetický tremor, klepanie prstami, deficit chôdze. Na konci cvičenia Apkinson vyhodnotí výkon pacienta, pričom vedie register výsledkov z predchádzajúcich cvičení. Táto analýza umožní posúdiť progresiu ochorenia u jednotlivých pacientov [14].

2.2.2 StrivePD

StrivePD je dôveryhodná iOS aplikácia, zaoberajúca sa manažovaním Parkinsonovej choroby u pacientov ako aj tímov klinickej starostlivosti. Táto aplikácia funguje v spojení so smarthodinkami Apple Watch a zariadeniami na hĺbkovú mozgovú stimuláciu. Pacienti môžu sledovať a zdieľať svoje súhrnné skúsenosti s ochorením (symptómy, lieky, aktivity) s tímom starostlivosti mimo kliniky.

Aplikácia je zadarmo, avšak bohužiaľ rovnako ako napríklad **Apkinson** spomínaný predtým, tak aj **StrivePD** je dostupný iba na jeden druh zariadenia, spomínaný iOS.

2.2.3 Limitácie existujúcich riešení

Najčastejšou limitáciou existujúcich riešení, je že nemajú lokalizáciu v českom jazyku a väčšinou sú dostupné iba na jednom operačnom systéme, buď Android, alebo iOS. Problémom môžu byť často aj neintuitívne ovládanie, či zbytočne komplikovaný dizajn.

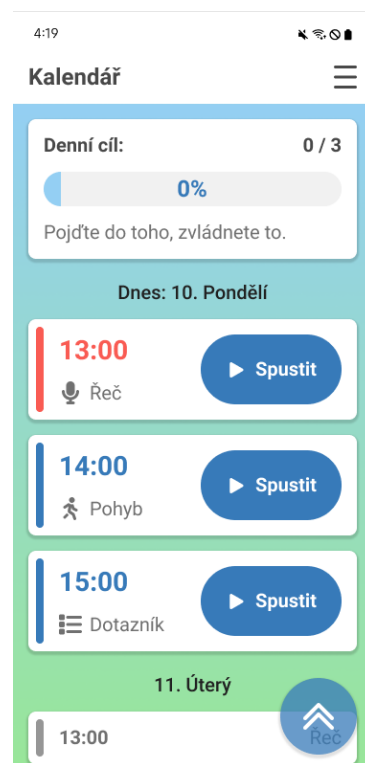
2.3 Návrh a koncepty systému

V existujúcich riešeniach chýbajú aplikácie, ktoré sú schopné fungovať "všade", na každom zariadení. To má za cieľ vyriešiť návrh, ktorý je pripravený tak, aby apli-

kácia bola postavená na technológii progresívnej webovej aplikácie. Pacientova časť aplikácie je navyše implementovaná v českom jazyku, keďže existujúce riešenia neposkytujú ani túto možnosť.

2.3.1 Pacient

Návrh pacientovej časti aplikácie je pomerne jednoduchý. Vo svojom profile bude pacient schopný vidieť iba špeciálne priradené ID, pohlavie a ročník. Z pohľadu pacienta je dôležité, aby bol jeho účet zabezpečený aby nedochádzalo ku stratám súkromných zdravotných informácií. Po prihlásení sa pacientovi zobrazí kalendár (obr. 2.1) s jeho predpísanými vyšetreniami, ktoré má splniť. Kalendár bude raz denne synchronizovaný so serverom, prípadne tak bude môcť spraviť pacient sám prostredníctvom tlačítka.



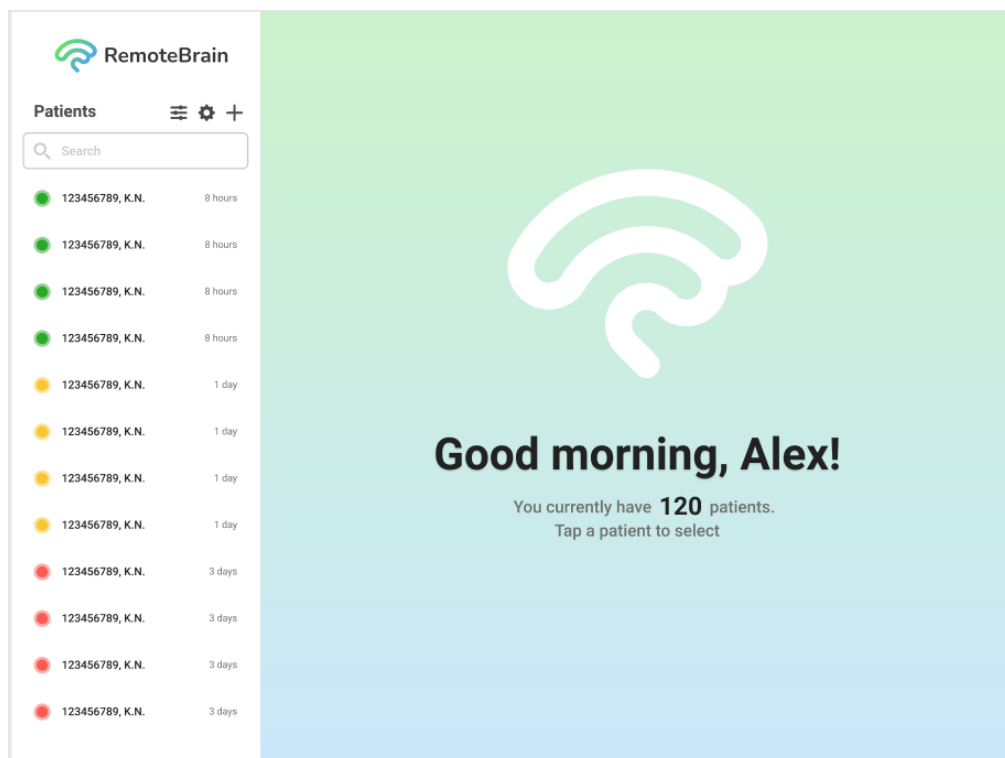
Obr. 2.1: Návrh pacientovho kalendára

Motiváciou pre pacienta bude znázornený ukazateľ s názvom denný cieľ, ktorý bude ukazovať, koľko percent vyšetrení splnil nachádzajúci sa v spomínanom kalendári na obr. 2.1. Ďalším motivujúcim a aj informatívnym prvkom pre pacienta budú notifikácie, ktoré pacienta budú informovať o tom, aby vykonal vyšetrenie. Akonáhle pacient zapne vyšetrenie, tak pred každým z vyšetrení sa mu zobrazí obrazovka s informáciami, ako by malo vyšetrenie správne prebiehať. Napríklad pri

rečovým vyšetrením sa zobrazí oznam, aby sa pacient presunul do tichého miesta, mal mikrofón v správnej vzdialenosti od úst, a podobne. Takéto informácie pre pacienta sú určite dôležité z pohľadu toho, že ak by vykonal vyšetrenie bez týchto doporučených informácií, tak pre neho tak povediac tento systém monitorovania stráca efekt. Po úspešnom ukončení vyšetrenia sa dáta vytvorené počas vyšetrenia odošlú na server, kde sa spracujú. Následne sú výsledky týchto jednotlivých vyšetrení uverejnené a neurológ má možnosť ich vidieť.

2.3.2 Neurológ

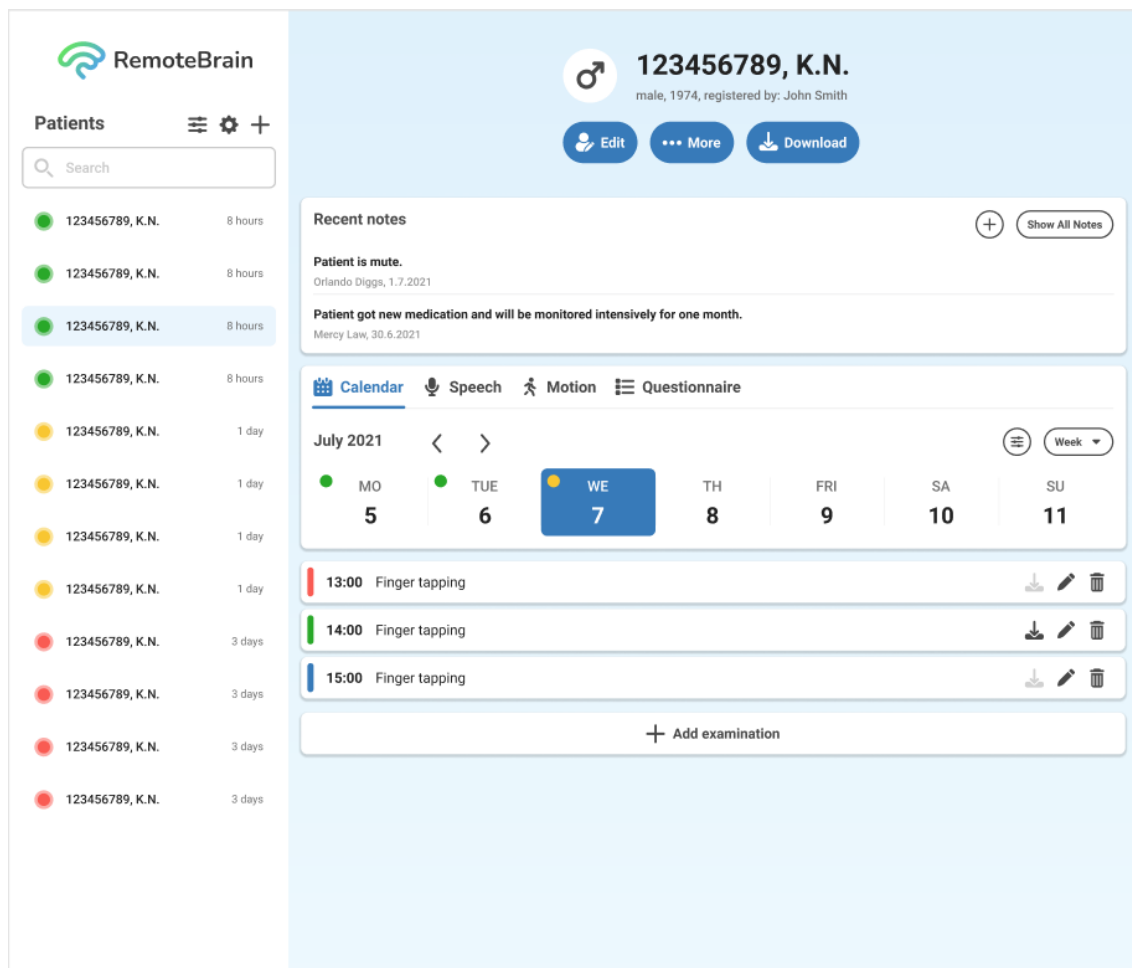
Z pohľadu neurológa je systém výhodnou voľbou pre monitovanie priebehu choroby pacientov. Neurolog v systéme vedie evidenciu jeho pacientov. Prihlasovacie údaje do aplikácie dodáva lekárovi administrátor. Administrátor nie je z pohľadu aplikácie až tak podstatný, keďže bude mať rovnaké práva ako neurolog a navyše bude iba schopný pridávať, mazať a editovať účty neurologov. Neurológ vidí účty svojich pacientov, ktoré môže pridávať, mazať a editovať. Pri jednotlivých pacientoch vidí výsledky ich vyšetrení, pričom ich bude môcť filtrovať. Samozrejme neurológ bude môcť pacientovi pridávať nové vyšetrenia. Takisto bude mať farebne rozdelených pacientov, podľa toho, či plnia dané vyšetrenia. Tieto časti návrhu sú dobre viditeľné na obr. 2.2a obr. 2.3



Obr. 2.2: Uvítacia stránka neurológa

2.3.3 Typy vyšetrení

Pacient bude v aplikácii absolvovať rôzne typy vyšetrení. Ako bolo spomínané tak tieto vyšetrenia mu zadáva neurolog, ktorý pacienta monitoruje. Dve hlavné skupiny vyšetrení budú dotazníky a aktívne úlohy. Systém bude navrhnutý tak, aby bolo možné pridať rôzne ďalšie dotazníky, prípadne aktívne úlohy.



Obr. 2.3: Kalendár neurológa

Dotazníky

Tento typ vyšetrenia obsahuje otázky na rôzne dôležité otázky ohľadom pacientovho stavu. Bude sa jednať o dotazníky **The Parkinson's Disease Questionnaire (PDQ-39)** [10], **The REM sleep behavior disorder screening questionnaire (RBDSQ)** [12] a **Parkinson's Disease Sleep Scale 2 (PDSS-2)** [11]. Dotazníkové vyšetrenie dáva neurológovi dobrú spätnú väzbu v tom, ako sa pacient cíti v poslednej dobe a či nemá nejaké problémy.

Aktívne úlohy

Aktívne úlohy sú interaktívnejšou voľbou vyšetrení, kde je zapojený priamo pacient, na základe fyzickej aktivity. Po ukončení týchto vyšetrení dostane pacient otázky, či počas vykonávania úlohy pozoroval dyskinízie, prípadne či užil lieky.

Reč a hlas

Táto úloha bude mať viacero podúloh. Každá úloha začína inštrukciami, ako má pacient vyšetrenie absolvovať. Medzi jednotlivé podúlohy budú zaradené rozprávanie na voľnú tému, predĺžená samohláska a, alebo diadochokinetická úloha.

Klikanie prstami

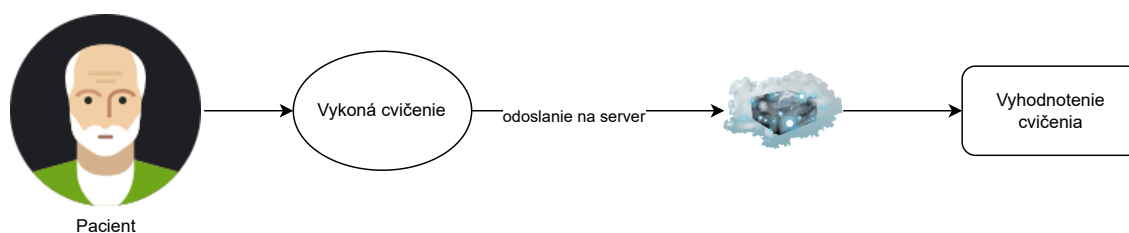
Jedná sa o monitorovanie motorických funkcií a rýchlosti prsta, tak, že pacient kliká dvomi prstami na dotykový displej vo vymedzenej časti obrazovky. Pri tomto cvičení sú zbierané dáta o časovaní a pozicioní jednotlivých dotykov s obrazovkou [9].

2.4 Prípady použitia aplikácie (Use cases)

Ako bolo spomínané v návrhu, tak v aplikácii sa nachádzajú dvaja hlavní aktéri. Jedná sa o pacientov a neuroológov. Z pohľadu prípadov použitia je dôležité začať najprv pacientom, pretože bez jeho aktivity by neuroológ s aplikáciou nemal čo robiť.

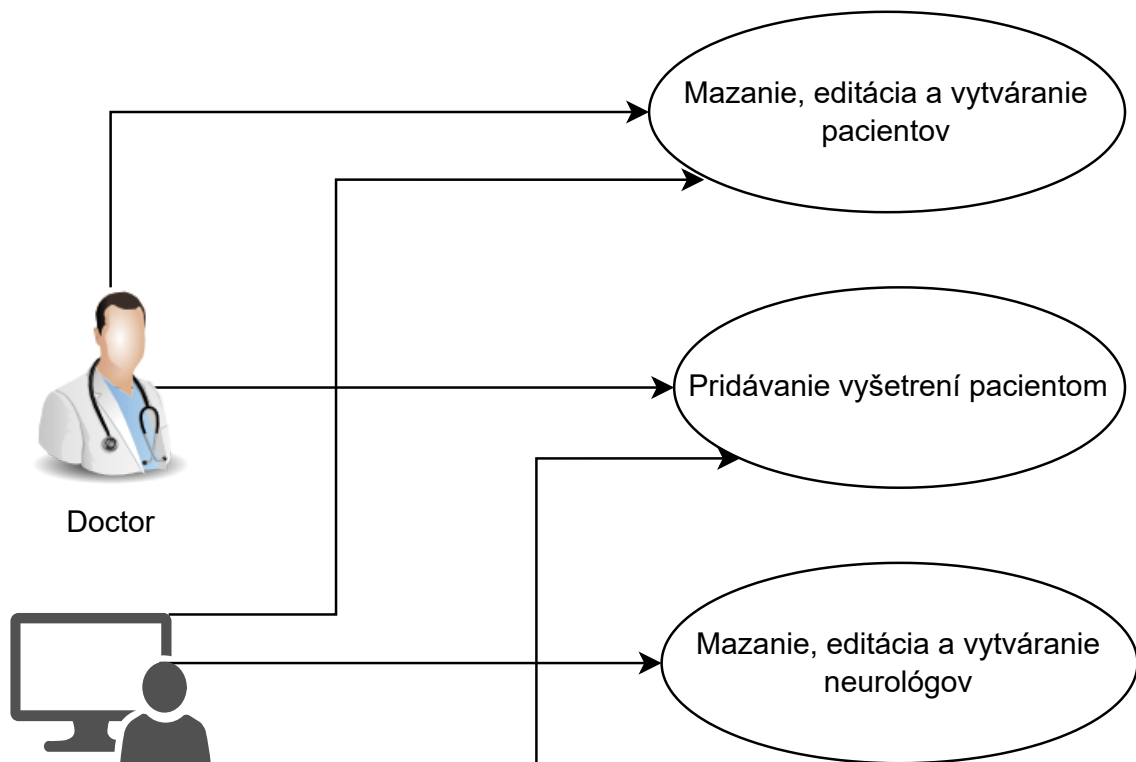
2.4.1 Pacient

Pacient je z pohľadu aplikácie teda dôležitým faktorom, keďže jeho výsledky doktor má možnosť vidieť a na základe nich prispôbovať liečbu. Tou najdôležitejšou aktivitou z pohľadu efektívnosti aplikácie je, aby pacient vykonával vyšetrenia.



Obr. 2.4: Princíp fungovania pacientovej časti aplikácie

Vykonávanie vyšetrení je v podstate najhlavnejší prípad užitia, bez ktorého by aplikácia nemohla fungovať. Pacient preto potrebuje potrebnú motiváciu v podaní



Obr. 2.5: Use case neurolog a admin

spomínaného denného cieľa, ktorý môže byť vidieť na obr. 2.1, rovnako ako aj notifikácie o tom, že by mal absolvovať vyšetrenie. Po absolvovaní vyšetrenia môže pacient spokojne zavrieť aplikáciu, výsledky sa totiž ďalej spracujú na serveri a následne ich môže pozorovať neurológ.

2.4.2 Neurológ

Úloha neurológa v tejto aplikácii je monitorovať svojich pacientov a sledovať tak priebeh parkinsonovej choroby u nich. Potrebným faktorom, aby toho bol schopný je to aby pacienti vykonávali vyšetrenia v aplikácii. Po celom procese ktorý bol popisovaný na obr. 2.4, princípu fungovania pacientovej časti aplikácie, si môže neurológ overovať výsledky vyšetrení u pacientov, kontrolovať, či pacienti na svojich vyšetreniach aktívne pracujú, alebo editovať, mazať a pridávať vyšetrenia. Admin má všetky práva ako neurológ, navyše dokáže ešte vytvárať, mazať a editovať neurológov v systéme. Prípady použitia neurológa a admina je možné vidieť na obr. 2.5.

3 Progresívna webová aplikácia (PWA)

V dnešnej dobe existuje viacero riešení, akými vytvoriť webovú aplikáciu, respektíve natívnu aplikáciu. Existuje však riešenie v podobe spomínanej PWA, alebo napríklad natívne riešenia pomocou frameworku akými sú react native, flutter či iné. Pre účely tejto aplikácie je však veľmi vhodné zvoliť PWA. Pre vytvorenie PWA aplikácie je potrebné použiť **HTML**, **CSS** a **JavaScript** [6] [7].

Progresívna webová aplikácia, alebo v skratke **PWA** je aplikácia, ktorá by mala byť schopná fungovať takmer všade, to znamená skoro na každom zariadení. Nejedná sa o žiadny nový framework či jazyk, ale naopak miesto toho PWA je súbor **stratégií, techník a aplikačných programovateľných rozhraní (API)**, ktoré vývojárom dávajú možnosť poskytnúť užívateľom zážitok ako pri bežnom používaní mobilných aplikácií. Na vytvorenie progresívnej webovej aplikácie sú potrebné znalosti **HTML**, **CSS** a **JavaScriptu**. Vytvorenie progresívnej webovej aplikácie je možné aj za pomoci využívaných JavaScriptových frameworkov, akými sú napríklad **React**, **Vue**, **Angular** a podobne [6].

Sú vytvárané tak, aby boli:

- **Rýchle:** Často renderujú nejaký obsah na užívateľovom zariadení v priebehu pár sekúnd.
- **Spôľahlivé:** Schopné pracovať pri horšom internetovom pripojení a na starších zariadeniach.
- **Aktívne zapoujú užívateľov:** Užívateľom sa dá aplikácia jednoducho nainštalovať do ich zariadení a vďaka povoleným notifikáciám je užívateľ informovaný o tom, čo sa v aplikácii odohráva a to aj pokiaľ nemá aplikáciu zapnutú [6].

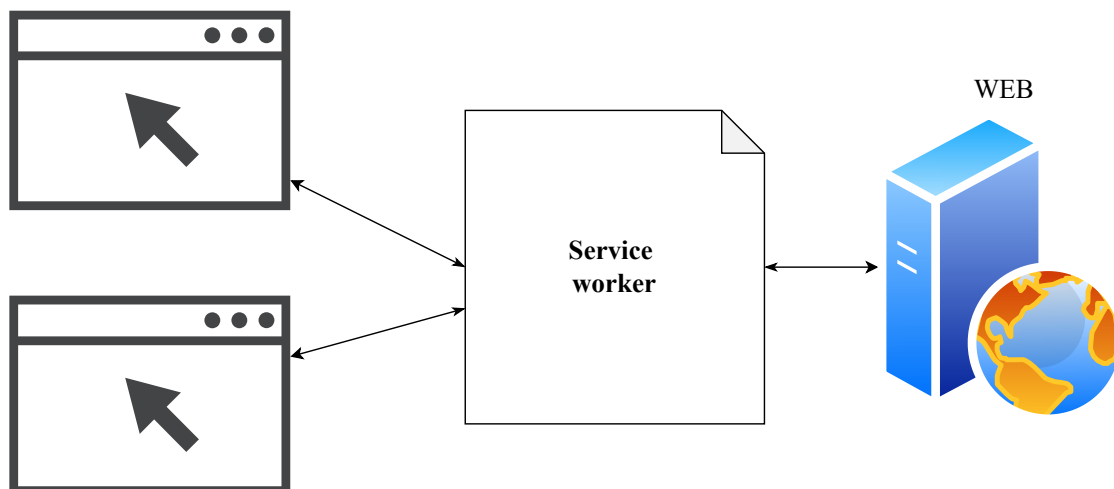
3.1 Service worker

Dôležitou súčasťou **PWA** aplikácii je takzvaný **Service worker**. Service worker predstavujú posun v tom, ako sa pozeráť na vývoj webov. V minulosti, bežali aplikácie na serveri alebo v okne prehliadača, ale service worker predstavuje ďalšiu vrstvu. Service worker je veľmi užitočným a prináša do aplikácií radu benefitov, ako napríklad

- ukladanie prostriedkov do vyrovnávacej pamäte, ako sú obrázky, skripty alebo štýly,
- ukladanie celej stránky do vyrovnávacej pamäte,
- synchronizácia stránky potom, ako sa znovu obnoví internetové spojenie,
- push notifikácie [6].

Service worker je popisovaný, ako skript, ktorý možno zaregistrovať, aby dokázal ovládať jednu alebo viacero stránok na konkrétnom webe. Po inštalácii je service

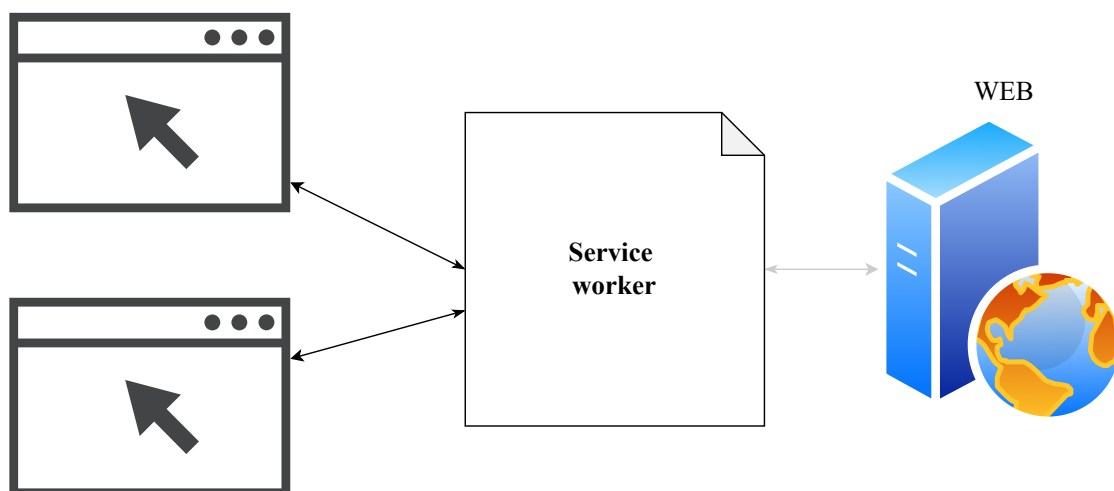
Karty prehliadača



Obr. 3.1: Karty prehliadača, service worker a web

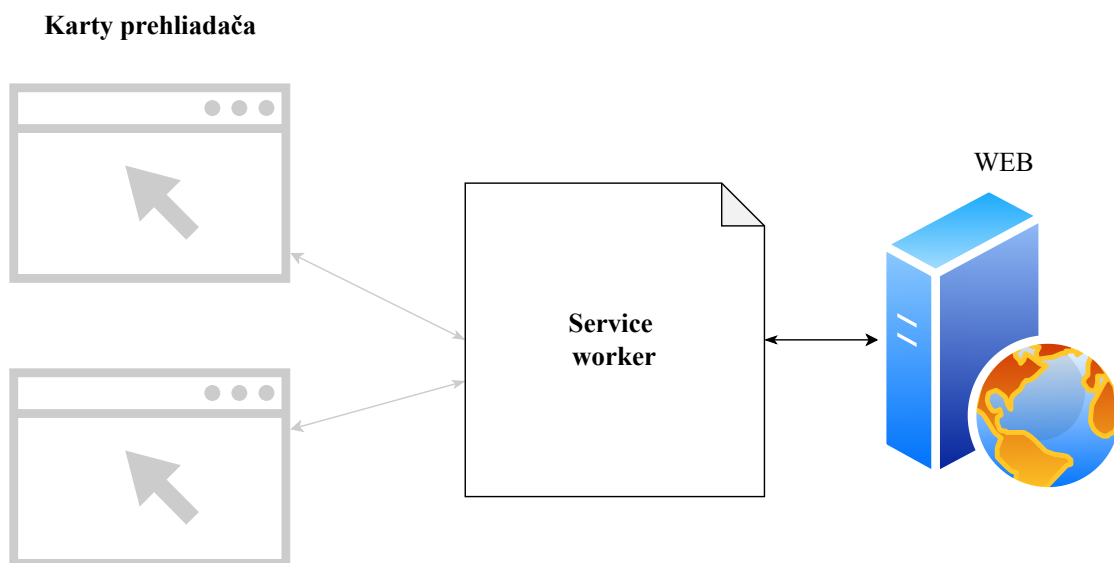
worker nasadený mimo akéhokoľvek okna prehliadača, či karty. Vďaka tomu dokáže počúvať a reagovať na udalosti zo všetkých stránok, ktoré má pod kontrolou. Udalosti, ako sú požiadavky na súbory z webu, môžu byť zachytené, upravené, odovzdané a vrátené na stránku, ako je možné vidieť na obr. 3.1 To znamená, že medzi stránkou a webom je vrstva, na ktorú môžu reagovať požiadavky nezávisle od sieťového pripojenia. Vrstva, ktorá funguje, aj keď je používateľ je offline. Táto vrstva dokáže detekovať offline stav alebo pomalé reakcie servera a namiesto toho vrátiť obsah uložený vo vyrovnávacej pamäti (obr. 3.2). To znamená, že ak užívateľ zavrie

Karty prehliadača



Obr. 3.2: Stránky komunikujúce so service workerom, pokiaľ je užívateľ offline

všetky karty spustené v prehliadači, existuje stále vrstva, ktorá môže komunikovať so serverom (obr. 3.3). Môže prijímať a zobrazovať push notifikácie, alebo sa ubezpečiť, že nejaké akcie vykonané užívateľom doputujú na server [7].



Obr. 3.3: Service worker komunikujúci so serverom po uzavretí stránky užívateľom

Service worker je teda skvelým pomocníkom vo chvíľach, kedy internetové spojenie nefunguje podľa našich predstáv. Dôležitým faktorom pri jeho použití je, že funguje iba na HTTPS protokole. Service worker potrebuje zabezpečené pripojenie z toho dôvodu, pretože jeho funkcionality nápadite pripomína **man in the middle** útok. Je to tým, že zachytáva jednotlivé sieťové požiadavky [6].

3.2 Web app manifest

Manifest webovej aplikácie je funkcia PWA, ktorá je zameraná na mobilné zariadenia. Vďaka tejto technológii je možné zadať podrobnosti o aplikácii, ktoré pomôžu zariadeniam poskytnúť používateľom čo možno najlepší užívateľský zážitok. Web app manifest je súbor typu JSON, ktorý obsahuje viaceré parametre, akými sú napríklad názov aplikácie, krátky názov aplikácie, orientáciu, začiatočnú url, ikony, popis aplikácie a podobne. Po úspešnom vytvorení manifestu, v takom rozmedzí aby spĺňal požiadavky PWA je možné aplikáciu inštalovať na rôznych zariadeniach [6].

3.3 Push notifikácie

Výhoda push notifikácií je, že progresívna webová aplikácia vie poslať používateľom notifikácie, dokonca aj po pár dňoch, čo naposledy navštívil stránku. Takéto

upozornenia majú úplne natívny vzhľad a sú na nerozoznanie od natívnych upozornení. Ďalším prínosom týchto notifikácií je určite aj to, že dokážu opätovne zapojiť užívateľa do aplikácie [7].

3.4 Dostupnosť bez ohľadu na pripojenie

Ako už bolo spomínané v sekcii 3.1, tak PWA aplikácie fungujú aj bez pripojenia na internet, respektíve nie sú na ňom závislé. Používateli sa zaregistrujú service worker, ktorý detekuje a reaguje na zmeny. Užívateľ môže byť teda na miestach, kde internet nie je dostupný, prípadne nefunguje úplne spoľahlivo, ale aj tak vie, že jeho udalosti, ktoré vykonal, napríklad zasielanie správ, komentovanie príspevkov, v našom prípade odoslanie vyšetrenia, budú skompletované v momente akonáhle bude internetové spojenie spoľahlivé a to aj v prípade, že aplikáciu uzatvorí. Tým pádom je možné tvrdiť, že sú spoľahlivejšie a u užívateľov budujú dôveru v tom, že sa ich akcia vykoná, čo bola doteraz výhoda iba natívnych aplikácií.

Užívateľ vie, že môže otvoriť napríklad aplikáciu akou je WhatsApp, napísať správu a následne mobilný telefón uložiť na miesto odkiaľ ho zobral, bez obáv o aktuálny stav pripojenia. Webové aplikácie nemali doteraz takúto úroveň dôvery, čo je jeden z dôvodov, prečo užívatelia často volia pre svoje používanie natívne aplikácie [7].

3.5 Odkaz na domovskej obrazovke

V momente, keď používateľ prejaví záujem o progresívnu webovú aplikáciu, tak mu prehliadač automaticky navrhne, že si môže pridať odkaz na domovskú stránku [7]. Tým pádom vidíme rozdiely v procese inštalácie, kde natívne aplikácie sú riadené takýmto prístupom: Prístup k referenčnému obchodu (Google Play, alebo App Store), vyhľadanie a následne nainštalovanie aplikácii, nastavenie povolení aplikácie a následné otvorenie a spustenie. Na druhej strane PWA inštalácia zahŕňa tieto činnosti: návšteva stránky, pridanie na domovskú obrazovku (voliteľná možnosť), otvorenie a následné použitie aplikácie [5].

3.6 Bezpečnosť progresívnych webových aplikácií

Pri pohľade na bezpečnosť progresívnych webových aplikácií je potrebné sa zamyslieť nad tým, že sa jedná stále pomerne o novú technológiu. Nové technológie ponúkajú potencionálnym útočníkom nové možnosti útokov. Pravdou však je, že aj keď sa jedná pomerne o novú technológiu, tak sa v podstate jedná iba o webovú aplikáciu

obohatenú o nejaké zaujímavé funkcie navyše. Avšak práve v dôsledku toho môžu byť **progresívne webové aplikácie potenciálne zraniteľné voči všetkým známym formám webových útokov.**

Dôležitejšie a zaujímavejšie z pohľadu bezpečnosti je pozrieť sa na Web app manifest a Service worker. Čo Web app manifest a Service worker je bolo popísané vyššie.

Množstvo škôd, ktoré je možné spôsobiť prostredníctvom **manifestu** je **obmedzené**. To však neznamená, že **bezpečnosť** by v tomto prípade nemala byť braná vážne. Kybernetickí útočníci napríklad radi využívajú cross-site scripting útoky, pri ktorých sa snažia do aplikácie vložiť škodlivý skript. Pokiaľ ide o manifest, tak prehliadače používajú prvú inštanciu manifestu, bez ohľadu na to, koľko manifestov v kóde je. Týmto pádom útočníci nevedia prepísať manifest. V prípade že nie je manifest nakonfigurovaný, tak dochádza k problému. V tejto chvíli by útočník mohol prepojiť aplikáciu so škodlivým manifestom. Takéto poškodenie je však obmedzené na estetiku, akými sú farby, názov, ikona aplikácie a podobne. Webové prehliadače v dnešnej dobe však obmedzujú domény, z ktorých je možné načítať manifest, takže tento krok takisto ďalej znižuje množstvo potenciálnych škôd, ktoré môžu byť prostredníctvom manifestu spôsobené.

Naopak v prípade **Service workeru** sa jedná o oblasť, ktorá je atraktívna na útok, pretože útočníci môžu zachytiť spojenia, alebo poskytnúť používateľom upravené odpovede. Je dôležité, aby kybernetickí útočníci nemohli upravovať vašich servisných pracovníkov. Ak kybernetický útočník môže prevziať kontrolu nad servisným pracovníkom, môže neustále útočiť na prichádzajúce a odchádzajúce informácie. Tento typ kybernetického útoku je známy ako muž uprostred. Škodlivý servisný pracovník môže mať vážne následky pre aplikáciu a používateľov. Kybernetický útočník, ktorý používa škodlivého servisného pracovníka, by mohol aktívne monitorovať a kontrolovať všetku komunikáciu medzi backendom aplikácie a používateľom frontendu. Útočník by mohol ľahko posielat používateľom phishingové správy, ktoré kompromitujú ich údaje a osobné informácie. Servisní pracovníci nemajú prístup k DOM alebo súborom cookie, aby obmedzili množstvo škôd, ktoré môže spôsobiť zlomyseľný servisný pracovník. Aplikácia by však mala využívať a podporovať rozhranie postMessage na komunikáciu medzi servisnými pracovníkmi a stránkami, ktoré ovládajú. Takýmto spôsobom je možné minimalizovať potenciálne škody spôsobené škodlivým servisným pracovníkom a zabezpečiť, že nebudú mať prístup k DOM.

Z väčšej časti sú PWA bezpečné, pretože sa riadia protokolmi HTTPS rovnako ako akékoľvek iné webové stránky alebo webové aplikácie. Navyše, keďže sú prístupné cez webové prehliadače, PWA využívajú všetky moderné bezpečnostné funkcie zabudované aj do webových prehliadačov. Samozrejme, pri PWA je potrebné brať do

úvahy všetky bežné webové útoky, ktoré sa používajú[2].

3.7 React

Framework react je v dnešnej dobe veľmi obľúbený a používaný na vytváranie webových aplikácií. Je vyvíjaný spoločnosťou Meta a komunitou ľudí. Vhodnosť jeho použitia sa dá vyjadriť tým, že poskytuje riešenia, ktoré dokážu zrýchliť vývoj aplikácie, fungujú bez problémov, ak sú správne použité a samozrejme aj uľahčujú vývoj webovej aplikácie. Reč je napríklad o funkciách reactu akými sú rôzne "hooky", ktoré dokážu pracovať so stavom premennej v aplikácii veľmi jednoducho.

React je:

- **Deklaratívny:** Zľahčuje vytváranie užívateľských rozhraní. Veľmi dobre pracuje so stavmi a aktualizuje efektívne komponenty, keď sa dáta menia
- **Komponentný:** Využíva zapúzdrené komponenty, ktoré spravujú svoj vlastný stav, potom sa z nich zostaví komplexné užívateľské rozhranie. Vzhľadom k tomu, že logika komponent je písaná v JavaScripte miesto šablón, tak jednoducho predáva data prostredníctvom aplikácie a udržuje stav mimo DOM [8].

3.8 Docker

Docker je otvorená platforma pre vývoj, odosielanie a spúšťanie aplikácií. Docker umožňuje užívateľom oddeliť aplikácie od infraštruktúry, aby sa tak mohol softvér rýchlejšie a jednoducho dodávať. S Dockerom je možné spravovať infraštruktúru rovnakým spôsobom, akým sú spravované aplikácie. Využitím metodológií Dockeru na rýchle odosielanie, testovanie a nasadzovanie kódu je možné výrazným spôsobom skrátiť oneskorenie medzi napísaním kódu a jeho spustením v produkcii.

4 Implementácia

Implementáciu je možné rovnako ako návrh rozdeliť na dve hlavné časti a to pacientovú časť aplikácie a neurológova časť. Medzi týmito časťami je postavená spoločná databáza a server. Obe aplikácie majú frontendovú časť aplikácie implementovanú v React a spoločnú backendovú v Node.js a Express frameworku. Spoločná databáza je vytvorená v PostgreSQL.

4.1 Server a databáza

Server spoločne s databázou sú kľúčovým bodom celého systému. Samotný server slúži na ukladanie jednotlivých výsledkov vyšetrení a ďalšie potrebné spracovanie týchto dát.

Pre potreby spojenia pacientovej a neurológovej časti aplikácie je potrebná databáza, do ktorej sú ukladané dôležité informácie z aplikácii. Z pohľadu návrhu je vhodné uvažovať databázu s tabuľkami pacient, úloha, poznámky a neurológ. Správne navrhnutá databáza je pre potreby aplikácie veľmi dôležitá.

4.2 Pacientová časť aplikácie

Implementácia pacientovej časti aplikácie je ako už bolo spomínané postavená na technológiach React, Node.js, Express. Dôležitou súčasťou pacientovej časti je prihlásenie. Pacient sa do aplikácie neprihlasuje svojím menom, ale prideleným špeciálnym identifikačným kódom a heslom. Prihlasovanie funguje za pomoci JSON Web Token technológie. Pre komunikáciu medzi Backendom a Frontendom je dôležité JavaScriptové Fetch API, ktoré spracováva jednotlivé požiadavky na Backend a následne na tieto požiadavky vracia odpovede. Na štýlovanie komponent je využité MaterialUI. Ikony v aplikácii sú čerpané z Iconify, čo je open source framework s ikonami.

4.2.1 Prihlasovanie

Z pohľadu pacienta je dôležité zabezpečiť aplikáciu, čo najlepším možným spôsobom, aby nedochádzalo k únikom jeho osobných informácií. Pacientove prihlasovanie je implementované za pomoci JSON Web Tokenu.

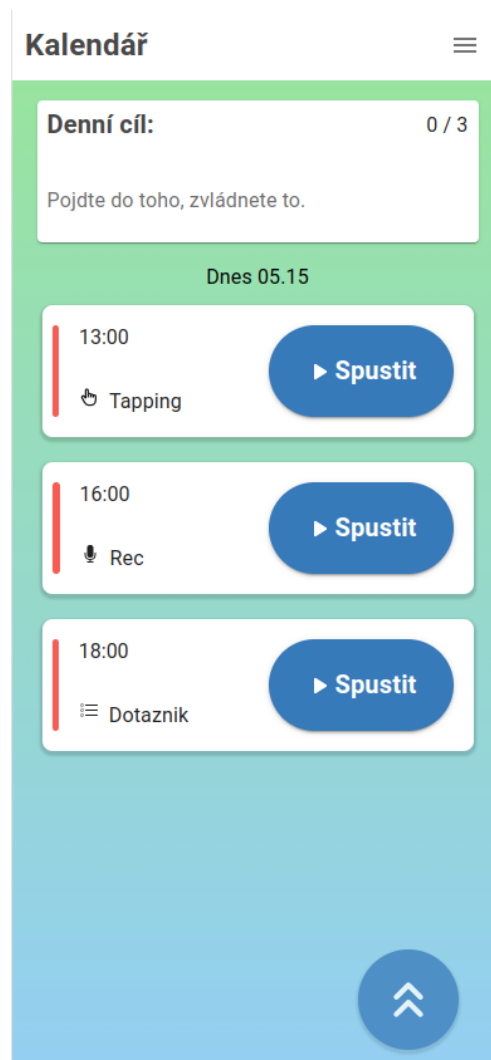
JSON Web Token

JSON Web Token (JWT) je otvorený štandard podľa RFC 7519, ktorý definuje kompaktný a samostatný spôsob zabezpečeného prenosu informácií medzi stranami.

To je realizovateľné za pomoci JSON objektu. Tieto informácie môžu byť overené a dôveryhodné, pretože sú digitálne podpísané. JWT môže byť podpísaný pomocou tajného (HMAC algoritmus), alebo za pomoci páru verejný/ súkromný kľúč pomocou RSA, alebo ECDSA.sa [15].

4.2.2 Kalendár

Implementácia kalendáru (obr. 4.1) je postavená tak, že sa posiela GET požiadavka na server, na získanie všetkých pacientových úloh. Po získaní týchto úloh z databáze, sú následne zobrazené jednotlivé úlohy a je možné ich vypracovávať. V kalendári je rovnako implementovaný aj denný cieľ v percentách, ktoré sú vypočítané z toho, koľko cvičení pacient v daný deň vykonal. Dôležitou súčasťou kalendáru, ale aj celej aplikácie z pohľadu implementácie je Fetch API.



Obr. 4.1: Implementovaný kalendár s tromi nedokončenými vyšetreniami

Fetch API

Fetch API poskytuje rozhranie na načítanie zdrojov (aj cez sieť). Nové API poskytuje výkonnejšiu a flexibilnejšiu sadu funkcií ako napríklad XMLHttpRequest. Fetch poskytuje všeobecnú definíciu objektov Request a Response (a ďalších vecí súvisiacich so sieťovými požiadavkami). To im umožní použiť ich všade tam, kde budú v budúcnosti potrebné, či už je to pre servisných pracovníkov, Cache API a iné podobné veci, ktoré spracúvajú alebo upravujú požiadavky a odpovede.

Metóda fetch() má jeden povinný argument, cestu k zdroju, ktorý je potrebné načítať. Vracia promise, ktorý sa vyhodnotí ako odpoveď na túto požiadavku – hneď ako server odpovie hlavičkami. Dochádza k tomu aj vtedy, keď je odpoveďou servera chybový stav HTTP [16].

4.2.3 Vyšetrenia

Jednotlivé vyšetrenia majú dizajnovú štruktúru implementovanú univerzálne. Je to tak kvôli tomu, že v budúcnosti bude jednoduché pridať do aplikácie nový typ vyšetrenia a tým pádom sa bude vedieť aplikácia prispôbiť prípadným požiadavkam neuroológov.

Dotazníkové vyšetrenia

Implementácia dotazníkového vyšetrenia (obr. 4.2) zahŕňa vytvorenie jednotlivých dotazníkov, ako súboru otázok uložených v súbore typu JSON. Na základe odpovedí na otázky, ktoré pacient zvolí, sa súbor postupne prepisuje a po skončení dotazníku sa uloží na server, rovnako vo formáte JSON, akurát už navyše aj s odpoveďami na jednotlivé otázky.

Z pohľadu Frontendu, je potrebné prihliadať na návrh tak, aby boli dotazníky pre pacientov jednoznačné, teda napríklad nepovoliť viac správnych odpovedí, ak je možná iba jedna a podobne.

V prípade, že bude potrebné do aplikácie pridať nejaké nové dotazníky, sa jedná o jednoduchý proces. Na vytvorenie nového dotazníka je potrebné iba vytvoriť nový JSON súbor, podľa štruktúry už vytvoreného dotazníka. Ten má štruktúru typu otázka a možné odpovede.

Pohybové vyšetrenia

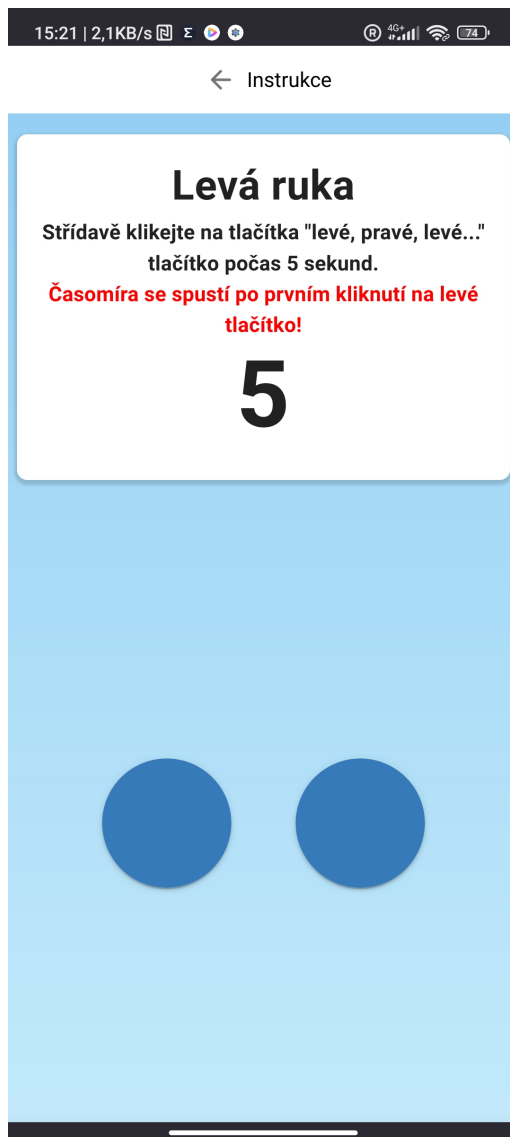
Zatiaľ implementované pohybové vyšetrenie je iba klikanie prstami (obr. 4.3), ktoré je pomerne jednoduché na implementáciu. Jedná sa o vyšetrenie, ktoré na realizáciu potrebuje dve tlačítka, na ktorých sa monitoruje, kedy na ne pacient klikol v rozmedzí piatich sekúnd. Tým pádom je implementovaný časovač, ktorý monitoruje



Obr. 4.2: Ukážka dotazníkového vyšetrenia

jednotlivé kliky na konkrétne tlačítka umiestnené na stránke. Po každom kliknutí sa počítadlo klikov zvýši o jedna a po dokončení tohoto vyšetrenia je následne výsledok uložený do JSON súboru. Tento súbor je následne poslaný na server, odkiaľ je výsledok vyšetrenia ďalej poskytovaný neurológovi.

V budúcnosti bude do aplikácie pomerne jednoduché implementovať ďalšie pohybové vyšetrenia a to aj vďaka navrhnutému riešeniu implementácie, kde sa v podstate iba vytvorí nová komponenta s potrebným dizajnom a funkčnosťou, ktorá sa následne napojí do aplikácie. Samozrejmosťou je, že sa nejedná o tak jednoduchý spôsob ako pri pridávaní dotazníka, keďže dotazníkové vyšetrenie je v podstate vždy iba o tom dostať do aplikácie nejaké tlačítka s odpoveďami, kdežto pri pohybových vyšetreniach je potrebné implementovať aj inú funkcionality daného vyšetrenia.



Obr. 4.3: Ukážka vyšetrenia Finger Tapping

Hlasové vyšetrenia

Pre vyšetrenia hlasom, bolo potrebné implementovať nahrávač zvuku, ktorý si samozrejme vypýta od pacienta povolenie na využitie jeho mikrofónu. Rovnako ako pri iných cvičeniach sú tieto cvičenia ukladané na server. Po tomto vyšetrení je nahrávka uložená vo formáte WAV. Jedná sa o bezstrátový formát, ktorý je vhodné použiť, aby sa zachovala kvalita nahrávky.

4.3 Neurológova časť aplikácie

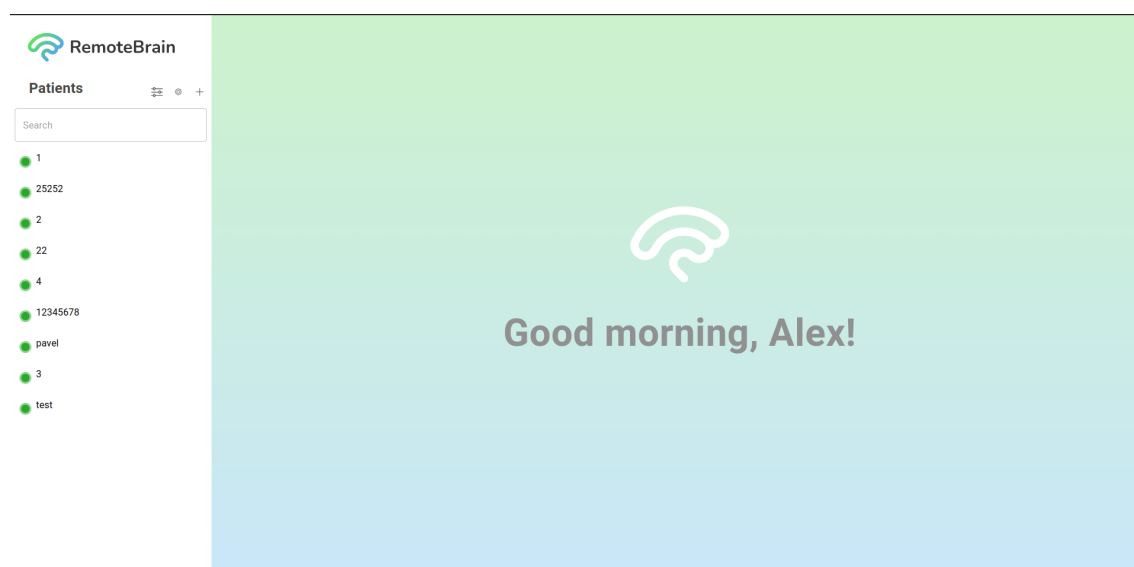
Implementácia tejto časti aplikácie zahŕňa viaceré prvky, akými sú kalendár, pridávanie nových pacientov, pridávanie vyšetrení, výsledky jednotlivých vyšetrení u pacientov a podobne. Každá z týchto častí je samozrejme priamo spojená s pacientovou časťou aplikácie, pretože bez implementácie tejto časti by nemohla v reálnej aplikácii fungovať ani druhá časť. Jednotlivé funkčné prvky tejto časti sú preto podrobnejšie opísané v podsekciiach. Rovnakým spôsobom sa do aplikácie prihlasuje aj admin.

4.3.1 Prihlasovanie

Neurológ sa do aplikácie prihlasuje podobne ako pacient za pomoci JSON Web Tokenu. Na prihlasovanie je využitý rovnaký návrh ako pri prihlasovacej stránke pacientov. Následne po prihlásení už neurológ vidí uvítaciu stránku, ktorá je zachytená na obr. 2.2. Na tejto stránke je z databázy vytiahnutá informácia, koľko pacientov aktuálne neurológ má a na ľavej časti obrazovky menu s pacientami, ktoré je z pohľadu implementácie dôležitejšie.

4.3.2 Menu s pacientami

V ľavej časti aplikácie pre neurológov je implementované menu s pacientami, ako je možné vidieť na obr. 4.4. V menu neurológ okamžite vidí, v akom stave sa pacientove vyšetrenia momentálne nachádzajú, či ich vykonáva, alebo naopak či náhodou nemá nejaké vyšetrenia zameškané.

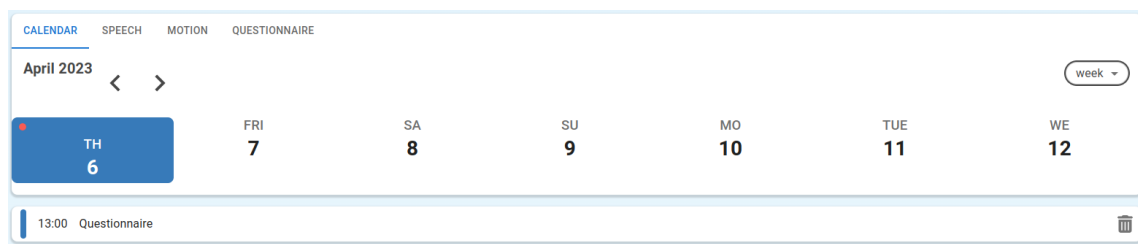


Obr. 4.4: Úvodná strana s ľavým menu obsahujúcim zoznam pacientov.

Po kliknutí na pacientove ID v menu sa zobrazí neurológovi kalendár s pacientovými vyšetreniami a prípadne nejaké poznámky. Z kalendára sa dá potom prístupovať aj ku iným dôležitým prvkom aplikácie, ktorými sú zobrazovania jednotlivých výsledkov vyšetrení, ktoré pacient vykonal.

4.3.3 Kalendár

Takisto ako pri pacientovej časti aplikácie hrá kľúčovú úlohu kalendár, tak sa dá povedať, že aj u neurológov bude najčastejšie používanou funkciou aplikácie práve kalendár, samozrejme v spojení s inými funkcionalitami popísanými nižšie. Samotný kalendár (obr. 4.5) totiž slúži na pridávanie nových vyšetrení a prezeranie výsledkov bývalých už splnených prípadne nesplnených vyšetrení. Jeho implementácia sa riadi návrhom a je teda implementované ako mesačné zobrazenie kalendáru, tak aj týždňové. Kalendár je implementovaný v komponente Calendar, ktorá riadi celé chovanie kalendáru z hľadiska frontendu. Pôvodne bol plán využiť na komponentu kalendáru nejakú vopred napísanú knižnicu, ale kvôli dizajnu, ktorý je špecifický sa rozhodlo, že sa kalendár napíše ako vlastná komponenta v aplikácii.



Obr. 4.5: Kalendár u neurológa

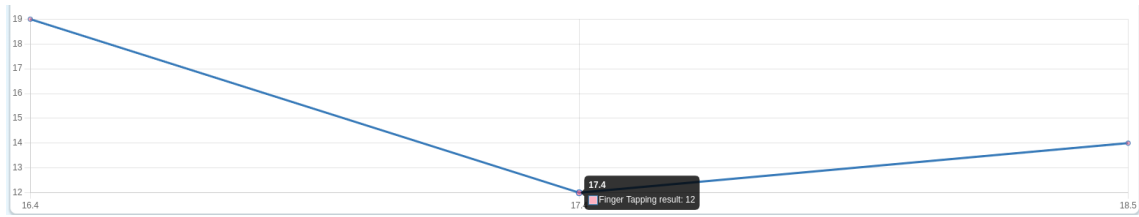
Na kalendári úzko závisí aj pridávanie nových vyšetrení, či prezeranie výsledkov jednotlivých vyšetrení. Jednotlivé vyšetrenia je možné zobrazit pri každom dni a je možné vidieť výsledky v type grafov, odpovedí na otázky, alebo prostredníctvom prehrávača zvuku (obr. 4.6).



Obr. 4.6: Záznam vyšetrenia reči a hlasu

Komponenta postavená na zobrazovanie grafov využíva dostupnú knižnicu Chart.js na vytvorenie grafu. Vďaka tejto knižnici je možné graf zobrazit tak aby spĺňal navrhované riešenie (obr. 4.7). Samozrejmosťou je, že jednotlivé údaje o grafoch sú

dostupné pri vyšetreniach pohybového typu. Jednotlivé údaje o týchto vyšetreniach sú zo servera získavané pomocou GET požiadavky, ktorá zašle údaje o tom ako vyšetrenie prebehlo.



Obr. 4.7: Vyhodnotenie vyšetrenia klikania prstami u pacienta

Dotazníkové vyšetrenie je zobrazené ako otázka- odpoveď vo webovom rozhraní, ako je možné vidieť na obr. 4.8.

Questionnaire 1 - 4.15-4.46.json

1. Bylo pro Vás obtížné se zabývat ve svém volném čase činnostmi, které byste rád/a dělal/a?-[vždy]
2. Bylo pro Vás obtížné se starat o domácnost, např. drobné opravy, úklid a vaření?-[vždy]
3. Bylo pro Vás obtížné nosit tašky s nákupem?-[vždy]
4. Dělal/a Vám potíže ujít 1 kilometr?-[vždy]
5. Dělal/a Vám potíže ujít 100 metrů?-[vždy]
6. Dělal/a Vám potíže se pohybovat doma tak snadno, jak byste si přál/a?-[vždy]
7. Bylo pro Vás obtížné se pohybovat na veřejnosti?-[vždy]
8. Potřeboval/a jste někoho, aby Vás doprovázel, když jdete ven?-[vždy]
9. Dělal/a Vám potíže ujít 1 kilometr? -[nikdy]
10. Dělal/a Vám potíže ujít 1 kilometr? -[vždy]

Obr. 4.8: Vyhodnotenie dotazníkového vyšetrenie pacienta

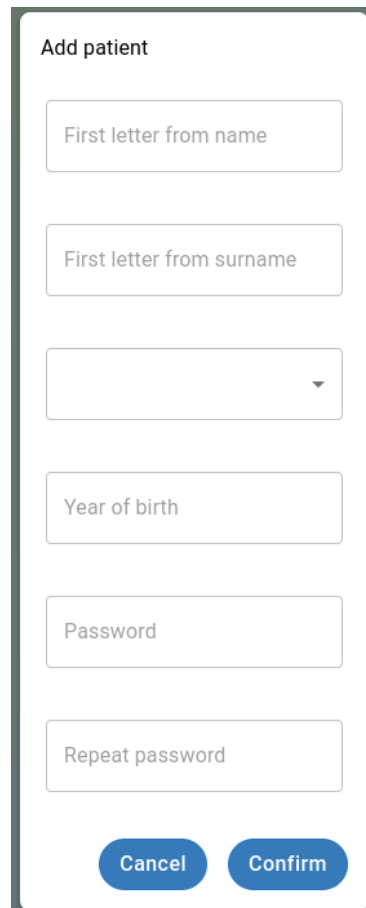
4.3.4 Pridávanie nových pacientov a neuroológov

Implementácia pridania pacienta do systému je postavená ako modálne okno, ktorého ukážka sa nachádza na obr. 4.9, v ktorom sú použité textové polia, ktoré neuroológ v okne vyplní. Tlačítko potvrdzujúce vyplnené údaje potom vykoná akciu poslania POST požiadavky na server. Požiadavka je následne spracovaná a ak

sú všetky údaje validné, je pacient pridaný do aplikácie. Po pridaní do aplikácie je možné okamžite pacientovi pridávať potrebné vyšetrenia.

Neurológova časť aplikácie potom obsahuje aj tlačítka na vymazanie pacienta zo systému, rovnako aj tlačítka na uvoľnenie pacienta zo štúdie.

Dôležitá funkcionálnosť, ktorú má ako jediný **admin** je, že dokáže pridávať do systému neurológov. Dizajn modálneho okna pre pridávanie neurológov je veľmi podobný tomu, aký využíva neurológ na pridávanie pacientov.

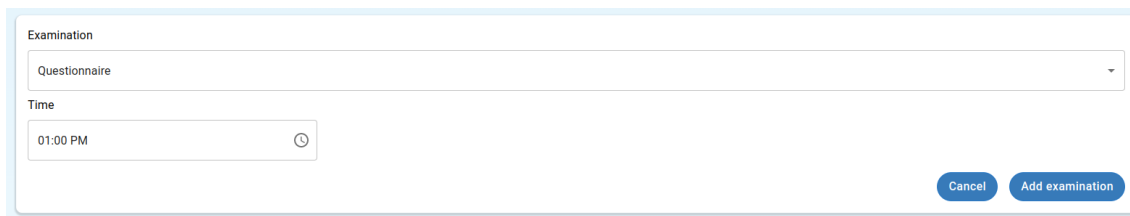


The image shows a mobile application dialog box titled "Add patient". It features a vertical stack of input fields: a text field for "First letter from name", another for "First letter from surname", a dropdown menu, a text field for "Year of birth", a text field for "Password", and a text field for "Repeat password". At the bottom of the dialog are two rounded buttons: "Cancel" and "Confirm".

Obr. 4.9: Dialógové okno slúžiace na pridávanie vyšetrení.

4.3.5 Pridávanie vyšetrení

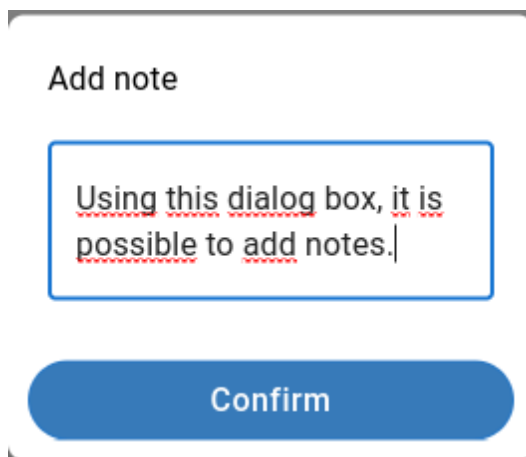
Pridávanie vyšetrení úzko súvisí samozrejme s komponentou kalendáru. Ten totiž určuje, na ktorý deň má byť vyšetrenie určené. Tento údaj je následne cez argumenty odoslaný do komponenty, ktorá odošle vyšetrenie na server. Pri vyplňaní údajov o vyšetrení sa vyplňa čas, typ vyšetrenie a prípadne opakovania. Na obr. 4.10 je znázornené, akým spôsobom neurológovia pridávajú pacientom vyšetrenia.



Obr. 4.10: Časť aplikácie, slúžiaca na pridávanie vyšetrení.

4.3.6 Pridávanie poznámok

Poznámky môže neurológ pridať k pacientovmu profilu pomocou modálneho okna, rovnako ako v prípade pridávania vyšetrení. Poznámka je v zásade iba text odoslaný pomocou POST requestu do databázy, kde sa uloží s jedinečným ID, dátumom a s údajom, kto poznámku vytvoril. Dôležité je ešte samozrejme aj vytvorenie databázového spojenia ku ktorému pacientovi sa daná poznámka vzťahuje. Poznámky je možné potom zobrazíť za pomoci tlačítka "Show all notes". Na obr. 4.11 je znázornené dialógové okno, slúžiace na pridávanie poznámok.



Obr. 4.11: Dialógové okno slúžiace na pridávanie poznámok.

4.4 Zabezpečenie aplikácie

Aby sa zaistila bezpečnosť, tak je nutné do aplikácie sa prihlasovať. Pre autentizáciu je nutné rovnako ako v podaní pacienta, tak aj v podaní neurológa je potrebné zadať **id** a **heslo**. Po úspešnej autentizácii je následne vygenerované sessionId, ktorý sa uloží do cookies.

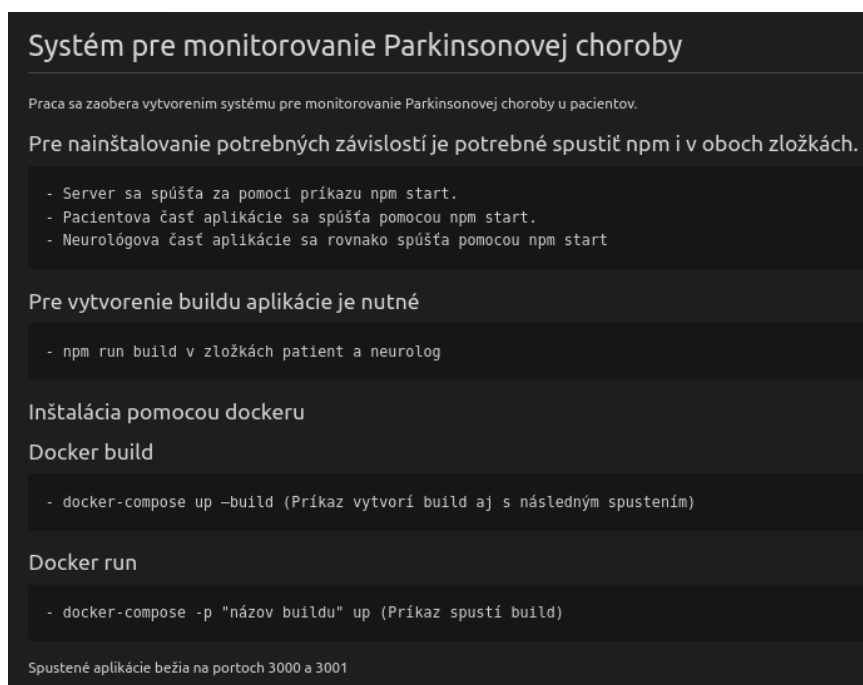
4.5 Testovanie implementácie

Aplikácia bola testovaná počas vývoja manuálne. Z jednoduchosti použitia aplikácia nebolo nutné písať nejaké rozsiahle automatické testy. Testovaním sa postupne zistilo viacero nedostatkov, ktoré boli postupne riešené. Rovnako tak sa napríklad počas testovania zistilo, že jedno tlačítko z návrhu je zbytočné a tak bolo z aplikácie odstránené. Aplikácia bola testovaná na rôznych operačných systémoch a to konkrétne MacOS, Windows, Ubuntu a Android.

4.6 Inštalácia systému

Vďaka dockeru je možné systém nainštalovať pomerne jednoducho. Build aplikácie je možné docieľiť za pomoci príkazu: **docker-compose up --build**, prípadne v súbore **package.json** je definovaný aj príkaz **npm run docker:build**, ktorý vykonáva to isté. K spusteniu aplikácie pomocou dockeru je možné využiť nasledujúci príkaz **npm run docker:up**.

Samozrejme build častí aplikácie implementovaných v reacte je možné získať aj bez pomoci dockeru, a to jednoduchým príkazom **npm build**. Následne je možné spustiť túto aplikáciu tak, že sa spustí node server, pomocou príkazu **npm start**, ktorý si build dohľadá a bude vedieť fungovať spolu s **frontendom**.



Obr. 4.12: Readme súbor s popisom inštalácie

Po spustení aplikácie beží pacientova časť na porte **3000** a neurológova časť systému na porte **3001**.

Podrobnejšie informácie k inštalácii a spúšťaniu aplikácie sú popísané aj v súbore **README.md**, priamo v koreňovom adresári projektu. Na obr. 4.12 je možné vidieť spomínaný Readme súbor.

5 Legislatívna úprava

Práca sa dotýka viacerých legislatívnych úprav. Tými sa zaoberá táto kapitola. Jedná sa najmä o ochranu osobných údajov. Dôležitou súčasťou je však aj zabezpečenie prenosu dát.

5.1 Ochrana osobných údajov - zákony

Asi každý v poslednej dobe zaregistroval pojem GDPR, čo je skratka z angličtiny, ktorá sa dá v plnom znení preložiť ako všeobecné nariadenia o ochrane osobných údajov. GDPR obsahuje pár základných pojmov, ktoré je dobré vysvetliť.

5.1.1 Osobný údaj

Definícia osobného údaje v originálnom znení podľa ministerstva vnútra Českej Republiky je nasledovná: „*Osobním údajem je každá informace o identifikované nebo identifikovatelné fyzické osobě (subjektu údajů). Identifikovatelnou fyzickou osobou je fyzická osoba, kterou lze přímo či nepřímo identifikovat, zejména odkazem na určitý identifikátor (jméno, číslo, síťový identifikátor) nebo na jeden či více zvláštních prvků fyzické, fyziologické, genetické, psychické, ekonomické, kulturní nebo společenské identity této fyzické osoby.*“ [13]

5.1.2 Spracovanie osobných údajov

Definícia spracovania osobných údajov v originálnom znení podľa ministerstva vnútra Českej Republiky je takáto: "*Zpracováním je jakákoli operace nebo soubor operací s osobními údaji nebo soubory osobních údajů, který je prováděn pomocí či bez pomoci automatizovaných postupů, jako je shromáždění, zaznamenání, uspořádání, strukturování, uložení, přizpůsobení nebo pozměnění, vyhledání, nahlédnutí, použití, zpřístupnění přenosem, šíření nebo jakékoli jiné zpřístupnění, seřazení či zkombinování, omezení, výmaz nebo zničení*

Zpracování ve smyslu Obecného nařízení však nelze chápat jako jakékoli nakládání s osobním údajem. Zpracování osobních údajů je nutné považovat již za sofistikovanější činnost, kterou správce s osobními údaji provádí za určitým účelem a z určitého pohledu tak činí systematicky. Pro nakládání s osobními údaji způsobem, který není zpracováním, poskytuje ochranu např. zákon č. 89/2012 Sb., občanský zákoník. Obecným nařízením se tak jako správci řídí pouze subjekty, které osobní údaje zpracovávají ve smyslu definice zpracování." [13]

5.1.3 Osobné údaje a spracovanie v aplikácii

V súčasnosti sa zvyšujú požiadavky užívateľov na zabezpečenie súkromia v aplikáciách. V aplikácii sa preto veľmi dbá na anonymitu užívateľov, to znamená, že identita užívateľov zostáva skrytá. Aj vďaka tomu, že aplikácia funguje tak, že do jej prostredia môžu byť zaregistrovaní iba pacienti neurológa, ktorý ich zaregistruje, sa dajú osobné údaje v aplikácii, čo možno najviac obmedziť. Pacientom je priradený na prihlasovanie vygenerovaný kód, ktorý nijako nesúvisí s ich menom, dátumom narodenia či podobnými osobnými údajmi, to zaručuje anonymitu užívateľa. Jediné dostupné údaje o pacientovi v aplikácii sú jeho iniciály. To však neplatí o neurológoch. U jednotlivých pacientov totiž bude viditeľné meno neurológa, ktorý ich zaregistroval.

Ako bolo už spomínané v kapitole implementácie tak rovnako, ako pacienti, tak aj neurológovia sa do aplikácie potrebujú prihlasovať. Po samotnej registrácii je heslo uložené v databázi ako hash. V aplikácii je využitá hashovacia funkcia **Bcrypt**.

Dôležité pri tvorbe hesla sú aj nejaké povinné pravidlá, ktoré heslo musí spĺňať. Samozrejmosťou je napríklad dĺžka hesla, kde heslo musí mať aspoň 8 znakov.

5.2 Zabezpečený prenos dát

Dáta jednotlivých pacientov a neurológov je možné odosielať, až po prihlásení sa do aplikácie. Tým sa zaisťuje, že prenos dát v aplikácii je umožnený iba oprávneným osobám, ktoré majú právo vykonávať potrebnú akciu.

K jednotlivým súborom sa pristupuje priamo. Po zavolaní potrebnej požiadavky je poslaní užívateľovi obsah alebo zoznam súborov, podľa toho, kde na serveri sa súbor nachádza.

5.3 Ikony

Pri tvorbe aplikácie bolo potrebné využiť ikony. V aplikácii sú použité ikony zo stránky <https://iconify.design/>. Iconify je open-source projekt, takmer všetky časti projektu využívajú **MIT licenciu**, až na niektoré staršie balíčky, využívajúce licencie **Apache 2.0**, alebo **GPL 2.0**. Podľa týchto licencií je možné ikony využiť aj na komerčné účely.

Záver

Hlavným cieľom diplomovej práce bolo implementovať aplikáciu, ktorá umožní neuroológom monitorovať priebeh Parkinsonovej choroby u pacientov. Dizajn a funkcionálnosť aplikácie boli implementované na základe vopred stanoveného návrhu, ktorý sa snaží byť vzhľadom na cieľovú skupinu, čo možno najjednoduchší. Akékoľvek zmeny od pôvodného návrhu boli konzultované a následne sa vybralo vhodné riešenie. Jednotlivé časti implementácie sú v diplomovej práci podrobne opísané. Implementovaný systém je možné označiť ako funkčný na základe návrhu.

Pred implementovaním celého systému boli preskúvané technológie, ktoré následne boli predstavené. Rovnako tak boli preskúvané aj podobné riešenia monitorovania pacientov s Parkinsonovou chorobou, ktoré následne takisto boli predstavené a porovnané. V práci bolo takisto nutné zakomponovať aj legislatívnu úpravu, keďže implementovaný systém sa dotýka najmä ochrany osobných údajov u pacientov. Implementovaný systém zahŕňa požiadavky na bezpečnosť užívateľov a spĺňa tak nároky na zabezpečenie.

Systém vytvorený v rámci tejto diplomovej práce by mal byť čoskoro využívaný neuroológmi a pacientmi. Systém bude pomáhať pacientom a zároveň urýchľovať prácu neuroológom. V budúcnosti je určite vhodné na základe spätnej väzby od užívateľov aplikácie, implementovať prípadné ďalšie funkcionality, alebo upraviť dizajn.

Literatúra

- [1] Parkinson's Disease: Causes, Symptoms, and Treatments, National Institutes of Health [online]. [cit. 28. 11. 2022]. Dostupné z URL: <https://www.nia.nih.gov/health/parkinsons-disease>
- [2] PWA Security: All You Need to Know, Nov 22, 2021 Koombea. [online]. [cit. 16. 4. 2023]. Dostupné z URL: <https://www.koombea.com/blog/pwa-security/>
- [3] Parkinson's disease, 2022 Mayo Foundation for Medical Education and Research (MFMER). [online]. [cit. 28. 11. 2022]. Dostupné z URL: <https://www.mayoclinic.org/diseases-conditions/parkinsons-disease/symptoms-causes/syc-20376055>
- [4] PWA stats. Cloud Four, Inc. [online]. [cit. 3. 11. 2022]. Dostupné z URL: <https://www.pwastats.com>
- [5] Čo je to vlastne PWA. EEA s.r.o. [online]. [cit. 3. 11. 2022]. Dostupné z URL: <https://www.eea.sk/zdielame-know-how/pwa-progressive-web-application/>
- [6] Sheppard, D. Beginning Progressive Web App Development: Creating a Native App Experience on the Web, Apress, 2017. ISBN 9781484230909
- [7] Ater, T. Building Progressive Web Apps: Bringing the Power of Native to the Browser, O'Reilly Media 2017. ISBN 9781491961650
- [8] Oficiálne stránky frameworku react, 2022. Meta Platforms, Inc.[online]. Dostupné z <https://reactjs.org/>
- [9] Finger tapping, 2022. Sage Bionetworks[online]. Dostupné z https://sagebionetworks.org/tools_resources/finger-tapping/
- [10] The Parkinson's Disease Questionnaire (PDQ-39), 2022. Parkinson's UK[online]. Dostupné z <https://www.parkinsons.org.uk/professionals/resources/parkinsons-disease-questionnaire-pdq-39>
- [11] Parkinson's disease sleep scale (PDSS - 2), 2022. Parkinson's UK[online]. Dostupné z <https://www.parkinsons.org.uk/professionals/resources/parkinsons-disease-sleep-scale-pdss-2>
- [12] Stiasny-Kolster K, Mayer G, Schäfer S, Möller JC, Heinzel-Gutenbrunner M, Oertel WH. The REM sleep behavior disorder screening questionnaire—a new diagnostic instrument. Mov Disord. 2007 PMID: 17894337.

- [13] Ministerstvo vnitra České republiky: Základní pojmy v GDPR[online]. 2022, [cit. 28. 11. 2022]. Dostupné z URL: <<https://www.mvcr.cz/gdpr/clanek/zakladni-pojmy-v-gdpr.aspx>>.
- [14] Juan Camilo Vasquez Correa, Apkinson [online]. [cit. 28. 11. 2022]. Dostupné z URL: <https://jcvasquezc.github.io/software/apkinson/>
- [15] Introduction to JSON Web Tokens, Auth0 [online]. [cit. 28. 11. 2022]. Dostupné z URL: <https://jwt.io/introduction>
- [16] Fetch API, mdn web docs, mozilla foundation [online]. [cit. 28. 11. 2022]. Dostupné z URL: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API