VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DIGITALIZACE NÁSTROJE PRO ZACHYCENÍ NEURODEGENERATIVNÍCH ONEMOCNĚNÍ BĚHEM LÉKAŘSKÉHO SEZENÍ

DIGITALIZATION OF TOOLS USED FOR SCREENING FOR NEURODEGENERATIVE DISEASE DURING MEDICAL EXAMINATIONS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — PAVOL PÁNISZ
AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. VÍTĚZSLAV BERAN, Ph.D.
SUPERVISOR

BRNO 2025

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

Ústav:	Ústav počítačové grafiky a multimédií (UPGM)	161412
Student:	**Pánisz Pavol**	
Program:	Informační technologie	
Název:	**Digitalizace nástroje pro zachycení neurodegenerativních onemocnění během lékařského sezení**	
Kategorie:	Uživatelská rozhraní	
Akademický rok:	2024/25	

Zadání:

1. Prostudujte principy tvorby mobilnich aplikací a návrhu uživatelských rozhraní. Dále se seznamte s metodikou screeningu primarní progresivní afázie.
2. Analyzujte existující aplikace používané v logopedické praxi pro účely detekce onemocnění. Navrhněte uživatelské rozhraní pro SW aplikaci, která adaptuje momentálně využívané metody screeningu primární progresivní afázie.
3. Implementujte navržené řešení s využitím existujících technologií.
4. Testujte vytvořené řešení s uživateli a iterativně je vylepšujte.
5. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu; vytvořte plakát nebo krátké video pro prezentování projektu.

Literatura:

- Tidwell et al. *Designing Interfaces: Patterns for Effective Interaction Design*. O'Reilly, 2020, ISBN: 9781492051961
- HARTSON Rex. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. 2012. ISBN 9780123852427
- Dále dle pokynu vedoucího.

Při obhajobě semestrální části projektu je požadováno:
Body 1., 2. a částečně 3.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce:	**Beran Vítězslav, doc. Ing., Ph.D.**
Vedoucí ústavu:	Černocký Jan, prof. Dr. Ing.
Datum zadání:	1.11.2024
Termín pro odevzdání:	14.5.2025
Datum schválení:	12.11.2024

Abstrakt

Cieľom tejto práce bolo navrhnúť a vytvoriť aplikáciu, ktorá zefektívňuje logopédom vykonávanie testu Mini Linguistic State Examination. Podstata riešenia spočíva v zjednodušovaní alebo automatizácií procesov počas testovania, ktoré by logopéd musel inak vykonávať manuálne. Vytvorené riešenie deleguje tieto procesy na aplikáciu, čo má za dôsledok efektívnejšiu administrácie testu a zvýšenie kvality samotného testovania.

Abstract

The goal of this thesis was to design and develop an application which simplifies performing the Mini Linguistic State Exam screening test by speech therapists. The essence of the solution lies within the optimization or automation of multiple processes during testing that a speech therapist would otherwise have to perform manually. The implemented solution delegates these processes onto the application, which makes the administration of the test more efficient and also increases the quality of the testing itself.

Kľúčové slová

používateľské rozhranie, webová aplikácia, primárna progresívna afázia, Mini Linguistic State Examination, používateľská skúsenosť, Vue, IndexedDB

Keywords

user interface, web application, primary progressive aphasia, Mini Linguistic State Examination, user experience, Vue, IndexedDB

Citácia

PÁNISZ, Pavol. *Digitalizace nástroje pro zachycení neurodegenerativních onemocnění během lékařského sezení.* Brno, 2025. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Ing. Vítězslav Beran, Ph.D.

Digitalizace nástroje pro zachycení neurodegenerativních onemocnění během lékařského sezení

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením doc. Ing. Vítězslava Berana, Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

. . . . . .
Pavol Pánisz
14. mája 2025

Poďakovanie

Týmto by som chcel poďakovať doc. Ing. Beranovi, Ph.D., za jeho cenné rady, trpezlivosť a nasmerovanie počas tvorby tejto práce. Ďalej by som chcel poďakovať Tomášovi Benkovi, ktorý informoval vývoj aplikácie po logopedickej stránke.

Obsah

1 Úvod **3**

2 Teória **4**
2.1 Mini Linguistic State Exam a jeho Slovenská adaptácia 4
2.2 Používateľská skúsenosť 10
2.3 Aplikácie používané v logopedickej praxi 14
2.4 Implementačné technológie 17

3 Návrh **19**
3.1 UX analýza papierovej verzie MLSE 19
3.2 Modelovanie požiadaviek 21
3.3 Používateľské rozhranie 24
3.4 Dátový model 30

4 Implementácia **32**
4.1 Výber cieľových zariadení 32
4.2 Výber implementačnej technológie 33
4.3 Architektúra aplikácie 34
4.4 Architektúra GUI 35
4.5 Architektúra backendu 36
4.6 Formatívne testovanie a jeho dopad na implementáciu 38
4.7 Tlačidlo „všetko bez chyby“ 40
4.8 Problémy s nahrávaním na tablete iPad 40
4.9 Sumatívne testovanie 41
4.10 Výsledky testovania 44

5 Záver **45**

Literatúra **46**

Zoznam obrázkov

2.1 Prvý hárok ukázaný pacientovi v subteste 1. 6
2.2 Prvý hárok používaný administrátorom v subteste 1. 7
2.3 Hárok pre vypočítanie celkového súčtu bodov do všetkých domén. 9
2.4 Rozhodovací strom z pôvodného MLSE 10
2.5 Goal Directed Design, preložený do Slovenčiny. 11
2.6 Kolobeh návrh – implementácia – vyhodnocovanie – analýza 14
2.7 iScreen Aphasia - ukážka obrazovky pre pacienta 15
2.8 MoCA Duo - ukážka obrazovky počas testovania pacienta 16

3.1 Obrazovka testovania pacienta 25
3.2 Obrazovka vyhodnocovania testu 26
3.3 Obrazovka Výsledkov 28
3.4 Obrazovka Zoznamu Pacientov 29
3.5 Obrazovka Karty Pacienta 30
3.6 Dátový model testu a vyhodnotenia 31

4.1 Architektúra systému z pohľadu oddelenia zodpovednosti 35
4.2 Prvý návrh pre vylepšenú komponentu ovládania inštrukcie 39
4.3 Druhý návrh pre vylepšenú komponentu ovládania inštrukcie, ako ho vidno v implementácií 39
4.4 Tlačidlo pre zaškrtnutie / odfajknutie všetkých políčok stĺpca „bez chyby“, ako ho vidno v implementácií 40

Kapitola 1

Úvod

Digitálne technológie postupom času prenikli do a transformovali značnú časť ľudskej činnosti. Bez zmeny sa nezaobišla ani disciplína logopédie, v ktorej sa tieto technológie čoraz väčšmi stávajú neoddeliteľnou súčasťou dennej praxe. Každý rok sú vytvárané aplikácie, ktoré majú za účel pomôcť logopédom pri ich činnosti, či sa už jedná o skríning, diagnostiku alebo terapiu.

Táto práca bola vytvorená s cieľom prispieť do narastajúcej kolekcie aplikácií používaných logopédmi. Zameranie bolo na vytvorenie aplikácie, ktorá je digitalizovanou verziou experimentálnej slovenskej adaptácie testu Mini Linguistic State Examination. Aplikácia bola vytvorená v nádeji, že pomôže logopédom s administráciou a administratívou spojenou s testovaním pomocou tohoto nástroja, a tým pádom skvalitní ich služby voči klientom.

Táto technická správa dokumentuje proces tvorby tejto aplikácie, s nasledovným členením kapitol: Kapitola Teória sa zaoberá analýzou východísk pre túto prácu – predstavuje sa v nej súčasná podoba testu MLSE, zhodnocujú sa trendy aplikácií v logopedickej praxi a približuje sa metodika návrhu a implementácie používateľsky prívetivej aplikácie. V kapitole Návrh sa na týchto poznatkoch stavia a vytvára sa špecifikácia pre takúto aplikáciu. V kapitole Implementácia sa napokon opisuje najdôležitejšie rysy implementačného procesu konkrétnymi technológiami, pričom výsledok nadobúda formu webovej aplikácie. Testovanie a vyhodnocovanie tejto aplikácie je poslednou súčasťou kapitoly.

Kapitola 2

Teória

V tejto kapitole sa postupne zoznámime s najdôležitejšími východiskami pre splnenie cieľu tejto práce – vytvoriť digitalizovanú verziu už existujúceho logopedického testu. Nebudeme si pri tom detailne vysvetľovať koncepciu testu z logopedického hľadiska, či ako konkrétne sa aplikácie programujú pomocou rozličných technológií. Namiesto toho ponúkneme plytký, avšak dostatočný prehľad do všetkých zásadných oblastí týkajúcich sa tejto práce. Konkrétne sú to 4 oblasti, každá obsiahnutá v jednej podkapitole.

V prvej podkapitole je čitateľovi predstavený východiskový logopedický test, zvaný Mini Linguistic State Examination. Pochopenie tohoto testu, jeho formy a procesov spätých s ním, je najzásadnejší bod pre celú prácu, nakoľko je postavená na jeho základe. Naša práca vo svojej podstate tento test vylepšuje, z pohľadu používateľskej skúsenosti.

V druhej podkapitole približujeme práve tematiku používateľskej skúsenosti. Túto oblasť, zaoberajúcu sa navrhovaním používateľsky prívetivých (nie len digitálnych) produktov, je taktiež dôležité poznať pre našu prácu – to, či budú logopédi túto aplikáciu chcieť skutočne používať namiesto súčasnej papierovej verzie, závisí z veľkej väčšiny práve od kvality používateľskej skúsenosti.

Hoci je používateľská skúsenosť subjektívna, do istej miery je informovaná predošlými skúsenosťami – v našom prípade skúsenosťami s inými aplikáciami, ktoré sa využívajú na podobné účely, ako tá naša. Aj preto je predposlednou podkapitolou rozbor už existujúcich aplikácií a technologických trendov v oblasti logopédie.

Poslednou podkapitolou je prehľad nástrojov využiteľných na tvorbu navrhnutej aplikácie. Približujeme súčasný stav metodík na tvorbu aplikácií, so zameraním na metodiky využívajúce webové technológie.

2.1 Mini Linguistic State Exam a jeho Slovenská adaptácia

Nakoľko táto práca stavia na existujúcom postupe administrácie Mini Linguistic State Exam (skrátene MLSE), je nevyhnutné načrtnúť podstatu tohoto testu - akú má test formu a ako testovanie v súčasnosti prebieha.

Uvedenie do problematiky

Z dôvodu, že sa pohybujeme v informatikom cudzej oblasti logopédie, poďme si najskôr objasniť niektoré pojmy. Logopédia je odbor, ktorý sa zaoberá diagnostikou, terapiou a prevenciou porúch komunikácie, reči, hlasu a jazyka. Afázia je získaná porucha produkcie a porozumenia reči, ktorá vzniká pri poškodení rôznych častí mozgu. Primárna progresívna afázia (ďalej len PPA) je jeden z typov afázie, pričom sa v súčasnosti člení na 3 varianty: agramatický, sémantický a logopenický. PPA sa oproti iným druhom afázie líši najmä pomalým stupňovaním syndrómov. Priemerný vek nástupu ochorenia je 55 až 60 rokov [12].

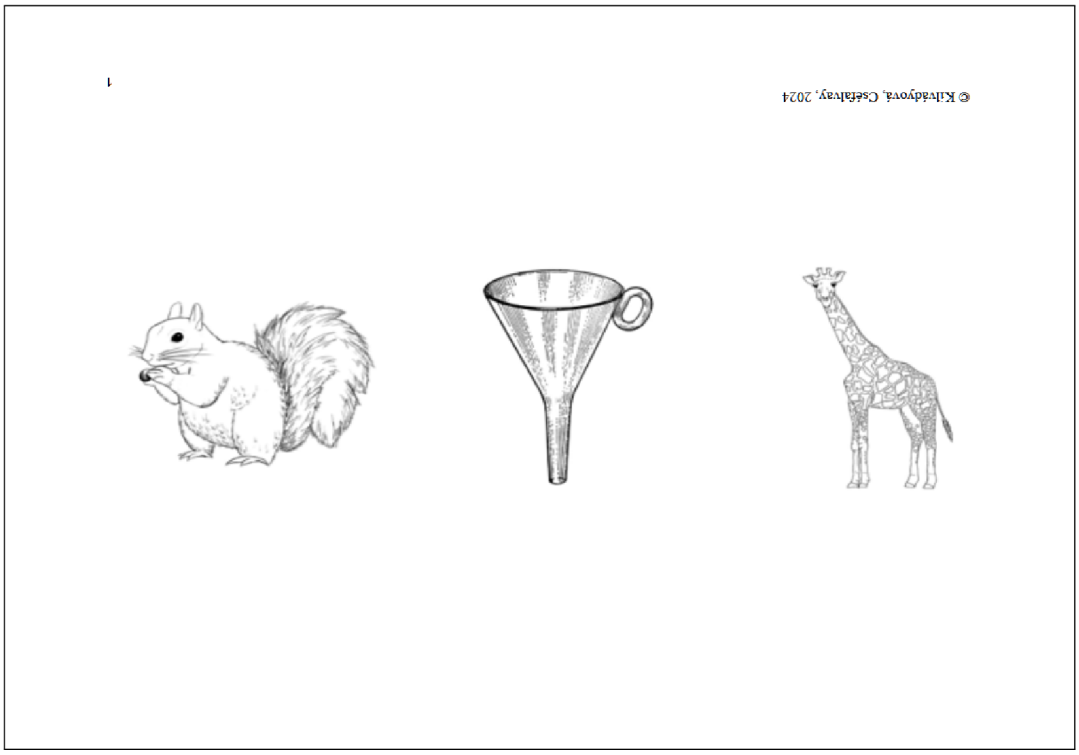
MLSE je logopedický skríningový test, ktorý bol predstavený v roku 2019 vo Veľkej Británii ako jednodušší spôsob skríningu proti PPA [16]. Oproti iným skríningovým testom proti PPA má za úlohu byť menej náročný na administráciu – okrem iného, administrátor nemusí byť trénovaný v oblasti afaziológie, avšak istá miera odbornosti musí byť beztak zachovaná. Tento test teda nie je mienený pre laickú verejnosť.

Nakoľko je MLSE navrhnuté pre populáciu Veľkej Británie, zohľadňujúc kultúrne poznatky tamojšej populácie (napr. familiaritu s konkrétnymi slovami), musí prejsť procesom tzv. adaptácie predtým, než sa môže používať v iných krajinách. Proces adaptácie zahŕňa jazykový preklad, úpravu obsahu a validovanie na populácií danej krajiny. Adaptácia prebehla aj na Slovensku, kde bola v roku 2024 vytvorená experimentálna verzia s názvom MLSEsk [13]. V čase písania je táto experimentálna verzia dokončená, avšak nie je ešte použiteľná v praxi. Je potrebné jej platnosť overiť na väčšom počte respondentov, než sa využívalo počas procesu adaptácie.

Forma testu

Nástroj MLSE má formu testovacej brožúry. V praxi sa táto brožúra tlačí na papier, čím vzniká 36 hárkov jednostranných výtlačkov formátu A4. Z tohoto dôvodu v tejto práci momentálnu podobu MLSE nazývame aj **papierová verzia MLSE**. Isté vytlačené hárky sú mienené pre pacienta, a iné sú mienené len pre administrátora. Tieto hárky dokopy utvárajú test, ktorý sa skladá z 11 subtestov. Každý testuje pacienta iným spôsobom a vykonávajú sa sekvenčne. Administrátor v rámci vyhodnocovania sleduje výkon pacienta a podľa neho udeľuje body do 6 domén. Tieto domény majú kvantitatívny charakter a každému subtestu prislúcha istá podmnožina z tejto množiny. Na základe súčtov bodov pre jednotlivé domény sa vytvára celkové skóre.

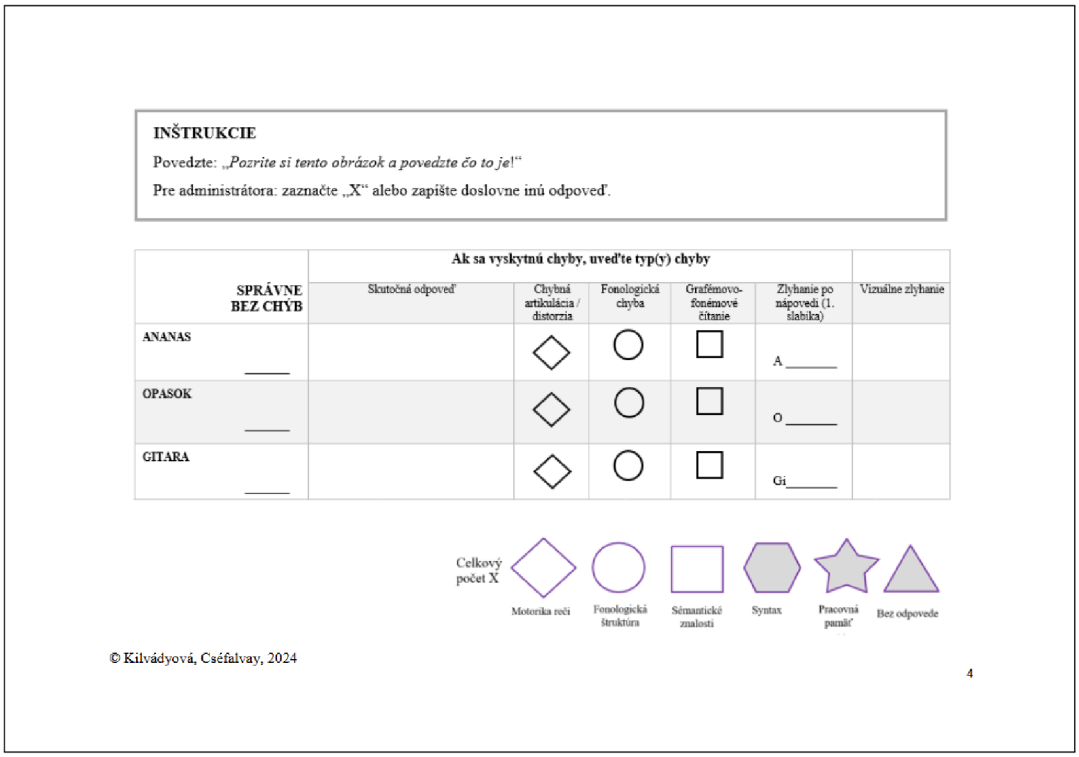
Nasledujú ukážky hárkov zo slovenskej adaptácie MLSE a detailný opis ich obsahu. Pre prípad potreby vidieť všetky hárky, prikladáme pôvodnú anglickú verziu do prílohy tejto práce.



Obr. 2.1: Prvý hárok ukázaný pacientovi v subteste 1.

Počas všetkých subtestov administrátor zadáva pacientovi **inštrukcie**. V niektorých subtestoch sa inštrukcie vzťahujú na hárky, ktoré administrátor predloží pacientovi, s príkladom takéhoto hárku na obrázku vyššie. V iných subtestoch sa inštrukcie vzťahujú na slová alebo vety, ktoré administrátor vyslovil. V našej práci tieto objekty, na ktoré sa inštrukcie vzťahujú, kolektívne nazývame **položky**.

Pre príklad, v subteste 1 má pacient pomenovať objekty na predloženom hárku. Celkovo má pomenovať 6 objektov, no nakoľko sa na 1 hárok autori rozhodli umiestniť len 3 objekty, je potrebné 2 hárky.



INŠTRUKCIE

Povedzte: *„Pozrite si tento obrázok a povedzte čo to je!“*

Pre administrátora: zaznačte „X“ alebo zapíšte doslovne inú odpoveď.

		Ak sa vyskytnú chyby, uveďte typ(y) chyby					
	SPRÁVNE BEZ CHÝB	Skutočná odpoveď	Chybná artikulácia / distorzia	Fonologická chyba	Grafémovo-fonémové čítanie	Zlyhanie po nápovedi (1. slabika)	Vizuálne zlyhanie
ANANAS	____		◇	○	□	A _____	
OPASOK	____		◇	○	□	O _____	
GITARA	____		◇	○	□	Gi_____	

Obr. 2.2: Prvý hárok používaný administrátorom v subteste 1.

Každému subtestu prinalieha jeden alebo viac hárkov pre administrátora. Tieto hárky sa pacientovi nikdy neukazujú. Počet hárkov sa odvíja od počtu položiek v subteste, no všetky takéto hárky obsahujú vždy 2 oddelené celky:

1. Inštrukciu (na vrchu hárku), ktorú administrátor nahlas číta pacientovi.
2. Maticu pre zaznamenávanie chýb vykonaných pacientom počas aplikovania inštrukcie na každú položku.

Z obrázku vyššie sú zrejmé názvy spomínaných domén: Motorika reči, Fonologická štruktúra, Sémantické znalosti, Syntax, Pracovná pamäť a Bez odpovede.

Z obrázku je tiež možné vďaka šedým piktogramom dedukovať spomenutý fakt, že pre rôzne subtesty sa položky hodnotia len v rámci niektorých domén. Hodnotením položky máme na mysli zaznačenie „X“ do niektorých piktogramov domén jej riadku. Pre jednotlivé stĺpce sa tieto „X“ neskôr sčítajú a tak sa pre daný subtest pre danú doménu vytvorí celková bodová hodnota. Vo väčšine subtestov má jedno „X“ bodovú váhu 1.

Pri hodnotení existujú aj iné špecifiká:

- Pre každú položku, ako aj pre celé subtesty, bývajú prítomné polia na anotácie, ktoré však nemajú priamy vplyv na body.
- Niektoré položky môžu mať pre niektorú z domén väčšiu bodovú váhu.
  - Napríklad v subteste 9, v ktorom má pacient opakovať vety vyslovené administrátorom, sú niektoré vety dlhšie ako iné. Tým pádom sa prihliada miernejšie na

chyby spravené počas vyslovovania týchto viet – do domény *Pracovná pamäť* sa pri nich pridelí napr. 1 bod namiesto 4.

- Pre každú položku sa prideľovanie bodov v rôznych doménach môže vylučovať.
  - Napríklad je logické, že keď sa pridelí 1 bod v doméne Bez odpovede, sa pre túto položku nemôže prideliť body do zvyšných domén: Nie je možné zároveň nič nepovedať a zároveň spraviť chybu výslovnosti.

Administrácia testu

Testovanie sa skladá z 2 fází: Testovania pacienta a vyhodnocovania výkonu pacienta.

Počas testovania sedia administrátor a pacient oproti sebe. Jednotlivé subtesty sa vykonávajú sériovo. To znamená, vždy sa číta a vykonáva nasledujúci hárok len po tom, čo sa vykonávanie inštrukcie predošlého dokončilo. Pre demonštráciu je uvedený priebeh subtestu 1:

- V 1. subteste sú pacientovi ukázané 3 obrázky.
- Pacienta sa požiada, aby vymenoval, čo na obrázkoch vidí.
- Po splnení tejto časti sa pacientovi ukážu ďalšie 3 obrázky, ktoré tvoria ďalšiu časť.
- Znova sa ho žiada, aby vymenoval, čo na obrázkoch vidí.

Nakoľko sa subtest 1 skladá len z týchto 2 častí, je po splnený 2. časti subtest 1 celkovo splnený a prechádza sa na subtest 2. Rovnakým štýlom sa prejde všetky zvyšné subtesty, v rastúcom poradí. Testovanie je možné pozastaviť a pokračovať neskôr, avšak ten istý subtest počas sedenia nemožno opakovať.

Druhá fáza, teda vyhodnocovanie výkonu pacienta, sa môže ale neodporúča prevádzať súbežne s fázou testovania pacienta. Počas tejto fázy administrátor udeľuje body pre položky jednotlivých subtestov. Okrem toho si administrátor môže písať anotácie, napríklad ako konkrétne vyslovil pacient nejaké slovo.

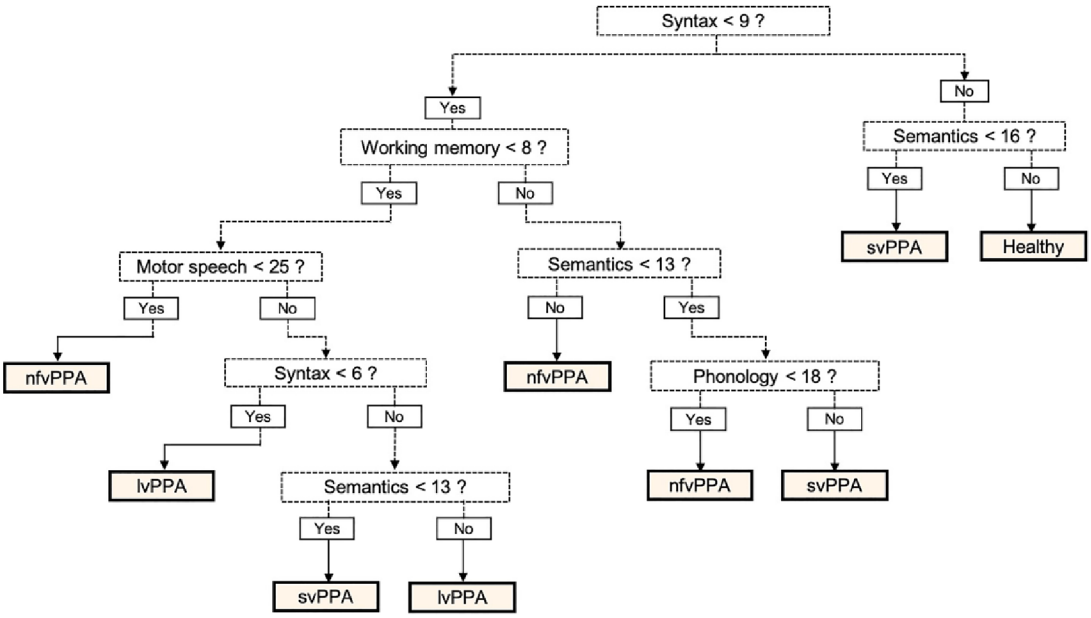
- Ak sa táto fáza deje súbežne s prvou, v praxi to znamená, že administrátor počúva pacienta a počas počúvania si zaznačuje chyby a prideľuje skóre jednotlivým doménam. Keďže počas tohoto nevenuje plnú pozornosť pacientovi, pacient sa môže cítiť pod stresom, čo dokáže zhoršiť jeho výkon. Tento prístup má však tú výhodu, že administrátor si nijakým spôsobom nemusí zapamätať výkon pacienta.
- Ak sa táto fáza deje až po tom, ako sa s pacientom prešlo všetkých 11 subtestov a pacient odišiel z miestnosti, je dôležité si zapamätať pacientov výkon. Za týmto účelom sa zvykne nahrávať buď audio- alebo videozáznam pacienta počas fázy testovania. Videozáznam poskytuje najviac kontextu pre administrátora a tým pádom je preferovaný. Výhoda tohoto prístupu je tá, že administrátor sa môže počas testovania plne venovať pacientovi. Nevýhoda je však získanie súhlasu s tvorbou nahrávky a taktiež možné rozptýlenie v dôsledku vedomia toho, že je človek nahrávaný. Ďalšia nevýhoda je tá, že administrátor musí okrem narábania s hárkami papiera taktiež narábať so samotným nahrávacím zariadením.

Vyhodnotením všetkých subtestov sa administrátor dostáva do situácie, kedy má vo vyhodnocovacích hárkoch na rôznych miestach zaznačené chyby pomocou „X“ a prípadné komentáre. Nasleduje sčítanie bodov pre jednotlivé domény, kedy je kritické, aby sa administrátor nepomýlil. Pre sčítanie týchto bodov slúži samostatný hárok, znázornený na obrázku nižšie.

Test	Strana	◇	○	□	⬡	☆	△	
Pomenovanie obrázkov	2	3	3	3	X	X	X	Celkový počet X na stránku
	4	3	3	3	X	X	X	
Opakovanie slabík	6	3	3	X	X	X	6	
Opakovanie a porozumenie slov	8	1	1	1	X	X	2	
	10	1	1	1	X	X	2	
	12	1	1	1	X	X	2	
Opakovanie pseudoslov	14	3	3	X	X	X	6	
Sémantické asociácie	16	X	X	2	X	X	X	
	18	X	X	2	X	X	X	
Porozumenie viet I.	20	X	X	X	4	X	X	
Porozumenie viet II.	22	X	X	X	2	X	X	
	24	X	X	X	2	X	X	
Čítanie slov a pseudoslov	26	10	10	5	X	X	25	
Opakovanie viet	28	4	4	X	X	10	18	
Písanie	30	X	X	X	1	X	X	
Opis obrázku	32	1	1	2	1	X	5	
Chyby spolu		◇	○	□	⬡	☆	△	

Obr. 2.3: Hárok pre vypočítanie celkového súčtu bodov do všetkých domén.

V pôvodnej verzií MLSE sa dá na základe bodov v jednotlivých doménach stanoviť, či má pacient PPA a ak áno, aký konkrétny subtyp [17]. Toto stanovovanie je graficky znázornené na obrázku nižšie.



Obr. 2.4: Rozhodovací strom z pôvodného MLSE

Pre experimentálnu slovenskú adaptáciu podobný rozhodovací strom v čase písania neexistuje, a tak môže byť výsledkom celého testu MLSE len počet bodov v jednotlivých doménach, ako aj tzv. celkové skóre, vypočítané ako 100 - súčet bodov všetkých domén. Tieto čísla si administrátor následne interpretuje sám. Keďže pacient môže byť časom testovaný aj viackrát, za účelom sledovania gradácie choroby, je žiadúce výsledky testovaní uchovávať.

2.2 Používateľská skúsenosť

V predošlej kapitole sme si predstavili povahu testu MLSE, a nakoľko máme v tejto práci za cieľ vytvoriť digitálnu verziu tohoto testu, nastáva čas položiť si otázku: Aby sme vytvorili aplikáciu vychádzajúcu z tohoto testu, akými princípmi sa máme riadiť? Konkrétne, ako zaistiť, aby sa táto aplikácia skutočne dobre používala? Aby bola užitočná, ba dokonca aby ju chceli logopédi používať radšej, než papierovú verziu? Týmito otázkami sa zaoberá vedná disciplína používateľskej skúsenosti, ktorú si napokon priblížime.

Na začiatku nástupu počítačov boli ich používatelia technicky orientovanej povahy, ochotní, ba dokonca aj hrdí na to, že sa museli učiť pracovať s týmito ťažko použiteľnými nástrojmi. Avšak postupom času, ako sa ich používanie začalo čoraz viac dostávať pomedzi masy ľudí, sa vytvorila potreba navrhovať digitálne produkty tak, aby sa dalo ľahšie naučiť s nimi pracovať, aby netrvalo zbytočne dlho vykonávať rôzne úkoly pomocou nich, ba dokonca aby ich používanie vytváralo pocit spokojnosti. V súčasnosti, keď je na trhu v rôznych kategóriách digitálnych aplikácií hneď niekoľko produktov s veľmi podobnou funkcionalitou, býva dobrá použiteľnosť často rozhodujúcim faktorom pri tom, ktorý z produktov bude najviac používaný [7].

Princípy použiteľnosti

Na aké aspekty systému teda treba dbať, aby bola jeho použiteľnosť čo najlepšia? Jakob Nielsen, jeden z popredných výskumníkov v oblasti použiteľnosti, vo svojom blogu sumarizuje hlavné princípy použiteľnosti nasledovne [15]:

- Naučiteľnosť: Ako jednoducho sa používateľovi robí rôzne úkoly so systémom, keď ho vidí po prvý krát? Je samozrejmé, že čím komplexnejšie úkoly majú byť splniteľné pomocou systému (napr. účtovný softvér), tým viac bude táto metrika trpieť. Avšak aj v prípade komplexných používateľských rozhraní sú rôzne techniky na minimalizovanie kognitívnej záťaže , napr. pomocou možnosti postupného odhalenia ovládacích prvkov.
- Efektívnosť: Ako rýchlo dokáže používateľ plniť rôzne úkoly po tom, čo sa už naučil nástroj používať? Táto metrika závisí od povahy plneného úkolu, avšak existujú všeobecné zásady, ktoré prispievajú k efektívnosti: - Často robené akcie by mali byť čo najrýchlejšie dostupné - Používateľ by nemal dlho čakať, nakoľko to narúša ich flow. - Tu prichádza pojem „neskoršia zmeniteľnosť“ (anglicky *fudgeability*).
- Zapamätateľnosť: Ako rýchlo sa dokáže používateľ znova naučiť pracovať so systémom po tom, čo ho dlhšiu dobu nepoužíval?
- Chybovosť: Koľko chýb robí používateľ pri používaní aplikácie, aká je ich závažnosť a ako rýchlo ich dokáže napraviť?
- Spokojnosť: Aké príjemné je používanie produktu? Táto metrika zahŕňa rôzne aspekty používateľskej skúsenosti, od estetiky produktu až po pocit zmajstrovania produktu. Pocit zmajstrovania produktu (expertízy) je o to väčší, čím je appka komplexnejšia. Adobe Software..

Goal Directed Design

Princípy použiteľnosti máme vytýčené, avšak ako máme navrhnúť produkt tak, aby tieto požiadavky spĺňal do čo najväčšej miery? Kým neexistuje spôsob ako garantovať, aby produkt ponúkal dobrú používateľskú skúsenosť, existujú rôzne metodiky návrhu digitálnych produktov, ktoré majú používateľskú skúsenosť v stredobode pozornosti. Toto je v silnom kontraste s niektorými inými metodikami návrhu, ktoré majú v stredobode pozornosti funkcionalitu a sú preto viac riadené softvérovými inžiniermi, než dizajnérmi [11].

Jednou z metodík návrhu, ktoré sú poháňané zamýšľaním sa nad používateľskou skúsenosťou, je Goal Directed Design, po slovensky Cieľom Smerovaný Dizajn [7]. Goal Directed Design je pomenovaný podľa faktu, že kladie dôraz na pochopenie cieľov, ktoré má produkt umožňovať splniť. Výsledok nasledovania tohoto procesu je špecifikácia vzhľadu a správania, ktorá je následne odovzdaná softvérovým inžinierom na samotný vývoj. Tým pádom sa tento proces javí mať viac lineárnu štruktúru, podobnú tej, akú v softvérovom inžinierstve poznáme pod pojmom vodopádový model životného cyklu vývoja.

Obr. 2.5: Goal Directed Design, preložený do Slovenčiny.

Fázy tohoto dizajnového procesu sú nasledovné:

- Výskum: Vo výskumnej fáze je kľúčové previesť nasledovné úlohy:
  - Stanovenie celkového času na vývoj produktu a hrubej vízie pre produkt.
  - Získanie poznatkov od cieľových používateľov v doméne, ktorej sa produkt týka. Toto zahŕňa rozhovory s budúcimi používateľmi ako aj pozorovacie techniky. Cieľom je zistenie zvyklostí a potrieb používateľov.
  - Analýza súčasných produktov plniacich podobný účel tomu nášmu. Toto má za cieľ vyhnúť sa „opätovnému vynachádzaniu kolesa“, ako aj zisteniu, na aké vzorce interakcie s produktom sú používatelia už zvyknutí.
  - Rozhovory s expertmi v oblastiach, ktoré sa týkajú produktu. Tu je žiadúce získať znalosti nie len z technologického hľadiska, no samozrejme aj z oblasti expertízy cieľových používateľov.
- Modelovanie:
  - V tejto fáze sa na základe poznatkov z predošlého štádia vytvára obraz o archetype používateľa, prípadne používateľov produktu. Odborne sa tento model nazýva persona a je len jedným zo spôsobov zhlukovania používateľov do osobitných kategórií. Zmyslom vytvorenia persony je abstrahovanie si nespočetného množstva rôznych používateľov do menších celkov, pre lepšie pochopenie a prioritizovanie ich požiadaviek.
  - Okrem modelu používateľov sa v tejto fáze taktiež vytvára doménový model, pozostávajúci z rôznych opisov a diagramov priebehu činností, ktoré používatelia môžu robiť. Je dôležité podotknúť, že táto fáza má skôr opisný charakter, než kreatívny. To znamená, dôraz je kladený na hodnoverné opísanie používateľov a ich potrieb.
- Definícia požiadaviek:
  - Až v tejto fáze sa modely používateľov a domény môžu stretnúť a začína sa navrhovať. Deje sa to konkrétne pomocou iteratívne spresňovaných kontextových scenárov [7]. Kontextové scenáre opisujú deje zasadené do života konkrétnej persony, berúc do úvahy jej motivácie, potreby a požiadavky. Dôležité pri týchto naratívoch je, že sa nezameriavame na formu ani konkrétnu funkcionalitu produktu – predstavujeme si ho takmer ako čarovnú skrinku, ako vec ktorá už existuje a je tak samozrejmá, že nie je potrebné opisovať konkrétne detaily jej používateľského rozhrania. Opisuje sa iba, aké ciele persona skrz produkt dosahuje a aké úkoly pri tom plní, no nie, ako presne ich plní.
  - Po iteratívnom upresňovaní kontextových scenárov sa ocitneme v situácií, kedy sme schopný vyslovovať požiadavky na systém za použitia objektov, akcií a kontextov: Zavolaj (akcia) osobe (objekt) priamo z termínu v kalendári (kontext). Tieto 3 zložky sa následne používajú pre spísanie dátových, funkčných a kontextuálnych požiadaviek.
    - Dátové požiadavky budú stanovovať, aké informácie si systém musí pamätať.

    - Funkčne požiadavky budú stanovovať, aké akcie budú v systéme prevediteľné.
    - Kontextové požiadavky budú stanovovať, ktoré informácie spolu súvisia

  Spoločne tieto 3 druhy požiadaviek tvoria základný stavebný kameň, od ktorého sa odráža všetko, počínajúc s návrhom používateľského rozhrania a stanovenia jednotlivých okien aplikácie, až po návrh dátového modelu a výber implementačnej technológie.

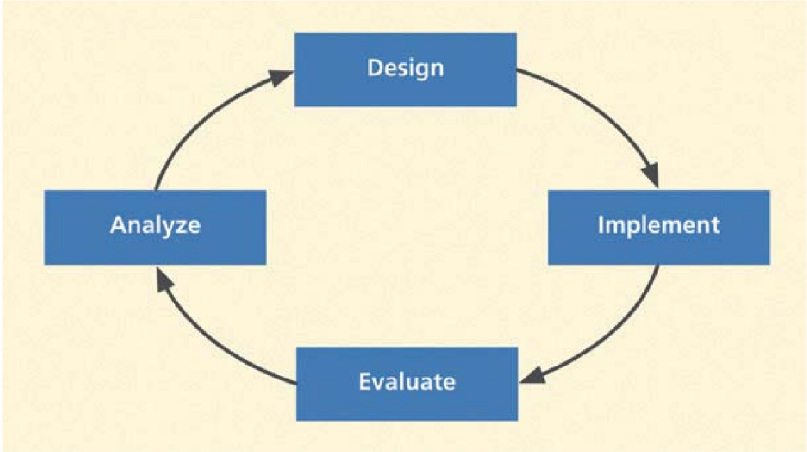
- Definícia rámca:
  - Stavajúc na textovom opise požiadaviek z predošlej fázy návrhového procesu, je táto fáza prvou, v ktorej sa niečo navrhne graficky (resp. fyzicky), najskôr nahrubo. Pri prvotných skicách aplikácie sa pritom opiera o princípy a vzory interaktívneho dizajnu.
  - Návrhové vzory sú zaužívané prvky používateľských rozhraní, ako napríklad dropdown-menu, slider alebo často opovrhované modálne okno.
  - Návrhové princípy sú všeobecné odporučenia, ktorými je rozumné sa riadiť pri vytváraní používateľských rozhraní. Dajú sa hierarchicky usporiadať pomedzi *konceptuálne*, *behaviorálne* a *na úrovni používateľského rozhrania.*
  - Konceptuálne opisujú, ako by sa mal produkt správať. Príkladom je postúra aplikácie - má sa produkt používať dlho alebo krátko? Beží na pozadí systému alebo je na popredí zariadenia?
  - Behaviorálne princípy, laicky povedané, zodpovedajú otázku „Keby bol môj produkt človek, ako by sa správal?“ – Zapamätával by si veci, neprerušoval ma, a podobne.
  - Princípy na úrovni používateľského rozhrania - Ako názov napovedá, tieto sa týkajú praktík vytvárania zmysluplných vzťahov medzi ovládacími prvkami a odvíjajú sa od implementačnej technológie (desktop vs. mobil, napríklad). Patria sem výroky ako „Súvisiace veci by mali byť blízko seba“ alebo „Pri dotykových obrazovkách dizajnujte pre nespresnosť prstov“.

- Spresnenie
  - V tejto fáze sa dolaďujú detaily z hrubých náčrtov vytvorených v predošlej fáze. Vytvára sa vizuálna identita produktu obsahujúca fonty, farby, ikony a podobne. Taktiež sa validuje a testuje návrh, za použitia scenárov *kľúčových ciest.* Tieto scenáre síce majú svoj zárodok v kontextových scenároch, no využívajú fakt, že v tomto bode vývoja už navrhnutú aj formu produktu. Z tohoto dôvodu už nie je nutne produkt považovať za spomínanú čarovnú skrinku a namiesto toho sa v scenároch kľúčových ciest modeluje rôzne situácie, ktorými si persona prechádza, používajúc vytvorený jazyk a dizajn produktu. Inými slovami, v tomto bode sa už sútredíme na opisovanie nie len toho čo robíme, ale aj ako konkrétne to robíme, s akými prvkami rozhrania interagujeme a v akej sekvencií. Tuto sa jasne ukazuje, že tieto scenáre sú viac úlohovo, než cieľovo orientované.
  - Ako je zrejmé z predošlých foriem modelovania systému, naratív je mocným princípom na navrhovane systému. Preto sa ako ďalší nástroj vo fáze spresňovania používa aj storyboarding, prebratý z filmovej industrie. Počas storyboardingu sa

rôzne sekvencie interakcií medzi prvkami a oknami zobrazujú pomocou za sebou idúcich nákresov používateľského rozhrania.

- Podpora vývoja - Finálna fáza vývoja, presahujúca sa s procesom implementácie dizajnu produktu. V tejto fáze sa návrh upravuje v dôsledku zistených technických, časových alebo iných problémov. Hoci v tejto fáze návrhársky team zastáva viac pasívnu úlohu, musí byť stále dostupný pre prípadne zopakovanie tohoto procesu v zmenšenej podobe, na adresovanie meniacich sa požiadaviek na produkt.

Hoci sme sa pri tvorbe tejto práce inšpirovali metodikou Goal Directed Design, nepreberáme jej postupy doslovne - táto metodika sa nám javí ako príliš komplexná pre naše účely, a preto sme si zvolili viac agilný postup: Zakomponovali sme jej myšlienky do všeobecného procesu tvorby produktu [11] a až tento proces sme využili. Konkrétne, tu zmienené praktiky používame najmä v etapách návrhu a testovania:



Obr. 2.6: Kolobeh návrh – implementácia – vyhodnocovanie – analýza

2.3 Aplikácie používané v logopedickej praxi

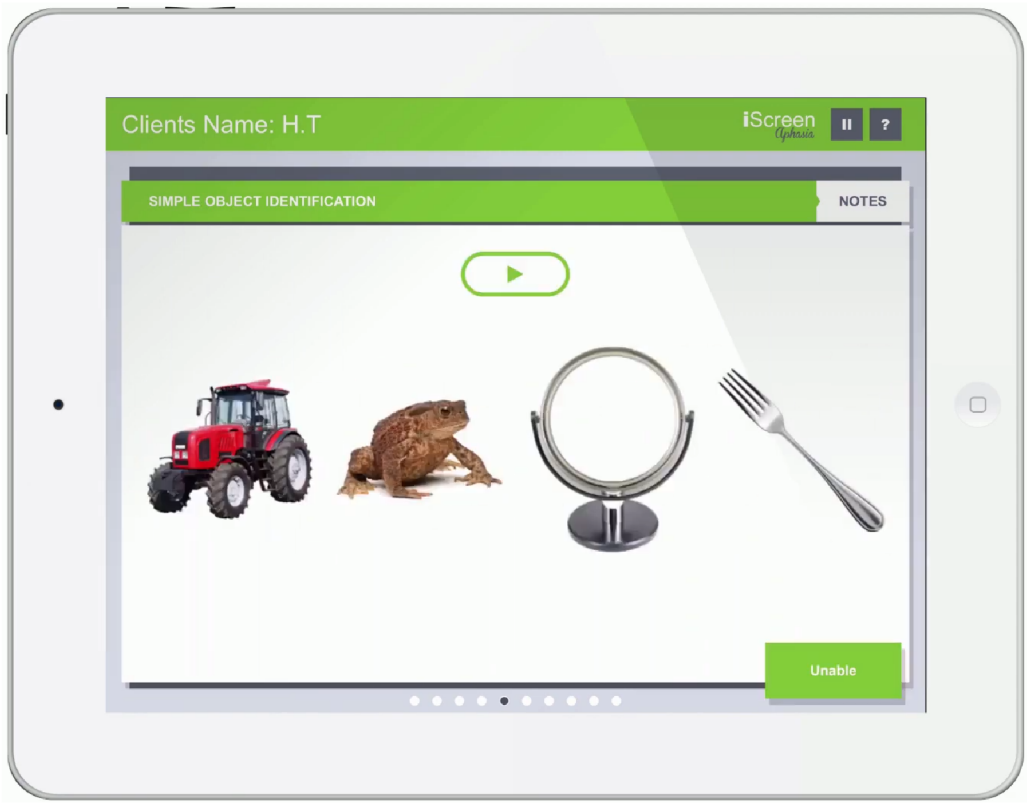
V predošlej kapitole sme si nie len vysvetlili, čo znamená mať dobrú používateľskú skúsenosť, no taktiež sme si priblížili proces, ktorým sme sa riadili pri vývoji výslednej aplikácie. Prvou etapou v tomto cyklickom procese je analýza, v rámci ktorej sme si už predstavili súčasnú podobu testu MLSE. Avšak v oblasti digitalizácie logopedických testov nie sme prví, a preto považujeme za užitočné sa bližšie pozrieť na niektoré aplikácie, ktoré majú podobné ciele tej našej - nie len aby sme zistili, na aké rozhrania sú logopédi už zvyknutí, no aj preto, aby sme spoznali priestor na zlepšenie oproti týmto aplikáciám a inšpirovali sa ich silnými stránkami.

Na túto analýzu sme si vybrali 2 aplikácie, a to iScreen Aphasia a MoCA Duo. Obe aplikácie sú totiž nástroje na administráciu logopedických testov a na ich vyhodnotenie. Najskôr si ich analyzujeme jednotlivo a následne vyzdvihneme ich spoločné črty.

iScreen Aphasia

Aplikácia iScreen Aphasia je skríningová aplikácia pre diagnostikovanie rôznych typov afázie. Ťarcha identifikácie konkrétneho typu spočíva na administrátorovi, čo znamená, že jej

používanie vyžaduje detailné znalosti oblasti afaziológie. Avšak s MLSE má veľa spoločného čo sa týka typov požadovaných úkolov od pacienta - ako MLSE preveruje rôzne oblasti produkcie reči a porozumeniu hovoreného slova.



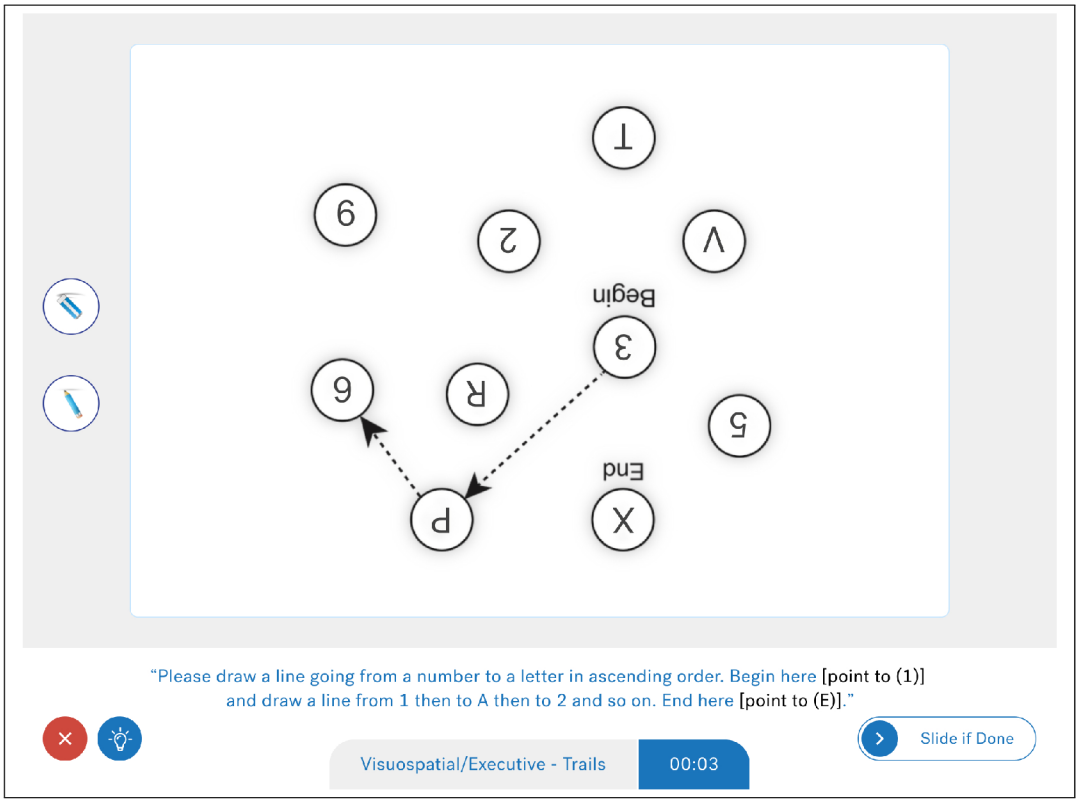
Obr. 2.7: iScreen Aphasia - ukážka obrazovky pre pacienta

Nasledovné črty tejto aplikácie nám prídu dôležité na zmienku:

- Pacient vykonáva podobné úkoly tým v MLSE. Avšak na rozdiel od MLSE sa väčšina položiek v subtestoch hodnotí nie v rámci viacerých domén, no jednoducho ako "chybne"/ "bez chyby"/ "bez odpovede".
- Hodnotenie pacientovho výkonu a samotné vykonávanie inštrukcií pacientom sa často striedajú. Tým pádom vznikajú "momenty hlucha", kedy pacient čaká, až administrátor dohodnotí jeho/jej výkon. Toto vyhodnocovanie počas testovania, ako sme si zmienili v podkapitole o MLSE, sa neodporúča práve kvôli diskomfortu pre pacienta.
- Aplikácia je určená len pre tablety, čo si vysvetľujeme aj potrebou používateľské rozhranie aplikácie niekedy zahaliť a naspäť odhaliť pacientovi - nie len počas vyhodnocovania výkonu pacienta, ale aj počas čítania inštrukcií administrátorom. Tieto inštrukcie si administrátor buď musí pamätať alebo si ich musí prečítať z okna nápovedy.

MoCA Duo

Hoci MoCA Duo (Montreal Cognitive Assessment) sa netýka afaziológie, vybrali sme si túto aplikáciu kvôli pozoruhodnej používateľskej skúsenosti.



Obr. 2.8: MoCA Duo - ukážka obrazovky počas testovania pacienta

- Z obrázku je pozoruhodné, že časť používateľského rozhrania je otočená o 180 stupňov. Takýmto spôsobom je docielené jasné oddelenie a "zatajenie"časti rozhrania pre pacienta a administrátora. Toto je z dôvodu, že pacient a administrátor počas využívania aplikácie sedia oproti sebe, podobne ako sa to deje pri papierovej verzií MLSE.
- Ďalším zaujímavým úkazom je riešenie nevyžiadaných interakcií od pacienta. Predpokladáme, že pri testovaní pacienta by sa mohlo stať, že by omylom zmenil subtest, napr. dotknutím sa nesprávneho tlačidla. Usudzujeme, že práve z tohoto dôvodu sa zmena momentálneho subtestu deje potiahnutím tlačidla v dolnom pravom rohu - akcia potiahnutia je omnoho menej pravdepodobná než klasický dotyk od pacienta.
- MoCA Duo je aplikácia pre tablety, ktorá si na rozdieľ od iScreen Aphasia nevyžaduje časté menenie polohy tabletu - z dôvodu unikátneho návrhu používateľského rozhrania je žiadúce mať tablet položený vodorovne na stole. Jediný prípad, kedy je vhodné tabletom hýbať je počas úloh vyžadujúcich odfotenie kresby pacienta.
- Z technického hľadiska je pozoruhodné, ako sa vyhodnotenie niektorých subtestov vykonáva automaticky. Pri iných, naopak, vyhodnotenie vykonáva administrátor, čo si vyžaduje spomínaný nežiadúci "moment hlucha"pre pacienta.

Spoločné črty

Hoci sú obe aplikácie od seba odlišné, k niektorým veciam sa stavajú rovnakým spôsobom.

- Azda najväčšiu podobnosť zdieľajú obe aplikácie z dôvodu, že sú to nástroje na prehrávanie testu a jeho vyhodnocovanie. I keď v prípade MoCA Duo sa vyhodnocovacia fáza menej prelína s testovacou, ani tu tieto 2 fázy nie sú kompletne oddelené.
- Nakoľko je pri vykonávaní akýchkoľvek lekárskych testov žiadúce archivovať výsledky, ponúkajú obe aplikácie túto funkcionalitu. Za týmto účelom sa v oboch aplikáciách vytvárajú profily pre pacientov. Pri zobrazení profilov je v oboch aplikáciách vidno históriu testovaní týchto pacientov. Takáto funkcionalita sa bez pochýb zíde aj pri MLSE, nakoľko je užitočné testovať progresiu ochorenia v priebehu rokov.
- V oboch aplikáciách spadá povinnosť správne podať inštrukcie na administrátora - opakujú ich buď z pamäti alebo ich čítajú. Nesprávne podanie inštrukcie môže mať nepatrné, no predsa značné následky na výkon pacienta. Pri tzv. intaktných pacientoch nie je problém pochopiť aj trochu vágnu alebo zle vyslovenú inštrukciu. Avšak pri pacientoch s ochorením môže znamenať zlé pochopenie inštrukcie rozdiel medzi správnou a nesprávnou diagnózou.

Posledným a rozhodujúcim črtám oboch aplikácií je, že sú pre tablety. Eliášová [8] vo svojej diplomovej práci poníma o príčinách pre rozšírenosť práve tabletových aplikácií u logopédov a o ich zastúpení v Českej republike - dokazuje, že tvoriť aplikácie práve pre túto platformu nie je márne. Zmieňuje, že tabletové aplikácie sa využívajú v ČR najmä s deťmi, avšak ich výhody ako zabudovaná kamera, nízka váha alebo podobnosť k papieru (kde prvotne vzniká veľa logopedických testov), ich robia ideálnym kandidátom pre omnoho širšie obecenstvo, než len deti. O tejto vhodnosti tabletov ako cieľovú platformu budeme aj my diskutovať v kapitole Implementácia.

2.4 Implementačné technológie

Na záver tejto kapitoly, po tom čo sme si utvorili prehľad ako o teste MLSE, tak aj o metodike návrhu používateľsky prívetivej aplikácie, si poďme predstaviť rôzne možnosti, ako z návrhu spraviť realitu. Implementačné technológie, ktoré si tu predstavíme, budú len malou podmnožinou všetkých možností, avšak ponúknu dobrý prehľad o možných prístupoch.

Vzhľadom na viackrát preukázaný trend využitia tabletov pre logopedické aplikácie (ktorý si pripomenieme opäť pri výbere tu zvolenej implementačné technológie), si ako prvé východisko stanovme nasledovné: Chceme, aby implementačná technológia ponúkala možnosť vytvárať aplikácie pre tablety. V čase písania sú dva dominantné operačné systémy pre tablety iOS a Android [3], takže sa zameriame na ne.

Ako prvé je vhodné zvážiť vývoj pomocou natívnych technológií. Natívne technológie dokážu vytvárať aplikácie len pre ich stanovenú platformu, čo na jednej strane poskytuje najväčší prístup k podliehajúcim technológiam zariadenia, no na druhej strane sa takto tvorené aplikácie nedá spustiť na iných platformách [4]. Natívne technológie pre iOS obnášajú využitie programovacích jazykov ako Swift alebo Objective-C, ako aj frameworkov ako UIKit [14]. Pre Android zahŕňa natívny vývoj využitie napr. programovacieho jazyka

Kotlin či IDE Android Studio [9].

Z dôvodu, že by sme museli vytvárať dva rozličné projekty pre obe platformy, sa poďme pozrieť na multiplatformové technológie – teda také, kde vieme ten istý zdrojový kód použiť pre obe platformy. Jednou možnosťou je framework QT s využitím jazyka C++ [6]. Ponúka vyvíjanie aplikácií nie len pre iOs a Android, avšak aj veľké množstvo ďalších platforiem vrátane desktopu. Jednou nevýhodou je však nemožnosť vytvárať webové aplikácie.

Prečo však spomínať web? Webové aplikácie sú v dnešnej dobe veľmi populárne, z dôvodov siahajúcich od trendov, jednotnosti využívania jazyka javascript, až po nástup business modelu Software as a Service [18]. Pre túto prácu je najviac relevantný fakt jednoduchá distribúcia – potenciálnym testerom stačí poslať odkaz na webstránku. Toto kompletne obchádza nutnosť využívania spoplatneného Apple Developer Program za účelom dostania aplikácie na tablet potenciálnych testerov skrz programy ako TestFlight [10].

Avšak webstránkami to nekončí – ďalšia výhoda využívania webových technológií je ich flexibilita pre tvorbu aplikácií, ktoré nebežia vo webovom prehliadači ale priamo na zariadení a tým pádom majú prístup k rozhraniam, ktoré pre webové prehliadače nie sú dostupné – dobrým príkladom je súborový systém. Jednou z technológií umožňujúcich zobrať zdrojový kód pre prehliadače a dostať ho von z prehliadača priamo na zariadenie je Capacitor.

Capacitor umožňuje vytvárať mobilné aplikácie (vrátane tých pre tablety) tak, že zoberie zdrojový kód určený pre webový prehliadač a spúšťa ho v prostredí nazývanom WebView. Hoci ide o aplikáciu, ktorú si používateľ nainštaluje priamo do zariadenia, jej obsah sa v skutočnosti zobrazuje cez webové technológie. Zároveň však, keďže ide o „skutočnú“ nainštalovanú aplikáciu, získavame prístup aj k natívnym API systému [2] — čím sa obchádza spomenuté obmedzenie webových prehliadačov. To nás ale opäť privádza k problému s distribúciou a k otázke: Naozaj potrebujeme natívny prístup, napríklad k súborovému systému? Nie je pre naše potreby postačujúce vytvoriť aplikáciu bežiacu len na webe? Túto otázku riešime v práci neskôr, avšak moderné webové prehliadače toho dnes dokážu skutočne veľa, napriek faktu, že nemajú prístup k natívnym API systému. Naprieč moderným browserom (Google Chrome, Safari, Firefox, Edge) existujú štandardizované rozhrania:

- Veľké množstvo dát je možné ukladať lokálne v prehliadači pomocou IndexedDB.c̃iteindexedDB - Prístup ku kamere a mikrofónu (získanie mediálneho streamu) zabezpečuje rozhranie getUserMedia z Media Capture and Streams APIc̃itesteamsAPI - Nahrávanie mediálnemu streamu je realizovateľné cez rozhranie MediaRecorder API, taktiež zo spomínaného API - Animácie je možné implementovať prostredníctvom Web Animations API.c̃iteanimationsAPI

Kapitola 3

Návrh

Cieľ tejto kapitoly je postupne analyzovať problém, špecifikovať ciele dosiahnuteľné aplikáciou a navrhnúť zodpovedajúce riešenie. Pri tomto procese sa budeme inšpirovať štruktúrou metodiky Goal Directed Design. Táto metodika, vysvetlená v kapitole Teória, si ako prvý krok pokladá fázu analýzy problémovej domény ako aj existujúcich riešení - čo sme taktiež v predošlej kapitole už spravili, čiastočne. V tejto kapitole preto pokračujeme ďalšími krokmi procesu návrhu aplikácie:

V prvej podkapitole si dokončíme analýzu tým, že použijeme poznatky z oblasti používateľskej skúsenosti na zhodnotenie papierovej verzie MLSE - ktoré aspekty narábania s týmto nástrojom považujeme za pozitívne a chceme ich skopírovať do digitálnej podoby, a naopak, ktoré aspekty sa dajú unikátne vylepšiť digitálnymi technológiami.

Po tomto nasleduje podkapitola odzrkadľujúca fázu modelovania a definície požiadaviek: Postupne si načrtneme požiadavky na náš systém z rôznych uhlov pohľadu. Pri stanovovaní požiadavok sa nebudeme ešte zamýšľať nad tým, ako konkrétne sa dajú splniť. Po ich stanovení si predstavíme, že produkt, ktorý ich splňuje, už existuje - a vytvoríme si imaginárne príbehy o používaní aplikácie.

V predposlednej kapitole sa budeme zaoberať otázkou, ako tieto scenáre dosiahnuť. Budeme navrhovať mockup používateľského rozhrania, ktorý ich umožní splniť čo najefektívnejšie. Prezentovaný mockup bude finálnou verziou mockupu s už dokončenou vizuálnou identitou - farbami, fontami a tvarmi. Tým pádom bude táto kapitola v sebe zahŕňať ciele Goal Directed Design-u nielen z fáze definície rámca, ale aj z fáze spresnenia.

V poslednej kapitole sa vrátime späť ku požiadavkám, a to technickým. Pomocou ER diagramu si načrtneme dátový model aplikácie, ako z hľadiska schémy uložených dát, tak aj z hľadiska dát opisujúcich celý test.

3.1 UX analýza papierovej verzie MLSE

Požiadavky na aplikáciu z produktového hľadiska sú vytvorenie takej aplikácie, ktorej užitočnosť a kvalita presahujú súčasný stav v dostatočnom merítku. Aplikácia musí byť dostatočne presvedčivá na to, aby sa logopédom ňou oplatilo nahradiť používanie súčasnej podoby testovania, teda testovanie za pomoci testovacej brožúry. Je dôležité mať na mysli,

že sú aspekty práce s testovacou brožúrou, ktoré jednoducho nie je možné digitálnou verziou poraziť. Tak napríklad, digitálna verzia si vyžaduje prístup k počítaču, elektrine a inému príslušenstvu. Cieľom je tým pádom vytvorenie takého produktu, ktorého klady a zápory v konečnom súčte poskytujú stále lepší používateľský zážitok.

Na začiatok je dôležité zmieniť výhody papierovej verzie MLSE a potrebu ich premietnuť aj do digitálnej verzie.

Výhody papierovej verzie a ich premietnutie do digitálnej verzie

- **Jednoduchosť:** Jedna z obrovských výhod testovacej brožúry je jej neskutočná jednoduchosť. Test MLSE bol schválne navrhnutý tak, aby jeho administráciu zvládol aj personál bez zdĺhavého zaúčacieho procesu. Z tohto dôvodu musí byť digitálna verzia MLSE taktiež dostatočne jednoduchá na používanie a ľahko naučiteľná.

- **Povedomosť:** Každý človek bez ohľadu na vek pozná pero a papier. Oproti tomu sú počítače relatívne novým technologickým výdobytkom. Nakoľko je predpoklad oboznámenia s technológiami u administrátorov MLSE vysoký, treba brať zreteľ aj na pacienta. Veková skupina ľudí, u ktorých sa vykonáva MLSE, má priemernú dolnú hranicu približne 55 rokov. Navyše, MLSE sa vykonáva nielen u intaktnej populácie pre účely skríningu, ale taktiež u pacientov, pri ktorých sa už vie o výskyte neurodegeneratívnych ochorení. Tým pádom musí byť test v maximálnej možnej miere pochopiteľný aj pre takéto osoby.

- **Archivácia:** V praxi nie je nezvyčajné, že pre toho istého pacienta sa test MLSE vykonáva viackrát, napríklad každý rok. Toto opakovanie testovania môže mať rôzne dôvody, napríklad monitorovanie stavu pacienta. Archivovanie hárkov papiera nie je nič zložité a preto musí byť digitálna verzia v tomto ohľade rovnako ak nie viac nápomocná. Pri archivácii je nutné zmieniť, čo má význam archivovať:

  - Minimálne je dobré archivovať výsledok hodnotenia celého testu, teda akumulované body v spomínaných 6 doménach.
  - Taktiež sa oplatí archivovať konkrétne skóre udelené v jednotlivých subtestoch, spolu s chybami, ktoré nastali. Takýmto spôsobom sa dá identifikovať, či sa chyby v priebehu rokov u pacienta stupňujú napríklad len v tom istom subteste (čo je z klinického hľadiska veľmi relevantný fakt), alebo či nastávajú aj v iných subtestoch.
  - V neposlednom rade môže k ešte obšírnejšiemu pohľadu na pacienta dopomôcť archivácia audio alebo videozáznamu z testovania. Pre čo najviac holistický pohľad na pacienta, ktorý sa ťažko zachytáva jednoduchými číslami alebo poznámkami na papier, je taktiež prínosné mať možnosť si spätne prehrať svedectvo o tom, ako pacient rozpráva, ako sa tvári a podobne.

  Je očividné, že jednoduchosť a spoľahlivosť archivácie týchto údajov musí byť ďalším silným pilierom digitálnej verzie MLSE.

Nevýhody papierovej verzie MLSE a ako ich vie digitálna verzia adresovať

- **Aritmetické chyby:** Pri súčte bodov v jednotlivých doménach môže nastať aritmetická chyba. Toto môže mať, pochopiteľne, katastrofálne následky. Ak by sa spočítava-

nie týchto bodov digitálne automatizovalo a patrične otestovalo, mal by administrátor MLSE o starosť menej.

- **Nešikovnosť nahrávania a rozptýlenie pacienta:**
  - Počas testovania pacienta sa administrátor musí venovať 2 zariadeniam zároveň: testovaciemu hárku a audio/video záznamníku, najčastejšie mobilu. V dôsledku takéhoto multi-taskingu sa môže zabudnúť na zahájenie nahrávania.
  - Problém úzko spätý s nahrávaním je taktiež rozptýlenie pacienta. Keď je na pacienta namierený mobil počas nahrávania videa, vie to pôsobiť rušivo. Z praxe je známe, že niektorí pacienti mávajú tendenciu hľadieť na kameru namiesto toho, aby venovali svoju plnú pozornosť hárku papiera.
  - Ďalší problém spätý s nahrávaním pacienta je fakt, že nastáva rozdelenie celkového výstupu z testovania. Výstup z testovania obsahuje položky, ktoré sa archivujú: výsledok hodnotenia a hárky, na ktorých sa skórovali položky jednotlivých subtestov, ktoré sú v papierovej podobe. Audio/video nahrávka je uložená inou technológiou, napríklad v mobilnom zariadení, na počítači, externom disku či cloudovom úložisku.

  Digitálna aplikácia by vedela tieto nedostatky procesu nahrávania mitigovať prirodzeným spôsobom. Na nahrávanie by sa dalo použiť vstavanú kameru zariadenia, ktorým sa zároveň testuje. Nahrávanie by sa mohlo diať automaticky počas testovania, čím by mal administrátor opäť menej úkonov na vykonanie. Ďalšou výhodou takéhoto prístupu je, že v prípade tabletu je kamera priamo nad displejom a nepôsobí rušivo. Poslednou výhodou využitia digitálnych technológií je možnosť všetky dáta agregovať na jednom mieste v pamäti zariadenia. S týmito dátami by sa dalo narábať ako s celkom a ukladať ich zjednotene.

- **Chyba výslovnosti u administrátora:** Počas toho, ako administrátor nahlas číta inštrukcie pacientovi, sa môže vyskytnúť zakoktanie, chyba výslovnosti, nedostatočná hlasitosť a pod. Aj takýto malý preklep vie ovplyvniť kvalitu testovania, a preto je žiaduce, aby sa inštrukcie podávali vždy s čo najlepšou artikuláciou. Tento problém sa dá opäť vyriešiť vytvorením digitálnej verzie MLSE, v ktorej by sa inštrukcie čítali nie administrátorom, ale namiesto toho by zazneli z predom nahratej nahrávky. Toto odbremeňuje administrátora od ďalšieho úkonu.

3.2 Modelovanie požiadaviek

Používateľské požiadavky

Z hľadiska používateľských požiadaviek treba najskôr stanoviť, kto je používateľom aplikácie. Digitálna verzia MLSE by mala v skutku dvoch používateľov, a to ako administrátora, tak i pacienta. Primárne so systémom interaguje administrátor, čo znamená, že zahajuje testovanie, určuje spúšťanie subtestov a vyhodnocuje výkon pacienta. Pacient však nesmie byť opomenutý, hoci jeho interakcia so systémom je viac pasívna.

Berúc do úvahy produktové požiadavky a súčasnú podobu testovania, sme vyhodnotili nasledovné požiadavky pre administrátora:

- **Požiadavky späté s testovaním pacienta:**

- Vytvorenie profilu pacienta:
  Táto tvorba profilu sa mapuje priamo na nultý hárok testovacej brožúry.
- Zahájenie testovania:
  Pre existujúceho pacienta sa má vytvoriť nové sedenie, počas ktorého sa pacienta testuje a následne sa vyhodnocuje jeho výkon.
- Navigovanie v rámci subtestov a ich častí:
  Administrátor musí mať kontrolu nad tým, ktorý subtest sa momentálne prevádza.
  Administrátor potrebuje vedieť stanoviť, ktorá časť je momentálne zobrazená.
- Spustenie časti subtestu a opätovné prehratie inštrukcie:
  Spustením sa rozumie to, že sa prečíta inštrukcia pre danú časť a zobrazia sa prípadné obrázky späté s touto inštrukciou. Je potrebné mať možnosť znova prehrať inštrukciu.
- Automatické zahájenie nahrávania a jeho pozastavenie:
  Počas testovania pacienta sa musí nahrávanie automaticky spustiť. Nahrávanie sa však musí dať pozastaviť a následne znova spustiť, napríklad pre prípad, keď pacient potrebuje dočasne odísť z miestnosti.

- **Požiadavky späté s vyhodnocovaním výkonu pacienta:**
  - Prehratie nahrávky pacienta pre jednotlivé subtesty:
    Po odchode pacienta sa začína vyhodnocovanie výkonu pacienta, pri ktorom je nevyhnutné si vedieť nahrávku spustiť pre pripomenutie si výkonu pacienta. Toto prehrávanie a s tým späté skórovanie sa musí dať previesť v ľubovoľný čas, nie len tesne po odchode pacienta.
  - Skórovanie v rámci domén subtestu:
    Administrátor potrebuje mať používateľské rozhranie, v ktorom vie na základe výkonu pacienta zaznačovať chyby a prideľovať body do jednotlivých domén pripadajúcich na subtest. Kľúčové je poskytnúť také riešenie, ktoré poskytuje efektívne zaznačovanie – naprieč všetkým 11 subtestom je naozaj veľké množstvo okienok na zaznačenie „X“ – ako spraviť tento zážitok zaznačovania čo najmenej nepríjemný?

- **Požiadavky späté s archiváciou testovania:**
  - Možnosť pozrieť si a editovať informácie o pacientovi:
    Sú to informácie o pacientovi vyplnené pri zostavovaní profilu pacienta.
  - Možnosť zobraziť si prehľad vykonaných testovaní pre vybraného pacienta:
    Dátumy jednotlivých výkonov testovania, čiže jednotlivých sedení.
  - Možnosť zobraziť si archivované dáta pre vybrané sedenie:
    Zobrazenie nahrávky, skórovania a celkového výsledku testovania.
  - Možnosť editácie archivovaných dát:
    Editovanie skóre v jednotlivých doménach pre jednotlivé položky subtestov.
  - Možnosť exportovať všetky dáta o pacientovi:
    To znamená vedieť si stiahnuť všetky nahrávky a výsledky testovania pre vybratého pacienta. Výsledky by mali byť exportovateľné do tlačiteľného formátu, ako pdf.

  - Možnosť vymazať všetky dáta o pacientovi a vymazať profil pacienta.

- **Požiadavky pacienta:**
  - Oboznámenie o nahrávaní:
    Pacient potrebuje byť informovaný o tom, že nahrávanie prebieha.
  - Súhlas alebo nesúhlas s nahrávaním videa:
    Pacient potrebuje vedieť administrátora požiadať o to, aby sa zhotovovanie videozáznamu nedialo. Tu je nutné zdôrazniť, že zhotovovanie audio záznamu je nevyhnutné pre skórovanie, a teda musí sa vykonávať za každých okolností.
  - Prehľad o dianí v rámci subtestu:
    Počas testovania s papierovou verziou sa v niektorých subtestoch používa obrázky. Pacientovi sú postupne predložené, a takéto postupné odhalenie pomáha pacientovi chápať, čo sa deje. Zachovať takéto postupné zmenenie výzoru obrázku (viditeľný / neviditeľný) je potrebné aj v našej aplikácií, čo naznačuje potrebu funkcionality animácií.

Funkčné požiadavky aplikácie

Nadväzujúc na potreby používateľov, zmienených v predošlej sekcii, boli identifikované nasledovné požiadavky na systém ako taký:

- Nahrávanie audio a video záznamu
- Ukladanie stoviek MB dát tohoto záznamu
- Export audio a video záznamov
- Prehrávanie záznamu
- Zaznamenávanie odpovedí
- Vytvorenie a export pdf súboru s výsledkami
- Úložisko a spätné pozretie predošlých testov
- Profily pacientov
- Animovanie výzoru obrázkov

Používateľské scenáre

Pred návrhom používateľského rozhrania je vhodné predstaviť si typické scenáre, ktorými si používatelia pri aplikácií budú prechádzať. Pre našu aplikáciu sú to nasledovné:

- Logopédovi príde do praxe nový pacient, ktorému logopéd vytvorí nový používateľský profil. Následne pacienta otestuje a ten po skončení testu odchádza. Logopéd si nechal aplikáciu otvorenú a tak okamžite začína test vyhodnocovať.
- Logopéd má nabitý deň, čaká ho testovanie 4 pacientov tesne po sebe. Do praxe príde prvý, pacientka ktorá MLSE už raz u tohoto logopéda absolvovala. Logopéd si otvorí aplikáciu, nájde si meno pacientky v zozname pacientov a začne nové testovanie. Pacientka po testovaní odíde, no hneď v zápätí príde ďalší pacient. Logopéd preto

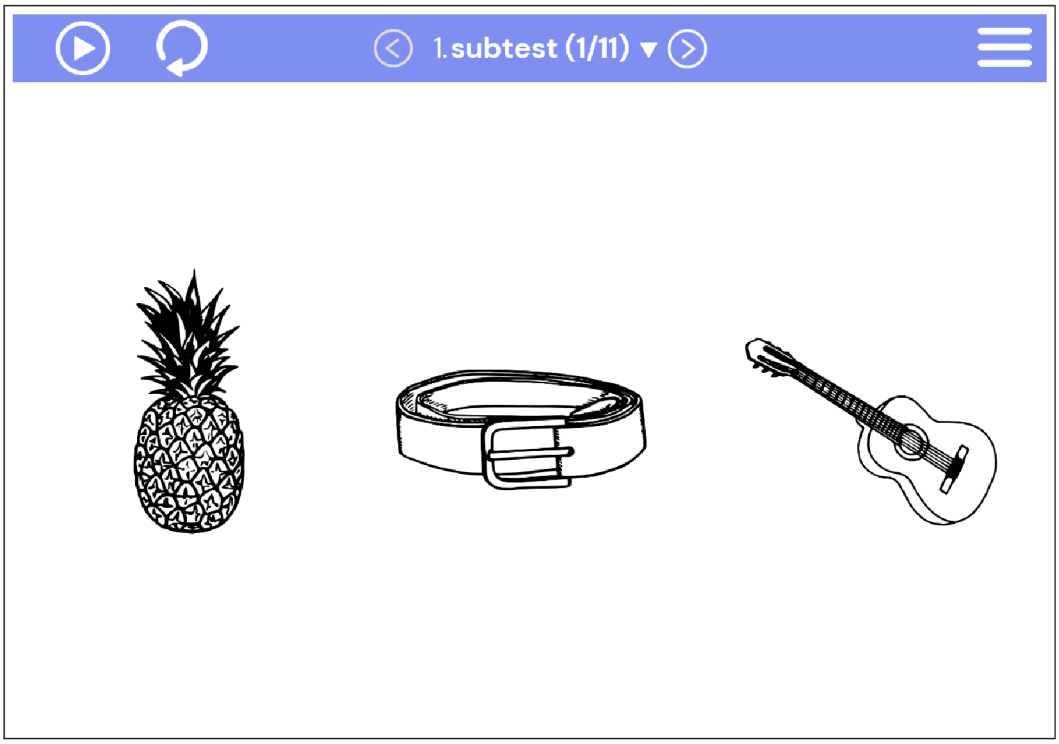
vyhodnotenie nechá na neskôr - vráti sa na domovskú obrazovku, otvorí / vytvorí nový profil pacienta a znova testuje. Po otestovaní posledného pacienta na domovskej stránke vidí, že má u každého z týchto 4 pacientov 1 nevyhodnotený subtest. Vyhodnocuje ich rad za radom.

- Logopéd si nechal vyhodnocovanie testov na koniec dňa. Pri vyhodnocovaní posledného si uvedomil, že v predošlom spravil chybu. Okamžite sa vráti na domovskú stránku, nájde si meno pacienta, na ktorého chybne opravený test spadá a zobrazí si vyhodnotenie. Z obrazovky vyhodnotenia sa dostane na obrazovku vyhodnocovania subtestu 4, kde opraví chybu. Taktiež sa prepne na subtest 11, v ktorom tiež opraví chybu. Teraz, keď opravil všetky chyby, sa vráti na domovskú obrazovku a prepne sa na vyhodnocovanie posledného testu, ktoré prerušil.

- Do praxe logopéda príde 3. rok po sebe pacientka, u ktorej si logopéd pamätá, že mala potiaže s niektorými subtestami počas predošlých 2 rokov. Pred začatím testovania si zobrazí profil tejto pacientky a pozrie sa na výsledky testu spred 2 rokov. Následne sa vráti späť na profil pacientky a pozrie si výsledky testu z predošlého roku. Po tomto má vytvorenú predstavu, pri ktorých subtestoch musí dávať obzvlášť pozor, nakoľko pri nich pacientka robila predošlé roky chyby. Začína nový test.

3.3 Používateľské rozhranie

Berúc do úvahy používateľské požiadavky a taktiež možné používateľské scenáre, si najskôr zobrazíme grafické návrhy pre najdôležitejšie časti používateľského rozhrania. Následne si každú z navrhnutých obrazoviek podrobne opíšeme – aké požiadavky zmienené v kapitolách vyššie spĺňa a ako ich spĺňa. Zakaždým budú jednotlivo vysvetlené komponenty GUI pre danú obrazovku.



Obr. 3.1: Obrazovka testovania pacienta

Obrazovka Testovania Pacienta

Na tejto obrazovke, znázornenej na obrázku 3.1 sa zobrazuje momentálne vykonávaný subtest. Rieši najdôležitejšiu časť celej aplikácie, čo je administrácia jednotlivých subtestov pacientovi a ich správne pochopenie pacientom. Stredobodom pozornosti pre pacienta sa tu stáva celá časť obrazovky pod hornou lištou, kde sa zobrazuje grafický obsah subtestov - obrázky. Oproti tomu sa administrátor však sústreďuje práve na hornú lištu, kde mu nasledovné GUI komponenty umožňujú splňať väčšinu cieľov spätých s testovaním:

- **Tlačidlá na prehratie a zopakovanie inštrukcie**: Tieto 2 tlačidlá v hornom ľavom rohu splňajú požiadavok prehrávať momentálnu inštrukciu pacientovi, prípadne ju zopakovať. Treba podotknúť absenciu akéhokoľvek umožnenia vrátiť sa k predošlej inštrukcií – toto rozhodnutie vychádzalo z faktu, že v papierovej verzií, akonáhle pacient splní inštrukciu, sa k nej nesmie vracať; iba prvotná odpoveď je platná.

  Táto komponenta si prešla istými zmenami po tom, ako bola otestovaná funkčná implementácia obrazovky testovania. Avšak pre terajší návrh, zhotovený pred testovaním v praxi, nastáva otázka – ak neumožňujeme vracať sa k iným inštrukciám ako terajšej (zopakovanie), ako sa má zamedziť napr. omylným kliknutiam na tlačidlo prehrávania? Odpoveď je, že pôvodne bolo toto tlačidlo koncipované tak, aby sa s ním nedalo na istý čas interagovať po jeho využití. Aplikácia mala tým pádom stiesňujúci charakter, administrátor musel zakaždým chvíľu počkať, až mohol prejsť na ďalšiu inštrukciu. Hoci sa tento dizajn zdá protichodný s filozofiou dať používateľovi nefrustrujúci zážitok, pri ktorom má pocit kontroly, aj tak sme prvotne uprednostnili menej

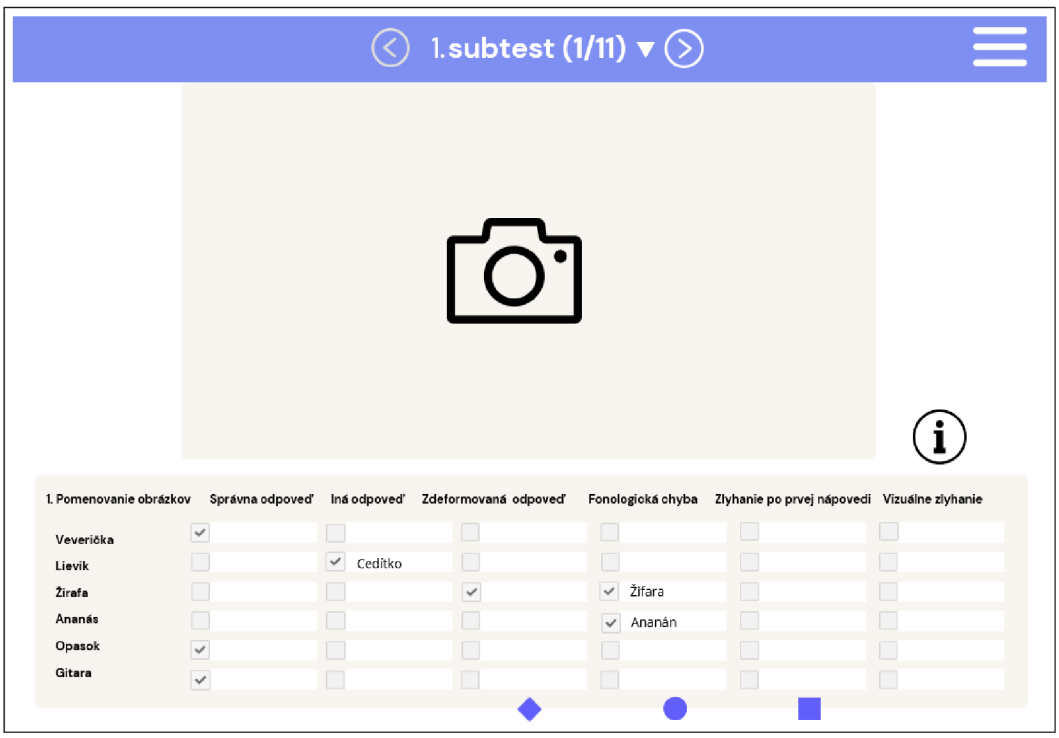
kontroly. Dôvod bol ten, že počas testovania papierovou verziou sa vždy postupuje vo veľmi lineárnom štýle - inštrukcia po inštrukcií, subtest po subteste.

- **Navigácia v subtestoch**: V strede hornej lišty sa zobrazuje, aký subtest je momentálne vykonávaný. Sú tu aj tlačidlá pre navigáciu medzi jednotlivými subtestami (šípky doľava a doprava), no pre nelineárnu navigáciu medzi subtestami tu je aj indikátor, že sa jedná o dropdown.

  Podobne ako pri predošlom ovládacom prvku, aj tento si prešiel iteráciou – z počiatku sme opäť stavali na zaužívanom lineárnom postupe pri testovaní a až neskôr sme pridali dropdown. Otvorením dropdownu má administrátor na výber, na ktorý subtest chce preskočiť. Toto lepšie spĺňa jeden z požadovaných UX princípov: Aby vedel používateľ vykonávať úkony efektívne. Čiže ak administrátor predsalen uzná za vhodné preskočiť hneď na posledný subtest, nemusí 11 krát klikať šipku doprava.

- **Menu**: Interakciou s posledným tlačidlom na hornej lište sa zobrazí okno umožňujúce sa vrátiť na domovskú obrazovku 3.4. V tomto menu sa taktiež zobrazí možnosť ukázať náhľad audio/video nahrávania - tým pádom si vie administrátor overiť, či sa nahrávanie náhodou nepokazilo. Treba podotknúť, že nahrávanie by sa malo dať počas testovania aj pozastaviť a znova spustiť. Tým pádom sa spĺňa požiadavok, aby sa administrátor a pacient necítili pod časovým nátlakom – dojmom, že testovanie musia čo najrýchlejšie dokončiť.

Obrazovka Vyhodnocovania Výkonu Pacienta



Obr. 3.2: Obrazovka vyhodnocovania testu

Cieľom tejto obrazovky, zobrazenej na obrázku 3.2, je administrátorovi čo najviac uľahčiť vyhodnocovanie výkonu pacienta. Vychádzajúc z potrieb spätých s týmto cieľom sa na tejto obrazovke nachádzajú 2 hlavné GUI komponenty:

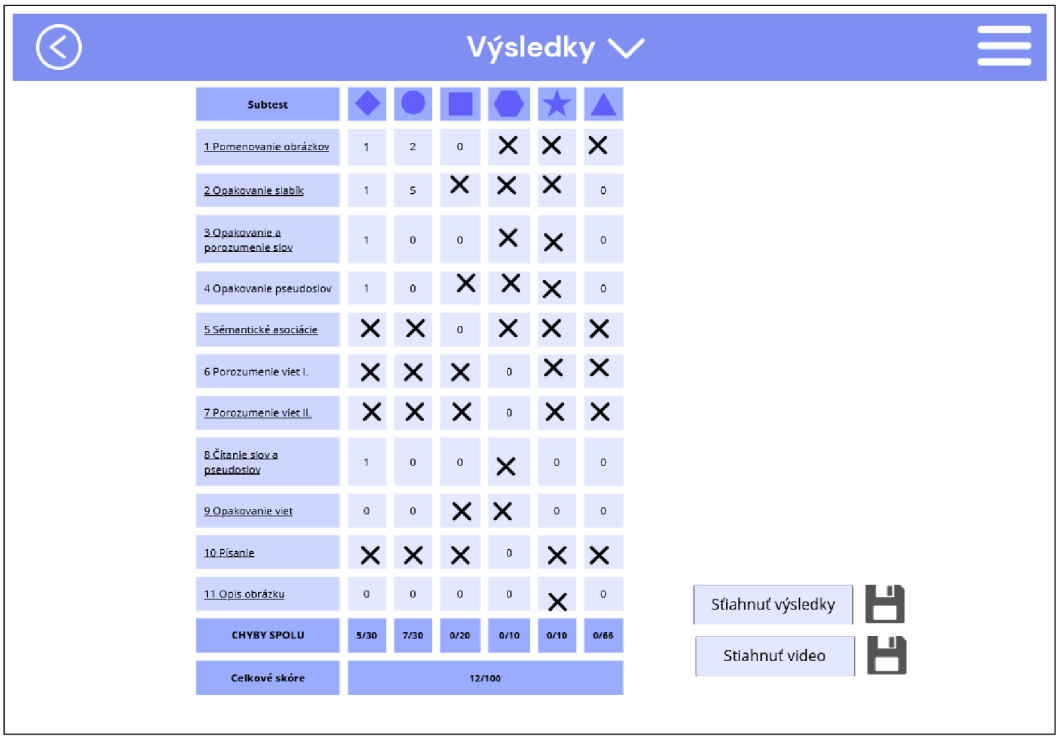
- **Zobrazenie nahrávky**: Za účelom pripomenutia si výkonu pacienta sa v strede obrazovky nachádza rozhranie pre spustenie nahrávky. V tomto rozhraní sa zobrazuje vždy len časť nahrávky súvisiaca s momentálne vyhodnocovaným subtestom – na ktorom subteste sa momentálne nachádzame opäť vidno na hornej lište.
  Zobrazením vždy len nahrávky z jedného subtestu (na rozdiel od celej nahrávky pre všetky subtesty) sa dosahuje 2 ciele: Jednak sa dá rýchlejšie zobraziť a vyhodnotiť výkon v rámci konkrétneho subtestu a druhak takéto rozdelenie intuitívne nadväzuje na rozhranie obrazovky testovania, v ktorom sa taktiež vždy zobrazuje len 1 subtest.

- Pod videom sa nachádza rozhranie, v ktorom administrátor skóruje výkon pacienta - udeľuje body pre každú položku do domén. Sú tu však len zaškrtávacie políčka, a to z toho dôvodu, že pridelenie bodov do domény ma binárny charakter. Buď je pridelené 0 bodov alebo 1, prípadne sú subtesty v ktorých sa prideľuje 0/2, 0/3 alebo 0/4.

Na tejto obrazovke je dôležité zmieniť informačné tlačidlo napravo od zobrazenia nahrávky. Toto tlačidlo otvorí okno, v ktorom sa zobrazí nápoveda pre administrátora, pre prípad, že nevie ako ohodnotiť nejakú chybu. Taktiež sú pod každou doménou geometrické symboly - tieto korešpondujú s označením domén v pôvodnej papierovej verzii MLSE a pomáhajú administrátorovi si jednoduchšie vytvoriť súvislosť medzi digitálnou a papierovou verziou, čím sa zlepšuje naučiteľnosť.

Oproti tomuto návrhu pribudlo vo finálnej verzií aplikácie pod zobrazením nahrávky aj tlačidlo „Uložiť“. Toto má umožniť administrátorovi uložiť nahrávku zobrazovaného subtestu do zariadenia, či už ako audio alebo video súbor. Bez tejto funkcionality by nahrávky boli vždy späté len s aplikáciou. Toto by nesmierne obmedzovalo slobodu administrátora narábať s dátami aplikácie podľa jeho osobných potrieb.

Obrazovka Vysledku Testu



Obr. 3.3: Obrazovka Výsledkov

Na obrazovke 3.3 vidno celkové hodnotenie testu. Grafický dizajn tejto obrazovky úzko súvisí s dizajnom celkového hodnotenia pri papierovej verzií. Obrazovka sa zobrazí po vyhodnotení všetkých subtestov, poskytujúc administrátorovi rýchly prehľad nad výkonom pacienta.

To, že názvy jednotlivých subtestov sú podčiarknuté, má naznačovať, že sa jedná o hyperlinky - kliknutím na názov subtestu sa zobrazí obrazovka vyhodnocovania subtestu pre vybraný subtest. Toto rieši potrebu administrátora vedieť si detailne pripomenúť pre každý subtest, aké konkrétne chyby pacient vykonal.

Taktiež sa tu nachádzajú tlačidlá pre export pdf týchto výsledkov – účel tohoto je vedieť dostať sumárne informácie o výkone pacienta von z aplikácie. Naopak, pre export nahrávky akéhokoľvek subtestu je nutné sa najskôr prekliknúť na daný subtest a tam stlačiť tlačidlo „Uložiť“.

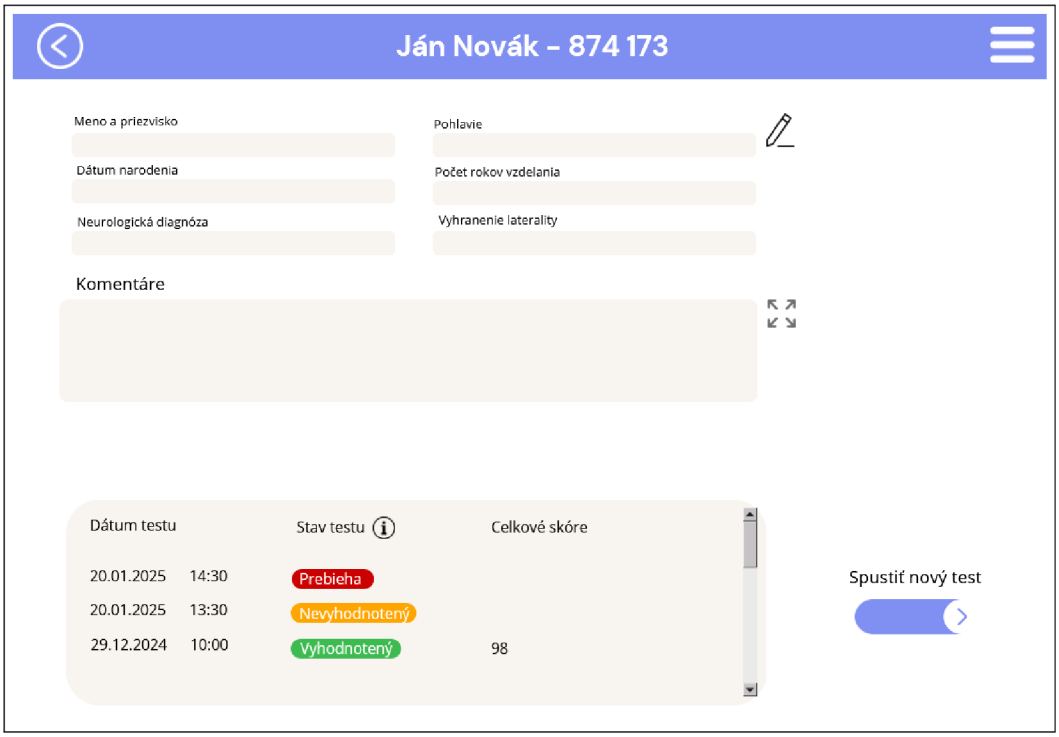
Domovská Obrazovka

Obr. 3.4: Obrazovka Zoznamu Pacientov

Na obrázku 3.4 sa nachádza domovská obrazovka aplikácie, v ktorej má administrátor prehľad o všetkých pacientoch a pri každom vidí, či má daný pacient nejaké subtesty, ktoré je potrebné vyhodnotiť. I keď má testovanie pacienta prebiehať počas jedného sedenia, môžu sa stať rôzne okolnosti, ktoré spôsobia, že administrátor bude musieť (chcene alebo nechcene) aplikáciu zatvoriť a znova otvoriť počas testovania pacienta, napríklad v dôsledku vybitia batérie zariadenia. Pacienti, ktorých testy boli uzatvorené počas testovania pacienta, majú z tohto dôvodu výstražné červené označenie s počtom prebiehajúcich subtestov.

Naopak, ak už testovanie je dokončené, no administrátor sa z rôznych dôvodov rozhodne nevyhodnotiť test okamžite, ostane tomuto pacientovi na domovskej obrazovke žltý varovný štítok, pripomínajúci túto skutočnosť. Tento štítok, narozdiel od výstražného, nemá administrátora popohnať aby okamžite test vyhodnotil. Totižto, zatiaľ čo testovanie je žiadúce ukončiť čo najskôr (počas prítomnosti pacienta), s vyhodnocovaním sa môže počkať, no nemalo by sa naň zabudnúť.

Karta Pacienta



Obr. 3.5: Obrazovka Karty Pacienta

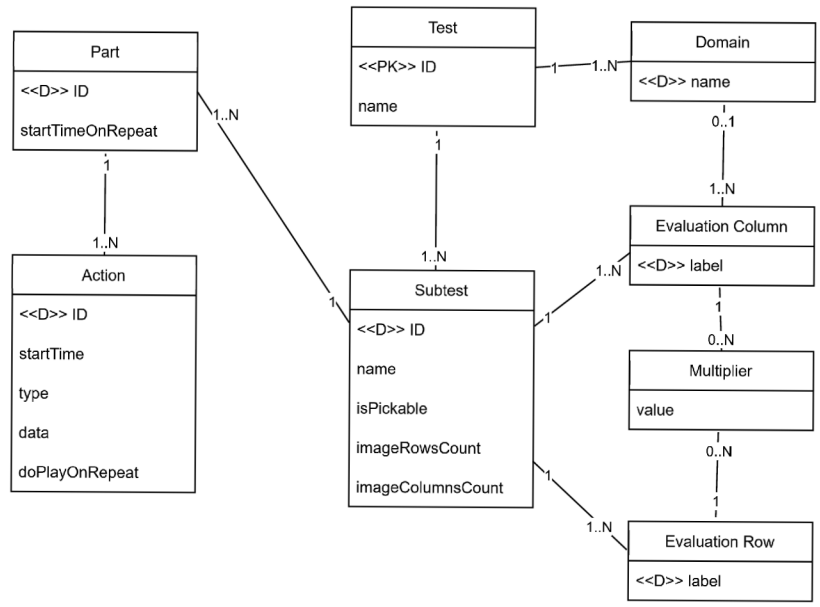
Táto obrazovka 3.5 sa zobrazí po tom, čo sa na domovskej obrazovke buď vyberie existujúceho pacienta alebo sa vytvorí nového pacienta.

V karte pacienta sa zobrazuje okrem informácií o pacientovi (ktoré je možné upravovať) aj históriu jeho testov. Vybratím hociktorého z týchto testov sa zobrazí obrazovka výsledkov z obrázku 3.2, v ktorej je možné sa prekliknúť na detail hociktorého subtestu a znova si pozrieť nahrávku pacienta a ako bol jeho výkon vyhodnotený.

Cieľom tejto obrazovky je vyhovieť potrebe administrátora mať prehľad o výkone pacienta naprieč viacerým sedeniam. Klesajúce hodnoty skóre totiž naznačujú progresiu patológie. Taktiež sa tu objavujú štítky z predošlej obrazovky, avšak tentoraz už vidno pre každé sedenie jednotlivo, či si vyžaduje pozornosť administrátora a ak áno, akú.

3.4 Dátový model

Na obrázku 3.6 je znázornený dátový model formou ER diagramu, zobrazujúci vzťahy medzi dátami týkajúcimi sa testovacej a vyhodnocovacej časti aplikácie. Sú to dáta definujúce obsah jednotlivých subtestov ako inštrukcie či obrázky, ako aj dáta týkajúce sa vyhodnocovacej časti – čiže ako vyzerá obrazovka vyhodnotenia pre daný subtest. Tu sa definuje názvy riadkov a stĺpcov, aké domény sú na stĺpce naviazané, či aké majú bodové váhy.



Obr. 3.6: Dátový model testu a vyhodnotenia

Kapitola 4

Implementácia

So špecifikáciou vytvorenou v predošlej kapitole sme teraz schopní systém implementovať do reality. Keď sa bližšie pozrieme na tieto požiadavky, zistíme, že hoci je aplikácia špecificky mienená na digitalizáciu konkrétneho testu MLSEsk, ak je vytvorená správne, malo by byť jednoduché pomocou nej administrovať a vyhodnocovať rôzne typy testov. Inými slovami, aplikácia by mala mať povahu **všeobecného prehrávača a vyhodnocovacieho nástroja** rôznych testov. Konkrétne zo špecifikácie vyplýva, že by sme mali byť schopní toto spraviť pre akýkoľvek test, ktorý obsahuje nasledujúce črty:

- zobrazovanie obrázkov,
- prehrávanie zvuku,
- funkcionalitu výberu obrázka spomedzi množiny obrázkov,
- fotenie kamerou,
- zaznamenávanie slovnej odpovede pacienta,
- zaznamenávanie videa,
- vyhodnocovanie v rámci určitého počtu domén.

Tým pádom, jeden zo spôsobov, ako verifikovať úspešnosť aplikácie, je ten, že sa pokúsime vytvoriť iný test než len MLSEsk. V podkapitole Testovanie sa venujeme presne tejto problematike, ako aj testovaniu samotnej používateľskej skúsenosti. V podkapitolách, ktoré testovaniu predchádzajú, sa venujeme postupne tomu, ako konkrétne implementujeme takýto systém.

4.1 Výber cieľových zariadení

Pred tým, ako sme sa mohli začať zamýšľať nad výberom implementačnej technológie, sme si najskôr museli ozrejmiť, na aké zariadenia našou aplikáciou cielime.

Pri zamýšľaní sa nad používateľskými požiadavkami nám bolo zrejmé, že počas testovania pacienta má aplikácia 2 používateľov: testovanú osobu a administrátora testu. Zároveň sme vychádzali zo zaužívaných postupov z logopedickej praxe, v ktorej pacient a logopéd sedia oproti sebe počas testovania - papierová verzia MLSE je koncipovaná presne s týmto

požiadavkám, čo sa odráža na dizajne testovacej brožúry, v ktorej je časť textu mienená pre logopéda vytlačená otočená o 180 stupňov. A tak nastala otázka: Ako vytvoriť aplikáciu, ktorá má 2 rozhrania: Jedno, ktoré vidí len testovaný a druhé, ktoré vidí len administrátor? Obaja musia vedieť interagovať s testom, no ako toto dosiahnuť, keď sedia oproti sebe?

Náš prvotný návrh bol vytvoriť aplikáciu naprieč 2 zariadeniam: Monitoru (alebo tabletu), ktorý je otočený k pacientovi a počítaču, ktorého monitor je otočený k logopédovi a ktorým sa ovláda test. Takýto prístup sa ukázal ako príliš náročný a tak sme sa rozhodli upustiť od zaužívanej praktiky sedenia oproti sebe: Dovolili sme, aby logopéd a pacient sedeli vedľa seba počas testovania.

S týmto odbremením nám prišlo logické pokračovať v trende využívania tabletov: Majú v sebe zabudovanú kameru a mikrofón, svojou formou pripomínajú papier, čo môže byť viac intuitívne pre staršie generácie a vďaka dotykovému displeju umožňujú prirodzenejší štýl interakcie s obsahom, než počítačová myška. Jednoznačne sme sa už tu zamýšľali ako nad personou pacienta, tak aj personou administrátora.

Zároveň bolo potrebné zohľadniť skutočnosť, že nie každý logopéd má tablet. Preto bolo myslené aj na desktopový počítač alebo laptop ako cieľové zariadenie.

4.2 Výber implementačnej technológie

V kapitole Teória sme si predstavili niektoré z možností, ako vytvoriť multi-platformové aplikácie. Pri výbere nami zvolenej technológie sme zohľadňovali nasledovné subjektívne požiadavky:

- Chceli sme, aby sa dalo aplikáciu ľahko dostať na zariadenia potenciálnych testerov. Tento bod bol veľmi dôležitý, nakoľko naši testeri mohli byť ľudia netechnickej povahy, ktorým by bolo nutné vedieť aplikáciu spojazdniť ideálne i len pomocou videokonferenčných technológií.
- Chceli sme, aby mala aplikácia čo najjednotnejší základ kódu, t.j. aby sme museli písať čo najmenej platformovo-špecifického kódu. Toto by hlavne šetrilo úsilie, ktoré by sa mohlo následne vyžiť pre zveľaďovanie používateľskej skúsenosti.
- Chceli sme ťažiť z predošlých znalostí s konkrétnymi implementačnými technológiami, nadobudnutými zo štúdia doposiaľ. Vychádzajúc z predošlej kapitoly sme chceli, aby implementačná technológia umožňovala vytvárať aplikácie pre tablety aj desktopy.

Berúc tieto požiadavky do úvahy sme sa napokon rozhodli ísť cestou webových technológií. Avšak tu sa otvárala nová otázka: Máme si zvoliť prístup čisto webových aplikácií alebo takých, ktoré využívajú webové technológie, no bežia na zariadení samotnom? Spočiatku sme preskúmavali druhú možnosť, nakoľko sme vedeli, že prístup k natívnym rozhraniam zariadenia, ako je kamera, je prvoradý a potrebujeme veľkú flexibilitu pri ich využívaní. Čo nás však odradilo od tohto prístupu bol fakt, že takto vytvorenú aplikáciu by si potenciálni testeri museli inštalovať - čo samo o sebe nie je až tak náročný proces. Avšak manažovanie častých aktualizácií, ako aj nutnosť zaplatiť 100€ pre Apple Developer Account, nás prinútili podstúpiť riziko:

Rozhodli sme sa použiť menej flexibilný prístup čisto webovej aplikácie. Výhoda bola, že pri každom zadaní adresy webstránky by sa používateľovi hneď načítala najaktuálnejšia verzia, t.j. distribúcia aplikácie by bola veľmi jednoduchá. Riziko spočívalo v tom, že sme mohli naraziť na prípad, kedy by napriek moderným webovým rozhraniam nebolo možné implementovať nejakú funkcionalitu. Na takýto okrajový prípad sme skutočne narazili, avšak tomuto sa venujeme v podkapitole Formatívne Testovanie.

Pri prvotnom zhodnotení moderných webových rozhraní sme zhodnotili, že všetky naše funkčné požiadavky sú realizovateľné aj napriek tomu, že sme vytvárali progresívnu webovú aplikáciu:

- Ukladanie dát (vrátane audio/video nahrávky) by sa dalo riešiť buď vytvorením webserveru, s ktorým sa komunikuje napr. prostredníctvom RESP API alebo ukladaním dát lokálne pomocou databázy IndexedDB.
- Prístup k hardvéru pre získanie streamu audia a videa je možné vďaka Media Capture and Streams API.
- Nahrávanie audio / video streamu do súborov by sa dalo riešiť pomocou rozhrania MediaStream Recording API.
- Prehrávanie audio / video súborov sa dá zaistiť pomocou HTML video alebo audio elementu, resp. pomocou zásuvných modulov stavajúcich na týchto elementoch.

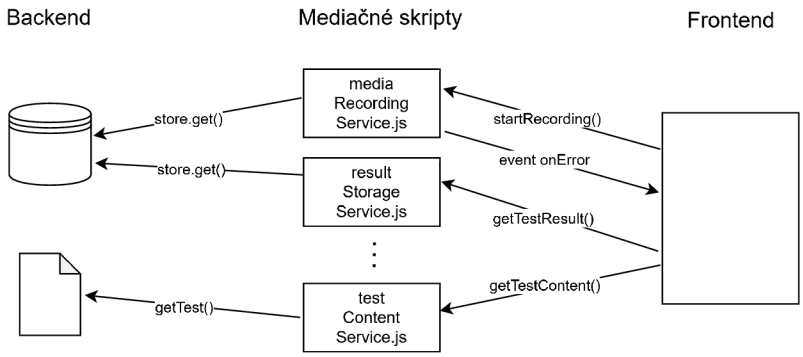
Po zvolení webových technológií nastal čas sa rozhodnúť, či a aký framework použiť pre tvorbu používateľského rozhrania. Hoci naše používateľské rozhranie nepovažujeme za nadmerne komplexné, predsa len sme sa rozhodli použiť javascriptový framework, konkrétne Vue. Dôvodmi bola predošlá skúsenosť s týmto frameworkom, rastúca popularita a subjektívne vyhovujúce abstrakcie ako komponenty alebo tzv. reaktivita.

Čo sa týka backendových technológií, nemenné dáta ako obsah testu sme zvažovali uchovávať tak, že by boli zabalené spoločne s aplikáciou. Avšak úložisko nahrávok sedení a výsledkov testov sme zvažovali ukladať oddelene na webserveri, s ktorým by sa komunikovalo prostredníctvom REST API. Tento prístup sa nám napokon zmenil, o čom sa budeme bližšie vyjadrovať neskôr.

4.3 Architektúra aplikácie

Po výbere implementačných technológií bolo nutné sa zamyslieť nad samotnou architektúrou aplikácie. Základom bolo vymyslieť spôsob, ako mať čo najväčšiu flexibilitu meniť front-end alebo back-end, pričom zmena jedného si nemala vyžadovať zmenu druhého. Zvolili sme prístup tvorby mediačných skriptov, ktorý mal nasledovnú filozofiu:

Mediačné skripty slúžia ako rozhranie medzi front-endom a back-endom, čo pripomína MVVM návrhový vzor. Pri zmene back-endu sa musí prispôsobiť len časť skriptu zodpovedná za komunikáciu s back-endom. Funkcie a udalosti využívané front-endom zostávajú nezmenené. Podobný princíp platí aj pri zmene front-endu.



Obr. 4.1: Architektúra systému z pohľadu oddelenia zodpovednosti

Z ilustrácie je patrné, že pri získavaní rôznych dát sa zvykne volá funkcie, avšak odpoveď neprichádza napr. skrz udalosti. Namiesto toho je väčšina funkcií pre získavanie dát asynchrónnych a využíva javascript Promise, pre hlásenie úspechu alebo chyby pri vykonávaní funkcie. Jediný prípad použitia udalostí nastáva pri situáciách, kedy backend sám bez vyzvania potrebuje zaslať správu - napr. správu o chybe pri čítaní z kamery, čo sa môže stať keď sa kamera odpojí počas jej využívania.

Teraz, po vysvetlení architektúry celkového systému, sa poďme bližšie pozrieť na organizáciu jednotlivých častí systému, počínajúc s frontendom.

4.4 Architektúra GUI

Nakoľko boli pri tvorbe používateľského rozhrania použité web technológie ako HTML, CSS a Javascript, s dodatkom frameworku Vue, bolo prirodzené používať nimi ponúkané konštrukty.

Jednotlivé obrazovky aplikácie a prechody medzi nimi sú riešené pomocou konceptov View a Router vo Vue. Celkovo boli vytvorené 4 Views, každý obsiahnutý v samostatnom súbore, odkazujúce sa na návrh používateľského rozhrania v kapitole Návrh: HomePage.vue - zodpovedá obrazovke zoznamu pacientov, Obr. 3.4 Test.vue - zodpovedá obrazovke testovania pacienta, Obr. 3.1 Evaluation.vue - zodpovedá obrazovke vyhodnocovania testu, Obr. 3.2 Results.vue - zodpovedá obrazovke výsledkov, Obr. 3.3

Nakoľko bola niektorá funkcionalita aplikácie zdieľaná naprieč viacerými obrazovkami, bolo potrebné túto funkcionalitu obsiahnuť v samostatných znovapoužiteľných celkoch - komponentoch. Komponenty sme vytvárali aj pre funkcionalitu, ktorú nebolo potrebné zdieľať s viacerými obrazovkami. Účel tohoto bol lepšia priehľadnosť zdrojového kódu. Niektoré z vytvorených komponentov, pričom každý je obsiahnutý v samostatnom súbore, sú:

- VideoRecorder.vue - komponent na nahrávanie a ukladanie audio alebo video záznamu
- VideoPlayer.vue - komponent na prehrávanie audio a video záznamov
- SubtestPartPlayer.vue - komponent ukazujúci obsah daného subtestu, t.j. vytvára CSS grid zorazujúci obrázky subtestu a zahŕňa v sebe funkcionalitu prehrávanie audio inštrukcií

- VisualTimer.vue - komponent zobrazujúci zbývajúci čas pri subtestoch, kde to je potrebné (subtesty 10 a 11). Tento komponent bol zahrnutý vo vnútri predošlého komponentu
- PatientDetail.vue - komponent zobrazujúci detail pacienta, ako ich história testov

Funkcionalita niektorých komponentov si vyžadovala komunikáciu s backendom. Napríklad, VideoRecorder prirodzene potreboval spôsobiť uloženie videa (ako aj audia, avšak nevýstižné pomenovanie zostalo). Riešili sme to tak, že priamo v komponente sme importovali potrebný mediačný skript, v tomto prípade mediaRecordingService.js.

4.5 Architektúra backendu

Pri koncipovaní backendu sme sa museli zamyslieť nad povahou dát v našej aplikácií - všetky dáta potrebné v aplikácií majú buď nemenný charakter alebo sú vytvárané počas využívania aplikácie.

Nemennými dátami konkrétne myslíme obsah samotného testu:

- Aké obrázky a audio súbory napr. pre inštrukcie potrebuje
- Ako sú tieto súbory usporiadané do subtestov a jednotlivých častí vo vnútri subtestov
- Ako sa nazývajú domény, v rámci ktorých sa dá test hodnotiť
- Aké domény priliehajú ktorým subtestom (toto tvorí stĺpce vo vyhodnocovacom rozhraní vidnom v Návrhu, Obr. 3.2)
- Aké sú názvy položiek a aké majú bodové váhy pre domény (toto tvorí riadky vo vyhodnocovacom rozhraní)

Dáta vzniklé počas behu aplikácie, ktoré je potrebné uložiť, sú nasledovné:

- Identifikačné dáta o testovaní ako meno testovaného, dátum, a podobne
- Nahrávky jednotlivých subtestov (či už zvukovej stopy alebo videa)
- Výsledky hodnotenia

Nemenné dáta

Nemenné dáta sme sa rozhodli mať uložené v rovnakej zložke, v ktorej pod-priečinkoch sú uložené zdrojové súbory projektu. Konkrétne sa na toto využíva špeciálny priečinok public. Obrázky a audio súbory sú uložené ako .svg a .mp3 súbory. Štruktúra testu, t.j. usporiadanie týchto obrázkov a súborov do testu, je uchovaná vo formáte json.

Formát json bol zvolený z dôvodu, že pri lepšom pohľade na ER diagram z návrhu je zjavná štruktúra zanorenia, napr.: Akcie sú existenčne závislé od častí (part), ktoré sú závislé od subtestov, ktoré sú závislé na samotnom teste. Pre takto zanorenú štruktúru nám prišlo využiť json reprezentáciu ako prirodzenú voľbu, nehovoriac o fakte, že json je v javascripte jednoduchý na použitie.

Čo sa týka dát vzniklých počas používania aplikácie, museli sme sa rozhodnúť, ako sa bude ukladať nemalé nahrávky z testovania ako aj výsledky - stav checkboxov v jednotlivých vyhodnoteniach subtestov. Pôvodne sme mienili ukladanie dát riešiť pomocou REST API komunikujúcej s web serverom obsahujúcim databázu. Avšak citlivá povaha nahrávok (jedná sa o dáta podliehajúce lekárskemu tajomstvu) nás donútili ukladať dáta na samotnom zariadení, pomocou ktorého sa testovalo.

Ukladanie veľkého množstva dát vo webovom prehliadači

Nakoľko webové aplikácie nemajú prístup k úložisku zariadenia (resp. nedá sa im dať povolenie na ukladanie viacerých súborov naraz, okrem browseru Chrome s experimentálnou File System Access API), rozhodli sme sa využiť ukladanie dát priamo v prehliadači používateľa. Existujú rôzne mechanizmy pre ukladania dát na strane klienta, avšak pokiaľ ide o veľkosť dát, najviac sa dá uložiť pomocou IndexedDB.

IndexedDB je transakčný databázový systém, ukladajúci dáta na strane klienta, ktorý avšak nepoužíva relačný dátový model. Namiesto toho sú dáta ukladané v podobe objektov a pristupuje sa k nim pomocou kľúčov - databáza je teda objektová databáza a má teda charakter úložiska kľúč-hodnota. Jednotlivé dvojice kľúč-hodnota sú ukladané v rámci tzv. skladov (stores), ktoré prislúchajú konkrétnej inštancií databázy. Každá inštancia je spätá s konkrétnou doménou a dá sa k nej pristupovať len z tejto domény (same-origin policy).

V rámci skladov sa dá jednoducho pridávať, meniť alebo odstraňovať páry kľúčov a hodnôt, avšak pre vytváranie nových skladov je potrebné zmenu schémy databázy. Tento poznatok je kľúčový pre našu aplikáciu, nakoľko spôsob získavania a ukladania dát nahrávok je najviac podstatný pre navrhnutie schémy databázy. Ako teda funguje nahrávanie?

Nahrávanie audio/video obsahu vo webovom prehliadači

V moderných browseroch už je možné pristupovať ku kamere a mikrofónu a zaznamenávať z nich idúci tok dát, buď vo forme zvukovej stopy alebo videa. Táto funkcionalita je štandardizovaná naprieč všetkým moderným browserom v dôsledku projektu WebRTC a je obsiahnutá vo viacerých javascriptových API, z ktorých si zmienime dve:

- Na získanie prístupu k mediálnemu hardvéru slúži funkcia getUserMedia() rozhrania Media Devices and Streams API. Tejto funkcií sa posiela argumenty súvisiace s typom požadovaného zariadenia, bitrate a prípadne rozlíšením. Zavolaním tejto funkcie sa v prehliadači zobrazí modálne okno, v ktorom vie používateľ povoliť alebo zamietnuť požadovaný prístup. Úspešným zavolaním tejto funkcie dostaneme stream dát z požadovaného zariadenia.
- Pre účely nahrávania získaného streamu sa vytvára inštancia objektu MediaRecorder z MediaStream Recording API, ktorej sa v konštruktore posiela tento stream. Následne sa na tomto objekte volá členské metódy ako start alebo stop. Samotné dáta nahraté zo streamu sa uchovávajú v bloboch, ktoré sú v pravidelných intervaloch sprístupňované eventom dataavailable.

Je dôležité podotknúť, že každé spustenie tohoto eventu sprístupňuje blob obsahujúci nahrávku audia alebo videa, ktorá sa začína od posledného volania eventu dataavailable a

končí sa nasledujúcim volaním tohoto eventu. Tým pádom, ak všetky takto získavané bloby konkatenujeme, dostaneme jednu ucelenú nahrávku.

Presne týmto prístupom nahrávame subtesty. Vždy na začiatku subtestu zahájime nahrávanie volaním start(), pravidelne získavané bloby v dataavailable ukladáme sekvenčne do IndexedDB a pri ukončení subtestu voláme stop(), čo vyvolá jedno konečné spustenie dataavailable a uloženie ním získaného blobu.

4.6 Formatívne testovanie a jeho dopad na implementáciu

Neoddeliteľná súčasť tvorby tejto práce bolo overovanie si, či je koncepcia aplikácie vôbec validna. Za týmto účelom bolo prevádzané testovanie ako formatívne (počas návrhu a vývoja aplikácie), tak aj sumatívne (po tom, čo bola aplikácia vytvorená). V tejto podkapitole si priblížime prvú, viac iteratívnu formu testovania. Z dôvodu, že táto forma testovania prebiehala už počas vývoja aplikácie, sa všetky tu zmienené zmeny aj skutočne premietli do výslednej implementácie, minimálne do podoby, kde ľahko pochopiť ich cielený výzor a funkcionalitu. To znamená, tu ukázané obrázky pochádzajú z implementácie, nie návrhu.

Formatívne testovanie ovplyvnilo aplikáciu aj po technickej stránke, čo je priblížené ku koncu tejto podkapitoly. Je tu zdôvodnené, z akého dôvodu nadobudla implementáciu formu webovej aplikácie mienenej najmä pre desktopové počítače a nie pre tablety, ako bolo pôvodne zamýšľané.

Podľa metodiky Goal Directed Design je vhodné v počiatočnom štádiu návrhového procesu robiť rozhovory s expertami v oblasti práce – pre túto prácu to znamenalo, že bolo žiadúce súčasnú podobu MLSE konzultovať so znalou osobou. Touto osobou bol študent logopédie, Tomáš Benko, ktorý tu navrhovanú aplikáciu v jeho diplomovej práci ponímal z hľadiska logopedického, zameriavajúc sa na testovanie pacientov [5]. Takýmto spôsobom sme sa dostali k formatívnemu testovaniu.

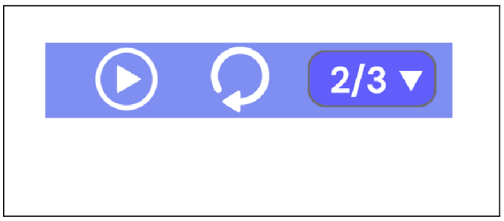
V rámci formatívneho testovania bol najmä Tomáš Benko, no i jeho kolegovia, pravidelne spytovaný na názor na rôzne iterácie návrhu aplikácie, od skíc až po implementáciu. Je samozrejmé, že opakované konzultovanie s jednou osobou u nej spôsobuje značné oboznámenie so systémom a tým pádom ľahšie prehliadnutie nedostatkov použiteľnosti. Tento problém testovania adresujeme v podkapitole o sumatívnom testovaní. Avšak už pri formatívnom testovaní boli postupom času zistené nasledovné problémy s návrhom prezentovaným v predošlej kapitole:

Prvky na ovládanie inštrukcie

V kapitole Návrh bolo spomenuté a odôvodnené, že spočiatku sa po kliknutí na tlačidlo „Play“ s ním nedalo istý čas interagovať. Tento čas bol úmerný dĺžke inštrukcie. Postupom času, ako sa testovaciu časť aplikácie znova a znova testovalo, sa hromadila frustrácia z takéhoto odobratia slobody. Ukázalo sa, že by bolo predsa len lepšie používateľovi umožniť v prípade nesprávneho kliknutia sa k inštrukcii vrátiť. Prehranie momentálnej inštrukcie už možné bolo vďaka tlačidlu „Zopakovať“. No ako docieliť, aby sa dalo vrátiť napr. o 3 inštrukcie naspäť alebo preskočiť k poslednej inštrukcií? A ak sa dá preskakovať, ako dať

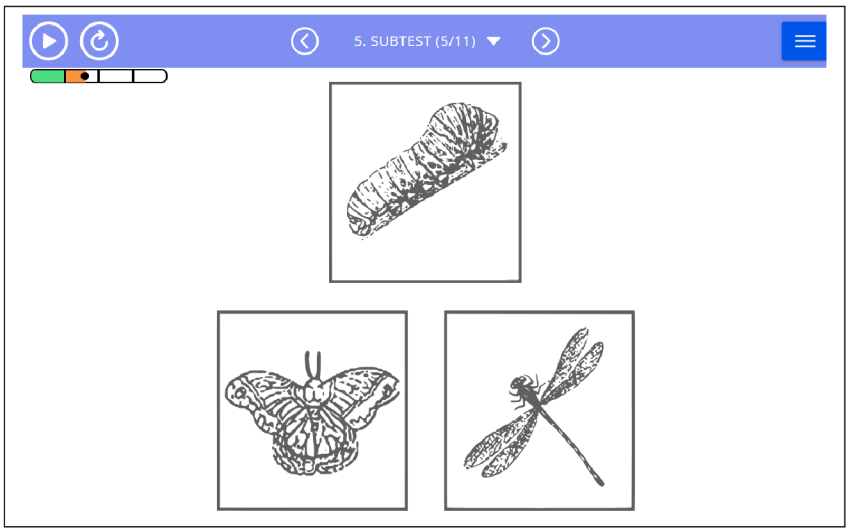
administrátorovi vedieť, ktoré inštrukcie už boli prehraté? Koniec koncov, v rámci správnej administrácie subtestu musia byť prehraté všetky inštrukcie.

Zhotovené boli 2 návrhy pre komponentu ovládania inštrukcie, ktoré riešia tieto požiadavky:



Obr. 4.2: Prvý návrh pre vylepšenú komponentu ovládania inštrukcie

Návrh na obrázku 4.2 pridáva k pôvodnému návrhu číselné zobrazenie inštrukcie vo formáte „momentálne číslo inštrukcie“ / „celkový počet inštrukcií“. Takéto zobrazenie len informuje administrátora o tom, na ktorej inštrukcií sa nachádza – čiže pri omylnej zmene inštrukcie by si mal všimnúť nepožadovanú zmenu čísla. Pre účel zmeny inštrukcie na ľubovoľnú sa tento text zobrazuje v tlačidle s ikonou naznačujúcou dropdown. Po kliknutí na tento dropdown by sa malo zobraziť okno, v ktorom si vie administrátor kliknutím vybrať požadovanú inštrukciu – čísla inštrukcií by boli zobrazené pod sebou v riadkoch a pri každom čísle by bola fajka alebo krížik indikujúci, či daná inštrukcia už bola spustená.



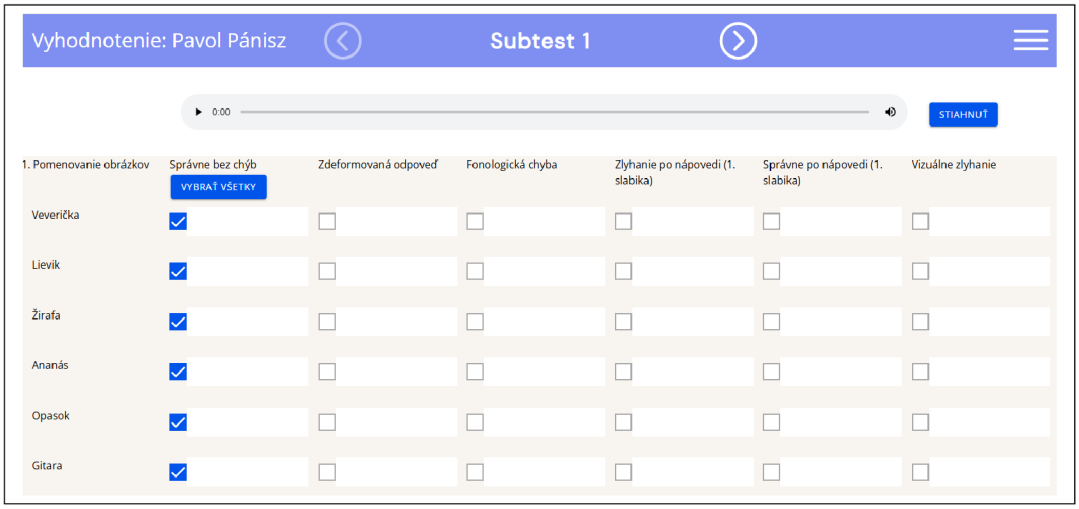
Obr. 4.3: Druhý návrh pre vylepšenú komponentu ovládania inštrukcie, ako ho vidno v implementácií

Návrh na obrázku 4.3 plní rovnaký účel, avšak robí to graficky, bez textu. Toto grafické zobrazenie má pripomínať slider podobný slidera v prehrávačoch videí. Slider je priečnymi čiarami členený na články, ktorých počet je rovný počtu inštrukcií pre daný subtest. Po spustení inštrukcie sa jej priliehajúci článok slideru začne postupne zapĺňať oranžovým obdĺžnikom s čiernym kruhom na konci. Tento obdĺžnik ako aj kruh sa rozširujú doprava, pričom doba zaplnenia zodpovedá trvaniu inštrukcie. Kruh má naznačovať, že sa s ním dá

interagovať – dá sa potiahnuť na začiatok (ľavú stranu) každého článku. Touto interakciou sa mení momentálna inštrukcia. Hoci je tento návrh na prvý pohľad ťažší na pochopenie, bol zvolený. Dôvod je ten, že oproti dropdownu z neho už na prvý pohľad vidno všetky potrebné informácie. Známa chyba: V implementácii sa pomocou slidera nedá meniť inštrukciu a doba trvania animácie kruhu nezodpovedá trvaniu inštrukcie.

4.7 Tlačidlo „všetko bez chyby“

Počas testovania na pacientoch sa ukázalo, že veľa pacientov malo väčšinu subtestov bez chyby. Ak aj mali nejakú chybu, väčšinou stále prevládali riadky bez chyby. Aj pri štandardizácii experimentálnej adaptácie MLSE by sa muselo testovať na veľkom počte ľudí, u ktorých sa nepredpokladá patológia, a tak je šanca na bezchybný subtest vysoká. Z týchto dôvodov bola do vyhodnocovacej časti pridaná malá pozornosť pre administrátora – tlačidlo na označenie celého prvého stĺpca, ako to vidno na obrázku 4.4.



Obr. 4.4: Tlačidlo pre zaškrtnutie / odfajknutie všetkých políčok stĺpca „bez chyby“, ako ho vidno v implementácií

4.8 Problémy s nahrávaním na tablete iPad

Hoci bola digitálna verzia MLSE mienená pre tablety, ja tablet nevlastním. Z toho dôvodu prebiehal vývoj aplikácie buď na počítači s operačným systémom Windows 11 alebo MacOS Sequoia. Väčšinu času sa aplikácia tým pádom spúšťala na desktopovom zariadení a len počas konzultácií s Tomášom Benkom bývala aplikácia testovaná na jeho tablete iPad Pro, 7. generácie, s operačným systémom iPadOS 18.

Jeden veľmi špecifický, no na prvý pohľad nie zjavný technický požiadavok, je nahrávanie za špeciálnych podmienok – nahrávanie sa deje počas testovania, kedy jednak aplikácia sama od seba môže generovať zvuk (počas prehrávania inštrukcií), no druhak aj pacient môže vytvárať zvuk (keď odpovedá). Môžu teda nastať situácie, kedy aj aplikácia aj pacient naraz „rozprávajú“. Audio výstup zariadenia je pri tom predpokladaný, že nastáva z reproduktorov – a v dôsledku nahrávania je tento audio výstup zachytený tým istým zaria-

dením, z ktorého pochádza.

Pri nahrávaní na desktope táto situácia nikdy nespôsobovala problém. Na nahrávkach bolo vždy počuť ako inštrukcie, tak aj pacienta, čo bol požadovaný výsledok. Zvuk inštrukcií je žiadaný v nahrávke, keďže sa vďaka nemu dá v nahrávke identifikovať, ktorú inštrukciu pacient práve vykonáva. Avšak na tablete iPad sú nahrávky bez zvuku.

Z experimentovania s nahrávaním na tablete iPad bolo pozorované, že na začiatku nahrávky počuť aj zvuk. Avšak tento zvuk sa v zapätí postupne stíši, zhruba v priebehu sekundy. Nie je úplne zjavné, prečo sa to deje, no predpokladá sa akýsi interný mechanizmus potláčania ozveny, ktorý sa využíva (a zíde) počas videohovorov. Pre naše účely je tento hypotetizovaný mechanizmus kontraproduktívny, avšak nepodarilo sa ho vypnúť. Pokus o mitigáciu tohoto problému spočíval v konfigurácií rôznych parametrov počas získavania prístupu k mediálnemu hardvéru, napr.:

```
const mediaConstraints = {
    video: videoSettings,
    audio: {
        noiseSuppression: false,
        echoCancellation: false,
        autoGainControl: false
    }
};

mediaDevices.getUserMedia(mediaConstraints);
```

Keďže sa tento problém nepodarilo vyriešiť, nastalo rozhodnutie pokračovať vo vývoji aplikácie primárne pre desktop. Možné riešenie pre iPad by mohlo spočívať vo vytvorení hybridnej aplikácie – na front-end by sa využíval rovnaký zdrojový kód ako v momentálnej verzii aplikácie. Pre back-end by sa prepísala časť súboru mediaRecorderService.js tak, aby získavala prístup k mediálnemu hardvéru pomocou natívneho frameworku AVFoundation [1]. Muselo by sa preskúmať, či vďaka rozhraniu ponúkanému týmto frameworkom by sa dalo docieliť nahrávanie s požiadavkami stanovenými v tejto práci.

4.9 Sumatívne testovanie

Počas formatívneho testovania s tou istou osobou sa z nej časom stáva skúsený používateľ. Nadobudnuté informácie o testovanom systéme u tejto osoby môžu spôsobovať, že prehliada potenciálne problémy s použiteľnosťou aplikácie. Z tohto dôvodu bolo prevedené testovanie aj na ďalších osobách, avšak až po tom, čo bola aplikácia z veľkej časti funkčná. Cieľom takéhoto sumatívneho testovania bolo zistiť ich perspektívu na aplikáciu z pohľadu jednak použiteľnosti a druhak užitočnosti oproti papierovej verzii. Z dôvodu, že toto testovanie prebehlo až keď bola aplikácia hotová, sa tu zmienené návrhy na zmeny neimplementovali.

Toto testovanie bolo prevedené na 3 naivných používateľoch, ktorí aplikáciu nikdy predtým nevideli a nepohybovali sa v sfére logopédie. Napriek ich neznalosti tejto odbornej domény však boli vyhodnotení ako vhodní kandidáti na testovanie, a to z dôvodu, že testovanie

sa nezameriavalo na obsah samotného logopedického testu, ale na interakciu so systémom.

Testovací protokol

Pôvodne mal testovací protokol mať ako kvalitatívny, tak aj kvantitatívny charakter. V rámci kvantitatívneho testovania bolo plánované meranie času vyhodnotenia testu: Ako dlho trvá vyhodnotenie testu pomocou digitálnej verzie MLSE oproti papierovej verzií? Okrem merania času bolo plánované klásť otázky z dotazníkov ako NASA TLX a SUS, pre vypočítanie celkového skóre použiteľnosti aplikácie. Napokon sa z tohoto plánu ustúpilo z dôvodu malého počtu účastníkov na testovaní. Avšak otázky kladené v rámci testovacieho protokolu boli inšpirované tými z týchto dotazníkov.

Testovací protokol sa skladal z 3 častí:

1. Úvod používateľa do problematiky
2. Testovanie aplikácie z role administrátora
3. Diskusia a otázky o dojmoch z aplikácie

Úvod používateľa do problematiky

Ako prvé bola respondentovi, ktorý aplikáciu testoval, priblížená tematika aplikácie. Bolo vysvetlené, že sa jedná o aplikáciu využívanú logopédmi za účelom administrácie a vyhodnotenia istého testu. Následne bolo stručne vysvetlené, o aký test sa jedná – aký je cieľ MLSE, akou formou sa pacienta testuje a ako sa test vyhodnocuje. Za týmto účelom boli osobe ukazované hárky z pôvodnej experimentálnej verzie MLSEsk.

V bode úvodu do problematiky by sa dalo zvoliť aj alternatívny prístup – vysvetliť osobe len slovne, o čom je aplikácia a nechať ju samu, nech si dotvorí obraz pomocou „hrania“ sa s aplikáciou. A skutočne, v imaginárnom svete kde digitálna verzia MLSE je dostupná v digitálnych obchodoch by sa človek dozvedel o teste takýmto spôsobom, resp. pomocou fotiek aplikácie. Avšak úlohou tohoto testovania bolo aj zistiť, či ľuďom príde zmysluplnejšie používať aplikáciu namiesto papierovej verzie – a práve preto padlo rozhodnutie ukázať aj časti papierovej verzie.

V tejto fáze boli taktiež kladené otázky, ktorých približné znenie bolo:

- Ako by ste sa cítili, keby Vás testuje logopéd a počas odpovedania by ste videli, ako si niečo píše bokom na papier?
- Ak by ste boli logopéd, ako by ste zaistili, nech sa pacient cíti pohodlne počas testovania?
- Ak by ste boli logopéd, ako by ste zaistili, aby pri sčítavaní bodov nenastala chyba?
- Keďže sa PPA môže aj viac krát v priebehu rokov administrovať, ako zaistiť, aby ste ako logopéd mali k dispozícií informácie z predošlých sedení, pre ich porovnanie?

Tieto otázky boli kladené nie len pre väčšiu záživnosť testovania, no taktiež pre uvedomenie si nevýhod papierového testovania, zmieneného v kapitole UX Analýza papierovej verzie MLSE.

Testovanie aplikácie z role administrátora

Druhá fáza aplikácie prebehla formou hry na role, v ktorej som ja nadobudol rolu pacienta a osoba testujúca aplikáciu zaujala rolu administrátora. Pred začatím tejto fáze bolo túto osobu povzbudené, aby rozmýšľala nahlas a nebála sa vyslovovať dojmy nadobudnuté počas používania aplikácie. Taktiež bolo dôležité osobe neradiť – nechať ich, nech samy prídu na to, ako sa aplikácia používa. Toto malo jasný účel zistenia, ako naučiteľná je aplikácia skutočne.

Postupovalo sa tak, ako by používanie aplikácie prebiehalo v skutočnosti:

- Najskôr sa vytvoril profil pre nového pacienta.
- Následne sa začal nový test pre tohoto pacienta. V záujme stručnosti sa netestovalo všetkých 11 subtestov. Namiesto toho sa testovalo len prvé 3, po čom bolo podľa uváženia vybrané ešte spomedzi posledných 2 subtestov, pre demonštráciu všetkých spôsobov interakcie s aplikáciou počas testovania.
- Počas testovania bolo administrátorovi prikázané, aby aplikáciu zatvoril a znova otvoril, čo malo simulovať nečakanú udalosť počas testovania, ako napr. vybitie batérie zariadenia, a podobne. Po opätovnom otvorení aplikácie si mal administrátor uvedomiť, že sa vie k testovaniu v takomto prípade vrátiť.
- Po ukončení testovania pacienta nasledovalo vyhodnocovanie výsledkov, pri čom bolo opäť prikázané aplikáciu zatvoriť a otvoriť.
- Po ukončení tejto fázy nasledovalo zobrazenie výsledkov, stiahnutie pdf súboru s výsledkami a vrátenie sa späť na domovskú obrazovku. Na domovskej obrazovke aplikácie bola položená administrátorovi otázka, ako by si vedel/a pripomenúť výsledky testovania – či pochopia, že v karte pacienta je história sedení s nahrávkami.

Diskusia a otázky o dojmoch z aplikácie

Posledná fáza testovania mala formu hĺbkového rozhovoru, s nie presne definovanou formou. Ako prvé však bola vždy položená otázka v zmysle „Aký máte z toho celého dojem?“. Toto zvyklo rozprúdiť diskusiu, ktorej interpretáciu si načrtneme v nasledovnej podkapitole. Avšak po diskusií nasledovali otázky, ktoré boli inšpirované spomenutými dotazníkmi a mali približne nasledovné znenie:

- Prišla Vám aplikácia intuitívna?
- Aký ste mali pocit, čo sa týkalo robenia chýb? Povoľovala Vám aplikácia vrátiť sa, kam ste chceli?
- Myslíte si, že bežný človek sa vie rýchlo naučiť ako používať aplikáciu?
- Cítili ste sa niekedy počas používania aplikácie v strese alebo v časovej tiesni?
- Vedeli ste vždy všetko vykonať rýchlo a efektívne? Alebo Vám niekedy niečo prišlo zdĺhavé, než to docielite?

Tieto otázky cielili na princípy použiteľnosti, no po nich nasledovala diskusia o porovnávaní digitalizovanej verzie MLSE s papierovou. Pri tejto diskusií si respondenti uvedomovali (alebo sa pýtali, či sú ich domnienky správne), ako digitálna verzia adresuje nedostatky papierovej.

4.10 Výsledky testovania

Celkovo mali všetky osoby zúčastnené na testovaní kladný pohľad na aplikáciu. Vyzdvihovali najmä efektívnosť oproti papierovej verzií – automatické nahrávanie pacienta, uniformitu vyslovovania inštrukcií, automatické sčítavanie bodov a jednotnosť ukladania a vyhľadávania archivovaných výsledkov.

Čo sa týka používateľskej skúsenosti, kým bolo vidno isté váhanie pri odpovedaní, celkovo hodnotili používateľskú skúsenosť kladne: Aplikácia im prišla v celku intuitívna, najmä pretože na úvodnej obrazovke je spočiatku len vždy 1 možnosť na výber – vytvoriť nového pacienta, začať nový test, stlačiť tlačidlo na prehratie inštrukcie. Vďaka simulácií vybitia batérie používatelia uznali, že aplikácia povoľuje administrátorovi byť uvoľnený a nebáť sa straty dát / momentálneho. Z aplikácie nikdy nemali pocit, že by sa s niečím museli ponáhľať alebo naopak, že by im bránila v tom, aby robili veci efektívne (rýchla navigácia).

Sú isté nedostatky s použiteľnosťou, ktoré však nepovažujeme za tak závažné, aby robili celú koncepciu systému neplatnú. Tieto nedostatky sa dajú odvodiť z parafrázií zapísaných výrokov testerov:

- „Nechápem, prečo v main menu (v karte pacienta) vypisuje, že prebieha (pozerajúc sa na históriu testov, je tam jeden prebiehajúci test)... jaj ahá!“ – Tu si dotyčný až po chvíli uvedomil, že stav sedenia je prebiehajúci, pretože neprešiel si všetky subtesty. Lepším riešením by bolo v histórií sedení ukazovať nie len či test prebieha alebo či čaká na vyhodnotenie, no taktiež informáciu o tom, *ktorý* subtest ešte treba administrovať a vyhodnotiť.

- „Nepáči sa mi, že keď kliknem na play (tlačidlo na prehratie inštrukcie), tak niekedy musím znova kliknúť, aby to začalo hrať.“ – Tu dotyčný narážal na fakt, že keď sa dokončí subtest, kliknutím tlačidla na prehratie inštrukcie sa *len* prejde na ďalší subtest, no *nezačne sa zároveň* aj rovno hrať prvá inštrukcia toho subtestu. Používateľ je teda zjavne naučený očakávať, že interakciou s týmto tlačidlom by vždy malo nastať prehratie inštrukcie. Ďalší tester mal tiež podobný problém s týmto tlačidlom a navrhoval, aby sa ikonka tlačidla inštrukcie menila podľa toho, akú akciu má vykonať.

- „Jaj, to som nevedel.. že musím kliknúť na dalmatínca (výber obrázku)“ – Tu bolo narážané na fakt, že ani inštrukcia „Dalmatínec.. Teraz ukážte, ktorý z obrázkov to je?“, ani samotná aplikácia nedajú administrátorovi vedieť, že pri takomto subteste treba na obrázky klikať. Riešením by mohlo byť to, že keď administrátor ide spustiť ďalšiu inštrukciu bez toho, aby bol vybratý niektorý z obrázkov, tak dostane o tom upozornenie.

Kapitola 5

Záver

Cieľom tejto práce bolo vytvoriť aplikáciu, ktorá je digitalizovanou verziu logopedického testu MLSE.

Tento cieľ sa podarilo z veľkej časti splniť. Bola študovaná oblasť návrhu a tvorby aplikácií s dôrazom na používateľskú skúsenosť. Analyzovaná bola súčasná podoba MLSE a existujúce aplikácie využívané logopédmi. Na základe nadobudnutých poznatkov bol zhotovený návrh aplikácie, ktorá je digitálnou verziou tohoto testu. Návrh bol následne implementovaný formou webovej aplikácie, ktorá bola testovaná ako počas vývoja, tak aj po ukončení vývoja. Testovanie bolo vykonávané aj s osobou dobre oboznámenou s tematikou MLSE a výsledky testovania naznačujú, že práca má skutočný prínos pre logopédov.

Výsledná implementácia je aplikácia, ktorá vo svojej podstate zahŕňa všetky aspekty práce s testom MLSE a vylepšuje ich. Umožňuje testovanie pacientov, pričom inštrukcie testu sú nahlas prehrávané aplikáciou, čím sa zaisťuje maximálna konzistentnosť a zrozumiteľnosť. Počas testovania sa automaticky zhotovuje audio alebo video-nahrávka. Po dokončení testovania si vie logopéd túto nahrávku spustiť, za účelom pripomenutia a pohodlného vyhodnotenia výkonu pacienta. Počas vyhodnocovania aplikácia automaticky sčítava body pre jednotlivé domény testu. Tieto výsledky sa dajú sumárne zobraziť a stiahnuť vo formáte pdf. Rovnako sa dá sťahovať aj nahrávky, i keď aplikácia tieto výstupy z testovania ukladá interne. Pre každého pacienta sa dá výsledky a nahrávky ich testov kedykoľvek pozrieť za účelom sledovania stavu pacienta naprieč viacerým sedeniam.

Osobne ma práca naučila veľa nového, nie len z pre mňa do nedávna neznámej oblasti používateľskej skúsenosti – presvedčil som sa na vlastnej koži o zložitosti kompletného cyklu tvorby digitálneho produktu. V procese písania tejto technickej správy som sa musel zlepšiť v triedení si myšlienok a ich spísaní do súvislého celku textu.

Čo sa týka výhľadov do budúcna pre túto prácu, rozhodne by som chcel popracovať na aspekte súvisiacom s vytváraním a ukladaním nahrávok – toto by vyžadovalo prehodnotenie zmysluplnosti aplikácie ako webovej. V širšom časovom horizonte by sa dalo prácu využiť a prispôsobiť na podporu prebiehajúcej štandardizácie slovenskej adaptácie MLSE. V prípade úspešnej štandardizácie by sa digitálna verzia MLSE mohla stať preferovanou formou využívania nástroja MLSE na Slovensku.

Literatúra

[1] *AVFoundation* Online: https://developer.apple.com/av-foundation/. Accessed: 2025-05-10.

[2] *How Capacitor Works* online. 2025. Dostupné z: https://capacitorjs.jp/blog/how-capacitor-works.

[3] *StatCounter* online. 2025. Dostupné z: https://gs.statcounter.com/os-market-share/tablet/worldwide.

[4] *Wikipedia* online. 2025. Dostupné z: https://en.wikipedia.org/wiki/Native_(computing).

[5] Benko, T. *Návrh digitalizovanej verzie skríningu primárnej progresívnej afázie MLSEsk*. 2025. Diplomová práca. Univerzita Komenského v Bratislave PdF PdF.KL. Vedúci práce Csépalvay, Z. Dostupné z: https://opac.crzp.sk/?fn=detailBiblioFormChildG1AJR&sid=6F9F3F51667F0CBC3944BEEE069E&seo=CRZP-detail-kniha.

[6] Chroboczek, M. *Grafická uživatelská rozhraní v Qt a C++ : [plně kompatibilní s Qt 5]*. 1. vyd. Brno: Computer Press, 2013. ISBN 978-80-251-4124-3.

[7] Cooper, A. *About face : the essentials of interaction design*. Fourth edition. Indianapolis: Wiley, 2014. ISBN 978-1-118-76657-6.

[8] Eliášová, M. *Využití tabletových aplikací v praxi logopedů [online]*. 2020. Bakalářská práce. Masarykova univerzita, Pedagogická fakulta, Brno. Vedúci práce Jůzová, I. Dostupné z: https://is.muni.cz/th/q009p/.

[9] Guo, L. *The First Line of Code: Android Programming with Kotlin*. 1. vyd. Singapore: Springer, 2022. ISBN 981191799X.

[10] Hajian, M. *Progressive web apps with Angular : create responsive, fast and reliable PWAs using Angulra*. New York: Apress, 2019. ISBN 978-1-4842-4447-0.

[11] Hartson, R. a Pyla, P. S. *UX Book - Process and Guidelines for Ensuring a Quality User Experience*. 1. vyd. San Francisco: Elsevier, 2012. ISBN 9780123852410.

[12] Kerekrétiová, A. *Logopédia*. 2. vyd. Univerzita Komenského v Bratislava, 2020. ISBN 978-80-223-4835-5.

[13] Kilvádyová, D. *Hodnotenie jazykových deficitov pri primárnej progresívnej afázii*. 2024. Diplomová práce. Univerzita Komenského v Bratislave, Pedagogická fakulta.

Vedúci práce Csséfalvay, Z. Dostupné z: https://opac.crzp.sk/?fn=detailBiblioFormChildGD0G0&sid=E866CA411CD4E5149FB08769BB2C&seo=CRZP-detail-kniha.

[14] Neuburg, M. Part I. Views. In: *Programming IOS 11*. United States: O'Reilly Media, Incorporated, 2017. ISBN 9781491999226.

[15] NIELSEN, J. *Usability 101: Introduction to Usability* online. 2011. Dostupné z: https://web.archive.org/web/20110408184029/http://www.useit.com/alertbox/20030825.html.

[16] Patel, N.; Peterson, K. A.; Ingram, R. U.; Storey, I.; Cappa, S. F. et al. A 'Mini Linguistic State Examination' to classify primary progressive aphasia. *Brain Communications*, December 2021, zv. 4, č. 2, s. fcab299. ISSN 2632-1297. Dostupné z: https://doi.org/10.1093/braincomms/fcab299.

[17] Patel, N.; Peterson, K. A.; Ingram, R. U.; Storey, I.; Cappa, S. F. et al. A 'Mini Linguistic State Examination' to classify primary progressive aphasia. *Brain Communications*, December 2021, zv. 4, č. 2, s. fcab299. ISSN 2632-1297. Dostupné z: https://doi.org/10.1093/braincomms/fcab299.

[18] Taivalsaari, A. a Mikkonen, T. Return of the Great Spaghetti Monster: Learnings from a Twelve-Year Adventure in Web Software Development. In: *Web Information Systems and Technologies*. Switzerland: Springer International Publishing AG, 2018, sv. 322, s. 21–44. Lecture Notes in Business Information Processing. ISBN 9783319935263.