



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

VÝVOJ FRAMEWORKU POČÍTAČOVÉ TERAPIE A APLIKACE PRO ALTERNATIVNÍ KOMUNIKACI TYPU ANO/NE
THE DEVELOPEMENT OF COMPUTER THERAPY FRAMEWORK AND APPLICATION FOR ALTERNATIVE COMMUNICATION OF YES/NO TYPE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAN KALINA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JIŘÍ FIALA

BRNO 2016

Zadání diplomové práce

Řešitel: **Kalina Jan, Bc.**

Obor: Informační systémy

Téma: **Vývoj i-CT frameworku a jeho aplikace pro komunikaci typu ANO/NE
Development of i-CT Framework and Application for Communication of YES/NO
Type**

Kategorie: Softwarové inženýrství

Pokyny:

1. Seznamte se s problematikou "*special educational needs*" (SEN) u osob s mentálním hendikepem.
2. Prostudujte stávající vhodné aplikace pro SEN zaměřené na zjednodušenou komunikaci na platformách iOS a Android. Analyzujte zejména jejich cenovou dostupnost, rozšiřitelnost a použitelnost.
3. Seznamte se s požadavky na návrh a vývoj aplikací z pohledu "*Human Computer Interaction*" (HCI) a požadavky návrhových principů "*počítačové terapie*" (i-CT) a stejnojmenného frameworku.
4. Proveďte návrh rozšíření i-CT frameworku a aplikace pro komunikaci typu ANO/NE postavené na tomto frameworku. Aplikace musí obsahovat praktickou funkcionalitu (viz body 1 a 2). Aplikace bude funkční na systémech iOS a Android a dostupná jako open source.
5. Dle požadavků vedoucího testujte aplikaci v prostředí osob s mentálním hendikepem. Proveďte vyhodnocení dle metod HCI a diskutujte přínos a další rozvoj.

Literatura:

- Projekt I-SEN (otevřená komunita rodičů, pedagogů, terapeutů, IT odborníků), [online], [cit. 2015-09-10]. Dostupné na: <http://www.i-sen.cz>
- Jacobson Ivar, et. al. The Unified Software Development Process, Addison Wesley, 2000
- Krajíček Jiří a Kočí Radek. *Počítačová terapie jako koncept nové formy terapie pro osoby s mentálním postižením: teorie i praxe*. Journal of Technology and Information Education. Olomouc: Universita Palackého, 2014, roč. 6, č. 1, s. 89-103. ISSN 1803-537X
- i-CT Framework (počítačová terapie) v 1.1.0. [online], [cit. 2015-09-10]. Dostupné na: <http://www.fit.vutbr.cz/~ikrajice/prods.php?id=405¬itle=1>

Při obhajobě semestrální části projektu je požadováno:

- První 3 body zadání

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Krajíček Jiří, Ing., UIFS FIT VUT**

Datum zadání: 1. listopadu 2015

Datum odevzdání: 25. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, Božetěchova 2

doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Tato diplomová práce se zabývá problematikou vývoje aplikací pro alternativní a augmentativní komunikaci (AAK) v prostředí speciálního vzdělávání mentálně postižených žáků s poruchami komunikace. Představuje současné problémy vývoje aplikací a navazuje na návrhové principy projektu Počítačové terapie, jejichž dodržení při návrhu a implementaci aplikací by mělo současným problémům zabránit. Cílem práce je vytvoření aplikace pro AAK typu ANO/NE a i-CT frameworku – rámce zjednodušujícího tvorbu dalších podobných aplikací pro uvedené prostředí. Obě části práce jsou funkční na systémech iOS a Android a byly otestovány v prostředí osob s mentálním hendikepem.

Abstract

This master thesis deals with issues of development of applications for augmentative and alternative communication (AAC) in environment of special education for mentally challenged pupils with communication disorders. Thesis presents issues of development contemporary applications and follows on design principles of Computer therapy project, which compliance during the design and implementation should prevent the current issues. The aim of the thesis is creation of application for AAC of YES/NO type and i-CT framework – framework simplifying creation of similar apps for described environment. Both apps works on iOS and Android systems and were tested in an environment of people with mental disabilities.

Klíčová slova

Počítačová terapie, alternativní komunikace, AAK, poruchy komunikace, mentální postižení, speciální vzdělávací potřeby, tablet, multiplatformní vývoj mobilních aplikací, Android, iOS, Cordova, Oracle MAF, Ionic

Keywords

Computer therapy, alternative communication, AAC, communication disorders, mental disability, special educational needs, tablet computer, multi-platform mobile application development, Android, iOS, Cordova, Oracle MAF, Ionic

Citace

KALINA, Jan. *Vývoj frameworku počítačové terapie a aplikace pro alternativní komunikaci typu ANO/NE*. Brno, 2016. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Fiala Jiří.

Vývoj frameworku počítačové terapie a aplikace pro alternativní komunikaci typu ANO/NE

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Jiřího Fialy.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jan Kalina
23. května 2016

Poděkování

Zde bych rád poděkoval panu Ing. Jiřímu Fialovi za odborné vedení práce, paní Mgr. Lence Říhové a Mgr. Ivě Jelínkové z odborné komunity iSEN za ověření použitelnosti implementovaných aplikací v cílovém prostředí a společnosti Red Hat za laskavou podporu této práce.

© Jan Kalina, 2016.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Speciální vzdělávací potřeby	4
2.1	Mentální postižení	4
2.1.1	Klasifikace mentálního postižení	5
2.2	Poruchy komunikace	6
2.3	Augmentativní a alternativní komunikace (AAK)	6
2.3.1	Systém Znak do řeči (TTT systém)	7
2.3.2	Piktogramy	7
2.3.3	Komunikační tabulky	8
2.3.4	Znaková řeč Makaton	8
2.3.5	Výměnný obrázkový komunikační systém (VOKS/PECS)	8
2.3.6	Komunikace typu ANO/NE	8
2.3.7	Kdy použít AAK	9
2.4	Počítačová terapie	10
2.4.1	Návrhové principy počítačové terapie	11
2.4.2	Aplikace pro komunikaci typu ANO/NE	12
3	Multiplatformní vývoj mobilních aplikací	13
3.1	Platforma Android	13
3.1.1	Dalvik a Android Runtime	13
3.1.2	Aplikační rámec platformy Android	13
3.1.3	Aktivity a Intenty	14
3.2	Platforma iOS	14
3.2.1	Cocoa a Cocoa Touch	14
3.2.2	Objective-C a Swift	15
3.3	Rámec Apache Cordova	15
3.4	Oracle Mobile Application Framework	16
3.5	Ionic	17
3.6	Zamezení opuštění aplikace	17
3.6.1	Potlačení tlačítek Zpět a dalších na platformě Android a Cordova	17
3.6.2	Potlačení tlačítka Domů na platformě Android	18
3.6.3	Znepřístupnění systémových nabídek na platformě Android	19
4	Analýza požadavků	21
4.1	Požadavky na i-CT framework	21
4.2	Stávající aplikace pro AAK	22
4.2.1	Klábosil	22

4.2.2	Yes/No (I Can Do Apps)	23
4.2.3	YesNoCommunication (Počítačová terapie)	24
4.2.4	Zhodnocení stávajících aplikací pro AAK	26
4.3	Návrh způsobu splnění požadavků	26
5	Návrh	27
5.1	Návrh i-CT frameworku	27
5.1.1	Rozdělení systému na i-CT framework a aplikace	28
5.1.2	Komunikace i-CT frameworku a výukových aplikací	28
5.1.3	Navržená posloupnost obrazovek i-CT frameworku	31
5.1.4	Návrh struktury databáze i-CT frameworku	32
5.2	Návrh aplikace ANO/NE	33
5.2.1	Online katalog	34
6	Implementace	36
6.1	Implementace i-CT frameworku	36
6.2	Implementace aplikace pro komunikaci typu ANO/NE	39
6.2.1	Online katalog	41
7	Zhodnocení aplikace a testování	42
7.1	Porovnání se současnými řešeními	42
7.2	Testování s reálnými uživateli	43
7.2.1	Způsob měření použitelnosti	44
7.2.2	Výsledky testování i-CT frameworku	45
7.2.3	Výsledky testování Aplikace ANO/NE	46
7.2.4	Zpracování připomínek z testování a zhodnocení	48
8	Závěr	50
	Literatura	51
	Přílohy	55
	Seznam příloh	56
A	Obsah CD	57
B	Instalační manuál – Android	58
C	Instalační manuál – iOS	59
D	Uživatelský manuál i-CT frameworku	60
D.1	Vytvoření a nastavení účtu uživatele – klienta	60
D.2	Práce se záznamy profilů klienta	62
E	Uživatelský manuál Aplikace ANO/NE	63
E.1	Vytváření karet možností	64
E.2	Práce s katalogy	65
E.3	Vytváření seznamů párů	66

Kapitola 1

Úvod

V oblasti speciální pedagogiky, ve vzdělávání dětí s mentálním postižením se již dlouho využívá alternativní a augmentativní komunikace – využívání gest a různých pomůcek jako prostředku pro rozvoj komunikačních dovedností a myšlení dětí s komunikačními vadami nebo mentálním postižením.

V poslední době nastal v této oblasti překotný růst ve využívání prostředků IT. Tento překotný růst byl pravděpodobně zapříčiněn cenovou dostupností elektronických pomůcek, zejména tabletů. Jak ale bylo zjištěno v rámci projektu Počítačové terapie (blíže popsán v kapitole 2.4), růst dostupnosti tabletů nebyl doprovázen adekvátním nárůstem v kvalitě a kvantitě dostupných softwarových produktů umožňujících využít tyto elektronické pomůcky k naplnění speciálních vzdělávacích potřeb žáků.

Hlavním cílem práce je vytvoření **Aplikace pro AAK typu ANO/NE** a aplikaci podporující rozvoj dalších aplikací pro AAK a vzdělávání mentálně postižených klientů – **i-CT framework**. Druhá uvedená aplikace byla vyvíjena ve spolupráci s Bc. Vojtěchem Mecou, který se na vývoji rovněž podílel v rámci své diplomové práce.

Součástí práce je studium problematiky speciálních vzdělávacích potřeb osob s mentálním postižením, které je popsáno v kapitole 2. Navazující seznámení s požadavky návrhových principů Počítačové terapie je v kapitole 2.4. Kapitola 3 se zabývá možnostmi v oblasti vývoje aplikací pro mobilní zařízení, zejména možnostmi současného vývoje pro více mobilních platforem (tzv. multiplatformní vývojem) a specifickými potřebami speciální pedagogiky v této oblasti (podkapitola 3.6).

Kapitola 4 se zabývá analýzou požadavků kladených na implementaci aplikací pro alternativní a augmentativní komunikaci (AAK) se zaměřením na aplikace pro komunikaci typu ANO/NE a také analýzou požadavků na i-CT framework, který by měl tyto aplikace propojovat. Kapitola 5 popisuje návrh uvedených aplikací a kapitola 6 jejich implementaci. Kapitola 7 se pak zabývá zhodnocením úspěšnosti realizace uvedeného řešení.

Kapitola 2

Speciální vzdělávací potřeby

Za žáky se speciálními vzdělávacími potřebami (special educational needs – SEN) jsou podle školského zákona považovány osoby: [53]

- se zdravotním postižením (tělesným, smyslovým, s vadami řeči, s autismem atd.)
- se zdravotním znevýhodněním (oslabením, dlouhodobou nemocí atd.)
- se sociálním znevýhodněním (rodinným prostředím nepodporujícím vzdělávání, neznalostí vyučovacího jazyka atd.)

Tato práce má být primárně zaměřena na osoby s mentálním hendikepem. Právě oni jsou proto předmětem následujících podkapitol.

2.1 Mentální postižení

Mentálním postižením (nebo mentální retardací) je nazýváno trvalé snížení rozumových schopností, které vzniklo v důsledku organického poškození mozku, v důsledku abnormálního vývoje mozku nebo v důsledku nedostatečné funkce centrální nervové soustavy. [53]

Za mentálně postižené (mentálně retardované) jedince se považují jedinci, u nichž dochází k zaostávání vývoje rozumových schopností, k odlišnému vývoji některých psychických vlastností a k poruchám adaptačních schopností. Porucha adaptačních schopností se projevuje zřetelnými problémy v přizpůsobování se nárokům běžného života na odpovídající vývojové úrovni. Hloubka a míra postižení jednotlivých funkcí mozku je u různých postižených jedinců individuálně odlišná. [53]

Podle vývojového období, v němž vzniklo, se mentální postižení rozděluje na oligofrenii a demenci: [53]

- **Oligofrenie** je opoždění duševního vývoje v prenatálním, perinatálním a časně postnatálním období života.
- **Demence** je důsledkem poškození mozku již v průběhu života jedince, zpravidla po dovršení 2. roku života.

Lidé s mentálním postižením netvoří snadno definovatelnou skupinu a rozdíly v jejich mentální úrovni jsou velké. Lidé s lehkým postižením mohou za velmi příznivých okolností úspěšně dokončit školu, uplatnit se v zaměstnání a založit rodinu. Lidé s těžkým a hlubokým postižením jsou naproti tomu prakticky ve všech svých potřebách odkázáni na cizí pomoc a nejsou leckdy schopni samostatného pohybu ani komunikace s okolím. [53]

2.1.1 Klasifikace mentálního postižení

Mentální postižení představuje výrazně sníženou úroveň inteligence. Úroveň inteligence se často vyjadřuje za pomoci inteligenčního kvocientu (IQ). Ten se vyjadřuje jako poměr mentálního věku (určeného výkonem v testovacích úlohách) a chronologického věku. Informuje tak o celkové rozumové úrovni jedince. Nevypovídá ale o kvalitativních zvláštностech inteligence dané osoby a nelze přeceňovat jeho diagnostickou hodnotu. [53]

Pro klasifikaci mentálního postižení se používá 10. revize Mezinárodní klasifikace nemocí (MKN-10), zpracovaná Světovou zdravotnickou organizací v Ženevě. Podle této klasifikace se mentální postižení dělí do 6 základních kategorií. Pro jejich níže uvedené charakteristiky platí, že se vztahují vždy jen na většinu jedinců trpících daným stupněm mentálního postižení. [53]

- **Lehké mentální postižení, IQ 50–69 (F70):** Vývoj řeči bývá opožděný, ale jedinec dosáhne schopnosti užívat řeč účelně v každodenním životě. Dosáhne nezávislosti v osobní péči a praktických domácích dovednostech. Hlavní potíže se projevují při teoretické práci ve škole – vzdělávací program základní školy nejsou schopni plně zvládnout. Mnozí mají specifické problémy se čtením a psaním. Většinu lze zaměstnat prací, jež vyžaduje spíše praktické než teoretické schopnosti včetně málo kvalifikované manuální práce. Obtížně se přizpůsobují kulturním normám a očekáváním a nedokážou samostatně řešit problémy vyplývající z nezávislého života. Obecně jsou ale jejich potřeby a potíže blízké potřebám jedinců s normální inteligencí.
- **Středně těžké mentální postižení, IQ 35–49 (F71):** Výrazně opožděn rozvoj chápání a užívání řeči. Jejich konečné schopnosti v této oblasti jsou individuální. Opožděna a omezena je také soběstačnost (schopnost starat se sám o sebe) a zručnost. Pokroky ve škole jsou limitované ale při kvalifikovaném vedení si osvojí trivium – čtení, psaní a počítání. Jsou schopni jednoduché manuální práce, jestliže jsou úkoly pečlivě strukturovány a je zajištěn odborný dohled. Tyto podmínky splňují chráněné dílny. Zřídka jsou schopni samostatného života, ale obvykle jsou zcela mobilní a fyzicky i sociálně aktivní. Mezi jedinci této skupiny jsou propastné rozdíly v úrovni sensoricko-motorických a v úrovni sociální interakce a komunikace. Někteří jsou schopni jednoduché komunikace, jiní se stěží dokáží domluvit ohledně svých základních potřeb.
- **Těžké mentální postižení, IQ 20–34 (F72):** Úroveň schopností je snižena výrazněji než u středně těžkého mentálního postižení. Zpravidla nezvládnou ani školní trivium – čtení, psaní a počítání. Často trpí výraznou poruchou motoriky a jinými vadami spojenými s poškozením nebo vadami vývoje ústředního nervového systému. Přestože jsou možnosti jejich vzdělávání omezené, včasná kvalifikovaná péče může významně přispět k rozvoji jejich motoriky, rozumových schopností, komunikačních dovedností i jejich samostatnosti.
- **Hluboké mentální postižení, IQ < 20 (F73):** Těžce omezení i ve své schopnosti porozumět řeči. V nejlepším případě jsou schopni pouze rudimentální neverbální komunikace. Většinou imobilní nebo výrazně omezení v pohybu. Vyžadují stálou pomoc a stálý dohled. Možnosti jejich vzdělávání i výchovy jsou velmi omezené, ale existují metodické postupy umožňující rozvoj motoriky i komunikačních schopností. Běžné jsou neurologické i jiné nedostatky postihující hybnost, zrak nebo sluch.

Zbývající 2 kategorie (Jiné mentální postižení – F78 a Nespecifikované mentální postižení – F79) jsou určeny pro případy, kdy je stanovení stupně postižení nesnadné nebo

nemožné provést pomocí obvyklých metod – například pokud je jedinec neslyšící, nevidící nebo nemluvící.

2.2 Poruchy komunikace

Mentální postižení je prakticky vždy doprovázeno opožděným vývojem řeči. Odchytky od normy, jako pomalý rozvoj a deformace ve všech jazykových rovinách, se přitom projevují již v raném věku. Kromě nejlehčích forem mentálního postižení nedosáhne vývoj řeči ani v dospělosti normy. [52]

- **Vývojová dysfázie** označuje specificky narušený vývoj řeči. Projevuje se ztíženou schopností nebo neschopností naučit se verbálně komunikovat, přestože jsou podmínky pro rozvoj řeči přiměřené. [52]
- **Dyslalie (patlavost)** označuje neschopnost používat jednotlivé hlásky nebo jejich skupiny v mluvené řeči podle ortoepických norem. Dítě hlásku vynechává, nahrazuje jinou hláskou nebo hlásku tvoří chybně. Obvykle nesouvisí s mentálním postižením a vyskytuje se i u dětí s nadprůměrným IQ. [52]
- **Dysartrie** je poruchou motorické realizace řeči na základě organického poškození centrální nervové soustavy. Zahrnuje poruchy způsobené obtížemi ve **svalové kontrole** řečových mechanismů – dýchání, fonaci, rezonanci nebo artikulaci. [52]
- **Anartrie** se projevuje poruchami programování řečových prvků, hlásek, slabik, jejich záměnami, vynecháváním či opakováním. Tyto projevy nejsou konstantní a nebývají spojeny s poruchami dalších motorických řečových činností. [52]

Související s organickým poškozením nebo onemocněním mozku (tedy pravděpodobně i s mentálním postižením):

- **Afázie** (od *phasis* – zápor) je ztráta exprese (vyjadřování) nebo percepce (vnímání) lidské řeči v důsledku postižení mozku. Rozlišuje se na expresivní a percepční afázii. Fatické funkce se dělí podle fluence (plynulosti řeči) na nonfluentní (méně než 50 slov za minutu), fluentní (přibližně 120 slov za minutu – odpovídá normální produkci řeči) a hyperfluentní (více než 200 slov za minutu). [52]
- **Poruchy řečové komunikace u demence** (viz 2.1) jsou výsledkem globálního úbytku paměťových a intelektových schopností. Dominují progredující (postupující) či stacionární poruchy verbální paměti a schopnosti užití řeči pro smysluplnou obsažnou komunikaci s okolím. Mezi symptomy patří mimo poruch řeči také poruchy neverbální komunikace, poruchy kognitivní, poruchy chování a psychické poruchy. [52]

2.3 Augmentativní a alternativní komunikace (AAK)

Termín augmentativní komunikace (z latinského *augmentare* – rozšiřovat) označuje komunikační systémy, které nějakým způsobem podporují nedostatečně rozvinuté komunikační dovednosti a doplňují mluvenou řeč, je-li málo rozvinutá a málo srozumitelná. [53]

Termín alternativní komunikace označuje komunikační systémy, které mají plně nahradit mluvenou řeč v případech, kdy není vůbec rozvinutá nebo zanikla. Z výše definovaných kategorií se tedy týká zejména Těžkého (F72) a Hlubokého (F73) mentálního postižení. [53]

Souhrnně tyto metody sociální komunikace označujeme zkratkou **AAK – Augmentativní a alternativní komunikace**. Tyto metody mají sloužit ke kompenzaci nedostatku řečových schopností, ale neměly by potlačovat projevy mluvené řeči. Cílem metod je umožnit lidem s postižením vytěžit maximum ze svých komunikačních dovedností a kompenzovat tak svoje znevýhodnění v oblasti komunikace. [53]

Komunikační systémy se dělí na statické a dynamické podle toho zda využívají pohyb. Mezi statické systémy se řadí systémy nevyužívající pohyb – např. Piktogramy. Mezi dynamické systémy se řadí systémy využívající pohyb (typicky gesta) – např. Znak do řeči nebo Makaton. [53]

2.3.1 Systém Znak do řeči (TTT systém)

Systém Znak do řeči (známý také jako TTT systém) je kompenzační prostředek pro lidi s mentálním a řečovým postižením. Vychází z myšlenky, že použití znaků napomáhá komunikaci založené na mluvené řeči, gestikulaci a mimice. [53]

Znakem se myslí specifické tvary a pohyby rukou, podobné znakům vizuálně-motorických komunikačních systémů pro komunikaci se sluchově postiženými, jako je Znaková řeč. [41] V případě systému Znak do řeči ale nejde o samostatný jazyk s vlastními mluvnickými pravidly. Používá se v kombinaci s gramaticky správnou mluvenou řečí. Přitom se znakuji pouze slova, která jsou důležitá pro pochopení významu věty, nebo které je obtížné jinak vyjádřit. [53]

2.3.2 Piktogramy

Piktogramy jsou maximálně zjednodušená zobrazení předmětů, činností a vlastností. Jejich cílem je umožnit rychlou orientaci všude, kde by slovní vyjádření mohlo být překážkou v porozumění. [39]

Piktogramy se často používají k vyjadřují instrukcí, příkazů a varování. Jsou rozšířené v dopravě, nemocnicích i jinde. Není proto překvapením, že stejnojmenná metoda patří k nejpoužívanějším metodám alternativní a augmentativní komunikace. [39]

Tato metoda alternativní a augmentativní komunikace je založena na předkládání symbolů v dvourozměrné podobě. Typicky se jedná o bílé symboly na černém pozadí na papírových kartičkách. [53]

Sady symbolů jsou uživatelům poskytovány ve formě učebnic piktogramů, z nichž pro každého jednotlivce pedagogičtí pracovníci sestavují **individuální slovníky piktogramů** (také komunikační knihy). Ty v případě potřeby obohacují o vlastní stylizované kresby, obrázky, loga apod. [53]

Pro úspěšné zavedení komunikačního systému je vhodné, aby dítě obrázek pouze neukazovalo, protože děti nemusí pochopit že jde o vzájemnou interakci – že obrázek ukazují druhému člověku. Je proto doporučeno vyžadovat od dětí podávání kartiček symbolů do ruky, díky čemuž by měly snáze pochopit, že musí někoho skutečně kontaktovat. [40]

Současně je důležité, aby bylo dítě ke komunikaci motivováno. Obrázky zahrnované zpočátku do individuálního slovníku piktogramů by měly reflektovat skutečná přání dítěte. Jejich plnění druhou stranou komunikace pak má fungovat jako odměna. [40]

Raná slovní zásoba je převážně obrázková (symboly jsou obrazem reálných předmětů nebo činností), později se může přecházet k obecnějším (třídy předmětů, vlastnosti) a abstraktnějším symbolům. [53]

2.3.3 Komunikační tabulky

Komunikační tabulky jsou prostředkem sdružujícím skupinu symbolů v ploše. Forma tabulky a rozmístění symbolů v její ploše je přizpůsobeno potřebám a pohybovým a zrakovým možnostem uživatele.

Přizpůsobit je možné velikost, umístění symbolů a barevný kontrast symbolů, přičemž se zohledňují především jazykové potřeby uživatele.

Tabulky mohou mimo symbolů známých z Piktogramů obsahovat také písmena, slova, celé věty nebo číslice.

Nemluvící osoba může symboly označovat ukázáním prstem, pěstí, pohledem, hlavovým ukazovátkem, světelným paprskem nebo i jinak – podle motorických možností uživatele.

Stejně jako v případě slovníku Piktogramů, i obsah komunikační tabulky by měl reflektovat zájmy, osobnost a věk uživatele a jeho vrstevníků. Tvorba komunikační tabulky je proto nikdy nekončící proces, který musí zajistit nezbytné rozšiřování slovní zásoby. Počet symbolů by měl být stále úměrný schopnostem uživatele.

Podle zkušeností škol a zařízení využívajících komunikační tabulky postačuje některým uživatelům jediná tabulka se symboly jako jediný komunikační prostředek. [53]

2.3.4 Znaková řeč Makaton

Makaton je dynamický komunikační systém založený na znakovém jazyce neslyšících. Jeho znaky jsou ale vždy doprovázeny artikulovanou řečí. [53]

Jednotlivá slova jsou vyjadřována pohyby ruky, hlavy apod. Přestože se znakuje pouze klíčová slova vět, doprovázejí se úplnou gramaticky správnou řečí. Znaky jsou rovněž doprovázeny odpovídajícími mimickými výrazy obličeje. [53]

Makaton je využíván zejména pro komunikaci dětí i dospělých s mentálním postižením, kteří jsou neslyšící, nekomunikují mluvenou řečí nebo špatně rozumějí mluvené řeči. Užívá se pro rozvíjení komunikačních schopností u dětí s autismem nebo s problémy artikulace nebo jako dočasný prostředek během výuky mluvené řeči. Slovník znakové řeči Makaton je mezinárodní, ale jednotlivé znaky jsou v každé zemi standardizovány zvlášť. [53]

2.3.5 Výměnný obrázkový komunikační systém (VOKS/PECS)

Výměnný obrázkový komunikační systém (VOKS/PECS – *Picture Exchange Communication System*) je komunikační prostředek vytvořený pro nemluvící děti trpící autismem. Je možné jej ale použít také pro děti s výraznými problémy s verbální komunikací – trpícími dětskou mozkovou obrnou (DMO), Downovým syndromem či dysfázií. S obměnami lze systém použít také s dospělými jedinci. [28]

Jako komunikační prostředek jsou používány fotografie, barevné piktogramy nebo jiné specifické symboly, které uživateli vyhovují. [28]

2.3.6 Komunikace typu ANO/NE

Komunikace typu ANO/NE je nejzákladnější a nejuniverzálnější formou komunikace vůbec. Používá se pokud nebo dokud ostatní formy komunikace včetně výše uvedených selhávají. Dítě je vedeno k tomu, aby si umělo vybrat ze dvou možností, které mu jsou nabídnuty. [22]

Komunikace typu ANO/NE se přitom ve výuce využívá dvěma způsoby:

- Vzdělávání a formování komunikace: dítě je vyzváno, aby vybralo obrázek např. míče, a ukázalo tak že rozumí významu slova "míč".
- Samotné dorozumívání: umožnit mentálně postiženému klientovi vyjádřit své přání. (A tím současně motivovat ke snaze komunikovat – třebaže jen formou výběru ze dvou možností.)

Existuje široká škála způsobů, jak může v komunikaci omezený člověk vybrat jednu ze dvou nabízených možností: [34]

- **Slovní odpovědi** – vyslovením jednoho ze dvou slov, typicky „ano“ a „ne“ v mateřském jazyce žáka.
- **Pohybem hlavy** – zavrtěním nebo kývnutím.
- **Jazykem těla** – úsměvem/zamračením, pohledem na lektora/pryč, sáhnutím na lektora/odstrčením.
- **Ukázáním symbolu v tabulce** – prstem, rukou, nohou, hlavou (typicky s náhlavním ukazovátkem), očima nebo jinak.
- **Aktivací přepínače/tlačítka** – opět prstem, rukou, nohou, hlavou nebo očima. V tomto případě je ale spojeno se strojovou hlasovou odezvou.
- **Otáčením zápěstí s náramkem** – náramek z jedné strany vyjadřuje kladnou (zelený nápis ANO) a z druhé strany zápornou (červený nápis NE) odpověď.
- **Ukázáním kartičky** – samostatná kartička pro „ano“ a pro „ne“, obdoba VOKS.
- **Ukázáním palcem** – nahoru pro ANO, dolů pro NE.
- **Sevřením pěsti** – sevřená pěst pro ANO, otevřená dlaň pro NE.
- **Pohybem očí** – pohledem doleva/doprava nebo nahoru/dolů.
- Modifikacemi a kombinacemi výše uvedených způsobů.

Tímto způsobem nelze komunikovat v plné šíři a děti se bez zvládnutí vyšší formy komunikace nenaučí dávat najevo cokoli, co by tímto způsobem bylo zdlouhavé nebo zcela nemožné sdělit. O komunikaci se pak snaží jen pokud něco hodně potřebují nebo chtějí. Cílem by tedy mělo být rozvinout komunikační schopnosti natolik, aby bylo možné přejít k o stupeň vyšší formě komunikace, jako je VOKS nebo komunikační tabulky. [22]

2.3.7 Kdy použít AAK

Před zavedením kteréhokoli alternativního způsobu komunikace je nutné zvážit všechny přednosti a omezení vyplývající z používání alternativních a augmentativních systémů komunikace.

Výhodou alternativních způsobů komunikace je, že rozšiřují komunikační možnosti, pomáhají rozvoji rozpoznávacích a jazykových dovedností a umožňují osobám s potížemi při vyjadřování aktivně se účastnit konverzace, v které by jinak byli jen opomíjenými posluchači. Snižují tak tendenci žáků k pasivitě a zvyšují aktivitu během vzdělávacích činností i ve volném čase. [53]

Nevýhodou alternativních způsobů komunikace naopak je, že jsou společensky méně uplatnitelné než mluvená řeč. Znesnadňují navíc integraci, protože vyžadují osvojení systémů všemi potenciálními účastníky komunikace – jejich uživatelé jsou izolováni od většiny společnosti, která tyto systémy neovládá. Používání alternativních komunikačních systémů navíc často vzbuzuje nechtěnou pozornost veřejnosti. Zavedení systému může být také považováno za důkaz, že dítě se nikdy nenaučí mluvit. [53]

2.4 Počítačová terapie

Počítačová terapie je název projektu, který zavádí i novou formu terapie pro osoby s mentálním postižením. Představuje novou formu výzkumu v oblasti informačních technologií cílenou na osoby s mentálním postižením. Autorem projektu je Ing. Jiří Fiala, vedoucí této práce. Projekt byl zformován počátkem roku 2013 v prostředí osob s mentálním postižením. Nyní pokračuje v zázemí Fakulty informačních technologií VUT v Brně za podpory společnosti Red Hat [30], odborné komunity iSEN [49] a neziskového sektoru zaměřeného na vzdělávání a terapie osob s mentálním postižením za pomocí prostředků ICT. [36][35]

Z pohledu programu formou terapie, který byl při projektu Počítačová terapie zaveden, se počítačovou terapií myslí využití dostupného potenciálu IT nástrojů a metod k rozvoji rozumových schopností osob s mentálním postižením a kombinovanými vadami (lehkým a středně těžkým mentálním postižením) ve smyslu kompenzace a reedukace. [36]

- **Kompenzace** (od *compensatio* – vzájemná náhrada) označuje vyvážení nedostatku vyvinutím jiné činnosti nebo zdůrazněním přednosti. [50]
- **Reedukace** (doslova nová výchova) označuje speciální metody upravující nebo napravující porušení nebo nevyvinutí funkce nebo činnosti. [50]

Obsahem Počítačové terapie jako programu formou terapie pak je rozvoj osob s mentálním postižením v několika základních oblastech s ohledem na jejich individuální vzdělávací i jiné potřeby. [36]

Přes tato sociálně pedagogická a sociálně pracovní východiska projektu se z informatického pohledu jedná o aplikovaný výzkum v oblasti softwarového inženýrství s využitím dalších přidružených informatických metod, zejména z oblasti interakce člověka s počítačem (HCI – „human–computer interaction“). [42] Práce je zaměřena na dosud nepokrytou oblast výzkumu ICT pro osoby s mentálním postižením. Ta je dlouhodobě v neuspokojivém stavu.

První nedostatky ve vývojovém procesu aplikací pro osoby s mentálním postižením byly identifikovány již na počátku roku 2013. Projekt počítačové terapie vychází ze zkušeností sociálních pedagogů převážně z odborné komunity iSEN [49], z odborných studií na toto téma, z kritického zhodnocení v současnosti dostupných aplikací a ze zkušeností vedoucího práce ze 4 let technického vedení osob s mentálním postižením v rámci programu na jejich podporu skrze prostředky ICT.

Pro účely speciální pedagogiky existuje množství specializovaných aplikací, které ale v mnoha ohledech nesplňují nároky cílové skupiny (mentálně postižených a neziskových organizací) nebo jim jsou cenově nedostupné. Některé prostředky IT (tablety iPad) jsou sice používány v rámci speciální pedagogiky na některých speciálních školách, žáci je však mohou využívat jen v rámci školní výuky, kde tráví jen malou část svého denního režimu. Ukazuje se jako nezbytné pro další pokrok umožnit používání těchto zařízení také v přirozeném sociálním prostředí žáků – za hranicemi škol. [36]

Za účelem identifikace nedostatků a požadavků na ICT prostředky pro tuto oblast byla vedoucím práce vedena stejnojmenná terapie v prostředí osob s mentálním postižením, která sloužila k testování a verifikaci navrhovaných řešení. [36]

2.4.1 Návrhové principy počítačové terapie

Jako jeden z dosud získaných výsledků v projektu počítačové terapie byly školitelem definovány návrhové principy, jejichž cílem je odstranění identifikovaných nedostatků a zajištění efektivního vývoje aplikací, které jsou cílem projektu. Cílem je také zajištění vyšší přístupnosti a použitelnosti, než jakou poskytují stávající řešení.

První z principů byly definovány v [36] a v průběhu výzkumu se stále vyvíjí a ověřuje se jejich prospěšnost. Pro potřeby této práce lze návrhové principy počítačové terapie stručně shrnout do následujících bodů: [36]

- **Multiplatformita:** Aplikace by měly být dostupné pro více mobilních platform. I jejich vývoj by ale měl být veden multiplatformním způsobem, aby jednou vytvořenou aplikaci bylo možné zkompileovat pro všechny podporované platformy.
- **Otevřenost:** Vývoj by měl být postaven na otevřených technologiích a zdrojové kódy aplikací by měly být veřejně zdarma dostupné, aby kdokoli mohl aplikace rozšiřovat, vylepšovat nebo na nich stavět další užitečné aplikace. Vývojáři by také měli být otevření začleňování takových vylepšení zpět do svých aplikací.
- **Rozšiřitelnost:** Vývoj by měl být realizován podle současných standardů. Aplikace by měly být implementovány ve vyšším programovacím jazyce, mít jasně definovanou architekturu a reflektovat zásady objektově orientovaného vývoje.
- **Stanovený cíl:** Aplikace by měly být vyvíjeny s definovaným cílem – např. kompenzovat nedostatečnost v řeči.
- **Použitelnost:** Aplikace by měly být snadno použitelné ke svému účelu i v běžném prostředí (sociálním prostředí) – mimo školu. Neměly by např. ke svému fungování vyžadovat internetové připojení.
- **Konfigurovatelnost:** Aplikace by měly umožňovat maximální přizpůsobení individuálním potřebám svých uživatelů. Měly by také podporovat přepínání mezi konfiguracemi více uživatelů.
- **Bezpečnost:** Aplikace by měly rozlišovat režim klienta (žáka), asistenta (sociálního pracovníka) a technického správce. Aplikace a její vývojáři by měly dbát na dodržování bezpečnostních zásad a zásad pro ochranu osob s mentálním postižením.
- **Přístupnost:** Aplikace by měly být cenově dostupné, aby si je mohli dovolit i méně majetní klienti a neziskové organizace.

Návrhové principy byly uplatněny při návrhu a implementaci první verze *Frameworku počítačové terapie* (i-CT frameworku), který měl být společným základem více aplikací pro osoby s mentálním postižením, které jsou dále označovány jako „nadstavbové aplikace“. Ty byly v první verzi systému s i-CT frameworkem nedílně propojeny. [36]

První verze i-CT frameworku s aplikacemi byly vytvořeny již v rámci dřívějších bakalářských a diplomových prací – viz [37][45][44][43].

Předmětem této práce je nová implementace i-CT frameworku, která je vyžadována současným stavem výzkumu, a vývoj nad ní postavené nadstavbové aplikace pro AAK typu ANO/NE. Nový i-CT framework má zajišťovat správu databáze uživatelů aplikací, která zahrnuje také informace o edukaci, terapii a sociální péči jednotlivých mentálně postižených klientů. Nadstavby tak nebudou zatíženy nutností správy a ukládání těchto informací.

2.4.2 Aplikace pro komunikaci typu ANO/NE

Aplikace pro komunikaci typu ANO/NE by měla být elektronickou náhradou pomůcek popsaných v kapitole 2.3.6 věnované právě komunikaci typu ANO/NE. Oproti papírovým a jiným pomůckám by však aplikace dostupná na zařízení s dotykovou obrazovkou měla přinášet přidanou hodnotu v podobě grafické vizualizace obou možností, z nichž si mentálně postižená osoba volí odpověď, a zvukové odezvy po výběru možnosti.

Kapitola 3

Multiplatformní vývoj mobilních aplikací

3.1 Platforma Android

Android je mobilní platforma určená pro široké spektrum mobilních zařízení od telefonů, přes tablety až po televize. Typicky se jedná o zařízení ovládaná skrze dotykovou obrazovku, čemuž je podřízen návrh jeho uživatelského rozhraní. Přesto je možné k ovládní využít také klasickou myš a klávesnici. (typicky v případě televizí) [6]

Systém Android je tvořen sadou open source softwarových produktů vyvíjených společností Google. Součástí nejdůležitější z pohledu této práce popisují následující kapitoly. [6]

3.1.1 Dalvik a Android Runtime

Dalvik je virtuální stroj Javy (JVM – Java Virtual Machine), který na platformě Android zajišťuje běh uživatelských aplikací. Od verze Androidu 5.0 je nahrazen běhovým prostředím Android Runtime (ART). [33]

Aplikace pro platformu Android jsou typicky vytvářeny v jazyce Java. Zdrojové soubory tříd v jazyce Java jsou překládány do Dalvik bajtkódu, který je ukládán do souboru `classes.dex` (classes – třídy, DEX – Dalvik Executable) v kořeni balíčku aplikace (APK – Android Package). Společně s ostatními soubory tvořícími obsah balíčku (pomocnými soubory jako jsou obrázky) je při instalaci aplikace nakopírován do adresáře aplikace. [33]

Při spuštění aplikace je vytvořen nový proces virtuálního stroje (Dalvik/ART) a v jeho rámci je spuštěna samotná aplikace. Zatímco v rámci běhového prostředí Dalvik je bajtkód Dex nejprve zkompileován do bajtkódu Odex který je následně Dalvikem interpretován, v rámci běhového prostředí ART je bajtkód Dex zkompileován přímo do nativního kódu a přímo vykonáván. Rozdíl mezi JVM Dalvik a Android Runtime (ART) je z pohledu této práce nepodstatný. Pro zjednodušení bude proto v dalším textu uváděn jen Dalvik. [33]

3.1.2 Aplikační rámec platformy Android

Aplikační rámec poskytuje služby aplikacím běžících v rámci platformy a zjednodušuje tak jejich vývoj. Aplikacím běžících v rámci JVM Dalvik umožňuje komunikovat se systémovými službami platformy. Jednotlivé systémové služby pak zpřístupňují různé komponenty platformy, jako je správce oken nebo správce upozornění (notifikací). [5]

Samotný aplikační rámec běží v rámci JVM Dalvik. Systémové služby naproti tomu běží v samostatných procesech operačního systému. Komunikují proto skrze Mechanismus meziprocesní komunikace Binder (Binder Inter-Process Communication). [5]

3.1.3 Aktivity a Intenty

Aktivita (Activity) je komponenta aplikace, která poskytuje jednu obrazovku uživatelského rozhraní. Aktivita může běžet na celou obrazovku (mimo nebo včetně systémového panelu) nebo ve vyskakovacím okně nad jinou aktivitou. Aplikace na platformě Android se zpravidla skládají z více aktivit. [1]

Aktivity se mohou vzájemně spouštět a to i napříč aplikacemi. K tomuto účelu se využívá tzv. Intent. Intent je objekt zprávy, která může být zaslána mezi komponentami aplikací. V tomto případě je zpráva zaslána aktivitě, která má být spuštěna. [2]

Intenty se podle způsobu stanovení příjemce zprávy dělí na explicitní a implicitní. Explicitní Intent má příjemce jednoznačně určeného skrze objekt třídy aktivity/služby (objekt třídy `Class` odpovídající třídě aktivity/služby, která má být instanciována). Implicitní Intent naproti tomu stanovuje pouze akci, která se má provést, a příjemce zprávy je vyhledán prostřednictvím Intent-filtru. Intent-filtr je součástí definice aktivity/služby, která popisuje akce, které aktivita/služba umožňuje provést. [2]

Pro návrat ze vzájemného volání aktivit se využívá **návratový zásobník** („back stack“). Při startu nové aktivity je nová aktivita vložena na vrchol návratového zásobníku. Při opouštění aktivity (např. tlačítkem Zpět) je aktivita ze zásobníku odstraněna a na popředí je zobrazena aktivita, která je novým vrcholem zásobníku. [3]

Systém Android je navíc víceúlohový. To znamená že uživatel se může přepínat mezi více současně spuštěnými úlohami. Úloha představuje skupinu vzájemně se volajících aktivit. Každá úloha přitom disponuje vlastním návratovým zásobníkem. Vzájemné volání aktivit je proto nezávislé na mechanismu přepínání úloh. Výjimkou je případ, kdy je za pomoci Intentu spuštěna aktivita do nové úlohy. [4][3]

Příklad: Uživatel spustí aktivitu A, z ní přejde do aktivity B a z ní do nové úlohy spustí aktivitu C. Skrze funkci přepínání úloh se bude uživatel schopný přepínat jen mezi aktivitami B a C. Použitím tlačítka zpět v rámci aktivity B se vrátí do aktivity A. Použitím tlačítka zpět v rámci aktivity C se vrátí na domovskou obrazovku systému (launcher).

3.2 Platforma iOS

iOS je operační systém společnosti Apple pro mobilní zařízení (telefony, tablety apod.) založený na operačním systému Macintosh OS X – operačním systému společnosti Apple pro stolní počítače.

Oproti platformě Android aplikace platformy iOS jsou typicky kompilovány do nativního kódu. Jazykem zdrojových souborů přitom typicky jsou Objective-C a nově také Swift. Aplikace mohou využívat aplikační programové rozhraní (API) Cocoa a Cocoa Touch.

3.2.1 Cocoa a Cocoa Touch

Cocoa a Cocoa Touch jsou prostředí pro běh aplikací pod OS X a iOS. Cocoa zahrnuje rámce Foundation a AppKit a využívá se pro vytváření aplikací na OS X. [11] Cocoa Touch zahrnuje rámce Foundation a UIKit a využívá se pro vytváření aplikací na iOS. [18]

Rámec Foundation poskytuje základní třídy pro vývoj v Objective-C – kořenový objekt `NSObject`, který je společnou nadtrídou všech tříd v Objective-C, třídy reprezentující primitivní datové typy a kolekce. Poskytuje přístup k systémovým objektům a službám – portům, vláknům, zámkům a procesům. Poskytuje také podporu: [11]

- Internacionalizace (přizpůsobování aplikace jednotlivým národním zvyklostem, zejména překládání uživatelského rozhraní do různých jazyků)
- Persistence objektů (ukládání stavu objektů do databáze)
- Správy souborů
- Zpracování XML

Rámec `UIKit` naproti tomu poskytuje infrastrukturu pro implementaci grafické, událostmi řízené aplikace pro prostředí iOS. Zahrnuje správu událostí v hlavní běhové smyčce, podporuje rozdělení aplikace podle architektury Model-View-Controller (MVC), nástroje pro práci se službami Apple a nástroje pro vývoj grafického uživatelského rozhraní. [18]

3.2.2 Objective-C a Swift

Objective-C je původní primární programovací jazyk pro vývoj aplikací na OS X a iOS. Poskytuje schopnosti dynamického objektově orientovaného jazyka. Přitom zachovává plnou kompatibilitu s jazykem C, z kterého vychází. Oproti jazyku C zahrnuje podporu pro definici tříd, metod, pro objektovou práci s grafy a objektovými literály. [29]

Swift je nový programovací jazyk pro vývoj aplikací na iOS, OS X, watchOS a tvOS. Vychází z jazyků C a Objective-C, přičemž představuje zjednodušenou správu paměti založenou na automatickém počítání referencí na objekt. Swift přejímá snadnou čitelnost pojmenovaných parametrů dynamického objektového modelu Objective-C. [31]

Vzhledem k zaměření práce na multiplatformní vývoj bude nadále uvažován pouze jazyk Objective-C, se kterým počítají rámce pro multiplatformní vývoj mobilních aplikací. (viz kapitola 3.3 a 3.4)

3.3 Rámec Apache Cordova

Apache Cordova je otevřený rámec (framework) pro vývoj multiplatformních mobilních aplikací. Dovoluje využívat webové technologie (HTML5, CSS3 a Javascript) k tvorbě mobilních aplikací bez potřeby učit se programovací jazyky specifické jednotlivým platformám. [9]

Samotná aplikace je implementována jako webová stránka využívající kaskádové styly (CSS), Javascript, obrázky a další soubory typické pro vývoj webových aplikací. Tato aplikace je spuštěna v rámci **WebView** – zjednodušeného webového prohlížeče. [9]

Cordova zajišťuje vytvoření ikony aplikace v menu aplikací dané platformy. Po aktivaci ikony je spuštěn zjednodušený prohlížeč bez ovládacích prvků a v něm webová stránka aplikace. Mimo systémového panelu (který je běžně vidět i nad nativními aplikacemi) vyplňuje webová stránka aplikace celou obrazovku zařízení. Nejsou zobrazeny ovládací prvky typické pro webový prohlížeč, jako je adresní řádek.

Javascriptu běžícímu v rámci `WebView` je mimo API dostupných v rámci HTML5 přístupné také API umožňující přístup k funkcím zařízení, které jsou jinak webovým aplikacím

nepřístupné. Tato API musí být utvářena používanými Cordova pluginy – částmi aplikace vytvořenými v nativním kódu platformy. [9]

Pluginy typicky zpřístupňují části API platformy aplikaci v Javascriptu. Díky tomu jsou ale samy platformě závislé. Plugin který má být dostupný na více platformách musí být vytvořen pro každou platformu zvlášť. V praxi tak často narážíme na pluginy podporující jen několik málo platforem (v krajním případě jen jednu), bez kterých se musíme obejít, pokud nejsme schopni sami doplnit implementaci pro ostatní platformy které potřebujeme.

Pluginy rámce Cordova se využívají také v rámcích založených na Cordova – zejména v Oracle MAF, který je popsán v kapitole 3.4, a v Ionic, který je popsán v kapitole 3.5.

3.4 Oracle Mobile Application Framework

Oracle Mobile Application Framework (MAF) je hybridní platforma pro vývoj mobilních aplikací. Snaží se vést aplikace k architektuře Model-View-Controller (MVC) přenesením technologie Java Server Faces (JSF) známé z platformy Java EE do prostředí mobilních aplikací.

Java Server Faces je aplikační rámec poskytující stavební komponenty pro vývoj mobilních aplikací nad platformou Java Servlet. Těmito stavebními kameny jsou: [19]

- Základní komponenty webového uživatelského rozhraní
- Navigace mezi stránkami (s přechody definovanými mimo kód samotných stránek)
- Validace uživatelského vstupu a ošetření chyb a událostí
- Správa Java Bean (správa objektů spojených s webovým uživatelským rozhraním)
- Podpora internacionalizace (zobrazování uživatelského rozhraní ve více různých jazycích a s různými formáty čísel, data a času)

Část View je tvořena šablonami HTML stránek tvořících uživatelské rozhraní aplikace. Část Model, zaměřená na byznys logiku (práci s databází a komunikaci se serverem), a Controller jsou tvořeny kódem tříd v Javě. [24]

Zatímco v případě platformy Java EE běží klientská část aplikace ve webovém prohlížeči klienta a zbytek aplikace na straně serveru, v rámci Java EE aplikačního serveru, v případě mobilní aplikace je nutné aby celá aplikace běžela přímo na mobilním zařízení.

Roli webového prohlížeče hraje v případě Oracle MAF komponenta WebView běžící v rámci rámce Apache Cordova (viz kapitola 3.3). Prvky uživatelského rozhraní jsou zde zobrazovány na základě HTML kódu generovaného serverovou částí aplikace.

Roli aplikačního serveru hraje v případě Oracle MAF zjednodušený aplikační server, který je nedílnou součástí rámce Oracle MAF. Ten ke svému běhu (a také běhu samotných provozovaných aplikací) vyžaduje virtuální stroj Javy. Ten je sice součástí platformy Android, ale není součástí platformy iOS. Virtuální stroj Javy (Java Virtual Machine – JVM), který je nezbytný k provádění kódu v Javě, je proto přibalen do každého balíčku aplikace v Oracle MAF.

Tento virtuální stroj je sám v nativním kódu platformy. Jeho přibalení se negativně projevuje na velikosti balíčku aplikace – balíček i velmi jednoduché aplikace běžně zabírá více než 30 MB. JVM je navíc přibalen i k balíčkům platformy Android, přestože zde je k dispozici JVM Dalvik. To aby bylo zajištěno identické běhové prostředí nad oběma platformami. [24] [23]

Aplikace tak mohou obsahovat dva druhy kódu v Javě – kód části Cordova pluginu specifického platformě Android, který je prováděn přímo v rámci platformy, a kód prováděný v JVM přibalené k aplikaci. Nevýhodou platformy Oracle MAF je, mimo nadměrné velikosti balíčku s aplikací, také komerční licence – zdrojové kódy většiny součástí nejsou veřejně k dispozici a některé případy použití jsou podmíněny placenou verzí.

3.5 Ionic

Ionic je rámec postavený nad platformou Apache Cordova/PhoneGap a rámcem AngularJS. Platforma Apache Cordova poskytuje prostředky pro vytváření mobilních aplikací z webových aplikací a byla již blíže popsána v kapitole 3.3.

AngularJS je naproti tomu rámec pro strukturování dynamických webových aplikací. Podporuje strukturování aplikace v Javaskriptu po vzoru architektury Model-View-Controller (MVC). Na rozdíl od Oracle MAF ale staví výhradně na webových technologiích (HTML a Javaskriptu). [7]

Ionic je navíc sám o sobě CSS rámcem a Javaskriptovou knihovnou pro tvorbu uživatelského rozhraní. Umožňuje tak za pomoci webových technologií snadno vytvářet uživatelské rozhraní vhodné pro mobilní zařízení.

Ionic tedy celkově poskytuje balík nástrojů usnadňujících tvorbu mobilních aplikací stejně jako rámec Oracle MAF. Základem je Apache Cordova zajišťující vytvoření samotného balíku s webovou aplikací ve WebView. Strukturování aplikace podporuje rámec AngularJS. Uživatelské rozhraní je možné tvořit z CSS a Javaskriptových komponent podobných komponentám Oracle MAF. Stejně jako Oracle MAF, ani Ionic sám o sobě neposkytuje podporu pro objektově-relační mapování nebo jinou vysokoúrovňovou práci s databází.

3.6 Zamezení opuštění aplikace

Předpokládá se, že aplikace bude nasazena v prostředí práce s mentálně postiženými lidmi, vedené sociálními pracovníky, speciálními pedagogy a asistenty pedagogů. V tomto prostředí se považuje za nezbytné zamezit, aby mentálně postižení uživatelé opustili prostředí aplikace nebo dokonce zasáhli do konfigurace zařízení.

Postupy jak opuštění aplikace uživatelem zamezit jsou často kontroverzní, neboť mohou být snadno zneužity autory škodlivých aplikací. Budeme se přesto snažit nalézt postupy co nejvíce univerzální, fungující na většině zařízení a neaspírující na potlačení v budoucích verzích operačních systémů pro mobilní zařízení.

3.6.1 Potlačení tlačítek Zpět a dalších na platformě Android a Cordova

```
public class MyActivity extends Activity {
    @Override
    public void onBackPressed() {
        // nothing here to disable Back button
    }
}
```

Ukázka kódu 3.1: Potlačení tlačítka Zpět v kódu aktivity

Tlačítko Zpět umožňuje uživateli opustit běžící aktivitu a vrátit se do předchozí aktivity. Tou je aktivita, která bude na vrcholu návratového zásobníku aktuální úlohy po odstranění opouštěné aktivity. [3]

Aktivita však může nejen upravovat svůj návratový zásobník, ale i zcela změnit chování navázané na tlačítko Zpět a původní zcela odstranit. Na úrovni kódu aktivity v Javě k tomuto stačí přepsat metodu `onBackPressed()`, jako v ukázce kódu 3.1. [3]

Potlačení tlačítka Zpět není problematické ani v případě aplikace postavené nad rámcem Cordova, jenž byl představen v kapitole 3.3. Stačí odchytit událost `backbutton`, jako v ukázce kódu 3.2. Použití metody `preventDefault()` události zabrání provedení původní akce navázané na událost – zabrání opuštění aktivity/stránky. [8]

```
document.addEventListener("deviceready", function(){
    document.addEventListener("backbutton", function(event){
        event.preventDefault();
    }, false);
}, false);
```

Ukázka kódu 3.2: Potlačení tlačítka Zpět v rámci Cordova

Nutno poznamenat, že přidání funkce reagující na událost je možné úspěšně provést až po úplném načtení stránky a inicializaci rámce Cordova. Již v rámci uvedené ukázky je proto přidání provedeno až v rámci funkce reagující na událost úplného načtení rámce Cordova.

Stejným způsobem je možné potlačit také další hardwarová tlačítka: [8]

- tlačítko Hledat (událost „`searchbutton`“)
- tlačítka zahájení/ukončení hovoru („`startcallbutton`“ a „`endcallbutton`“)
- tlačítka zvýšení/snížení hlasitosti („`volumeupbutton`“ a „`volumedownbutton`“)

Tímto způsobem lze tedy blokovat drtivou většinu hardwarových tlačítek. na platformě Cordova i Oracle MAF. Výjimkou je Tlačítko napájení, tlačítko Domů a tlačítko Menu (známé také jako tlačítko přepínání úloh).

3.6.2 Potlačení tlačítka Domů na platformě Android

Tlačítko Domů (Home) umožňuje opustit běžící aplikaci a vrátit se na domovskou obrazovku systému (tzv. launcher).

Opuštění běžné aplikace tlačítkem Domů není možné na platformě Android účinně zabránit. Vzhledem k již zmíněné možnosti zneužití škodlivými aplikacemi není ani v zájmu vývojářů platformy tento nedostatek odstranit. Omezení lze ale obejít nastavením vlastní aplikace jako domovské obrazovky – uživateli se tak po použití tlačítka Domů zobrazí (respektive zůstane zobrazená) naše aplikace. [25]

Domovská obrazovka systému je na platformě Android realizována běžnou uživatelskou aplikací, přesněji řečeno, jednou její aktivitou. Ta obvykle umožňuje uživateli spouštění ostatních aplikací, které spouští za pomoci Intent požadavku v rámci nových úloh. Za pomoci menu pro přepínání úloh pak je možné přepínat mezi všemi a aktivitami spuštěnými z domovské obrazovky, případně dalšími které byly spuštěny v samostatné úloze.

Aby bylo možné nastavit vlastní aktivitu jako domovskou obrazovku systému, je nutné této aktivitě přiřadit Intent-filtry ukázané v ukázce kódu 3.3.

```
<intent-filter>
  <action android:name="android.intent.action.MAIN" />
  <category android:name="android.intent.category.DEFAULT" />
  <category android:name="android.intent.category.HOME" />
</intent-filter>
```

Ukázka kódu 3.3: Intent-filtry dovolující aktivitě stát se domovskou obrazovkou [46]

Budou-li se navíc všechny ostatní aktivity spouštět v rámci stejné úlohy (tedy bez Intent příznaku `FLAG_ACTIVITY_NEW_TASK`), potlačíme tlačítko Menu zcela. Tlačítko Menu je totiž ignorováno, jestliže se aktivita domovské obrazovky již nachází na návratovém zásobníku aktuální úlohy. (viz kapitola 3.1.3)

Nevýhodou tohoto způsobu blokování tlačítka Domů je nutnost skutečně nahradit domovskou obrazovku systému. Není možné jej zablokovat v rámci běžné aplikace, bez takto drastického zásahu do systému. Zařízení při každém svém spuštění bude startovat přímo na naši domovskou obrazovku. Toto řešení je tedy přinejmenším problematické, pokud má dané zařízení sloužit i k jiným účelům. Tento nedostatek lze z části odstranit umožněním spuštění původní domovské obrazovky z naší upravené.

3.6.3 Znepřístupnění systémových nabídek na platformě Android

I po potlačení všech klíčových hardwarových tlačítek je na většině verzí systému Android uživatel schopný dostat se do alespoň jedné systémové nabídky:

- Podržetím tlačítka napájení je možné zobrazit **Vypínací dialog**, který často mimo vypnutí/restartu zařízení umožňuje také zapínat/vypínat Režim letadlo (odpojit zařízení od sítě GSM), zapínat/vypínat Tichý režim (vypnout zvuky zařízení) nebo dokonce restartovat zařízení do režimu pro obnovu systému (Recovery mode).
- Od verze Android 4.4 je možné zobrazit **Stavový panel (Statusbar)** i pokud je běžící aplikace v režimu celé obrazovky – gestem tažení od horního okraje obrazovky. Odtud je pak možné se dostat až do nastavení zařízení.
- Tlačítkem přepínání úloh je možné zobrazit **Nabídku úloh**. I pokud zajistíme že naše úloha bude jediná běžící (k čemuž postačuje omezení z kapitoly 3.6.2), stále tato nabídka překrývá aplikaci a je proto žádoucí zabránit jejímu zobrazení.

Ani jedné z událostí není možné účinně zabránit bez tzv. rootnutí zařízení. To umožní změnit jinak nepřístupnou konfiguraci systému, skrze vývojářskou konzoli (`adb shell`) nebo instalací aplikace označené jako systémové, tedy opatřené vyššími oprávněními. Rootnutí zařízení ale znamená ztrátu záruky u většiny výrobců zařízení, proto je žádoucí vyřešit znepřístupnění nabídek jinak.

Alternativním způsobem řešení, které nevyžaduje žádná zvláštní oprávnění, je zasílání všesměrového Intent požadavku `ACTION_CLOSE_SYSTEM_DIALOGS`. Ten dovoluje zavřít všechny aktuálně zobrazené systémové dialogy.

Současně je možné navázat libovolnou akci na událost ztráty zaměření (focus) okna aktivity aplikace. Je tak možné uzavřít všechny zobrazené systémové dialogy v reakci na přesun okna naší aktivity do pozadí, ke kterému při každém zobrazení systémového dialogu dochází.

```
public void onWindowFocusChanged(boolean hasFocus) {
    super.onWindowFocusChanged(hasFocus);
    if(!hasFocus) {
        Intent closeDialog = new Intent(Intent.ACTION_CLOSE_SYSTEM_DIALOGS);
        sendBroadcast(closeDialog);
    }
}
```

Ukázka kódu 3.4: Metoda Aktivita zavírající systémové dialogy při jejich objevení [16]

Ukázka kódu 3.4 ukazuje způsob integrace zasílání uvedeného Intent požadavku v kódu aktivity. Požadavek je zasílán v rámci metody `onWindowFocusChanged()`, která je volána při změně zaměření (focus). Ztratí-li aktivita zaměření (dostane se do pozadí), aktivita zareaguje zasláním uvedeného Intent požadavku a všechny zobrazené systémové dialogy tak zavře.

Přestože tak není zabráněno zobrazení uvedených systémových dialogů, tím že jsou po svém zobrazení ihned uzavřeny je zabráněno využití jejich ovládacích prvků k opuštění aplikace nebo změně nastavení zařízení.

Kapitola 4

Analýza požadavků

Cílem kapitoly je shrnout požadavky kladené na implementaci vyplývající z poznatků z předcházejících kapitol a navrhnout jejich řešení.

4.1 Požadavky na i-CT framework

První částí implementační části práce je část i-CT frameworku. Ten by měl poskytovat společný základ pro jednotlivé výukové aplikace, poskytovat společnou databázi klientů a asistentů a umožňovat spravovat záznamy o klientech.

Na základě návrhových principů Počítačové terapie (přehledově uvedených v kapitole [2.4.1](#)) byly požadavky na i-CT framework ve spolupráci s vedoucím práce definovány následovně:

- Systém (i-CT framework a aplikace) bude rozlišovat tři úrovně uživatelů – administrátory, asistenty a klienty.
- Asistenti budou moci spravovat všechny informace o svých klientech, o svých podřízených asistentech a jejich klientech. Jen administrátoři budou moci spravovat všechny klienty. Důvodem je ochrana klientů před zneužitím záznamů o nich.
- Asistenti budou moci svoje klienty nejen spravovat, ale také zakládat nové.
- Informace o klientech se skládají z identifikačních údajů, zdravotního profilu, vzdělávacího profilu a sociálního profilu.
- Každý profil se skládá z 0 až N záznamů. Každý záznam popisuje stav klienta ve specifikovaném čase. Mimo informací o aktuálním stavu klienta tak jsou uchovávány také záznamy o jeho stavech v minulosti.
- Záznam zdravotního profilu popisuje zdravotní problém klienta a odpovídající anamnézu.
- Záznam sociálního profilu popisuje vztahy klienta s lidmi v jeho okolí a dále zejména problémy klienta se stravováním a hygienou.
- Záznam vzdělávacího profilu popisuje speciální potřeby klienta a jeho kompetence (znalosti, dovednosti a postoje).
- Při vytváření záznamu profilu bude možné záznam předvyplnit z jiného záznamu stejného nebo jiného klienta, ke kterému mu daný uživatel oprávnění.

- Asistent bude moci nastavit restriktce (maximální čas souvislé práce s aplikací) a přizpůsobení (velikost písma), která se budou uplatňovat na jednotlivé uživatele pro každou aplikaci zvlášť.
- i-CT framework bude poskytovat nástroj pro spouštění výukových aplikací, kterým bude poskytovat informaci o přihlášeném uživateli včetně jeho restrikcí a přizpůsobení pro danou aplikaci.
- i-CT framework po opuštění výukové aplikace zaznamená do profilu klienta informace předané z výukové aplikace.

4.2 Stávající aplikace pro AAK

Součástí analýzy požadavků je také analýza a srovnání stávajících aplikací pro AAK, zejména ve vztahu k návrhovým principům Počítačové terapie. Srovnání je primárně zaměřeno na aplikace pro komunikaci typu ANO/NE, jejíž vytvoření je součástí implementační části této práce.

4.2.1 Klábosil

Klábosil je česká aplikace pro AAK pro platformu iOS. Byla vytvořena v rámci projektu „Sociální integrace žáků s těžkým zdravotním postižením v Poděbradech“ (CZ.1.07/1.2.05/04.0009) z prostředků ESF (Evropského sociálního fondu) a státního rozpočtu České republiky. Uživatelům je dostupná zdarma prostřednictvím distribuční platformy Apple App Store. [20]



Obrázek 4.1: Hlavní obrazovka aplikace Klábosil

Aplikace je elektronickou variantou komunikačních tabulek představených v kapitole 2.3.3. Symboly jsou zde tříděny do hierarchicky zanořovaných kategorií a symboly i kategorie lze volně přidávat a upravovat. Po výběru symbolu je symbol přečten za pomoci hlasového syntetizátoru a umístěn na větný řádek – vrchní panel aplikace, kde se za sebe umísťují vybrané symboly v pořadí, v němž byly vybrány. Kliknutím na větný řádek je možné nechat přečíst popisky symbolů na větném řádku jako jednu větu.

Aplikaci ve srovnání vyzdvihuje právě hlasový syntetizátor, který věty předčítá korektní češtinou. To je hlavní výhoda aplikace ve srovnání s obdobnými konkurenčními aplikacemi, které nejsou primárně zaměřeny na české uživatele.

Aplikace v původní verzi využívala hlasový syntetizátor „Eliška“, který byl vyvíjený společností Acapela. Vzhledem k tvrdým podmínkám použití (při použití hlasu Eliška by nebylo možné poskytovat aplikaci trvale zdarma) byl v pozdějších verzích nahrazen. Aplikace nyní využívá hlasový syntetizátor vyvíjený Západočeskou univerzitou v Plzni, který je v současnosti dostupný v ženské a mužské variantě hlasu. [14]

Splnění návrhových principů počítačové terapie

Aplikace splňuje návrhové principy počítačové terapie v následujících hlediscích:

- **Stanovený cíl:** Aplikace je vyvíjena s definovaným cílem – kompenzovat nedostatečnost řeči.
- **Použitelnost:** Aplikace je snadno použitelná i v běžném prostředí – mimo školu.
- **Přístupnost:** Aplikace je k dispozici zdarma.

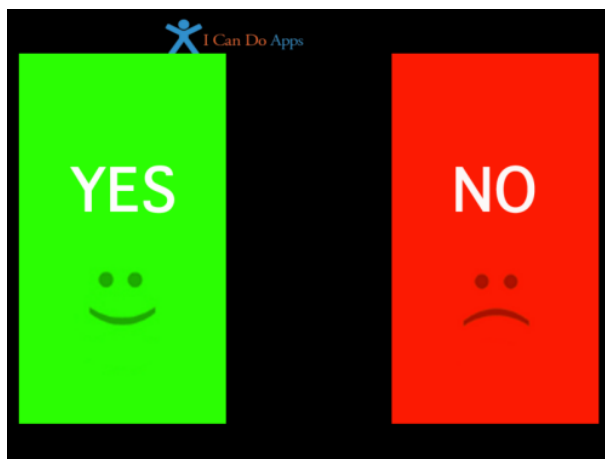
Aplikace ale na druhu stranu nespĺňuje následující hlediska:

- **Multiplatformita:** Aplikace podporuje výhradně platformu iOS. Není multiplatformní.
- **Otevřenost:** K aplikaci nejsou volně dostupné zdrojové kódy ani není postavena na otevřených technologiích.
- **Konfigurovatelnost:** Aplikace sice umožňuje přizpůsobovat dostupné symboly a jejich kategorie, neumožňuje ale přepínat mezi konfiguracemi pro více různých uživatelů.
- **Bezpečnost:** Aplikace umožňuje znepřístupnit obrazovku pro správu symbolů. Ostatní možnosti znepřístupnění ponechává na funkci Asistovaný přístup platformy iOS, což by nebylo na závadu. Aplikace má ale zcela nechráněnou historii všech vyslovených vět, kterou lze navíc mazat jen po jednotlivých větách.

4.2.2 Yes/No (I Can Do Apps)

Aplikace je označena jako vzdělávací nástroj vyvinutý ve spolupráci s odborníky na vady řeči. [17] Sestává ale z jediné obrazovky s tlačítky Ano a Ne, které aktivují příslušnou hlasovou odezvu. Jediným rozšířením placené verze aplikace (Yes/No Data) je počítání počtu stisků jednotlivých tlačítek.

Aplikace vůbec neumožňuje vytvářet vlastní páry ani jinak přiřazovat význam tlačítkům. Nepodporuje tak komunikaci typu ANO/NE v plné šíři, kde (navzdory názvu) nemusí být párem (dvěma možnostmi mezi kterými se vybírá) právě Ano a Ne. Aplikace také není přeložená do českého jazyka, což je v českém prostředí v této kategorii softwarových produktů problém.



Obrázek 4.2: Hlavní a jediná obrazovka aplikace Yes/No společnosti I Can Do Apps

Splnění návrhových principů počítačové terapie

Aplikace splňuje návrhové principy počítačové terapie v následujících hlediscích:

- **Stanovený cíl:** Aplikace je vyvíjena s definovaným cílem – kompenzovat nedostatečnost a podporovat rozvoj komunikačních dovedností.
- **Použitelnost:** Aplikace je snadno použitelná i v běžném prostředí – mimo školu.
- **Bezpečnost:** Aplikace neuchovává žádná data která by bylo nutné chránit.
- **Přístupnost:** Aplikace je k dispozici zdarma a i její placená verze je velmi levná (kolem 1 €).

Aplikace ale nespĺňuje následující hlediska:

- **Konfigurovatelnost:** Aplikace neumožňuje tvořit vlastní páry pro jednotlivé uživatele.
- **Multiplatformita:** Aplikace podporuje výhradně platformu iOS. Není multiplatformní.
- **Otevřenost:** K aplikaci nejsou volně dostupné zdrojové kódy.

4.2.3 YesNoCommunication (Počítačová terapie)

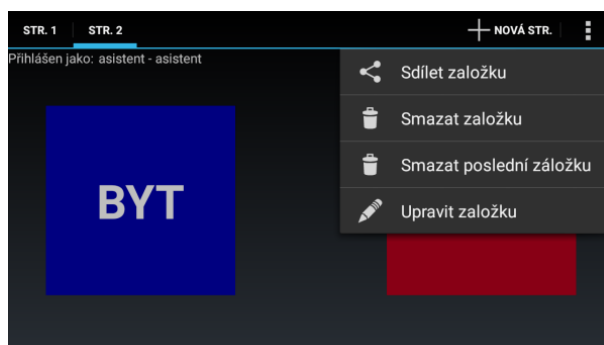
Aplikace YesNoCommunication je aplikací pro AAK pro platformu Android a iOS. Byla vytvořena v rámci projektu počítačové terapie v rámci bakalářské práce Bc. Jana Tůmy vedené Ing. Radkem Kočím, Ph.D. a dále vedoucím této práce jako školitelem specialistou. [44] Jedná se o aplikaci pro alternativní komunikaci typu ANO/NE, jejíž princip byl již nastíněn v kapitole 2.4.2.

Kombinace dvou možností ze kterých se v rámci jedné obrazovky vybírá se označuje jako pár. Obou možnostem může být nastaven nápis a obrázek, které se objeví na tlačítku a zvukový záznam, který se přehraje při stisku tlačítka. Páry jsou sdruženy do karet – každý pár je obsažen v jedné kartě a karta může obsahovat libovolný počet párů.

Aplikace rozlišuje tři úrovně uživatele:



Obrázek 4.3: Přehrávání karty v aplikaci YesNoCommunication.



Obrázek 4.4: Úprava karty v aplikaci YesNoCommunication.

- Administrátor může být v systému jen jeden a může spravovat všechny asistenty, klienty i karty.
- Asistenti mají přidělené klienty, jejichž karty mohou spravovat i přehrávat. Mohou také přidávat své další klienty.
- Klienti se obvykle sami nepřihlašují, ale pokud ano, mohou jen přehrávat své karty.

Splnění současných návrhových principů počítačové terapie

Aplikace splňuje návrhové principy počítačové terapie v následujících hlediscích:

- **Multiplatformita:** Aplikace je dostupná pro platformu Android i iOS, přestože se technicky vzato jedná o dvě nezávisle vyvíjené aplikace.
- **Otevřenost:** K aplikaci jsou volně dostupné zdrojové kódy.
- **Stanovený cíl:** Aplikace je vyvíjena s definovaným cílem – kompenzovat nedostatečnost a podporovat rozvoj komunikačních dovedností.
- **Použitelnost:** Aplikace je snadno použitelná i v běžném prostředí – mimo školu.
- **Konfigurovatelnost:** Aplikace umožňuje vytvářet vlastní páry a sdružovat je do karet a to pro každého uživatele aplikace zvlášť.
- **Přístupnost:** Aplikace je k dispozici zdarma.

Aplikace ale nespĺňuje následující hlediska:

- **Rozšiřitelnost:** Aplikaci je nutno udržovat a rozšiřovat ve dvou samostatných provedeních pro jednotlivé platformy. Všechny databázové operace nad všemi objekty jsou prováděny v rámci metod jedné třídy. Aplikace nereflktuje současné standardy pro vývoj aplikací.
- **Bezpečnost:** Aplikace sice zpřístupňuje karty klientů jen oprávněným uživatelům, umožňuje ale klientovi, jemuž přehrávání párů spustil jeho asistent, opustit kdykoli přehrávání a dostat se k datům jiných klientů nebo manipulovat s konfigurací zařízení mimo aplikaci.

4.2.4 Zhodnocení stávajících aplikací pro AAK

Z analýzy stávajících aplikací pro AAK vyplývá, že žádná z nalezených aplikací nespĺňuje všechny aktuálně definované požadavky vyplývající z principů Počítačové terapie. Potvrzují tak závěry z kapitoly 2.4, podle kterých nejsou na trhu dostupné vyhovující nástroje pro augmentativní a alternativní komunikaci, zejména ne pro komunikaci typu ANO/NE.

4.3 Návrh způsobu splnění požadavků

Tato podkapitola popisuje návrh způsobu splnění požadavků daných principy Počítačové terapie.

- **Multiplatformita:** Aplikace budou vyvíjeny za pomoci rámce podporujícího multiplatformní vývoj – rámec Oracle MAF nebo rámec Ionic.
- **Otevřenost:** Repozitář verzovacího systému bude veřejně umístěn na některém z k tomu určených veřejných úložišť a vývojáři aplikací budou dbát na začleňování vylepšení vzešlých z open source komunity.
- **Rozšiřitelnost:** Aplikace budou navrženy po vzoru architektury MVC, budou objektově navržené a implementované a to s důrazem na efektivitu a zkvalitnění vývojového procesu.
- **Stanovený cíl:** Před vytvořením aplikace bude nejprve definován její cíl.
- **Použitelnost:** Aplikace bude možné používat off-line – bez internetového připojení.
- **Konfigurovatelnost:** Aplikace budou umožňovat maximální přizpůsobení potřebám klienta – zejména budou umožňovat asistentům klientů vytvářet nové páry/sady piktoqramů a přiřazovat je zadaným klientům.
- **Bezpečnost:** Aplikace budou rozlišovat administrátory, asistenty a klienty. Klienti budou moci jen spouštět výukové aplikace které jim jejich asistent zpřístupní, asistenti budou moci pracovat s daty jen svých klientů a jen administrátoři budou mít oprávnění neomezená. Aplikace neumožní klientům zasáhnout do konfigurace zařízení.
- **Přístupnost:** Aplikace bude k dispozici zdarma a volně ke stažení jak mentálně postiženým klientům a neziskovým organizacím, tak i jakémukoli zájemci o zkvalitňování výuky.

Kapitola 5

Návrh

Předmětem kapitoly je návrh implementace systému – i-CT frameworku (dále jen frameworku) a aplikace pro AAK komunikaci typu ANO/NE (dále jen Aplikace ANO/NE) dle požadavků na návrh plynoucích z návrhových principů počítačové terapie, popsanych v kapitole 4.3.

5.1 Návrh i-CT frameworku

První součástí navrhovaného řešení je i-CT framework. Předmětem této podkapitoly je stanovit způsob fungování frameworku a také systému (frameworku a aplikací) jako celku.

Z principů Počítačové terapie má na návrh aplikace vliv konfigurovatelnost a rozšiřitelnost. Asistenti musí být schopni volně upravovat páry pro jednotlivé klienty. Jelikož je ale samotná karta znovupoužitelnou jednotkou, není vázána na jednotlivé klienty ale na asistenty.

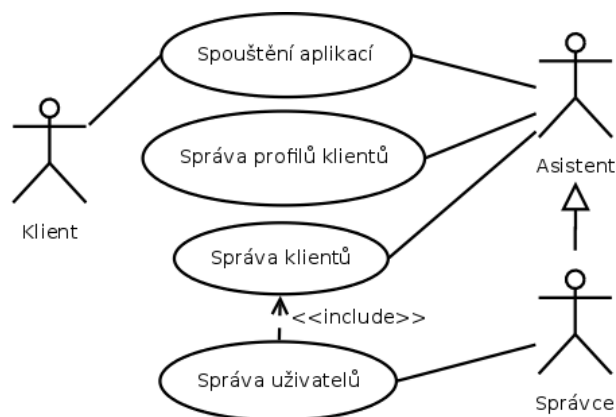
Návrhový princip rozšiřitelnosti požaduje podporu pro vývoj s ohledem na současné metodiky softwarového inženýrství. I při aplikaci všech pravidel vyplývajících z návrhových principů existuje více modelů navrhovaného systému splňujících požadavky. Pod vedením vedoucího práce a s ohledem na metodiku známou jako Domain driven design (viz [51]) byl sestaven níže popsáný model aplikace odrážející hlavní rysy návrhových principů.

Úkolem frameworku je zajistit přihlašování uživatelů (asistentů klientů, klientů a správců), umožnit asistentům evidovat informace o klientech a spouštět jednotlivé výukové aplikace. Výukové aplikace by přitom měly být schopny zjistit roli i identitu přihlášeného uživatele.

Z pohledu vývoje přináší framework společný základ aplikací zajišťující vedení databáze uživatelů a přihlašování. Vývoj aplikací tak je víceméně zbaven nutnosti práci s uživateli řešit. Z pohledu uživatele zajišťuje framework jedinou databázi, v rámci které je nutné uživatele spravovat. Jakoukoli změnu v uživatelských systémech tak stačí realizovat v rámci frameworku – není třeba ji řešit v rámci každé výukové aplikace zvlášť.

Za hlavní případ použití frameworku lze považovat spuštění výukových aplikací. Přitom se rozlišuje spuštění aplikace v režimu klienta a v režimu správy. Je-li výuková aplikace spuštěna pro klienta, měla by klientovi bránit v zásahu do konfigurace aplikací i zařízení. Je-li výuková aplikace spuštěna asistentem, měla by asistentovi umožnit připravovat v aplikaci materiály pro svoje klienty. Asistent klientů by měl mít také možnost spustit aplikaci pro některého svého klienta, aniž by se tento klient musel přihlašovat. Interakce frameworku a výukových aplikací je blíže popsána v kapitole 5.1.1.

Mezi další případy použití frameworku patří správa profilů klientů (zdravotních, sociál-



Obrázek 5.1: Diagram případů užití i-CT frameworku

ních a vzdělávacích) a správa uživatelů frameworku (klientů, asistentů a správců). Asistenti přitom mohou pracovat jen se svými klienty, svými podřízenými asistenty a s klienty svých podřízených asistentů. Správci mohou pracovat se všemi uživateli systému.

5.1.1 Rozdělení systému na i-CT framework a aplikace

V rámci návrhu i-CT frameworku je nezbytné stanovit, jakým způsobem bude systém rozdělen na i-CT framework a samotné aplikace. Jako nejjednodušší řešení se jeví vytvářet celý systém jako jedinou mobilní aplikaci, která by byla jen interně rozdělena na logické celky odpovídající jednotlivým výukovým aplikacím a i-CT frameworku.

V takovém případě ale není možné instalovat jednotlivé aplikace – je možné instalovat jen celý systém najednou. Na většině mobilních platform je odepřena možnost sebemodifikace – aplikace nemohou zapisovat do adresářů se svými spustitelnými soubory. Díky webové podstatě multiplatformních webových aplikací by sice bylo možné soubory aplikací ukládat do datových adresářů, součástí aplikace by pak ale nemohl být žádný nativní kód. Aplikace by pak mohly využívat jen služeb Cordova pluginů, které by byly součástí i-CT frameworku. Systém byl tedy navržen jako skupina zcela nezávislých aplikací.

Stále však je nutné stanovit způsob, jakým bude moci jedna aplikace spustit druhou (zejména i-CT framework výukovou aplikaci) a jakým budou mezi sebou předávat data.

5.1.2 Komunikace i-CT frameworku a výukových aplikací

V rámci platformy Android je možné pro komunikaci mezi aplikacemi využít Intent požadavek, popsáný v kapitole 3.1.3. Ten je ale dostupný jen na platformě Android. Na platformě iOS je vzájemné volání aplikací možné jen prostřednictvím odkazu na URL s URL schématem registrovaným cílovou aplikací.

URL schéma je předpona URL adresy určující protokol a tedy i program, pomocí kterého bude cílová URL otevřena. Aplikace instalované v rámci operačních systémů si mohou URL schémata registrovat podobně jako typy souborů, které mají být danou aplikací otvírány. Při jakémkoli pokusu o přechod na URL s daným schématem pak je spuštěna aplikace, která si dané URL schéma zaregistrovala.

Za URL schématem může navíc v adrese následovat libovolný text, což lze využít právě k předávání dat mezi aplikacemi. Vzájemné volání aplikací prostřednictvím URL schématu

je navíc možné také na platformě Android. URL schéma se tak jeví jako nejlepší (a vzhledem k požadavku na multiplatformitu také jedině) možné řešení.

V rámci fáze návrhu tedy bylo rozhodnuto vytvářet i-CT framework a výukové aplikace jako zcela nezávislé aplikace, z nichž každá bude mít zaregistrované své URL schéma. i-CT framework bude spouštět aplikace přesměrováním uživatele na URL se schématem cílové aplikace a v parametrech předá informace z databáze frameworku nutné k fungování výukových aplikací. Aplikace při svém ukončení zase uživatele přesměruje zpět do aplikace i-CT frameworku, přičemž v parametrech může zase předat informace i-CT frameworku.

Jedním z cílů i-CT frameworku je odstínit výukové aplikace od potřeby zajišťovat autentizaci uživatelů a také zabránit potřebě opakovaného přihlašování uživatele ke každé aplikaci zvlášť. Uživatelé tak nebudou nuceni se znovu přihlašovat při každém přechodu mezi aplikacemi, čímž by jinak byli neúměrně zdržováni. Je tedy nezbytné, aby i-CT framework předával výukovým aplikacím informace o přihlášeném uživateli – asistentovi klienta, popř. asistentovi samotném.

Zároveň je nutné mít na paměti, že délka URL skrze kterou jsou informace předávány může být na různých platformách různě omezena. Není tedy vhodné přenášet při spuštění aplikace asistentem informace o všech klientech přihlášeného asistenta. Z tohoto důvodu bylo rozhodnuto ustoupit od původního záměru provádět výběr klienta se kterým chce asistent pracovat v rámci výukové aplikace. Namísto toho bylo zvoleno řešení, kdy je klient asistentem zvolen ještě před spuštěním výukové aplikace. Aplikace pak při svém spuštění obdrží nejen informace o přihlášeném asistentovi, ale také informace o vybraném klientovi. Aplikace pak umožní asistentovi konfigurovat aplikaci pro vybraného klienta.

Stejným způsobem je možné realizovat také spuštění aplikace asistentem pro vybraného klienta. Asistent tak při spuštění aplikace volí nejen klienta, ale také režim, v jakém bude aplikace spuštěna:

- **Režim správy** asistent zvolí v případě, kdy chce upravit nastavení aplikace pro vybraného klienta.
- **Režim klienta** asistent zvolí v případě, kdy chce aplikaci spustit pro klienta. Zařízení pak může bezpečně předat klientovi, aniž by hrozilo, že klient úmyslně nebo neúmyslně zasáhne do konfigurace aplikace nebo zařízení.

Druhý uvedený režim vyžaduje stanovení mechanismu, jakým bude možné výukovou aplikaci spuštěnou v režimu klienta ukončit. Opuštění aplikace by mělo být nemožné pro klienta, ale snadné pro jeho asistenta. Jako vhodný se pro tento účel jeví použití odemykacích gest, podobných těm využívaných na zamykací obrazovce mobilních zařízení.

Přechod do výukové aplikace

Pro přechod z i-CT frameworku do výukové aplikace byly definovány následující parametry URL:

- **return** – URL schéma pro návrat z výukové aplikace do i-CT frameworku
- **userId** – ID zvoleného uživatele (klienta pro kterého byla výuková aplikace spuštěna)
- **userLogin** – přihlašovací jméno zvoleného uživatele
- **userName** – křestní jméno zvoleného uživatele

- `userSurname` – příjmení zvoleného uživatele
- `userLevel` – úroveň zvoleného uživatele (`CLIENT` – klient / `ASSISTANT` – asistent / `ADMIN` – správce)
- `userManagerId` – ID nadřízeného zvoleného uživatele (asistenta klienta)
- `userLanguage` – jazyk zvoleného uživatele
- `loggedId` – ID přihlášeného uživatele (asistenta který aplikaci spustil)
- `loggedLogin` – přihlašovací jméno přihlášeného uživatele
- `loggedGesture` – přihlašovací gesto přihlášeného uživatele (je-li požadováno pro opuštění aplikace)
- `loggedName` – křestní jméno přihlášeného uživatele
- `loggedSurname` – příjmení přihlášeného uživatele
- `loggedLevel` – úroveň přihlášeného uživatele (viz `userLevel`)
- `loggedManagerId` – ID nadřízeného přihlášeného uživatele (nadřízeného asistenta)
- `loggedLanguage` – jazyk přihlášeného uživatele
- `canEscapeApp` – zda-li může uživatel aplikaci opustit bez gesta asistenta (přestože aplikace běží v rámci kiosk a v režimu klienta)
- `allowedTime` – čas, po který je uživateli dovoleno zůstat v aplikaci (celé číslo v minutách, -1 pokud není omezeno)
- `manage` – `true` pokud je aplikace spuštěna v režimu správy (jinak není definováno)

```
myapp://?return=ictframework&userId=3&userLogin=klient&userName=Jana&
userSurname=Vešela&userLevel=CLIENT&userManagerId=2&userLanguage=cs&
loggedId=2&loggedLogin=asistent&loggedGesture=123&loggedName=Pavel&
loggedSurname=Smutny&loggedLevel=ASSISTANT&loggedManagerId=1&
loggedLanguage=en&canEscapeApp=false&allowedTime=15
```

Ukázka kódu 5.1: URL pro spuštění výukové aplikace

Celkově tak URL, kterou i-CT framework spouští výukovou aplikaci může vypadat například jako v ukázce kódu 5.1. Výukovou aplikaci zde spouští asistent Pavel Smutný (ID uživatele 2, lokalizace česká) pro klientku Janu Veselou (ID uživatele 3, lokalizace anglická). Pavel je přímým asistentem Jany. (Jana není klientem jeho nadřízeného.) Nadřízeným asistentem Pavla je uživatel s ID 1.

Pro opuštění aplikace je nutné zadat gesto asistenta. Klientovi je dovoleno aplikaci používat nejdéle 15 minut, pak je přeměrován na přihlašovací obrazovku frameworku. Poslední uvedená omezení jsou ale ignorována, pokud aplikace neběží v rámci kiosk na Android a nelze je tak vynutit.

Návrat do i-CT frameworku

Parametry pro opačný směr (návrat z výukové aplikace do i-CT frameworku) souvisejí s ověřením uživatele. Spustil-li asistent aplikaci pro klienta, mělo by být pro návrat z aplikace vyžadováno zadání gesta asistenta. Pokud výuková aplikace takové restriktce nepodporuje, měl by alespoň framework po opuštění aplikace provést odhlášení a zamezit tak přístupu klienta k profilu asistenta.

Je-li systém používán na platformě iOS, kde aplikace nemá možnost zabránit uživateli ve svém opuštění, je předpokládáno ruční spouštění Asistovaného režimu systému iOS při předání zařízení klientovi. Výše uvedená pravidla pak nemají kýžený efekt a na návrat z aplikace pak lze pohlížet stejně jako na návrat z režimu správy.

Pro návrat z výukové aplikace do i-CT frameworku je proto možné použít následující URL:

- `ictframework://?afterGesture=true` – jde-li o návrat po zadání gesta asistenta (není nutné uživatele odhlásit – již prokázal že je přihlášeným asistentem)
- `ictframework://?fromManage=true` – jde-li o návrat ze správy aplikace (opět lze ihned pokračovat pod účtem asistenta)
- `ictframework://?becauseNotInKiosk=true` – byl-li požadavek na zadání gesta přeskóčen, protože jde o platformu iOS (blokování opuštění aplikace není k dispozici a spoléhá se na Asistovaný režim systému iOS – viz kapitola 6.1)

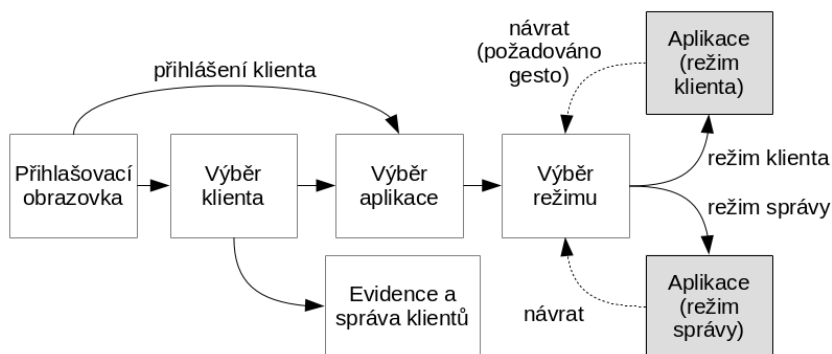
Registace výukové aplikace

URL schéma se využívá také pro registraci výukové aplikace v rámci i-CT frameworku. Aplikace se voláním frameworku přidá do nabídky i-CT frameworku. Specifikuje přitom URL schéma, za pomoci kterého ji může framework spustit, a svůj název.

```
ictframework://?action=register&scheme=myapp&name=MojeVyukovaAplikace
```

Ukázka kódu 5.2: Příklad URL pro registraci aplikace

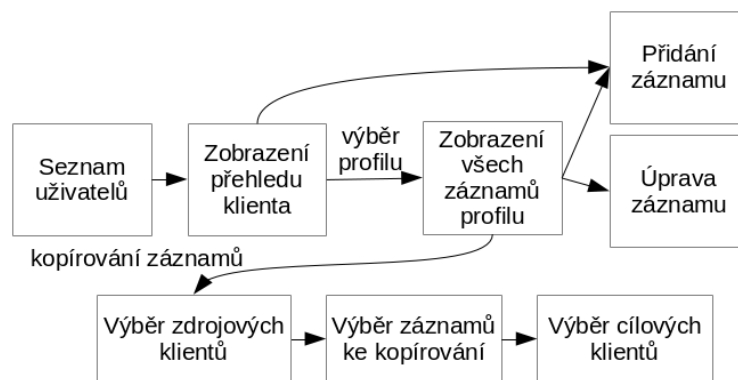
5.1.3 Navržená posloupnost obrazovek i-CT frameworku



Obrázek 5.2: Sled obrazovek pro spouštění výukových aplikací

K použití výukového systému je vyžadováno přihlášení uživatele. Jde-li o asistenta, po přihlášení je zobrazen výběr uživatele k úpravě. Zvolením klienta je zobrazen přehled

klienta, z kterého je možné pokračovat k výběru výukové aplikace. Tu může následně spustit v režimu správce (pak může v rámci aplikace upravovat materiály vztahující se k vybranému klientovi), nebo v režimu klienta. Po spuštění výukové aplikace předá zařízení klientovi a po dokončení výuky jej od něj zase převezme. Pro opuštění aplikace se v takové situaci musí prokázat svým odemykacím gestem. Tím je zabráněno tomu, aby mentálně postižený klient opustil výukovou aplikaci a neoprávněně manipuloval s konfigurací zařízení nebo daty uživatelů. Popsaný sled obrazovek znázorňuje obrázek 5.2.



Obrázek 5.3: Obrazovky frameworku pro správu profilů

Z přehledu klienta je možné také spravovat zdravotní, sociální a vzdělávací profil klienta. V samotném přehledu klienta je zobrazeno jen několik posledních záznamů vybraného profilu. Kliknutím na řádky posledních záznamů je možné zobrazit úplný seznam záznamů daného profilu. Odtud je pak možné přejít k úpravě jednotlivých záznamů včetně vytváření nových a jejich kopírování.

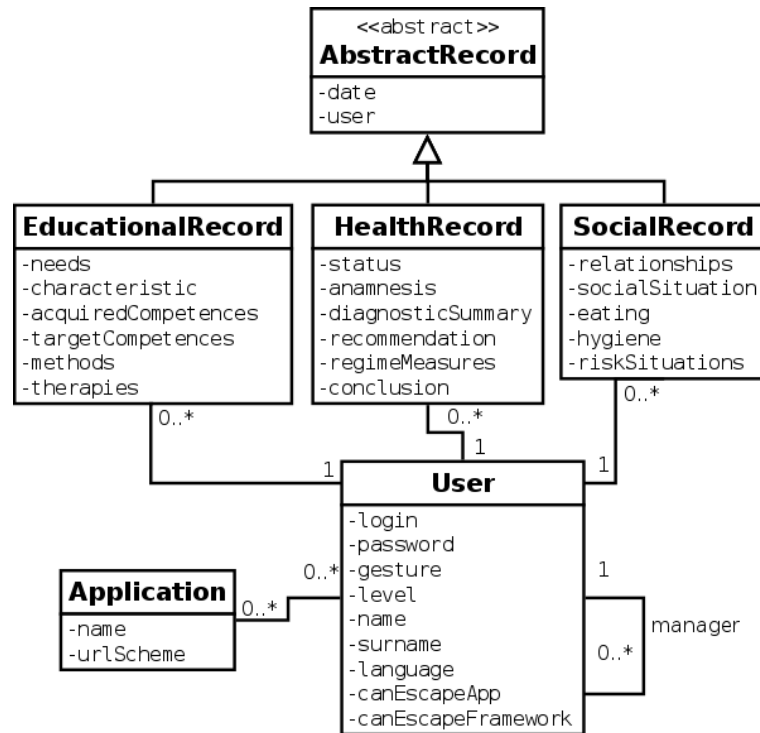
Pro kopírování záznamů je nejprve uživatel vyzván k výběru klientů, z jejichž profilu budou záznamy kopírovány. Na další obrazovce jsou pak zobrazeny záznamy vybraných uživatelů daného typu, z nichž uživatel vybírá záznamy, které mají být kopírovány. Na poslední obrazovce kopírování pak uživatel vybírá klienty, do jejichž profilů budou záznamy zkopírovány. Je tak možné kopírovat více záznamů, z více profilů klientů i do více profilů klientů současně. Odborní asistenti by tak měli být zbaveni maxima práce s vyplňováním profilů uživatelů.

5.1.4 Návrh struktury databáze i-CT frameworku

Účelem databázové části i-CT frameworku je uchovávat informace o uživatelích frameworku, zejména o klientech – profily klientů a v neposlední řadě také informace o aplikacích asociovaných s i-CT frameworkem.

Každý klient má přitom zdravotní, sociální a vzdělávací profil. Každý z těchto dílčích profilů je tvořen seznamem záznamů profilu, kde jednotlivé záznamy reprezentují stav klienta v různých časových okamžicích. Protože jsou záznamy jednotlivých profilů značně odlišné ve svých položkách, byly navrženy jako samostatné třídy entit. Protože ale mají některé společné vlastnosti (časový okamžik ke kterému se záznam vztahuje a vztah s klientem), byly založeny na společné abstraktní nadtrídě **AbstractRecord**, která reflektuje tyto společné vlastnosti.

Z návrhových principů počítačové terapie (viz kapitola 2.4) vyplývá nezbytnost restrikcí – omezení pohybu uživatelů v systému. Většina z uplatňovaných restrikcí nemá vliv



Obrázek 5.4: Diagram tříd modelu domény

na strukturu databáze s jednou výjimkou – omezení aplikací, které je možné spouštět pro jednotlivé klienty. Vztah mezi aplikacemi a uživateli je N:M – každému uživateli může být přiřazeno více aplikací, které je pro něj možné spustit, a každá aplikace může být přiřazena více uživatelům.

5.2 Návrh aplikace ANO/NE

Druhou součástí implementace této práce je aplikace pro alternativní a augmentativní komunikaci (AAK) typu ANO/NE (dále jen Aplikace ANO/NE). Cílem této kapitoly je navrhnout způsob fungování aplikace.

Cílem Aplikace ANO/NE je usnadnit komunikaci mentálně postiženého člověka (klienta) s jeho asistentem. Asistent by přitom měl vybrat pár možností a předat elektronickou pomůcku klientovi, kterému aplikace umožní co nejnázornější výběr z těchto dvou možností tak, aby se asistent dozvěděl, která možnost byla klientem zvolena. Tento hlavní případ použití bude dále označován jako **přehrávání**.

Další případy použití souvisí se samotnou přípravou párů pro klienty. Páry jsou tvořeny z karet možností. Karta možnosti zahrnuje vše související s jednou stranou páru. Může být přitom používána současně ve více párech, čímž je znovupoužitelná. To je nedocenitelné, protože jejich příprava je pro asistenty pracná. Jednotlivé páry jsou dále spojovány do seznamů souvisejících párů.

Případy užití aplikace shrnuje diagram případů užití na obrázku 5.5.

Každá karta možnosti by měla zahrnovat:

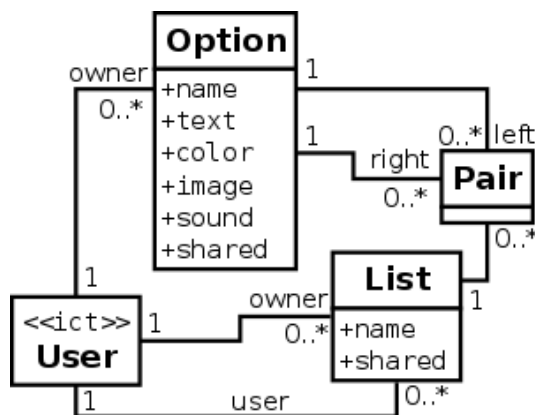
- Text zobrazovaný na kartě



Obrázek 5.5: Diagram případů užití Aplikace ANO/NE

- Barvu pozadí karty
- Obrázek nebo animaci zobrazovanou na kartě
- Zvukový záznam přehrávaný při zvolení karty

Karty možností musí být následně možné skládat do párů. Každý pár obsahuje referenci na dvě karty možností – na pravou a levou polovinu páru. Ty asistenti vybírají ze svého katalogu karet. Páry jsou dále sdružovány do seznamů párů, které shlukují související páry a jsou přiřazeny jednotlivým klientům. Navržená struktura databáze je nastíněna přehledovým diagramem tříd na obrázku 5.6.



Obrázek 5.6: Přehledový diagram tříd Aplikace ANO/NE. (Stereotyp <<ict>> označuje entitní množinu existující v rámci i-CT frameworku.)

Jak pro karty možností tak pro seznamy párů se ukládá uživatel (asistent), který entitu vytvořil. Pro seznamy párů se ukládá navíc i uživatel (klient), pro kterého byl seznam vytvořen. Entitní množina uživatelů ale není součástí databáze aplikace ANO/NE – zde se ukládají jen reference na entity uživatelů existující v rámci aplikace i-CT frameworku.

5.2.1 Online katalog

Online katalog umožňuje sdílení obsahu v rámci aplikace ANO/NE napříč více zařízeními. Jde o webovou aplikaci běžící na serveru v internetu nebo místní síti, která poskytuje

zařízením s výukovou aplikací karty možností a seznamy párů, které jsou na něm uloženy.

Uživatelé mohou z online katalogu karty a seznamy nejen stahovat, ale mohou je do něj také nahrávat. Pro přístup do online katalogu je vyžadováno přihlášení, na základě kterého jsou určena oprávnění uživatele.

Přihlašuje se přihlašovacím jménem a heslem uživatele serveru, který není pevně vázaný na účet uživatele i-CT frameworku. Uživatelé i-CT frameworku se připojují k online serverům pomocí přihlašovacích údajů, které sami zadají. V rámci aplikace si uživatelé i-CT frameworku sami tvoří seznam serverů, které chtějí mít možnost používat. V rámci něj jsou mimo URL adres serverů ukládány právě také přihlašovací údaje k serveru.

Uživatelé mají možnost karty a seznamy párů nahrávat a zase je odstraňovat. Odstraňovat ale mohou jen karty a seznamy párů, které sami nahráli na server.

Kapitola 6

Implementace

Kapitola popisuje implementaci i-CT frameworku (jehož návrh je popsán v kapitole 5.1) a výukové aplikace pro AAK typu ANO/NE (jejíž návrh je popsán v kapitole 5.2).

S ohledem na práci s modely získanými na základě definovaných návrhových principů lze celkově hovořit o vývoji s využitím techniky Model-Driven Development. (viz [38])

6.1 Implementace i-CT frameworku

Framework byl na základě požadavku vedoucího práce implementován za pomoci rámce Oracle MAF, popsaného v kapitole 3.4. Implementačním jazykem je tedy Java a obrazovky uživatelského prostředí jsou definovány prostřednictvím AMX UI.

Databáze a ORM

Aplikace frameworku vyžaduje ke svému fungování databázi – pro ukládání záznamů klientů, jejich asistentů a profilů. Víceméně jasnou volbou je databáze SQLite, která je jedinou relační databází dostupnou na obou požadovaných platformách (Android a iOS) a prostřednictvím rozhraní JDBC. JDBC je poskytováno rámcem Oracle MAF. Umožňuje provádět SQL příkazy z kódu v Javě, ale samo o sobě neposkytuje objektově-relační mapování (ORM). [12]

Pro zjednodušení práce s databází byla proto zároveň použita knihovna OrmLite. Ta aplikacím v Javě poskytuje objektově-relační mapování a zprostředkovává tak komunikaci s databází dostupnou skrze JDBC. Není tak pro uložení objektu do databáze nutné ručně psát aktualizací SQL dotazy a při čtení z databáze zase ručně plnit atributy objektů daty z databáze.

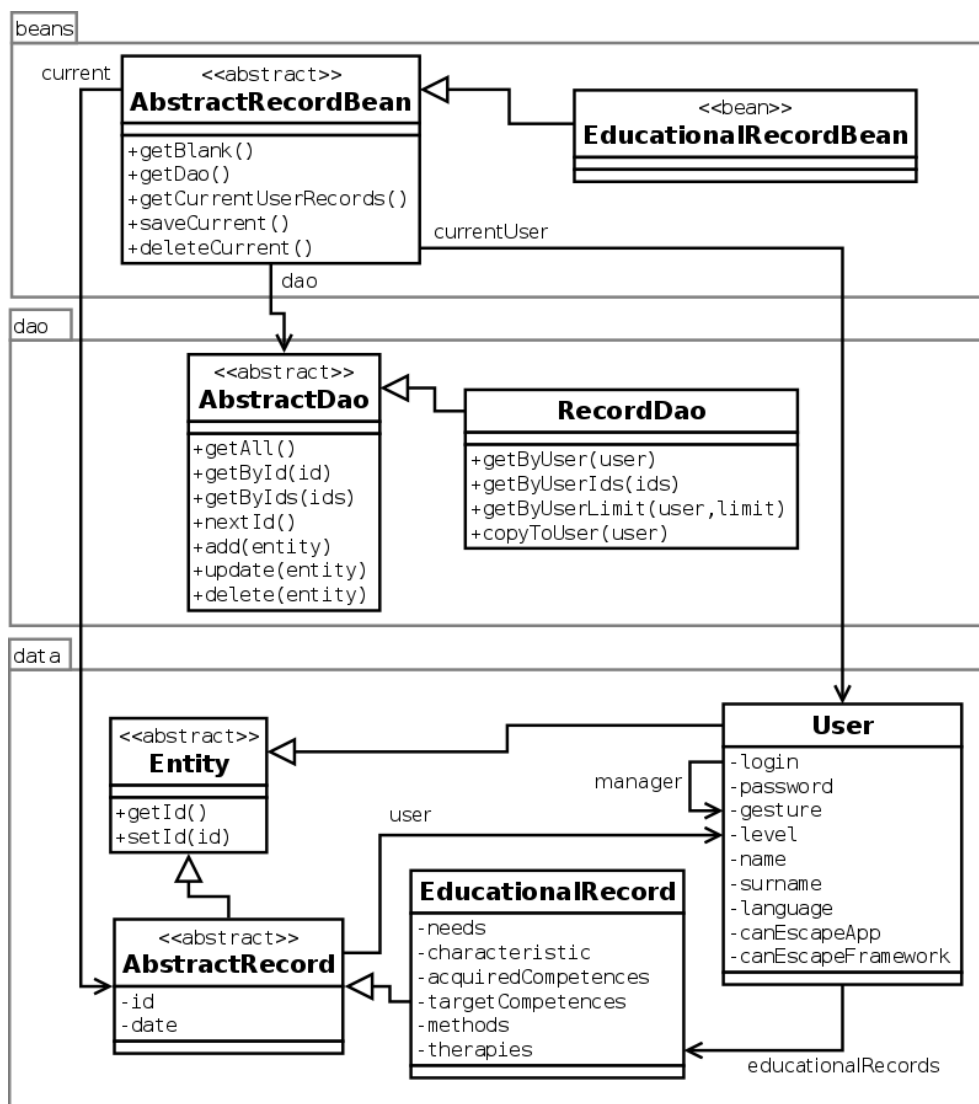
Knihovna OrmLite byla vybrána jako nejmenší postačující knihovna tohoto typu. Zatímco obecně nejpoužívanější ORM knihovna Hibernate zabírá sama o sobě více než 100 MB, knihovna OrmLite si bez problému vystačí s méně než 400 kB. To je samozřejmě vyváženo mnoha omezeními – např. není možné plnohodnotně využívat dědičnost mezi objekty domény. Není například možné dotazovat se na objekty napříč více podtřídami jedné nadtřídy. Není tak například možné procházet chronologicky všechny záznamy o klientovi napříč zdravotními, sociálními a vzdělávacími profily.

Problémem při použití knihovny ORMLite pro přístup k SQLite databázi skrze JDBC je bohužel také chyba, díky které není možné použít standardní generované primární klíče – v této kombinaci neproběhne korektně jejich generování. V rámci implementace i-CT frameworku je to řešeno ručním generováním primárního klíče (ID) za pomoci dotazu na

maximální ID nacházející se v dané tabulce. To je jen navýšeno o 1 a manuálně nastaveno jako ID nového záznamu.

Vzhledem k tomu, že převod modelových tříd na tabulky relační databáze je zajišťován knihovnou ORMLite, struktura databáze je definována za pomoci diagramu tříd na obrázku 5.4.

Implementační diagram tříd na obrázku 6.1 znázorňuje architekturu aplikace. Ukazuje rozdělení tříd do vrstev a vazby mezi zobrazenými třídami. Zobrazuje všechny abstraktní třídy frameworku a třídy související se záznamy vzdělávacího profilu. Třídy pro práci s ostatními typy databázových objektů mají vazby obdobné.



Obrázek 6.1: Implementační diagram tříd části i-CT frameworku – pro záznamy vzdělávacího profilu klienta (pro zjednodušení nejsou zakresleny settery a gettery)

Jak již bylo stanoveno v rámci návrhu, třídy záznamů klienta mají společného abstraktního předka `AbstractRecord`. Pro zjednodušení implementace tříd pro manipulaci s databází byl vytvořen také společný předek všech entitních tříd – `Entity`. Tento typ se následně využívá v implementaci objektů pro přístup datům (DAO – Data access object).

Jak již bylo nastíněno, z jistého pohledu by mohlo být výhodné mít jediný seznam záznamů o klientovi a ten mít možnost procházet chronologicky napříč typy záznamů. Právě to by už ale bylo nad schopnosti odlehčeného ORM OrmLite.

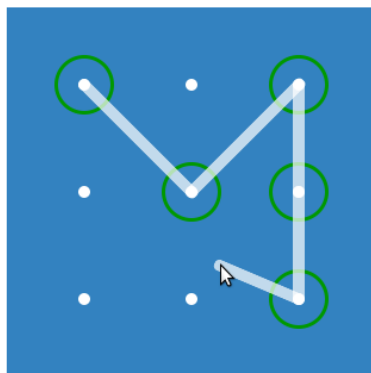
Omezení klienta a odemykací gesto

Jak již bylo naznačeno v kapitole 3.6, protože bude zařízení sloužit lidem s mentálním postižením, je nezbytné zamezit volnému procházení obsahu zařízení jeho držiteli.

Všechny možnosti zabezpečení zařízení proti nežádoucímu zásahu byly již zdokumentovány v kapitole 3.6. Cílem této podkapitoly je upřesnit, jaká opatření a jak byla implementována v rámci i-CT frameworku a jaká by měla být implementována v rámci každé výukové aplikace používané s i-CT frameworkem.

Jak vyplývá z poznatků z kapitoly 3.6, pro omezení pohybu klienta nad platformou Android je nezbytné ustanovit aplikaci frameworku spouštěcí obrazovkou systému, zabránit zobrazování systémových oken a odchytit události hardwarových tlačítek. Za tímto účelem byl již v rámci kapitoly 3.6 vytvořen plugin rámce Cordova, který umožňuje zabránit opuštění aplikace na platformě Android. Jeho použití v aplikaci vytvořené za pomoci Oracle MAF však nebylo tak bezproblémové jak se původně předpokládalo.

Plugin bylo nutné upravit, aby třídou kterou rozšiřuje nebyla `org.apache.cordova.CordovaActivity` ale `oracle.adfmf.Container` – aktivita která je součástí rámce MAF, která mimo vykreslení WebView zajišťuje také start JVM, na které běží části aplikace v Javě. Vzhledem k nedostupnosti zdrojových kódů Oracle MAF (jde o proprietární produkt) bylo nutné jméno třídy rozšiřované aktivity získat dekompilací APK balíčku generovaného Oracle MAF.



Obrázek 6.2: Komponenta pro zadání gesta

Ověření asistenta za pomoci odemykacího gesta bylo realizováno za pomoci Javaskriptové knihovny `patternLock.js`. Tato knihovna je postavená na knihovně jQuery a do specifikovaného blokového elementu dokáže vykreslit komponentu pro zadání gesta podobnou zamykací obrazovce systému Android. Výstupem komponenty je textový řetězec – posloupnost číselic označujících jednotlivé body kreslicí plochy. Tento řetězec lze dále použít stejným způsobem jako heslo uživatele. [27]

Komponenta pro zadání gesta byla v rámci aplikace frameworku implementována ve formě znovupoužitelné AMX UI komponenty. Pro vložení další instance komponenty do aplikace frameworku postačuje AMX kód uvedený v ukázce kódu 6.1 a deklarace odpovídajícího XML jmenného prostoru v hlavičce AMX dokumentu, do kterého je komponenta vkládána.

```
<ict:gesture destination="UserBean.current.gesture"></ict:gesture>
```

Ukázka kódu 6.1: Příklad použití implementované komponenty pro zadání gesta

Jak je z ukázky vidět, implementované komponentě stačí předat referenci na atribut Java Bean, do kterého má být uloženo zadané gesto. Komponenta může být vložena do prakticky jakékoli AMX stránky. V aplikaci frameworku je komponenta použita na přihlašovací obrazovce a na obrazovce pro změnu přihlašovacího gesta uživatele.

Na platformě iOS je zabránění opuštění aplikace zajištěno za pomoci Asistovaného režimu systému iOS. V nastavení systému stačí nastavit použití Asistovaného režimu a po každém spuštění výukové aplikace pak trojím stiskem tlačítka menu Asistovaný režim spustit. Opuštění aplikace pak není možné, dokud není Asistovaný režim ukončen – trojím stiskem tlačítka menu a zadáním nastaveného kódu.

Asistovaný režim systému iOS nemůže být řízen aplikací a brání i opuštění aplikace iniciované aplikací samotnou. Asistent klienta tak po spuštění výukové aplikace musí ručně Asistovaný režim spustit a před opuštěním aplikace jej zase ukončit. Protože se přitom musí prokazovat číselným kódem systému iOS, je požadování gesta asistenta aplikací na platformě iOS neúčinné a je zde proto potlačeno.

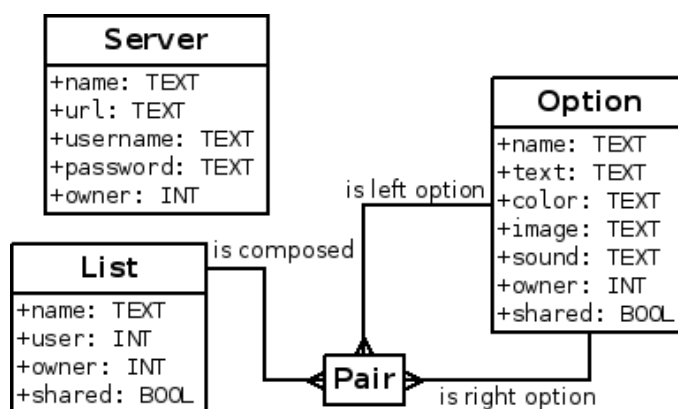
6.2 Implementace aplikace pro komunikaci typu ANO/NE

Pro implementaci výukové aplikace pro komunikaci typu ANO/NE byla zvolena platforma Ionic, která je blíže popsána v kapitole 3.5. Důvodem pro volbu platformy Ionic byla zejména větší otevřenost platformy (jde o open source software) a snazší práce s částmi aplikace psanými v nativním kódu. Ta by byla výhodou při implementaci volby možnosti z páru za pomoci speciálního hardware (fyzických spínačů) nebo počítačového vidění. Na implementaci těchto volitelných rozšíření ale nakonec nezbyl prostor a nejsou implementovány.

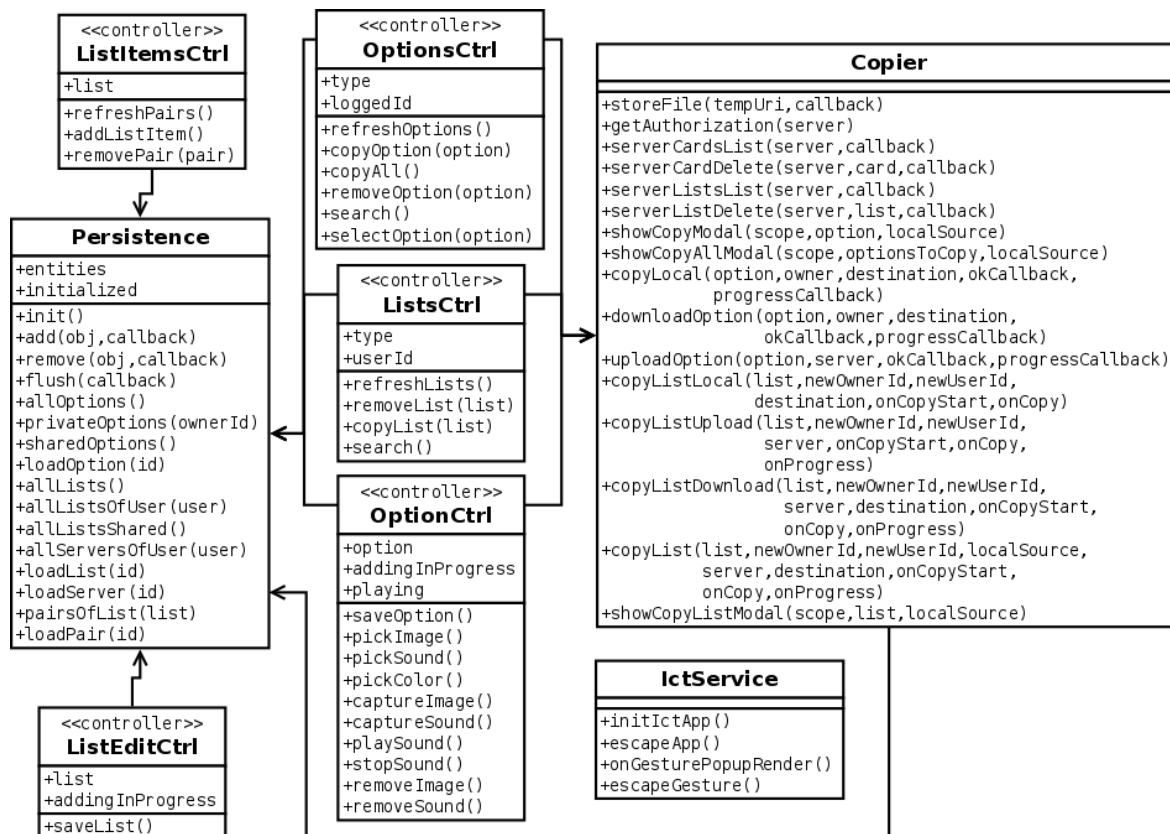
Aplikace nad platformou Ionic jsou tvořeny šablonami stránek (obrazovek, view) a programovou částí (modelem) v jazyce Javascript. Programová část je tvořena moduly, které jsou tvořeny řadiči (controller) a dalšími součástmi modelu.

Výuková aplikace byla rozdělena do následujících modulů:

- **App:** Hlavní modul aplikace. Definuje adresy stránek obrazovek aplikace a zajišťuje načtení ostatních modulů.



Obrázek 6.3: ER diagram databáze Aplikace ANO/NE



Obrázek 6.4: Implementační diagram tříd Aplikace ANO/NE (pro zjednodušení vynechány řadiče méně významných obrazovek aplikace)

- **Color:** Modul poskytující komponentu pro výběr barvy z nabídky. Využívá se pro výběr barvy pozadí karty.
- **ICT:** Modul zajišťující interakci výukové aplikace s frameworkem, včetně požadování gesta asistenta, je-li vyžadováno k opuštění aplikace.
- **Options:** Modul zajišťující správu karet možností, zejména jejich úpravu včetně přiřazování obrázků a zvukových záznamů.
- **Playing:** Modul zajišťující přehrávání seznamů párů, tedy samotnou interakci aplikace s mentálně postiženými klienty.
- **Database:** Modul zpřístupňující databázi aplikace. Definiuje strukturu databáze výukové aplikace.
- **Copier:** Modul zajišťující kopírování karet a seznamů párů, jak lokálně, tak v interakci s online katalogem.

Strukturu databáze výukové aplikace popisuje ER diagram na obrázku 6.3. Součástí všech tří tabulek je cizí klíč do tabulky uživatelů v i-CT frameworku (atributy **owner** a **user**), která není součástí databáze (a tedy ani ER diagramu) výukové aplikace.

Strukturu aplikace ukazuje implementační diagram tříd na obrázku 6.4. Každá obrazovka aplikace má vlastní řadič (controller). Definice databázové struktury a funkce pro

základní operace nad databází jsou součástí služby **Persistence**. Funkce související s kopírováním karet i seznamů párů poskytuje za tímto účelem vytvořená služba **Copier**. Funkce související s komunikací s i-CT frameworkem poskytuje služba **ictService**.

6.2.1 Online katalog

Online katalog je implementován jako webová aplikace na platformě Node.js. Jednotlivé operace nad uloženými kartami a seznamy párů umožňuje provádět prostřednictvím rozhraní REST. Jednotlivými operacemi pro práci s kartami možností v katalogu přitom jsou:

- GET `/cards/list` – získání seznamu karet
- GET `/cards/get/:id` – získání vlastností karty
- GET `/download/:id` – stažení souboru obrázku/zvuku podle identifikátoru z vlastností karty
- POST `/cards/create` – vytvoření karty (bez souboru obrázku a zvuku)
- POST `/cards/update` – úprava vlastností karty (bez změny souboru obrázku a zvuku)
- POST `/cards/attach` – přiložení/nahrazení souboru obrázku/zvuku ke kartě
- DELETE `/cards/delete/:id` – odstranění karty

Podobná sada operací je poskytována pro práci se seznamy párů:

- GET `/lists/list` – získání seznamu všech seznamů párů
- GET `/lists/get/:id` – získání párů seznamu
- POST `/lists/addpair` – přidání páru do seznamu
- POST `/lists/create` – vytvoření nového seznamu
- DELETE `/lists/delete/:id` – odstranění seznamu

Sada operací pro práci s katalogem není úplná – server např. neumožňuje bez náhrady odstranit soubor obrázku/zvuku karty, nebo odstranit pár ze seznamu. Protokol je primárně určen jen ke čtení, kopírování seznamů a karet ze serveru a na server a k jejich odstraňování ze serveru.

Pro práci s databází je využita ORM knihovna **persistence.js** [15], která objektovým způsobem zpřístupňuje MySQL databázi aplikaci. REST rozhraní je realizováno za pomoci knihovny **express** [13], jejíž možnosti jsou rozšířeny o podporu nahrávání souborů na server za pomoci knihovny **multer** [21], vyvíjené v rámci stejného projektu. Zabezpečení REST služeb proti neoprávněnému použití je zajištěno za pomoci knihovny **passport** [26], která jednotlivé REST služby doplňuje o BASIC HTTP autentizaci.

Online katalog byl nasazen na infrastrukturu služby OpenShift, kde byl dostupný po celou dobu testování. Ze zdrojových souborů je možné kdykoli vytvořit nový online katalog nasazením na nově vytvořený virtuální server (gear) platformy OpenShift nebo na jiný server poskytující prostor pro Node.js aplikace.

Kapitola 7

Zhodnocení aplikace a testování

Cílem kapitoly je zhodnocení úspěšnosti implementace a návrhu aplikace. Zatímco první část zhodnocuje úspěšnost porovnáním se stávajícími aplikacemi tohoto druhu, druhá popisuje testování aplikace v reálných podmínkách – v prostředí speciální školy, s cílem ověření praktické použitelnosti při využití metod měření použitelnosti z oblasti interakce člověka s počítačem (HCI – human–computer interaction). [42]

7.1 Porovnání se současnými řešeními

Dle dostupných zdrojů neexistuje v současnosti aplikace, kterou by bylo možné přímo porovnat s nástrojem i-CT frameworku – která by plnila požadovanou funkci pro naši cílovou skupinu. (Proto je tato aplikace sama o sobě podstatným přínosem, viz podkapitola 7.2.) Z tohoto důvodu se tato podkapitola dále zabývá jen aplikací pro AAK typu ANO/NE.

Cílem podkapitoly je porovnání nově vytvořené Aplikace ANO/NE se stávajícími aplikacemi pro AAK typu ANO/NE: S jednoduchou aplikací **Yes/No** od společnosti I Can Do Apps, s na jiný způsob AAK zaměřenou aplikací **Klábosil** a s aplikací vzniklou již v rámci projektu Počítačové terapie na FIT VUT, **YesNoCommunication** Jana Tůmy. Všechny uvedené aplikace byly již představeny v rámci kapitoly 4.2. Předmětem této kapitoly je tedy jen jejich porovnání s nově implementovanou aplikací.

Aplikace zabývající se AAK jsou zpravidla platformě závislé, ať již jde o aplikaci Klábosil, nebo aplikaci Yes/No od I Can Do Apps. Jak dříve vytvořená aplikace YesNoCommunication od Jana Tůmy, tak Aplikace ANO/NE vytvořená v rámci této práce, podporují obě hlavní mobilní platformy. Zatímco však v případě dříve vytvořené aplikace jde ve skutečnosti o dvě samostatné implementace, aplikace vytvořená v rámci této práce má jedinou multiplatformní implementaci.

S tímto souvisí také naplnění principů Počítačové terapie, zde zejména principu **multiplatformity** a **rozšiřitelnosti**. Zatímco v případě dřívější aplikace YesNoCommunication je nutné udržovat samostatné implementace pro obě podporované platformy, nová aplikace má většinu kódu společného pro obě platformy a pro obě platformy současně tak je nutné pouze testovat. Vývoj probíhá pro obě platformy současně. Platforma Ionic navíc do budoucna umožňuje přidat i další podporované platformy. Škála platform podporovaných frameworkem Oracle MAF je naproti tomu výrazně užší. Proprietární aplikace nebylo možné z hlediska rozšiřitelnosti objektivně porovnat.

Všechny porovnávané aplikace jsou **použitelné** i v prostředí mimo prostory školy, tedy v offline režimu, bez připojení k serveru v místní síti nebo k internetu. Zatímco však Yes/No

a YesNoCommunication za to vděčí tomu, že ani jinou než lokální databázi nepoužívají, jen v případě nově implementované aplikace je dostupný i síťový server, kam lze připravené výukové materiály ukládat, avšak činnost samotné aplikace na dostupnosti spojení se serverem nezávisí.

Kapitolou samou pro sebe je **bezpečnost**. Tu je možné rozdělit na ochranu dat uložených v aplikaci (zejména z hlediska ochrany osobních údajů) a na ochranu před opuštěním aplikace a neoprávněnou manipulací s konfigurací zařízení nesouvisející s aplikací.

Zatímco aplikace určené výhradně pro iOS se mohou v ochraně před opuštěním aplikace spoléhat na Asistovaný režim systému iOS, řešení podporující platformu Android si tento luxus dovolit nemůže. Dřívější aplikace YesNoCommunication opuštění aplikace nijak nebrání. Nově implementovaná aplikace oproti tomu řešení poskytuje, byť k tomu vyžaduje součinnost i-CT frameworku. (blíže viz kapitola 6.1)

Porovnání naplnění jednotlivých principů Počítačové terapie a tedy i potřeb koncových uživatelů jednotlivými aplikacemi ukazuje tabulka 7.1.

	Yes/No (I Can Do Apps)	YesNoCommunication (Počítačová terapie)	Aplikace ANO/NE (produkt této práce)
Multiplatformita	✗jen iOS	✓Android a iOS ¹	✓Android a iOS
Otevřenost	✗propietární	✓otevřené	✓otevřené
Rozšiřitelnost	nelze posoudit	✗dvě samost. impl.	✓jediná impl.
Stanovený cíl	✓stanoven	✓stanoven	✓stanoven
Použitelnost	✓offline	✓offline	✓offline (+ online)
Konfigurovatelnost	✗nelze tvořit páry	✓lze tvořit	✓lze tvořit
Bezp. – dat	✓není co chránit	✓přihlášení	✓přihlášení
Bezp. – opuštění	✓Asist. režim iOS	✗na Android ne	✓opouštěcí gesto
Přístupnost	✓zdarma	✓zdarma	✓zdarma

Tabulka 7.1: Porovnání naplnění principů Počítačové terapie aplikacemi pro AAK typu ANO/NE

7.2 Testování s reálnými uživateli

Cílem testování s reálnými uživateli je ověřit **použitelnost** systému (i-CT frameworku a aplikace pro AAK typu ANO/NE) v praxi. Tato forma testování byla vytvořena na základě návrhu vedoucího této práce, při využití metod zmíněné oblasti HCI, odpovídající tomuto druhu testování.

Testování probíhalo v prostorách Speciální základní školy Poděbrady. V této speciální škole není použití tabletů novinkou – zdejší pedagogičtí pracovníci využívají iPady pro potřeby výuky již déle než 5 let. V květnu 2011 zde dokonce založili iniciativu **iSEN**, jejíž cílem je právě podpora využití iOS zařízení ke vzdělávání a rozvoji komunikace dětí a k AAK. [48]

Komunita iSEN není ve světě neznámou – v roce 2013 byla pozvána k uvedení mezinárodní události **Apple Special Needs Summit, Prague 2013**, konferenci pro výměnu zkušeností s účastníky ze 17 zemí včetně zástupců 12 ministerstev školství a společnosti Apple. [47] Úspěchy speciální školy v Poděbradech se prezentuje i společnost Apple ve svých

¹Každá platforma má ale vlastní nezávislou implementaci

videích demonstrujících využití iPadů ve speciální pedagogice, kde je tato škola uvedena bok po boku se speciální školou v Japonsku a speciální školou v New Yorku. [10]

Škola vzdělává žáky různého věku a různého postižení. V rámci speciální školy se osvědčilo sdružovat žáky těžšího postižení než lehkého mentálního postižení do tříd víceméně bez ohledu na věk – v rámci jedné třídy se společně vzdělávají děti s až 5letým rozdílem věku. Žáci jsou řazeni do tříd spíše podle druhu a míry postižení. Ani druh postižení však není zcela směrodatný – dle zkušeností zdejších pedagogů se jeví i sdružování žáků různých typů podobně orientovaných postižení jako bezproblémové.

Testování použitelnosti systému bylo prováděno zdejšími pedagogickými pracovníky v období 11.5. – 20.5.2016 se 6 pedagogy a 14 žáky. Pět žáků trpí lehkým mentálním postižením a účastní se rámcového vzdělávacího programu (RVP) základního vzdělávání a 9 se účastní RVP speciálního vzdělávání.

7.2.1 Způsob měření použitelnosti

Metrika použitelnosti (usability) stanovuje, jak snadné je pro uživatele uživatelské rozhraní používat. Dle Jakoba Nielsena je použitelnost definována jako souhrn následujících měřítek: [32]

- **Naučitelnost** (learnability) – jak snadné je pro uživatele provádět základní úlohy při prvním setkání s rozhraním
- **Efektivita** (efficiency) – jak rychle je uživatel schopný provádět základní úlohy po obeznámení se s rozhraním
- **Zapamatovatelnost** (memorability) – jak snadné je pro uživatele provádět základní úlohy po návratu po delší době nepoužívání rozhraní
- **Chybovost** (errors) – kolik chyb uživatelé při používání rozhraní dělají a jak snadno se z nich zotaví
- **Uspokojení** (satisfaction) – jak příjemné je pro uživatele používání rozhraní

Pro potřeby stanovení použitelnosti systému byli testující pedagogičtí pracovníci požádáni o vyplnění dotazníků, kde hodnotili jednotlivé aspekty použitelnosti obou aplikací na stupnici 1 – 5, kde 5 je nejlepší. (Respektive 1 – 4, kde 4 je nejlepší, pro Zapamatovatelnost.) Tato stupnice byla převedena do rozsahu 0 – 1 (respektive 0% – 100%), kde 1 (respektive 100%) je nejlepší, podle vztahu:

$$x_{0..1} = \frac{x_{min..max} - x_{min}}{x_{max} - x_{min}},$$

kde x_{min} označuje spodní mez (nejhorší hodnocení), x_{max} horní mez (nejlepší hodnocení) a $x_{min..max}$ samotnou vyplněnou hodnotu. Samotná použitelnost pak byla určena váženým průměrem:

$$u = \frac{\sum_{i=0}^n w_i x_i}{\sum_{i=0}^n w_i},$$

kde x_i představuje jednotlivá měřítka použitelnosti a w_i jim odpovídající váhy. Předpokládáme-li že součet vah je roven 1, lze výpočet zjednodušit:

$$u = \sum_{i=0}^n w_i x_i$$

Stanovené váhy jednotlivých měřítek jsou zaznamenány v tabulce 7.2.

Naučitelnost (learnability)	0,15
Efektivita (efficiency)	0,3
Zapamatovatelnost (memorability)	0,15
Chybovost (errors)	0,3
Uspokojení (satisfaction)	0,1

Tabulka 7.2: Stanovené váhy měřítek použitelnosti

Nejmenší váha je dána míře uspokojení, protože ta je nejvíce zatížená chybou způsobenou aktuální náladou uživatele. Největší váha je naopak přiřazena míře efektivity, kterou lze považovat za nejvíce vypovídající. Měří totiž rychlost práce již zkušeného uživatele při běžném používání. Není navíc ani tolik ovlivněna zkušenostmi předcházejícími testování. Zrovna tak důležitá je také míra chybovosti. Začínající uživatel může v rozhraní bloudit, nikdy by pro něj ale nemělo být obtížné se vracet. Na pomezí pak leží naučitelnost a zapamatovatelnost, které měří, jak rychle se v rozhraní zorientuje nový uživatel a uživatel, který se k rozhraní vrací po delší době. (S ohledem na časový rozsah testování byl za „delší dobu“ uvažován následující pracovní den.)

7.2.2 Výsledky testování i-CT frameworku

Obě části systému, i-CT framework a Aplikace ANO/NE, byly uživateli hodnoceny samostatně. V případě i-CT frameworku byla aplikace pro hodnocení rozdělena do tří oblastí:

- **Správa asistentů** – vlastního účtu a účtů jiných pedagogů (např. pokud má pedagog pod sebou několik jiných pedagogů)
- **Správa klientů** – účtu a profilu žáka/klienta, včetně jeho záznamů dělených na vzdělávací, sociální a zdravotní
- **Správa restrikcí** – nastavení zpřístupnění jen některých aplikací pro žáky/klienty a omezení doby spuštění výukové aplikace

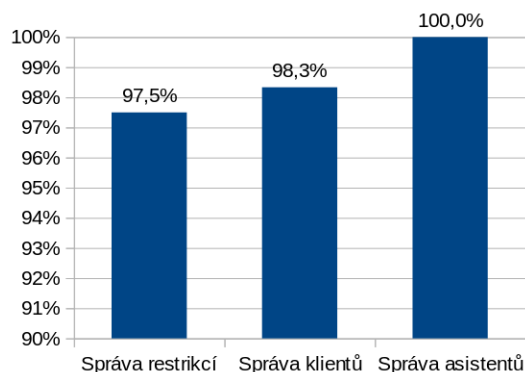
Nutno podotknout, že testování probíhalo na platformě iOS (z důvodu v současnosti největšího využití této platformy v této oblasti), kde je množina podporovaných restrikcí klienta silně zúžená (s ohledem na část restrikcí dostupných přímo v rovině systému iOS, jak je uvedeno v kapitole 6.1).

Celkové hodnocení použitelnosti ukazuje graf na obrázku 7.1. Průměrné hodnocení jednotlivých složek ukazuje graf na obrázku 7.2.

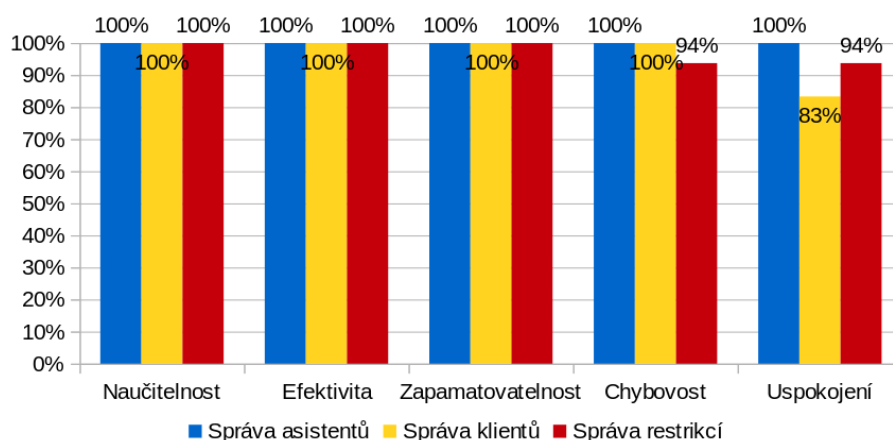
Z hodnocení použitelnosti vyplývá, že množství času věnované implementaci i-CT frameworku se vyplatilo – aplikace i-CT frameworku je z hlediska použitelnosti hodnocena velice dobře, což lze přisoudit i následování návrhových principů Počítačové terapie.

V rámci otevřených otázek uživatelé sdělili, že spatřují přínos v přihlašování uživatelů a v souvisejícím systému oprávnění uživatelů – v přiřazování klientů jednotlivým pedagogům.

Většina (5 ze 6) pedagogů ale v tuto chvíli nenachází využití pro vedení agendy (vzdělávacího, sociálního a zdravotního profilu) ve spojení s jednou, zde ukázanou, aplikací pro



Obrázek 7.1: Celková použitelnost aplikace i-CT frameworku



Obrázek 7.2: Jednotlivé složky použitelnosti aplikace i-CT frameworku

AAK typu ANO/NE. Podobně je tomu také u správy restrikcí (3 ze 4 pedagogů). Rozsah agendy je ale také určen i pro oblast sociálních služeb. Lze tedy očekávat, že v prostředí speciální školy bude využita jen část celé agendy. I toto se ale jeví jako přínosné.

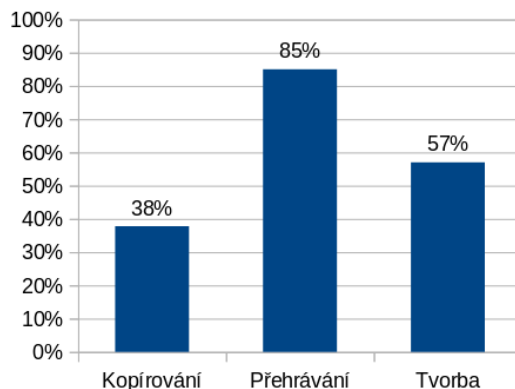
Jak plyne z jejich dalších odpovědí, využití možnosti vedení agendy či správy restrikcí v centrálním nástroji frameworku má smysl až v okamžiku, kdy bude dostupných více aplikací spojených s tímto nástrojem frameworku. I proto lze vývoj dalších aplikací propojitelných s centrálním nástrojem frameworku očekávat jako vhodné pokračování této práce.

Vedení administrativy v souvislosti s jedinou výukovou aplikací skutečně praktické není. Právě malé množství spolupracujících aplikací uvádí také jako další negativum většina (3 ze 4) pedagogů hodnotících správu restrikcí.

Z odpovědi jedné z učitelek lze odhadnout, že ve správě profilů klientů by byl vítaný výběr jednotlivých položek z číselníků.

7.2.3 Výsledky testování Aplikace ANO/NE

Protože byl v rámci této práce kladen větší důraz na vývoj i-CT frameworku, nebylo možné věnovat stejné úsilí s ohledem na úplné využití všech možných návrhových principů počítačové terapie i při vývoji Aplikace ANO/NE. To odráží i skutečnost, že aplikace ANO/NE byla vyvíjena s cílem vytvoření první ukázkové aplikace, která bude schopna komunikovat s nástrojem frameworku. Jedná se tedy zejména o ukázkovou aplikaci. Tato skutečnost se



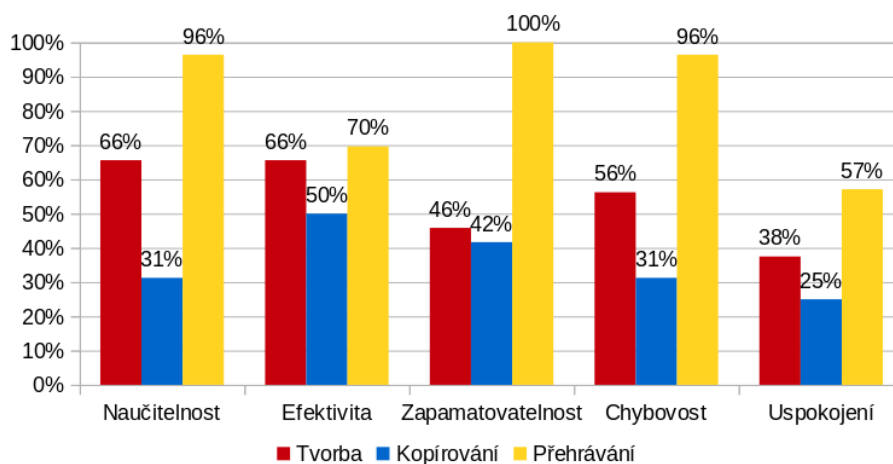
Obrázek 7.3: Celková použitelnost Aplikace ANO/NE

proto promítla u zjištěné koncové použitelnosti aplikace ANO/NE, která byla dostupná v okamžiku testování.

Aplikace ANO/NE byla pro potřeby hodnocení rozdělena do následujících tří oblastí:

- **Tvorba** – Tvorba karet a seznamů párů karet
- **Kopírování** – Kopírování karet a seznamů párů mezi uživateli a online i offline katalogy
- **Přehrávání** – Přehrávání párů při práci s klientem a vlastní volba ze dvou možností

Celkové hodnocení použitelnosti ukazuje graf na obrázku 7.3. Průměrné hodnocení jednotlivých složek ukazuje graf na obrázku 7.4.



Obrázek 7.4: Jednotlivé složky použitelnosti Aplikace ANO/NE

Z odpovědí na otevřené otázky vyplývá, že hlavním nedostatkem Aplikace ANO/NE je nepřehledný systém katalogů. Ten byl přehledný v době kdy existovaly různé katalogy párů, ale jen jediný katalog seznamů párů. Později byla přidána stejná struktura katalogů pro seznamy párů. Tím bylo následně umožněno kopírovat a sdílet kompletní seznamy párů.

Právě tato možnost kopírovat kompletní seznamy párů mezi klienty (a pravděpodobně i mezi zařízeními) byla pro pedagogy skutečně klíčová – 3 ze 4 pedagogů uvedli možnost kopírovat celé seznamy párů mezi přínosy aplikace.



Obrázek 7.5: Levé menu se strukturami katalogů seznamů a párů

Na druhou stranu, všichni pedagogové testující kopírování prohlásili systém katalogů za nepřehledný: „Jednotlivé kategorie se mezi sebou pletou.“

To lze přisuzovat právě dvěma samostatným strukturám katalogů – katalogů karet a katalogů seznamů párů. Odpovídající součást aplikace, levé menu, je zachycena na obrázku 7.5. Je nutno přiznat, že tato dvojitá struktura katalogů skutečně není ideální.

Uživatelé hodnotí jako zbytečně složitý také způsob přidání vlastní karty – nelíbí se jim, že pro úpravu/vytvoření karty musí nejdříve opustit úpravu seznamu párů a přejít do katalogu karet. Nejspíše by bylo vhodné umožnit přejít do úpravy karty přímo z úpravy seznamu párů a umožnit při výběru karty rovnou vytvořit novou kartu.

Uživatelé by rovněž rádi operativně měnili páry v průběhu přehrávání. To by ale přineslo jiné obtíže spojené s možným neoprávněným zásahem klienta do práce vytvořené asistentem. Toto riziko lze považovat za větší ztrátu, než možný přínos z udávané možnosti změny. Proto je i z návrhových principů Počítačové terapie takový zásah znemožněn.

Uživatelé potvrdili že aplikaci lze úspěšně využívat pro komunikaci, podporu globálního čtení i výběr možností klientem. Jako pozitivum uvedli hned ve 4 případech ze 14 velkou plochu karet – na rozdíl od slabikářových aplikací (běžně používaných v tomto prostředí jako substitut uvedené aplikace – dle informací z osobní návštěvy) dává Aplikace ANO/NE oběma kartám plochu celé poloviny dotykové obrazovky. Výběr tak jsou schopni provádět i klienti s horší motorikou, kteří mají problém se na menší karty zacílit.

Z hlediska tvorby karet uživatelé ocenili možnost využití vlastního materiálu – obrázků, fotek a zvukových nahrávek – zejména možnost namluvení zvukové nahrávky samotným klientem.

Přestože se jedná o aplikaci specializovanou na AAK typu ANO/NE, uživatelé by ocenily, kdyby bylo možné nastavovat počet karet v rámci jednoho „páru“ – pravděpodobně pro jednotlivé seznamy párů nebo i jednotlivé „páry“.

7.2.4 Zapracování připomínek z testování a zhodnocení

Přestože byly výsledky testování a závěry z nich dostupné nedlouho před odevzdáním této práce, podařilo se některé zjištěné nedostatky v uživatelském rozhraní alespoň z části odstranit. Další zjištěné nedostatky jsou tak motivací pro další vývoj aplikace a vzhledem k otevřenosti a volné dostupnosti řešení lze toto do budoucna očekávat.

Jednoduchou úpravou obrazovky pro výběr karty byla umožněna úprava a přidávání karet možností přímo z obrazovky pro výběr karty do páru. Namísto aby se karta vybrala ihned kliknutím na kartu, je po kliknutí zobrazena nabídka nabízející mimo výběr karty také její úpravu a odstranění. Současně bylo do hlavičky obrazovky vloženo tlačítko pro

přidání karty, čímž byla obrazovka pro výběr karty rozšířena o většinu možností obrazovky samotného seznamu karet.

Přestože výsledky testování použitelnosti Aplikace ANO/NE vykazují jisté nedostatečnosti u této metriky (z výše uvedených důvodů), lze celkově i zde vytvořené řešení považovat za přínosné – nejen jako příklad propojení výukové aplikace s i-CT frameworkem, ale i s ohledem na šíři poskytovaných funkcí. Ty sice nejsou nejlépe použitelné, ale jsou v tomto řešení alespoň dostupné, což se bohužel v této míře nevyskytuje u ostatních aplikací tohoto druhu (viz srovnání se stávajícími aplikacemi v kapitole 7.1).

Kapitola 8

Závěr

Cílem diplomové práce bylo seznámit se s problematikou speciálních vzdělávacích potřeb u osob s mentálním postižením, s požadavky na vývoj aplikací z pohledu interakce člověka s počítačem a s požadavky vyplývajícími z návrhových principů Počítačové terapie. Prostudovány byly také dostupné platformy pro vývoj mobilních aplikací. V rámci této snahy byly prověřeny možnosti naplnění požadavků kladených na aplikace pro speciální pedagogiku – byly prověřeny způsoby umožňující zabránit mentálně postiženým klientům v rámci práce se vzdělávací aplikací opustit aplikaci a zasáhnout do konfigurace mobilního zařízení.

Výsledkem práce je soustava aplikací – i-CT frameworku a Aplikace pro AAK typu ANO/NE. Je určena k pozdějšímu rozšiřování o další aplikace pro AAK a speciální pedagogiku. Nové aplikace tak mohou čerpat z připraveného základu ve formě i-CT frameworku a nemusejí se zabývat správou, přihlašováním a evidencí klientů a asistentů.

Aplikace ANO/NE byla podrobena srovnání se stávajícími aplikacemi pro AAK typu ANO/NE, z něhož vyšla nejlépe v míře splnění požadavků vyplývajících z principů Počítačové terapie. (viz 7.1)

Jak bylo popsáno v kapitole 7.2, obě aplikace byly otestovány v praxi s relativně pozitivními výsledky. Zatímco v případě aplikace i-CT frameworku výtky směřovaly spíše jen na nedostupnost dalších aplikací spolupracujících s frameworkem, v případě Aplikace ANO/NE nebyli uživatelé spokojeni s oddělením správy karet a správy seznamů párů. Potvrdili však, že aplikaci je možné úspěšně využívat pro komunikaci, podporu globálního čtení i výběr možností klientem.

Závěrem lze považovat obě aplikace za přínosné. Aplikaci i-CT frameworku jak s ohledem na poskytované funkce, tak i s ohledem na zjištěnou použitelnost. Aplikaci ANO/NE lze rovněž považovat za přínosnou. Třebaže jí poskytované funkce nejsou nejlépe použitelné, jsou v tomto řešení alespoň dostupné, na rozdíl od jiných aplikací tohoto druhu. (viz kapitola 7.1).

Obě aplikace jsou nyní k dispozici jako freeware a open source, konkrétně pod licencí GNU GPL verze 3, čímž jsou naplněny požadavky na otevřenost i přístupnost vyplývající z návrhových principů Počítačové terapie.

Nakonec lze přínos této práce spatřovat i v ověření přínosu návrhových principů Počítačové terapie, bez jejichž modelu aplikovaného ve vývojovém procesu při užití SW techniky „Model Driven Development“ by nebylo možné dosáhnout takto pozitivních výsledků.

Literatura

- [1] *Android Developers: Activities* [online]. [cit. 2015-12-31] Dostupné z: <<http://developer.android.com/guide/components/activities.html>>.
- [2] *Android Developers: Intents and Intent Filters* [online]. [cit. 2015-12-31] Dostupné z: <<http://developer.android.com/guide/components/intents-filters.html>>.
- [3] *Android Developers: Providing Proper Back Navigation* [online]. [cit. 2015-12-31] Dostupné z: <<http://developer.android.com/training/implementing-navigation/temporal.html>>.
- [4] *Android Developers: Tasks and Back Stack* [online]. [cit. 2015-12-31] Dostupné z: <<http://developer.android.com/guide/components/tasks-and-back-stack.html>>.
- [5] *Android Interfaces and Architecture* [online]. [cit. 2016-03-08] Dostupné z: <<https://source.android.com/devices/index.html>>.
- [6] *Android Open Source Project* [online]. [cit. 2016-03-08] Dostupné z: <<https://source.android.com/>>.
- [7] *AngularJS Tutorial: What is AngularJS?* [online]. [cit. 2016-04-09] Dostupné z: <http://www.tutorialspoint.com/angularjs/angularjs_overview.htm>.
- [8] *Apache Cordova: Events* [online]. [cit. 2016-01-01] Dostupné z: <<https://cordova.apache.org/docs/en/latest/cordova/events/events.html>>.
- [9] *Apache Cordova: Overview* [online]. [cit. 2015-12-10] Dostupné z: <<https://cordova.apache.org/docs/en/latest/guide/overview/>>.
- [10] *Apple UK – Special Education* [online]. [cit. 2016-05-22] Dostupné z: <<http://www.apple.com/uk/education/real-stories/special-education/>>.
- [11] *Cocoa Core Competencies: Cocoa (Touch)* [online]. [cit. 2016-01-05] Dostupné z: <<https://developer.apple.com/library/ios/documentation/General/Conceptual/DevPedia-CocoaCore/Cocoa.html>>.
- [12] *Developing Mobile Applications with Oracle MAF: Using the Local Database in MAF AMX* [online]. Dostupné z: <<http://docs.oracle.com/middleware/maf213/mobile/develop-maf/maf-sqlite-db.htm>>.
- [13] *Express – Node.js web application framework* [online]. [cit. 2016-05-14] Dostupné z: <<http://expressjs.com/>>.

- [14] *Facebook: Klábosil* [online]. [cit. 2016-01-09] Dostupné z: <<https://www.facebook.com/pages/Klábosil/421887401231584>>.
- [15] *Github: persistence.js* [online]. [cit. 2016-05-14] Dostupné z: <<https://github.com/coresmart/persistencejs>>.
- [16] *How-To Create a Working Kiosk Mode in Android* [online]. StackOverFlow.com, [cit. 2016-01-02] Dostupné z: <<http://www.andreas-schrade.de/2015/02/16/android-tutorial-how-to-create-a-kiosk-mode-in-android/>>.
- [17] *I Can Do Apps: Yes/No from I Can Do Apps* [online]. [cit. 2016-01-05] Dostupné z: <<http://www.icandoapps.com/apps.html>>.
- [18] *iOS Technology Overview: Cocoa Touch Layer* [online]. [cit. 2016-01-05] Dostupné z: <<https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSTechnologies/iPhoneOSTechnologies.html>>.
- [19] *Java Server-Side Programming* [online]. [cit. 2016-04-09] Dostupné z: <<https://www3.ntu.edu.sg/home/ehchua/programming/java/JavaServerFaces.html>>.
- [20] *Klábosil - Oficiální web* [online]. [cit. 2016-01-09] Dostupné z: <<http://www.klabosil.cz/>>.
- [21] *Multer / Expressjs middleware* [online]. [cit. 2016-05-14] Dostupné z: <<https://ewiggin.gitbooks.io/expressjs-middleware/content/multer.html>>.
- [22] *Nikolčina touha: Augmentativní a alternativní komunikace* [online]. [cit. 2015-12-28] Dostupné z: <http://www.nikolcinatouha.cz/?page_id=118>.
- [23] *Oracle Brings Java to iOS Devices (and Android too)* [online]. The Oracle Mobile Platform Blog, [cit. 2016-04-12] Dostupné z: <https://blogs.oracle.com/mobile/entry/oracle_brings_java_to_ios>.
- [24] *Oracle Mobile Application Framework Release 2.2.0.0.0: Get Started* [online]. [cit. 2016-03-02] Dostupné z: <<https://docs.oracle.com/middleware/maf220/mobile/>>.
- [25] *Overriding the Home button - how do I get rid of the choice?* [online]. StackOverFlow.com, [cit. 2015-12-28] Dostupné z: <<http://stackoverflow.com/questions/2079691/overriding-the-home-button-how-do-i-get-rid-of-the-choice>>.
- [26] *Passport – Simple, unobtrusive authentication for Node.js* [online]. [cit. 2016-05-14] Dostupné z: <<http://passportjs.org/>>.
- [27] *patternLock.js* [online]. [cit. 2016-04-03] Dostupné z: <<http://ignitersworld.com/lab/patternLock.html>>.
- [28] *Picture Exchange Communication System and Autism* [online]. [cit. 2015-12-07] Dostupné z: <<http://researchautism.net/interventions/36/picture-exchange-communication-system-and-autism>>.

- [29] *Programming with Objective-C: About Objective-C* [online]. [cit. 2016-01-05] Dostupné z: <<https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>>.
- [30] *Red Hat Mobile Application Platform* [online]. [cit. 2015-12-28] Dostupné z: <<http://www.redhat.com/en/technologies/mobile>>.
- [31] *The Swift Programming Language: About Swift* [online]. [cit. 2016-01-05] Dostupné z: <https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/>.
- [32] *Usability 101: Introduction to Usability* [online]. [cit. 2016-05-14] Dostupné z: <<https://www.mngroup.com/articles/usability-101-introduction-to-usability/>>.
- [33] *What is... the Dalvik virtual machine?* [online]. [cit. 2015-12-28] Dostupné z: <<http://www.electronicweeky.com/blogs/eyes-on-android/what-is-the-dalvik-virtual-machine-2011-10/>>.
- [34] *The Yes/No Series* [online]. [cit. 2015-12-28] Dostupné z: <<http://teachinglearnerswithmultipleneeds.blogspot.cz/2012/11/the-yesno-series-part-one.html>>.
- [35] Fiala, J.; Kočí, R. : Computer as Therapy in role of alternative and augmentative communication. *Proceedings of 4th International Conference on Advanced in Computing and Emerging E-learning Technology (ICACET 2014)*. Singapore, roč. 18, č. 1, 2014: s 34–42, iISSN 2091-1610.
- [36] Fiala, J.; Kočí, R. *Počítačová terapie jako koncept nové formy terapie pro osoby s mentálním postižením: teorie i praxe* [online]. roč. 6, 2014 Dostupné z: <http://www.fit.vutbr.cz/research/view_pub.php.cs?id=10718>. ISSN 1803-537X.
- [37] Fiala, J.; Kočí, R.; Vejtasa, O.; aj. *i-CT Framework (počítačová terapie) verze 1.1.0* [online]. [cit. 2016-01-09] Dostupné z: <<http://www.fit.vutbr.cz/~koci/prods.php.cs?id=405¬itle=1>>.
- [38] Fowler, M. *Model Driven Software Development* [online]. Dostupné z: <<http://martinfowler.com/bliki/ModelDrivenSoftwareDevelopment.html>>.
- [39] Hejlová, D. *Rozvoj komunikační schopnosti u žáků s poruchou autistického spektra* (diplomová práce). Brno, Masarykova univerzita, Pedagogická fakulta., 2011. Dostupné z: <http://is.muni.cz/th/209855/pedf_m/Rozvoj_komunikacni_schopnosti_u_zaku_s_poruchou_autistickeho_spektra.doc>.
- [40] Kouřil, J. *Syn se nenaučil komunikovat pomocí piktogramů. Existují jiné možnosti alternativní komunikace?* [online]. [cit. 2015-12-05] Dostupné z: <<http://www.praha.apla.cz/syn-se-nenaucil-komunikovat-pomoci-piktogramu-existuji-jine-moznosti-altern-komunikace.html>>.
- [41] Langer, J. *Znakový jazyk jako prostředek komunikace mezi učitelem a žákem* [online]. [cit. 2016-04-03] Dostupné z: <http://ruce.cz/clanky/_702.pdf>.

- [42] Lazar; Feng; Hochheiser. *Research methods in Human Computer Interaction*. Wiley, 2010. ISBN 0470723378.
- [43] Meca, V. *Aplikace pro výuku znaku do řeči pro osoby s mentálním postižením* (bakalářská práce). Brno, FIT VUT v Brně, 2014.
- [44] Tůma, J. *Aplikace pro YES/NO alternativní komunikaci pro osoby s mentálním postižením* (bakalářská práce). Brno, FIT VUT v Brně, 2014.
- [45] Vejtasa, O. *Aplikace pro alternativní a augmentativní komunikaci pro osoby s mentálním postižením* (diplomová práce). Brno, FIT VUT v Brně, 2014.
- [46] Yaghmour, K. *Embedded Android: Porting, Extending, and Customizing*. O'Reilly Media, Inc., 2013. ISBN 978-1-4493-0829-2.
- [47] Říhová, L. *Apple special needs summit Praha* [online]. [cit. 2016-05-22] Dostupné z: <<http://www.i-sen.cz/clanky/apple-special-needs-summit>>.
- [48] Říhová, L. *Co je iSen?* [online]. [cit. 2016-05-19] Dostupné z: <<http://www.i-sen.cz/onas>>.
- [49] Říhová, L.; Jelínková, I. *iSEN in SEN* [online]. [cit. 2015-12-28] Dostupné z: <<http://www.i-sen.cz/rubriky/clanky/specialni-potreby>>.
- [50] Šafrová, A. *Reedukace a kompenzace specifických poruch učení (a chování) u žáků základní školy* [online]. [cit. 2015-12-20] Dostupné z: <https://is.muni.cz/el/1421/podzim2012/PGVCH10/um/Reedukace_a_kompenzace_SPU_CH__OSTRAVA_5_2012.ppt>.
- [51] Ševčík, D. *Domain Driven Design* (diplomová práce). Brno, Masarykova univerzita, Fakulta informatiky., Brno, 2009. Vedoucí práce Ráček Jaroslav. Dostupné z: <http://is.muni.cz/th/173376/fi_b/domain_driven_design.pdf>.
- [52] Škodová, E.; Jedlička, I. *Klinická logopedie*. Portál, 2003. ISBN 80-7178-546-6.
- [53] Švarcová, I. *Mentální retardace*. Portál, 2011. ISBN 978-80-7367-889-0.

Přílohy

Seznam příloh

A	Obsah CD	57
B	Instalační manuál – Android	58
C	Instalační manuál – iOS	59
D	Uživatelský manuál i-CT frameworku	60
	D.1 Vytvoření a nastavení účtu uživatele – klienta	60
	D.2 Práce se záznamy profilů klienta	62
E	Uživatelský manuál Aplikace ANO/NE	63
	E.1 Vytváření karet možností	64
	E.2 Práce s katalogy	65
	E.3 Vytváření seznamů párů	66

Příloha A

Obsah CD

- `sources.zip` – zdrojové soubory implementovaných aplikací
- `iCTframework.apk` – aplikace i-CT frameworku zkompilevaná pro Android
- `YesNoApp.apk` – Aplikace ANO/NE zkompilevaná pro Android
- `xkalin03.zip` – zdrojové soubory textu této práce
- `xkalin03.pdf` – elektronická verze textu této práce

Příloha B

Instalační manuál – Android

Příloha popisuje způsob instalace aplikace i-CT frameworku a Aplikace ANO/NE na zařízení platformy Android.

1. Proveďte instalaci APK balíčků i-CT frameworku a aplikace ANO/NE:
 - (a) Proveďte instalaci APK balíčku `iCTframework.apk`
 - (b) Proveďte instalaci APK balíčku `YesNoApp.apk`
2. Vytvořte asociaci mezi i-CT frameworkem a Aplikací ANO/NE:
 - (a) Otevřete Aplikaci ANO/NE a v menu zvolte **Instalovat do iCT-frameworku**.
 - (b) Měla by se zobrazit aplikace i-CT frameworku a měla by se zobrazit zpráva potvrzující úspěšnou instalaci aplikace v rámci frameworku.
 - (c) Aplikaci frameworku můžete nyní opustit.
3. Vypněte zamykací obrazovku systému Android. Postup se liší mezi zařízeními různých výrobců a verzemi systému. Na testovaných zařízeních je postup následující:
 - (a) Alcatel: **Settings – Security – Set up screen lock – None**
 - (b) Xiaomi: **Settings – Additional settings – Developer options – Skip screen lock**
4. Nastavte aplikace i-CT frameworku jako domovskou obrazovku systému (launcher).
 - (a) Alcatel: **Settings – Applications – All – iCTFramework / Launcher – Clear defaults**, následně, po použití tlačítka Domů, budete dotázáni na aplikaci, která má být domovskou obrazovkou – vyberte **iCTFramework**
 - (b) Xiaomi: **Settings – Installed apps – Defaults – Launcher – iCTFramework**

Příloha C

Instalační manuál – iOS

Příloha popisuje způsob instalace aplikace i-CT frameworku a Aplikace ANO/NE na zařízení platformy iOS. Vzhledem k požadavkům na podpisy balíčků je nutné aplikaci přeložit ze zdrojových kódů.

1. Za pomoci JDeveloper otevřete soubor projektu i-CT frameworku a zkompilujte:
`iCTFramework/iCTFramework.jws`
2. Jde-li o první kompilaci, otevřete vygenerovaný soubor projektu XCode:
 - (a) `iCTFramework/deploy/iOS1/temporary_xcode_project/Oracle_ADFmc_Container_Template.xcodeproj`
 - (b) Při pokusu o spuštění aplikace na zařízení nabídne XCode opravu problému: chybějící klíč pro nasazení – potvrďte opravu jeho vygenerováním.
 - (c) Nyní by již mělo být možné provádět nasazení do iTunes přímo, bez XCode.
3. V rámci JDeveloper proveďte kompilaci a nasazení aplikace do iTunes:
 - (a) **Application – Deploy – iOS1 – Deploy to iTunes**
 - (b) Aplikace by se nyní měla zobrazit v iTunes mezi aplikacemi zařízení.
 - (c) Tlačítky **Install** a **Apply** proveďte instalaci do zařízení.
4. Proveďte stažení závislostí, kompilaci a spuštění Aplikace ANO/NE:

```
git submodule init
git submodule update
cd YesNoApp
bower install
ionic config build
cordova platform add ios
ionic run ios
```
5. První pokus o spuštění selže – je třeba opakovat postup výše pro vygenerování klíče pro nasazení a spuštění opakovat. V tomto případě je soubor projektu v adresáři `YesNoApp/platforms/ios`
6. Vytvořte asociaci mezi i-CT frameworkem a Aplikací ANO/NE – v menu Aplikace ANO/NE zvolte **Instalovat do iCT-frameworku**.

Příloha D

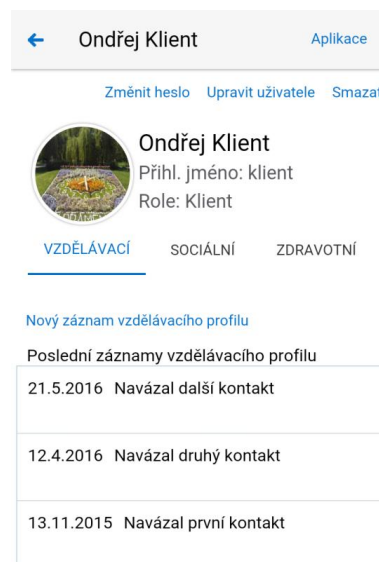
Uživatelský manuál i-CT frameworku

Příloha popisuje způsob používání aplikace i-CT frameworku. Postup předpokládá již nainstalovaný i-CT framework a související aplikace podle přílohy C.

- Po spuštění aplikace i-CT frameworku by se měla zobrazit přihlašovací obrazovka. Přihlaste se:
 - Zadejte uživatelské jméno uživatele.
 - Pokud je uživatelský účet nastavený aby vyžadoval k přihlášení heslo, zadejte heslo a potvrďte tlačítkem Přihlásit.
 - Jinak zadejte přihlašovací gesto tažením po matici teček.
- Po přihlášení asistenta nebo administrátora je zobrazen seznam uživatelů.
- Kliknutím na uživatele v seznamu uživatelů je zobrazena obrazovka profilu uživatele. (obrázek D.1)
- Tlačítkem **Aplikace** se dostanete na seznam aplikací uživatele. Zobrazené aplikace závisejí na nainstalovaných aplikacích (viz příloha C) a na nastavení zobrazených aplikací daného uživatele (viz část D.1).

D.1 Vytvoření a nastavení účtu uživatele – klienta

- V seznamu uživatelů použijte tlačítko **Přidat**.
- Na obrazovce úpravy uživatele (obrázek D.2) zadejte údaje uživatele, jehož účet chcete vytvořit:



Obrázek D.1: Obrazovka profilu uživatele

- **Úroveň uživatele** určuje oprávnění uživatele – zda-li jde o klienta, asistenta nebo správce.
- **Správce uživatele** určuje, které uživateli je asistentem daného klienta nebo nadřízeným daného asistenta.
- **Jazyk** určuje jazyk uživatelského rozhraní, které bude použito při přihlášení daného uživatele.
- Je-li přepínač **Může opustit framework** v pravé modré poloze, uživateli je po přihlášení v seznamu uživatelů zobrazeno tlačítko **Ukončit**. To umožňuje opustit prostředí i-CT frameworku a jeho aplikací a spouštět následně libovolné aplikace zařízení. Tato volba by měla být zapnuta jen pro administrátory, protože umožňuje uživateli zasahovat do konfigurace zařízení, ale je-li nutné umožnit spouštět aplikace neintegrováné s i-CT frameworkem, je nezbytné ji zapnout i pro asistenty.
- Položka **Kopírovat profily z** umožňuje při vytváření účtu nového klienta zkopírovat kompletní profily klienta z účtu jiného klienta.
- Položka **Aplikace uživatele** umožňuje vybrat aplikace, které bude možné pro uživatele spouštět. Nejsou-li zde v nabídce žádné aplikace, ověřte že byl správně proveden krok 2 instalačního manuálu z přílohy C.

3. Po uložení tlačítkem **Uložit** v pravém horním rohu je nově vytvořený účet zobrazen v seznamu uživatelů.

Pokud má mít vytvořený uživatel možnost se samostatně přihlásit, je nutné mu nastavit přihlašovací gesto a volitelně také heslo. Tento krok přeskočte pro klienty, kteří nebudou pracovat se zařízením samostatně – pokud budou aplikace klientovi spouštěny jeho asistentem, nebude heslo ani gesto nikdy potřeba. Gesto naopak bude potřebovat každý asistent nebo správce.

1. V seznamu uživatelů klikněte na uživatele, jehož gesto nebo heslo chcete nastavit – bude zobrazena hlavní strana profilu uživatele.
2. Na hlavní straně profilu uživatele klikněte na tlačítko **Změnit heslo** – dostanete se na stránku pro nastavení hesla a gesta:
 - Chcete-li aby pro přihlášení uživatele bylo požadováno heslo, zaškrtněte pole **Požadovat heslo k přihlášení**. Tím bude zobrazeno pole pro zadání hesla. Zde heslo vyplňte.
 - I pokud vyžadujete aby k přihlašování bylo používáno heslo, je nutné nastavit i gesto – pro návrat z aplikací spuštěných v režimu klienta může být vyžadováno zadání gesta asistenta. Pro nastavení gesta pouze nakreslete gesto na matici na stránce změny hesla.

Obrázek D.2: Obrazovka úpravy uživatele

- Obojí potvrďte tlačítkem **Uložit** vpravo nahoře.

Z hlavní strany profilu uživatele se můžete dostat také do úpravy uživatele tlačítkem **Upravit uživatele**. Zde je možné měnit údaje uživatele nastavené při jeho vytvoření.

D.2 Práce se záznamy profilů klienta

1. V seznamu uživatelů zvolte klienta, jehož záznamy chcete spravovat.
2. Zvolte záložku **Sociální**, **Zdravotní** nebo **Vzdělávací**, podle toho, s kterým profilem klienta chcete pracovat.
3. Za pomoci odkazu **Nový záznam** můžete vytvořit nový záznam:
 - **Datum** určuje, ke kterému datu se vytvářený záznam vztahuje. Záznamy jsou podle data také řazeny. Záznamy které jsou podle data nejnovější se také zobrazují na hlavní straně profilu uživatele.
 - Ostatní položky záznamy jsou textové a jejich vyplnění je plně v kompetenci asistenta – aplikace na jejich obsah neklade žádné požadavky.
 - Vytváření položky dokončete pomocí tlačítka **Uložit** v pravém horním rohu obrazovky.
4. Kliknutím na libovolný z posledních záznamů profilu se dostanete k úplnému seznamu záznamů daného profilu daného klienta.
5. Kliknutím na záznam v úplném seznamu můžete záznam upravit. Postup úpravy je totožný s vytvářením nového záznamu. Na obrazovce úprav záznamu je možné záznam také smazat tlačítkem **Smazat**.
6. V úplném seznamu záznamů najdete také tlačítko **Kopírovat**, které umožňuje kopírovat jednotlivé záznamy profilů mezi klienty nebo i v rámci jednoho klienta:
 - (a) Použijte tlačítko **Kopírovat** v pravém horním rohu obrazovky úplného seznamu.
 - (b) Na obrazovce „Převzít z klienta“ vyberte klienty, z kterých chcete záznamy kopírovat a potvrďte tlačítkem **Pokračovat**.
 - (c) Na obrazovce „Kopírovat záznamy“ vyberte konkrétní záznamy, které chcete kopírovat.
 - (d) Na obrazovce „Kam kopírovat“ vyberte klienty, do jejichž profilu mají být kopie zvolených záznamů uloženy.

Příloha E

Uživatelský manuál Aplikace ANO/NE

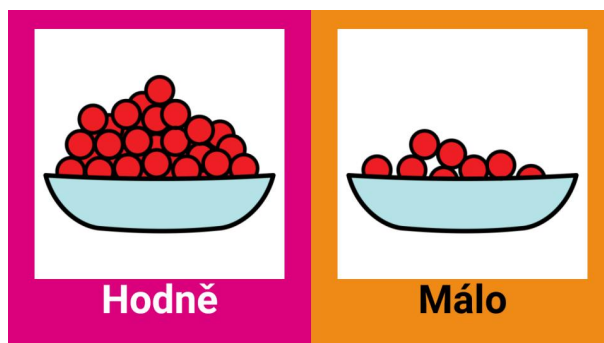
Příloha popisuje způsob používání Aplikace ANO/NE (Aplikace pro AAK typu ANO/NE). Postup předpokládá již nainstalovanou Aplikace ANO/NE. Aplikace nemusí být spuštěna skrze i-CT framework. Je-li však spuštěna skrze i-CT framework, zobrazený obsah (seznamy párů a karty možností) bude záviset na uživateli vybraném v rámci frameworku.



Obrázek E.1: Obrazovka seznamů párů a levé menu

1. Po spuštění aplikace ANO/NE je zobrazena nabídka seznamů párů (obrázek E.1). Je-li aplikace spuštěna skrze i-CT framework, budou zobrazeny seznamy párů vybraného klienta.
2. Kliknutím na seznam párů je zobrazena nabídka umožňující zobrazit **Náhled přehrávání**, **Upravit seznam** nebo seznam **Odstranit**. Je-li aplikace spuštěna v režimu klienta, je po kliknutí na seznam párů rovnou spuštěno přehrávání seznamu.
3. V přehrávání nebo v náhledu přehrávání (obrázek E.2) je možné se pohybovat mezi jednotlivými páry seznamu tažením po obrazovce doleva nebo doprava. Aktuální pozici v seznamu lze sledovat na indikátoru dole uprostřed.
4. Kliknutím na kartu možnosti (levou nebo pravou polovinu obrazovky páru) je zvýrazněna zvolená možnost a přehrána připojená zvuková nahrávka.

5. Přehrávání je možné ukončit za pomoci hardwarového tlačítka zpět. V případě přehrávání v režimu klienta je pro opuštění přehrávání nutné zadat gesto asistenta, který aplikaci pro klienta spustil.



Obrázek E.2: Obrazovka přehrávání seznamu karet

E.1 Vytváření karet možností

Karta možnosti tvoří jednu polovinu každého páru. Zahrnuje nápis, obrázek, barvu a zvuk. Může být využita v různých kontextech v rámci různých seznamů párů. Jejich vytváření je proto odděleno od vytváření samotného seznamů párů. Dostupné karty nezávisí na klientovi vybraném při spuštění aplikace ANO/NE ale jen na přihlášeném asistentovi.

1. Otevřete menu kliknutím na symbol tří linek v levém horním menu.
2. Zvolte položku **Moje karty** pro práci s katalogem karet, který můžete využívat při tvorbě seznamů párů pro své klienty.
3. Kliknutím na kartu v katalogu zobrazíte nabídku, která umožní vybranou kartu **Upravit**, **Kopírovat** do jiného katalogu (katalogu sdíleného mezi asistenty nebo on-line katalogu) nebo ji **Odstranit**.
4. Tlačítkem **Přidat** v pravém horním rohu můžete vytvořit novou kartu:
 - (a) **Název karty** určuje označení karty v seznamu karet. Je viditelné pro asistenta a lze podle něj vyhledávat karty.
 - (b) **Nápis na tlačítku** určuje text, který bude v přehrávání páru zobrazen pod obrázkem. Jeho barva bude černá nebo bílá, v závislosti na světlosti barvy pozadí.
 - (c) **Barva pozadí** určuje barvu pozadí karty - části karty která není vyplněna obrázkem. Barvu vyberte pomocí výběrového nástroje skrytého pod tlačítkem **Vybrat barvu**.
 - (d) **Obrázek** definuje soubor obrázku, který bude umístěn uprostřed karty. Tlačítkem **Vyfotit** lze spustit aplikaci fotoaparátu a skrze ni obrázek vyfotit. Tlačítko **Z galerie** umožňuje vybrat obrázek z úložiště (např. SD karty) v zařízení.
 - (e) **Zvuk po stisknutí** definuje soubor zvukové nahrávky, která bude přehrána při aktivaci karty – když mentálně postižený klient zvolí danou kartu. Tlačítkem **Nahrát** lze pořídit zvukovou nahrávku z mikrofonu zařízení.
 - (f) Vytváření nebo úpravu karty dokončíte tlačítkem **Uložit** v pravém horním rohu.

E.2 Práce s katalogy

Katalogy jsou úložiště karet. V rámci aplikace jsou k dispozici katalogy **Moje karty** a **Sdílené karty**. Zatímco první uvedený katalog zahrnuje karty aktuálně přihlášeného asistenta, druhý uvedený je sdílený mezi asistenty používajícími stejné zařízení. K těmto katalogům je možné přidávat Online katalogy – katalogy umístěné na serveru v internetu nebo v místní síti. Následující postup předpokládá, že máte k dispozici již zprovozněný server, jeho adresu a k němu platné přihlašovací údaje.

1. Otevřete správu online katalogů – otevřete menu symbolem tří linek v levém horním menu a zvolte položku **Online katalogy**.
2. Tlačítkem **Přidat** přejděte k nastavení údajů pro připojení k přidávanému serveru:
 - **Pojmenování serveru** definuje označení serveru v seznamu katalogů – v levém menu a v nabídce kopírování karet.
 - **Adresa serveru** definuje URL adresu pro přístup k nainstalované instanci serveru. Např: `http://192.168.1.12/yesno/`
 - **Přihlašovací jméno** a **Heslo** definují přihlašovací údaje pro přihlášení k serveru. Nemají žádný vztah k přihlašovacím údajům uživatelů i-CT frameworku.
3. Nastavení serveru uložte a zavřete tlačítkem **Uložit**.
4. Zobrazte karty poskytované serverem – v levém menu (po kliknutí na ikonu tří linek v levém horním rohu) klikněte na položku nově vytvořeného katalogu/záznamu serveru.
5. Pokud se karty zobrazily, je spojení se serverem nakonfigurované správně. Jinak se vraťte zpět k nastavení serveru.

Obdobným postupem je možné servery ze seznamu zase odstraňovat nebo upravovat jejich údaje pro připojení. Po nastavení připojení je možné kopírovat karty z online katalogu do místních katalogů:

1. V seznamu karet online katalogu klikněte na kartu kterou chcete kopírovat.
2. V zobrazené nabídce vyberte cíl kopírování – **Moje karty** – a potvrďte tlačítkem **Kopírovat**.

Karty je možné kopírovat také z místních katalogů zpět:

1. V místním katalogu klikněte na kartu kterou chcete kopírovat.
2. V nabídce vyberte volbu **Kopírovat**.
3. V zobrazené nabídce vyberte cíl kopírování a potvrďte tlačítkem **Kopírovat**.

E.3 Vytváření seznamů párů

Seznam párů sdružuje páry, které jsou součástí jediného přehrávání – je možné snadno přecházet mezi páry v rámci jednoho seznamu. Pro přechod mezi seznamy může být naproti tomu požadováno gesto asistenta.

1. Přejděte na obrazovku „Seznamy párů“ – otevřete menu symbolem tří linek v levém horním menu a zvolte položku **Seznamy párů**.
2. Tlačítkem **Přidat** přejděte na „Vlastnosti seznamu“ nového seznamu.
3. Vyplňte název seznamu a potvrďte tlačítkem **Uložit**.
4. V **Seznamu párů** klikněte na nově vytvořený seznam a v nabídce zvolte **Upravit seznam**.
5. Tlačítkem **Přidat pár** přidávejte řádky – páry seznamu.
6. Kliknutím na pozici (u nového páru označeny nápisem „Klikněte pro výběr karty...“) proveďte výběr karty pro danou stranu daného páru. Karty se vybírají z katalogu **Moje karty**, jehož plnění je popsáno v částech **E.1** a **E.2**.
7. Tlačítkem **Náhled** se dostanete na **Náhled přehrávání**, kde si můžete prohlédnout výsledek své práce – seznam párů tak, jak jej uvidí klient.
8. Tlačítkem **Vlastnosti** můžete upravit nastavení seznamu – v současnosti pouze jeho název.
9. Změny jsou ukládány okamžitě – úpravu seznamu párů můžete kdykoli opustit tlačítkem s ikonou šipky vlevo nahoře.

Stejně jako karty možností je možné i seznamy párů kopírovat:

1. Na obrazovce seznamů párů klikněte na seznam který chcete kopírovat.
2. V nabídce vyberte volbu **Kopírovat**.
3. V zobrazené nabídce vyberte cíl kopírování a potvrďte tlačítkem **Kopírovat**.



Obrázek E.3: Obrazovka pro úpravu seznamu párů