



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**APLIKACE AAK SPECIÁLNÍ PEDAGOGIKY PRO I-CT
FRAMEWORK**

APPLICATION OF AAC IN SPECIAL EDUCATION FOR I-CT FRAMEWORK

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VEDOUcí PRÁCE

SUPERVISOR

MARTIN ČAJÁNEK

Ing. JIŘÍ FIALA

BRNO 2017

Abstrakt

Tato bakalářská práce se zaměřuje na analýzu potřeb komunikace ve speciální pedagogice při práci s osobami s mentálním postižením a na aplikaci, která danou problematiku řeší. Vyvíjená aplikace umožňuje sestavování vět pomocí výběru obrázkových symbolů reprezentujících slova, či jejich vlastní tvorbu a následnou hlasovou syntetizaci hotové věty. Je určena pro mobilní zařízení se systémy Android a iOS, byla vyvíjena v souladu s návrhovými principy počítačové terapie a slouží jako nadstavba pro i-CT Framework, rámec zastřešující a zjednodušující práci s aplikacemi určené pro speciální pedagogiku.

Abstract

This bachelor thesis focuses on analysis of communication needs in special education during sessions with mental disabled people and also focuses on application, which addresses this problematic. Developed application enables users to create sentences by choosing picture symbols representing words or their own creating and subsequent voice synthesizing of constructed sentence. This application is designed for mobile devices with Android and iOS operating systems and was develop in accordance with “Computer as Therapy” design principles and serves as extension (add-on) for i-CT Framework, which encapsulates and simplifies interaction and management among applications designed for special education.

Klíčová slova

Mentální postižení, porucha komunikace, speciální vzdělávací potřeby, počítačová terapie, alternativní a augmentovaná komunikace, i-CT Framework, vývoj na mobilní zařízení, Oracle MAF, Android, iOS

Keywords

Mental disability, communication disorder, special educational needs, computer therapy, alternative and augmentative communication, i-CT Framework, mobile device development, Oracle MAF, Android, iOS

Citace

ČAJÁNEK, Martin. *APLIKACE AAK SPECIÁLNÍ PEDAGOGIKY PRO I-CT FRAMEWORK*. Brno, 2017. 62 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jiří Fiala

APLIKACE AAK SPECIÁLNÍ PEDAGOGIKY PRO I-CT FRAMEWORK

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jiřího Fialy. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Martin Čajánek
17. května 2017

Poděkování

Tímto bych rád poděkoval panu Ing. Jiřímu Fialovi, za ochotu a trpělivost při vedení této bakalářské práce. Dále bych rád poděkoval pedagogům ze Speciální zvláštní školy v Poděbradech za možnost otestování použitelnosti implementované aplikace v cílovém prostředí, jmenovitě zde zejména Mgr. L. Říhové a I. Jelínkové.

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 3 |
| 2 | Speciální pedagogika a její potřeby | 4 |
| 2.1 | Speciální pedagogika | 4 |
| 2.2 | Mentální postižení | 5 |
| 2.2.1 | Klasifikace mentálního postižení | 5 |
| 2.2.2 | Kombinované postižení | 6 |
| 2.2.3 | Výchova a vzdělávání mentálně postižených | 6 |
| 2.2.4 | Poruchy komunikace | 8 |
| 2.3 | Speciální pedagogické potřeby | 9 |
| 2.3.1 | Principy AAK | 9 |
| 2.3.2 | Používané prostředky pro AAK | 10 |
| 2.4 | Analýza požadavků | 14 |
| 3 | Mobilní platformy a prostředky pro vývoj | 15 |
| 3.1 | Platformy na mobilní zařízení | 15 |
| 3.1.1 | Android | 15 |
| 3.1.2 | iOS | 16 |
| 3.2 | Vývojové prostředky | 17 |
| 3.2.1 | Xamarin | 17 |
| 3.2.2 | Ionic | 17 |
| 3.2.3 | Oracle Mobile Application Framework | 17 |
| 3.3 | Architektura Model-View-Controller | 18 |
| 3.4 | Návrhové principy a framework počítačové terapie | 18 |
| 3.4.1 | O projektu počítačové terapie | 18 |
| 3.4.2 | Návrhové principy počítačové terapie | 19 |
| 3.4.3 | Framework počítačové terapie | 20 |
| 4 | Návrh aplikace | 22 |
| 4.1 | Návrh struktury aplikace | 22 |
| 4.2 | Návrh uživatelského prostředí | 23 |
| 4.2.1 | Návrh obrazovek aplikace | 25 |
| 4.3 | Návrh databáze | 30 |
| 4.3.1 | Ukládané informace a omezení | 30 |
| 4.4 | Správa symbolů | 32 |
| 4.5 | Objektový návrh | 35 |
| 4.5.1 | Implementační balíček <i>ApplicationController</i> | 35 |
| 4.5.2 | Implementační balíčky <i>ViewController</i> | 35 |

| | | |
|----------|---|-----------|
| 4.6 | Použitá licence aplikace | 36 |
| 5 | Implementace | 37 |
| 5.1 | Životní cyklus aplikace a prezenční vrstvy | 37 |
| 5.1.1 | Implementace hlavní obrazovky | 39 |
| 5.1.2 | Implementace obrazovky historie | 41 |
| 5.1.3 | Implementace správy symbolů | 41 |
| 5.1.4 | Implementace kreslení a psaní symbolu | 42 |
| 5.2 | Databáze | 43 |
| 5.2.1 | Inicializace databáze | 43 |
| 5.2.2 | Vložení záznamu o novém uživateli | 44 |
| 5.2.3 | Přístup k databázi | 44 |
| 5.3 | Převod textu na řeč | 44 |
| 6 | Testování a zhodnocení aplikace | 46 |
| 6.1 | Zhodnocení | 46 |
| 7 | Závěr | 47 |
| | Literatura | 48 |
| A | Obsah přiloženého CD | 51 |
| B | Testování použitelnosti aplikace a porovnání s existujícími řešeními | 52 |
| B.1 | Ověření použitelnosti | 52 |
| B.1.1 | Popis prostředí ověřování | 52 |
| B.1.2 | Fáze ověřování použitelnosti | 53 |
| B.1.3 | Zhodnocení výsledků ověření použitelnosti | 57 |
| B.2 | Porovnání se současnými řešeními | 58 |
| B.2.1 | Zhodnocení porovnání aplikace | 59 |

Kapitola 1

Úvod

Speciální pedagogové potřebují při práci s klienty s mentálním postižením či s poruchou komunikace použít různé pomůcky alternativní a augmentativní komunikace k dosažení uspokojivého výsledku. Zatímco dříve bylo standardem využívání papírových piktogramů, dnes, díky rozvoji a nižším cenám mobilních zařízení, již mohou speciální pedagogové využít moderní a efektivnější nástroje než bylo v minulosti možné. Mezi nejpokročilejší z nich patří multimediální aplikace ovládané pomocí tabletů a chytrých mobilních telefonů. Nicméně velké množství těchto aplikací má pro české uživatele vážné nedostatky a chyby, například chybějící podpora českého jazyka, vysoká cena a komplikovaná správa dat uživatelů.

Cílem této bakalářské práce je tedy bližší seznámení se specifiky alternativní a augmentativní komunikace a speciálních vzdělávacích potřeb u osob s mentálním postižením. Také je cílem navrhnout aplikaci, která řeší s těmito osobami možnost komunikace pomocí sestavování vět výběrem piktogramů včetně hlasového výstupu složených vět a správy piktogramů, tedy jejich přesouvání mezi jejich kategorií či uživateli a jejich odstraňování a přidávání nových. A následně pak i tuto aplikaci implementovat ji na platformy Android a iOS.

Text této práce je možné rozdělit do dvou částí – teoretické a praktické. Teoretická část se zabývá v kapitole 2 stavem a specifickými potřebami mentálně postižených a tedy i poruchou komunikace a vysvětluje důvody existence aplikací či jiných řešení pro alternativní a augmentativní komunikaci. Následující kapitola 3 se týká analýzy existujících a použitých prostředků pro vývoj mobilních multiplatformních aplikací a také shrnuje návrhové principy a framework počítačové terapie, což je jeden z nejdůležitějších důvodů vývoje této aplikace. Poslední teoretická část práce se nachází v kapitole 4, kde je popisován návrh vyvíjené aplikace AAC.

Praktická část této práce se nachází v kapitole 5, týkající se implementace aplikace AAC v Oracle Mobile Application Framework, a řešení vyvstávajících problémů během samotného vývoje. Kapitola 6 a část v příloze A, se zabývá testováním vyvíjené aplikace v cílovém prostředí klientů s mentálním postižením, zhodnocením míry použitelnosti aplikace a jejího možného rozšíření.

Kapitola 2

Speciální pedagogika a její potřeby

V této kapitole se pojednává o oblasti speciální pedagogiky, osobách, které ji vykonávají a jaké překážky musí řešit. Také se zabývá mentálním postižením a jeho klasifikací podle závažnosti. Toto postižení též negativně postihuje řeč, hlavní složku mezilidské komunikace a v mnoha případech ji znemožňuje. Tato kapitola se tedy také zaměřuje na principy a možnosti alternativní a augmentativní komunikace, kterými lze problémy s komunikací postižených osob kompenzovat.

2.1 Speciální pedagogika

Na současný obor speciální pedagogiky lze nahlížet ze dvou úhlů:

- V užším pojetí je speciální pedagogika pedagogickou disciplínou, která se zabývá edukací dětí, žáků a dospělých osob se speciálními vzdělávacími potřebami a zkoumáním formativních (výchovných) a informativních (vzdělávacích) vlivů na tyto jedince.
- V širším pojetí je možno na speciální pedagogiku nahlížet jako na interdisciplinární obor zabývající se péčí o jedince minoritních skupin, obyvatelstva se zřetelem na edukaci, reedukaci a kompenzaci, diagnostiku, terapeuticko-formativní intervenci, rehabilitaci, inkluzi (integraci) a socializaci, či resocializaci, prevenci a prognostiku osob se zdravotním postižením a zdravotním či sociálním znevýhodněním [35].

Speciální pedagogové pečují a jednájí s klienty, kteří mají nějaký druh postižení či znevýhodnění. Osoby, jež mají zájem vykonávat toto povolání, musí získat danou kvalifikaci, tedy splňovat standardy, které jsou specifikovány zákonem 563/2004 Sb., o pedagogických pracovních. Podle § 18 tohoto zákona získává speciální pedagog odbornou kvalifikaci vysok školským vzděláním získaným studiem v akreditovaném magisterském studijním programu v oblasti pedagogických věd zaměřených na speciální pedagogiku či zaměřené na pedagogiku předškolního věku a doplňujícím studiem k rozšíření odborné kvalifikace, případně studijního oboru pedagogika a doplňujícím studiem k rozšíření odborné kvalifikace uskutečňovaném vysokou školou [17].

Pracovní náplň speciálního pedagoga je naplňována ve čtyřech zásadních oblastech[17]:

1. **Depistážní činnost** – vyhledávání dětí se speciálními vzdělávacími potřebami a jejich zařazení do speciální pedagogické péče.
2. **Diagnostická činnost** – diagnostika speciálních vzdělávacích potřeb dané osoby, včetně shromažďování údajů o osobní a rodinné anamnéze, analýza získaných údajů a jejich vyhodnocení. Stanovení individuálního plánu podpory.

3. **Intervenční činnost** – provádění či zajištění krátkodobé i dlouhodobé speciální pedagogické vzdělávací činnosti. Průběžné vyhodnocování účinnosti navržených opatření a dle potřeby realizace úprav. Zabezpečení průběžné komunikace a kontaktů s odinou dítěte (případně se zákonným zástupcem). Úpravy školního prostředí, zajištění speciálních pomůcek a didaktických materiálů. Konzultace s pracovníky specializovaných poradenských a dalších pracovišť.
4. **Metodická a koordinační činnost** - příprava a průběžná úprava podmínek pro integraci žáků se speciálními vzdělávacími potřebami ve škole – koordinace speciálně pedagogických poradenských služeb ve škole.

2.2 Mentální postižení

Osoby trpící mentální retardací jsou postiženy snížením úrovně rozumových schopností, v psychologii označované jako inteligence. Ačkoliv stále nebylo na definici tohoto často užívaného termínu dosaženo shody a psychologové se již velmi dlouhou dobu pokoušejí o jeho přesnější vymezení, obecně je chápána inteligence jako schopnost učit se z předchozí zkušenosti a přizpůsobit se novým životním podmínkám a situacím.

2.2.1 Klasifikace mentálního postižení

Při klasifikaci mentálního postižení se v současné době užívá 10. revize Mezinárodní klasifikace nemocí, zpracovaná Světovou zdravotnickou organizací v Ženevě, která vstoupila v platnost roku 1992. Podle této klasifikace se mentální postižení dělí do těchto základních kategorií[33]:

- **Lehké mentální postižení** (klasifikace F70, IQ v rozmezí 50–69) – U osob, charakterizovaných touto úrovní postižení lze očekávat, že dosáhnou schopnosti užívat řeč účelně v každodenním životě, udržovat smysluplnou konverzaci a dosáhnout částečné nezávislosti v osobním životě (jídlo, oblékání, hygienické návyky), ačkoliv jejich vývoj je oproti normě mnohem pomalejší. Toto postižení se týká přibližně 85 % mentálně postižených.
- **Středně těžké mentální postižení** (F71, IQ 35–49) – Jedinci, zařazení do této kategorie mají výrazně opožděn rozvoj chápání a užívání řeči. Také je opožděna jejich zručnost a tedy i schopnost se o sebe starat. Jejich pokroky ve škole jsou do určité míry limitované, ale někteří z nich si mohou osvojit základy čtení, psaní a počítání, nicméně tyto schopnosti nezvládnou vylepšit na běžnou úroveň, takže zřídka mají možný úplně samostatný život. V dospělosti nicméně jsou osoby se středním stupněm retardace většinou schopni vykonávat jednoduchou manuální práci, pokud jsou jejich úkoly dostatečně strukturovány a je zajištěn odborný dozor.
- **Těžké mentální postižení** (F72, IQ 20–34) – Většina jedinců z této kategorie (na rozdíl od těch z kategorie předchozí) trpí značnou poruchou motoriky či jinými přidruženými vadami, prokazující přítomnost klinicky signifikantního poškození případně vadného vývoje ústředního nervového systému. U těchto jedinců je možnost výchovy a vzdělávání velice omezené, nicméně včasná systematická a dostatečně kvalifikovaná rehabilitační a obecně vzdělávací péče může výrazně přispět k rozvoji jejich motoriky, rozumových a komunikačních schopností a tedy i k celkovému zlepšení kvality jejich života.

- **Hluboké mentální postižení** (F73, IQ je nižší než 20) - Jedinci postižení tímto stupněm mentální retardace jsou těžce omezeni ve schopnosti porozumět požadavkům a instrukcím ostatních osob nebo jim vyhovět. Většina těchto osob jsou výrazně omezených v pohybu, případně úplně imobilní. Postižení jsou schopni pouze primitivní neverbální komunikace. Mají minimální schopnost pečovat o své základní potřeby a vyžadují stálou asistenci.
- **Jiné mentální postižení** (F78) – Tato kategorie by měla být použita pouze v případech kdy je stanovení stupně intelektové retardace pomocí obvyklých metod obzvláště nesnadné či nemožné. Takové případy mohou nastávat například v případě přidruženého sensorického či somatického poškození – u nevidomých, neslyšících, nemluvicích, u osob s těžkými poruchami chování, s autismem či u těžce tělesně postižených osob.
- **Nespecifikované mentální postižení** (F79) – Označením touto kategorií se používá, když je u daného pacienta mentální postižení prokázáno, ale není dostatek informací, aby bylo možné pacienta do některé ze dříve uvedených kategorií zařadit.

2.2.2 Kombinované postižení

V některých případech netrpí klienti pouze mentální retardací, ale mohou mít i kombinované postižení, čímž se následně jejich stav zhoršuje a možnosti omezují.

Autismus

Autismus je podle Mezinárodní klasifikace nemocí zařazován mezi pervazivní vývojové poruchy. Je běžně charakterizován trojicí příznaků, které se většinou objeví do 3 let věku dítěte [23]:

- Neschopností vzájemné společenské interakce.
- Omezenou komunikací (ne nutně porucha řeči).
- Omezeným a opakujícím se repertoárem zájmů a aktivit či agresivitou.

Ačkoliv je onemocnění poměrně vzácné, obecně se uvádí, že je diagnostikováno v poměru asi u jednoho až pěti z 10 tisíc narozených dětí [33], velká část autistů trpí také mentální retardací. Je diagnostikována přibližně u tří čtvrtin lidí trpících autismem (lehká až střední okolo 24 %, těžká až hluboká u zhruba 47 % pacientů) [34]. Ačkoliv může mít běžný pacient trpící autismem určitou mírou poruchy řeči, pokud trpí zároveň i mentální retardací, jeho porucha řeči se prohlubuje.

2.2.3 Výchova a vzdělávání mentálně postižených

U lidí s mentálním postižením je výchova a vzdělávání chápána jako celoživotní proces. Ačkoliv se i u nepostižených jedinců uznává nutnost celoživotního vzdělávání, zvyšování kvalifikace, prohlubování jazykových a dalších znalostí, u mentálně postižených probíhají kognitivní procesy podstatně pomaleji než u ostatní populace, tudíž stále výrazněji vystupuje potřeba neustálého vzdělávání. Pouze permanentní rozvíjení, stálé opakování a prohlubování jejich znalostí a dovedností a soustavné vedení ke stále komplexnějšímu poznávání okolních skutečností je může přiblížit k lepší integraci do společnosti.[33]

Podle zákona č. 561/2004 o předškolním, základním, středním a vyšším vzdělávání (školského zákona) je vzdělávání v České republice založeno na zásadách [33]:

1. rovného přístupu každého státního občana České republiky nebo jiného členského státu Evropské unie ke vzdělávání bez jakékoliv diskriminace z důvodu rasy, barvy pleti, pohlaví, jazyka, víry a náboženství, národnosti, etnického nebo sociálního původu, majetku, rodu a zdravotního stavu nebo jiného postavení občana,
2. zohledňování vzdělávacích potřeb jednotlivce,
3. vzájemné úcty, respektu, názorové snášenlivosti, solidarity a důstojnosti všech účastníků vzdělávání.

Osoby s mentálním postižením mohou během svého života navštěvovat různé vzdělávací a výchovné instituce [33]:

- Předškolní vzdělávání dětí s mentální retardací institucionálně zajišťují mateřské školy a speciální vzdělávací školy.
- Žáci s mentální retardací ve věku povinné školní docházky se vzdělávají[33]:
 - Formou individuální integrace v základních školách.
 - Formou skupinové integrace v základních školách.
 - Ve škole samostatně zřízené pro žáky se zdravotním postižením (speciální škole), tedy v základní škole praktické (dříve zvláštní škola) nebo v základní škole speciální (dříve pomocná škola).
 - Kombinací výše uvedených forem.
- Po ukončení povinné školní docházky v základních školách a základních školách praktických mohou navštěvovat další instituce:
 - Odborná učiliště.
 - Praktické školy.
 - * Dvouleté.
 - * Jednoleté (určeno pro žáky s těžkým zdravotním postižením).
- Celoživotní vzdělávání lidí s mentálním postižením:
 - Večerní školy (určeny nejen pro další vzdělávání absolventů pomocných, případně praktických škol, ale otevřeny i pro ty, kteří povinnou školní docházku neabsolvovali).
 - Kurzy k doplnění vzdělání (jedna z dalších forem vzdělávání dospělých lidí s mentálním postižením).
 - Aktivační centra (školská zařízení zaměřená na poskytování celoživotního vzdělávání mladistvým a dospělým se závažnými formami mentálního postižení, kteří nemohou využívat jiné formy celoživotního vzdělávání určeného ostatní populaci).

Jednou z netradičních forem péče o mentální postižené je i pěstounská péče. Ačkoliv není tato forma více vžitá, v některých případech je možné, že si rodina ke svým vlastním dětem přijme do pěstounské péče i dítě s mentálním postižením, o které se jeho vlastní rodiče nemohou starat. Někteří lidé se v rámci pěstounské péče starají za úplatu či za úhradu

nutných nákladů i o dospělé jedince s mentálním postižením [33].

Dalším typem netradiční péče o osoby s mentálním postižením jsou služby dočasné výpomoci. Jsou určeny zejména pro rodiny nebo osamělé osoby, starající se o mentálně postižené jedince, kteří se dostanou do situace, kdy danou péčí nemohou zvládat, nejčastěji kvůli úrazu, náhlému onemocnění nebo jinému závažnému zhoršení zdravotního stavu [33]. Právě ve výše zmíněných institucích lze při práci s klienty používat aplikaci vyvíjenou v rámci této práce. Kompenzovala by jejich částečnou či plnou poruchu komunikace a díky tomu by mohli lépe vyjadřovat své pocity a myšlenky.

2.2.4 Poruchy komunikace

„Řeč lze definovat jako biologickou vlastnost člověka, systém, kterým lze přenášet informace pomocí jazyka“ [25].

Řeč je stále ještě nejdůležitějším prostředkem mezilidské komunikace a právě osoby s mentálním postižením mají prakticky vždy problémy s rozvojem řeči a jsou oproti normě výrazně opožděni [25].

Narušená komunikační schopnost v důsledku poruch vývoje řeči

Vývoj řeči nelze odlišit od rozvoje a zdokonalování ostatních schopností člověka. A stejně tak jako v ostatních schopnostech se v něm může vyskytnout porucha.

V případě osob s mentálním postižením dochází nejčastěji k omezenému vývoji řeči, kdy se výrazně projevuje opoždění vyjadřovacích schopností. Tato porucha je nejvíce patrná v obsahové stránce řeči (chápání významu slov). Opoždění vývoje řeči v důsledku mentální retardace pak typicky doprovází i opoždění jemné a hrubé motoriky.

U mentálně postižených dětí se liší úroveň opoždění podle stupně postižení intelektu dané osoby [25]:

- Pacienti s lehkým mentálním postižením mají ve většině případů dlouho přetrvávající vadnou výslovnost, ačkoliv někteří ji zvládnou dobrou napodobovací schopností kompenzovat a chybná výslovnost se tak nemusí vůbec projevit. Na rozdíl od závažnějších forem mentálního postižení může jejich řeč v dospělosti dosáhnout normy.
- Pacienti se středně těžkým postižením mají nejnápadnější symptomy v morfologicko-syntaktické a lexikálně-sémantické jazykové rovině, tj. hovoří v jednoduchých, obsahově chudých a gramaticky nesprávných větách.
- Pacienti s nejtěžšími formami mentální retardace rozvinou svou řeč pouze částečně (postižení užívají pouze nejjednodušší věty, slovní spojení či jednotlivá slova) ale v nejextrémnějších případech nejsou schopni využívat řeč efektivně ke komunikaci vůbec a zůstává tak na prvosignální úrovni (nemluví, ale motoricky vyjadřuje porozumění) nebo na úrovni pudového žvatlání.

Mezi nejčastější poruchy řeči patří:

- **Dyslálie (patlavost)** je neschopnost používat jednotlivé hlásky či skupiny v mluvené řeči podle stanovených ortoepických norem. Hlávka je tvořena na nesprávném místě. Je to nejčastější poruchou komunikačních schopností u dětí [25].

- **Afázie** je úplná nebo částečná ztráta schopnosti produkce nebo porozumění řeči v důsledku ložiskového poškození mozku, tato porucha může postihnout i dospělé jedince s vybudovanou řečí [25].
- **Dysfázie** je porucha postihující vývoj řeči, který je tedy nedostatečný či žádný. Projevuje se zejména u dětí a trvá již od narození dítěte. Její příčinou bývá porucha centrálního nervového systému (mozku), zejména jeho sluchové a řečové oblasti. Jejími znaky je porucha motoriky, sluchového a zrakového vnímání a samozřejmě i opožděný vývoj řeči [18].
- **Mutismus (nemluvnost)** není řečovou vadou, stav, kdy dítě či osoba umí mluvit, ale odmítá to. Existuje i poddruh selektivní mutismus, kdy dítě v určitém prostředí či s určitými osobami mluví, ale v jiných případech nikoliv. Případně může znamenat i dočasnou či trvalou ztrátu již vyvinutého řečového projevu následkem šoku či psychického úrazu [25].
- **Balbuties (kocktavost)** je postižení zabraňující plynulé řeči dané osoby. Jeho nejčastějším projevem je opakování první slabiky slova, také doprovázen netypickými pohyby hlavou, křečovitě sevření obličeje atd. [25].

2.3 Speciální pedagogické potřeby

Cílem augmentativní a alternativní komunikace (AAK) je přechodně nebo trvale kompenzovat projevy poruchy a postižení u osob se závažným postižením řeči, jazyka a psaní. [11]. Alternativní komunikační systémy se používají jako náhrada mluvené řeči. Jejich výhodou je snižování tendence k pasivitě klientů, zvyšuje jejich aktivitu a rozšiřuje možnosti, umožňuje komunikovat i osobám, které mají velké potíže při vyjadřování. [24]

Nicméně alternativní způsoby komunikace mají i své nevýhody, jsou společensky méně využitelné, jejich uživatelé jsou jejím používáním odděleni od majoritní části společnosti, jež tyto systémy neovládá.

2.3.1 Principy AAK

Při budování podpůrných nebo náhradních komunikačních systémů je třeba využívat všech schopností uživatele. Zásadou je vždy to, aby způsob dorozumívání byl co nejpřirozenější, jak je to jen v dané situaci možné. Alternativní komunikace tedy využívá [11]:

- Systémy bez pomůcek:
 - cílené pohledy očí, mimika,
 - gesta, gestikulace, manuální znaky.
- Systémy s pomůckami:
 - předměty,
 - fotografie,
 - systémy grafických symbolů,
 - komunikační tabulky,
 - písmena a psaná slova,
 - technické pomůcky s hlasovým výstupem,
 - počítače, tablety a jiné elektronické přístroje.

2.3.2 Používané prostředky pro AAK

V této kapitole jsou popsány prostředky, které se používají nyní pro alternativní a augmentativní komunikaci. Na konci této podkapitoly jsou tyto prostředky porovnány s aplikací vyvíjenou v rámci této bakalářské práce.

VOKS

Výměnný obrázkový komunikační systém (celosvětově známý jako *Picture Exchange Communication System* - PECS) využívá piktogramy, zpravidla natištěné na papíru, znázorňující běžná slova, pomocí kterých mohou klienti sestavovat věty či větné celky, reprezentující jejich sdělení, které by nebyli schopni jiným způsobem předat ostatním.

Jednotlivé piktogramy systému VOKS mohou být umístěny na tzv. komunikačních tabulkách (obrázek č. 2.1). Existuje několik komunikačních tabulek různé barvy podle toho, pro jaký druh obrázku jsou určeny, například oranžová komunikační tabulka je určena pro podstatná jména, modrá pro přídavná jména a zelená pro slovesa. Výměnný obrázkový komunikační systém má nicméně i své nevýhody. V případě jeho potřeby je nutné ho mít neustále k dispozici a může ho využívat pouze jeden klient v jeden okamžik. Stejně tak nelze okamžitě kdekoliv vytvořit nové piktogramy a ty samotné časem degradují v kvalitě a je nutné je vyrobít znovu.



Obrázek 2.1: Výměnný obrázkový komunikační systém¹

¹Zdroj obrázku: <http://luckyho-svet.bloger.cz/obrazky/luckyho-svet.bloger.cz/sdc18953.jpg> (dostupné online)

Klábosil

Klábosil je původní česká elektronická aplikace pro iPad (tablet od společnosti Apple), určená pro AAK, byla navržena za pomoci pracovníků Speciální základní školy Poděbrady, kteří mají dlouhodobou zkušenost v oblasti využití tabletů ve speciální pedagogice (zakladatelé odborné komunity i-SEN)[9][1].

Přináší typicky české výrazy a vychází z českých tradic, nicméně pro podobnost jazyka a možnost změnit textu u symbolů může být využívána také na Slovensku.

Uživatel aplikace, tedy osoba se speciálními potřebami, může pomocí piktogramů skládat vlastní věty a následně je nechat přečíst hlasovým syntetizátorem.

Aplikace obsahuje více než tisíc symbolů a umožňuje třídit tyto pojmy do kategorií, podkategorií či kontextů. Také je možné rozšířit slovní zásobu libovolným množstvím slov, případně vytvořenou fotografií či obrázkem z vlastního alba. Aplikace je dostupná zdarma z obchodu iTunes.



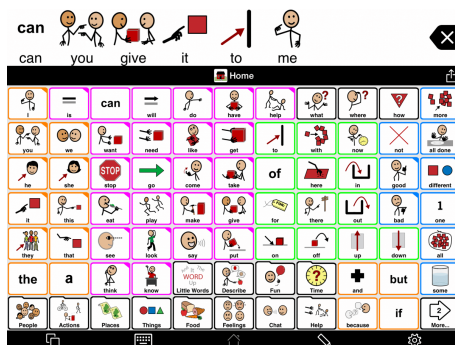
Obrázek 2.2: Ukázka prostředí aplikace Klábosil²

²Zdroj obrázku: <http://2.bp.blogspot.com/-G9Tt95CHbN0/UStYoTfgmoI/AAAAAAAAACT8/MZmDsrp7CqA/s1600/Klábosil+2.PNG> (dostupné online)

Proloquo2Go

Proloquo2Go je považována za jednu z nejpropracovanějších a nejpoužívanějších aplikací pro AAK, ačkoliv je dostupná pouze pro zařízení s operačním systémem iOS. Stejně jako ve výše zmíněné aplikaci Klábosil, si uživatel pomocí piktogramů skládá věty, nicméně tato aplikace podporuje pouze angličtinu, francouzštinu a španělštinu, tedy neexistuje její česká lokalizace.

Na druhé straně, aplikace obsahuje více než 20000 symbolů a nabízí až šedesát různých hlasů (včetně dětských či možnosti vytvoření z vlastního hlasu) na zvukovou syntetizaci [13]. Proloquo2Go lze zakoupit za \$249.99 z obchodu iTunes.



Obrázek 2.3: Ukázka prostředí aplikace Proloquo2Go³

AAC speech communicator

Tato aplikace je alternativa předchozích aplikací dostupná na operační systém Android. Podporuje francouzský a anglický jazyk a obsahuje přes 5000 symbolů. Aplikaci lze stáhnout zdarma v obchodě Google Play.



Obrázek 2.4: Ukázka prostředí aplikace AAC speech communicator⁴

³Zdroj obrázku: <http://www.assistiveware.com/sites/default/files/image1-optim.png> (dostupné online)

⁴Zdroj obrázku: https://i3.wp.com/img.android-apk.org/imgs/7/e/1/7e1e475ccd13707142a6351e0a75ac05_screen.png?w=300 (dostupné online)

EasyTalk

EasyTalk je aplikace se stejným účelem jako předchozí uvedené a byla vyvíjena v rámci diplomové práce *Aplikace pro alternativní a augmentativní komunikaci pro osoby s mentálním postižením* na fakultě informačních technologií VUT v Brně v roce 2014 pod vedením školitele této práce a Dr. Kočího [17]. Vývoj zde byl prováděn dle první verzí návrhových principů počítačové terapie [26].

Aplikace je dostupná pro platformy iOS a Android a umožňuje sestavení vět na základě výběru piktogramů včetně hlasového výstupu sestavovaných vět, jež nicméně je implementován pouze pomocí hlasových nahrávek nikoliv akustické syntézy řeči. Obsahuje českou lokalizaci a možnost tvorby vlastních symbolů. Aplikaci lze zdarma stáhnout na stránce projektu počítačové terapie na serveru portálu Sourceforge [36].



Obrázek 2.5: Ukázka prostředí aplikace Easy Talk⁵

Srovnání AAK aplikací

Zde, v tabulce č 2.1 jsou porovnány vlastnosti existujících aplikací určených pro alternativní a augmentativní komunikaci s vlastnostmi aplikace, jež je vyvíjena v rámci této bakalářské práce.

| Název aplikace/vlastnosti | Klábosil | Proloque2Go | AAC speech communicator | Easy Talk | AAC |
|---------------------------------|----------|-------------|-------------------------|-----------|-----|
| Dostupné zdarma | Ano | Ne | Ano | Ano | Ano |
| Česká lokalizace | Ano | Ne | Ano | Ano | Ano |
| Vytvoření vlastních symbolů | Ano | Ano | Ano | Ano | Ano |
| Přizpůsobení prostředí aplikace | Ne | Ano | Ne | Ano | Ano |
| Dostupné pro Android | Ne | Ne | Ano | Ano | Ano |
| Dostupné pro iOS | Ano | Ano | Ne | Ano | Ano |
| Bez nutnosti správy uživatelů | Ne | Ne | Ne | Ne | Ano |
| Možnost výukového režimu | Ne | Ne | Ne | Ne | Ano |

Tabulka 2.1: Srovnání vlastností AAK aplikací

⁵Zdroj obrázku: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=119286 (dostupné online)

2.4 Analýza požadavků

Aplikace vyvíjená v rámci této práce má vycházet z návrhových principů počítačové terapie, které se ukazují jako vhodné pro zkvalitnění vývojového procesu aplikací speciální pedagogiky, naplnění jejích potřeb a nakonec zvýšení vlastní koncové použitelnosti v praxi [26]. Některé z těchto návrhových principů, jsou u řady aplikací speciální pedagogiky interpretovatelné obdobným způsobem, proto byly již dříve převedeny na společné modely, které daly za vznik tzv. frameworku počítačové terapie – i-CT Framework. Ten zakládá automatizaci vývoje aplikací speciální pedagogiky postaveném na tomto frameworku a vede vyvíjené koncové aplikace k vyšší použitelnosti. S ohledem na výhody tohoto přístupu [26] budeme z tohoto přístupu vycházet i v této práci.

S přihlédnutím na výše uvedené a zjištěné nedostatky uvedené v předchozí kapitole, byly definovány následující požadavky na aplikaci:

- Schopnost zpracovat URL schéma přijaté při spuštění aplikace z i-CT Frameworku.
- Možnost spustit aplikaci v režimu správy, kdy lze upravit nastavení aplikace, či v režimu přehrávání, kdy je možné pouze používat základní funkce aplikace a klient nemá možnost žádné změnit nastavení.
- Zobrazení uživatelského prostředí aplikace podle informací získaných z URL schéma.
 - Přizpůsobení textů uživatelského rozhraní podle informace o používaném jazyku klienta.
- Možnost upravit uživatelské prostředí.
 - Možnost změny barvy pozadí uživatelského prostředí a permanentní uložení této informace pro daného uživatele.
 - Možnost změny velikosti piktogramů a permanentní uložení této informace.
- Možnost vytvořit nový piktogram, případně novou kategorii piktogramů, způsob manipulace s piktogramy a kategoriemi – možnost přidávání jich uživatelům či jejich mazání.
- Možnost spuštění výukového režimu aplikace – omezený režim, kdy se může uživatel lépe seznámit s prostředím aplikace.
- Sestavení věty pomocí piktogramů reprezentujících slova či větné celky.
- Možnost vyslovení sestavené věty pomocí hlasového syntetizátoru.
- Uložení vyslovené věty do historie, každý uživatel má vlastní historii vyslovených vět.
- Pokud byla při spuštění aplikace z i-CT Frameworku zvolena možnost k časovému omezení běhu aplikace, po jejím vypršení se aplikace ukončí a navrátí zpět do i-CT Frameworku.

Kapitola 3

Mobilní platformy a prostředky pro vývoj

V této kapitole jsou popsány hlavní platformy na mobilních zařízeních, na které je aplikace vyvíjena a také vývojové prostředky, které existují na trhu.

3.1 Platformy na mobilní zařízení

Mezi mobilními zařízeními se v nynější době prosazuje pouze několik operačních systémů, což také zjednodušuje práci vývojářům, jelikož mohou zaměřit svou pozornost na méně platforem, nicméně se stále dostatečně velkým počtem potencionálních uživatelů. V nynější době (rok 2017) ovládají více než 98 % trhu s mobilními zařízeními pouze tři platformy, Android, iOS a Windows Phone [16]. Aplikace vyvíjená v rámci této bakalářská práce je vyvíjena k používání na prvních dvou zmíněných platformách.

3.1.1 Android

Android je operační systém určený zejména pro mobilní zařízení, který je založený na jádře Linuxu. Vývoj tohoto systému je veden společností Google pod hlavičkou konsorcia firem Open Handset Alliance (OHA) a je dostupný jako otevřený software (*open source*), tedy je možné ho při dodržení stanovených podmínek upravovat pro své účely. Tohoto využívají zejména výrobci mobilních telefonů a tabletů, jako například Samsung či Huawei.

Architektura

Architektura operačního systému Android je rozdělena do pěti vrstev a každá z nich má svůj účel a nemusí být přímo oddělena od ostatních vrstev.

- Linuxový Kernel – jádro operačního systému a nejnižší vrstva architektury. Tvoří abstraktní vrstvu mezi využívaným hardwarem (zařízení se systémem Android se mohou vybaveným hardwarem výrazně lišit) a software vyšších vrstev [8].
- Knihovny – nativní knihovny operačního systému napsány v C/C++ jazycích a obsahující základní funkce systému, například Open GL a SGL pro práci s grafikou, SQLite pro práci s daty, Webkit vykreslovací jádro pro webový prohlížeč, Freeetype pro vykreslování písma či SSL pro šifrování a zabezpečení přenosu dat [8].

- Android Runtime – tato vrstva slouží primárně pro běh aplikací. Obsahuje Dalvik Virtual Machine (aplikační virtuální stroj – stará se o převod kódu aplikace do nativního kódu). Toto řešení bylo zvoleno především z licenčních důvodů, jelikož Java VM a jeho součásti nejsou volně šiřitelné a také bylo třeba optimalizovat virtuální stroj pro mobilní zařízení, k čemuž Java Virtual Machine nebylo uzpůsobeno. Překlad aplikace napsané pro Android starší než verze 4.4 probíhá zkompileováním zdrojového Java kódu do Java byte kódu pomocí stejného kompilátoru jako je používán v případě překladu Java aplikací a následně se překompiluje Java byte kód pomocí Dalvik kompilátoru a výsledný Dalvik byte kód je spuštěn na Dalvik Virtual Machine (DVM). Od verze Android 4.4 se již používá dopředná kompilace, tedy zdrojový kód aplikace se přeloží do Java byte kódu a následně do byte kódu Dalviku. Během instalace se pak jedinkrát finálně zkompileje do nativního kódu daného zařízení [3].
- Application Framework – vrstva obsahující další knihovny, v Javě, tvořící vlastní systémové API, tedy soubor funkcí umožňující programátorovi pracovat s prvky operačního systému. Jedná se zejména o přístup ke grafickým prvkům uživatelského rozhraní, zpřístupnění dat v jiných aplikacích, práci s notifikacemi v upozorňovacím stavovém řádku atd. [30].
- Aplikace – na nejvyšší vrstvě architektury operačního systému Android jsou samotné aplikace, ať už ty defaultní systémové (správa kontaktů, posílání SMS zpráv, kalendář atd.) či tzv. aplikace třetích stran.

3.1.2 iOS

iOS je mobilní operační systém vytvořený a používaný exkluzivně společností Apple Inc. Původně využitý pouze pro mobilní telefony iPhone, nyní se již používá i v tabletech iPad, multimediálních přehrávačích iPod touch, a operační systémy digitálních mediálních přijímačů Apple TV a chytrých hodinkách Apple Watch jsou také založeny na iOS.

Architektura

Mobilní operační systém iOS je odlehčená verze operačního systému macOS, využívaného v notebookech MacPro a osobních počítačích iMac a Mac Pro, tedy se jedná o systém UNIXového typu. Systém se dělí na čtyři základní vrstvy zajišťující jeho základní funkčnost a poskytují vývojářům API pro vývoj aplikací:

- Cocoa Touch – vrstva obsahující nejdůležitější frameworky při vývoji aplikací. Mezi služby poskytované v této vrstvě patří např. multitasking, ochrana dat, notifikace, rozpoznávání gest, sdílení souborů či podpora externích zobrazovacích zařízení.
- Media – tato vrstva umožňuje vytváření graficky pokročilejších aplikací využívající přehrávání animací, videí či zvuků.
- Core Services – tato vrstva podporuje vysokoúrovňové služby jako například lokální služby, SQLite databázi, podporu zpracování XML či plateb v aplikaci.
- Core OS – poskytuje nízkoúrovňové funkce ostatním technologiím, které na ní staví.

Systém iOS je velice uzavřený systém neumožňující uživateli podobně velkou svobodu s úpravou systému a práci s daty jako například výše uvedený systém Android. Aplikace

je možné instalovat pouze z oficiálního obchodu App Store, kde se nachází pouze aplikace, které prošly schvalovacím procesem společnosti Apple. Také není možné do telefonu nahrát hudební soubory, ty je nutné stáhnout z obchodu iTunes.

3.2 Vývojové prostředky

Jedním z posledních trendů při vývoji mobilních aplikací je zaměření se na multiplatformní vývoj, kdy většina zdrojového kódu je sdílána pro více platforem. Na trhu je několik různých softwarových řešení stručně popsanych v této podkapitole.

3.2.1 Xamarin

Xamarin je nástroj, umožňující vývojářům vytvářet aplikace v jazyku C# pro tři nejrozšířenější platformy (iOS, Android a Windows). Xamarin je původně založen na open source projektu Mono X.

Při práci s Xamarin frameworkem lze používat i další nástroje k vývoji – Xamarin Test Cloud (pro Unit a UI testy), Xamarin Profiler (monitoring paměti) a další. Od roku 2016, kdy byla společnost Xamarin zakoupena společností Microsoft je možné využívat Xamarin pro studentské účely a projekty v menších týmech zdarma [22], dříve bylo využití zpoplatněno od určité délky zdrojového kódu.

3.2.2 Ionic

Ionic je SDK, nebo-li soubor nástrojů, pomocí něhož je možné vyvíjet hybridní aplikace. Od nejnovější verze 2.0.0 (vydáno v prosinci 2016) je nově podporován vývoj aplikací pro Windows 10. Dále lze vyvíjet pro iOS 7 a výše a pro Android 4.1 a výše. Nejdůležitější součástí – Ionic framework je postaven na bázi AngularJS a pro sestavení aplikací na cílovou platformu využívá Cordovu.

Ionic framework ctí přístup „*Write once, deploy anywhere*“ (neboli „napiš jednou, nasad kamkoliv“), tedy se snaží vyřešit odlišnosti mezi platformami přímo na úrovni frameworku, aby nebylo nutné je řešit při vývoji vlastních aplikací.

Kromě Ionic frameworku také obsahuje další nástroje jako například Ionic Creator (nástroj pro tvorbu uživatelského rozhraní), analytické nástroje přizpůsobené pro hybridní aplikace či nástroj pro nasazování aktualizací do produkce bez nutnosti čekat na schválení na App Store/Google Play [37].

3.2.3 Oracle Mobile Application Framework

Oracle Mobile Application Framework (Oracle MAF) je hybridní mobilní framework, dostupný zdarma, na kterém lze vyvíjet aplikace pro různé platformy (iOS, Android, Windows 10), jež sdílí velkou část zdrojového kódu. Při vývoji v Oracle MAF je využíván programovací jazyk Java, skriptovací jazyk JavaScript a značkovací jazyk HTML5. Také podporuje práci s databází SQLite.

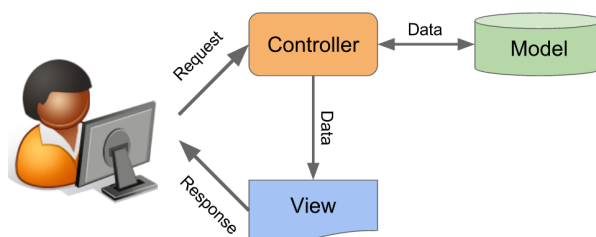
Tento framework je založen na principech softwarové architektury Model-View-Controller, který je popsán v kapitole 3.3.

3.3 Architektura Model-View-Controller

Ve výše zmíněném frameworku Oracle MAF je využívána softwarová architektura Model-View-Controller. Tento návrhový vzor rozděluje aplikaci logicky do tří hlavních částí:

- *Model* – data a jejich související operace.
- *View* – prezentace dat.
- *Controller* – řadič, řídící tok událostí v programu (aplikační logika).

Toto rozdělení bylo vytvořeno tak, aby každou z částí šlo upravovat samostatně a dopad změn byl na ostatní části co nejmenší [21].



Obrázek 3.1: Diagram Model-View-Controller¹

3.4 Návrhové principy a framework počítačové terapie

V této podkapitole jsou popsány principy a myšlenky počítačové terapie a také i-CT Framework, který byl na jejich základě vytvořen.

3.4.1 O projektu počítačové terapie

Projekt počítačové terapie představuje nový směr výzkumu v oblasti informačních technologií pro potřeby speciální pedagogiky a zároveň přináší novou formu terapie pro osoby s mentálním postižením. Autorem projektu počítačové terapie je Ing. Jiří Fiala [26].

Zmíněný projekt vznikl přímo v prostředí osob s mentálním postižením už od roku 2012 a je dále rozvíjen na Fakultě informačních technologií VUT v Brně s podporou společnosti Red Hat [14], neziskového sektoru zaměřeného na vzdělávání a terapie osob s mentálním postižením za pomoci prostředků ICT a s odbornou pomocí komunity iSEN [12].

Projekt počítačové terapie má v obecném slova smyslu za cíl vhodnou aplikaci prostředků informačních technologií na vznik jednotného standardu pro vývoj SW i HW, který bude snadno plikovatelný do praxe (speciální pedagogiky, resp. sociálních služeb), bude zdarma veřejně přístupný a povede k efektivnímu použití bez potřeby důkladného školení v přirozeném prostředí osob s mentálním postižením.

Při využití výše uvedeného ve speciální pedagogické edukaci nebo terapii má pak taková terapie dlouhodobý charakter a slouží ke kompenzaci nedostatků, včetně napomáhání ke snížení dopadu mentálního postižení (například poruchy komunikace). Dochází tak k syntéze metod z různých odvětví [26][27][28]. V konkrétním slova smyslu se ve výzkumu počítačové

¹Zdroj obrázku: <https://daveh.io/blog/the-model-view-controller-pattern/the-mvc-pattern.png> (dostupné online)

terapie jedná o předložení stejnojmenných SW/HW návrhových principů [28], které při užití vhodných metod např. i ze softwarového inženýrství (např. MDE [29]) vedou také ke zkvalitnění vývojového procesu i u koncové aplikace, která je pak při dodržení jednotného standardu na daném HW lépe použitelná a dostupná při práci v cílové skupině.

3.4.2 Návrhové principy počítačové terapie

Návrhové principy počítačové terapie byly dlouhodobě utvářeny na základě analýzy reálných požadavků a expertních zkušeností získaných z oblasti speciální pedagogiky. Vhodnost jejich závaznosti pro vývoj aplikací speciální pedagogiky je opakovaně ověřována [19].

Tyto principy jsou dále rozšiřovány. Současnou verzi čtrnácti návrhových principů lze stručně shrnout do několika priorit [10]:

- **Daný účel aplikace:** Před začátkem vývoje aplikace by měl být jasně definovaný cíl, kterým může být například kompenzace určitých nedostatků jedince či rozvinutí jeho schopností.
- **Bezpečnost informací:** Vývojáři by měli při návrhu aplikace dodržovat zásady pro bezpečnost a ochranu klientů, tedy například osob s mentálním postižením. Aplikace by měla dále podporovat různé režimy v závislosti na přihlášeném uživateli, konkrétněji režim pro klienta, asistenta a technického správce a zamezit snadnému či nechtěnému přechodu mezi nimi.
- **Otevřenost:** Při vývoji by měl být kladen důraz na využití otevřených technologií, společně s veřejným zpřístupněním zdrojových kódů, aby mohla být aplikace volně rozšiřována, vylepšována a mohly na ni navázat další užitečné aplikace. Vývojář a použitá licence aplikace by neměli být nepřístupní k začlenění takových vylepšení zpět do své původní aplikace.
- **Multiplatformita:** Aplikace by měla být dostupná na více mobilních platformách. Vývoj aplikace by měl být veden multiplatformním způsobem, což znamená, že zdrojový kód aplikace bude možné zkompilovat pro všechny podporované platformy dle potřeby bez nutné úpravy kódu.
- **Rozšiřitelnost:** Vyvíjená aplikace by měla mít jasně definovanou architekturu a měla by být implementována ve vyšším programovacím jazyce s použitím objektově orientovaného vývoje dle moderních standardů.
- **Přizpůsobitelnost:** Aplikace by měla brát ohled na individuální preference uživatelů a umožnit maximální přizpůsobení těmto potřebám.
- **Přístupnost:** Koncová aplikace by měla být v cenové kategorii, kterou si mohou dovolit i méně majetní klienti a neziskové kategorie případně zcela zdarma.
- **Použitelnost:** Aplikace by neměly vyžadovat ke svému fungování internetové připojení. Měly by být tedy použitelné ke svému účelu i v běžném prostředí – mimo školu.

Tyto priority vedly k návrhu a vývoji nového frameworku a z něj odvozených aplikací. Společným základem těchto aplikací je pak tzv. framework počítačová terapie (dále jen i-CT

Framework), na kterém staví i tato práce a je také detailněji popsán [29]. Jednotlivé aplikace, nadstavby frameworku, následně mohou být využity ve speciální pedagogické edukaci či terapiích, stálých aktivitách, které mohou být nabízeny pravidelně dle individuálního plánu.

Otevřenost tohoto přístupu pak cílové skupině v důsledku přináší hned několik možností využití zařízení s novými aplikacemi [26][28]:

- **Slouží jako vzdělávací pomůcka** – výuka předmětů, čtení, psaní.
- **Rozvoj rozumových schopností** – zvládání praktických potřeb denního života.
- **Kompenzační pomůcka** – stává se přidruženou částí osoby – např. alternativní a augmentativní komunikační aplikace.
- **Volnočasová aktivita** – slouží k uvolnění a odpočinku.
- **Propojení s jinými terapiemi** – např. použití pro muzikoterapii.

3.4.3 Framework počítačové terapie

V roce 2016 byl pod vedením školitele této práce v rámci diplomové práce vytvořen framework založený na principech počítačové terapie nazvaný i-CT Framework s využitím metodologie SW inženýrství MDE [29].

Účel

Účelem i-CT Frameworku bylo vytvoření a poskytování společného základu pro všechny jednotlivé výukové aplikace, navržené ke spolupráci s i-CT Frameworkem. Framework také obsahuje databázi asistentů, jejich klientů a zdravotních záznamů, správu oprávnění, restrikcí a jiné společné pro všechny výukové aplikace.

Správa koncových aplikací (nadstaveb)

Jednotlivé výukové aplikace (nadstavby) musí obsahovat způsob registrace aplikace v i-CT Frameworku. Provádí se za pomoci invokací i-CT Frameworku včetně předaných informací o nadstavbě potřebných k její jednoznačné identifikaci mezi ostatními aplikacemi v zařízení. Po registraci aplikace a povolení dané aplikace pro určené uživatele ji lze spustit. Dojde k invokaci nadstavby a znovu také k předání určitých potřebných informací o přihlášeném uživateli. Podrobnější informace o předaných parametrech se nachází v kapitole 5.1.

Správa uživatelů

Samotné výukové aplikace pracují pouze s informacemi o aktuálním přihlášeném klientovi či asistentovi, předanými do aplikace při spuštění nadstavby. Veškerá ostatní manipulace a organizace uživatelů aplikace náleží i-CT Frameworku. Ten rozlišuje tři úrovně uživatelů – administrátory, asistenty a klienty. Asistenti mohou spravovat informace o svých klientech a vytvářet nové klienty. Administrátoři mohou spravovat veškeré klienty a asistenty.

Přínos pro vyvíjenou aplikaci

Díky správě uživatelů (tzv. „*user manager*“) a většině informací o nich, není nutné implementovat mechaniky starající se o toto téma a aplikace je tak vždy připravena přímo pro daného uživatele.

Vedle správy uživatelů poskytuje i-CT Framework také správu dané aplikace (tzv. „*apps manager*“) přímo ze svého prostředí aniž by bylo nutno koncovou aplikaci spustit a nastavit zde tak potřebné restriktce, oprávnění, která jsou často společná pro řadu koncových aplikací. Odpadá tak nezbytnost v dané koncové aplikaci toto implementovat.

Podle návrhových principů počítačové terapie spolu také mohou v určitých případech sdílet jednotlivé aplikace své informace a tím tak znásobit jejich vlastní hodnotu.

Také díky *Cordova Kiosk* pluginu implementovaném v i-CT Frameworku (pro naplnění priority zabezpečení jako součást tzv. „*security manager*“) se není potřeba starat o chování tabletu po uzavření aplikace, aniž by mohlo dojít k nechtěnému vypnutí aplikace či přechodu do jiného prostředí v OS.

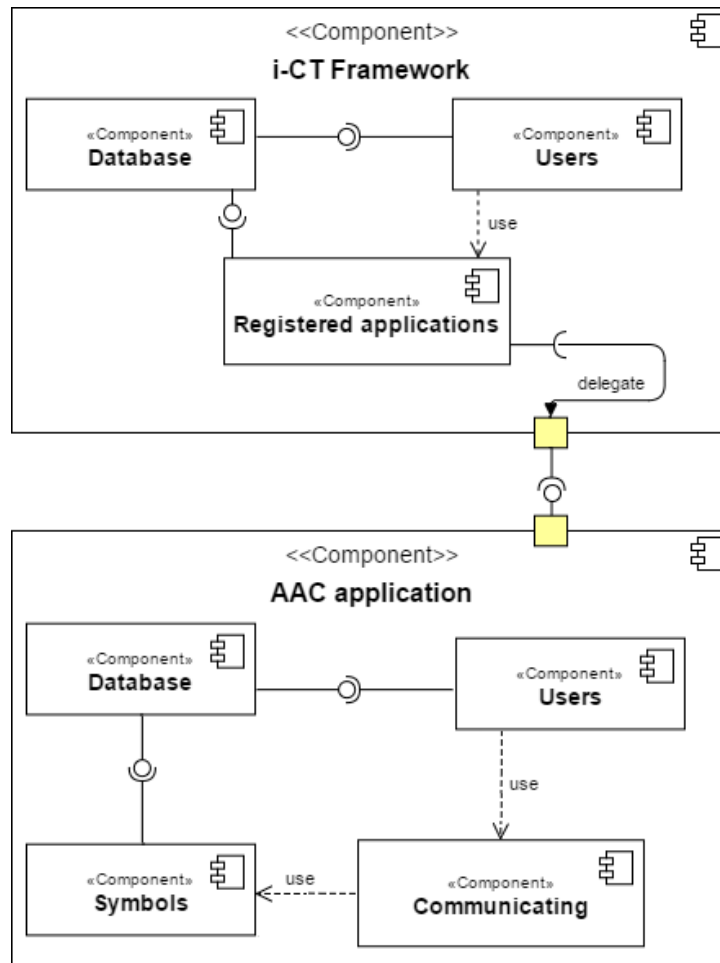
Kapitola 4

Návrh aplikace

V této kapitole se pojednává o problémech návrhu aplikace AAC podle požadavků na návrh aplikace vyjmenovaných v kapitole 2.4. Jak jsme již uvedli v kapitole analýzy požadavků 2.4, návrhové principy počítačové terapie představují prostředek pro naplnění požadavků ze strany vývoje i cílové skupiny. Budeme proto z těchto principů a nich postaveném frameworku (i-CT Framework) dále vycházet i v návrhu této koncové aplikace. Návrh bude tak kromě jiného odrážet i komunikační rozhraní pro spolupráci s i-CT Frameworkem a bude dále reflektovat z toho vyplývající mechanismy na straně této aplikace.

4.1 Návrh struktury aplikace

Jak je možné vidět na diagramu komponent na obrázku č. 4.1, vyvíjená aplikace je nadstavba i-CT Frameworku a je z něj spouštěna (důvody popsány v kapitole 3.4.3). Tato nadstavba obsahuje svou vlastní databázi, ve které jsou uloženy informace, které jsou využívány ke své hlavní úloze – poskytnout uživatelům alternativní způsob komunikace.



Obrázek 4.1: Diagram komponent aplikace a rámce i-CT Frameworku

4.2 Návrh uživatelského prostředí

Aplikaci vyvíjenou v rámci této bakalářské práce může používat více uživatelů s různými oprávněními. Proto bylo nutné tomu přizpůsobit návrh aplikace, aby se neoprávněný uživatel nedostal do části aplikace, kam by neměl mít přístup.

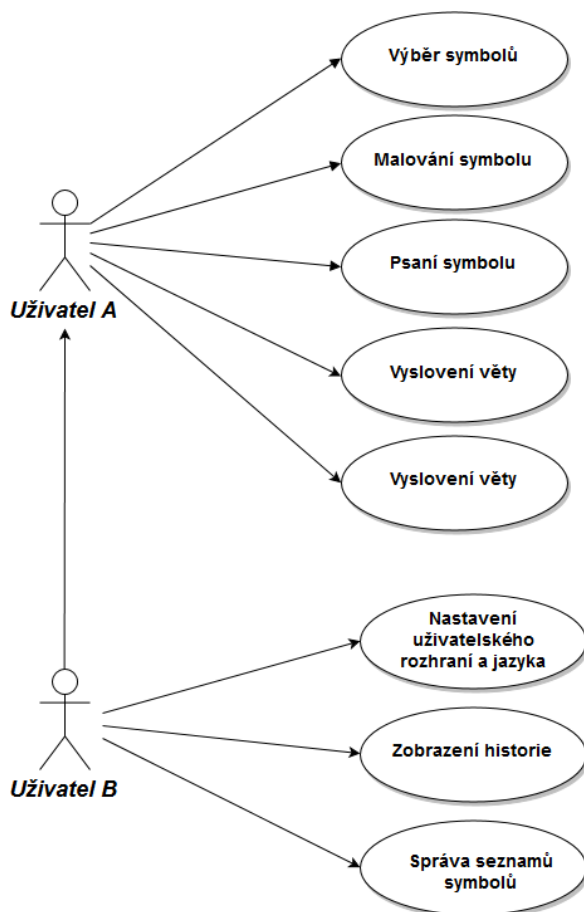
Souhrnně existuje několik různých způsobů jak spustit tuto AAC aplikaci z i-CT Frameworku:

1. Uživatel přihlášený do i-CT Frameworku je klient a může pouze spustit aplikaci v režimu klienta.
2. Uživatel přihlášený v i-CT Frameworku není klient a spustí aplikaci pod svým profilem v režimu klienta.
3. Uživatel přihlášený v i-CT Frameworku spustí aplikaci pod svým profilem v režimu správy.
4. Uživatel přihlášený v i-CT Frameworku vybere jiného z existujících uživatelů a spustí aplikaci pod jeho profilem v režimu klienta.

5. Uživatel přihlášený v i-CT Frameworku vybere jiného z existujících uživatelů a spustí aplikaci pod jeho profilem v režimu správy (lze použít např. jako tzv. asistovaný režim pro daného klienta jeho pedagogem).

V případě, že uživatel spustí aplikaci v režimu klienta (případy 1, 2 a 4), zobrazí se přednastavená pro uživatele, pod jehož profilem byla spuštěna, s možností běžného používání aplikace, bez dalších možností správy (nastavení), tedy v systému existuje pouze „uživatel A“ z diagramu případů užití na obrázku č. 4.2. Tento způsob se zvolí v případě, kdy chce aplikaci spustit pro vybraného klienta k běžnému použití a poté mu předat zařízení k používání.

Pokud naopak uživatel spustí aplikaci v režimu správy (případy 3 a 5), zobrazí se přednastavená pro uživatele, pod jehož profilem byla spuštěna s možností úpravy seznamů symbolů, zobrazení historie a úpravy uživatelského prostředí. V systému tento uživatel reprezentuje zároveň „uživatel A“ i „uživatel B“ z diagramu případů užití na obrázku č. 4.2. Tento způsob se zvolí v případě, kdy chce v aplikaci upravit nastavení vybraného klienta či spravovat jeho symboly. Zobrazí se tak tlačítko pro přechod do nastavení, které je při spuštění v režimu klienta skryto.



Obrázek 4.2: Diagram případů užití aplikace

4.2.1 Návrh obrazovek aplikace

Jelikož aplikace lze rozdělit do dvou částí – klientská a administrátorská, je možné i odlišit přístup v jejich návrhu. Obrazovky, přístupné pouze asistentům (např. speciálním pedagogům) či administrátorům mohly být navrženy více komplexně než obrazovky určené vlastním klientům (např. žákům speciální školy).

Hlavní obrazovka

Hlavní obrazovka je první a pro běžné uživatele – klienty s mentálním postižením i nejdůležitější obrazovka, na které stráví v rámci aplikace většinu času. Bylo tedy nutné ji vzhledem k budoucím uživatelům navrhnout intuitivní a přehlednou.

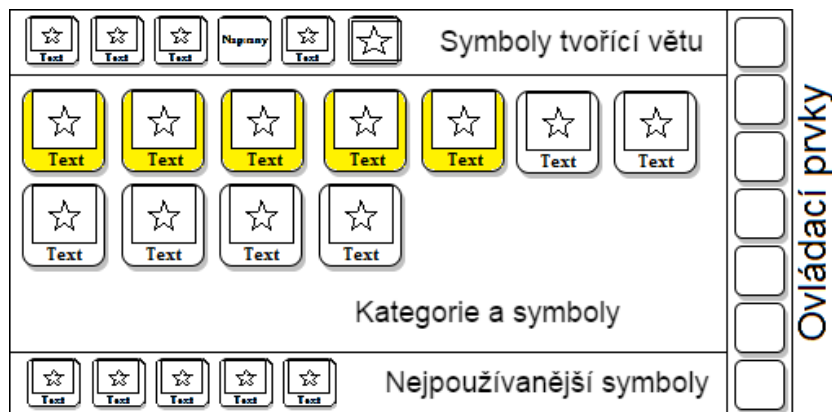
Na této obrazovce klienti vybírají symboly (v podobě piktogramů a jejich textového popisu), pomocí kterých poté tvoří celé věty. Každou vytvořenou větu poté mohou nechat nahlas vyslovit. Nemusí používat pouze předem připravené piktogramy, ale mohou také do věty přidat vlastní nakreslený piktogram případně celé slovo napsané na klávesnici.

Tato obrazovka může být zobrazena i v tréninkovém režimu, kdy uživatel neskládá větu pomocí symbolů, ale ve společnosti asistenta se učí s prostředím (důraz na získání kompetencí pro práci se piktogramy zařazených v daných kategoriích) pracovat. Během tohoto režimu se mu na obrazovce zobrazují pouze kategorie a symboly, které má k dispozici a tlačítka na navigaci mezi nimi. Při výběru kategorie či symbolu je mu také automaticky vysloven jeho popis. Ačkoliv je po spuštění aplikace zobrazen automaticky běžný režim, je možné přejít do tréninkového režimu po jeho povolení v nastavení klienta v aplikaci.

Z pohledu návrhových principů počítačové terapie a v kontextu návrhu aplikace pro alternativní komunikaci existují jisté prvky, které by měla základní obrazovka optimálně splňovat:

- **Jednoznačná akce** – kdykoliv uživatel chce provést danou akci, měla by mít stejný efekt při svém každém provedení.
- **Neměnné rozvržení rozhraní** – během celé doby používání aplikace by měly být veškeré ovládací prvky na stejném místě. Spolu s dříve zmíněným bodem by se takto mělo předejít zmatení uživatelů a vést k rychlejšímu zvyknutí si prozatím neznámé prostředí.
- **Přizpůsobení rozhraní** – ačkoliv základní rozvržení rozhraní je neměnné, může si uživatel přizpůsobit rozhraní ve dvou směrech:
 - **Změna velikosti piktogramů** – Asistent může nastavit pro svého klienta různou velikost symbolů. Může to pomoci jak zjednodušit práci pro klienty s oční vadou, tak přizpůsobit rozhraní pro různě velké obrazovky.
 - **Změna barevného pozadí obrazovky** – Asistent také může změnit na třech nezávislých místech hlavní obrazovky barvu rozhraní.
- **Responzivní efekt** – ať už uživatel provede jakoukoliv akci, daný efekt by měl být okamžitě viditelný.

Podle těchto požadavků je hlavní obrazovka aplikace navržena s obsahem následujících funkčních částí:

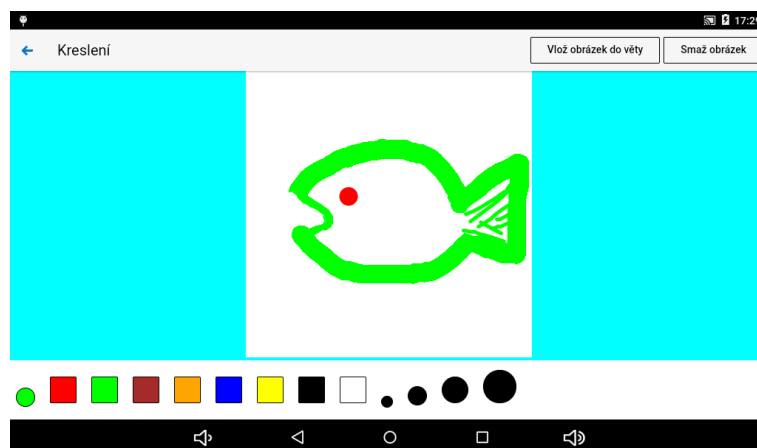


Obrázek 4.3: Návrh uživatelského rozhraní hlavní obrazovky v běžném režimu

- **Vybrané symboly tvořící větu** – symboly, vybrané uživatelem ze seznamu vlastních symbolů, případně nově vytvořené kreslením či napsáním.
- **Kategorie a symboly** – zobrazují uživatelské symboly a podkategorie v nynější vybrané složce, které mu jsou právě k dispozici. V jejich výčtu jsou kategorie označeny žlutou barvou a samotné symboly barvou bílou. Pokud uživatel zvolí symbol, přidá se k vybraným symbolům s právě vytvářenou větou. Pokud naopak zvolí kategorii, zobrazí se symboly a podkategorie nově vybrané kategorie.
- **Nejpoužívanější symboly** – zobrazují nejpoužívanější symboly daného uživatele. Berou se v potaz pouze věty uložené v jeho historii, tedy každý uživatel má jiné nejčastější symboly a navzájem se nijak neovlivňují.
- **Ovládací prvky** – obsahuje tlačítka na mazání symbolů z právě tvořené věty či smazání celé věty kompletně, navigaci mezi kategoriemi, tlačítka pro přechod k vytváření nového symbolu pomocí psaní či kreslení, tlačítka pro nastavení aplikace a tlačítka na vyslovení vytvořené věty. Na některých zařízeních s menší obrazovkou nejsou vidět najednou všechna tlačítka, zobrazí se tedy šipka, indikující, že je možné tahem prstu po dané části zobrazit i ostatní tlačítka.

Nakreslení symbolu

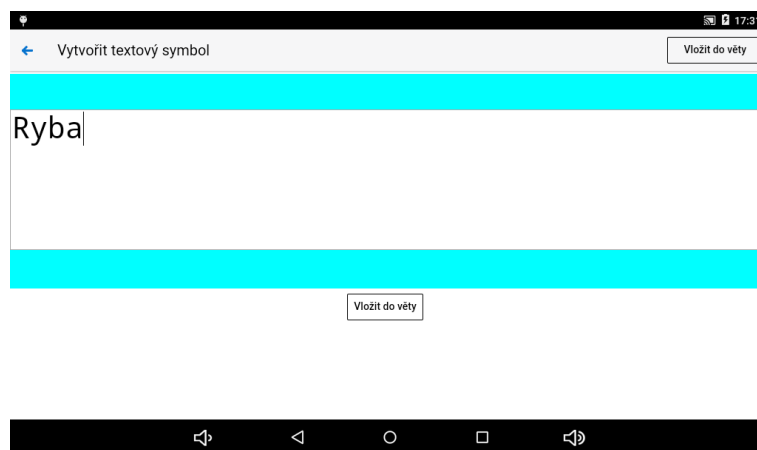
Pokud uživatel (i bez možnosti správy symbolů přes položku nastavení) nenalezne odpovídající symbol a nedokáže jej slovně popsat, může nakreslit vlastní. Využije k tomu právě tuto obrazovku (na obrázku č. 4.4), kde může vybrat z několika velikostí štětce a několika možných barev včetně bílé, která představuje efektivně také gumu. Vytvořený obrázek poté může uživatel vložit do právě tvořené věty jako zástupný symbol.



Obrázek 4.4: Obrazovka kreslení symbolu

Napsání symbolu

Kromě malování existuje také další způsob přidání symbolu s vlastním významem do věty – pomocí zadání textu z klávesnice (opět vhodné pro okamžité přidání symbolu bez nutnosti správy symbolů přes položku nastavení). Jelikož tuto aplikaci mohou používat i zrakově znevýhodnění, je nutné, aby uživatel mohl vidět zadávaný text v dostatečné velikosti. Obrazovku, řešící napsání napsání symbolů je možné vidět na obrázku č. 4.5.



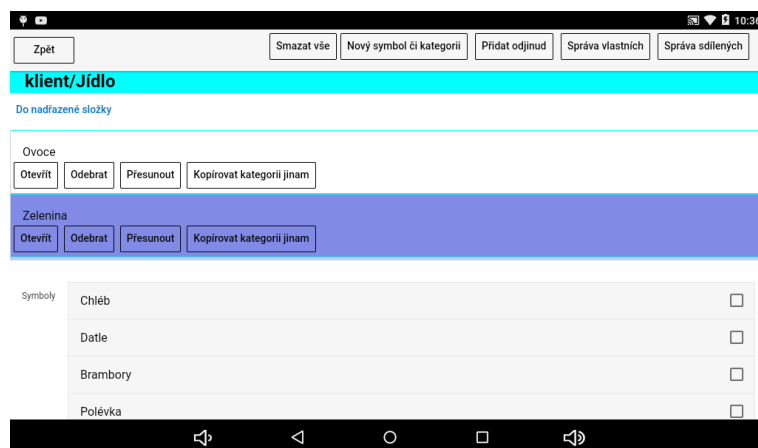
Obrázek 4.5: Obrazovka napsání symbolu

Nastavení

Pokud aplikaci AAC spustil uživatel, který je přihlášený v i-CT Frameworku jako asistent či jako administrátor, může zde spravovat tři různé prostory dat:

- Data klienta (pokud byla spuštěna aplikace přes profil klienta).
- Vlastní data (data asistenta či administrátora).
- Společná data – sdílená pro všechny uživatele.

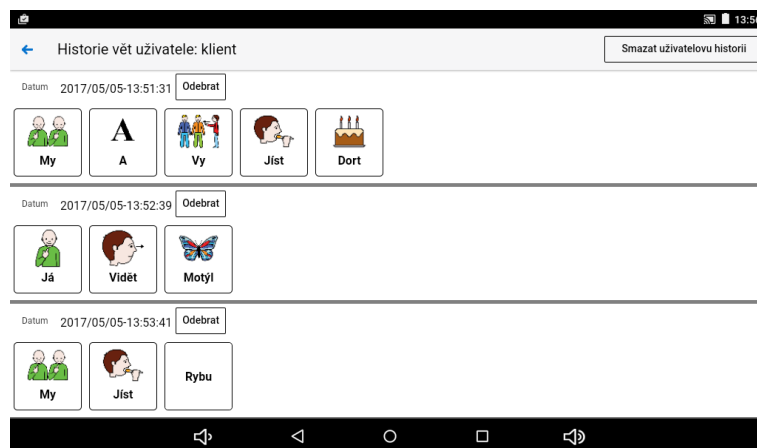
Pro všechny tři různé datové prostory bylo navrženo co nejvíce podobné a konzistentní rozhraní. Na hlavní liště obrazovky se v běžném režimu zobrazují tlačítka pro přechod do správy jiných prostorů dat, či pro přidání odjinud a pro vytvoření nového symbolu či kategorie. Nicméně v případě aktivní změny obsahu dat (kopírování, přesouvání) se rozhraní aplikace přizpůsobí danému kontextu a zobrazí pouze tlačítka pro vložení symbolů či kategorie do nyní zobrazené kategorie, případně tlačítko na zrušení změny, což pozitivně ovlivňuje přehlednost práce na obrazovce (na obrázku č. 4.6).



Obrázek 4.6: Uživatelské rozhraní správy symbolů

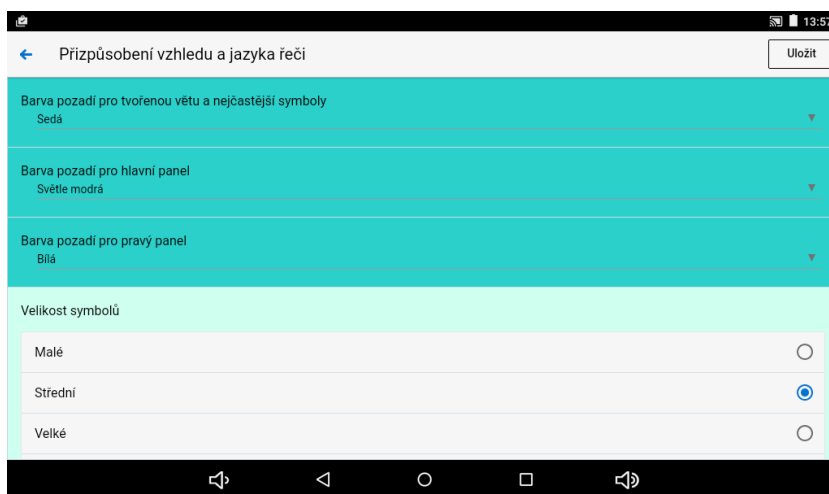
Vzhledem k rozsahu této problematiky je věnován prostor týkající se správy symbolů v samostatné kapitole 4.4.

Na obrazovce historie (obrázek č. 4.7) je možné vidět veškeré věty, které uživatel během svého používání aplikace sestavil a nechal vyslovit. Podle těchto záznamů o vyslovených větách se mění seznam nejčastěji používaných symbolů na hlavní obrazovce.



Obrázek 4.7: Obrazovka historie sestavených vět

Na obrazovce přizpůsobení aplikace (obrázek č. 4.8) je možné změnit barvu pozadí na hlavní obrazovce a velikost symbolů a kategorií. Také lze aktivovat tréninkový režim, který je v defaultním nastavení vypnutý. A rovněž lze zvolit jazyk pro hlasový výstup, stejně jako jeho rychlost.



Obrázek 4.8: Obrazovka přizpůsobení uživatelského rozhraní a jazyka hlasového výstupu

4.3 Návrh databáze

Tuto aplikaci může využívat několik asistentů (např. pedagogů), kteří pracují s více klienty (např. žáky) a pro každého z těchto uživatelů je třeba spravovat množství informací přímo v aplikaci, tedy je nutné je trvale ukládat do databáze na straně této aplikace.

4.3.1 Ukládané informace a omezení

Ačkoliv část informací se předává při spuštění aplikace v komunikačním URL schématu, s ohledem na vhodnost omezení rozsahu dat předaných tímto způsobem a uchovávaných na straně i-CT Frameworku, získá koncová aplikace povědomí pouze o základních údajích uživatele (jméno, příjmení, přihlašovací jméno (login), používaný jazyk a role uživatele, omezení v čase apod.) a zbylou část potřebných informací je tedy vhodné ukládat na straně koncové aplikace.

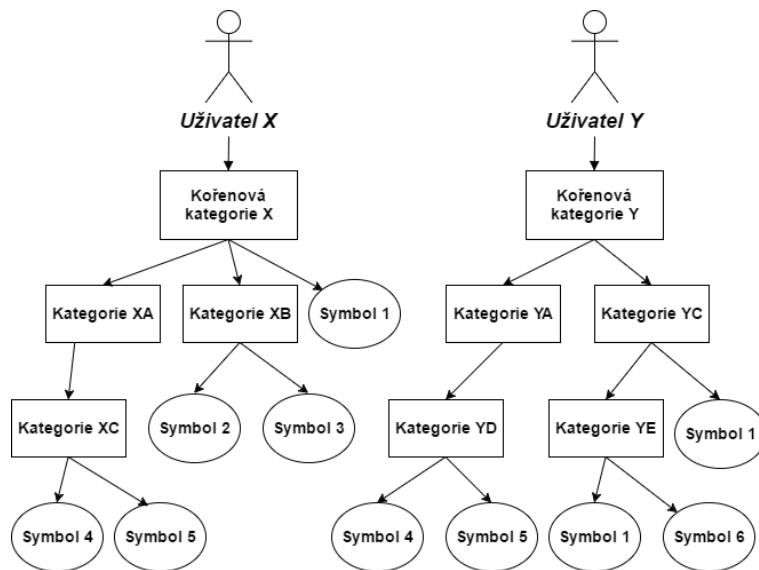
Je tedy třeba v databázi udržovat informace o nastavení aplikace pro daného uživatele. Týká se to například barvy pozadí, velikosti symbolů na hlavní obrazovce či používaného jazyka pro hlasový výstup aplikace.

Také je nutné ukládat informace o kategoriích a symbolech, které každý z uživatelů může vlastnit. Symboly (v podobě piktogramů) reprezentující slova by měl mít každý uživatel roztríděné do různých kategorií a samotná kategorie může obsahovat i další podkategorie, tedy vzniká určitá stromová struktura. Ačkoliv si mezi sebou mohou uživatelé kategorie sdílet, nesmí změny provedené jedním uživatelem ve své kategorii nijak ovlivnit kategorii stejného jména, kterou vlastní jiný uživatel.

S ohledem na smysluplnost pro potřeby alternativní komunikace byla také stanovena určitá následující omezení týkající se vlastnictví kategorií a symbolů:

- Uživatel nesmí mít kategorii s duplicitním názvem.
- Daný symbol může existovat ve více kategoriích v rámci jednoho uživatele, nicméně pouze jednou v rámci jedné kategorie.
- Kategorie druhé úrovně (např. kořenová složka (0. Úroveň) – > (Jídlo 1.) – > (Zelenina 2.)) již nesmí obsahovat další kategorii, pouze symboly, tedy kategorie mohou být zanořeny nejvýše do druhé úrovně.

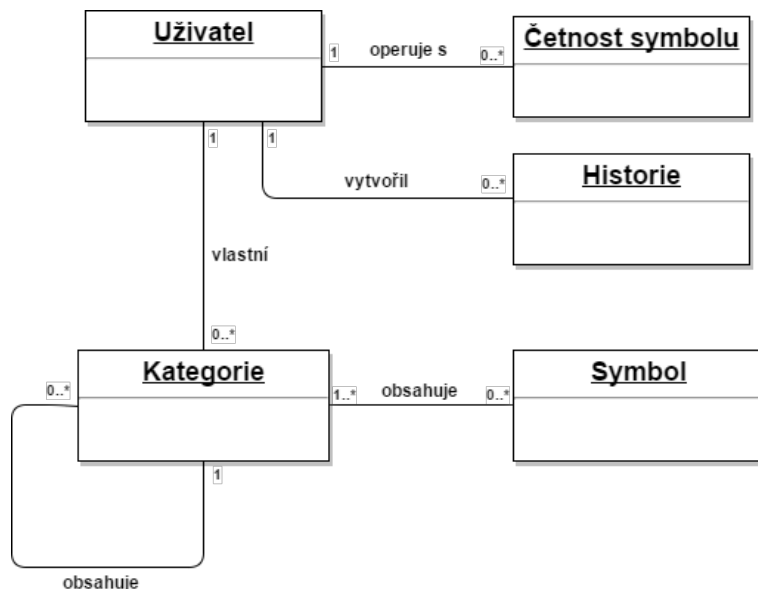
Bylo tedy navrženo následující schéma ukládání symbolů a kategorií, schematicky naznačené na obrázku č. 4.9



Obrázek 4.9: Návrh ukládání symbolů a kategorií pro uživatele

Každý uživatel zde má vlastní unikátně identifikovatelnou kořenovou kategorii. A tedy jelikož může vlastnit kategorii se stejným jménem více uživatelů zároveň, musí být daná kategorie také specificky označena, aby bylo jí možné odlišit mezi uživateli. Jednotlivé symboly nicméně mohou uživatelé mezi sebou sdílet volně, a může ho uživatel vlastnit i několikanásobně - uložený v několika kategoriích.

Samotný návrh databáze je na obrázku č. 4.10 a reflektuje výše zmíněné požadavky a omezení. ER diagram databáze je poté na obrázku č. 5.4 v kapitole 5.2.



Obrázek 4.10: Návrh databáze aplikace AAC

4.4 Správa symbolů

Jednoduchá, rychlá a komplexní správa symbolů je pro asistenty důležitá, a proto bylo tomuto cíli vyhověno při návrhu jejich správy.

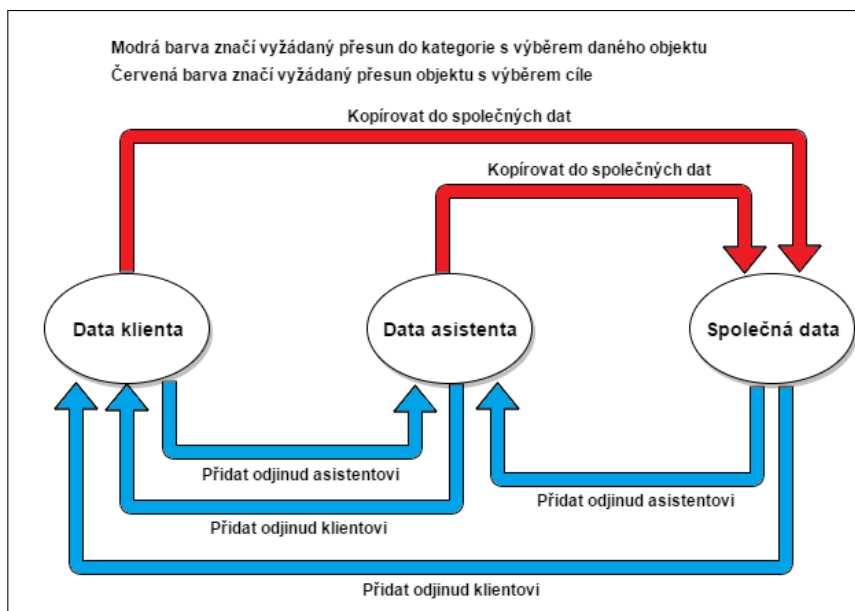
Jak již bylo zmíněno v kapitole 4.2.1, lze spravovat tři různé prostory dat:

- Data klienta (pokud byla spuštěna aplikace přes profil klienta).
- Vlastní data přihlášeného uživatele (data asistenta či administrátora).
- Společná data – sdílená pro všechny uživatele.

Ve všech třech prostorech dat lze provádět následující akce:

- Smazání symbolů či kategorie.
- Kopírování symbolu (nelze kopírovat kategorii v rámci jednoho prostoru dat).
- Přesun symbolů či kategorie.
- Vytvoření nové kategorie či symbolu.

Kromě samotného přesouvání či kopírování v rámci jednoho prostoru dat lze i kopírovat mezi nimi. Na obrázku č. 4.11 je možné vidět, jak lze dosáhnout daného cíle. Např. pro nakopírování kategorie od klienta asistentovi je nutné zvolit správu symbolů asistenta a kategorii, do které se má nakopírovat. Poté již uživatel zvolí možnost „Přidat odjinud“ a vybere z prostoru klienta zamýšlenou kopírovanou kategorii. Toto řešení bylo zvoleno z důvodu jednoznačnosti, kdy je vždy cílová kategorie klienta či asistenta pevně daná a pouze se volí kopírovaný objekt.



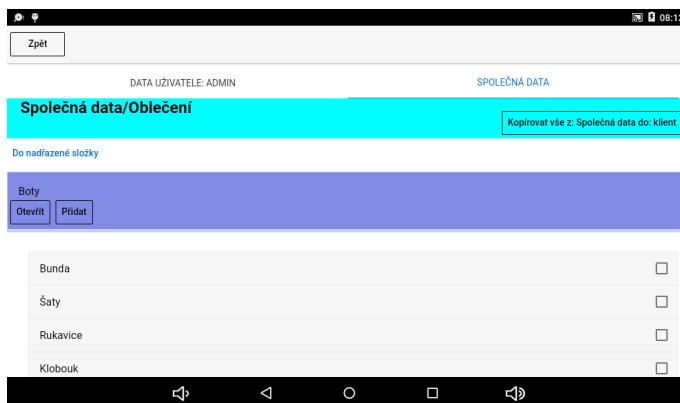
Obrázek 4.11: Diagram možností přesouvání mezi prostory dat

Jak je možné vidět na obrázku č. 4.12, zobrazují se na obrazovce správy symbolů vždy nejdříve podkategorie z nynější vybrané kategorie a až následně symboly, které nynější kategorie obsahuje. Na rozdíl od kategorií, kde jsou všechna tlačítka pro manipulaci s nimi viditelná vždy, pro práci se samotnými symboly je nutné některé z nich vybrat. Také je důležité na těchto obrazovkách skrývat či deaktivovat ovládací prvky v okamžicích, kdy je není možné použít. Jak je na obrázku č. 4.12 možné vidět, na rozdíl od obrázku č. 4.6 zde chybí několik ovládacích prvků, jelikož tento obrázek zachycuje obrazovku správy symbolů v okamžiku přesunu kategorie, tedy se zobrazují pouze prvky důležité pro uživatele v daný okamžik.



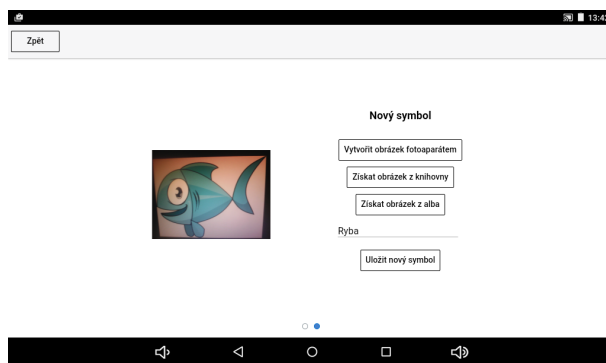
Obrázek 4.12: Obrazovka správy symbolů v okamžiku přesunu kategorie

Obrázek č. 4.13 poté zachycuje obrazovku správy symbolů v okamžiku přidávání symbolů odjinud, kdy si uživatel může vybrat mezi dvěma prostory dat (společná data nebo data uživatele v roli správce), a poté kategorii či symboly přidat do předem zvolené kategorie.



Obrázek 4.13: Obrazovka správy symbolů v okamžiku přidávání symbolů odjinud

Důležitou částí správy symbolů je také obrazovka vytvoření nového symbolu či kategorie, na obrázku č. 4.14 je možné vidět obrazovku, kde se vytváří nový symbol reprezentující slovo „Ryba“. Uživatel má možnost zvolit zdroj obrázku, tedy může získat obrázek pro reprezentaci slova „Ryba“ vyfotografováním obrázku v daném okamžiku, či získat obrázek již dříve vytvořený z alba či knihovny fotografií v zařízení.



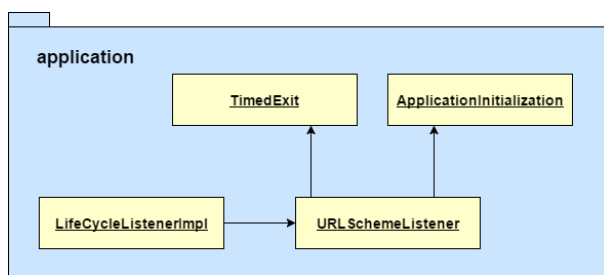
Obrázek 4.14: Obrazovka vytvoření nové kategorie a symbolu

4.5 Objektový návrh

Aplikace je navržena dle objektově orientovaného paradigmatu. Od této skutečnosti se proto odráží i modelové prostředky (UML) a jimi dosažené návrhové diagramy, které budou popsány v této kapitole. Při vytvoření nové aplikace založené na Mobile Application Frameworku se vytvoří automaticky dva podprojekty – *ApplicationController* a *ViewController*.

4.5.1 Implementační balíček *ApplicationController*

ApplicationController obsahuje pouze jediný balíček s názvem *application*. Tento balíček obsahuje především automaticky vytvořenou třídu *LifeCycleListenerImpl*. Tato třída obsahuje metody, které se provádí v různých fázích životního cyklu aplikace, například metoda *stop*, provedená při ukončení aplikace, metoda *deactivate* provedená při deaktivaci aplikace (uživatel začne pracovat s jinou aplikací) či metoda *start*, která se naopak provede právě při spuštění aplikace. V této metodě je volána metoda třídy *ApplicationInitialization*, jež se stará o případné vytvoření databáze a také je zde instanciována třída *URLSchemeListener*, která zpracovává přijaté URL schéma a v případě spuštění aplikace s nastaveným časovým limitem následně také vytváří instanci třídy *TimedExit*, která se stará o automatické ukončení aplikace po uplynutí časového limitu.



Obrázek 4.15: Zjednodušený konceptuální diagram tříd z části *ApplicationController*

4.5.2 Implementační balíčky *ViewController*

Vzhledem k velké orientaci na některé z komponent uživatelského rozhraní v průběhu vývoje aplikace došlo na vytvoření několika tříd specificky pro manipulaci s několika komponentami, např. třída *listViewSelectionHelper* je zaměřená na komponentu `amx:listView`, zobrazující ve správě symbolů jednotlivé kategorie.

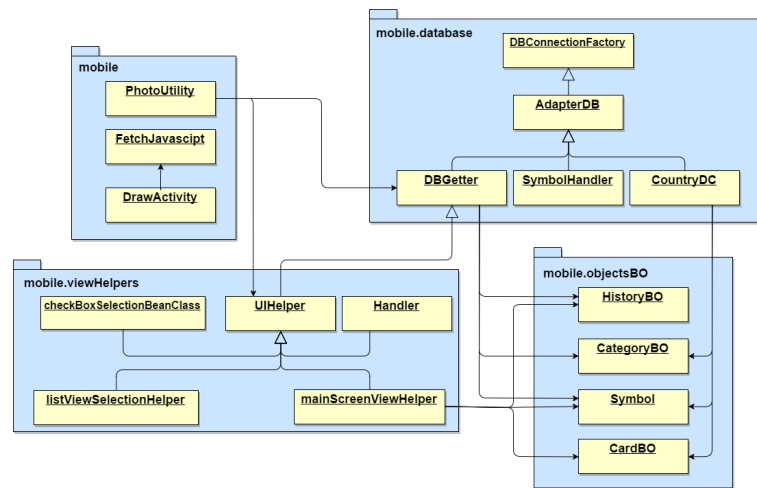
Diagram tříd na obrázku č. 4.16 zahrnuje i diagram balíčků, pouze pro ty balíčky, které obsahují alespoň jednu třídu.

Diagram tříd na obrázku č. 4.16 zahrnuje i diagram balíčků, pouze pro ty balíčky, které obsahují alespoň jednu třídu:

- **Balíček *mobile*** - Tento balíček se skládá z pomocných tříd pro některé aktivity (práce s kreslením obrázku a práce se získáním fotografie) a také obsahuje soubory popisující dostupné toky dat (*Data Controls*).
- **Balíček *mobile.database*** - V tomto balíčku se nachází podpůrné třídy pro práci s databází – pro navázání spojení či pro manipulaci s daty, které databáze obsahuje a pro je-

jich získání. Také obsahuje třídy s metodami, které přímo získávají data pro zobrazení v komponentách uživatelského rozhraní.

- **Balíček** *mobile.localization* - Tento balíček se skládá z lokalizačních souborů pro češtinu a angličtinu.
- **Balíček** *mobile.objectsBO* - V tomto balíčku jsou třídy, popisující objekty zobrazené uživateli v aplikaci. Například objekty *cardBO* jsou reprezentace pro symboly, se kterými pracuje uživatel na hlavní obrazovce.
- **Balíček** *mobile.pageDefs* - Do tohoto balíčku se ukládají soubory týkající se navázaného spojení toku dat (*Data Controls*) pro jednotlivé obrazovky.
- **Balíček** *mobile.viewHelpers* - Tento balíček obsahuje pomocné třídy aplikace týkající se reakce na práci s uživatelským rozhraním a jeho úpravou podle daného kontextu.



Obrázek 4.16: Zjednodušený konceptuální diagram tříd spolu s diagramem balíčků (pouze obsahující třídy).

4.6 Použitá licence aplikace

Po konzultaci s vedoucím práce a po zvážení požadavků principů počítačové terapie (v kapitole 3.4.2), byla zvolena licence GNU General Public License třetí verze (GPLv3) [6]. Byla zvolena zejména z důvodů podmínky veřejně dostupného zdrojového kódu a pro potřeby jeho dalších úprav či distribuce. Dalším důvodem je soulad s podmínkami licence Creative Commons (CC) [5], pod kterou byly uvolněny piktogramy ARASAAC [2] využití v aplikaci.

Kapitola 5

Implementace

Předmětem této kapitoly je samotná implementace vyvíjené aplikace AAC a popisuje řešené postupy a problémy během práci na této aplikaci. Jako vývojové prostředí bylo po domluvě s vedoucím práce zvoleno Oracle JDeveloper ve verzi 12.2.1.0.0, pro multiplatformní vývoj na mobilní platformy byl tedy vybrán Oracle Mobile Application Framework ve verzi 2.3.3, již dříve popsán v kapitole 3.2.3. Implementačním jazykem při vývoji je Java a částečně JavaScript a uživatelské prostředí je popsáno pomocí MAF AMX UI.

5.1 Životní cyklus aplikace a prezenční vrstvy

Při prvním spuštění aplikace se inicializuje databáze aplikace (více v kapitole 5.2.1). Při každém dalším spuštění aplikace (v případě neodinstalování aplikace) již databáze existuje a nemusí se znovu inicializovat.

Při spuštění aplikace AAC z i-CT Frameworku dojde k zavolání funkce start z třídy `LifeCycleListener`, kde se vytvoří instance třídy `URLSchemeListener`.

Při přijetí URL parametrů během invokace aplikace z i-CT Frameworku se pak vyvolá metoda `onMessage`. V té se získá dříve zmíněné URL schéma metodou `getPayload` a následně v metodě `extrakceParametruURL` se rozparsuje na URL parametry, jež obsahují základní informace o uživateli, jež spustili tuto aplikaci a určité údaje o způsobu spuštění. Příklad URL schéma pro tuto aplikaci je na ukázce kódu č. 5.1.

```
aac://?return=ictframework&userId=3&userLogin=klient&
userName=Ondrej&userSurname=Klient&userLevel=CLIENT&
userManagerId=2&userLanguage=en&loggedId=1&loggedLogin=admin&
loggedGesture=40bd001563085fc35165329ea1ff5c5ecbdbbeef&
loggedName=Ivana&loggedSurname=Spravcova&loggedLevel=ADMIN&
loggedLanguage=en&canEscapeApp=false&allowedTime=-1
```

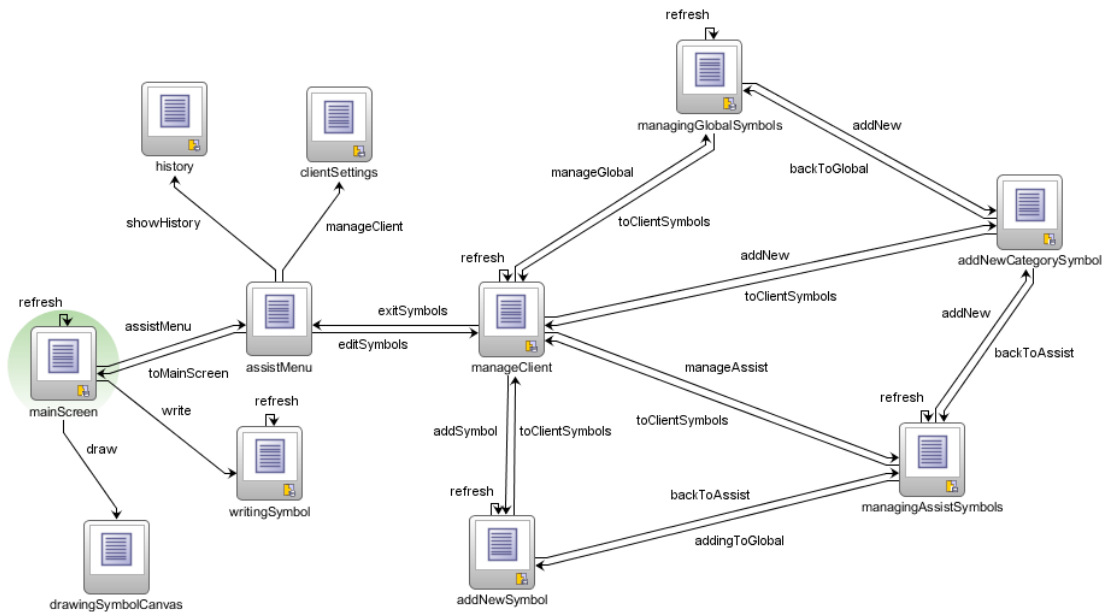
Ukázka kódu 5.1 - Příklad předaného URL schéma

Podle získaných informací je poté v metodách `checkIfUserExists` a `insertNewUserIntoDB` zjištěno zda daný uživatel či asistent již dříve aplikaci použil či ne. V případě, že se tak nestalo, se o nich do databáze aplikace přidá záznam (více v kapitole 5.2.2).

V případě, že uživatel používá aplikaci poprvé, je mu také dialogem na hlavní obrazovce nabídnuto nakopírování kategorií a symbolů z globálního prostoru do prostoru uživatele. Tento dialog je implementován komponentou `amx:popup`. Zde zobrazí uživateli dvě tlačítka a text s otázkou.

Obě tlačítka uzavřou dialog, nicméně kopírování celého globálního prostoru může trvat až několik desítek vteřin a zároveň uživatel nemá žádnou zpětnou vazbu. Proto se nyní po stisku tlačítka pro kopírování zobrazí vyskakovací okno se sdělením a až po jeho potvrzení se začne kopírovat.

Poté již může uživatel volně používat aplikaci. Na obrázku č. 5.1 je vidět posloupnost obrazovek, jež byla v rámci práce implementována a přechody (*Control Flows*) mezi nimi. Na obrázku je označena hlavní obrazovka (*Default activity*), jež je zobrazena automaticky jako první.

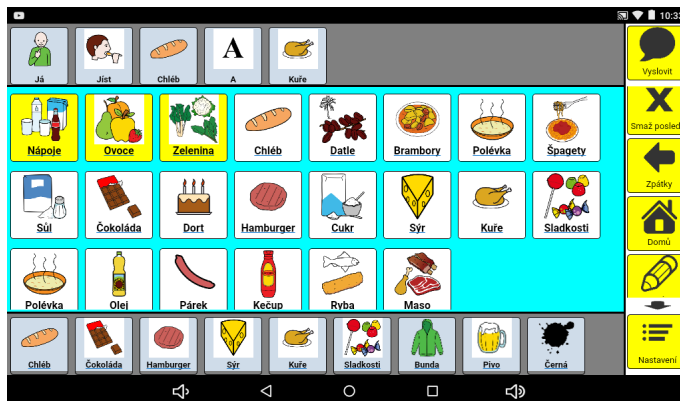


Obrázek 5.1: Diagram posloupnosti obrazovek (tzv. Task Flow Diagram) v implementované aplikaci.

Pokud je také aplikace spuštěna s časovým omezením, je po parsování URL parametrů inicializována třída `TimedExit`, kde se po uplynutí dané doby spustí metoda, jež ukončí aplikaci AAC a vrátí se zpět do i-CT Frameworku.

5.1.1 Implementace hlavní obrazovky

Hlavní obrazovku v běžném režimu, zobrazenou na obrázku č. 5.2, lze logicky rozdělit na tři proměnné prvky zobrazující uživateli symboly a jeden statický, který obsahuje ovládací tlačítka.



Obrázek 5.2: Hlavní obrazovka aplikace (v běžném režimu, spuštěná s přihlášeným asistentem)

Toto základní rozložení bylo v MAF AMX souborech vytvořeno pomocí komponenty `amx:tableLayout` a jejich zanořených komponent `amx:rowLayout` a `amx:cellFormat`. Pro možnost posouvání daných prvků v případě přesažení velikosti komponenty bylo využito `amx:panelGroup` s parametrem `overflow-x` nastaveným na hodnotu „`hidden`“ (případně `overflow-y` pokud se jedná o posun ve vertikálním směru).

Samotné zobrazení symbolů bylo realizováno komponentami `amx:rowLayout`, které obsahují seznam symbolů pro vykreslení. Dále je třeba zajistit obnovování obsahu jednotlivých komponent.

To je nutné provést v několika případech:

- U panelu s vybranými symboly do věty při přidání nového symbolu.
- U hlavního panelu při změně kategorie.
- U panelu s nejčastějšími symboly při vyslovení věty a tedy případné změně seznamu nejčastěji používaných symbolů.

Nejjednodušším způsobem lze obnovení vyvolat při interakci s uživatelským rozhraním akcí (*Action*), která v *Control Flows* ukazuje na obrazovku (*Activity*), ze které byla vyvolána a tím efektivně obnovuje celou obrazovku. Toto řešení je nicméně velice pomalé a při provedení se objeví na krátký okamžik grafické artefakty, které ruší práci.

Jiný způsob je vhodnější, a proto je implementovaný v aplikaci. Využívá třídu ADF Mobile frameworku `TreeBounding` a její metodu `refresh` pro aktualizaci dané zobrazující komponenty. Tímto se obnoví obsah pouze zvolené komponenty a ostatní zůstanou nedotčené. Například metoda `refreshTop` ze třídy `mainScreenViewHelper` aktualizuje seznam vybraných symbolů do věty.

Jedním z nejdůležitějších prvků na této obrazovce je také tlačítko pro vyslovení tvořené věty. Po jeho stisknutí se vysloví věta sestavená ze symbolů vybraných uživatelem.

Postupuje se následovně: Ve třídě `mainScreenViewHelper` v metodě `composeSentenceString` je získán seznam vybraných symbolů do věty a prochází se od prvního po poslední. V případě, že se jedná o běžný symbol, vloží se do výsledného řetězce název symbolu. Pokud je to symbol vytvořený uživatelem pomocí psaní, vloží se do výsledného řetězce uživatelem zadaná hodnota. A v případě, že je jde o nakreslený obrázek vloží se do něj řetězec „*nakreslený obrázek*“, případně „*own cartoon image*“, pokud je jazyk aplikace nastavený na angličtinu a následně je výsledný řetězec předán metodě `toSpeech`. Část algoritmu pro sestavení řetězce je v ukázce kódu 5.1.1 - Získání textu pro hlasový výstup. O samotném postupu převodu textu na řeč se pojednává v kapitole 5.3.

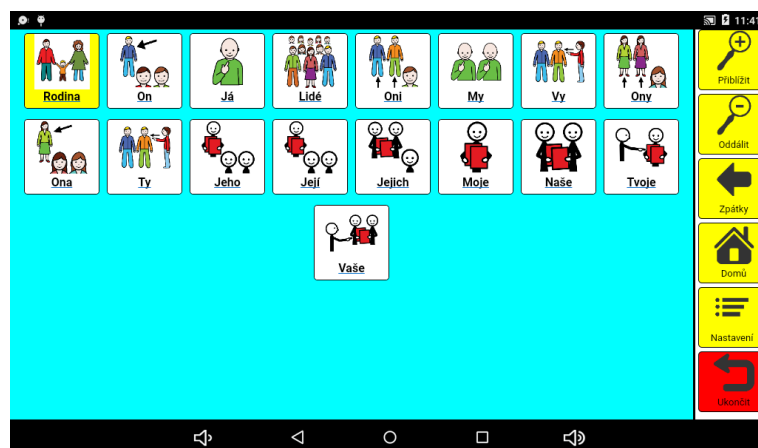
```

for (Integer i=0;i<basicIterator.getTotalRowCount();i++){
    basicIterator.setCurrentIndex(i);
    CardBO karta = (CardBO) basicIterator.getDataProvider();
    if (!vybranaKarta.getCardName().equals("USER_PICTURE"))
        retezecToSpeech+=vybranaKarta.getCardName() + " ";
    else {
        if (language.equals("english"))
            retezecToSpeech+="own_cartoon_image ";
        else
            retezecToSpeech+="nakresleny_obrazek ";
    }
}

```

Ukázka kódu 5.1.1 - Získání textu pro hlasový výstup

Také je možné pracovat na hlavní obrazovce v trénovacím režimu (na obrázku č. 5.3), kdy se nezobrazuje řádek pro vytvářenou větu ani řádek pro zobrazení nejčastěji používaných symbolů. Je tedy třeba pouze jedna komponenta `amx:iterator` pro zobrazení symbolů a kategorií. Při stisku symbolu či přechodu do nějaké kategorie je také okamžitě vysloven jeho název. Je zde i rozdíl v ovládacích prvcích. Některá tlačítka zde neexistují (pro kreslení a psaní symbolů) a naopak jiná jsou zde přidána – pro zvětšení či zmenšení velikosti symbolů bez nutnosti toto měnit v nastavení.



Obrázek 5.3: Trénovací režim aplikace

5.1.2 Implementace obrazovky historie

Na obrazovce historie bylo nutné použít `amx:iterator` zanořený v další komponentě stejného typu. První z nich slouží pro získání a zobrazení hlavičky věty (čas vytvoření věty a tlačítko pro smazání věty z historie) a druhý, zanořený iterátor, získává a vykresluje samotné symboly tvořící větu. Pro dosažení požadovaného chování iterátorů, bylo potřeba upravit soubor `historyPageDef.xml` s navázaným tokem dat pro danou obrazovku.

5.1.3 Implementace správy symbolů

Správa symbolů se dá rozdělit na dva různé typy obrazovek. První z nich zahrnuje práci se symboly a kategoriemi - jejich přesouvání, kopírování či mazání. Pro zobrazení kategorií byla využita

komponenta `amx:ListView`. Ta obsahuje seznam podkategorií, které obsahuje daná kategorie. Každou z nich v této komponentě reprezentuje prvek `amx:ListItem`. Symboly jsou zobrazeny komponentou `amx:selectManyCheckbox`. Díky využití této komponenty mohou uživatelé pracovat s více symboly najednou (např. mazat či přesouvat), nicméně to s sebou i nese nevýhody – nelze ji příliš upravovat a není možné v něm zobrazit další obsah, například obrázky.

Důležitou roli na této obrazovce hraje kontext, tedy zajištění, kdy zobrazovat určité ovládací prvky a kdy je naopak nechat skryté.

Ke komponentě zobrazující symboly je logicky přiřazených několik ovládacích prvků, které by se měly zobrazit pouze při výběru jednoho či více symbolů. Toho je dosaženo během provedení metody `onValueChange`, která je spuštěna při změně (selekcí či odznačení) této komponenty. Při provedení metody pak dojde k nastavení EL (*Expression Language*) výrazu rozhodujícího o viditelnosti daných ovládacích prvků, takže v případě žádného nevybraného symbolu budou skryty.

Pro zajištění určité bezpečnosti aplikace byl také vytvořen dialog, který se objeví při stisku tlačítka „Smazat vše“. Tento dialog je implementován komponentou `amx:popup`, stejně jako dialog na hlavní obrazovce při spuštění aplikace pod novým uživatelem.

Druhý typ obrazovky zobrazuje možnost vytvořit nový symbol či kategorii. Pomocí prvku `amx:filmStrip` se zobrazuje vždy jen možnost vytvořit jeden (symbol či kategorii) z nich a tahem po obrazovce zobrazit výběr druhého. Zatímco k vytvoření nového symbolu je nutné vybrat nějaký obrázek, k vytvoření kategorie nikoli.

Vytvořit obrázek lze několika způsoby: pomocí fotoaparátu, pomocí výběru z alba či z knihovny (poslední dvě možnosti se mohou na některých zařízeních shodovat). Díky MAF metodě `getPhoto` ze třídy `DeviceManager` lze získat URI schéma popisující umístění daného obrázku v dočasné složce aplikace.

```
storage/emulated/0/Android/data/com.company.AAC_085/cache/1.png
```

Ukázka kódu 5.1.2 - URI schéma umístění daného obrázku

Po potvrzení vytvoření daného symbolu či kategorie se vytvoří kopie daného obrázku ve složce trvalých souborů aplikace pomocí metody `saveIntoDB` ze třídy `PhotoUtility`. Cestu k dané složce získáme díky MAF metodě `getDirectoryPathRoot` z nativní třídy `AdfmfJavaUtilities`.

5.1.4 Implementace kreslení a psaní symbolu

Na obrazovce kreslení je nejdůležitější komponenta pro kreslení. Pro její implementaci bylo třeba použít komponentu `amx:verbatim`, do které lze vložit HTML elementy a skripty v jazyce JavaScript. Bylo zde využito skriptu *sketch.js* [15], díky kterému bylo implementováno jednoduché kreslicí plátno.

Uložení nakresleného obrázku funguje následujícím způsobem. Po stisku tlačítka na uložení obrázku je získán obrázek popsáný kódováním BASE64 a tato informace je uložena do EL výrazu. V případě, že uživatel vytvoří další obrázek a uloží ho ještě před návratem na hlavní obrazovku, uloží se daná informace do EL výrazu za dříve vytvořený. K vložení samotných obrázků do věty dojde až při návratu na hlavní obrazovku, kdy je zavolána metoda `escapeAndCreateSketch`, jež z kódování BASE64 vytvoří v dočasném úložišti aplikace obrázky ve formátu PNG, následně je přesune k ostatním trvalým souborům aplikace a vloží do věty nový symbol reprezentující tento obrázek. Algoritmus tohoto převodu je možné vidět na ukázce kódu č. 5.1.4.

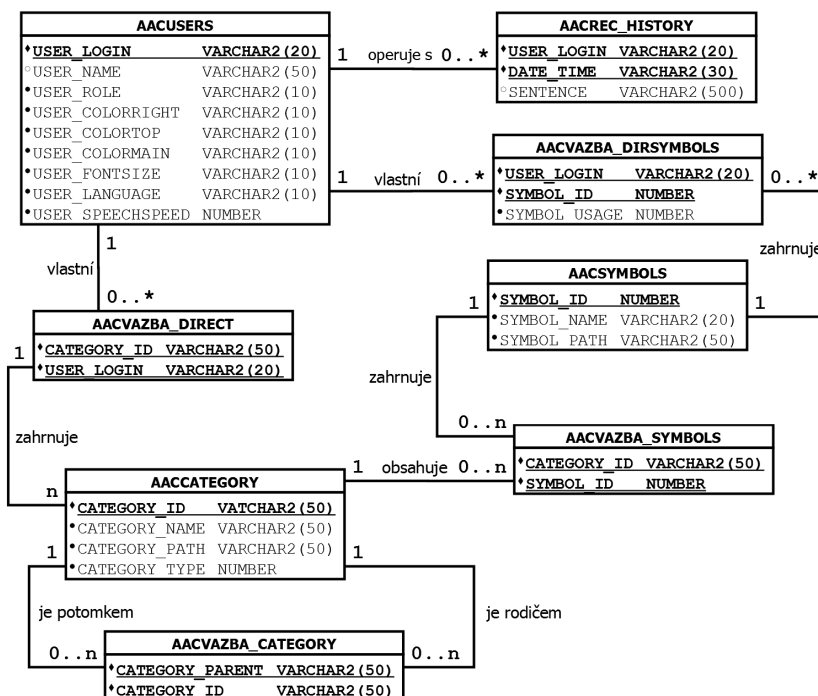
```
if ( pictureBase64.isEmpty() )
    continue ;
byte [] decoded = Base64.getDecoder().decode( pictureBase64 );
try ( OutputStream stream=new FileOutputStream( dir+"/01.png" ) ) {
    stream.write( decoded );
} catch ( IOException e ) {
    e.printStackTrace();
}
String symbolPath=newSymbol.saveIntoDB( dir+"/01.png" , "NOT" );
```

Ukázka kódu 5.1.4 - Převod obrázku do do formátu PNG

Na obrazovce psaní symbolů může vytvořit uživatel vlastní symbol do věty pomocí napsání, zde je nejdůležitější komponenta pro zadávání textu. Jelikož framework Oracle MAF obsahuje pouze jedinou komponentu pro zadávání textu - `amx:inputText`, u které ale není možné změnit velikost textu a jeho stávající velikost není dostačující, (jelikož jak je zmíněno v kapitole 4.2.1, uživatelé této aplikace mohou být i zrakově postižení), bylo i zde také nutné použít komponentu `amx:verbatim`. Do ní byl vložen HTML element `textarea` u kterého již je možné nastavit vyšší velikost textu. Díky parametru `placeholder` tohoto elementu také uživatel před napsáním prvního písmena vidí sdělení, kterého ho k tomu vybízí. Také je pomocí krátkého skriptu omezena délka zadávaného textu. Objevil se zde problém se získáváním textu z elementu `textarea`, a proto se každý vytvořený symbol nejdříve vloží do skryté a uživateli neviditelné komponenty `amx:inputText` a až při návratu na hlavní obrazovku se jeden či více uživatelem napsaných symbolů vloží do právě vytvářené věty.

5.2 Databáze

Zatímco většina údajů týkajících se uživatelů a jiné se ukládá na straně i-CT Frameworku, v této AAC aplikaci jsou ukládány údaje o nastavení uživatelského rozhraní a vlastnictví symbolů. Kvůli své dostupnosti na platformě Android i iOS byla vybrána databáze *SQLite*. Na obrázku č. 5.4 je ER diagram databáze podle požadavků uvedených v kapitole 4.3.



Obrázek 5.4: ER diagram databáze aplikace

5.2.1 Inicializace databáze

Databáze aplikace AAC je inicializována při svém prvním spuštění. Po zjištění zda databáze již není vytvořena (ukázka kódu 5.2.1) je v metodě `initializeDatabaseFromScript` ze třídy `ApplicationInitialization` databáze inicializována. Databáze je vytvořena do adresáře aplikace pod názvem `aacInfo.db`. SQL příkazy pro tento účel jsou obsaženy v souboru `\ApplicationController\src\initializedb.sql` a jsou prováděny jeden po druhém. Zahrnují nejen vytvoření databázových tabulek, ale i vložení informací o kategoriích a symbolech v globálním prostoru, tedy vždy je při prvním spuštění již dostupný obsah.

```

String docRoot = AdfmfJavaUtilities.getDirectoryPathRoot(1);
String dbName = docRoot + "/aacInfo.db";
File dbFile = new File(dbName);
if (dbFile.exists()) {
    return;
}
    
```

Ukázka kódu 5.2.1 - Zjištění existence nové databáze

5.2.2 Vložení záznamu o novém uživateli

V případě, že daný uživatel i-CT Frameworku ještě nepoužil aplikaci AAC, je nutné o něm vložit do databáze několik informací.

Zda záznam v databázi o uživateli existuje, se zjistí, pokud výsledek SQL dotazu na tabulku uživatelů aplikace (v ukázce kódu č. 5.2.2) bude neprázdný. V této tabulce je primárním klíčem login uživatelů předaný z i-CT Frameworku, jelikož i jejich login musí být unikátní.

```
SELECT * FROM AACUSERS WHERE USER_LOGIN=userLogin ;
```

Ukázka kódu 5.2.2 - Dotaz na existenci uživatele v databázi

Pro každého uživatele je automaticky při jeho prvním spuštění aplikace:

- Přidán záznam s informacemi o uživateli a defaultních hodnotách (velikost symbolů, barva pozadí rozhraní, jazyk řeči) do tabulky AACUSERS.
- Přidán záznam do databázové tabulky AACCATEGORY reprezentující kořenovou složku daného uživatele.

5.2.3 Přístup k databázi

Po inicializaci databáze se k ní již přistupuje pevně stanoveným přístupem. Je vytvořeno spojení s databází díky metodám ve třídě `DBConnectionFactory`. A pomocí metod `executeQuery` či `executeUpdate` ze třídy `adapterDB` lze provést nad databází vyhledávací dotaz či provést nějakou změnu. Ve třídě `DBGetter` potom jsou metody, které s databází pracují, například dotaz na ukázce kódu č. 5.2.3 vyjadřuje získání informací o nejčastěji používaných symbolech daného uživatele.

```
SELECT * FROM AACVAZBA_DIRSYMBOLS WHERE USER_LOGIN = userLogin  
ORDER BY SYMBOL_USAGE DESC LIMIT 9;
```

Ukázka kódu 5.2.3 - Dotaz pro získání nejčastěji používaných symbolů

5.3 Převod textu na řeč

Pro hlasovou syntetizaci byl využit *Cordova Text-to-Speech Plugin* [4], jelikož je dostupný pro obě podporované platformy (iOS od verze 7 a Android od verze 4.0.3).

Tento plugin nicméně sám neprovede převod textu na hlasový výstup, pouze poskytuje pomocnou vrstvu mezi aplikací a nainstalovaným softwarovým řešením převodu na dané platformě. Pro správnou funkci programu tedy musí být na zařízení nainstalován i odpovídající hlasový software podporovaný danou platformou.

Na platformě Android je od verze 4.0 přeinstalován modul Převod textu na řeč Google (Google TTS), které od června 2016 nabízí češtinu v přijatelné kvalitě, který je ale nutný dodatečně zdarma stáhnout. Alternativně je možné si z obchodu Google Play stáhnout i jiné moduly – například: Acapella TTS či SVOX Classic TTS a další, ale většina kvalitních českých hlasů je placených.

Na platformě iOS je možné využít nativní funkci Převod textu na řeč podporující češtinu a od verze 8 je i k dispozici vylepšená verze převodu textu na řeč - Alex.

Po stažení daného software či využití přeinstalovaného řešení je nutné dále stáhnout a nastavit požadovaný jazyk v nastavení v prostředí daného OS, který bude také následně nutné zvolit ještě volbou jazyka pro syntézu v nastavení této AAC aplikaci.

Při defaultním využití (tj. pouze při zavolání funkce s požadovaným textem bez zadání dalších parametrů) se tento plugin pokusí použít jako jazyk vždy angličtinu. Bylo proto nutné naimplementovat výběr z několika možných jazyků. To je možné zvolit v nastavení pozadí a jazyka pro syntézu v aplikaci. Je nutné, aby jazyk nastavený v software na převod textu do řeči (nastaveným v prostředí OS) byl shodný s jazykem nastaveným v této aplikaci. Na ukázce kódu č. 5.3 je vidět příklad volání funkce v JavaScriptu.

```
function toSpeech(desiredText , myLocale , speed) {
  document.addEventListener('deviceready', function () {
    TTS.speak( {
      text : desiredText , locale : myLocale , rate : speed
    },
    function () {
      ;
    },
    function (err) {
      alert(err);
    });
  },
  false );
}
```

Ukázka kódu 5.3 - Funkce pro spuštění převodu textu na řeč

Také bylo implementováno nastavení rychlosti výstupu řeči, je možné ji na obrazovce přizpůsobení vzhledu a jazyka řeči zvolit v rozmezí 50 % - 150 %. Informace o nastavení jazyka i o rychlosti výslovnosti hlasového výstupu se ukládají do databáze aplikace.

Kapitola 6

Testování a zhodnocení aplikace

Při vývoji každé aplikace patří mezi jeho nejdůležitější body testování. Během něj se programátor snaží nalézt většinu chyb v programu. Tato aplikace je vyvíjená na dvě odlišné platformy, bylo tedy nutné je otestovat na obou platformách.

Během každé iterace bylo opakováno testování správné funkcionality aplikace a produkovalo spustitelnou verzi vyvíjené aplikace. Díky využití této iterativní metody došlo k včasné eliminaci rizik, minimalizaci chyb a nedostatků, a celkově tak k lepší kvalitě výsledné aplikace.

Vedle tohoto průběžného testování funkcionality bylo navíc provedeno tzv. testování použitelnosti z pohledu oblasti "*Human-computer interaction*" [19], které ověřuje schopnost aplikace z pohledu praktického použití, a to i s koncovými uživateli. Bylo proto zvoleno několik metod pro ověření použitelnosti, které probíhalo v několika fázích. Více v samostatně v příloze B.

6.1 Zhodnocení

Díky počátečním fázím testování byly zjištěny určité nedostatky, které ovšem byly během následujících fází odstraněny či jejich interference minimalizovány. Během testování (zejména testování použitelnosti) se objevily i body a návrhy, jež byly z části již implementovány, případně navrhnuty jejich řešení. Z hlediska použitelnosti bylo možné ohodnotit aplikaci jako schopnou být využívána v cílovém prostředí.

Aplikace se také podařilo naplnit úspěšně prioritní body návrhových principů počítačové terapie, jako základní předpoklad pro dobrou použitelnost v praxi.

Aplikace byla porovnána s existujícími řešeními a z tohoto porovnání jasně vyplývá nejlepší hodnoty z pohledu použitelnosti i přístupnosti.

Pokud jde o vlastní testování použitelnost přímo v této aplikaci, většinu odhalených nedostatků se podařilo zdárně odstranit. K dořešení zůstali pouze některá minoritní rozšíření a nesprávné zobrazení přidaných obrázků na některých verzích systému iOS, pod systémem Android se vše podařilo zcela odladit. Tyto body zůstávají tak motivací pro další vývoj této aplikace, který lze s ohledem na otevřenost a volbu vhodné licence očekávat.

Z celkového hlediska míry použitelnosti a návrhových principů lze považovat aplikaci za úspěšně implementovanou.

Kapitola 7

Závěr

Mezi cíle této bakalářské práce patřilo seznámení se s problematikou alternativní a augmentativní komunikace a speciálních vzdělávacích potřeb u osob s mentálním postižením, a také s návrhovými principy počítačové terapie, a zejména s rámcem i-CT Framework, který zvyšuje použitelnost a přístupnost počítačové terapie pro cílové osoby. V průběhu této práce byly také prostudovány existující aplikace pro AAK na mobilních platformách a porovnány jejich vlastnosti s vyvíjenou aplikací, která řeší stejnou problematiku.

Aplikace, jejíž návrh a implementace jsou výsledkem této práce, realizuje komunikaci s i-CT Frameworkem a zobrazuje rozhraní aplikace přizpůsobené přihlášenému uživateli. Ten poté může tvořit věty pomocí symbolů, sdružených do kategorií. Také dovoluje tvořit vlastní symboly pomocí jejich kreslení či zadávání jako textových řetězců reálném čase komunikace. Hlasový výstup aplikace je pak implementován akustickou syntézou, což výrazně zjednodušuje práci oproti dříve vyvinuté aplikaci EasyTalk, jež sloužila jako vzor. V administrační části, jež je přístupná pouze pro speciální asistenty a pedagogy, pak je možné upravovat rozhraní aplikace, případně ji používat v jiných módech. Také se zde mohou spravovat symboly a kategorie a vytvářet nové pomocí fotografií či obrázků.

Pro vývoj této aplikace byl zvolen vývojový rámec Oracle Mobile Application Framework zejména z důvodu jeho dostupnosti, což přímo podporuje prioritní bod Otevřenost z návrhových principů počítačové terapie, a také jelikož podporuje vývoj na obě hlavní platformy iOS a Android, na které byla tato aplikace zaměřena. Bohužel tento rámec není ve vývojářské komunitě příliš rozšířený a je stále ve vývoji, a tedy prozatím nebylo jeho potenciálu plně využito.

Použitelnost této aplikace byla testována jak v přirozeném prostředí osob s mentálním postižením na Speciální zvláštní škole v Poděbradech, tak i v akademickém prostředí vedoucím práce, nezávislým expertem se zkušenostmi z tohoto oboru. Z výsledků tohoto testování vyplývá, že ačkoliv má aplikace stále potenciál k dalšímu vylepšení, je ji možné používat v cílovém prostředí osob s mentálním postižením jako prostředek alternativní a augmentativní komunikace.

Zdrojové kódy aplikace jsou dostupné pod vhodnou licencí jako open-source a aplikace tak může být dále rozvíjena. Díky multiplatformnímu řešení je také dostupná na široké množství zařízení (včetně budoucího nasazení na mobilní zařízení s Windows 10) a může se stát jedním ze základních kamenů aplikací pro i-CT Framework, které napomohou jeho rozšíření a nasazení v praxi.

Literatura

- [1] *AAK aplikace Klábosil. Speciální ZŠ Poděbrady* . [online]. [cit. 2017-03-11].
URL <http://www.spec-skola.cz/projekty-1/socialni-integrace-zakua-spostizenim/aakaplikaceklabosil>
- [2] *ARASAAC: Aragonese Portal of Augmentative and Alternative Communication* . [online]. [cit. 2017-05-10].
URL <http://www.arasaac.org>
- [3] *ART vs Dalvik - introducing the new Android runtime in KitKat. Infinum* . [online]. [cit. 2017-04-20].
URL <https://infinum.co/the-capsized-eight/art-vs-dalvik-introducing-the-new-android-runtime-in-kit-kat>
- [4] *Cordova Text-to-Speech Plugin. GitHub* . [online]. [cit. 2017-05-12].
URL <https://github.com/vilic/cordova-plugin-tts>
- [5] *Creative Commons Česká Republika* . [online]. [cit. 2017-05-10].
URL <http://www.creativecommons.cz>
- [6] *GNU General Public License. GNU Operating System* . [online]. [cit. 2017-05-10].
URL <https://www.gnu.org/licenses/gpl-3.0.en.html>
- [7] *How to Conduct a Cognitive Walkthrough. Interaction Design Foundation* . [online]. [cit. 2017-05-10].
URL <http://www.igi-global.com/dictionary/human-computer-interaction/13413>
- [8] *Jak vypadá Android uvnitř aneb co je ROM, kernel, bootloader a další? Android Market* . [online]. [cit. 2017-05-10].
URL <http://androidmarket.cz/android/jak-vypada-androiduvnitř-aneb-co-je-rom-kernel-bootloader-a-dalsi/>
- [9] *Ke vzdělávání autistů se v ČR začaly masově využívat tablety zpět. Zdravotnictví a medicína* . [online]. [cit. 2017-05-12].
URL <http://zdravi.euro.cz/clanek/priloha-pacientske-listy/ke-vzdelavani-autistu-se-v-cr-zacaly-masove-vyuzivat-tablety-473809?category=z-domova>
- [10] *Klábosil* . [Online; navštíveno 2017-03-19].
URL <http://www.klabosil.cz/>

- [11] *Metody a postupy AAK. SPC pro děti s vadami řeči*. [Online; navštíveno 2017-03-24].
URL <http://www.alternativnikomunikace.cz/stranka-metody-a-postupy-aak-7>
- [12] *Projekt I-SEN (otevřená komunita rodičů, pedagogů, terapeutů, IT odborníků)*. [online]. [cit. 2017-04-08].
URL <http://www.i-sen.cz>
- [13] *Proloque2Go. Assistive Ware*. [Online; navštíveno 2017-04-13].
URL <http://www.assistiveware.com/product/proloquo2go>
- [14] *Red Hat Česká republika*. [online]. [cit. 2017-05-11].
URL <https://www.redhat.com/en/global/czech-republic>
- [15] *Simple Canvas-based Drawing for jQuery. Sketch.js. Github.io*. [online]. [cit. 2017-05-02].
URL <http://intridge.github.io/sketch.js/>
- [16] *Smartphone OS Market Share, 2016 Q3. IDC: Analyze the Future*. [online]. [cit. 2017-05-11].
URL <http://www.idc.com/promo/smartphone-market-share/os>
- [17] *Speciální pedagog: Systémová podpora. Mají na to: Jak podpořit sociálně znevýhodněné děti na ZŠ*. [online]. [cit. 2017-04-26].
URL <http://www.majinato.cz/27-specialnipedagog.php>
- [18] *Vady řeči. Zdravě.cz*. [online]. [cit. 2017-05-02].
URL <https://vady-rci.zdrave.cz>
- [19] *What is Human-Computer Interaction. IGI Global*. [online]. [cit. 2017-05-10].
URL <http://www.igi-global.com/dictionary/human-computer-interaction/13413>
- [20] *What is Inspection Methods. IGI Global*. [online]. [cit. 2017-05-10].
URL <http://www.igi-global.com/dictionary/inspection-methods/14758>
- [21] Bernard, B.: *Úvod do architektury MVC. Zdroják: O tvorbě webových stránek a aplikací*. [online]. [cit. 2017-05-12].
URL <https://www.zdrojak.cz/clanky/uvod-doarchitektury-mvc/>
- [22] Burian, J.: *Co přesně znamená, že Xamarin je zdarma? TechNet Microsoft*. [online]. [cit. 2017-04-05].
URL <https://blogs.technet.microsoft.com/technetczsk/2016/04/13/co-presne-znamenazexamarin-je-zdarma/>
- [23] Chrenková, G.: *Problematika dětí s autismem v kombinaci s mentální retardací*. Diplomová práce, Univerzita Palackého v Olomouci, Pedagogická fakulta, Katedra psychologie a patopsychologie, Olomouc, 2010.
- [24] Chrástková, E. a. R. K.: *Komunikace u osob s mentálním postižením*. FIT VUT v Brně, Březen 2014, [online]. [cit. 2017-05-10].
URL <http://zdravi.euro.cz/clanek/sestra/komunikace-u-osob-s-mentalnim-postizenim-461759>

- [25] Škodová Eva: *Klinická logopedie I.*, ročník 616. Praha: Portál, první vydání, 2003, ISBN 80-7178-5466.
- [26] Fiala, J.; Kočí, R.: *Počítačová terapie jako koncept nové formy terapie pro osoby s mentálním postižením: teorie i praxe. Journal of Technology and Information Education*, 2014: s. 89–103.
- [27] 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*, 2015: s. 34–42.
- [28] Fiala, J.; Zendulka, J.: *Mentally challenged as design principles and models for their applications. Applied Computer Science* . Polish Association for Knowledge Promotion, 2016.
- [29] Kalina, J.: *Vývoj i-CT frameworku a jeho aplikace pro komunikaci typu ANO/NE*. Diplomová práce, Vysoké učení technické v Brně, Fakulta informačních technologií, Brno, 2016.
- [30] Lacko, L.: *Vývoj pre Android – Architektúra platformy Android / 4. časť. PC Revue* . [online]. [cit. 2017-04-30].
URL <https://www.pcrevue.sk/a/Architekturaplatformy-Android>
- [31] Mifsud, J.: *Usability Metrics: A Guide To Quantify The Usability Of Any System. UsabilityGeek* . [online]. [cit. 2017-05-13].
URL <http://usabilitygeek.com/usability-metrics-aguide-to-quantify-system-usability/>
- [32] Nielsen, J.: *Usability 101: Introduction to Usability. UsabilityGeek* . [online]. [cit. 2017-05-10].
URL www.nngroup.com/articles/usability-101-introduction-to-usability/
- [33] Švarcová Slabinová Iva: *Mentální retardace: vzdělávání, výchova, sociální péče* . Portál, 2011, ISBN 978-80-7367-889-0.
- [34] Thorová, K.: *Komplexní diagnostika autismu. Autismus.cz: Portál o poruchách autistického spektra* . [online]. [cit. 2017-03-29].
URL <http://www.autismus.cz/diagnostika/komplexni-diagnostika-autismu-4.html>
- [35] Valenta, M.: *Přehled speciální pedagogiky: rámcové kompendium oboru*. Praha: Portál, 2014, ISBN 978-80-262-0602-6.
- [36] Vejtasa, O.: *Aplikace pro alternativní a augmentativní komunikaci pro osoby s mentálním postižením*. Diplomová práce, Vysoké učení technické v Brně, Fakulta informačních technologií, Ústav inteligentních systémů, Brno, 2014.
- [37] Václavík, J.: *Vyvíjíme hybridní aplikace v Ionicu: Úvod a instalace. Zdroják* . [online]. [cit. 2017-03-15].
URL <https://www.zdrojak.cz/clanky/vyvijime-hybridni-aplikace-vionicu/>

Příloha A

Obsah přiloženého CD

- `sources.zip` - zdrojové soubory implementované aplikace + `readme` soubor
- `xcajan01.pdf` - zdrojové soubory textu této práce
- `xcajan01.zip` - elektronická verze textu této práce (formát $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$)
- `manual.pdf` - uživatelský manuál pro práci s aplikací

Příloha B

Testování použitelnosti aplikace a porovnání s existujícími řešeními

Cílem této přílohy je vyhodnocení úspěšnosti návrhu a implementace aplikace vyvíjené v rámci této práce. První část této přílohy se věnuje ověření míry použitelnosti („usability“, viz [31]) aplikace, a to i v případě použití v reálných podmínkách s cílem ověřit praktickou použitelnost při využití metod měření použitelnosti z podoblasti informatiky – interakce člověka s počítačem („human computer interaction“ viz [19]), druhá část zhodnocuje úspěšnost aplikace porovnáním s existujícími řešeními této problematiky.

B.1 Ověření použitelnosti

Cílem této části je popsat ověření použitelnosti vyvíjené aplikace jako systému v praxi. Ačkoliv je metrika použitelnosti podstatná i u softwarových řešení vytvářených ve všech ostatních odvětvích, v případě speciální pedagogiky je tato vlastnost zcela fundamentální. Proto je zhodnocení vhodnosti navržené a implementované aplikace součástí této práce. Vlastní ověření bylo pro potřeby této práce navrženo vedoucím práce ve shodě s obvyklými metodami pro ověření použitelnosti dle HCI u podobných systémů, jež se vyskytují i ve vědecké praxi.

V tomto případě ověření použitelnosti sestává souhrnně ze čtyř fází s využitím převážně inspekčních metod („usability inspection methods“ viz [20]). Před popisem jednotlivých fází je proveden nejdříve popis prostředí, ve kterém bylo toto ověřování probíhalo.

B.1.1 Popis prostředí ověřování

První dvě fáze ověřování byly provedeny v prostředí speciální základní školy a speciálně vzdělávacího centra v Poděbradech. Ve zdejší organizaci jsou elektronická zařízení (specificky tablety iPad) používány již delší dobu, pro účely výuky i více než sedm let. V roce 2011 byla dokonce pracovníky této organizace založena iniciativa iSEN, která má za cíl podporu využití a sdílení informací o využívání iPadů ve vzdělávání [12].

V této škole se vzdělávají žáci různého věku a postižení a jsou sdružováni dohromady spíše dle úrovně mentálního postižení než dle svého věku, a tedy v rámci jedné třídy se společně mohou vzdělávat i studenti s několikaletým rozdílem věku. V první fázi bylo ověřování použitelnosti prováděno zdejšími pedagogickými pracovníky, zejména autorkami iniciativy iSEN, které jsou také expertkami na interakci žáků z naší cílové skupiny s dotykovými za-

řízeními především na platformě iOS. Ve druhé fázi se do testování zapojili také zdejší žáci ve shodě s metodikou aplikovanou v rámci této fáze.

Ve třetí (předposlední) fázi byl vyžádán opětovný vstup ze strany vývoje na naší straně. Tato fáze byla provedena v podmínkách akademického prostředí. Poslední čtvrtá fáze ověřování pak probíhala opět v akademickém prostředí a ověřování bylo provedeno vedoucím práce jako nezávislým expertem na problematiku metriky použitelnosti a zpřístupnění SW (již od roku 2008 zahraniční stáž na Akademii věd v Singapuru a zde účast na HCI projektech, od roku 2010 zaměření na problematiku použitelnosti a zpřístupnění zejména u osob s mentálním postižením, realizace projektů, návrhy a implementace SW produktů pro tuto cílovou skupinu a v neposlední řadě také publikování ve významných mezinárodních, recenzovaných publikacích zaměřených na tuto problematiku).

B.1.2 Fáze ověřování použitelnosti

1. Fáze: kognitivní průchod („*cognitive walkthrough*“)

Kognitivní průchod [7] patří mezi inspekční metody a jeho cílem je odhalit možná rizika, nedostatky v použitelnosti dané aplikace.

V rámci této fáze byly pro potřeby tohoto ověření nejprve analyzovány strategické úlohy aplikace, které pokrývají základní případy použití této práce. Stanovené úlohy jsou následující:

- I. Nastavení prostředí této app (např. změna barev pozadí, velikosti piktogramů, rychlost řeči apod.).
- II. Přidání (vytvoření) nového piktogramu (symbolu) či kategorie (složka) danému uživateli.
- III. Kopírování piktogramů nebo celé složky mezi sadou žáka, sadou piktogramů na straně pedagoga nebo sdílenou sadou.
- IV. Výběr symbolů/piktogramů z hlavní plochy ve výukovém - trénovacím režimu (osvojování si významu piktogramů v daném uspořádání).
- V. Výběr symbolů/piktogramů z hlavní plochy nebo jejich zadání z menu (psaní, kreslení) v režim aplikace pro vlastní alternativní komunikaci (je aktivní větný řádek a ostatní menu).

Během této první fáze byly také při průběžně u každé úlohy pokládány standardní kritické otázky charakteristické pro tento druh metody [32]. V jednotlivých úlohách byly identifikovány následující možné nedostatky (výčet vždy souhrnně pro danou úlohu):

- I. Nastavení prostředí
 - a. Velikost kartiček - větší variabilita, i ty největší jsou příliš malé pro určitý typ klientů.
 - b. Problematický návrat z jednoho klienta do jiného – nutnost vše shodit a znovu zapnout a přihlásit se.
 - c. Nefunguje tlačítko – ukončit (návrat z administrace do prostředí klienta) bylo třeba aplikaci ukončit a znovu spustit
- II. Přidání (vytvoření) nového piktogramu
 - a. Přidání funguje, ale obrázek se nezobrazuje a text je vysloven po jednotlivých písmenech.

III. Kopírování piktogramů

- a. Nevidím možnost kopírovat od jednoho žáka do jiného, ale od správce nebo pedagoga ano.

IV. Výukový - trénovací režim

- a. Určitě by bylo vhodná možnost trvalého zapnutí tréninkového modulu, aby šlo přepínat mezi kategoriemi – po návratu „domů“ zmizí a je nutné se vrátit do nastavení.

V. Práce v režimu pro vlastní AAK

- a. Bylo by dobré přidat možnost vypnout řádek pro nejčastěji používané symboly (některé klienty může rušit – mnoho informací).
- b. Symbol tužky pro využití klávesnice je zavádějící, děti si myslí, že budou moct kreslit (funkce kreslení by byla dobrá).

2. Fáze: uživatelské testování („*usability testing, testing with real users*“)

Jako podpůrná metoda byla za první fázi zařazena metoda pro testování s reálnými uživateli, zde byli také zapojeni i žáci z výše uvedené školy. Metrika použitelnosti je v tomto případě (*usability*) hodnocena z pěti dílčích následujících měřítek dle Nielsen [32]: naučitelnost (*learnability*), efektivita (*efficiency*), zapamatovatelnost (*memorability*), chybovost (*errors*), naplnění (*satisfaction*).

Pro potřeby této práce lze souhrnně vyjádřit výstup z této fáze jako potvrzení nedostatků identifikovaných v první fázi. Dále se zde objevily jiné možné podněty ke zvýšení celkové použitelnosti, které nebylo dosažitelné identifikovat v první fázi zaměřené na základní úlohy. Zde je přehled nově identifikovaných bodů pro navýšení použitelnosti:

- A. Přidání podpory pro samostatný režim přímého spuštění této aplikace (bez propojení s i-CT Frameworkem) např. v případě kdy jde o uživatele jen s jedním nebo pár klienty.
- B. Možnost nastavení editace existujících symbolů/kategorií např. pro nastavení velkých tiskacích písmen u popisu obrázků vhodných pro globální čtení.
- C. Přidání dostupného obrázkového návodu pro rychlé zapojení pro uživatele, kteří nemusí mít s AAK aplikací zkušenosti.

3. Fáze: zpětná vazba („*re-design, re-implementation*“)

V rámci této fáze jsou reflektovány identifikované nedostatky z předešlých dvou fází. Během této fáze bylo vyvinuto úsilí reagovat na tyto nedostatky přepracováním jak na rovině návrhu, tak i na rovině implementace s tím spojeným testováním. Následně je uveden provedený způsob řešení přehledem v bodech dle kategorizace identifikovaných nedostatků. Tam kde nebylo dosažitelné zanesení změn do konečné implementace je alespoň uveden návrh postupu a odpovídající zdůvodnění (např. pokud by řešení značně přesahovalo požadavky na tuto bakalářskou práci nebo pokud by navržené řešení vedlo k výskytu jiných závažných nedostatků).

I. Nastavení prostředí

- a. **Problém:** Velikost kartiček - větší variabilita, i ty největší jsou příliš malé pro určitý typ klientů.
Řešení: Bylo implementováno více možností pro nastavení velikosti symbolů.

b. **Problém:** Problematický návrat z jednoho klienta do jiného – nutnost vše shodit a znovu zapnout a přihlásit se.

Řešení: Tato situace se vykytovala dříve, když nešlo aplikaci korektně opustit (viz také následný bod c), společně s funkcí pro EXIT byl navržen i přechod zpět do i-CT Frameworku, kde se po volbě jiného klienta je možno opět navrátit do prostředí této aplikace, kde bude vše už změněno a upraveno dle nastavení potřeb tohoto klienta.

c. **Problém:** Nefunguje tlačítko – ukončit (návrat z administrace do prostředí klienta), musela jsem aplikaci ukončit a znovu spustit.

Řešení: Problém vyřešen.

II. Přidání (vytvoření) nového piktogramu

a. **Problém:** Přidání funguje, ale obrázek se nezobrazuje a text je vysloven po jednotlivých písmenech.

Řešení: Problém snezobrazením obrázku se objevil na některých verzích siOS, vokamžiku dokončení práce se toto podařilo zatím zcela ošetřit pro verze sOS Android. Vlastní korektní vyslovení textu bylo zcela vyřešeno.

III. Kopírování piktogramů

a. **Problém:** Nevidím možnost kopírovat od jednoho žáka do jiného, ale od správce nebo pedagoga ano.

Řešení: Tato aplikace je navržena primárně jako nadstavba nad aplikací centrální správy - i-CT Frameworkem, to přináší své výhody neboť společný tzv. *class – user manager, apps manager a security manager* jsou řešeny jen na straně i-CT Frameworkem. Nadstavba proto nemůže mít v daném jednom okamžiku informaci o všech žácích neboť je také spouštěna v jinak požadovaném individuálním uzpůsobení pro daného žáka. Z toho důvodu je k dispozici sdílený prostor pro symbol, kategorie a také prostor asistenta (pedagoga), skrze kterýkoliv z těchto dvou prostorů dat, lze plnohodnotně zajistit přenesení piktogramů od jednoho žáka k jinému.

IV. Výukový - trénovací režim

a. **Problém:** Uvítala bych možnost trvalého zapnutí tréninkového modulu, aby šlo přepínat mezi kategoriemi – po návratu „domů“ zmizí a je nutné se vrátit do nastavení.

Řešení: Vyřešeno. Nyní aplikace může fungovat v tomto režimu trvale. Velikost si uživatel zvolí až na možnost přes celou plochu, ta pak zůstává aktivní.

V. Práce v režimu pro vlastní AAK

a. **Problém:** Bylo by dobré přidat možnost vypnout řádek pro nejčastěji používané symboly (některé klienty může rušit – mnoho informací).

Řešení: Implementováno. Nyní je možné používat aplikaci v režimu skrytého panelu nejčastějších symbolů.

b. **Problém:** Symbol tužky pro využití klávesnice je zavádějící, děti si myslí, že budou moct kreslit (funkce kreslení by byla dobrá).

Řešení: Vyřešeno. Ovládací tlačítko pro přechod na obrazovku napsání vlastního symbolu bylo změněno. Funkce nakreslení vlastního symbolu je implementována, ale na zařízeních s některými verzemi iOS se objevily obtíže s jejím správným zobrazením, v okamžiku dokončení práce se toto podařilo zatím zcela ošetřit pro verze s OS Android.

Návrhy pro navýšení použitelnosti:

- A. **Návrh:** Přidání podpory pro samostatný režim přímého spuštění této aplikace (bez nutnosti propojení s i-CT Frameworkem) např. v případě, kdy jde o uživatele jen s jedním nebo pár klienty.

Stanovisko: Během práce na programu ve fázi zpětné vazby bylo implementováno řešení, kdy je možné spustit aplikaci přímo bez nutnosti existence i-CT Frameworku. Lze takto aplikaci používat v režimu klienta bez jakýchkoli omezení.

- B. **Návrh:** Možnost nastavení editace existujících symbolů/kategorií např. pro nastavení velkých tiskacích písmen u popisu obrázků vhodných pro globální čtení.

Stanovisko: Je možné implementovat správu symbolů a kategorií s orientací na jejich úpravu a podrobnější správu, zatímco v této práci byla správa symbolů orientována zejména na co nejrychlejší a nejpohodlnější používání.

- C. **Návrh:** Přidání dostupného obrázkového návodu pro rychlé zapojení pro uživatele, kteří nemusí mít s AAK aplikací zkušenosti.

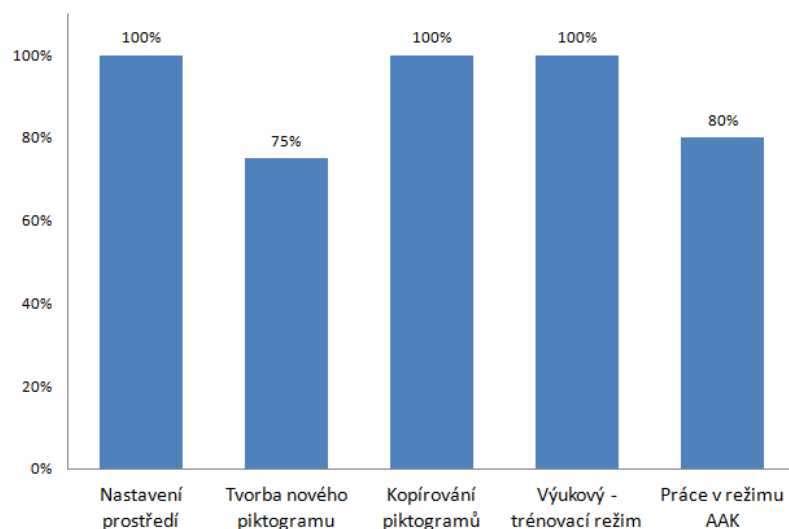
Stanovisko: V dalším rozvoji aplikace by jistě bylo vhodné vytvořit obrázkový návod, jež by pomohl uživatelům se seznámit s aplikacemi řešícími alternativní a augmentativní komunikaci, a zvláště s touto řešenou aplikací, aby ji byl uživatel schopný brzy používat.

4. Fáze: druhý kognitivní průchod („*cognitive walkthrough*“)

Zde u této konečné fáze pro ověření byla použita opět metoda pro kognitivní průchod. Namísto expertů z prostředí speciální pedagogiky hodnotících v rámci první fáze, zde byl zvolen hodnotitel, expert z akademického prostředí dle výše uvedeného.

Jak jsme již uvedli v popisu předchozí fáze, v rámci rozsahu této práce není dosažitelné reflektovat všechny body pro zlepšení použitelnosti zjištěné v rámci uživatelského testování s výjimkou bodu C (obrázkový návod) a bodu A, který je částečně naplnitelný (viz popis možnosti přímého spuštění této app i po propojení s i-CT Frameworkem).

Zároveň je zřejmé, že většina identifikovaných nedostatků se vyskytla právě na základě již vlastního testování kognitivním průchodem pro základní stanovené úlohy. I v průběhu této fáze byly při řešení jednotlivých úloh pokládány standardní kritické otázky, viz výše. Většina úloh byla nyní vyhodnocena jako splnitelná bez nedostatků, pro přehled následně uvádíme přehled míry naplnění použitelnosti u daných úloh.



Obrázek B.1: Graf celkové použitelnosti aplikace

Přestože nebylo plná použitelnosti zcela dosaženo, zbývající identifikované nedostatky představují minoritní podíl jak v rámci každé úlohy, tak v rámci celé aplikace. Podrobněji viz následná sekce zhodnocení výsledků ověření použitelnosti. Také jelikož je k dispozici zdrojový kód aplikace pod vhodnou licencí („*open source*“), je zde otevřena možnost navýšit použitelnost skrze další návrhy a implementace rozšíření prostřednictvím jiných vývojářů i z mezinárodních komunit.

B.1.3 Zhodnocení výsledků ověření použitelnosti

Ačkoliv byly výsledky ověřování použitelnosti a závěry z nich dostupné nedlouho před odevzdáním této práce, nakonec se podařilo větší část zjištěných nedostatků na straně použitelnosti odstranit. Celkem z pěti identifikovaných a řešitelných oblastí z kognitivního průchodu, byly 3 body zcela vyřešeny a dva body byly částečně naplněny. Dále byla pokryta i některá rozšíření zjištěná v uživatelském testování. Některé části této aplikace byly již v první fázi testování velmi pozitivně hodnoceny.

Patří mezi ně zejména:

- možnosti přizpůsobení uživatelského prostředí,
- vhodné a přehledné ikony použité z databáze ARASAAC,
- přehledná či vhodná grafika pro AAK,
- propojení s manažerem výukové třídy v podobě i-CT Frameworku, což bylo jedním z nejdůležitějších cílů této práce.

Souhrnně tak některé stávající nedostatky jsou motivací pro další vývoj a celkové zlepšení aplikace a vzhledem k otevřenosti a volné dostupnosti řešení lze toto do budoucna očekávat.

B.2 Porovnání se současnými řešeními

V této podkapitole je cílem porovnat vyvíjenou aplikaci **AAC** s jinými aplikacemi, které také řeší alternativní a augmentativní komunikaci. Byly vybrány aplikace **Klábosil**, využívaná v praxi v cílovém prostředí mentálně postižených klientů a aplikace **EasyTalk**, vytvořená na této fakultě v rámci diplomové práce od pana Ing. Ondřeje Vejtasy. Obě zmíněné aplikace byly již dříve popsány a funkčně porovnány v kapitole 2.3.2, nyní budou srovnány zmíněné aplikace s vytvořenou aplikací zejména podle priorit návrhových principů počítačové terapie (popsány v kapitole 3.4.2).

- **Daný účel aplikace** – Všechny aplikace splnily svůj účel, tedy poskytnout klientům s mentálním postižením způsob komunikace ve formě AAK.
- **Bezpečnost informací:** Vyvíjená aplikace využívá pro tento účel služeb i-CT Frameworku, jež umožňuje spravovat uživatele s různými oprávněními, což se přenáší i do nadstavby, tedy do aplikace AACC. Aplikace Klábosil umožňuje znepřístupnit obrazovku pro správu symbolů a aplikace EasyTalk pro danou obrazovku vyžaduje přihlášení pod specifickým uživatelem. Do této priority také spadá ochrana před opuštěním aplikace. Ve všech aplikacích na platformě iOS je toto vyřešeno Asistovaným režimem systému iOS. Aplikace EasyTalk na platformě Android nemá tuto část vyřešenou a aplikace AAC díky i-CT Frameworku je ochráněna pomocí Kiosk módu.
- **Otevřenost:** Aplikace Klábosil není vyvíjená v duchu open source a ani nebyla vyvíjená na otevřených technologiích. Obě ostatní aplikace (AAC a EasyTalk) mají volně dostupný zdrojový kód pod licencí která umožňuje další vývoj rozšíření a jsou postaveny na otevřených technologiích.
- **Multiplatformita:** Tento bod také splňují pouze aplikace EasyTalk a AAC. Obě jsou dostupné na platformách iOS a Android, zatímco aplikace Klábosil je dostupná pouze na dříve zmíněnou.
- **Rozšiřitelnost:** Vzhledem k nedostupnosti zdrojových souborů aplikace Klábosil nelze posoudit pro ni splnění tohoto bodu. Aplikace AAC a EasyTalk jsou ale obě implementovány ve vysokoúrovňovém jazyce a sdílí svou implementaci pro obě platformy. Tato aplikace je navíc rozšiřitelná i na mobilní platformy s OS Windows 10.
- **Přizpůsobitelnost:** Ačkoliv uživatelské rozhraní všech tří aplikací je možné přizpůsobit pro uživatele, aplikace AAC dovoluje přizpůsobit nejvíce a to nejen v rovině vlastního komunikačního obsahu ale i celkového prostředí.
- **Přístupnost:** Všechny tři aplikace jsou dostupné zdarma.
- **Použitelnost:** Všechny tři aplikace je možné používat bez nutnosti využití internetového připojení.

Porovnání splnění jednotlivých prioritních bodů z návrhových principů počítačové terapie a tedy i potřeb koncových uživatelů jednotlivých aplikací ukazuje tabulka č. B.1.

| | Klábosil | EasyTalk | AAC (produkt této práce) |
|------------------------------|------------------|---------------|---------------------------------------|
| Účel aplikace | splněno | splněno | splněno |
| Bezpečnost dat | heslo | přihlášení | i-CT Framework |
| Bezpečnost opuštění aplikace | Asist. režim iOS | pouze na iOS | i-CTFramework |
| Otevřenost | proprietární | otevřené | otevřené |
| Multiplatformita | pouze iOS | Android a iOS | Android a iOS |
| Rozšiřitelnost | nelze posoudit | plněno | Splněno a přesahuje možnosti EasyTalk |
| Přizpůsobitelnost | splněno | splněno | splněno |
| Přístupnost | zdarma | zdarma | zdarma |
| Použitelnost | offline | offline | offline |

Tabulka B.1: Tabulka 2 - Porovnání aplikací podle priorit návrhových principů počítačové terapie

B.2.1 Zhodnocení porovnání aplikace

Jak je patrné ze srovnávací tabulky č. B.1, aplikace AAC vyvíjená v rámci této bakalářské práce nejvíce splnila požadavky návrhových principů počítačové terapie, proto zde jsou také nejvíce naplněna očekávání z oblasti speciální pedagogie.