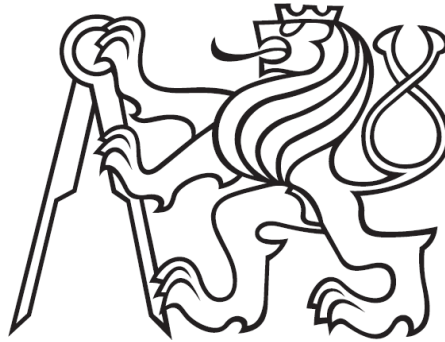


České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra kybernetiky



Bakalářská práce

Implementace audio-knihovny pro zrakově handicapované
uživatele

Lukáš Rubeš

Vedoucí práce: Ing. Daniel Novák, Ph.D.

Studijní program: Kybernetika a robotika

Obor: Robotika

15. května 2015

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Lukáš R u b e š
Studijní program: Kybernetika a robotika (bakalářský)
Obor: Robotika
Název tématu: Implementace audio-knihovny pro zrakově postižené uživatele

Pokyny pro vypracování:

1. Seznamte se s metodou user - center design pro návrh alternativních rozhraní.
2. Implementujte systém Bookshare jako mobilní aplikaci. Aplikace by měla umožňovat vyhledávání titulů podle jména a titulu, dále stažení elektronické knihy a její poslech.
3. Otestujte funkčnost aplikace minimálně na 5 uživateliích.

Seznam odborné literatury:

- [1] Suzor, Nicolas P., Harpur, Paul D., & Thampapillai, Dilan - Digital copyright and disability discrimination: From braille books to bookshare. (2008)
- [2] Vincent Gaudissart, Silvio Ferreira, Céline Thillou, Bernard Gosselin, SYPOLE - Mobile Reading Assistant for Blind People, SPECOM- 9th Conference Speech and Computer. (2004)

Vedoucí bakalářské práce: Ing. Daniel Novák, Ph.D.

Platnost zadání: do konce letního semestru 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 20. 1. 2015

Poděkování

Rád bych poděkoval své přítelkyni za psychickou podporu, své rodině za fyzickou podporu a svému vedoucímu práce Ing. Danielu Novákovi Ph.D. za praktickou podporu. Velké poděkování také patří sdružení SONS za konzultaci, vyplnění návrhového dotazníku, otestování mé aplikace a vyplnění hodnotícího dotazníku.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 15. 5. 2015

Lukáš Rubeš

Abstract

This work deals with the creation of an application for communication with on-line library Bookshare for blind and visually impaired users. The users have access to the database and are able to search for book according to various criteria. A book can be downloaded to the mobile device where the user is able to listen to it. To use this service one must create an account, provide proof of disability and pay a membership fee. Only people with physical disabilities can become members of the Bookshare community. The application is developed under the Android mobile platform.

Key words: Android; blind; on-line library; Bookshare.

Abstrakt

Tato práce se zabývá vytvořením aplikace pro komunikaci s on-line knihovnou Bookshare pro nevidomé a zrakově postižené uživatele. Uživatelé mají přístup do databáze knih a mohou vyhledávat díla dle různých kritérií. Libovolnou knihu si mohou stáhnout do svého mobilního zařízení a poslechnou si ji. K využívání této služby je nutné vytvořit si účet, doložit potvrzení o postižení a zaplatit členský poplatek. Členy komunity Bookshare se mohou stát pouze lidé s tělesným postižením. Aplikace je vyvíjena pod mobilní platformou Android.

Klíčová slova: Android; nevidomí; on-line knihovna; Bookshare.

Obsah

1	Úvod.....	13
1.1	Cíl práce.....	13
1.2	Existující aplikace.....	14
1.2.1	GoRead.....	14
1.2.2	Audible.....	15
1.2.3	LibriVox.....	16
2	Návrh řešení.....	17
2.1	Návrhový dotazník.....	18
3	Implementace.....	21
3.1	Třída BookshareWebservice.....	22
3.1.1	String convertStreamToString(InputStream inputStream).....	22
3.1.2	HttpResponse getHttpResponse(String wsPassword, String requestUri).....	22
3.1.3	InputStream getResponseStream(String wsPassword, String requestUri).....	23
3.2	Ověření uživatele.....	23
3.3	Vyhledávání.....	24
3.3.1	Vyhledávání dle zadaného textu.....	25
3.3.2	Výpis seznamů.....	26
3.4	Zobrazení výsledků hledání.....	27
3.5	Stažení knihy.....	29
3.6	Extrakce knihy.....	32
4	Testování.....	36
4.1	Hodnotící dotazník.....	36
5	Závěr.....	38
6	Použité zdroje.....	39
7	Seznam obrázků.....	40
8	Seznam kódů.....	41
9	Příloha A.....	42
10	Příloha B.....	45
11	Příloha C.....	48

12 Příloha D..... 49

1 Úvod

Na světě je více než 285 milionů lidí se zrakovým postižením¹), z toho přes 39 milionů je zcela nevidomých. Během života jsou všichni do jisté míry odkázáni na pomoc okolí a jejich každodenní aktivity mohou být dosti monotónní. Rozhodl jsem se zprostředkovat těmto uživatelům audio-knihovnu, odkud je možné stáhnout si do svého mobilního zařízení libovolnou knihu.

Umožnil jsem uživatelům využívat služeb databáze knih Bookshare²), která čítá přes 340 tisíc knih světových autorů. Česká díla tu zatím bohužel nejsou, ale to se snad do budoucna změní.

Roční poplatek činí \$50 a knihovnu mohou využívat pouze zrakově postižené osoby. Občané Spojených států amerických mají celou tuto službu zdarma (opět pouze zrakově postižení).

1.1 Cíl práce

Cílem mé práce je vytvoření mobilní aplikace pro Android, která zpřístupní uživatelům obrovskou databázi knih Bookshare. Aplikaci vytvářím jako součást mobilní aplikace Blindshell³). Ta využívá funkci text-to-speech, která převádí text na zvukový výstup. Nevytvářel jsem uživatelské rozhraní, ale pouze logiku vyhledávání a stahování. Aplikace Blindshell také obsahuje přehrávač textových souborů, proto jsem ani ten sám nevytvářel. Výsledkem je textový soubor, který obsahuje text dané knihy a který je strojově uživateli předčítán.

Pro textový soubor jsem se rozhodl z důvodu, že aplikace Blindshell již má integrovanou čtečku textových souborů. Tuto čtečku dále nebudu rozebírat, jelikož nejsem jejím autorem.

Aplikaci mi pomohou otestovat uživatelé ze společnosti SONS⁴ (Sjednocená organizace nevidomých a slabozrakých České republiky).

Vše implementuji v jazyce Java.

¹ <http://www.who.int/mediacentre/factsheets/fs282/en/>

² <https://www.bookshare.org/cms>

³ <http://www.blindshell.com/cs>

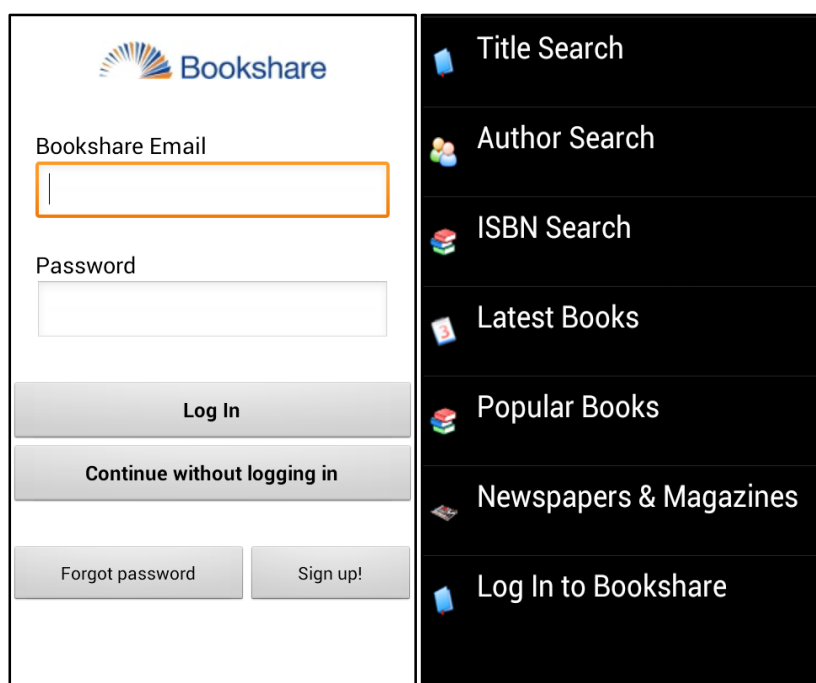
⁴ <http://www.sons.cz/>

1.2 Existující aplikace

Mobilních aplikací, které pracují s databázemi knih, je mnoho. Z pravidla každá aplikace přistupuje k jiné knihovně, ze které čerpá své knihy. Uživatel se zaregistruje, zaplatí členský příspěvek a může na cestách poslouchat stažená díla. Budu zde hovořit z mého pohledu o třech nejvýznamnějších aplikacích, které jsou volně stažitelné.

1.2.1 GoRead

Aplikaci GoRead [5] vyvinula společnost Google pod operačním systémem Android. Vytvoření této aplikace bylo na popud společnosti Benetech, vlastníci knihovnu Bookshare, aby rozšířila své pole působnosti z počítačů i na mobilní zařízení. Zanedlouho se na trhu objevily také aplikace pro zbylé operační systémy, IOS a Windows. Z vnějšku jsou všechny způsoby implementace stejné, záleží jen na systému mobilního telefonu.



Obrázek 1: Uživatelské prostředí aplikace GoRead

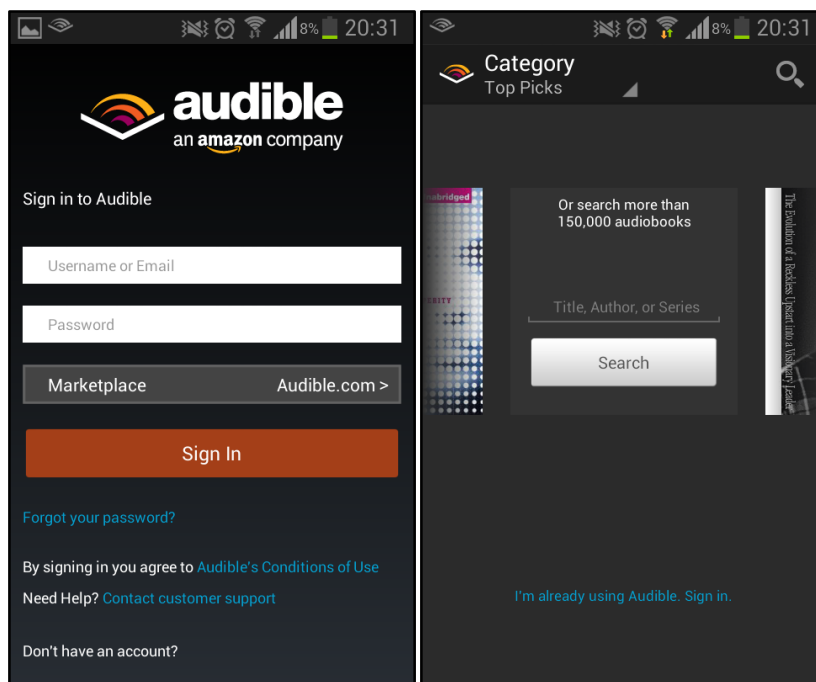
V popisu GoRead stojí, že je navržena pro uživatele s poruchou zraku. Rozumí se tím ale konečná fáze, a to poslech knihy. Aby se uživatel dostal do nabídky děl a nějakou knihu si stáhl, musí mu někdo asistovat. I kdyby měl uživatel zapnutý zvukový výstup, byla by manipulace s aplikací velmi obtížná. Pozitivní věcí na této aplikaci je, že při spuštění automaticky otevře naposledy čtenou knihu a stačí pak už jen spustit přehrávání.

Při testování této aplikace jsem zjistil, že stažené knížky předčítá strojově. To byl také jeden z důvodů, proč jsem se rozhodl pro uložení knih do textové podoby.

Stáhl jsem si knihu „An essay upon projects“ od spisovatele Daniel Defoe a zaměřil se na velikost zabrané paměti v mobilním telefonu. Pokud bych s knihou pracoval v aplikaci GoRead, její velikost je 745 472 bytů. Po převodu do textové podoby a odstranění všech původních souborů je velikost knihy 203 998 bytů. V tomto případě jsem ušetřil zhruba 72,6 % zabrané paměti.

1.2.2 Audible

Další aplikací, která uživatelům umožňuje přístup k mobilním online knihovnám, je aplikace Audible [6] od společnosti Amazon. Ta umožňuje členům (opět nutná registrace) procházet knihovnu čítající 180 tisíc knih. Je to tedy pravděpodobně druhá největší databáze na světě.



Obrázek 2: Uživatelské prostředí aplikace Audible

Uživatelské rozhraní je velmi nepřehledné, tlačítka jsou titěrná a není zde žádné základní menu. Audible necílí na zrakově postižené uživatele, a to se odráží v ceně. Měsíčně uživatel zaplatí \$14.95, ročně to tedy činí \$179.4.

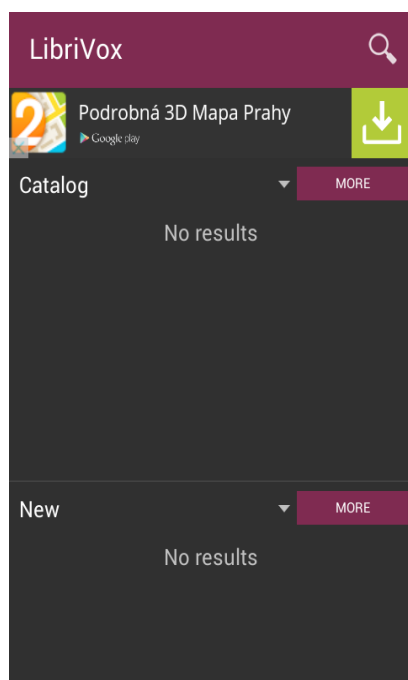
Otestovat Audible se mi bohužel nepodařilo. První věc, která se uživateli zobrazí při spuštění, je přihlašovací okno. Není možné přistupovat pomocí této aplikace ani k již staženým knihám,

aniž bych vlastnil placený účet na Amazonu. Nabízejí sice měsíční zkušební verzi, ale nevěřím, že bych poté mohl bezplatně skončit.

Jedinou věcí, kterou mohu bezplatně zkusit, je vyhledávání knih. Zkusil jsem autora „Capek“ a byl jsem velmi mile překvapen, jelikož mi to našlo pouze šest knih, a to od spisovatele Karla Čapka. Tento problém u databáze Bookshare podrobněji rozebírám v kapitole 3.4. Mohu jen konstatovat, že vyhledávání má server Amazon propracované mnohem lépe, než server Bookshare.

1.2.3 LibriVox

Poslední aplikací, o které budu hovořit, je LibriVox [7]. Tato kolekce čítá přes 15 000 knih a je úplně zdarma, namlouvají ji totiž dobrovolníci z celého světa. Z nějakého důvodu mají ale občané Spojených států amerických k dispozici více než 75 000 knih.













Obrázek 3: Uživatelské prostředí aplikace LibriVox

Při spuštění této aplikace mi to nabídlo přihlášení se přes účet od Googlu. Odmítl jsem a dostal se do nabídky, která je zobrazena na obrázku 3. Bohužel tedy nemohu nic dodat k ovládní, jelikož jsem neměl příležitost ho vyzkoušet. Není zde ani možnost se zpětně přihlásit.

Co se týče blikající reklamy, je to pravděpodobně jediný příjem společnosti a v tomto případě ji nebudu odsuzovat.

2 Návrh řešení

Při návrhu řešení jsem vycházel z dokumentace⁵⁾ webového portálu Bookshare. Ta obsahuje důležité informace o postupu při hledání a následném stažení knihy. Již na začátku jsem věděl, že nejtěžší část bude procházení dokumentací. Základní dotazování na server Bookshare je krásně popsáno, ale zbytek už není. Výsledkem předepsaného postupu stahování je kniha ve formátu DAISY⁶⁾. Tento formát je natolik nerozšířený, že se prakticky nedá dohledat, jak s ním zacházet. Pro představu, takto vypadá zmiňovaná kniha „An essay upon projects“ od spisovatele Daniel Defoe ve formátu DAISY:

Název položky	Typ	Velikost
 An_Essay_Upon_Projects.ncx	Soubor NCX	30 kB
 An_Essay_Upon_Projects.opf	Soubor OPF	2 kB
 An_Essay_Upon_Projects.xml	Dokument ve formátu XML	372 kB
 An_Essay_Upon_Projects00001.smil	Soubor SMIL	70 kB
 An_Essay_Upon_Projects00002.smil	Soubor SMIL	70 kB
 An_Essay_Upon_Projects00003.smil	Soubor SMIL	9 kB
 daisy.css	Šablona stylů CSS	13 kB
 daisyTransform.xsl	Šablona stylů ve formátu XSL	25 kB
 dtbook-2005-3.dtd	Soubor DTD	104 kB
 html.css	Šablona stylů CSS	14 kB

Obrázek 4: Kniha ve formátu DAISY

S těmito typy souborů jsem se předtím nikdy nenesetkal a nebylo jednoduché zjistit, co představují. Po dlouhém studování jsem nakonec kontaktoval osobu pracující ve společnosti Benetech, pana Nicholase. Ten mi poradil, abych pracoval se souborem formátu XML, který obsahuje text knihy. S touto informací jsem již byl schopen úspěšně dokončit implementaci celého procesu stažení knihy.

Dále jsem se snažil vytvořit přívětivé ovládání pro nevidomé uživatele. Vyhledávání knih v knihovně přes počítač je jednoduché. Můžu si u každé knihy rozkliknout detaily, podívat se na podobné knihy nebo například další knihy od daného autora. Některé části realizace tedy nebyly jednoznačné, a proto jsem vytvořil dotazník zaměřující se právě na ovládání a způsob zobrazení. Sestavený dotazník byl předložen uživatelům ze společnosti SONS. Je velmi důležité,

⁵ <http://developer.bookshare.org/docs>

⁶ http://en.wikipedia.org/wiki/DAISY_Digital_Talking_Book

aby výsledná realizace odpovídala představám cílových uživatelů a byli s tím spokojeni. Výsledky studie jsem zpracoval a inspiroval se požadavky uživatelů na koncept řešení.

2.1 Návrhový dotazník

Hlavní součástí mé práce je již zmíněný dotazník. Cílem dotazníku bylo zjistit, jak si uživatelé představují způsob ovládání. Účastnilo se jej celkem pět lidí, kteří jsou nevidomí nebo zrakově postižení a mají zkušenost s aplikací Blindshell. Tento počet byl zvolen z důvodu nedostatku času, neboť jsem potřeboval s výsledky velmi brzy pracovat, od výsledků se odvíjel celý návrh aplikace. Dotazník se všemi otázkami a odpověďmi je k nahlédnutí v příloze A.

V dotazníku jsem položil celkem pět otázek. Nejprve jsem se zajímal o jazykovou výbavu uživatelů, kteří používají aplikaci Blindshell, jejíž součástí je i má aplikace. Ptal jsem se na znalost cizího jazyka a na úroveň anglického jazyka. Knihy v databázi Bookshare jsou cizojazyčné, kdy většinu všech děl tvoří knihy anglické. Průzkum v této oblasti je u pěti lidí dosti nevyhovující, ale bral jsem to spíše jako informativní doplněk. Aplikace Blindshell je přeložena do několika světových jazyků a mohu předpokládat, že si budou chtít poslechnout knihu z databáze Bookshare i lidé z anglicky mluvících zemí. Zjistil jsem, že se část dotazovaných anglicky dorozumí, a má aplikace může být přínosem i pro ně.

Nejdůležitější otázky se ale týkaly možností vyhledávání a zobrazení. Ptal jsem se uživatelů na kritéria vyhledávání, dle kterých lze procházet katalog knih. Tuto nabídku jsem vybral podle možností, které server Bookshare podporuje a také podle svých vlastních zkušeností. Sám jsem častým návštěvníkem knihoven a vím, které položky využívám. Vybral jsem tedy šest kategorií, u kterých jsem si myslel, že by mohly uživatelům vyhovovat:

- Název knihy
- Jméno autora
- Kombinace názvu a autora
- Procházení novinek
- Procházení populárních knih
- Kategorie

K mému překvapení se všechny zmíněné položky respondentům zamlouvaly. Nejvíce hlasů dostaly položky novinky, populární a kategorie. Vysvětluji si to tím, že nevidomé uživatele zajímá, co se děje ve světě nového a chtějí být v obraze. Často dle svých slov nehledají konkrétní dílo, ale procházejí seznamy a dívají se, co by je zaujalo. Naopak když uživatel bude hledat konkrétní knížku, jako nejlepší způsob hledání mi přijde kombinace názvu a autora. Ušetří se tím spoustu času procházení ostatních výsledků. Tento případ si dovedu představit při doporučení jednoho čtenáře druhému. Když se někomu přečtená kniha zalíbí, to nejmenší je zmínit ji svým kamarádům.

Dalšími možnostmi je vyhledávání např. podle ISBN⁷⁾ knihy nebo podle ročníku ve škole. ISBN je mezinárodní standardní číslo knihy a používá se identifikaci výtisků. Uživatel by si musel

⁷ http://cs.wikipedia.org/wiki/International_Standard_Book_Number

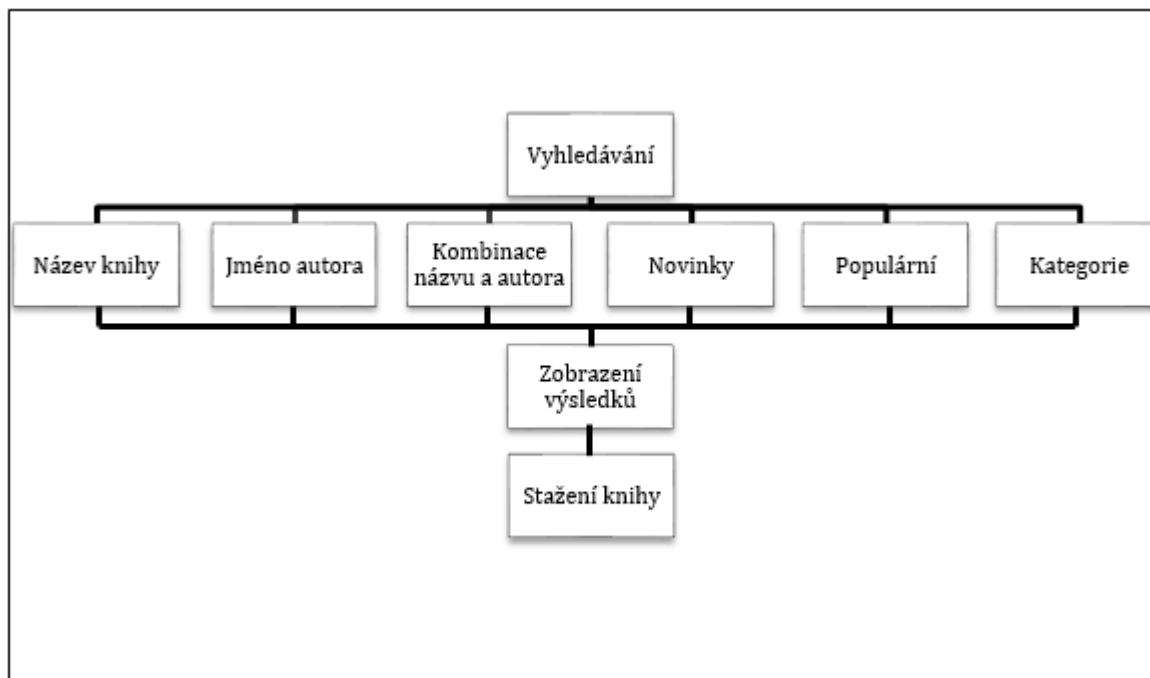
vyhledat požadovanou knihu, najít si její ISBN číslo a to poté zadat do aplikace. Toto mi přijde velmi nepraktické a zbytečné. ISBN číslo má deset až třináct číslic a je také velmi jednoduché udělat při zadávání chybu. Druhá zmíněná možnost je užitečná pro studenty, kteří si zde mohou dohledat učebnice a odbornou literaturu. Rozdělení ročníků je dle školního systému Spojených států amerických a na své si tu přijdou především studenti této země.

Nakonec jsem se zajímal o dalších způsobech zobrazení. Měl jsem myšlenku dialogového okna, které by se uživatelům zobrazilo při klepnutí na aktuální knihu. Opět jsem vybral několik možností, které by se zde mohly vyskytovat:

- Uložení knihy do mobilu
- Zobrazení anotace
- Zobrazení vydavatele
- Další vyhledávání podle zobrazených výsledků (např. vyhledávání nových knih podle autora dané knihy)
- Zobrazení roku vydání
- Zobrazení jazyka vydání

Po konzultaci výsledků s vedoucím práce jsem se rozhodl, že možnost dialogového okna do výpisu výsledků zatím nepřidám. Většina respondentů o tuto část nestojí, neboť tyto informace pro ně nejsou důležité. Mě osobně by zajímala možná tak anotace, ale také bych se dokázal bez této nabídky obejít. Je např. škoda, že server Bookshare nemá u svých knih hodnocení uživatelů. To si myslím, že by uživatele zajímalo ze všeho nejvíce. Kvalitní knihy se nemusí nutně zrovna nacházet mezi populárními knihami, které se denně mění. Pokud se tahle možnost v budoucnu změní, představoval bych si mít u výsledků dialog se záložkami hodnocení knihy a ohodnotit knihu. Zlepšila by se tak komunikace mezi čtenáři a mohlo by se pak i vyhledávat podle nejlépe hodnocených knih. Žebříčků nejlepších knih je samozřejmě velká spousta, nenašel jsem ale ani jeden, který by se zaměřoval na jednu konkrétní cílovou skupinu.

Konečný tvar vyhledávání jsem zvolil následovně:



Obrázek 5: Stromová struktura vyhledávání

Stažení knihy se provede klepnutím za zvolenou knihu. Celý proces je tak velmi směrový a uživatele nezatěžuju prodíráním se možnostmi, o které nestojí.

Aplikace je navržena tak, aby byla jednoduše rozšířitelná o další funkce. Např. přidání dalšího způsobu vyhledávání je záležitostí několika minut. To bylo také cílem mé práce, aby byla přehledná a co nejméně složitá. Pokud by na ní pracoval v budoucnu někdo jiný, bude muset mému kódu rozumět a umět ho přetvořit do jiné podoby. V průběhu času se mohou požadavky uživatelů měnit a je nutné vždy držet krok s těmito podněty. Také předpokládám, že server Bookshare na sobě bude dále pracovat a vymýšlet nové komponenty. Byla by škoda uživatelům zatajit např. nové užitečné funkce, které stará verze neobsahuje. Nenapadá mě žádný konkrétní případ, ale kdo ví, co bude za deset let. Některé nedostatky serveru Bookshare jsem tu již zmiňoval, a pokud by je opravili, bylo by nutné to změnit i v mé aplikaci.

Ačkoliv jsem nyní možnost zobrazení dalších informací pomocí dialogového okna zamítl, je možné, že ho jednou do mé aplikace přidám. Opět by se to odvíjelo od změn, které by server Bookshare provedl. Na aplikaci pracuji téměř jeden rok a je pravděpodobné, že jsem dosud neprozkoumal všechny možnosti této databáze. Teprve s uvedením na trh a s řádnou odezvou je možné aplikaci dotáhnout do dokonalosti. Hovořím především o spoustě chyb, které mi unikly, nebo jsem na ně v průběhu tvorby nenarazil.

3 Implementace

Základem komunikace s portálem Bookshare je třída BookshareWebservice⁸). Ta definuje metody, kterými posílám požadavky na server. Dále budu popisovat nejdůležitější metody mého kódu chronologicky, jak jdou za sebou. Rozsah celé aplikace je samozřejmě mnohem větší, ale nerad bych čtenáře nudil množstvím nepřehledného kódu. Nebudu také popisovat části, které jsou z hlediska programování dosti elementární. Jak jsem již na začátku zmiňoval, aplikaci vyvíjím jako součást aplikace Blindshell. Na začátku spolupráce jsem podepsal dohodu o mlčenlivosti, a proto také nebudu komentovat úplně všechny části, především co se zobrazení týče. Vše podstatné, z hlediska mé vlastní práce, je v této kapitole detailně popsáno.

Při spuštění aplikace kontroluji, zda je uživatel připojen k internetu. Celá komunikace se serverem Bookshare probíhá on-line a aktivní připojení je základní podmínkou.

```
WifiManager wm = (WifiManager) getSystemService(Context.WIFI_SERVICE);

WifiInfo info = wm.getConnectionInfo();

ConnectivityManager cm = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);

NetworkInfo netInfo = cm.getActiveNetworkInfo();

if (wm.isWifiEnabled() || netInfo.isAvailable()) {
    if ((info.getSSID() != null) || (netInfo != null && netInfo.isConnected())) {
        return class;
    } else {
        //not connected, inform user
        return null;
    }
} else {
    //not enabled, inform user
    return null;
}
```

Kód 1: Kontrola připojení

⁸ <https://github.com/benetech/Bookshare-API-Access-Library>

Musím kontrolovat jak připojení přes Wi-Fi, tak připojení přes mobilní síť. Doporučeno je samozřejmě připojovat se přes Wi-Fi, neboť odpovědi serveru mohou být obsáhlé. Nejprve v první podmínce zjišťuji, zda má uživatel na svém mobilním zařízení povolen přístup k internetu. Jednodušeji řečeno, zda má zapnuté Wi-Fi připojení nebo připojení přes mobilní síť. V druhé podmínce kontroluji, zda je mobilní zařízení k síti připojeno. Toto rozlišuji pro kvalitnější informaci o chybě. Pokud např. není dostatečný signál nebo se mobilní zařízení nenachází v dosahu signálu, informuji uživatele o skutečnosti, že se musí přesunout na jiné místo. Naopak, pokud je signál dostatečný, uživatel nerušeně spustí aplikaci.

3.1 Třída `BookshareWebservice`

Nejsem autorem této třídy (zprostředkovává ji server Bookshare), ale je velmi důležitá ve všech fázích komunikace se serverem. Rád bych tedy popsal některé metody, které používám ve svém kódu. Při registraci na serveru Bookshare, jakožto vývojář, mi bylo přiděleno heslo a odkaz na tuto třídu. Obsahuje mimo jiné i speciální kódování, bez kterého bych komunikaci nenavázal.

3.1.1 `String convertStreamToString(InputStream inputStream)`

Metoda převádí `InputStream` na `String`. Využívá se třídy `BufferedReader`, konkrétně metody `readLine()`, která prochází všechny řádky. Pomocí `StringBuilderu` se jednotlivé části poskládají do jednoho textového řetězce. Pro mě to je nejdůležitější metoda, protože překládá odpovědi ze serveru do čitelné formy a já s nimi mohu dále pracovat.

3.1.2 `HttpResponse getHttpResponse(String wsPassword, String requestUri)`

Metoda vrací `HttpResponse` pro požadované URI. `HttpResponse` si lze představit jako odpověď ze serveru, kterou si nejdříve vnitřně zpracuje, než ji zobrazí uživateli. URI je textový řetězec, jehož základem je internetová adresa s dalšími požadavky dotazovaného serveru. V další části této kapitoly je tvar URI rozebírán dopodrobna. Parametry metody jsou heslo uživatele a již zmiňovaný URI řetězec.

3.1.3 InputStream getResponseStream(String wsPassword, String requestUri)

Metoda vrací InputStream pro požadované URI. Vnitřně sama volá již zmíněnou metodu *getHttpResponse(String wsPassword, String requestUri)*. Vrací mi bytový InputStream jako odpověď ze serveru. Tuto metodu využívám při stahování knihy, ale mohl bych ji využít také i ve všech ostatních částech dotazování se. Chtěl jsem ale využít všech prostředků, které mi byly přiděleny a vyzkoušet si práci se všemi metodami. Parametry jsou opět heslo uživatele a URI řetězec.

3.2 Ověření uživatele

Před prvním použitím aplikace je nutné přihlášení uživatele. Ten se nejprve registruje na webových stránkách knihovny Bookshare, která mu zašle přihlašovací údaje. Tyto údaje uživatel v aplikaci vyplní a bude moci využívat všech služeb aplikace. Bez vyplnění správných údajů uživatele dále nepustím. Již v dalším kroku probíhá vyhledávání, a pokud by neměl platný účet, bylo by to bezpředmětné.

Po zadání přihlašovacích údajů kontroluji, zda jsou zadány správně.

```
public String login (String username, String password) throws URISyntaxException,
IOException {
    BookshareWebservice service = new BookshareWebservice();
    String uri = "https://api.bookshare.org/user/preferences/list/for/" + username;
    HttpResponse resp = service.getHttpResponse(password, uri);
    HttpEntity entity = resp.getEntity();
    String text = service.convertStreamToString(entity.getContent());
    if (text.contains("<status-code>400</status-code>")) {
        System.err.println("Login failed, please try again.");
    }
    loggedIn = true;
    return text;
}
```

Kód 2: Ověření uživatele

V metodě volám třídu BookshareWebservice a využívám její již zmíněné metody. V dokumentaci je také zobrazen přesný tvar textového řetězce *uri*, kterým se ptám na správnost zadaných

přihlašovacích údajů. Mohou nastat pouze dva případy. Pokud je v přihlašovacím jméně nebo hesle chyba, jako odpověď dostanu chybový kód 400. V opačném případě mi server pošle detaily uživatele, se kterými dále mohu pracovat.

Tuto metodu volám při každém vstupu do nabídky vyhledávání. Musím vždy kontrolovat, jestli daný uživatelský účet existuje a zda je stále časově platný. Pokud aplikaci spouštím poprvé nebo nejsem přihlášen, objeví se mi klávesnice pro zadání přihlašovacích údajů. Tyto údaje je nutné zadávat pouze jednou po dobu aktivního účtu. I když aplikaci vypnu a znovu zapnu, automaticky mě to přesměruje do hlavní nabídky vyhledávání.

Tímto si ošetřím situaci, kdy uživatel s účtem po expiraci bude chtít vyhledávat knihu a jako odpověď dostane chybovou hlášku. Takovýchto možných chyb v průběhu hledání je velké množství a jejich kontrolování by mi zbytečně znepráhlednilo kód. Některé situace samozřejmě ošetřit musím, snažím se ale předcházet chybám tak, že je kontroluji hned na začátku. Tyto případy budu komentovat pod jednotlivými částmi kódu, kde to bude aktuální.

U přihlašování rozlišuji, zda uživatel do nabídky teprve vstoupil nebo nezadal správné přihlašovací údaje. V obou případech se objeví klávesnice, ale rozdíl tu je. Informaci, že pro další využívání aplikace je nutné přihlášení stačí sdělit jen jednou, a to při vstupu do aplikace. Naopak po zadání chybných údajů (neexistující účet, účet po expiraci nebo překlep ve jméně nebo hesle) oznamuji uživateli, že přihlášení neproběhlo v pořádku a je nutné ho opakovat. Odhlášení se v tuto chvíli možné není, neboť se tato situace nepředpokládá. Do budoucna bych rád přidal možnost odhlášení se, ale skrytou před uživatelem. Pokud se při testování úspěšně přihlásím, musím znovu zkompileovat celý kód dvakrát. Nejprve přemažu přihlašovací údaje prázdnými řetězci a poté při další kompilaci opět umožním tyto údaje ukládat a automaticky zůstat přihlášen.

Veškeré odpovědi ze serveru Bookshare jsou ve formátu XML.

3.3 Vyhledávání

Možnosti procházení databáze knih jsou zobrazeny v kapitole 2.1 na *obrázku 5*.

Vyhledávání knih se dá rozdělit do dvou jakýchsi skupin. Uživatel může zadat text, podle kterého se bude hledat. Při nespokojenosti s výsledky hledání může tento text pozměnit a selektovat to, co opravdu hledá. Druhou možností je procházení seznamů, kde uživatel nemá kontrolu nad výsledky hledání. Analogií v praxi je, že se můžu zeptat zaměstnance knihovny, nebo jít hledat knihu v regálech sám.

V každém případě má uživatel možnost načíst další knihy dle stejného kritéria hledání, což je užitečné hlavně při procházení pevně daných seznamů knih. Zobrazuji vždy prvních 250 výsledků, což je maximum pro jednu stránku. Při vyhledávání dalších knih se přesunu na další stranu a opět zobrazím 250 výsledků. Pokud jich je méně než 250, došel jsem na konec a informuji uživatele, že se již nedá dále posouvat v seznamu.

3.3.1 Vyhledávání dle zadaného textu

Do první skupiny patří vyhledávání dle názvu knihy, dle názvu autora nebo kombinace názvu a autora. Zde uživatel zadává text, podle kterého se prochází seznamy knih. Nejprve tedy musí zadat přes klávesnici text a poté se zobrazí výsledky hledání.

```
public String searchTitle(BookEntity mBookEntity, String username, String password)
throws URISyntaxException, IOException {
    BookshareWebservice service = new BookshareWebservice();
    String title = "/book/search/title/" + mBookEntity.getTitle() + "/page/1/limit/250/form-
mat/xml";
    String uri = "https://api.bookshare.org/" + title + "/for/" + username;
    HttpResponse resp = service.getHttpResponse(password, uri);
    HttpEntity entity = resp.getEntity();
    String text = service.convertStreamToString(entity.getContent());
    return text;
}
```

Kód 3: Vyhledávání dle názvu knihy

Zbylé metody první skupiny jsou velmi podobné, liší se pouze v prvním řetězci. Parametrem této metody je Interface *BookEntity*, kam ukládám zadaný text od uživatele. V textovém řetězci *uri* je obsaženo kritérium hledání, počet výsledků a formát zobrazení.

V této metodě musím ošetřovat textový řetězec, který uživatel zadává. Jedná se hlavně o odstranění všech znaků typu otazník, vykřičník, apod. To provádím již při zadávání přes klávesnici. Také musím nahradit všechny mezery řetězcem „%20“, který je předepsaný v dokumentaci. Uživateli také nahradím všechny písmena s diakritikou, protože výsledek je zkreslený. V anglickém jazyce diakritika není a neexistuje kniha, která by ji obsahovala. Uživateli se zobrazí absolutně jiný výsledek, než by si představoval a opět bude zmatený, v čem může být problém.

Někdy může nastat situace, že je zadaný řetězec nevyhovující, např. pokud je zadané slovo příliš krátké. V tomto případě uživatele informuji o chybě a ten musí daný text změnit. Zadávaný řetězec může také obsahovat nesmysl nebo slovo s gramatickou chybou. To na vstupu ošetřit nedokážu, ale nic se nestane. Server Bookshare má tento případ ošetřený zevnitř a jednoduše mi jako odpověď vrátí prázdný seznam. Tato situace také nastane, pokud databáze danou, ani nikterak podobnou knihu neobsahuje.

3.3.2 Výpis seznamů

V této skupině uživatel nic nezadává, pouze se mu otevře seznam knih. Sem patří novinky, populární knihy a vyhledávání v kategoriích.

```
public String searchPopular(String username, String password) throws URISyntaxException, IOException {  
    BookshareWebservice service = new BookshareWebservice();  
    String popular= "/book/popular/page/1/limit/250/format/xml";  
    String uri = "https://api.bookshare.org/" + popular + "/for/" + username;  
    HttpResponse resp = service.getHttpResponse(password, uri);  
    HttpEntity entity = resp.getEntity();  
    String text = service.convertStreamToString(entity.getContent());  
    return text;  
}
```

Kód 4: Výpis populárních knih

Seznamy knih se neustále obměňují každý den. Populární knihy jsou řazeny podle počtu stažení uživateli ze serveru Bookshare. Nejsou zde tedy knihy všeobecně oblíbené, ale knihy oblíbené mezi komunitou nevidomých. K mému potěšení jsem dílo 50 odstínů šedi v tomto seznamu nenašel, nejčtenější knihou je první díl anglické verze Harryho Pottera.

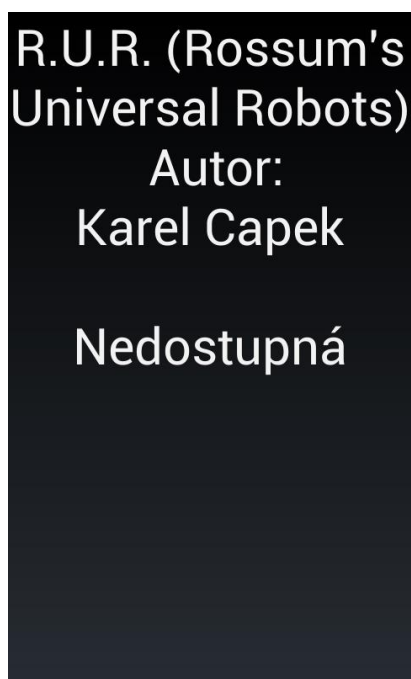
Seznam kategorií je pevně daný, celkem jich je 37. Uživatel si nejprve vybere libovolnou kategorii a poté se mu zobrazí výsledky. Bohužel si nejsem jistý, jakým způsobem probíhá vyhledávání. Je možné, že se zobrazují novinky v daných kategoriích, ale nikde v dokumentaci se o tom nezmiňují. Kategorie jsou stejné, jako všude jinde, jsou zde např. detektivky, sci-fi, vědecká literatura apod. Pokud by server Bookshare do své knihovny přidával nové kategorie, automaticky se uživateli zobrazí.

Já osobně bych si také ještě dokázal představit vyhledávání podle jazyka díla. Neznámější knihy jsou v databázi ve víceru vydání a je matoucí, když při výpisu je za sebou několik stejných děl. Na první pohled ani není jasné, kterým jazykem jsou psány. Rád bych si tedy nejprve vybral v kategorii jazyk, a poté pokračoval v dalším vyhledávání. Výsledky by pak samozřejmě byly relevantní pouze k danému jazyku. V internetovém prohlížeči si uživatel může na svém účtu nastavit preferovaný jazyk, ale nikde není tato možnost popsána pro mobilní aplikace. První výsledek hledání pravděpodobně odpovídá zvolenému jazyku, ale zbytek může být libovolný.

3.4 Zobrazení výsledků hledání

Ve všech případech hledání se uživateli načte seznam 250 knih. Na obrazovce mobilního telefonu je zobrazena vždy jen jedna kniha. Uživatel má možnost posouvat se po výsledcích tažením prstu zleva doprava nebo zprava doleva.

Množství zobrazených informací bylo zvoleno po vyhodnocení výsledků mého dotazníku. Nejčastěji si uživatelé stěžovali, že aplikace obsahuje spoustu informací a „moc mluví“. Snažil jsem se tedy ponechat pouze opravdu důležité informace. Není potřeba zobrazovat např. ID knihy, pod kterým je uložen záznam v databázi, protože to uživateli stejně nic neřekne.



Obrázek 6: Výpis výsledků hledání

Výpis textu začíná na displeji úplně odshora, protože některé výsledky byly příliš dlouhé. Velikost písma se automaticky mění tak, aby se vždy všechno vešlo na obrazovku telefonu. Při velkém množství informací tedy bylo písmo natolik malé, že se nedalo přečíst. Musím totiž předpokládat, že aplikaci Blindshell, jejíž součástí je i má aplikace, využívají mimo jiné i lidé s částečnou poruchou zraku. Nebo například zbylí členové rodiny, kteří mají zrak v pořádku.

Informace ohledně dostupnosti slouží především k testovacím účelům. Vlastním vývojářský účet, který má velmi omezený přístup a většina knih se mi jeví jako nedostupná. Je tedy důležité, abych na první pohled věděl, jestli danou knihu mohu stáhnout. V případě nedostupnosti informuji uživatele o situaci a nic se neprovede. Zároveň také nevím, jestli jsou všechna díla dostupná uživatelům v České republice.

Někdy se při zobrazení výsledků objeví i nerelevantní položky. Například když zadám autora „Capek“, dostanu následující:

- 1) R.U.R. (Rossum's Universal Robots). Author: Karel Capek
- 2) How to Handle Difficult Parents: A Teacher's Survival Guide. Author: Suzanne Capek Tingley
- 3) Difficult Parents. Author: Suzanne Capek Tingley
- 4) Adjudication in Religious Family Law. Author: Gopika Solanki
- 5) Calibrating Your Intuition: Capital Allocation for Market and Credit Risk. Author: Paul Kupiec

Obrázek 7: Hledání dle autora „Capek“

Vybral jsem prvních pět výsledků a již čtvrtý výsledek nemá se zadaných autorem nic společného. Tohle se stává například při neobvyklém jméně autora, kdy počet děl je malý. Nicméně si nemyslím, že je tento přístup správný. Je mnohem jednodušší říci uživateli, že nic jiného se nenašlo, než ho nechat probírat se nesmysly. To bohužel nemohu nijak ovlivnit, protože takhle je nastavené vyhledávání v databázi Bookshare. Psal jsem na technickou podporu e-mail, ale nemyslím si, že budou chtít svůj zaběhlý systém měnit. Nejhorší je, že si pak uživatel pomyslí, že jsem to já špatně nastavil.

Nakonec jsem se rozhodl kontaktovat přímo pověřenou osobu, již zmíněného pana Nicholase. Server Bookshare má prý mimo jiné i fonetické vyhledávání. Může se tedy stát, že vyhledávané slovo bude v anglické syntaxi znít podobně, jako jiné slovo. Pořád to ale nic nemění na faktu, že se koncovému uživateli zobrazí spousta výsledků navíc. Alespoň mě ujistil, že vyhledávání podle textového řetězce má větší prioritu a pokud na prvním místě nebude hledaná položka (nebo alespoň se stejným názvem), daná kniha v databázi neexistuje.

3.5 Stažení knihy

Stažení knihy probíhá v několika krocích. Nejprve se kontroluje, jestli má uživatel přístup k dané knize. Pokud je kniha dostupná, začne probíhat stahování.

```
public void download (Long id, String title, String username, String password)
throws URISyntaxException, IOException{

    String path = Environment.getExternalStorageDirectory().getAbsolutePath() + "/" +
Constants.READER_AUDIOBOOKS_FOLDER + "/" + title;

    String download = "download/content/" + id + "/version/1";

    String uri = "https://api.bookshare.org/" + download + "/for/" + username;

    InputStream in = service.getResponseStream(password, uri);

    java.io.File file = new java.io.File (path + ".zip");

    copy(in, file);

    unzip(path + ".zip", path);

    com.matapo.blindshell.utils.File.deleteFileOrDirectory(file); }
```

Kód 5: Stahování

Novými parametry této metody jsou název a ID knihy. Každá kniha má své unikátní číslo ID, což je nejdůležitější součást řetězce *uri* při stahování. Toto číslo si ukládám již při zobrazení výsledků. Název knihy využívám k tomu, abych v úložišti vytvořil soubor se stejným názvem. Tato část důležitá není, protože přesný název nakonec získávám při parsování obsahu knihy. Mohl bych tedy nový soubor značit jakýmkoliv konstantním názvem, jelikož na konci všechny přebytečné soubory mažu. Tento přístup ale není z programátorského hlediska korektní a problém by nastal např. v případě, že bych si chtěl některé soubory ponechat. Navíc jsem měl tuto třídu implementovanou mnohem dříve, než členění obsahu knihy (o tomto tématu hovořím v kapitole 3.6). Bylo by tedy zbytečné fungující části kódu předělávat.

Soubor ještě opatřím koncovkou ZIP, s tímto typem souboru je pak jednoduché pracovat. Nezapadlo mě jiné řešení, které by se dalo aplikovat na můj případ a bylo z hlediska složitosti jednodušší.

Nyní mám prázdný soubor typu ZIP a binární stream v proměnné typu `InputStream`. Vytvořím si nový `OutputStream` a dostatečně velký buffer. Obsah `InputStreamu` nakopíruji do připraveného souboru pomocí metody `write (byte[] b, int offset, int length)`. Při práci se streamy je důležité vždy vše na konci pozavírat pomocí metody `close()`.

```
private void copy (InputStream in, File file) {
    OutputStream out = new FileOutputStream(String.valueOf(file));
    byte[] buffer = new byte[1024];
    int length;
    while((length=in.read(buffer))>0) {
        out.write(buffer, 0, length); }
    out.close();
    in.close();
}
```

Kód 6: Kopírování `InputStreamu` do souboru

Nejhorší kus práce mám za sebou, neboť v tuto chvíli mám ZIP soubor obsahující všechny soubory knihy ve formátu DAISY. Pokud bych pracoval se všemi soubory, mohl bych je nechat zazipované a ušetřit si tak místo v paměti telefonu. Na tomto principu je založená již zmiňovaná aplikace GoRead, která si soubor odzipuje do mezipaměti, a na konci procesu paměť vymaže.

Já se ale potřebuji dostat pouze k jednomu souboru a s ním dále manipulovat. Přišlo mi nejjednodušší soubor typu ZIP rozbalit do složky. Na internetu je spousta návodů, jak tohoto docílit. Inspiroval jsem se dokumentací⁹⁾ Javy.

Nejprve se vytvoří složka, která se jmenuje stejně, jako zazipovaný soubor. Data se čtou pomocí třídy `ZipInputStream`, která je navržena přímo pro čtení dat ze souborů typu ZIP. Jakmile je tok dat `InputStream` otevřen, objekty uvnitř se načítají pomocí metody `getNextEntry()`, která vrací objekty typu `ZipEntry`. V každém kroku se přečte soubor ze zazipovaného souboru. V nově vytvořené složce vznikne kopie tohoto souboru, která má stejný název, ale nulovou velikost. Až v dalším vnořeném cyklu se pomocí třídy `FileOutputStream` překopírují data do nového souboru metodou `write (byte[] b, int offset, int length)`. Tento proces je stejný jako u metody `copy (InputStream in, File outputFile)`. To probíhá tak dlouho, dokud metoda `getNextEntry()` nepřerečte všechny soubory.

Poté co rozbalím ZIP soubor do složky stejného názvu, původní zazipovaný soubor odstraním. To provádím na konci metody `download (Long id, String title, String username, String password)` na začátku této kapitoly.

⁹ <http://www.oracle.com/technetwork/articles/java/compress-1565076.html>

```

public void unzip(String zipFile, String outputFolder){
    byte[] buffer = new byte[1024];
    try{
        File folder = new File (outputFolder);
        if (!folder.exists()){
            folder.mkdir();
        }
        ZipInputStream zis = new ZipInputStream(new FileInputStream(zipFile));
        ZipEntry ze = zis.getNextEntry();
        while(ze != null){
            String fileName = ze.getName();
            File newFile = new File (outputFolder + File.separator + fileName);
            FileOutputStream fos = new FileOutputStream(newFile);
            int len;
            while ((len = zis.read(buffer)) > 0) {
                fos.write(buffer, 0, len);
            }
            fos.close();
            ze = zis.getNextEntry();
        }
        zis.closeEntry();
        zis.close();
    } catch (IOException ex){
        ex.printStackTrace();
    }
}

```

Kód 7: Rozbalení zip souboru

3.6 Extrakce knihy

V tuto chvíli mám složku, která obsahuje spoustu souborů. Tuto situaci nastiňuje *obrázek 4* v kapitole 2. Zajímá mě pouze soubor s koncovkou XML, ve kterém se nachází obsah knihy. Vše ostatní mohu smazat a uvolním tak velké množství paměti, je to poslední krok celého procesu stažení knihy. *Kód 8* je součástí metody *download* (*Long id*, *String title*, *String username*, *String password*) kapitoly 3.5, proto není uveden hlavičkou. Navazuje přímo na poslední řádek *kódu 5*, kdy odstraňuji soubor typu ZIP. Ukázka knihy v podobě XML je zobrazena v příloze C.

```
BufferedReader br = new BufferedReader(new FileReader(path + "/" + fileName));
String line;
while((line = br.readLine()) != null) {
    if (line.contains("Title\ " content")){
        String name = getTitle (line, "\\");
    }
    if (line.contains("bodymatter")){
        while ((line = br.readLine()) != null) {
            if (line.contains("pagenum")){
                continue;
            }
            SplitUsingTokenizer(line,"<>");
        }
    }
    String pathname = Environment.getExternalStorageDirectory().getAbsolutePath() +
"/" + Constants.READER_BOOKS_FOLDER + "/" + name + ".txt";
    PrintWriter writer = new PrintWriter(pathname, "UTF-8");
    writer.print(text);
    writer.close();
    br.close();
    com.matapo.blindshell.utils.File.deleteFileOrDirectory(folder);
```

Kód 8: Vytvoření finálního souboru

Použiji `BufferedReader` k otevření datového toku souboru XML. XML soubor obsahuje opravdu velké množství řádků a spoustu toho nepotřebuji. Na začátku souboru jsou úplně všechny informace o knize. Mohl bych tedy uživateli o dané knize povědět spoustu, ale tento krok jsem přeskočil. Tyto informace se pro nevidomé lidi stávají zbytečnou přítěží, kdy trvá velmi dlouho, než hlasová syntéza přečte všechny údaje. Zajímá mě pouze název knihy, který je nejpřesnější. Když stahuji knihu a vytvářím soubory a složky s názvem knihy, mezery jsou nahrazovány pomlčkami a není to přehledné. Až výsledný textový soubor má název, který odpovídá názvu knihy. Ten získám metodou `getTitle(String subject, String delimiters)`. Název knihy je v souboru zobrazen ve tvaru:

```
<meta name="dc:Title" content="An Essay Upon Projects" />
```

Jakmile tedy narazím v řádku na řetězec „`Title content`“, zavolám metodu v *kódu 9*. Používám k parsování třídu `StringTokenizer`, která řádek rozkouskuje podle oddělovače, v tomto případě dle uvozovek. Získám tak pět částí, které vypadají následovně:

- `<meta name=`
- `dc:Title`
- `content=`
- `An Essay Upon Projects`
- `/>`

Stačí jen vzít čtvrtou část, která v sobě nese název knihy. Jazyk Java indexuje od nuly, proto v *kódu 9* beru část řetězce pod indexem tři.

```
public static String getTitle (String subject, String delimiters) {
    StringTokenizer st = new StringTokenizer(subject, delimiters);
    ArrayList<String> list = new ArrayList<String>(subject.length());
    While (st.hasMoreTokens())
        list.add (st.nextToken());
    return list.get(3);
}
```

Kód 9: Extrakce názvu knihy

Takovýto tvar řádku mají všechny knihy, proto mohu ponechat konstantní index. Dalším způsobem řešení by bylo, že bych určoval index podle postavení ostatních částí. Tuto část kódu bych v budoucnu vylepšit mohl, ale momentálně vše funguje bezchybně.

Dále v textu hledám klíčové slovo „*bodymatter*“, které označuje začátek knihy. Před tímto začátkem jsou různá ustanovení a autorská práva, která ale koncového uživatele nezajímají. Takto projdu celý zbytek textového souboru a v každém řádku volám metodu *splitTokenizer(String subject, String delimiters)*. Nejprve ale kontroluji, zda se v řádku neobjevuje řetězec „*pagenum*“. Je to pouze označení stránky a na ně to já nerozděluji. Text na displeji má nastavenou minimální velikost (ze zbytku textu, co se nevejde na displej, se vytvoří další stránka).

Metoda *splitTokenizer(String subject, String delimiters)* zobrazena v *kódu 10* funguje na stejném principu jako *kód 9*. Opět se rozdělují řádek textu podle daných oddělovačů, v tomto případě dle lomených závorek. Tentokrát ale nevracím žádnou hodnotu, metoda je typu `void`. Řádek knihy může vypadat například takto:

```
<p id="para_000065" smilref="An_Essay_Upon_Projects00001.smil#para_000065">Your most obliged, humble servant</p>
```

Výsledkem procesu parsování je opět několik textových řetězců:

- `p id="para_000065" smilref="An_Essay_Upon_Projects00001.smil#para_000065"`
- `Your most obliged, humble servant`
- `/p`

Nejprve kontroluji, zda má řádek alespoň tři rozdělené části. Některé řádky jsou totiž např. jen uvozující nebo ukončující a neobsahují text knihy. Pokud řádek text obsahuje, je vždy uvozen tagy a nachází se na druhém místě. Proto opět mohu využít konstantní index při výběru z `ArrayListu`.

Nezmínil jsem ještě proměnnou *text* v *kódu 8* a *kódu 10*. Jedná se o globální proměnnou typu `StringBuilder`. Globální proto, abych k ní mohl přistupovat ve všech metodách dané třídy. Umožňuje mi skládat textové řetězce za sebe a výsledkem je jeden dlouhý řetězec. Za každý řádek textu ještě přikládám znak pro konec řádku. Je to především kvůli přehlednosti a snazší kontrole správnosti výsledků. Také mi to člení text na stránky, ale o stránkování jsem již mluvil.

Nyní se opět vrátím ke *kódu 8*, kdy jsem rozparsoval celý text a jsem nyní připraven s ním dále pracovat. Nejprve si vytvořím textový soubor s koncovkou `TXT`, který bude mít název získaný z metody *kódu 9*. Vytvořím si proměnnou *writer* třídy `PrintWriter` a nastavím jí kódování typu `UTF-8`. Tento typ kódování je doporučený v dokumentaci serveru `Bookshare`. Využiji funkci *print(String s)*, která mi přepíše do nově vytvořeného souboru obsah proměnné *text*. Na konci všech datových toků pozavírám a smažu složku, která obsahovala soubory typu `DAISY`.

V tuto chvíli jsem se dostal na úplný závěr implementace. Kniha je uložena v textovém formátu a je připravena k použití. Výsledek parsování uvádím v příloze `D`, kde je porovnání textu původního XML souboru a textu, který jsem si ponechal.

```
public static void splitTokenizer(String subject, String delimiters) {
    StringTokenizer st = new StringTokenizer(subject, delimiters);
    ArrayList<String> list = new ArrayList<String>(subject.length());
    while(st.hasMoreTokens())
        list.add(st.nextToken());
    if (list.size()<3){
        return null;
    }else{
        text.append(list.get(1));
        text.append("\r\n");
    }
}
```

Kód 10: Parsování textu

4 Testování

Po dokončení aplikace přišlo na řadu testování. Cílem bylo najít co největší počet chyb a co nejvíce zjednodušit ovládání. Aplikaci jsem testoval na svém mobilním zařízení, Samsung Galaxy S3 mini. V této části nebudu hovořit o chybách, které jsem našel já nebo někdo jiný. Přijde mi zbytečné o tom jen hovořit, a proto jsem vše opravil ihned a okomentoval to v předchozích kapitolách, kde byly aktuální. Tato aplikace je výsledkem mé celoroční práce a snažil jsem se udělat vše pro to, aby fungovala bezchybně. Až postupem času se ukáže, jak bylo mé nynější snažení úspěšné.

Aplikaci jsem ale netestoval sám, pomáhalo mi při tom hodně lidí. Spoustu chyb objevila má přítelkyně a můj vedoucí práce, když zkoušeli nasimulovat různé situace. Já osobně jsem s mým kódem a aplikací natolik sžitý, že je pro mě takřka nemožné objevit něco nového. Vše беру jako samozřejmost a vím, že hlavní funkce fungují správně. Bohužel je to ale o malých nedostatcích, které ve výsledku snižují kvalitu práce.

Největší dík při testování aplikace zasluhují lidé ze společnosti SONS. V kapitole 2.1 jsem hovořil o návrhovém dotazníku, který jsem těmto uživatelům předložil. Po dokončení aplikace a odstranění největších problémů jsem opět požádal tuto společnost o finální otestování a o vyplnění hodnotícího dotazníku.

4.1 Hodnotící dotazník

Hodnotící dotazník jsem sestavil po dokončení aplikace a slouží k získání zpětné vazby od uživatelů, kteří mojí aplikaci používali. Průzkumu se opět účastnilo pět lidí s částečným nebo úplným zrakovým postižením. Domnívám se, že tento dotazník byl předložen stejným lidem, kteří mi vyplňovali návrhový dotazník. Mohli tedy sami posoudit, zda jsem se v aplikaci skutečně inspiroval jejich podněty. Hodnotící dotazník s otázkami a odpověďmi je k nahlédnutí v příloze B.

Opět jsem pokládal pouze pět otázek a zajímal se především o spokojenost a pohodlí uživatelů. Nejprve jsem se ptal, zda je má aplikace pro nevidomou komunitu užitečná. U této otázky jsem obdržel pouze pozitivní odpovědi (které snad nebyly ze solidarity). Nejde jen o konkrétní knihovnu Bookshare, ale o celkově nové možnosti vyhledávání knížek v telefonu. Vše je připravené na jednom místě a stačí se jen proklikat jednotlivými položkami.

Zajímal jsem se také o rychlost vyhledávání. Je totiž velmi důležité, aby při jednotlivých fázích nenastávaly prostoje, zvláště pokud uživatel na mobilní zařízení nevidí. Jde především o různé kontrolování podmínek a procházení seznamů. Na rychlosti dílčích procesů jsem se ale zaměřoval již v průběhu a i také uživatelé ocenili svižnost mé aplikace. Jsem dosti nervní člověk, pokud jde o čekání, až se něco stane a jistě nejsem sám. Takovéto pocity by lidé při používání mé aplikace jistě zažívat neměli. Rychlost je ale velmi relativní veličina, která se na různých

mobilních zařízeních projevuje různě. Nemohu tedy předvídat, jak rychle bude reagovat např. starší zařízení, jelikož to nemám jak porovnat.

Zobrazení výsledků jsem chtěl řešit již v návrhovém dotazníku, ale je těžké si to představit bez nějaké konkrétní ukázky. Ptal jsem se uživatelů, jak jsou spokojeni s množstvím informací zobrazených u každé knihy. Tato otázka je dost subjektivní, neboť každý má jiné měřítko, co je důležité, a co není. Výsledky napovídají, že skupina pěti testujících uživatelů byla poměrně spokojena. V poznámkách opět někdo poznamenával, že na jeho vkus to „moc mluví“. Já se domnívám, že jsem se dostal na nejnižší možnou hranici, kdy jsou pořád informace relevantní, ale není jich příliš. Tvar zobrazení knihy uživatelům je znázorněn v kapitole 3.4 na *obrázku 6*.

V další otázce jsem se uživatelů ptal, zda si dokáží představit používání této nebo jiné podobné aplikace v běžném životě. Všichni samozřejmě tuto variantu vítají, jelikož dotazovaní uživatelé ve svém životě dotykové telefony používají. Vše mohou dělat z pohodlí svého domova a nepotřebují k tomu žádnou jinou techniku, ke všemu jim postačí mobilní zařízení. Uživatelé mi v poznámkách vytýkali, že knihovna Bookshare není určena pro cílovou skupinu českých lidí. V tom s nimi souhlasím, protože české knihy se tu opravdu nenacházejí. Na druhou stranu musím brát v potaz, že mojí aplikaci, jakožto součást aplikace Blindshell, mohou využívat lidé po celém světě. Pro ně to jistý přínos mít může. Zároveň je to ale také průprava na možnost rozšíření pro další on-line knihovnu. Snad každý nevidomý uživatel, se kterým jsem svou aplikaci konzultoval, se mě ptal, proč raději neudělám aplikaci pro připojení na KDD¹⁰). KDD je knihovna digitálních dokumentů určená především pro zrakově postižené. Věřím, že s aplikací připojující se na KDD bych měl u českých uživatelů obrovský úspěch. Nemůžu se ale zavděčit všem, zvláště pokud mám jasně dané zadání práce. V budoucnu se ale nebráním vytvoření takové aplikace, když už vím, jak to zhruba funguje.

V poslední otázce jsem se zajímal o složitost celého ovládání. Jak jsem již v průběhu nastiňoval, mým cílem bylo vytvoření aplikace, která se jednoduše ovládá. Nechtěl jsem časově ani myšlenkově zatěžovat budoucí uživatele. Výsledky jasně ukazují, že má aplikace není složitá. Jak v průběhu vyhledávání, tak při finálním stažení knihy.

¹⁰ <http://www.kdd.cz/>

5 Závěr

V rámci této práce jsem implementoval zpřístupnění audio-knihovny Bookshare nevidomým uživatelům pomocí mobilní aplikace. Ti nyní mohou využívat plného potenciálu knihovny, který skýtá.

Při tvorbě jsem velkou část konzultoval s vedoucím práce a s lidmi ze společnosti SONS. Výsledkem pak byl kompromis vyhovující všem stranám a především realizovatelný z mého pohledu. Jednalo se ale vždy o detaily, v hlavní části na improvizaci nebyl prostor.

Aplikaci jsem otestoval na pěti nevidomých uživatelích a ti mi také pomohli vyplnit návrhový a hodnotící dotazník. Z hlediska výsledků dotazníků si dovoluji tvrdit, že jsem svou aplikaci navrhl a naprogramoval poměrně dobře. Odezva na jednotlivé úkony je rychlá, eliminoval jsem velké množství chyb a celý proces funguje od začátku do konce bezvadně.

O dalších možnostech rozšíření aplikace jsem hovořil v průběhu mé práce. Stále je zde několik nedostatků, které se dají vylepšit. Jsou to ale spíše drobnosti, vše podstatné jsem myslím vyřešil. Záleží také na serveru Bookshare, zda se bude v budoucnu někam vyvíjet a posouvat. Reakce na případné změny by pak byly na místě.

Snažím se dokončit veškerou dokumentaci k své aplikaci, aby mohla být vydána jako součást příští aktualizace aplikace Blindshell. Rád bych ji zpřístupnil koncovým uživatelům co nejdříve, aby byla používána a neupadla v zapomnění.

6 Použité zdroje

- [1] Suzor, Nicolas P., Harpur, Paul D., & Thampapillai, Dilan - *Digital copyright and disability discrimination: From braille books to bookshare*. (2008)
- [2] Vincent Gaudissart, Silvio Ferreira, Celine Thillou, Bernard Gosselin, SYPOLE - *Mobile Reading Assistant for Blind People, SPECOM- 9th Conference Speech and Computer*. (2004)
- [3] Schildt, Herbert. *Java 7, Výukový kurz*. ISBN 978-80-251-3748-2.
- [4] Android developers. (<http://developer.android.com/develop/index.html>).
- [5] Aplikace GoRead. (<https://play.google.com/store/apps/details?id=org.benetech.android>).
- [6] Aplikace Audible. (<https://play.google.com/store/apps/details?id=com.audible.application>).
- [7] Aplikace LibriVox. (<https://play.google.com/store/apps/details?id=biz.bookdesign.librivox>).

7 Seznam obrázků

Obrázek 1:	Uživatelské prostředí aplikace GoRead.....	14
Obrázek 2:	Uživatelské prostředí aplikace Audible.....	15
Obrázek 3:	Uživatelské prostředí aplikace LibriVox.....	16
Obrázek 4:	Knihy ve formátu DAISY	17
Obrázek 5:	Stromová struktura vyhledávání	20
Obrázek 6:	Výpis výsledků hledání.....	27
Obrázek 7:	Hledání dle autora „Capek“	28
Obrázek 8:	Výsledek odpovědi na otázku týkající se znalosti cizího jazyka.....	42
Obrázek 9:	Výsledek odpovědi na otázku týkající se úrovně anglického jazyka	42
Obrázek 10:	Výsledek odpovědi na otázku týkající se možností vyhledávání.....	43
Obrázek 11:	Výsledek odpovědi na otázku týkající se dialogového okna	43
Obrázek 12:	Výsledek odpovědi na otázku týkající se možností dialogového okna	44
Obrázek 13:	Výsledek odpovědi na otázku týkající se užitečnosti vyhledávání.....	45
Obrázek 14:	Výsledek odpovědi na otázku týkající se rychlosti vyhledávání.....	45
Obrázek 15:	Výsledek odpovědi na otázku týkající se kvality zobrazení výsledků.....	46
Obrázek 16:	Výsledek odpovědi na otázku týkající se využití aplikace v běžném životě	46
Obrázek 17:	Výsledek odpovědi na otázku týkající se složitosti vyhledávání.....	47

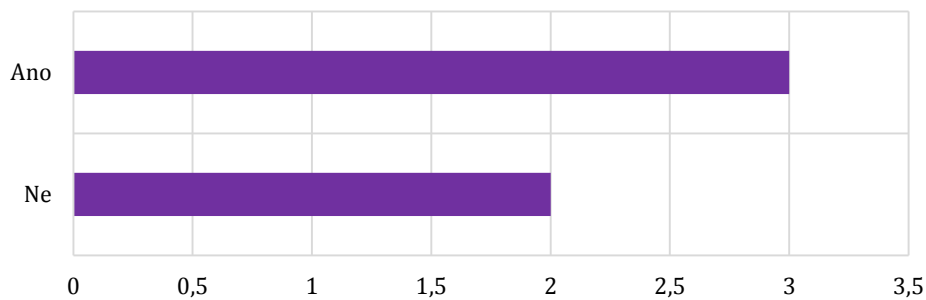
8 Seznam kódů

Kód 1:	Kontrola připojení	21
Kód 2:	Ověření uživatele	23
Kód 3:	Vyhledávání dle názvu knihy	25
Kód 4:	Výpis populárních knih	26
Kód 5:	Stahování.....	29
Kód 6:	Kopírování InputStreamu do souboru	30
Kód 7:	Rozbalení zip souboru	31
Kód 8:	Vytvoření finálního souboru	32
Kód 9:	Extrakce názvu knihy.....	33
Kód 10:	Parseování textu.....	35

9 Příloha A

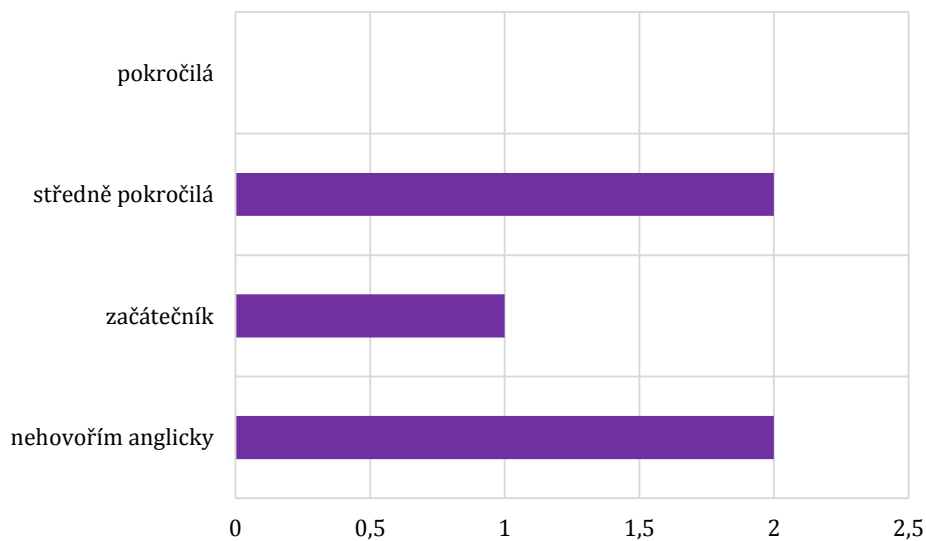
Příloha A obsahuje otázky a odpovědi návrhového dotazníku.

1) Hovoříte cizím jazykem?



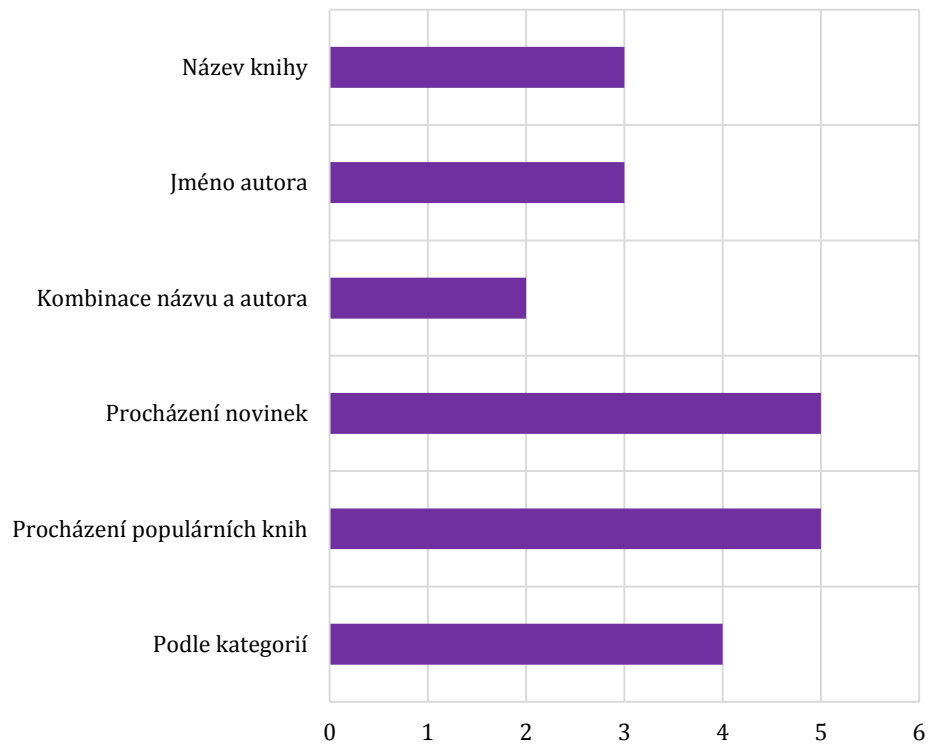
Obrázek 8: Výsledek odpovědí na otázku týkající se znalosti cizího jazyka

2) Jaká je Vaše úroveň anglického jazyka?



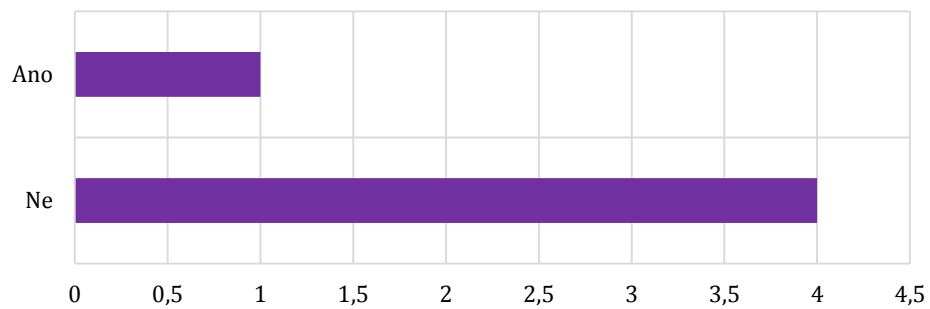
Obrázek 9: Výsledek odpovědí na otázku týkající se úrovně anglického jazyka

3) Dle jakých kritérií vyhledávání byste chtěli procházet katalog knih?



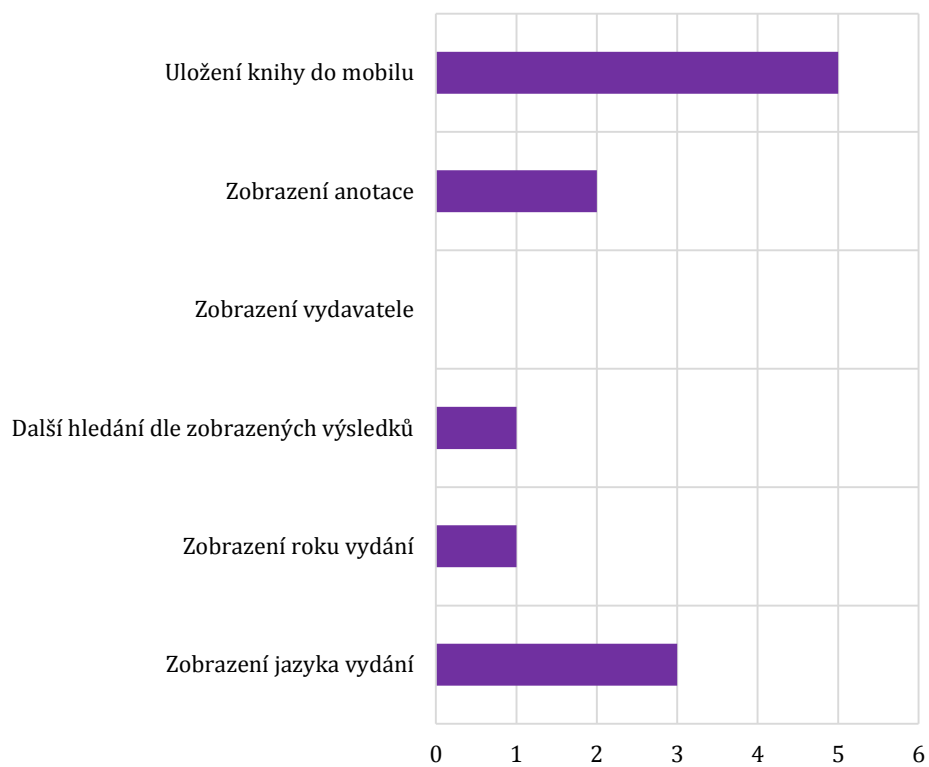
Obrázek 10: Výsledek odpovědí na otázku týkající se možností vyhledávání

4) Přejete si mít u výsledků hledání knih dialogové okno?



Obrázek 11: Výsledek odpovědí na otázku týkající se dialogového okna

5) Pokud ano, které možnosti byste ocenili?

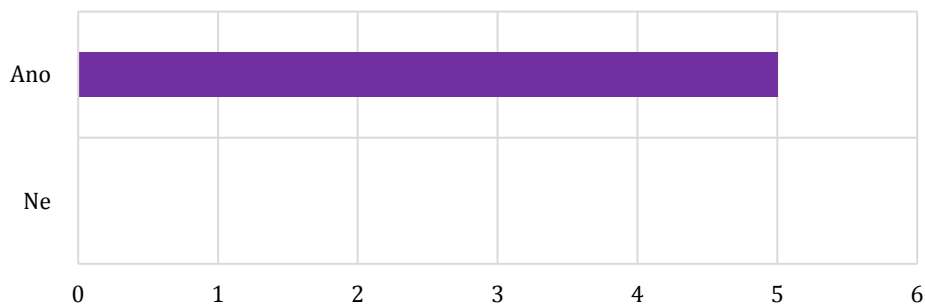


Obrázek 12: Výsledek odpovědí na otázku týkající se možností dialogového okna

10 Příloha B

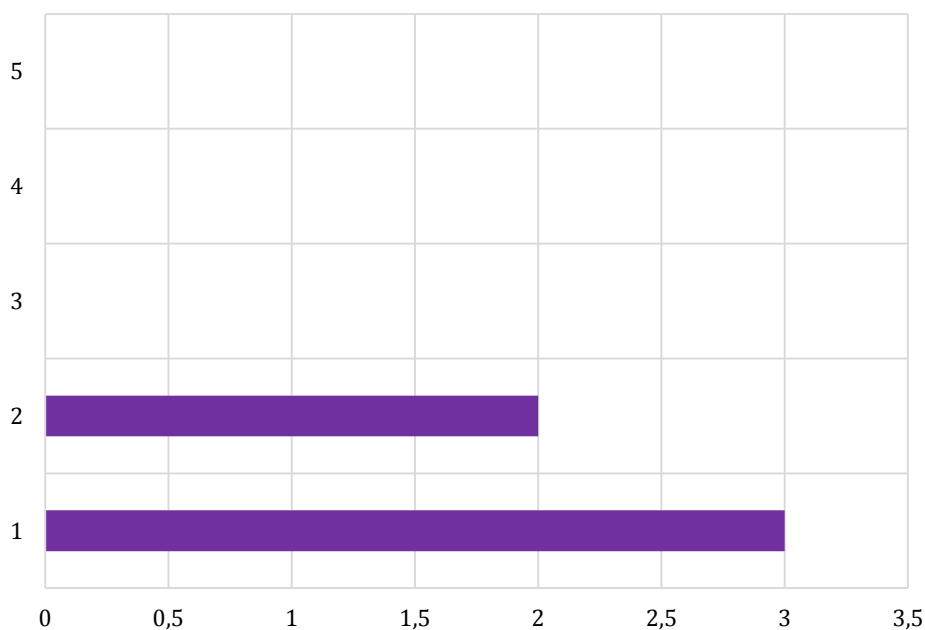
Příloha B obsahuje otázky a odpovědi hodnotícího dotazníku.

1) Přejde Vám metoda vyhledávání knih pomocí telefonu užitečná?



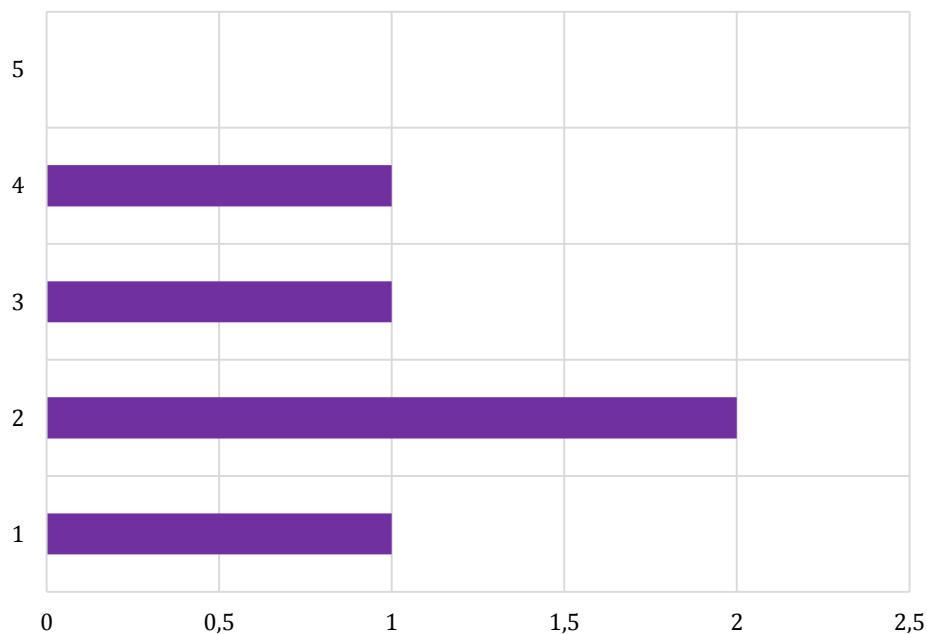
Obrázek 13: Výsledek odpovědí na otázku týkající se užitečnosti vyhledávání

2) Ohodnoťte prosím od 1 do 5 rychlost vyhledávání knih (1 velmi rychlé, 5 velmi pomalé)



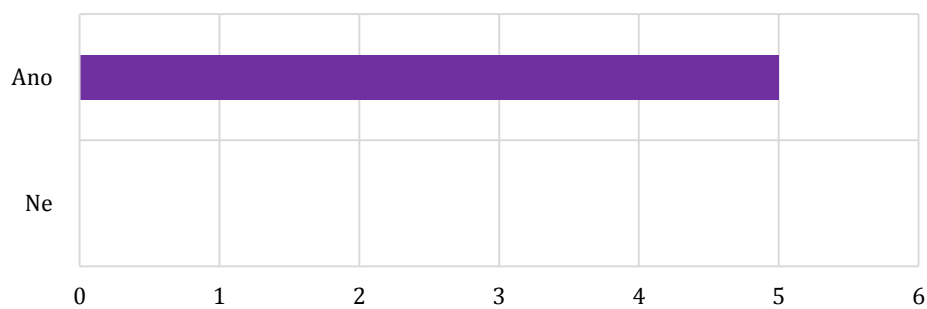
Obrázek 14: Výsledek odpovědí na otázku týkající se rychlosti vyhledávání

- 3) Ohodnoťe prosím od 1 do 5 kvalitu zobrazení výsledků (1 vše je podle představ, 5 vše je špatně)



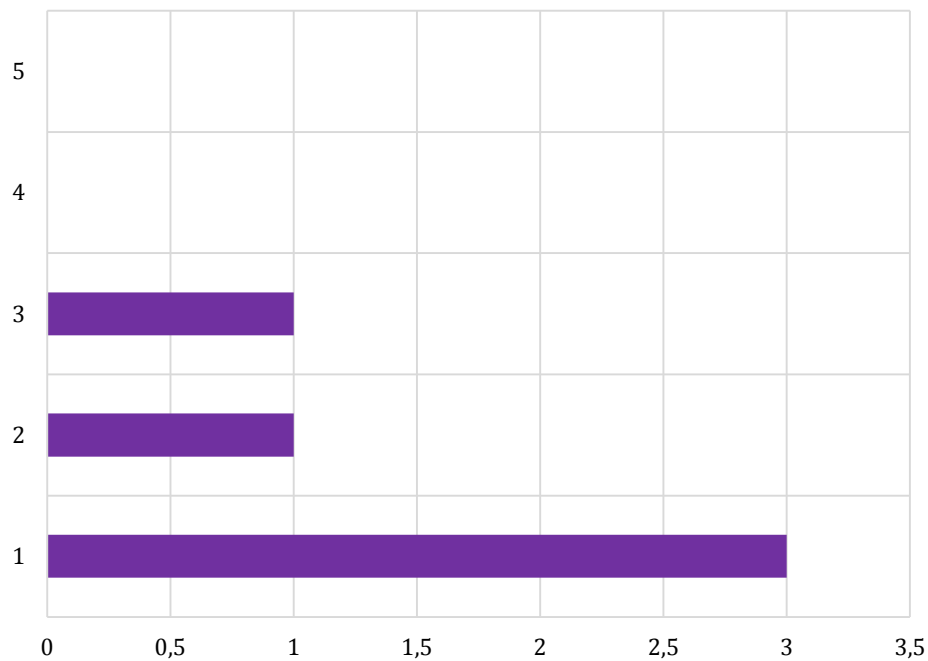
Obrázek 15: Výsledek odpovědí na otázku týkající se kvality zobrazení výsledků

- 4) Dokážete si představit, že byste vyhledávání a následné ukládání knih pomocí chytrého telefonu používal/a v běžném životě? (Jakákoliv knihovna, např. KDD)



Obrázek 16: Výsledek odpovědí na otázku týkající se využití aplikace v běžném životě

5) Ohodnoťe prosím od 1 do 5 složitost vyhledání a uložení knihy (1 velmi jednoduché, 5 velmi složité)



Obrázek 17: Výsledek odpovědí na otázku týkající se složitosti vyhledávání

11 Příloha C

Úryvek knihy „An essay upon projects“ od autora Daniel Defoe v původním tvaru:

```
<bodymatter id="bodymatter_000001">
```

```
  <level1 class="part" id="start">
```

```
    <h1 class="part" id="h1_000001" smilref="An_Essay_Upon_Pro-  
jects00001.smil#h1_000001">An Essay Upon Projects</h1>
```

```
    <pagenum id="p1" page="normal" smilref="An_Essay_Upon_Pro-  
jects00001.smil#p1">1</pagenum>
```

```
    <p id="para_000012" smilref="An_Essay_Upon_Projects00001.smil#para_000012">AN  
ESSAY UPON PROJECTS</p>
```

```
    <p id="para_000013" smilref="An_Essay_Upon_Projects00001.smil#para_000013">By  
Daniel Defoe</p>
```

```
    <p id="para_000017" smilref="An_Essay_Upon_Projects00001.smil#para_000017">Con-  
tents:</p>
```

```
    <p id="para_000018" smilref="An_Essay_Upon_Projects00001.smil#para_000018">In-  
troduction Author's Preface Author's Introduction The History of Projects Of Projectors Of  
Banks Of the Multiplicity of Banks Of the Highways Of Assurances Of Friendly Societies Of Se-  
amen Of Wagering Of Fools A Charity-Lottery Of Bankrupts Of Academies Of a Court Mer-  
chant Of Seamen The Conclusion</p>
```

```
    <p id="para_000019" smilref="An_Essay_Upon_Projects00001.smil#para_000019">IN-  
TRODUCTION.</p>
```

```
    <p id="para_000020" smilref="An_Essay_Upon_Projects00001.smil#para_000020">Defo-  
e's "Essay on Projects" was the first volume he published, and no great writer ever published  
a first book more characteristic in expression of his tone of thought. It is practical in the  
highest degree, while running over with fresh speculation that seeks everywhere the well-  
being of society by growth of material and moral power. There is a wonderful fertility of  
mind, and almost whimsical precision of detail, with good sense and good humour to form the  
groundwork of a happy English style. Defoe in this book ran again and again into sound  
suggestions that first came to be realised long after he was dead. Upon one subject, indeed,  
the education of women, we have only just now caught him up. Defoe wrote the book in 1692  
or 1693, when his age was a year or two over thirty, and he published it in 1697.</p>
```


12 Příloha D

Tentýž úryvek knihy „An essay upon projects“ od autora Daniel Defoe po úpravě:

An Essay Upon Projects

AN ESSAY UPON PROJECTS

By Daniel Defoe

Contents:

Introduction Author's Preface Author's Introduction The History of Projects Of Projectors Of Banks Of the Multiplicity of Banks Of the Highways Of Assurances Of Friendly Societies Of Seamen Of Wagering Of Fools A Charity-Lottery Of Bankrupts Of Academies Of a Court Merchant Of Seamen The Conclusion

INTRODUCTION.

Defoe's "Essay on Projects" was the first volume he published, and no great writer ever published a first book more characteristic in expression of his tone of thought. It is practical in the highest degree, while running over with fresh speculation that seeks everywhere the well-being of society by growth of material and moral power. There is a wonderful fertility of mind, and almost whimsical precision of detail, with good sense and good humour to form the groundwork of a happy English style. Defoe in this book ran again and again into sound suggestions that first came to be realised long after he was dead. Upon one subject, indeed, the education of women, we have only just now caught him up. Defoe wrote the book in 1692 or 1693, when his age was a year or two over thirty, and he published it in 1697.