



**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

F3

**Fakulta elektrotechnická
Katedra počítačů**

Bakalářská práce

Návrh a realizace jednoduchého systému pro zpracování dotazníků

Radek Mahdal

Softwarové inženýrství a technologie

Květen 2018

Vedoucí práce: Ing. Pavel Náplava



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Mahdal** Jméno: **Radek** Osobní číslo: **456886**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Návrh a realizace jednoduchého systému pro zpracování dotazníků

Název bakalářské práce anglicky:

Design and development of a simple survey system

Pokyny pro vypracování:

1. Analyzujte oblast tvorby a zpracování dotazníků.
2. Proveďte analýzu existujících řešení a vytvořte návrh jednoduchého systému, který umožní dotazníky vytvářet, sbírat a vyhodnocovat odpovědi. Klíčovým parametrem systému je jednoduché zprovoznění a používání uživateli bez hlubších IT znalostí. Zvažte variantu rozdělení systému do více samostatných aplikací. Systém musí být navržený tak, aby jej bylo možné v budoucnu rozšířit o další funkční moduly.
3. Implementujte minimálně funkční prototyp navrženého systému.
4. Společně s vedoucím práce navrhnete alespoň tři různé typy dotazníků a scénářů sběru dat, které následně použijete pro uživatelské testování vytvořeného prototypu systému.

Seznam doporučené literatury:

-Ariow, J., Neustat, I.: UML 2 a unifikovaný proces vývoje aplikací. Computer Press, ISBN: 978-80-251-1503-9, Praha 2007.
-Ron Patton, Testování softwaru, 2002, COMPUTER PRESS, ISBN: 8072266365
-Svobodová, H., Mynářová, L., Kozel R.; Moderní metody a techniky marketingového výzkumu, 2011, GRADA, ISBN: 978-80-247-3527-6

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Pavel Náplava, Centrum znalostního managementu

Jméno a pracoviště druhého(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **19.02.2018** Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce: **30.09.2019**

Ing. Pavel Náplava
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování / Prohlášení

Chtěl bych poděkovat především panu Ing. Pavlu Náplavovi, za velmi ochotnou spolupráci a trpělivost při vedení této práce, a také Lukáši Krýdovi, se kterým jsem spolupracoval při návrhu této práce v rámci Semestrálního projektu.

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 25. 05. 2018

.....

Abstrakt / Abstract

Tento dokument se zabývá analýzou dotazníků a testů. Popisuje jejich dělení, využití a způsob tvorby a plnění. Následně na základě těchto poznatků navrhuje systém, který umožňuje jeho uživatelům dotazníky vytvářet a vyplňovat. Také obsahuje popis vývoje prvního prototypu tohoto systému a jeho součástí, včetně vytvoření několika typů dotazníků za účelem uživatelského testování.

Klíčová slova: dotazník; test; návrh systému; implementace; prototyp.

This document analyses the topic of questionnaires and tests. It describes their types, usage and the process of constructing and filling them. Using the results of the analysis, it proposes a design of a system, which allows the construction of questionnaires and collection of answers. It also describes the implementation of a prototype of this system, including the construction of a few questionnaire types used for user-based testing.

Keywords: questionnaire; test; system design; implementation; prototype.

Title translation: Design and development of a simple survey system

Obsah /

1 Úvod	1	4.3.2 Okno výběru	22
1.1 Motivace	1	4.3.3 Okno označení	23
1.2 Cíle	1	4.3.4 Okno otázky	23
2 Zavedení do problematiky	2	4.3.5 Okno vyhodnocení	24
2.1 Rozdělení Dotazníků	2	4.3.6 Okno nabídky	25
2.1.1 Testy	2	5 Navržené dotazníky	26
2.1.2 Dotazníky	2	5.1 Matematický test	26
2.2 Využití dotazníků a testů	3	5.1.1 Stavba dotazníku	26
2.2.1 Průzkumy veřejného		5.1.2 Sledování	26
mínění	3	5.2 SUS	26
2.2.2 System Usability Scale	3	5.2.1 Stavba dotazníku	26
2.2.3 Belbinův test rolí	4	5.2.2 Sledování	27
2.3 Existující programy	4	5.3 BelbinAlt	27
2.3.1 Formuláře google	4	5.3.1 Stavba dotazníku	27
2.3.2 Quotev	5	5.3.2 Sledování	28
2.3.3 QuickTapSurvey	6	6 Závěr	29
2.3.4 Survio	6	6.1 Zhodnocení	29
2.4 Shrnutí	7	6.2 Návrh na pokračování vývo-	
3 Návrh	8	je systému	29
3.1 Předběžný návrh	9	Literatura	31
3.2 Požadavky	10	A Přílohy	33
3.3 Funkční požadavky	10	B Zkratky	34
3.3.1 Tvorba dotazníků	10		
3.3.2 Plnění dotazníků	10		
3.4 Nefunkční požadavky	10		
3.5 Případy užití	11		
3.6 Diagram tříd	12		
3.7 Soubor dotazníku	13		
3.8 Diagramy aktivit	14		
3.8.1 Založení dotazníku	14		
3.8.2 Vyplnění dotazníku	14		
3.9 Diagram nasazení	15		
4 Implementace	16		
4.1 Specifikace implementace	16		
4.1.1 Tvorba a čtení souboru ..	16		
4.1.2 Udržování dotazníku	17		
4.2 Prototyp aplikace pro vytvá-			
ření dotazníků	18		
4.2.1 Hlavní okno	18		
4.2.2 Okno dotazníku	18		
4.2.3 Okno výsledku	19		
4.2.4 Okno otázky	20		
4.2.5 Okno odpovědi	20		
4.3 Prototyp aplikace vyplňová-			
ní dotazníků	21		
4.3.1 Hlavní okno	21		

/ **Obrázky**

2.1.	Tvorba formuláře v aplikaci Formuláře Google	4
2.2.	Tvorba formuláře v aplikaci Quotev	5
2.3.	Vyplnění formuláře v aplikaci QuickTapSurvey	6
2.4.	Přidání otázky v aplikaci sur- vivo	6
3.1.	Předběžný návrh funkcionality ..	9
3.2.	Diagram případu užití	11
3.3.	Diagram tříd struktury	12
3.4.	Diagram aktivity založení do- tazníku.....	14
3.5.	Diagram aktivity vyplnění dotazníku	14
3.6.	Diagram nasazení	15
4.1.	Hlavní okno	18
4.2.	Okno dotazníku	19
4.3.	Okno výsledku	20
4.4.	Okno otázky	20
4.5.	Okno odpovědi	21
4.6.	Android Hlavní okno.....	22
4.7.	Android Okno výběru.....	22
4.8.	Android Okno označení	23
4.9.	Android Okno otázky	24
4.10.	Android Okno vyhodnocení ...	24
4.11.	Android Okno nabídky	25
5.1.	Vyhodnocení SUS	27
5.2.	Stavba dotazníku BelbinAlt ...	28

Kapitola 1

Úvod

Dotazník je v dnešní době naprosto všední slovo, se kterým se každý už určitě setkal. Navíc s rychle rostoucím rozšířením internetu se jejich návrh a použití stává mnohem snadnější. Není tedy divu, že se dotazníky staly součástí mnoha oblastí, ve kterých můžou přinést velmi cenné informace. Příkladem mohou být průzkumy veřejného mínění, ze kterých vycházejí mnohé společnosti zabývající se statistikou, nebo také dotazníky zjišťující spokojenost zákazníků.

Tato práce se zabývá analýzou problematiky dotazníků a následným návrhem a implementací prototypu systému sloužícímu k jejich vytváření, vyplňování a vyhodnocování. Důraz je kladen hlavně na jednoduchost provozu a použití systému. V rámci této práce budou také v systému vytvořeny alespoň tři druhy dotazníků za účelem otestování prototypu.

1.1 Motivace

Systémů zabývajících se stejnou problematikou už bylo vyvinuto nespočet, mohlo by se tedy zdát, že by bylo lepší použít nějaké z již existujících řešení. Hlavní motivací pro práci na novém systému je zadavatelem popsaná specifická situace, ve které potřebuje sbírat informace pomocí dotazníků. Touto situací je plnění dotazníků na konferencích, festivalech a podobných sjezdech. Zadavatel by potřeboval mít možnost předem si připravit dotazníky, které pak jen předloží účastníkům pro vyplnění na mobilním zařízení. Navíc jeden z klíčových parametrů je možnost používání i uživateli bez hlubších IT znalostí. Také byl při návrhu kladen důraz na kompletní funkčnost systému bez přístupu k internetu.

Z této situace jsme vycházeli při analýze již existujících řešení, kdy jsme se pokoušeli nalézt systém splňující tyto potřeby, nebo obsahující alespoň nějaké součásti, kterými bychom se mohli inspirovat při návrhu našeho systému.

1.2 Cíle

Již od počátku bylo o práci uvažováno jako práci pro dva studenty, rozdělení na dvě aplikace se tedy zdálo být logické nejen z pohledu požadavků, ale také rozdělení práce. Tento návrh je nakonec použit i přesto, že na práci nakonec zbyl student jen jeden. Místo přepracování na jednu aplikaci se se tedy hlavním cílem stal vývoj prototypu těchto dvou aplikací, s tím že finalizace se případně provede mimo rámec této práce.

Prototyp tedy musí splňovat alespoň tyto minimální požadavky:

- Možnost vyplnění dotazníků na mobilních zařízeních bez připojení k internetu
- Možnost tvorby dotazníků předem, ideálně na stolním počítači
- Schopnost vytvoření dotazníků pro alespoň 3 různé scénáře použití
- Použitelnost uživateli bez hlubších IT znalostí

Kapitola 2

Zavedení do problematiky

Tato kapitola slouží k základnímu obeznámení se s dotazníky a s nimi spojenými tématy. Popisuje složení a typy dotazníků, jejich rozdíly a způsoby používání. Uvádí také několik příkladů využití dotazníků v praxi a některé systémy zabývající se problematikou tvorby a plnění dotazníků. Na závěr poukazuje na důsledky těchto informací na návrh našeho systému.

2.1 Rozdělení Dotazníků

Co je to vlastně dotazník? Člověk si obvykle pod tímto pojmem představí formulář v papírové nebo digitální formě, obsahující řadu otázek pro vyplnění. Otázkou ale zůstává, k čemu potom tyto vyplněné odpovědi slouží. Podle H. J. Adéra[1] se tento proces sběru odpovědí nazývá průzkum (nebo také anketa) a formuláře použité lze rozdělit podle účelu do dvou základních druhů, dotazníků a testů.

2.1.1 Testy

Testy jsou formuláře, které po vyplnění nabídnou uživateli zpětnou vazbu. Tato zpětná vazba musí být závislá na odpovědích od uživatele. Výsledky testů jsou obvykle ukládány jednotlivě, aby si každý uživatel mohl zobrazit svůj vlastní. Velkou výhodou testů je jednoznačnost výsledků. Testy mají přímo definované, kdy vyjde který výsledek a co pro uživatele znamená. Vše tedy záleží na tvorbě formuláře.

Testovat se mohou například uživateli znalosti (testy ve školách, autoškolách apod.), takové testy obvykle spočítají počet správných odpovědí a zobrazí výsledek. Jiné testy mohou pomoci uživateli rozhodnout se při výběru mezi alternativami, kdy uživatel vybírá své preference ve formě odpovědí, které jsou pak přiřazeny k těmto alternativám. Často se také můžeme setkat s takzvanými osobnostními testy, ty mají odpovědi nastaveny tak, aby korespondovali k určitým osobnostem a podle vybraných odpovědí pak zobrazí výsledek.

2.1.2 Dotazníky

Dotazníky na rozdíl od testů neposkytují žádnou přímou zpětnou vazbu uživateli, který je vyplnil. Místo toho si ukládá počet uživatelů, kteří odpověděli na každou otázku u každé odpovědi. Tyto data pak obvykle využívá autor tohoto dotazníku jako jeden celek. Hlavními poznatky z dotazníků tedy mohou být například nejčastější odpovědi, nejméně časté, poměr mezi odpověďmi, rozdíl oproti očekávání nebo minulým průzkumům a také korelace mezi odpověďmi u více otázek.

U dotazníků se ovšem vždy setkáme s problémem objektivnosti. George Beam[2] se například zabývá problémem výběru respondentů, způsobu sběru odpovědí ale i samotného obsahu dotazníků. Je podle něj třeba vždy pečlivě vybírat všechny tyto parametry a hlavně je také patřičně zdokumentovat. Jedna vybraná cílová skupina může přinést úplně jiné výsledky, než skupina jiná, a bez informace o této skupině jsou výsledky zavádějící a mohou přinést více škody, než užitku.

Výsledky dotazníků mohou být využity k mnoha cílům. Vývojáři pomocí nich mohou zjistit nedostatky svých aplikací, společnosti mohou zlepšit svůj zákaznický servis na základě odpovědí v dotazníku o spokojenosti. Na dotaznících jsou také často založeny články obsahující statistické údaje o respondentech.

2.2 Využití dotazníků a testů

V této sekci jsou podrobněji popsány některá využití dotazníků a testů. Cílem je ukázat různé typy dotazníků a testů a jejich odlišnosti ve stavbě, způsobu sběru a manipulaci s výsledky.

2.2.1 Průzkumy veřejného mínění

Cílem průzkumu veřejného mínění je zjištění preferencí a názorů nejčastěji v oblasti politiky. Provádějí se předkládáním dotazníků respondentům (ti jsou vybírání buď náhodně, nebo tak, aby rozložení jejich vlastností alespoň přibližně odpovídalo rozložení v celkové populaci) s otázkami, které mají obvykle předdefinované odpovědi.

Veřejnost se nejčastěji setkává s předvolebními průzkumy. Tyto průzkumy se pokoušejí předpovědět výsledky voleb na základě výsledků průzkumu jen malé části voličů (nejčastěji 1000 respondentů). Z těchto průzkumů často bývá neobjektivnost dotazníků zjevná. Kristýna Chábová [3] poukazuje na rozdíly mezi finálními výsledky voleb a těmito průzkumy, ale i rozdíly napříč průzkumy různých autorů. Jako příčiny považuje právě třeba rozdíly mezi vybranými respondenty, lišící se počty respondentů, ale také výběr a počet otázek v dotazníku.

2.2.2 System Usability Scale

System Usability Scale (SUS) ¹ je způsob měření míry použitelnosti vybraného systému. Půžitelnost systému se v tomto případě skládá z několika aspektů: efektivity systému, jeho výkonnosti a spokojenosti jeho uživatelů. Dotazník pro SUS se skládá z 10 následujících tvrzení:

- 1. Myslím, že bych chtěl často tuto aplikaci používat.
- 2. Aplikaci jsem shledal zbytečně komplikovanou.
- 3. Aplikaci jsem shledal snadnou k používání.
- 4. Myslím, že bych pro používání aplikace potřeboval technickou podporu.
- 5. Funkcionality systému jsem shledal jako dobře propojené.
- 6. Myslím, že je aplikace příliš nekonzistentní.
- 7. Většina lidí se nejspíše naučí aplikaci používat velmi rychle.
- 8. Aplikaci jsem shledal složitou na ovládání.
- 9. Při používání aplikace jsem se cítil sebejistě.
- 10. Před používáním aplikace jsem se musel naučit mnoho věcí.

K těmto tvrzením pak mají účastníci vyjádřit míru souhlasu, od silného nesouhlasu s tvrzením po silný souhlas. U lichých otázek znamená souhlas kladné hodnocení (od 4 silný souhlas, do 0 silný nesouhlas) a u sudých naopak (od 0 silný souhlas do 4 silný nesouhlas). Součet těchto hodnocení je pak vynásoben 2.5 a výsledná hodnota mezi 0-100 vyjadřuje míru použitelnosti z pohledu daného uživatele. Z těchto výsledků se pak obvykle vypočítává průměr, pro který platí, že hodnota 68 a více značí nadprůměrnou, 67 a níže podprůměrnou použitelnost.

Tyto výsledky se také dají použít při porovnávání použitelnosti 2 funkčně podobných systémů a případně rozlišení jejich hlavních problémů nebo výhod.

¹ <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>

2.2.3 Belbinův test rolí

Belbinův test rolí je typickým příkladem osobnostního testu. Test má celkem 7 výroků s osmi možnostmi, mezi které respondenti rozdělují přesně 12 bodů podle toho, jak moc je daná možnost vystihuje. Pro každý ze sedmi výroků je nutno rozdělit všech dvanáct bodů, a to v libovolném poměru mezi dané možnosti. Je tedy možné dát všech dvanáct bodů k jedné možnosti a u zbytku nechat nula, pokud má respondent pocit, že ta jediná možnost na něj zcela sedí a ostatní vůbec.

Vyhodnocení těchto testů pak probíhá následujícím způsobem: Každá možnost u každého výroku odpovídá jedné konkrétní roli. Tyto body pak přímo vyjadřují zastoupený poměr dané role. Z těchto poměrů pak lze vyčíst respondentova jedna nebo více dominantních rolí.

2.3 Existující programy

2.3.1 Formuláře google

Obrázek 2.1. Tvorba formuláře v aplikaci Formuláře Google

První zkoumanou aplikací jsou formuláře Google¹. Jsou součástí souboru Dokumenty Google od společnosti Google. Nabízí velmi rozmanitou nabídku druhů otázek a možností odpovědí. K formulářům i otázkám zvláště lze také přidávat různá média, jako jsou zvukové stopy, obrázky nebo videa.

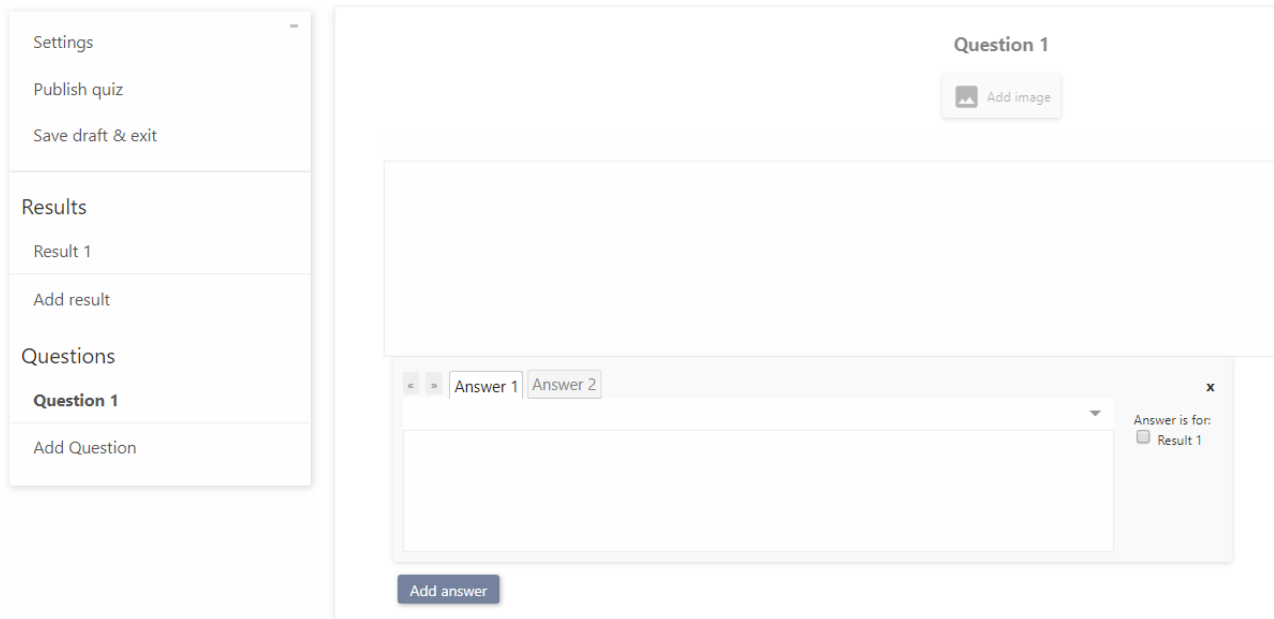
Aplikace je navíc velmi jednoduchá k používání, graficky příjemná, a protože je součástí Dokumentů Google pro používání stačí mít přístup k internetu a založený uživatelský účet Google. Hlavním účelem je tvorba dotazníků, v aplikaci nelze nastavit

¹ <https://www.google.com/forms/about/>

zpětnou vazbu jako výsledek, tvorba testů tedy není možná. Také nelze ukládat odpovědi jednotlivých uživatelů. Výsledky se ukládají pouze jako počet respondentů, kteří označili danou odpověď.

Výsledky si může zobrazit pouze autor dotazníku, respondentovi je po vyplnění zobrazeno pouze poděkování za vyplnění a možnost vyplnit znovu. Tvorba dotazníků, zobrazování výsledků i odpovídání probíhá v prohlížeči, není tedy možné ani jednu z těchto činností provádět bez připojení k internetu.

2.3.2 Quotev



Obrázek 2.2. Tvorba formuláře v aplikaci Quotev

Aplikace Quotev¹ slouží především pro tvorbu a vyplňování různých testů a kvízů online. Na Obrázku 2.2. můžeme vidět okno pro tvorbu nového testu, ve kterém se postupně přidávají možné výsledky, otázky a odpovědi na ně. Každé otázce lze přidat obrázek, a při přidávání odpovědí je potřeba vybrat, ke kterému z výsledků daná odpověď patří.

Respondentovi se pak při vyplňování zobrazuje jedna otázka za druhou s případným přidaným obrázkem, u kterých vybírá odpovědi podle toho, jak byli při tvoření zadány. Při ukončení plnění se pak jednoduše sečtou odpovědi patřící ke každému výsledku a ten s nejvyšším počtem je pak představen uživateli jako výsledek testu, i s hodnotami přiřazenými ke každému výsledku. Také jsou zobrazeny počty uživatelů, kteří dposud dosáhli všech výsledků, které test nabízí pro porovnání vlastního výsledku s výsledky ostatních respondentů.

V Quotevu lze také tvořit dotazníky sčítající počty odpovědí. Každému se po vyplnění zobrazí současné hodnoty u každé odpovědi. Autor si může tyto hodnoty zobrazit sám kdykoliv. Testy zaznamenávají pouze počet uživatelů, kteří došli ke každému z výsledků, výsledky samotné se neukládají. Aplikace běží celá v prohlížeči, je tedy třeba stále připojení k internetu.

¹ <https://www.quotev.com/>

2.3.3 QuickTapSurvey

Obrázek 2.3. Vyplnění formuláře v aplikaci QuickTapSurvey

QuickTapSurvey¹ nabízí oproti předchozím aplikacím navíc funkcionalitu založení dotazníku na stolním počítači a vyplněním zvláště na mobilním zařízení, včetně vyplňování offline. Lze také vybírat z mnoha druhů otázek, více způsobů jejich zobrazení a dokonce i využití šablon, na jejichž základě se založí nový dotazník.

Bohužel se specializuje pouze na dotazníky, testy vytvářet nelze. Výsledky se opět ukládají pouze jako součty. Hlavní nevýhodou je ale zpoplatnění: v trial verzi QuickTapSurvey lze využít pouze jedno mobilní zařízení a navíc pouze pro 10 respondentů. Placené verze navíc účtují navíc za každé použité zařízení.

2.3.4 Survio

Obrázek 2.4. Přidání otázky v aplikaci survio

¹ <https://www.quicktapsurvey.com/>

Poslední aplikací kterou zde uvedeme je Survio¹. Survio je další aplikací, která nabízí mnoho typů otázek, odpovědí a dalších nastavení (například ukončení dotazníku po uplynulém čase). Sestavení dotazníků je opět velmi jednoduché a rychlé. Trial verze je navíc mírnější než u jiných aplikací, největším omezením je 100 odpovědí měsíčně. Opět ale umožňuje tvorbu pouze dotazníků a vyžaduje připojení k internetu po celou dobu zacházení s aplikací.

2.4 Shrnutí

Jak je snad z této sekce patrné, problematika dotazníků je velmi rozmanitá. Výše uvedené využití a aplikace jsou pouze jedny z mnoha, tyto vybrané slouží právě pro ilustraci všech těchto různých možností, které se uživatelům nabízejí. Tato rozmanitost vede k tomu, že většina systémů je pro nás příliš obecná, nebo příliš složitá. Navíc takto velké aplikace jsou často placené. Také mezi aplikacemi převažují ty webové, které nesplňují požadavek na funkčnost bez internetu a tím je možnost jejich použití vyřazena.

Pro prototyp naší aplikace si z této sekce můžeme odnést alespoň způsob sestavování dotazníků: jednotlivé aplikace se v tomto ohledu liší jen málo přidáváním výsledků a otázek jednu po druhé do stavby dotazníku vyhovuje i pro naše potřeby.

¹ <https://www.survio.com/>

Kapitola 3

Návrh

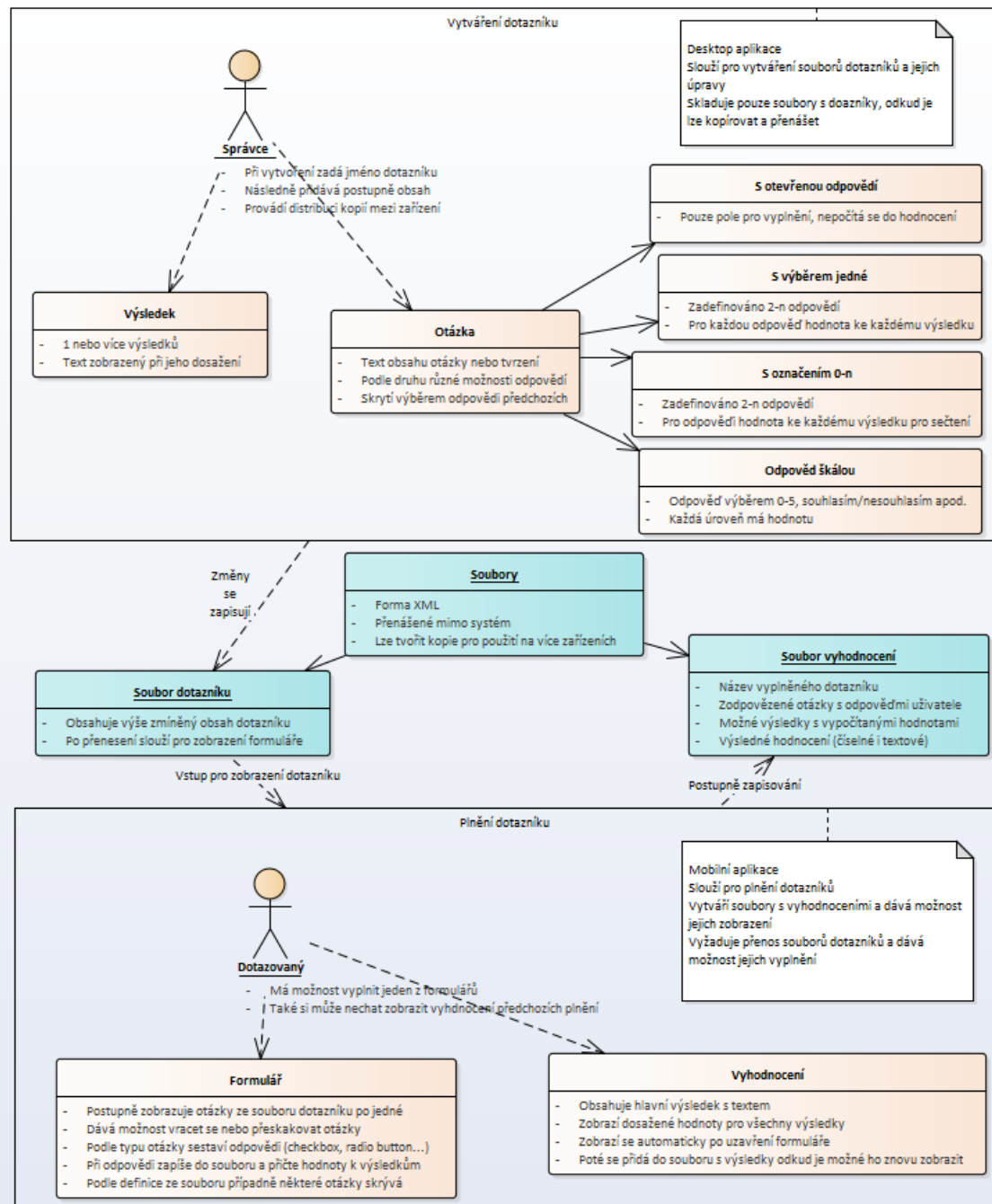
Tato kapitola popisuje plánování výsledné implementace prototypu. Soustředí se především na to, v jakém stavu by měl být prototyp po dokončení této práce. Obsahem jsou požadavky na funkcionalitu programu a také několik diagramů popisujících jeho stavbu a použití. Pro vytváření těchto diagramů byl použit nástroj Enterprise Architect¹, který umožňuje využití prvků UML pro sestavení diagramů.

Unified Modeling Language (UML) je jazyk sloužící pro vizualizaci, specifikaci, návrh a dokumentaci prvků softwarových systémů. UML lze použít pro naprostou většinu systémů různých velikostí, účelů nově vyvíjených i dlouho zaběhlých. Tento jazyk je velmi expresivní a obsáhlý, je ale také považován za přehledný a jednoduchý pro použití, stal se tak pro jeho účely naprosto dominantním. Popis UML byl volně přeložen z oficiální uživatelské příručky[4].

Jak už bylo zmíněno, systém je rozdělen do dvou částí (tvorba a vyplňování), v následujících sekcích tedy bude uvedeno která část má jakou funkcionalitu na starosti, často zvlášť popis každé z nich, nebo budou alespoň porovnány.

¹ <http://sparxsystems.com/products/ea/>

3.1 Předběžný návrh



Obrázek 3.1. Předběžný návrh funkcionality

Na počátku návrhu byl sestaven diagram (obrázek 3.1.), který sice využívá prvky UML, ale není žádným standardním diagramem. Sloužil hlavně k prvnímu pohledu na rozdělení systému na dvě části. Ukazuje hlavní stavbu každé části a informativně popisuje jednotlivé položky. Z tohoto diagramu se pak vycházelo při definici požadavků na systém a tvorbě diagramů, které už přesněji popisují implementované části systému.

3.2 Požadavky

Požadavky na systém popisují co má daný systém dělat: jaké služby poskytuje a jaká jsou omezení jeho činností. Odrážejí potřeby uživatelů na systém, který bude sloužit k určitému účelu (převzato z knihy Softwarové inženýrství[5]).

3.3 Funkční požadavky

Funkční požadavky uvádějí možnosti uživatele při používání prototypu a jejich stručný popis.

3.3.1 Tvorba dotazníků

Funkcionality části systému pro tvorbu dotazníků.

- **Založení dotazníku:** Systém bude umožňovat zakládání nových dotazníků a jejich zápis do souborů.
- **Editace dotazníku:** Systém bude umožňovat načtení již vytvořených dotazníků a provádění změn.
- **Smazání dotazník:** Systém bude umožňovat odstranění dotazníků včetně jejich souborů.
- **Práce s výsledky:** Systém bude umožňovat přidávání, odebrání a editaci výsledků dotazníků nových i existujících.
- **Práce s otázkami:** Systém bude umožňovat přidávání, odebrání a editaci otázek alepoň těchto tří typů: S otevřenou odpovědí, s možností výběru maximálně jedné, s možností neomezeného výběru.
- **Umístění otázek:** Systém bude umožňovat výběr pořadí otázek v dotazníku, ve kterém se budou při vyplňování zobrazovat.
- **Práce s odpověďmi:** Systém bude umožňovat přidávání, odebrání a editaci odpovědí k otázkám podle jejich typu, včetně přiřazení k výsledkům pro vyhodnocení.

3.3.2 Plnění dotazníků

Funkcionality části systému pro vyplňování dotazníků.

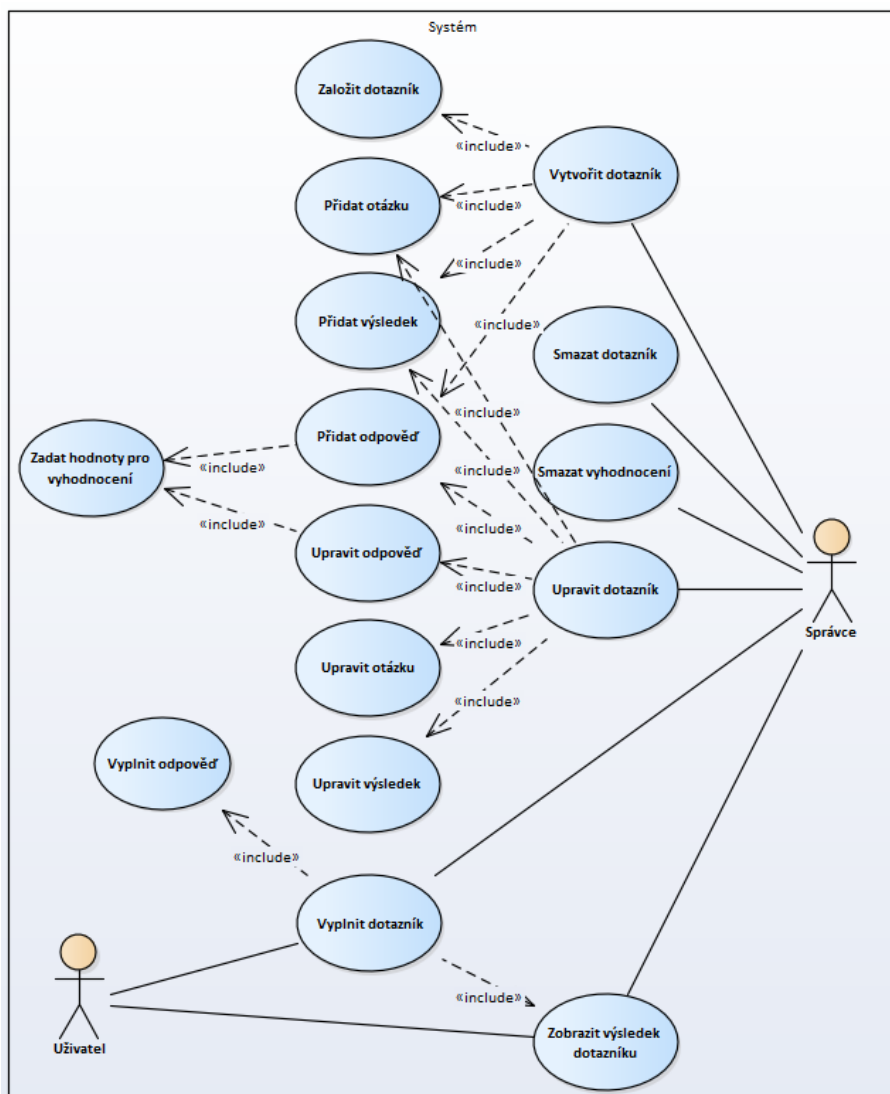
- **Zobrazení dotazníků:** Systém bude umožňovat zobrazení nabídky dotazníků dostupných na zařízení.
- **Vyplnění dotazníku:** Systém bude umožňovat výběr dotazníku pro vyplnění a zobrazení otázek.
- **Odpovídání na otázky:** Systém bude umožňovat vyplnění odpovědi podle typu otázky a uložení těchto odpovědí, včetně možnosti provádění úprav v již zodpovězených otázkách.
- **Vyhodnocení:** Systém bude umožňovat vyhodnocení vyplnění dotazníků, včetně jeho uložení a zobrazení tohoto vyhodnocení.
- **Zobrazení vyhodnocení:** Systém bude umožňovat zobrazení minulých vyhodnocení dotazníků.

3.4 Nefunkční požadavky

Nefunkční požadavky (někdy také mimofunkční) se přímo netýkají poskytovaných služeb. Mohou se vztahovat k samotným vlastnostem systému, nebo definovat určité omezení na jeho implementaci.

- **Čitelnost souborů:** Dotazníky a vyhodnocení musí být ukládány v lidsky čitelné formě a na pro uživatele dostupných místech na disku.
- **Grafické rozhraní:** Systém poskytne uživatelům grafické rozhraní pro usnadnění jeho používání.
- **Platforma tvorby:** Systém bude umožňovat tvorbu dotazníků na současně podporovaných verzích OS Windows.
- **Platforma plnění:** Systém bude umožňovat tvorbu dotazníků na současně podporovaných verzích OS Android.
- **Programovací jazyk:** Systém bude realizován v jazyce Java, s využitím značkovacího jazyka XML.

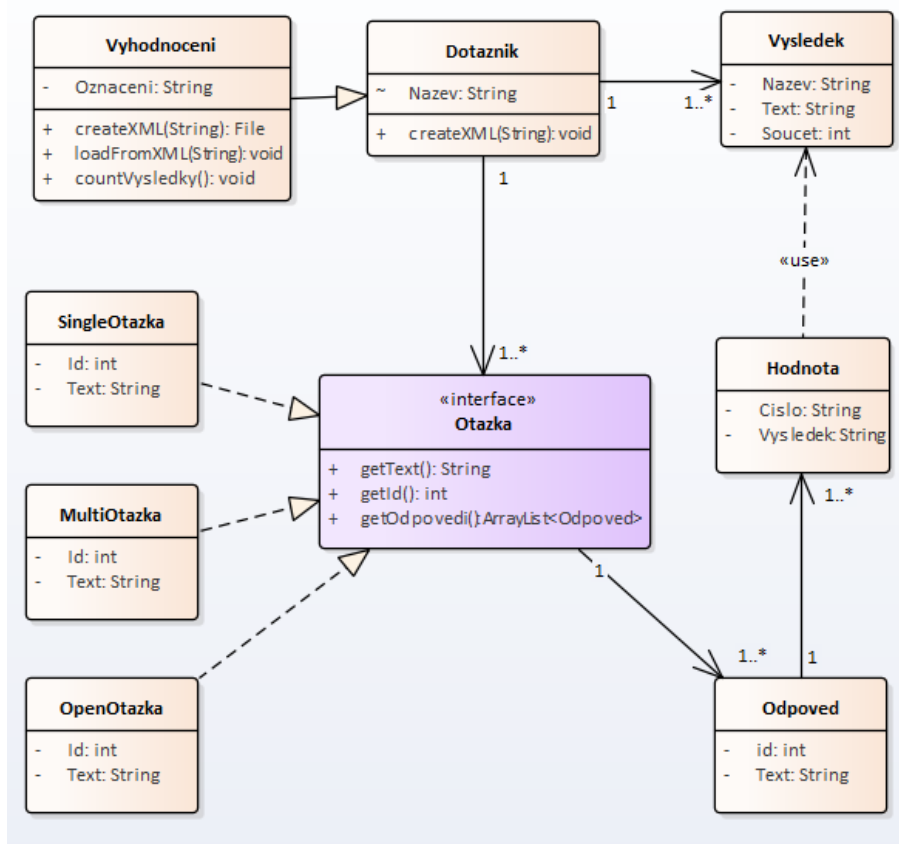
3.5 Případy užití



Obrázek 3.2. Diagram případu užití

Diagram případů užití slouží k zachycení funkčních požadavků do grafické formy. Uvádí veškeré činnosti, které jsou uživatelům (aktérům) v systému dostupné.

3.6 Diagram tříd



Obrázek 3.3. Diagram tříd struktury

Diagram tříd na obrázku 3.3 popisuje objekty v prototypu aplikace pro plnění dotazníků, které odpovídají obvyklé stavbě dotazníků a jejich součástí. Obsahem jsou také atributy a důležité metody v těchto třídách. Na základě diagramu předběžného návrhu (3.1) byly navrhunty tyto třídy, které lépe reprezentují dané objekty a slouží k popisu jejich vlastní implementace.

Na vrcholu je třída *Dotaznik*, reprezentující dotazník samotný. Udrží si název a také seznamy výsledků a otázek. Které do něj byli přidány. Třída *Vyhodnoceni* má velmi podobnou stavbu, přidává si navíc jen záznam *Oznaceni*, který slouží k identifikaci uživatele, kterému toto vyhodnocení odpovídá. Umí se také načíst ze souboru dotazníku pomocí metody *loadFromXML*, která vyžaduje zadání cesty k dotazníku parametru *path* a nastaví objekt tak, aby odpovídal plněnému dotazníku.

Třídy implementující rozhraní *Otazka* reprezentují druhy otázek, které lze do dotazníku přidat. Oproti diagramu 3.1 zde chybí otázka škálová, implementačně je totiž shodná s otázkou s 1 odpovědí (u obou má uživatel výběr, kde může vybrat jen jednu možnost). Tyto otázky tedy kromě identifikačního čísla a textu otázek obsahují seznam možných odpovědí. U otázky s otevřenou odpovědí je obvykle před plněním tento seznam prázdný, po vyplnění obsahuje jednu instanci třídy *Odpoved*.

Třída *Odpoved* má opět identifikaci a vlastní text odpovědi, navíc je zde přidána třída *Hodnota*. Každá odpověď může mít až tolik přiřazených objektů třídy *Hodnota*, kolik je v dotazníku výsledků. Tyto objekty pak jednoduše obsahují název výsledku a

číselnou hodnotu, kterou tomuto výsledku přiřazují. Při výběru těchto odpovědí jsou tyto objekty přidány do objektu Vyhodnoceni tak, aby obsahoval právě jen ty označené.

Konečně třída Vysledek obsahuje název výsledku, popis zobrazený uživateli v případě, že k tomuto výsledku dojde a součet hodnot. Tento součet je vypočten při ukončení plnění dotazíků voláním metody countVysledky ve třídě Vyhodnoceni, která projde všechny odpovědi u všech otázek, u každé z nich vezme list hodnot a přičítá tyto hodnoty do odpovídající instance třídy Vysledek.

Stavba v části aplikace pro tvorbu dotazníků je velmi podobná, hlavním rozdílem je chybějící třída Vyhodnoceni, která není v této části zapotřebí.

3.7 Soubor dotazníku

Soubor dotazníku má být čitelný jak pro člověka, tak i jednoduše pro obě aplikace. Proto byl zvolen pro jejich tvorbu značkovací jazyk Extensible Markup Language (XML), jehož jedním z hlavních účelů je právě tato vlastnost. Způsob práce s těmito soubory bude detailněji popsán v další kapitole a v příloze je možné najít několik těchto souborů.

Níže je uvedeno základní rozložení značek (tagů), které odpovídají stavbě v diagramu tříd (obrázek 3.3). Vyhodnoceni má stavbu identickou až na hlavní značku, kde se místo `<Dotaznik>` nachází `<Vyhodnoceni>`.

Dotazníky, výsledky, otázky a odpovědi mají každý jeden atribut, který slouží k jednoduchému označení, zbytek struktury je složen z vnořených elementů, tak jak i reprezentované třídy obsahují seznamy. Některé elementy, jako například Text vnořený v odpověď, může být přítomen jen jednou uvnitř rodičovského elementu, jiné i vícekrát, podle toho, jestli daný element odpovídá položce seznamu, nebo jen jednoduchému třídnímu atributu.

```

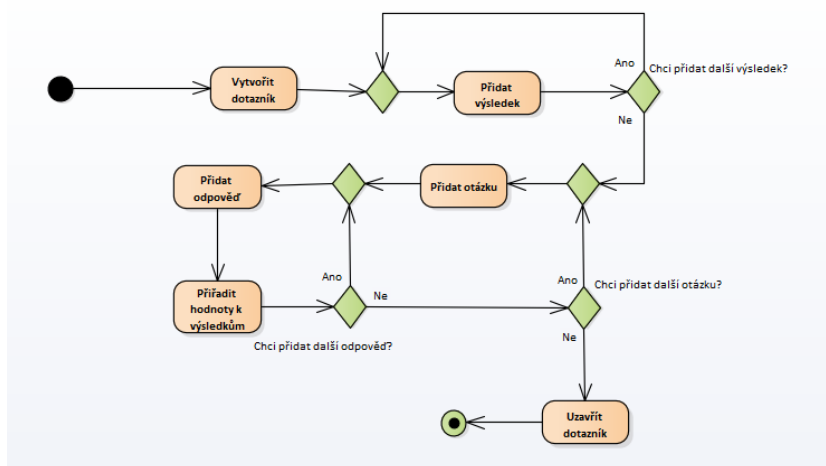
<Dotaznik Nazev="Název dotazníku">
  <vysledek Nazev="Název výsledku">
    <Text>
      Popis výsledku
    </Text>
  </vysledek>
  <typOtazka id="id">
    <Text>
      Text otázky
    </Text>
    <odpoved id="id">
      <Text>
        Text Odpovědi
      </Text>
      <hodnota>
        <Cislo>
          hodnota odpovědi
        </Cislo>
      </Vysledek>
    </typOtazka>
  </Dotaznik>

```

3.8 Diagramy aktivit

Diagramy aktivit slouží k definici postupu uživatelů při provádění nějaké činnosti se systémem. Obvykle obsahují začátek a konec činnosti s určitým množstvím aktivit a rozhodnutí, které jsou její součástí. Níže jsou takto popsány aktivity při založení dotazníku a při jeho plnění, jakožto hlavní využití systému.

3.8.1 Založení dotazníku

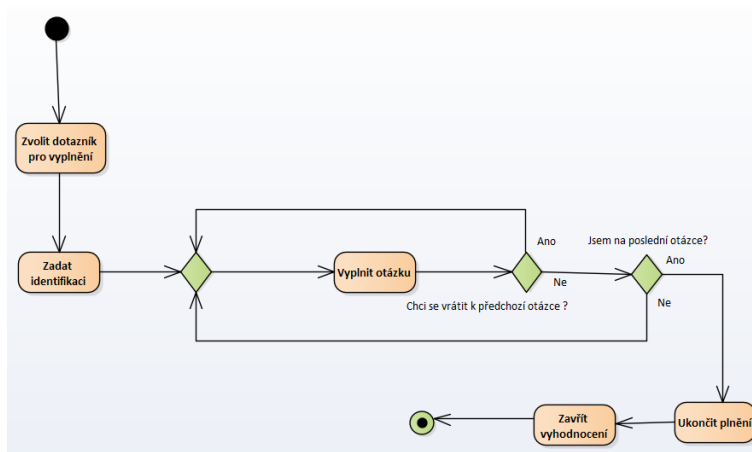


Obrázek 3.4. Diagram aktivity založení dotazníku

Uživatel na začátek vytvoří dotazník a začne do něj postupně přidávat jeho součásti. Přidání výsledků na začátek je lepší (později při přidávání hodnot k odpovědím budou již zobrazeny, kdyby byly vytvořeny nejdříve tyto odpovědi, musel by se k nim autor vracet), ale je uživateli umožněno přidávat i otázky a odpovědi nejdříve. Po dokončení vytváření všech součástí dotazníku ho uživatel uzavře a soubor je uložen.

Uživateli je také samozřejmě umožněno upravovat minulé dotazníky, a také měnit či mazat každou z již vytvořených souborů. Diagramy pro tyto postupy by byly téměř identické, je tu pro jednoduchost uveden jen tento.

3.8.2 Vyplnění dotazníku

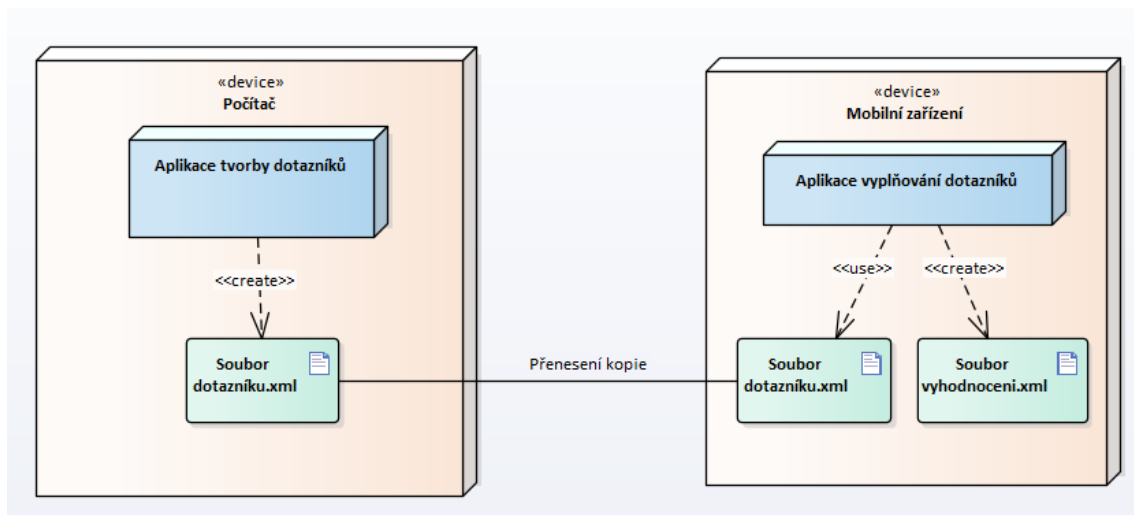


Obrázek 3.5. Diagram aktivity vyplnění dotazníku

Uživateli je zobrazen seznam dotazníků v zařízení, které může vyplňovat. Výběrem jednoho z nich je vyzván k potvrzení, že chce začít vyplňovat a poté zadání nějaké identifikační informace o sobě (jméno, email apod.). Po zadání této informace se mu zobrazí první otázka, kterou může vyplnit. Postupně se propracuje otázkami (má možnost se případně vracet a upravovat odpovědi) a nakonec plnění uzavře. Po uzavření se mu zobrazí jeho vyhodnocení, které si může přečíst a pak se vrátit zpět do zobrazení dotazníků.

Další užití této části je zobrazení minulých vyhodnocení. Zde stačí po vybrání dotazníku místo vyplňování vybrat možnost zobrazení vyhodnocení, kde jsou ty, co vznikli vyplněním vybraného dotazníku, uvedeny. Tato činnost je triviální a tak nemělo smysl ji znázorňovat modelem.

3.9 Diagram nasazení



Obrázek 3.6. Diagram nasazení

Diagram nasazení slouží k zobrazení fyzického nasazení systému na jednotlivá zařízení. Pro nás hlavně slouží pro zobrazení skutečnosti, že oba prototypy jsou nasazeny zvlášť na jiných zařízeních a nijak navzájem nekomunikují. Aplikace pro tvorbu vytváří soubory, které si uživatel sám musí přenést do zařízení s aplikací pro plnění. Způsob přenosu je tedy na volbě uživatele. Aplikace pro plnění pak přenesené soubory využívá pro načtení dotazníku a vytváří vlastní soubory pro vyhodnocení.

Kapitola 4

Implementace

Tématem této kapitoly je popis implementace systému. Uvede některé technologie použité při vývoji, porovná některé funkcionality mezi oběma částmi a poukáže na specifika každé z těchto částí. Především pak provede grafickým rozhráním obou prototypů, čímž ukáže, jak ji lze použít z pohledu uživatele.

4.1 Specifikace implementace

Jak již bylo uvedeno dříve, systém byl implementován jako dva zvlášť vyvíjené prototypy aplikací. Pro obě části bylo využito programovacího jazyku Java především proto, že k němu má autor nejbližší. Obě součásti jsou navzájem nezávislé, obě byly vyvíjené jako nativní na svůj cílový operační systém (Windows pro tvorbu, Android pro plnění dotazníků). Pro kompletní využití funkcionality systému jsou ale potřeba obě části. Soubory dotazníků, které uživatel vytvoří v části pro tvorbu jsou jejím hlavním a v podstatě jediným účelem a samotné nemají žádný užitek. Tyto soubory jsou nutné pro funkčnost v části druhé, která tyto soubory načte a na základě uživatelských odpovědí tvoří soubory s výsledky. Bez jediného souboru s dotazníkem je tedy nepoužitelná.

4.1.1 Tvorba a čtení souboru

Tvorba a čtení souboru jsou nejdůležitější částí systému, je tedy vhodné blížeji popsat, jak se s tímto úkolem systém vypořádává. Jedním z hlavních důvodů, proč je objektově orientované programování tak populární je právě možnost reprezentace reálných objektů pomocí tříd a objektů v kódu. Právě na tom je i založena práce s dotazníky v našich prototypy: Při vytváření dotazníku se v aplikaci vytvoří objekt, který ho reprezentuje. Každý výsledek, každá otázka, odpověď i hodnota, kterou uživatel přidá v grafickém rozhraní se také vytvoří jako instance příslušné třídy a přidá se do kompozice hlavního objektu dokumentu. Při tvoření dotazníku tedy má program k dispozici objekt, který přesně odpovídá dotazníku, který se má zapsat na soubor.

K serializaci těchto objektů do zapsatelné formy byly využity dvě rozhraní, zvlášť na jiných operačních systémech. Na Windows to je JAXB¹ a na Android Simple². Obě tyto API fungují velmi podobně: nabízejí třídy, které na základě XML tagů ve třídách jejich objekty převedou do UML tagů v souborech. Pro naši funkcionality bylo tedy potřeba pouze doplnit patřičné XML tagy k součástem dotazníku tak, aby výsledný soubor odpovídal návrhu. Příkladem takové třídy pro zacházení se Simple API může být třída dotazník. V kódu je vidět i metoda CreateXML, která ukazuje, jak je v Simple API jednoduché takto označené objekty serializovat. API JAXB je funkčně téměř identické, ale použité tagy se trochu liší a třída pro serializaci je také samozřejmě jiná. Dalším rozdílem je umístění tagů: v Simple jsou potřeba nad deklarací atribut, JAXB nad jejich gettery.

¹ <http://www.oracle.com/technetwork/articles/javase/index-140168.html>

² <http://simple.sourceforge.net/>


```

@Root(name="Dotaznik")
public class Dotaznik {
    @Attribute
    String Nazev;

    @ElementList(inline=true)
    ArrayList<Vysledek> vysledky = new ArrayList<>();

    @ElementListUnion
    ({
    @ElementList(inline = true, type = MultiOtazka.class, required=false),
    @ElementList(inline = true, type = SingleOtazka.class, required=false),
    @ElementList(inline = true, type = OpenOtazka.class, required=false)
    })
    ArrayList<Otazka> otazky = new ArrayList<>();

    public void CreateXML(File Path) throws Exception{
        Serializer serializer = new Persister();
        File result = new File(Path + "/" + getNazev()+".xml");
        serializer.write(this, result);
    }
}

```

4.1.2 Udržování dotazníku

V předchozí sekci je ukázáno, že pro funkčnost aplikace je nejdůležitější existence a korektnost objektu reprezentujícího dotazník. Otázkou tedy bylo, jak to v obou částech systému zajistit. V aplikaci tvoření to lze velmi jednoduše: tuto instanci dotazníku si vždy drží současný *Frame* (okno). K němu pak přistupují funkce které na něm provádí změny, nebo ho potřebují číst. Při změně okna je pak tato instance předána a formou přetíženého konstruktora uložena v dalším *Frame* hned při jeho vytvoření.

Na Android je situace mírně složitější, k přenosu informací mezi okny (reprezentovány třídami implementujícími *Activity*) a přechodu mezi nimi slouží takzvané *Intenty*. Ty ale obvykle nabízejí přenos pouze primitivních a několika základních datových typů. Při přechodu mezi okny by tedy instance dotazníku byla ztracena. Řešením tohoto problému se stal návrhový vzor Singleton, zde přesněji ve formě rozšíření třídy *Application*. Třída vytvořená podle vzoru Singleton je omezená tak, že lze vytvořit pouze jednu instanci této třídy. Tak je právě implementována i třída *Application*. Objekt této třídy existuje po celou dobu běhu programu, a protože je typu singleton, máme zajištěno že v něm uložené informace tam zůstanou, dokud je nezměníme.

```

public class MyApplication extends Application {
    private Vyhodnoceni vyhodnoceni;
    private Dotaznik dotaznik;
    private int soucasnaOtazka;
    private SparseBooleanArray[] odpovedi;
}

```

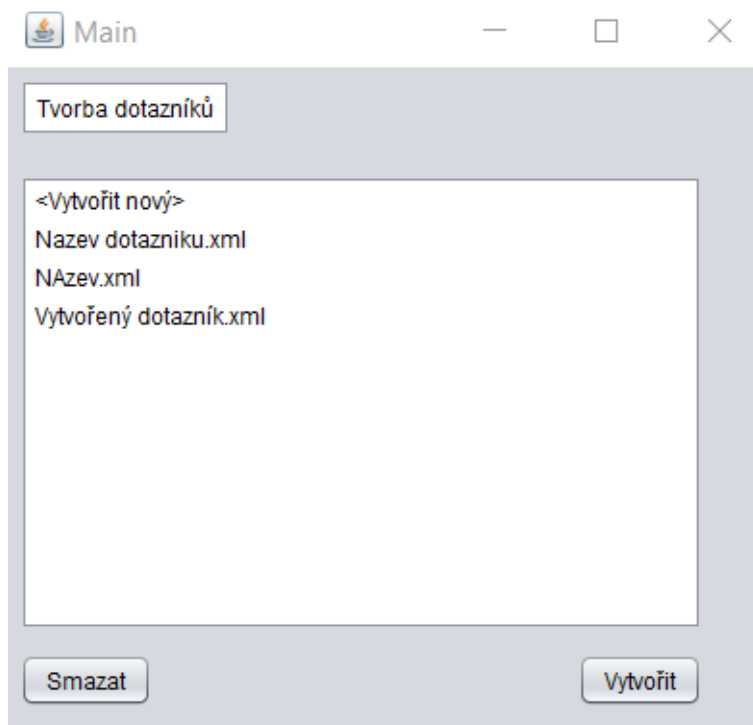
Všechny třídy implementující *Activity* mají přístup k metodě *getApplication*, která vrací právě tuto instanci. Při potřebě přístupu k těmto datům tedy stačí zavolat tuto funkci a k atributům přistoupit pomocí getterů a setterů.

4.2 Prototyp aplikace pro vytváření dotazníků

Vytváření tohoto prototypu proběhlo v prostředí NetBeans IDE¹, a i pro vytváření GUI bylo použito v něm zabudované funkcionality *GUI Builder*. Každé okno je tedy vlastní třídou rozšiřující třídu `javax.swing.JFrame`. První dvě okna jsou sestavena za použití třídy rozšiřující `swing.JPanel`, další, jednodušší okna mají své prvky definovány rovnou v `JFrame`. V oknech se tedy nachází základní prvky knihovny swing, jako jsou `JButton` (tlačítko), `JList` (seznam s možností výběru), `JTextArea` (pole pro uživatelský textový vstup) a další. Také je na místech, kde uživatel vyplňuje textová pole ochrana proti špatnému vyplnění (v našem případě se jedná jen o prázdná pole, a nečíselný znak při zadávání čísla v hodnotě) vyvoláním dialogu `JOptionPane`. Na uživateli činnosti obvykle reagují funkce jako je `JButtonActionPerformed`, ve které je zadefinována činnost systému při akci uživatele.

4.2.1 Hlavní okno

Po spuštění prototypu je zobrazeno hlavní okno, které uživateli zobrazí seznam již existujících dotazníků a umožňuje mu jeden z nich vybrat a upravit (nebo smazat), nebo založit nový pomocí tlačítek pod seznamem. Tím se přesune do okna dotazníku. Tento způsob výběru jednoho pro úpravy, nebo založení nového se objevuje v prototypu znovu pro každou potenciální instanci, se kterou je potřeba pracovat, aby byla aplikace v tomto ohledu konzistentní.



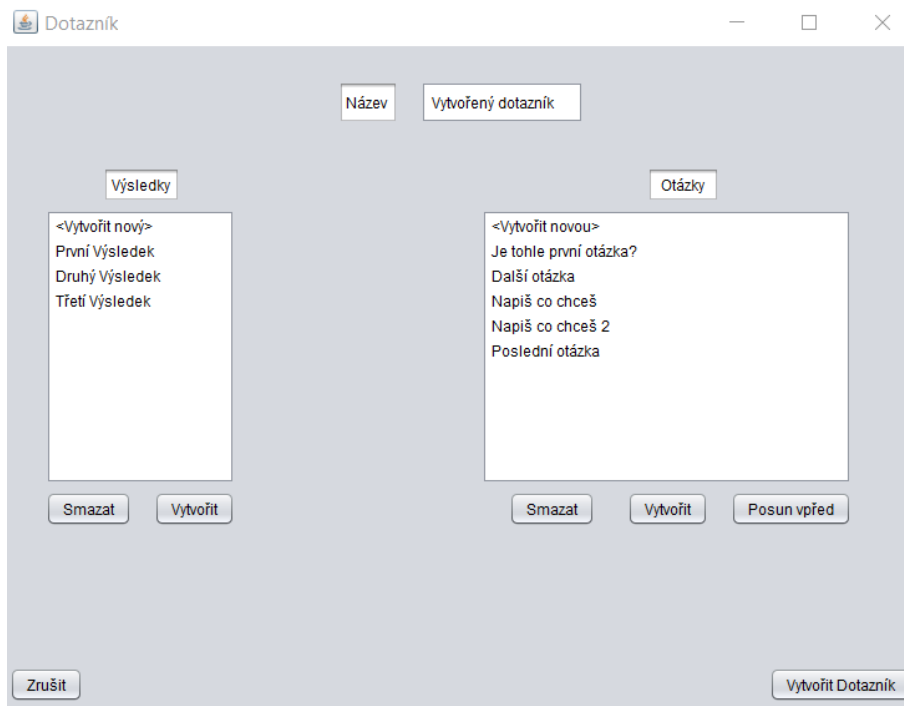
Obrázek 4.1. Hlavní okno

4.2.2 Okno dotazníku

Toto okno je uživateli zobrazeno po výběru dotazníku pro úpravy, nebo založení nového. Nabízí zadání názvu dotazníku (podle něj také bude pojmenován soubor), a práci

¹ <https://netbeans.org/>

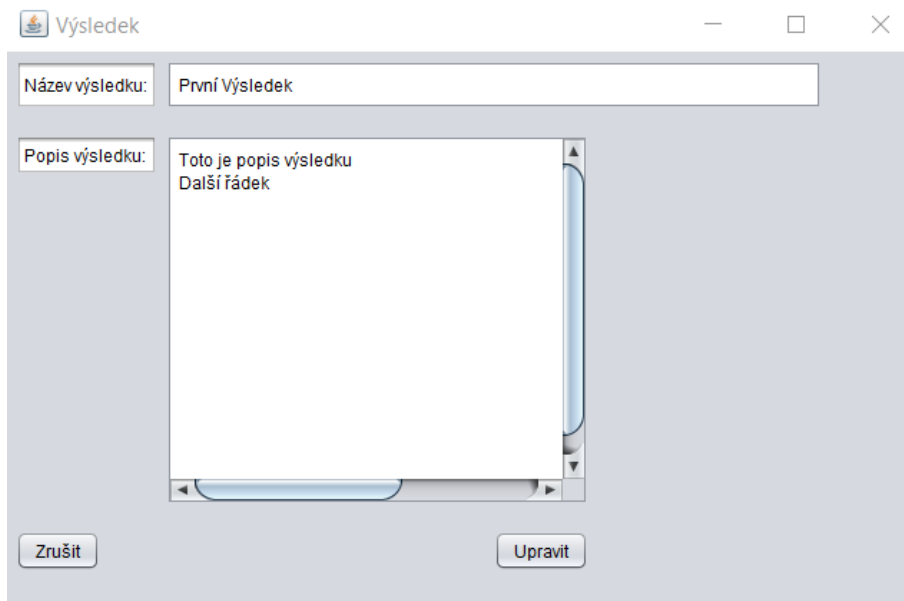
s výsledky a otázkami. U obou může opět vybrat položku již vytvořenou za účelem mazání nebo úprav, nebo založí položku novou. Zakládání a úpravy přesunou uživatele do příslušného okna. U otázek má navíc možnost pomocí tlačítka *Posun vpřed* posunout vybranou otázku dopředu, čímž může efektivně vybrat celkové pořadí všech otázek. Ko- nečně může také na této obrazovce zrušit svojí tvorbu, čímž zahodí všechny provedené změny, nebo potvrdit vytvoření, čímž se současná instance zapíše do souboru (způsob byl popsán v sekci 4.1.1.).



Obrázek 4.2. Okno dotazníku

■ 4.2.3 Okno výsledku

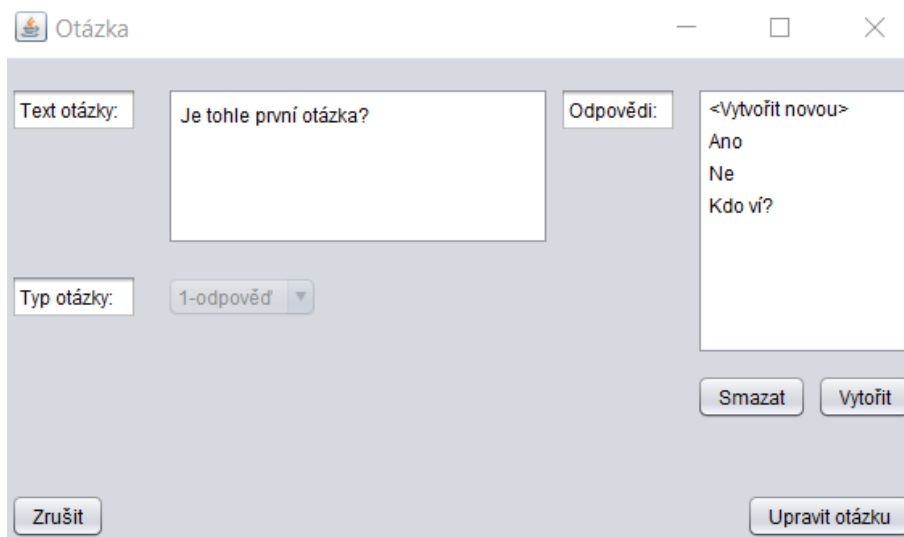
Okno výsledku je velmi jednoduché, uživatel pouze vyplní obě textová pole, jedno pro název výsledku a druhé pro jeho popis. Potvrzením se vytvoří instance třídy *Vysledek* se zadanými parametry, přidá se do seznamu v dotazníku a vrátí uživatele do okna dotazníku. Tlačítko zrušit tuto tvorbu přeskočí.



Obrázek 4.3. Okno výsledku

4.2.4 Okno otázky

V okně otázky uživatel vytváří instanci jedné z tříd implementující rozhraní Otazka. To se vybere pomocí JComboBoxu, kde se dá vybrat právě jedna z těchto tří možností typu otázek. Po výběru je instance vytvořena a typ už se nedá změnit. Dále je zde možné zadat text této otázky a pracovat s odpověďmi. V případě, že byla založena otázka s otevřenou odpovědí, sekce pro práci s odpověďmi je uživateli nepřístupná. V druhých dvou případech zde může uživatel jako obvykle vybrat jednu z jich založených, smazat ji nebo upravit, nebo založit zcela novou. Zakládání a úpravy opět probíhají ve vlastním okně.

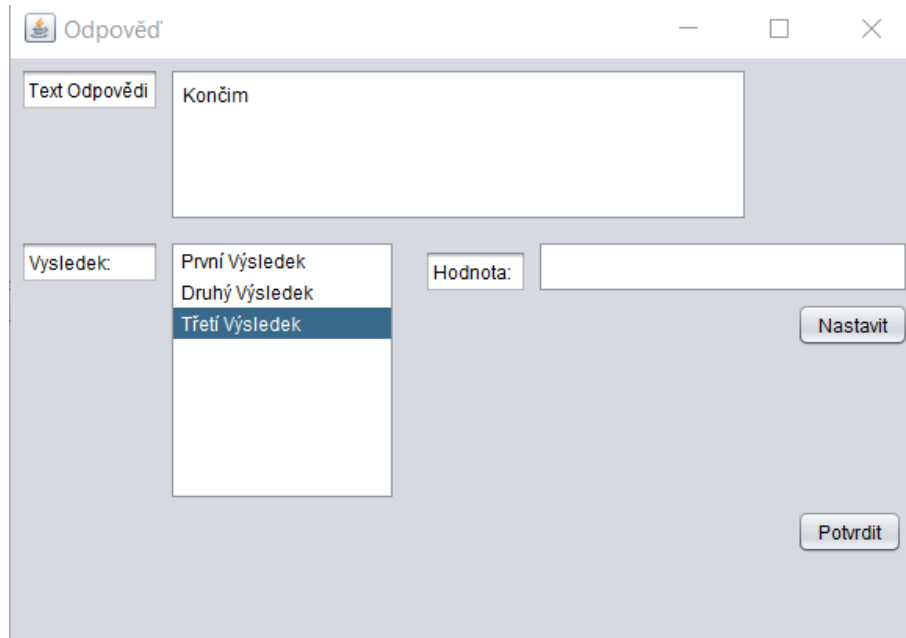


Obrázek 4.4. Okno otázky

4.2.5 Okno odpovědi

V okně odpovědi musí pro vytvoření příslušného objektu uživatel zadat text odpovědi a pak případně přidat hodnoty k výsledkům. To provede tak, že vybere výsledek, ke

kterému chce hodnotu přidat, číselnou hodnotu zadá do textového pole a tlačítkem potvrdí a vytvoří objekt *Hodnota*. Výchozí hodnota je pro každý výsledek 0, v takovém případě není nutné ani tuto hodnotu zadávat. Potvrzením se tento objekt přidá do příslušné otázky a je znovu zobrazeno okno otázky.



Obrázek 4.5. Okno odpovědi

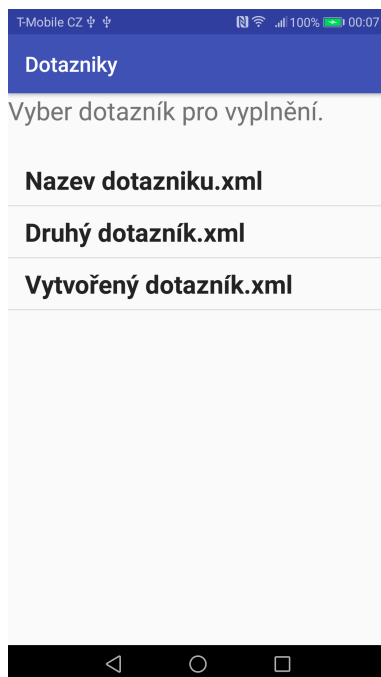
4.3 Prototyp aplikace vyplňování dotazníků

Programování prototypu na android proběhlo v IDE Android Studio ¹ a stejně jako v první části, pro sestavení vzhledu GUI bylo použito zabudované funkcionality, zde pojmenované *Layout Editor*. GUI na android vyžaduje trochu složitější strukturu než GUI desktopové. V *Layout Editoru* jsou vytvořené xml soubory popisující jednotlivé prvky v okně. Tento xml soubor je pak předán třídě dědicí z třídy *Activity*. V těchto třídách se definuje metoda *onCreate*, která je zavolána, když se má příslušné okno vytvořit. V této metodě se tedy nastaví layout na vytvořený xml soubor a případně může dále upravovat prvky v něm definované. Nejzajímavějším použitým prvkem je *RecyclerView*, který zobrazuje nějaký seznam položek. Tento seznam musí být zpracován třídou *Adapter*, která převádí seznamy na *ViewHoldery* (ty opět využijí xml layout, který musí programátor předem vytvořit), které už je *RecyclerView* schopen zobrazit.

4.3.1 Hlavní okno

První okno zobrazené uživateli při spuštění obsahuje právě výše zmiňovaný *RecyclerView*, obsahující všechny dostupné dotazníky a nad ním jen textový popis. Klepnutím na jeden z těchto dotazníků se načte tento dotazník do atributu *myApplication*, a otevře se nové okno výběru.

¹ <https://developer.android.com/studio/>



Obrázek 4.6. Android Hlavní okno

■ 4.3.2 Okno výběru

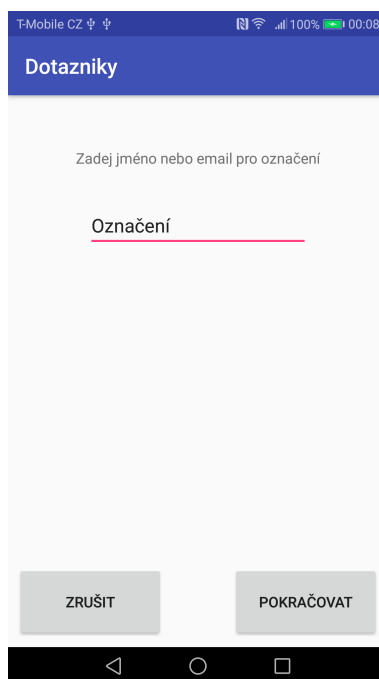
V okně výběru se nacházejí pouze tři tlačítka a popisek. První tlačítko spustí plnění dotazníku a vytvoří instanci `Vyhodnoceni`, kterou uloží do `myApplication`. Poté pro uživatele otevře nové okno označení. Druhé tlačítko slouží k zobrazování minulých vyhodnocení a vede do okna nabídky s vyhodnoceními současného dotazníku v `myApplication`. Tlačítko zpět vede zpět do hlavního okna.



Obrázek 4.7. Android Okno výběru

4.3.3 Okno označení

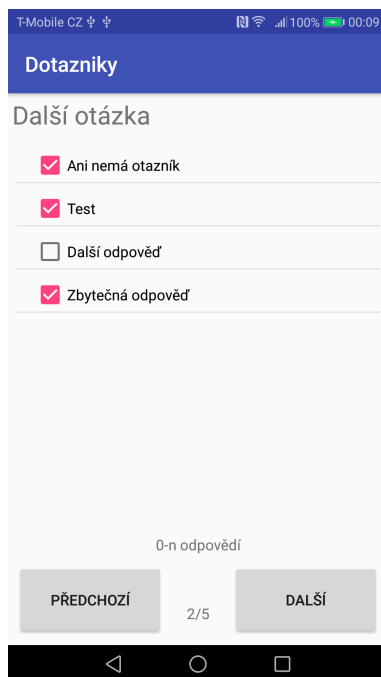
Okno označení pouze vyžaduje vyplnění textového pole. Po vyplnění je umožněno pokračování do vyplňování, čímž se označení uloží do instance Vyhodnoceni a zobrazí se okno otázky s první otázkou. Tlačítkem zrušit se lze vrátit do hlavního okna bez žádného ukládání.



Obrázek 4.8. Android Okno označení

4.3.4 Okno otázky

Okno otázky je nejsložitějším oknem v tomto prototypu. Layout na vrcholu obrazovky definuje textView, ve kterém se zobrazuje text současné otázky. Pod ním je definováno textové pole pro vyplnění, a zároveň RecyclerView. Pokud je typ současné otázky otevřená, RecyclerView se nezobrazí a uživatel pouze vyplňuje textové pole. Pro ostatní 2 typy se naopak schová textové pole a RecyclerView se naplní odpověďmi, které má otázka ve svém seznamu definované. Navíc podle typu umožní označení maximálně jedné možnosti, nebo více. Na spodu jsou dvě tlačítka, která slouží k přechodu mezi otázkami. Jejich stiskem se uloží současné odpovědi, posune se ukazatel otázky a znovu načte stejné okno (ale s novou otázkou). Levé tlačítko, pokud se uživatel nachází na první otázce, se mění na Zrušit. Stisknutím Zrušit je otevřeno hlavní okno a doposavadní vyplňování zahazeno. Stejně tak pravé tlačítko se na poslední otázce změní na Vyhodnotit. Klepnutím na něj se uloží poslední odpovědi, sečtou hodnoty k výsledkům a vyhodnocení se zapíše na soubor. Také je toto vyhodnocení zobrazeno v novém okně vyhodnocení.

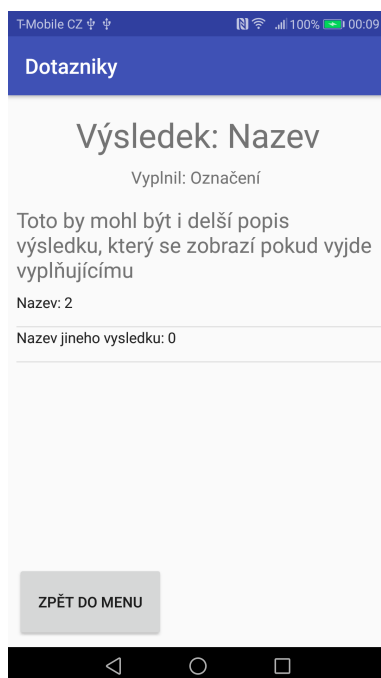


Obrázek 4.9. Android Okno otázky

4.3.5 Okno vyhodnocení

Okno vyhodnocení je nejméně interaktivní: nabízí pouze návrat do hlavního okna. Jsou v něm vypsány informace o příslušném vyplňování dotazníku: název dotazníku, kdo ho vyplnil, který výsledek vyšel a jeho popis.

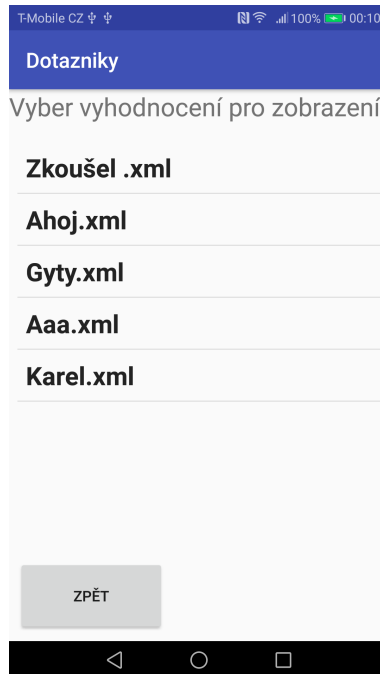
Nakonec také vypíše pro informaci všechny možné výsledky se sečtenými hodnotami odpovědí uživatele.



Obrázek 4.10. Android Okno vyhodnocení

4.3.6 Okno nabídky

Okno nabídky je podobné hlavnímu oknu, má jen navíc tlačítko pro možnost návratu právě do hlavního okna. RecyclerView je zde naplněn názvy souborů s vyhodnoceními příslušného dotazníku. Klepnutím na jedno z nich se toto vyhodnocení načte ze souboru a zobrazí v novém okně vyhodnocení.



Obrázek 4.11. Android Okno nabídky

Kapitola 5

Navržené dotazníky

V rámci práce bylo také úkolem navrhnout tři typy dotazníků a v prototypu je posléze vyzkoušet. Společně s vedoucím práce tedy byly s ohledem na možnosti prototypu a na požadavky na systém vytvořeny následující tři dotazníky. Soubory těchto dotazníků a vyhodnocení respondentů jsou k nahlédnutí přiloženy v přílohách.

5.1 Matematický test

Jedním z možných použití, jak už v tomto dokumentu bylo dříve uvedeno, je testování samotného respondenta. V tomto případě se jedná o otestování základních matematických schopností. Otázky jsou schválně triviální, aby respondenti netrávili mnoho času vyplňováním a soustředili se spíše na prototyp aplikace.

5.1.1 Stavba dotazníku

Jediný možný výsledek je nazván Výsledné body a využívá skutečnosti, že při vyhodnocení dotazníku je zobrazen i součet. Tento součet pak slouží k porovnání s maximálním počtem bodů, který je uveden v popisu výsledku. Jak už bylo uvedeno otázky jsou jednoduchého rázu, dotazník kombinuje otázky s 1 odpovědí a s více možnými.

5.1.2 Sledování

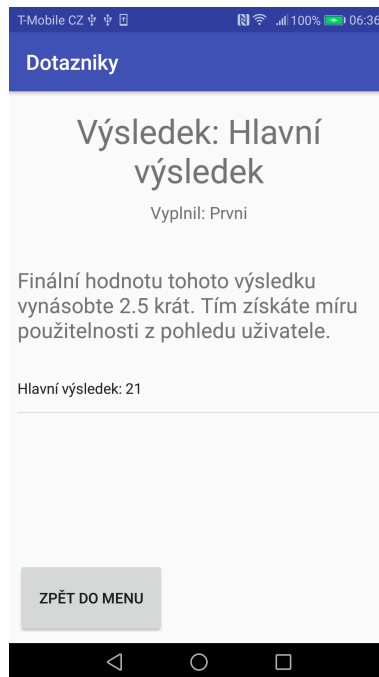
Plnění tohoto dotazníku proběhlo bez problémů, systém splnil očekávání respondentů. I při tvoření jsme nedošli k žádnému problému. V testech tohoto typu by se dalo také využít možnosti negativních hodnot pro případy špatných odpovědí. Tuto funkcionalitu prototyp podporuje.

5.2 SUS

Dalším dotazníkem vytvořeným pro testování byl dotazník pro použití měření použitelnosti SUS. K čemu se takový dotazník používá je popsáno v sekci 2.2.2. tohoto dokumentu.

5.2.1 Stavba dotazníku

Otázky a způsob vyhodnocení lze nalézt v sekci popisující SUS. V prototypu byly realizovány jako otázky s 1 odpovědí, kde uživatel mohl zvolit na škále od souhlasu po nesouhlasu u všech deseti výroků. Jediný výsledek reprezentující součet je vypočten z těchto odpovědí. Vynásobení 2.5 si uživatel musí udělat sám, jak je ukázáno na obrázku 5.1.



Obrázek 5.1. Vyhodnocení SUS

■ 5.2.2 Sledování

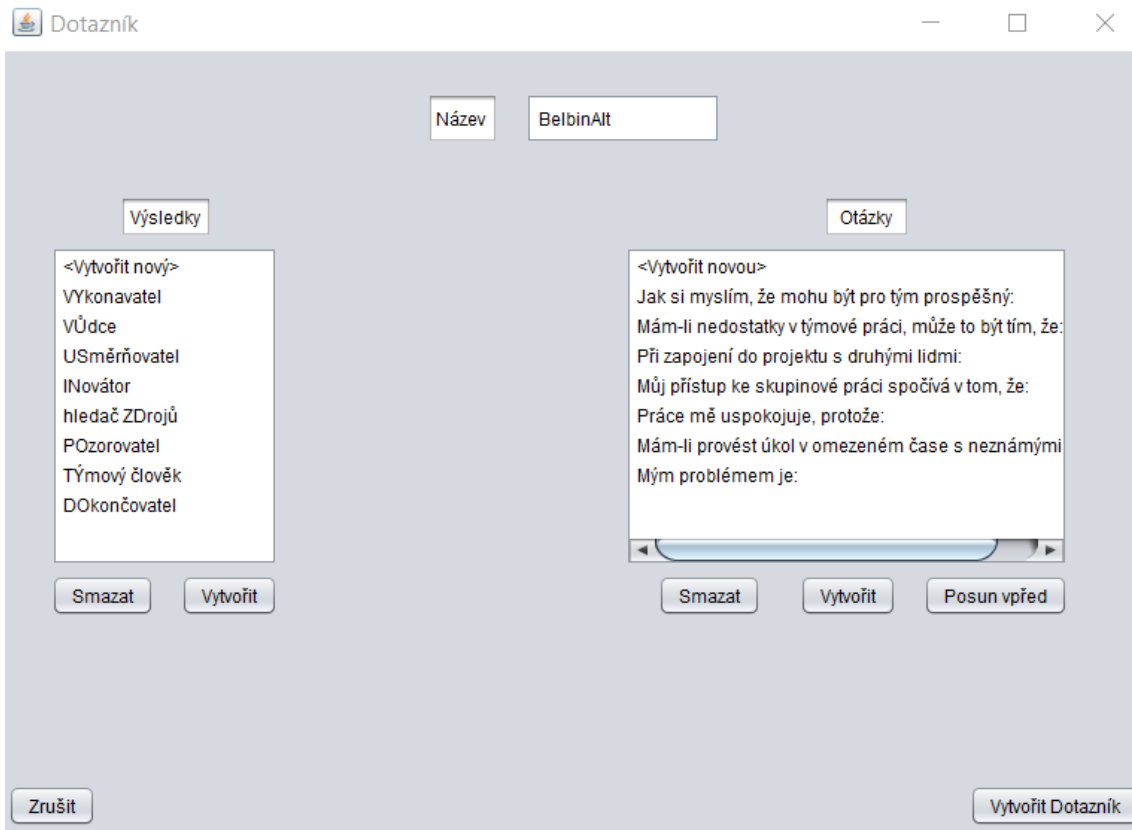
I pro SUS byl náš prototyp dostačující, dotazník byl sestaven tak jak má být a i vyplňování proběhlo podle očekávání. Při zakládání dotazníku vznikla myšlenka, že by mohlo být vhodné umožnit uživateli otázky také kopírovat: velmi by to zjednodušilo tvorbu dotazníku, ve kterém mají otázky stále stejné odpovědi (stačilo by otázku zkopírovat a jen upravit její text).

■ 5.3 BelbinAlt

Třetím dotazníkem tvořeným v rámci této práce je Belbinův test rolí. O něm si lze přečíst více v sekci 2.2.3. Zde ho nazýváme Belbinalt, protože jsme nebyli schopni přesně dodržet postupy u normálního vyplňování a dotazník je tedy mírně upraven.

■ 5.3.1 Stavba dotazníku

Tento test musel projít menšími úpravami ve způsobu vyplňování, protože náš prototyp neumožňuje požadovaný způsob plnění. Belbinův test se obvykle plní způsobem přidělování bodů mezi odpovědi u výroků a tyto body se pak sčítají mezi výsledky. Náš dotazník jsme poupravili tak, že ke každému výroku může respondent vybrat kolik chce odpovědí u každého výroku, a každá vybraná se přičte k výsledku, ke kterému patří. Na obrázku je vidět otevřené okno dotazníku s tímto dotazníkem načteným, kde můžeme vidět seznamy výsledků a otázek naplněné zadanými hodnotami.



Obrázek 5.2. Stavba dotazníku BelbinAlt

■ 5.3.2 Sledování

Jak bylo vidět už při stavbě, Belbinův test rolí není něco, co by náš prototyp mohl bez problému zvládnout. Dotazník vytvořený na jeho základě sice svou funkci také splní, chybí mu ale přesnost a spolehlivost, které dosahuje klasický Belbinův test. Výsledky se často lišili jen o jeden nebo dva body a to nemusí být dostačující pro definitivní určení osobnostní role.

Kapitola 6

Závěr

6.1 Zhodnocení

Problematika dotazníků je v dnešním světě velmi důležitým tématem. Míra používání dotazníků neustále stoupá a rozšiřují se i okruhy jejich využití. Není tedy divu, že se objevují i velmi nestandardní požadavky na aplikace, které se dotazníky zabývají. Pro takové požadavky mohou být obvyklé aplikace nedostačující. A právě takovou situaci řešila tato práce. V prvních částech proběhla analýza celkové problematiky, včetně pohledu na existující aplikace. Tyto poznatky poté byly porovnány s požadavky na nový systém. Na základě těchto rozdílů byl poté navržen a implementován první prototyp takto specifické aplikace. Rozdělení systému na dvě aplikace vedlo k zajímavému pohledu na rozdíly v implementaci s různými cílovými operačními systémy. Na závěr byly tyto prototypy použity pro tvorbu různých typů dotazníků, což posloužilo k otestování jejich funkčnosti.

6.2 Návrh na pokračování vývoje systému

Jak už několikrát bylo řečeno, systém je v současnosti pouze ve stavu prototypu. Dá se už sice použít pro požadovanou základní funkcionalitu, postrádá ale některé další funkce, které by mohli uživatelé očekávat. Také celá stavba systému je zatím pouze na hrubo postavená, rozhraní je neuhlazené a používá většinou výchozí nastavení. Zde jsou některé funkčnosti, které na základě původního návrhu a testování by mohly aplikaci posunout dál. Některé z těchto funkcností také patří mezi ty, které byly vyřazeny buď z důvodu redukce práce pro jednoho studenta, nebo že se došlo k závěru, že nejsou potřeba pro splnění požadavků.

- **Dokumentace a nápověda:** Kvalitní dokumentace jak systému tak kódu je v dnešní době standard a ve finální aplikaci by neměla chybět. Stejně tak nápovědy jsou obvyklou součástí a to i v případě podobně jednoduchých aplikací, jako má být tato.
- **Omezení přístupu běžnému uživateli:** V současnosti nelze nijak rozlišit, jestli s aplikací zachází respondent nebo autor dotazníku. Nemusí být vždy žádoucí, aby každý, kdo se k aplikaci dostane, měl plný přístup ke všem částem systému. Například by se dala schovat možnost zobrazení minulých dotazníků pro běžné uživatele.
- **Kopie otázek a výsledků:** Při zakládání nové otázky, nebo nového výsledku se musí všechny jejich součásti vyplnit od znovu. To může být zbytečně zdlouhavé, zejména v případě otázek, které se vzájemně liší jen v textu otázky.
- **Přenos souborů:** V brzkém návrhu systému se také přemýšlelo o umožnění přenosu těchto souborů v rámci aplikace (například v lokální síti, nebo přes internet, když by byl k dispozici).
- **Podpora dotazníků k testům:** I když zadavatel prozatím potřebuje pouze testy, nebylo by na škodu v takovéhle aplikaci zajistit i funkcionality dotazníků, sloužících ke sběru informací o respondentech.

- **Logika v dotazníku:** Logikou je myšleno přeskokování některých otázek na základě odpovědí na otázky dosavadně vyplněné. I když to opět nepatří mezi požadavky na systém, tato funkce by mohla najít své využití.
- **Otevřené odpovědi:** V prototypu nebyl implementován způsob, jak si zobrazit uživatelskou odpověď na otázky s otevřenou odpovědí. Zatím to tedy lze pouze náhlednutím přímo do souboru s vyhodnocením.



Literatura

- [1] Hermanus Johannes Ader. *Advising on Research Methods: A consultant's companion*. Johannes van Kessel Publishing, 2008. ISBN 9-079-41801-3.
- [2] George Beam. *The Problem with Survey Research*. Transaction Publishers, 2012. ISBN 1-412-84632-3.
- [3] Kristýna Chábová. *Komparace metodických postupů předvolebních průzkumů agentur CVVM, ppm factum, Median a STEM*. 2013.
https://cvvm.soc.cas.cz/media/com_form2content/documents/c3/a1395/f28/Komparace%20metodick%C3%BDch%20postup%C5%AF%20p%C5%99edvolebn%C3%ADch%20pr%C5%AFzkum%C5%AF.pdf .
- [4] Ian Sommerville. *Unified Modeling Language User Guide*. Pearson Education, 2004. ISBN 8-177-58530-4.
- [5] Grady Booch, James Rumbaugh a Ivar Jacobson. *Software engineering*. Druhé vydání. Addison-Wesley Professional, 2005. ISBN 0-321-26797-4.



Příloha **A**

Přílohy

- **Zdrojové kódy:** soubory obsahující zdrojové kódy obou částí systému
- **Soubory dotazníků a vyhodnocení:** soubory vytvořené v poslední kapitole, při uživatelském testování
- **Diagramy UML** projektový soubor EnterpriseArchitectu, obsahují veškeré diagramy.
- **Výsledný soubor textu:** PDF soubor tohoto dokumentu
- **Zdrojové soubory textu:** soubory použité pro tvorbu PDF



Příloha B

Zkratky

IT	Informační technologie
SUS	System Usability Scale
UML	Universal Modeling Language
OS	Operační systém
API	Application programming interface
XML	Extensible Markup Language
IDE	Integrated development environment
GUI	Graphical User Interface