



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

WEBOVÝ NÁSTROJ PRO TESTOVÁNÍ POMOCÍ Q-ŘAZENÍ

WEB TOOL FOR TESTING BASED ON Q-SORTING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ALEXANDR PERSICH

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. ADAM HEROUT, Ph.D.

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Persich Alexandr**

Obor: Informační technologie

Téma: **Webový nástroj pro testování pomocí Q-řazení**
Web Tool for Testing Based on Q-Sorting

Kategorie: Uživatelská rozhraní

Pokyny:

1. Seznamte se s technikou Q-sorting pro zjišťování subjektivního hodnocení.
2. Vyhledejte a prostudujte existující nástroje pro hodnocení pomocí Q-řazení.
3. Prostudujte vhodné webové technologie pro realizaci webového nástroje pro Q-sorting.
4. Navrhněte prvky uživatelského rozhraní pro Q-sorting.
5. Implementujte dílčí prototypy navrženého uživatelského rozhraní. Testujte prototypy na uživateli a navrhněte vhodná vylepšení.
6. Navrhněte a implementujte webový nástroj pro Q-řazení.
7. Realizujte několik experimentů s použitím vytvořeného nástroje včetně jejich vyhodnocení a demonstруйте tak jeho použitelnost.
8. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu; vytvořte plakátek a krátké video pro demonstrování výsledků projektu.

Literatura:

- dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3, značné rozpracování bodů 4 až 6.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

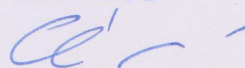
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo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

Vedoucí: **Herout Adam, prof. Ing., Ph.D.,** UPGM FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Q-řazení je jednou z metod používaných pro zjišťování názorů lidí na zkoumané téma. Cílem této práce je seznámit čtenáře s návrhem a tvorbou webové aplikace realizující Q-řazení. Zaměřil jsem se především na uživatelské rozhraní, které ve velké míře ovlivňuje počet úspěšně dokončených průzkumů. K řešení jsem použil běžně dostupné webové technologie. Výsledkem je webová aplikace, která zadavatelům umožňuje prostřednictvím administrace vytvoření vlastního průzkumu a respondentům poskytuje jednoduché uživatelské rozhraní, které jim usnadňuje proces řazení.

Abstract

Q-sorting is one of many methods used to determine people's subjectivity. The main goal of this thesis is to present ways used to design and create web application for Q-sort method. I have been focusing mainly on designing user interface which greatly affects the number of successfully finished surveys. To accomplish this, I have used common web technologies. The result is a web tool which allows to create and manage new surveys. For respondents it provides simple and intuitive user interface which makes the process of Q-sorting easier.

Klíčová slova

Q-řazení, Q-řazení online, Q-metodologie, zkoumání lidské subjektivity, grafické uživatelské rozhraní, webová aplikace, drag&drop

Keywords

Q-sort, Q-sort online, Q-methodology, study of subjectivity, GUI, UX, web application, web design, drag&drop

Citace

PERSICH, Alexandr. *Webový nástroj pro testování pomocí Q-řazení*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Herout Adam.

Webový nástroj pro testování pomocí Q-řazení

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana prof. Ing. Adama Herouta Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Alexandr Persich
16. května 2017

Poděkování

Děkuji panu prof. Ing. Adamu Heroutovi Ph.D. za odborné vedení, ochotu a cenné rady, jež mi v průběhu tvorby bakalářské práce věnoval. Mé poděkování patří též těm, kteří se podíleli na testování této práce.

Obsah

1 Úvod	3
2 Základy Q metodologie	4
2.1 Úvod	4
2.2 Možnosti využití Q-metodologie	5
3 Analýza existujících řešení	7
3.1 Dostupné aplikace	7
3.2 Výběr vhodných prvků	12
4 Použité technologie	13
4.1 Technologie na straně serveru	13
4.2 Technologie na straně klienta	13
5 Návrh	15
5.1 Cílové skupiny uživatelů	15
5.2 Filosofie návrhu uživatelského rozhraní	16
5.3 Prvky uživatelského rozhraní	16
6 Implementace	21
6.1 Použití šablon	21
6.2 Administrace	21
6.2.1 Přihlašování pomocí služby Google Sign-In	21
6.2.2 Vytváření průzkumu	22
6.2.3 Generování odkazů na průzkum	22
6.2.4 Odesílání e-mailů	23
6.2.5 Exportování výsledků průzkumu	23
6.3 Formát dat průzkumu	23
6.4 Načítání a ukládání stavu průzkumu	24
6.5 Dynamické vykreslování výsledkové mřížky	24
6.6 Bezpečnostní opatření	25
7 Testování	27
8 Závěr	29
Literatura	30
Přílohy	31

A Obsah CD	32
B Plakát	33

Kapitola 1

Úvod

Tato práce pojednává o návrhu, implementaci a testování webového nástroje na provádění průzkumů pomocí Q-řazení, což je metoda používaná na zjišťování názorů a postojů lidí na různá témata. Vytvořený nástroj bude používán jako náhrada fyzického Q-řazení pomocí papírových kartiček.

Hlavním cílem je vytvořit nástroj s propracovaným uživatelským rozhraním, které respondentům průzkumů umožní jednoduchou a pohodlnou realizaci Q-řazení. Uživatelské rozhraní bude důkladně testováno. Důraz bude rovněž kladen na celkový design. Vedlejším cílem je vytvoření administrace na přidávání a správu průzkumů.

V kapitole 2 se čtenář nejprve seznámí se základními principy Q-metodologie, zejména pak s Q-řazením. Kapitola 3 je věnována analýze nedostatků nebo naopak silných stránek některých již existujících aplikací realizujících Q-řazení. Technologie, které jsem použil k tvorbě nástroje, jsou shrnuty v kapitole 4. V kapitole 5 se věnuji návrhu uživatelského rozhraní pro cílové skupiny uživatelů. Uvedeno je zde i několik konkrétních prvků uživatelského rozhraní, které jsou ve vytvořeném nástroji použity. V kapitole 6 jsou popsány implementační detaily některých zajímavějších částí nástroje. Testování nástroje a jeho uživatelského rozhraní je věnována kapitola 7.

Kapitola 2

Základy Q metodologie

V této kapitole je zmíněn úvod do problematiky Q metodologie a do jejích částí. Dále se zde budu zabývat možnostmi jejího praktického využití. Obé bylo stěžejní pro pochopení tématu a úspěšné vytvoření webové aplikace, která realizuje Q-řazení.

2.1 Úvod

Q metodologie je způsob zjišťování lidské subjektivity, který byl vytvořen kolem roku 1935 fyzikem a psychologem Williamem Stephensonem [4]. Subjektivitou se zde myslí názor, domněnka, postoj či pohled veřejnosti na dané téma. Q metodologie zahrnuje dvě základní složky:

- způsob sběru dat (Q-sort, Q-řazení)
- způsob analýzy sesbíraných dat (typicky faktorová analýza)

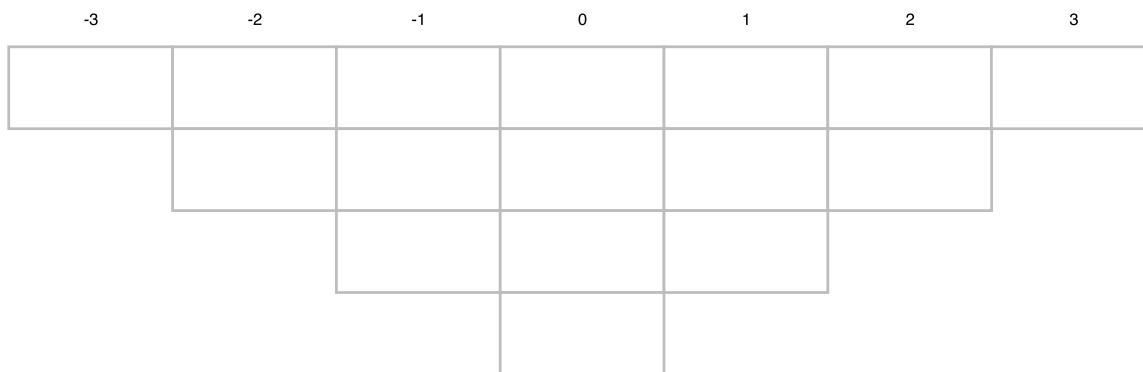
Základní informace a uvedení do principů a fungování této metodologie mimo jiné zmiňuje video z Leeds Beckett University produkované Timem Deignanem¹.

Q-řazení

Před započítáním samotného řazení je třeba z různých zdrojů sesbírat co nejvíce tvrzení o daném tématu, která by měla zahrnovat co možná největší množství názorů a postojů, které mohou lidé v souvislosti s tímto tématem mít. Tato obsáhlá množina se poté zredukuje na množinu tvrzení, která jsou pro téma podstatná. Ta je označována jako Q-set a měla by obsahovat dostatečné množství tvrzení na to, aby byl každý schopen vyjádřit svůj osobní názor na dané téma. Jednotlivá tvrzení v Q-setu jsou respondentům průzkumu předložena typicky ve formě kartiček za účelem jejich rozřídění do výsledkové mřížky (obr. 2.1), která tvarem připomíná pyramidu či rozložení podle Gaussovy křivky. Dříve se využívaly kartičky papírové, s rozvojem informačních technologií však přibývají různé adaptace, které převádí papírovou podobu Q-řazení do podoby elektronické.

Řazení typicky probíhá ve dvou krocích. Respondenti jsou nejprve požádáni o rozřídění kartiček do tří základních skupin – souhlasná, nesouhlasná, nerozhodná. Poté jsou požádáni o jejich další rozřídění do výsledkové mřížky, která přesněji zachytí jejich souhlasný či nesouhlasný postoj k danému tvrzení. K okrajům mřížky jsou umístěna tvrzení, na která

¹Leeds Metropolitan Quick-Q Animation – <https://www.youtube.com/watch?v=0AejeH6jw2c>



Obrázek 2.1: Příklad výsledkové mřížky pro umístění kartiček s tvrzeními.

má respondent vyhraněný názor, ve středu mřížky se naopak objeví tvrzení, jež respondent považuje za názorově neutrální.

Výsledkem celého Q-řazení je množina roztríděných Q-setů, kde každý z nich přesně zachycuje názory a postoje jednoho respondenta na dané téma. V samotném závěru může být o výsledku Q-řazení vedena diskuze, kde respondent vysvětlí, proč se rozhodl rozřadit tvrzení právě tímto způsobem.

Metoda Q-řazení je časově i mentálně náročná, získané výsledky ale mají vysokou výpovědní hodnotu. Proto je metoda Q-řazení označována jako kvalitativní, spíše než kvantitativní. Pro analýzu daného tématu typicky stačí kolem 60 roztríděných Q-setů [3, 1].

Analýza získaných dat a vyhodnocení

Podstatou analýzy dat získaných Q-řazením je faktorová analýza. Ve velkém množství roztríděných Q-setů se snažíme najít podobnosti s cílem identifikovat a seskupit respondenty, kteří mají na zkoumanou problematiku podobný názor. Z takovýchto skupin utvoříme obecnější Q-setsy, které reprezentují názory či postoje sdílené větší skupinou respondentů. Takto získaná data mohou poskytnout cenné informace o názorech převažujících ve společnosti [8].

Pro provedení analýzy Q-setů je k dispozici celá řada nástrojů jako například aplikace PCQ², vyvíjena Michaelem Stricklinem, balíček pro statistický jazyk R, qmethod³, jehož autorem je Aiora Zabala nebo program PQMethod⁴ vytvořen Johnem Atkinsonem na Kent State University.

2.2 Možnosti využití Q-metodologie

Stejně jako ostatní typy průzkumů má Q metodologie celou řadu odvětví a směrů, ve kterých se dá použít. Typickým příkladem může být zjišťování názoru veřejnosti na distribuci financí ve zdravotnictví, zjišťování politických preferencí, či průzkum postojů a názorů pedagogů na školách⁵. Využít se ale samozřejmě dá všude tam, kde je potřeba zkoumat převažující názor či postoj lidí.

²PCQ Soft – <http://www.pcqsoft.com/>

³CRAN, balíček qmethod – <https://cran.r-project.org/web/packages/qmethod/index.html>

⁴PQMethod Software – <http://schmolck.userweb.mwn.de/qmethod/#PQMethod>

⁵Q-řazení ve školství – <http://clanky.rvp.cz/clanek/o/z/920/Q-TRIDENI-JAKO-NASTROJ-ZPETNE-VAZBY-VE-SKOLE.html/>

Online aplikace využití Q metodologie zjednodušují, jelikož není potřeba fyzické přítomnosti respondentů a třídění tvrzení v Q-setu není vázáno na konkrétní čas, což vede k většímu pohodlí pro respondenty. To se může projevit pečlivěji roztríděným Q-setem. Negativem naopak může být absence závěrečné diskuze, jelikož diskusi nelze plně nahradit pouhým formulářem k vyplnění.

Kapitola 3

Analýza existujících řešení

Důležitou částí pro úspěšné vytvoření aplikace realizující Q-řazení byla analýza již existujících řešení. Cílem bylo odhalit jejich nedostatky nebo naopak silné stránky. Zaměřil jsem se především na prvky uživatelského rozhraní, intuitivnost, způsoby vizualizace kartiček s tvrzeními a design.

V této kapitole zmíním některé z dostupných aplikací a rozeberu jejich silné, či naopak slabé stránky, ze kterých jsem čerpal při návrhu vlastního řešení.

3.1 Dostupné aplikace

FlashQ

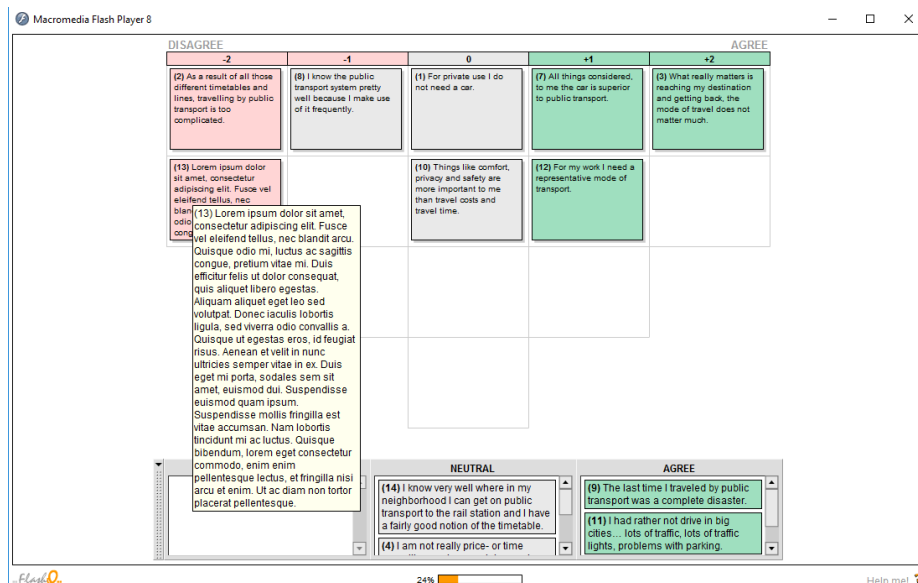
FlashQ¹, vyvíjen8 Christianem Hackertem, je Flash aplikace realizující Q-řazení. Dostupná je online i offline verze, nicméně kvůli zastaralosti platformy Flash není jednoduše spustitelná v novějších prohlížečích. Pro testování jsem tedy zvolil její offline verzi.

Průběh řazení je rozdělen do pěti kroků. V prvním kroku je provedeno třídění do tří základních kategorií – souhlasné, neutrální a nesouhlasné. Kartičky s tvrzeními lze do příslušných kategorií přesouvat myší nebo pomocí kláves. K dispozici je i počítadlo, které dává respondentovi informaci o počtu roztríděných kartiček.

Ve druhém kroku probíhá řazení do výsledkové pyramidy. Nevýhodou je, že sloupce, ze kterých se kartičky přesunují, jsou umístěny u vrcholu výsledkové mřížky, nikoliv u její základny, což znamená že se většina kartiček musí táhnout na větší vzdálenost. Je-li text karty příliš dlouhý, je zkrácen a jeho zbytek se zobrazí po najetí kurzoru na příslušnou kartu (viz. obrázek 3.1). Druhý krok celkově dobře imituje papírovou vezri Q-řazení. Ve třetím kroku je respondent požádán o pouhé překontrolování kroku druhého, přičemž pouze přibyla funkcionality záměny dvou kartiček mezi sebou. Tento krok mi připadá nadbytečný, jelikož jde jednoduše sloučit s krokem předchozím.

Ve čtvrtém kroku je připraven formulář obsahující dvě okrajová tvrzení (tvrzení, jež respondent označil jako nejjednoznačnější), ve kterém se očekává zdůvodnění jejich umístění ve výsledkové mřížce. Řazení končí pátým krokem, kde je potřeba vyplnit základní osobní údaje.

¹FlashQ, sorting via Internet – <http://www.hackert.biz/flashq/home/>



Obrázek 3.1: Druhý krok Q-řazení v aplikaci FlashQ s ukázkou kartičky s dlouhým textem.

HtmlQ

Nástroj HtmlQ² od společnosti Aproxima je novější verzí již zmíněné aplikace FlashQ. Je s ní zpětně kompatibilní a využívá stejné konfigurační soubory. Díky responsivnímu designu je možno provádět Q-řazení i na mobilních zařízeních.

Na první pohled nejsou od aplikace FlashQ vidět žádné výrazné vizuální změny, což bych od bezmála osm let novějšího nástroje očekával. Grafické zpracování tedy považuji za zastaralé. Velké mínus je uživatelsky nepřívětivé nastavování nového průzkumu, což zahrnuje manuální úpravy konfiguračních souborů, do kterých se kromě identifikátorů kartiček a textů jednotlivých tvrzení musí dopočítat i velikost jednotlivých sloupců výsledkové mřížky.

Průběh samotného Q-řazení je koncipován do pěti kroků stejně jako v případě FlashQ.

Q-Assessor

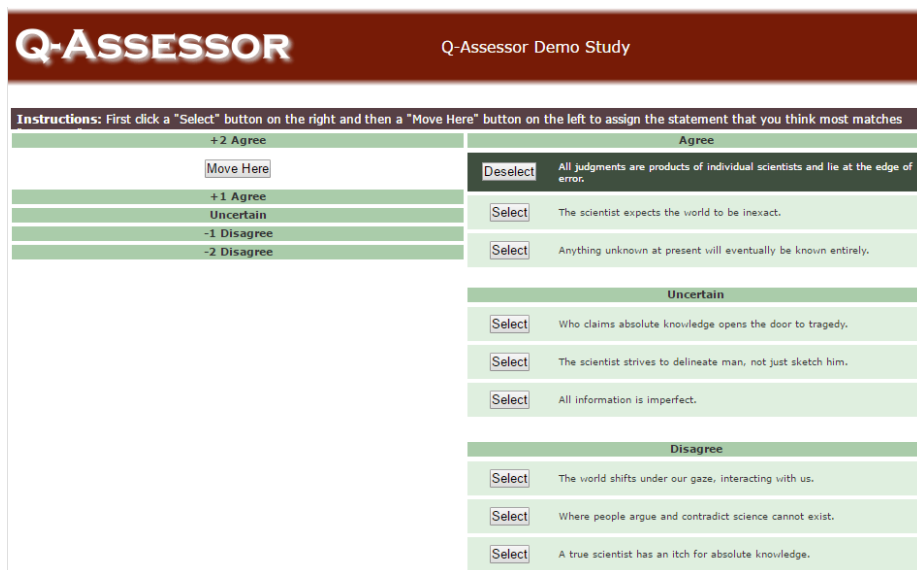
Q-assessor³ je webová aplikace od americké skupiny The Epimetrics Group⁴, která se mimo jiné zabývá průzkumem a analýzou dat z různých oborů. V jejich demonstračních příkladech mne zaujaly dvě aplikace používající různý přístup ke Q-řazení.

Jedna z nich, označena jako „Q-řazení se standardním rozhraním“, používá princip třídění karet prostřednictvím klikání. Nejprve se předložená tvrzení pomocí tlačítek roztřídí do obvyklých tří základních skupin. Poté jsou respondentovi prezentovány skupiny s cílem bližšího rozřazení, kde je vždy jedna předvolena a do ní je nutno zařadit některou z kartiček. Zde vidím hlavní nevýhodu v tom, že si respondent zpočátku nemůže zvolit, do jaké z těchto konkrétnějších kategorií bude kartičku s tvrzením umisťovat. Situace je zachycena na obrázku 3.2. Následuje krátký dotazník požadující vyplnění osobních údajů. Řazení klikem hodnotím pozitivně, jelikož může být časově podstatně efektivnější, než řazení tažením, nicméně zde zvolené uživatelské rozhraní tento potenciál plně nevyužívá.

²HtmlQ – <https://github.com/aproxima/htmlq>

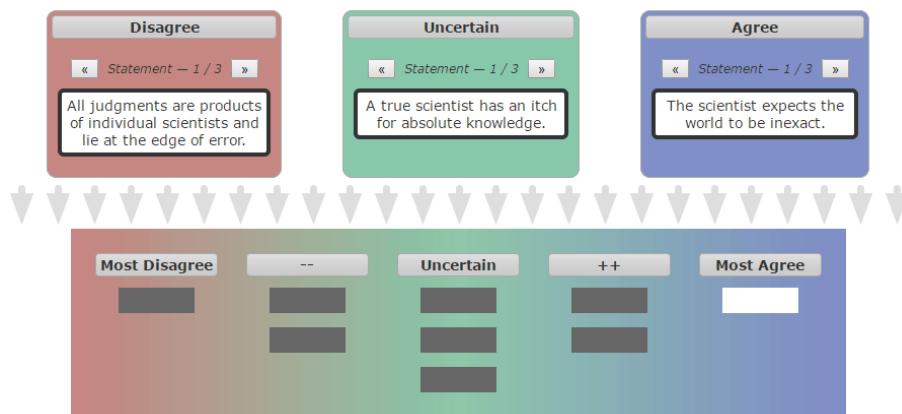
³Q-Assessor – <http://q-assessor.com/>

⁴The Epimetrics Group – <http://epimetrics.com/>

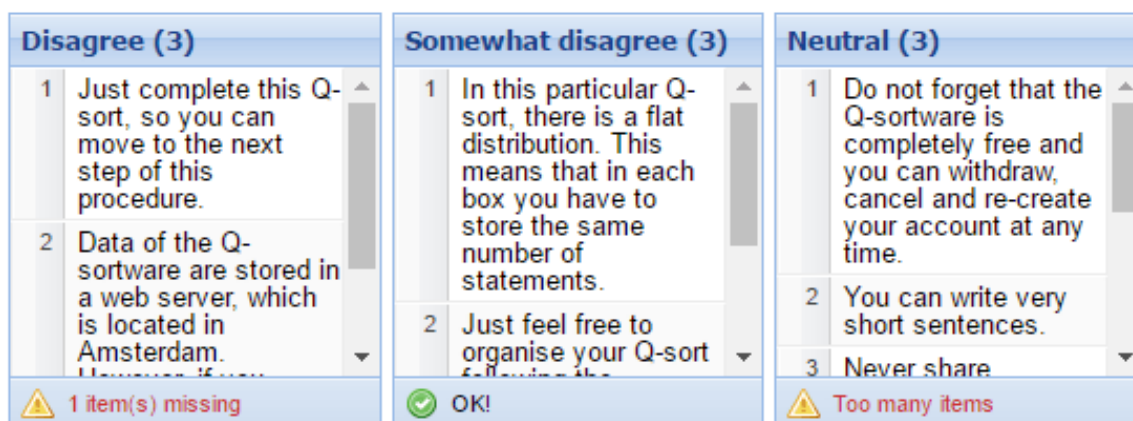


Obrázek 3.2: Vpravo výběr kartičky s tvrzením, vlevo předvolená skupina „+2 Agree“, do které se má kartička vložit.

Druhá z aplikací, označena jako „Q-řazení s rozhraním Drag-and-Drop“, využívá princip přetahování kartiček s tvrzeními do příslušných kategorií. V prvním kroku opět probíhá předtřídění do tří skupin, v kroku druhém je zapotřebí přetáhnout kartičky do výsledkové mřížky. Respondent je zde ale opět nucen začít s řazením tvrzení s nejsilnějším souhlasem a nesouhlasem. Za dobrý nápad považuji použití šipek v pozadí, které v obou krocích naznačují, co dělat (obr. 3.3). Nešťasný je naopak způsob zobrazení kartiček, které jsou ve výsledkové mřížce příliš malé, a neefektivní využití prostoru na stránce.



Obrázek 3.3: Q-Assessor Drag-and-Drop – druhý krok.



Obrázek 3.4: Výřez z aplikace Q-sortware s ukázkou zobrazení kapacity sloupce.

Q-sortware

Dalším z existujících řešení Q-řazení je Q-sortware⁵, který byl vyvíjen doktorem Alessio Prunedduem na University of York.

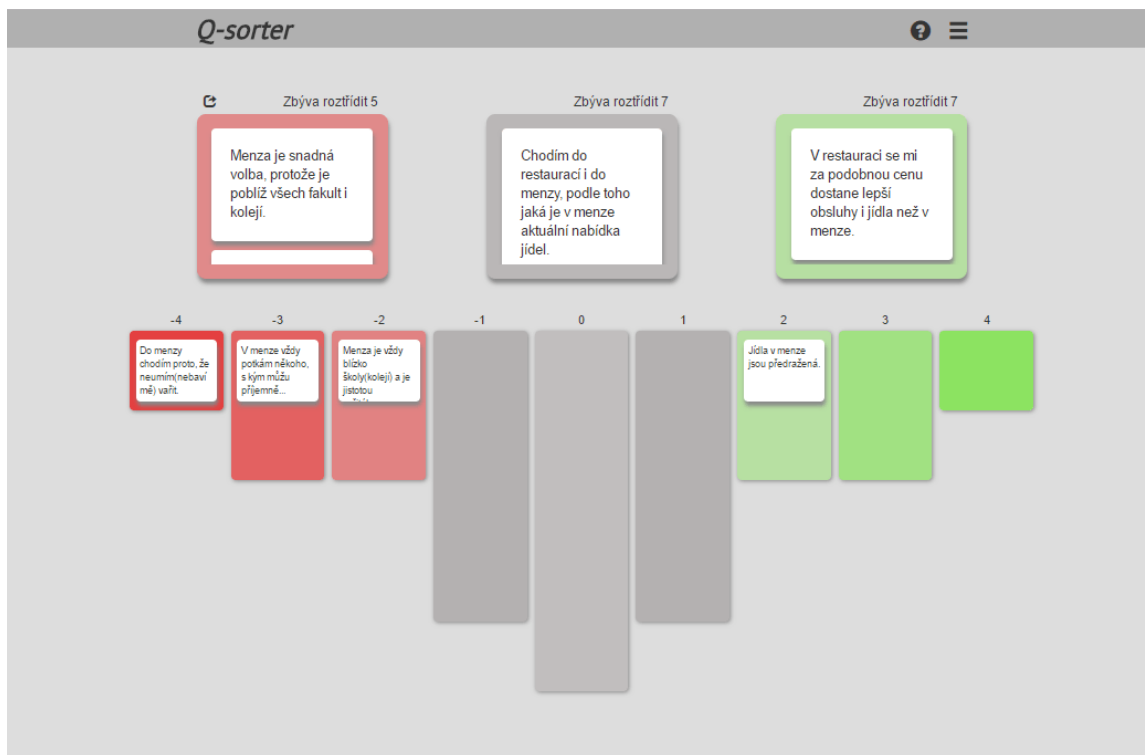
Tvrzení jsou řazena tažením. V prvním kroku probíhá standardně třídění do tří základních sloupců. V kroku druhém se karty řadí do sloupců, které nahrazují klasickou výsledkovou mřížku. Každý ze sloupců má maximální kapacitu vyznačenu pouze v textové formě. Z grafického návrhu tedy není hned na první pohled vidět, kolik do každého sloupce patří kartiček (obr. 3.4). Design působí zastarale, dobře je naopak vyřešeno umístění elementů na stránku, která je díky tomu plně využita.

Q-sorter

Online aplikaci Q-sorter⁶ vytvořil v akademickém roce 2015/2016 Vojtěch Mička jako součást bakalářské práce na Fakultě informačních technologií Vysokého učení technického v Brně.

⁵Q-sortware – <http://www.qsoftware.net/>

⁶Q-sorter – <http://q-sorter.vojtam.cz/>



Obrázek 3.5: Průběh druhého kroku řazení pomocí nástroje Q-sorter. Označení nejlevější sady jako výchozí pro řazení pomocí klikání.

Autorovi se zde podařilo doplnit některé nedostatky výše uvedených programů. Příkladem může být kombinace možností řazení kartiček jak tažením, tak i klikáním na příslušné sloupce. V prvním kroku, kde probíhá třídění do tří základních kategorií, chybí počítadlo kartiček, které by respondentovi sdělilo počet kartiček k rozřídění.

Ve druhém kroku probíhá třídění kartiček do výsledkové mřížky. Jednotlivé sloupečky jsou ohodnoceny čísly, která v kombinaci s barevným pozadím znázorňují, která strana je souhlasná resp. nesouhlasná. Větší jednoznačnosti by zde určitě pomohlo přidání slovních popisků. Při řazení klikáním se do příslušného sloupce výsledkové mřížky přesune první kartička z označené sady. Označení ale není dostatečné a lze ho snadno přehlédnout (obr. 3.5). Jako nešťastné považuji řešení zvětšování kartiček. Ty se po najetí kurzoru roztáhnou do stran, což kromě zvětšení způsobí postupné přeskakování slov mezi řádky. Po úplném rozřídění chybí možnost vzájemné záměny dvou kartiček. Pokud se respondent rozhodne změnit názor, musí jednu ze zaměňovaných kartiček vyjmout z výsledkové mřížky, nahradit ji, a poté zařadit do nového sloupce.

Další nevýhodou může být předem stanovená velikost Q-setu. Podporováno pět různých počtů tvrzení (23, 28, 41, 50 a 65). Zajímavý nápad je použití overlay nápovědy, která se zobrazí formou popisků přímo do pracovní plochy.

3.2 Výběr vhodných prvků

Z průzkumu výše uvedených aplikací jsem vybral některé prvky a přístupy, které považuji za přínosné, a které by se daly využít pro mou práci. Rovněž jsem odhalil nedostatky, kterým bych se při tvorbě vlastní aplikace měl vyhnout.

Jelikož je Q-řazení poměrně náročné, přijde mi rozdělení do obvyklých dvou kroků dostačující. Další formuláře sice mohou poskytnout dodatečné informace, jsou ale zároveň dalším zatížením respondenta. V programu Q-sorter je i přes drobné nedostatky dobře zkombinováno řazení pomocí tažení a řazení pomocí klikání. Za důležité též považuji počítadlo neroztříděných kartiček, ze kterého lze odhadnout, kolik času celé řazení zabere.

Při tvorbě vlastního řešení bude zapotřebí dbát na efektivní využití místa v okně prohlížeče, do kterého se často musí vejít velké množství prvků. Důležitou vlastností by mělo být dynamické vykreslování výsledkové mřížky, aby zadavatel průzkumu nebyl omezen předem definovaným počtem tvrzení, jež může zadat. Zde je třeba počítat s rozsáhlým průzkumem o velkém počtu kartiček. Jedním z dalších cílů bude navržení designu dle současných trendů, jelikož jsou v tomto ohledu dostupné aplikace pozadu.

Kapitola 4

Použité technologie

V této kapitole jsou popsány technologie, které jsem se rozhodl pro realizaci online nástroje pro Q-řazení použít. Volil jsem na základě použitelnosti pro daný účel a předešlých zkušeností.

4.1 Technologie na straně serveru

PHP

PHP (Hypertext Preprocessor) je víceúčelový skriptovací jazyk, typicky používáný pro vývoj dynamických webových stránek. Díky jeho rozšíření má dobře dostupnou a propracovanou dokumentaci¹, což usnadní práci a urychlí řešení problémů. Je nezbytný například pro komunikaci s databází, některé výpočty či dynamické vykreslování výsledkové mřížky. Dále bude využit pro ukládání a načítání dat jednotlivých průzkumů a vytvoření souboru k exportu dat z dokončených průzkumů.

MySQL

Pro uchování dat o průzkumech a informací týkajících se jednotlivých rozpracovaných Q-setů bude využito MySQL², což je databázový systém založený na relačním databázovém modelu. Pro komunikaci používá dotazovací jazyk SQL. MySQL jsem zvolil především kvůli dobré podpoře, rychlosti a bezpečnosti.

4.2 Technologie na straně klienta

HTML5

Jazyk HTML³ (HyperText Markup Language) je značkovací jazyk, který je typický pro webové stránky. Jeho nová verze, HTML5, v mnoha ohledech vylepšuje své předchůdce a vytváří plnohodnotné API (aplikační programové rozhraní) pro vývoj webových aplikací [2]. Tento jazyk je pro realizaci této práce zcela nezbytný a nenahraditelný.

¹PHP, manual – <http://php.net/manual/en/>

²MySQL, official website – <https://www.mysql.com/>

³w3schools, HTML5 Introduction – https://www.w3schools.com/html/html5_intro.asp

CSS3

Pro požadovaný vzhled a design webových stránek je nutné použít technologii CSS (Cascading Style Sheet), která upravuje způsob zobrazení HTML elementů. Konkrétně bude použit poslední vydaný standard, a to CSS3⁴, který přináší pokročilé selektory, stínování, zaoblení hran, transformace a rotace, barevný model RGBA a další. Využití CSS3 je nezbytné k dosažení vizuálně přijatelného uživatelského rozhraní.

JavaScript, jQuery

JavaScript⁵ je skriptovací jazyk, který se obvykle používá pro ovládání interaktivních prvků, jako jsou například tlačítka a vstupní pole, nebo na animace, přechody či jiné efekty.

Pro zjednodušení práce jsem použil JavaScriptovou knihovnu jQuery⁶. Nabízí rozšíření a zjednodušení běžných operací, animací, reakcí na události a zjednodušení manipulace s elementy DOM (Document Object Model). Díky rozšířené podpoře této knihovny fungují všechny její části na převážné většině dnešních prohlížečů bez problémů.

Ve velké míře bude rovněž používána technologie Ajax (asynchronous JavaScript and XML), která umožňuje na pozadí provádět komunikaci se serverem a překreslovat obsah stránky bez nutnosti jejího znovunačtení.

Bootstrap

Weboví vývojáři a designéři se dnes často přiklání k použití Bootstrapu⁷, což je HTML, CSS a JavaScriptový framework, pro tvorbu responsivních webových stránek či aplikací.

Nabízí velké množství komponent, které svým designem odpovídají dnešním požadavkům. Používá grid layout, který funguje na bázi řádků a sloupců. Tento layout sice poskytuje značnou volnost při tvorbě webových stránek, ale pro některá specifická řešení nemusí být vhodný. Proto ve své práci využiji Bootstrap pouze pro informační stránky a administraci. Pro samotnou realizaci Q-řazení nepovažuji Bootstrap za dobře použitelný.

Dragula

Pro realizaci funkcionality řazení pomocí tažení kartiček je vhodné použít knihovnu řešící funkcionalitu drag&drop. Pro účely této práce jsem se rozhodl použít JavaScriptovou knihovnu dragula⁸, vyvíjenou Nicolasem Bevacquaem.

Dragula disponuje velmi jednoduchým API, stručnou ale kvalitní dokumentací a několika vzorovými příklady. Použití je jednoduché – stačí definovat kontejnery, mezi kterými se budou prvky přesouvat, a případně upravit některá nastavení pro pokročilejší manipulaci s prvky. Jednoduché je též odchyťování událostí, jako například chycení či upuštění taženého elementu, které je nezbytné nejen pro aktualizaci počítadla přesunutých kartiček.

⁴w3schools, CSS3 Introduction https://www.w3schools.com/css/css3_intro.asp

⁵JavaScript – <https://www.javascript.com/>

⁶jQuery – <https://jquery.com/>

⁷Bootstrap framework – <http://getbootstrap.com/>

⁸Knihovna Dragula – <https://github.com/bevacqua/dragula>

Kapitola 5

Návrh

Před samotnou tvorbou práce bylo nutné seznámit se s různými typy uživatelů, kteří by mohli online nástroj pro realizaci Q-řazení používat, prostudovat problematiku tvorby uživatelského rozhraní, a poté navrhnout jednotlivé prvky, prostřednictvím kterých budou uživatelé aplikaci ovládat. Tomu se budu věnovat v této kapitole.

5.1 Cílové skupiny uživatelů

Obecně lze říci, že tvořit program bez znalosti požadavků lidí, kteří s ním mají pracovat, je prvním krokem k neúspěchu. Ze strany mé práce to znamenalo seznámit se s požadavky a nároky z pohledu tvorby nového průzkumu a také z pohledu jeho vyplňování.

Zadavatelé průzkumu

Zadavatel chce mít možnost volby vlastního textu popisek základních skupin, do kterých se v prvním kroku tvrzení třídí. Jejich význam může do jisté míry ovlivnit výsledky průzkumu. Nelze tedy aplikovat stejné popisky na všechny druhy průzkumů, například tvrzení „Dávám přednost jazyku C++ před jazykem Java.“ je jednodušší intuitivně zařadit do některé z kategorií „Ano, Možná, Ne“, než do některé kategorie z „Souhlasím, Nevím, Nesouhlasím“. Je tedy vhodné, aby aplikace umožňovala jejich změnu.

Další funkcionalitou by měla být možnost rozeslat průzkum respondentům na základě poskytnutého seznamu e-mailů, a to jak hromadně, tak i selektivně, jelikož se text zprávy různým respondentům může lišit.

Po ukončení průzkumu je očekáván export výsledků v takovém formátu, aby jej bylo možné vyhodnotit pomocí dostupných nástrojů, například těch, jež jsou zmíněny v kapitole 2.1, ve které se zabývám analýzou a zpracováním dat.

Respondenti

Převážnou většinu času budou s mou aplikací pracovat respondenti. Těm bude poskytnut unikátní odkaz, prostřednictvím kterého jim bude daný průzkum zpřístupněn.

Očekává se jednoduché rozhraní, jehož prostřednictvím bude celé Q-řazení probíhat. Respondent chce mít k dispozici jasnou, srozumitelnou a hlavně stručnou nápovědu, která bude dostupná ze všech kroků řazení.

5.2 Filosofie návrhu uživatelského rozhraní

Q-řazení je svou podstatou unikátní, což nutně vedlo na zcela individuální přístup k uživatelskému rozhraní. Při jeho návrhu jsem se zaměřil především na jeho použitelnost pro účel třídění kartiček s tvrzeními.

Aplikovat se zde ale dají i některá obecná pravidla. Snaha byla o eliminaci dlouhých textových pasáží, které mohou být důvodem uživatelského odchodu z webových stránek. Rozhodnutí, zda na stránkách setrvat, totiž většinou trvá prvních několik vteřin [7]. Dále jsem se snažil zachovat jednotnost a jednoduchost barev tak, aby neodpoutávaly pozornost respondenta od samotného průzkumu.

Při návrhu jsem se zaměřil na user-centered design, který začíná být v dnešních uživatelských rozhraních standardem. Principem je přizpůsobení systému uživateli s myšlenkou pomoci mu s prací a s naplněním jeho cílů v tomto systému. Základem je tedy vývoj programu s myšlenkou, jakým způsobem bude uživatel systém využívat, což přispívá k zefektivnění práce. Nutným předpokladem je vytváření uživatelského rozhraní ve spolupráci s uživatelem [10, 11].

5.3 Prvky uživatelského rozhraní

V této sekci uvedu několik konkrétních prvků uživatelského rozhraní, které jsou použity v mé práci.

Zpracování nápovědy

Známým problémem u nápovědy je fakt, že se jí uživatelé snaží za každou cenu vyhnout, a to i v případě, že neví, co mají dělat. Nápověda čistě textová sice může být výstižná, hrozí zde ale, že se po prvním pohledu na ni člověk raději rozhodne stránku opustit. Nápověda jako overlay na stránce může pro některé uživatele působit zmatečně.

Se znalostí výše zmíněné problematiky jsem se nápovědu rozhodl umístit na zvláštní stránku ještě před samotné Q-řazení, a to pod název průzkumu, jeho popis a tlačítko „Začít“. Tato volba přispěje k povědomí, že vůbec nějaká nápověda existuje, zároveň ale nebude zdržovat uživatele, kteří již s Q-řazením zkušenost mají. Hlavní částí jsou dvě animace demonstrující způsob, jakým lze Q-řazení provádět. Ty jsou doprovázeny krátkým popisem či doplněním některých vlastností (obr. 5.1).

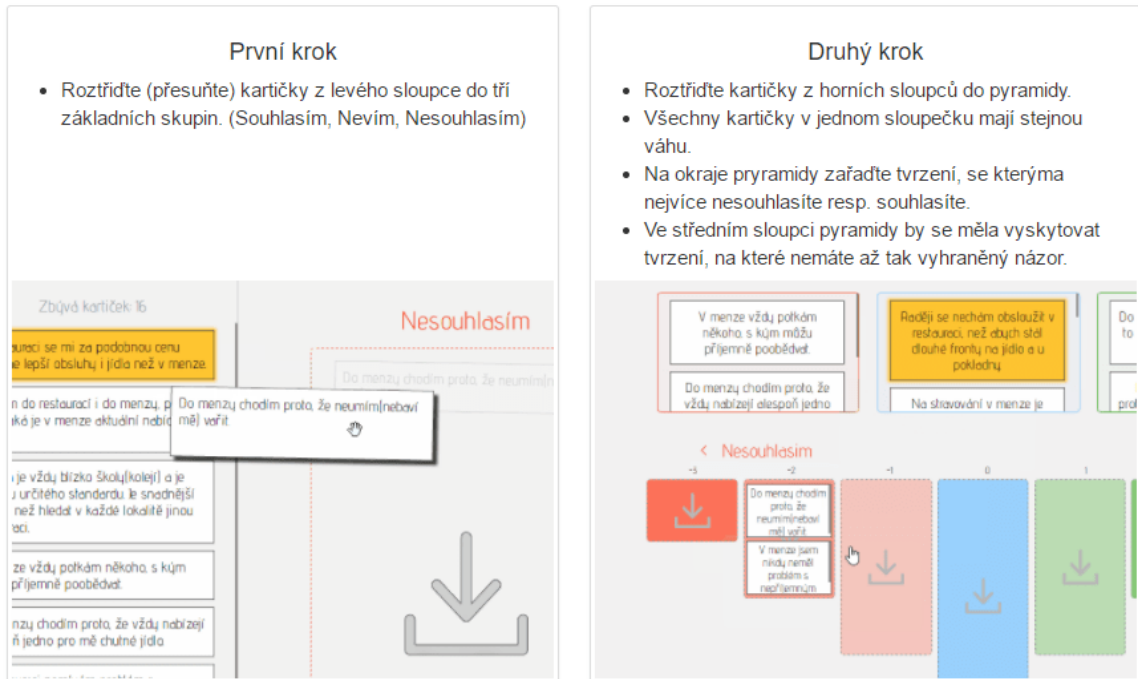
Kontejnery Drag&Drop

Pro základní kategorie a sloupce výsledkové mřížky, do kterých se kartičky řadí, jsem zvolil design typický pro funkcionalitu drag&drop. Jedná se o čárkovaný obrys, většinou ve spojení textového či obrázkového pozadí. Pro ještě větší zdůraznění sloupce jsem přidal zvýraznění obrázků v pozadí, pokud je některá z kartiček tažena, což je ilustrováno obrázkem 5.2.

Výsledková mřížka

Při návrhu výsledkové mřížky (obr. 5.3) jsem se rozhodl použít podobný přístup, jako v případě kontejnerů v prvním kroku. Ponechal jsem čárkované ohraničení s šipkou v pozadí. Sloupce jsem se rozhodl nerozdělovat na jednotlivé sloty (místa pro jednotlivé kartičky), jako tomu bylo v případě některých již hotových aplikací, ale ponechat je celistvé, což podpoří dojem rovnocennosti všech tvrzení v jednom sloupci.

Jak na to?



Obrázek 5.1: Ukázka zpracování části nápovědy.

Mřížka má tvar obrácené pyramidy. Tím se většina prostoru pro kartičky přesune blíže ke sloupcům, z nichž se kartičky přesunují. Při řazení s použitím funkcionality drag&drop se to projeví podstatným zkrácením vzdálenosti, kterou je třeba překonat k rozřídění všech kartiček. V opačném případě by respondent musel tvrzení, která chce umístit na kraje výsledkové mřížky, táhnout téměř přes celou obrazovku.

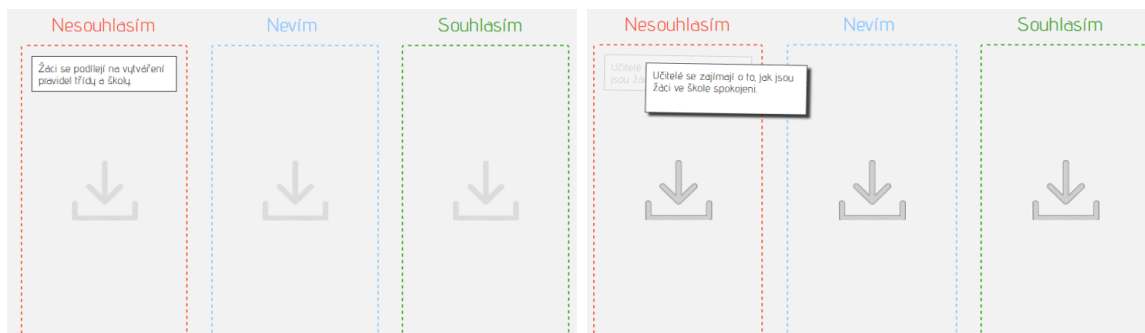
Pro odlišení, která strana pyramidy je souhlasná, a která nesouhlasná je užito třech prvků:

- popisků s šipkou
- číslování sloupců
- podbarvení mřížky

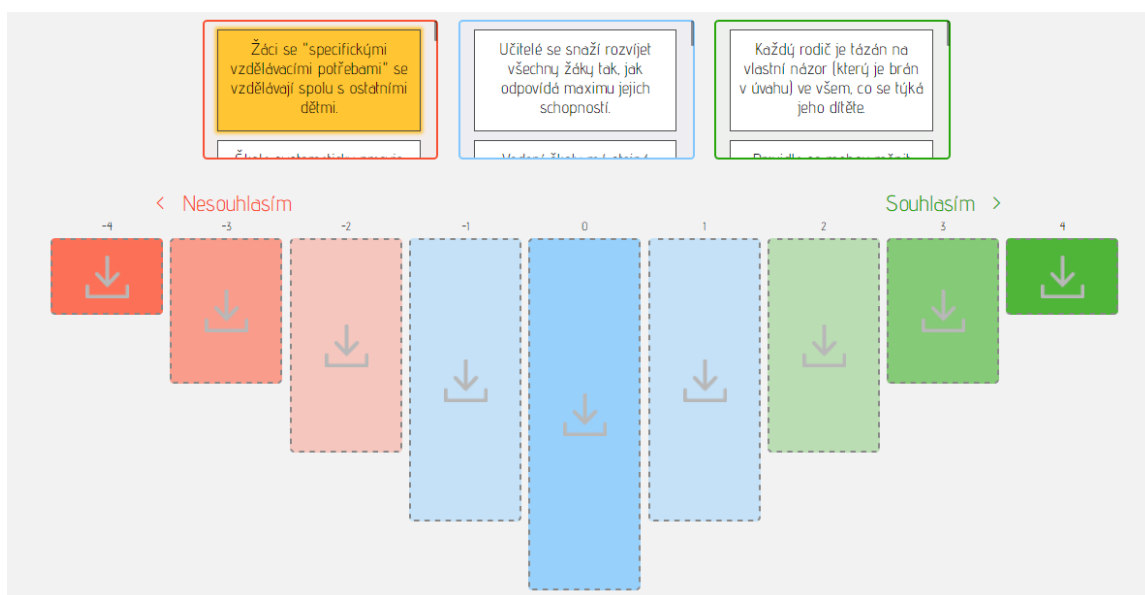
Pro pozadí sloupců v nesouhlasné části mřížky jsem použil červenou barvu, která v běžném životě reprezentuje nesouhlas, pro souhlasnou část jsem použil barvu významově opačnou, tedy zelenou. Jako neutrální barvu jsem použil barvu modrou. Sytost barvy v jednotlivých sloupcích reflektuje sílu nesouhlasu resp. souhlasu.

Zobrazení rozměrné výsledkové mřížky

V případě, že se zadavatel rozhodne pro větší počet tvrzení, nastává problém s nedostatkem místa na obrazovce a to kvůli nutnosti vykreslit rozměrnou výsledkovou mřížku. Toto nelze řešit pouhým zmenšováním mřížky, jelikož by se kartičky staly nečitelnými a sloupce příliš úzkými, což by vedlo ke snížení komfortu při řazení.



Obrázek 5.2: Vzhled drag&drop kontejnerů a zvýraznění obrázku v pozadí. Vlevo je obrázek v pozadí nezvýrazněn, vpravo (při tažení karty) je zobrazen tmavší barvou.



Obrázek 5.3: Vzhled výsledkové mřížky pro obvyklý počet kartiček.

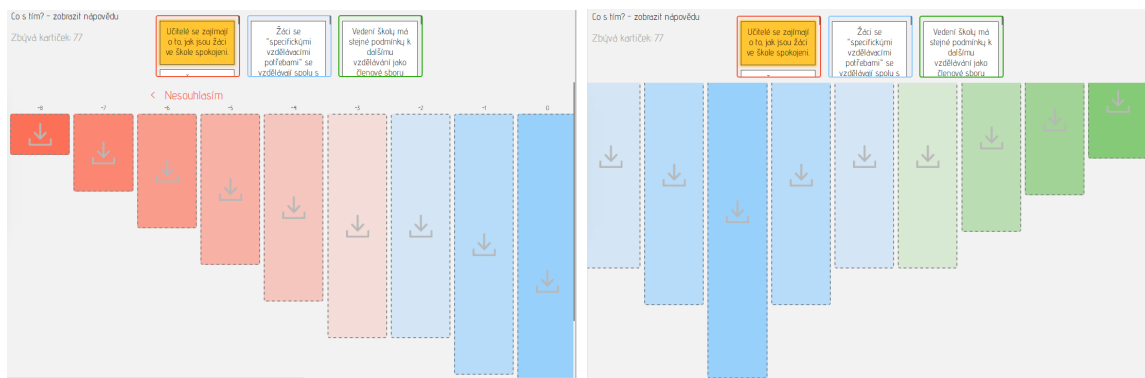
Jako vhodné řešení se osvědčilo rozdělení celé stránky na dvě části. První část, která obsahuje sloupce, ze kterých se kartičky přetahují, bude mít vždy fixní polohu na stránce. Druhá část, obsahující výsledkovou mřížku, se bude dle potřeby roztahovat. Pokud bude výsledková mřížka příliš široká nebo i vysoká, přidají se do druhé části scrollbarů, pomocí kterých se lze ve druhé části pohybovat. Tato situace je zachycena na obrázku 5.4, kde je zobrazena výsledková mřížka pro 77 kartiček.

Vzhled kartiček

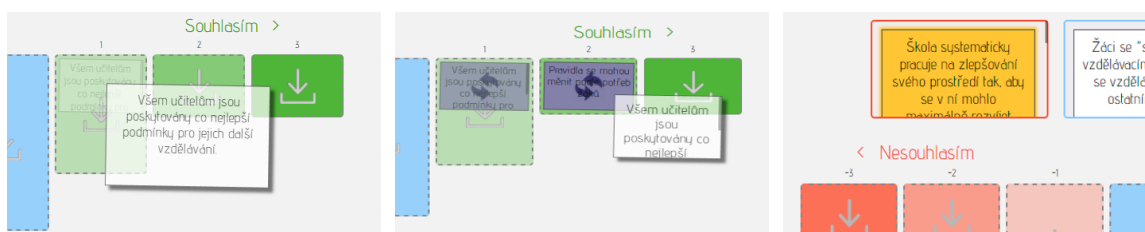
Při návrhu bylo potřeba odlišit taženou kartičku od kartiček ostatních. K tomu je využito stínování a drobné zvětšení, které dodává efekt zdvihnutí (obr. 5.5).

Dále bylo potřeba zvýraznit místo, kde tažená kartička při upuštění dopadne. K tomu je využito označení jemným stínem.

Je-li některý ze sloupců ve druhém kroku plný, nejde do něj vložit další kartička. Respondent ale může libovolnou kartičku z takového sloupce nahradit přetažením kartičky



Obrázek 5.4: Způsob zobrazení mřížky pro větší počet kartiček.



Obrázek 5.5: Vlevo tažená kartička, ve středu záměna dvou kartiček, vpravo zvýraznění kartičky pro řazení klikem.

jiné na tu nahrazovanou. Dojde ke zvýraznění obou zaměňovaných kartiček a k následnému prohození. Toto chování je zachyceno na obrázku 5.5.

Při rozhodnutí provádět Q-řazení kliknutím, se vždy přesune barevně zvýrazněná kartička (obr. 5.5). V prvním kroku je to vždy kartička na vrcholu sloupce, ve kroku druhém je výběr možno ovládat buď kliknutím na příslušný sloupeček s kartičkami k rozřazení nebo tažením některé kartičky z něj.

Animace tlačítka „upravit“

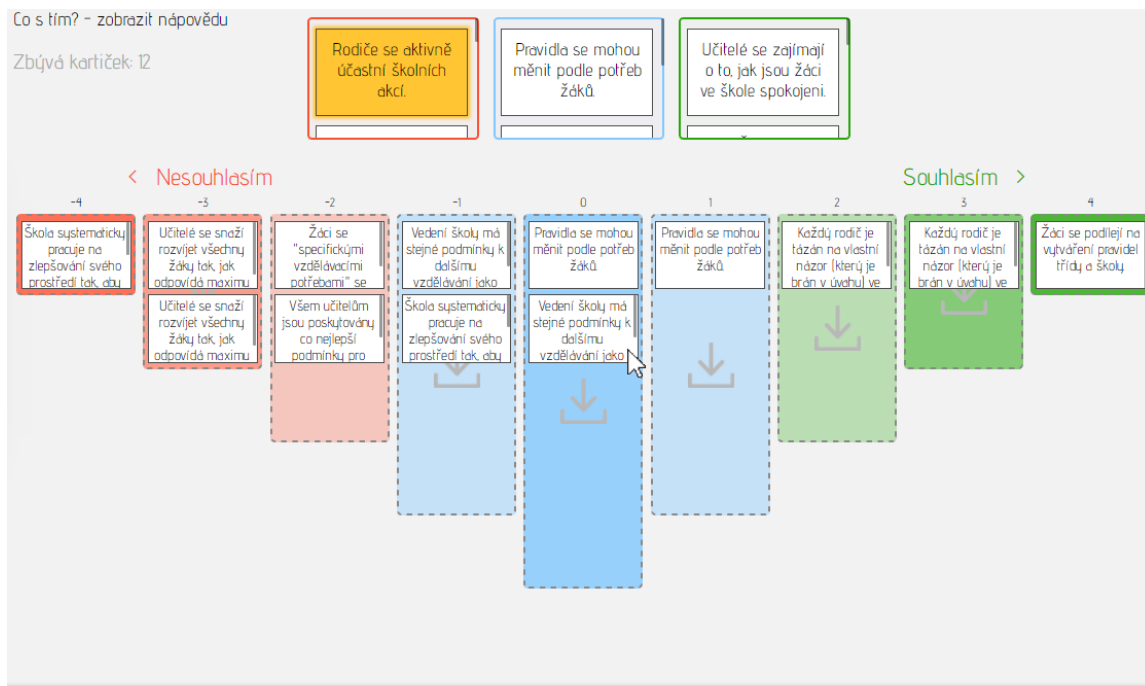
Po rozřazení všech kartiček v prvním, případně ve druhém kroku řazení se ze strany vysune nabídka, která obsahuje tlačítko pro dokončení aktuálního kroku a tlačítko „upravit“, které způsobí přesunutí této nabídky do levého horního rohu, kde nahradí odkaz na nápovědu.

Přesunutí bylo nutno realizovat pomocí animace postupného zmenšení a přemístění, aby tlačítko jednoduše nezmizelo z očí uživatele. Ten by pak nemusel vědět, jak pokračovat dále.

Zvětšování kartiček

Kartičky ve výsledkové mřížce mají pevně dané rozměry. Proto se může snadno stát, že se do nich nevejde celý text tvrzení. Další překážkou může pro respondenta být příliš malé písmo.

Problém s přetékačím textem je vyřešen jednoduchým vložením scrollbaru přímo do kartičky s tvrzením. Pro uživatele s horším zrakem je přidána dodatečná funkcionalita, která pokrývá oba výše zmíněné nedostatky zároveň. Po najetí a podržení kurzoru nad kartičkou se v dolní části obrazovky zobrazí kompletní text kartičky se zvětšeným písmem (obr 5.6).



Obrázek 5.6: Ukázka zvětšování kartiček ve druhém kroku řazení. Po podržení kurzoru nad kartičkou se její text zobrazí ve spodní části obrazovky.

Responsivita

Má práce se prioritně zaměřuje na Q-řazení prováděné na stolním počítači, nikoliv na mobilních zařízeních. Díky použití relativních jednotek na určování velikosti prvků na stránce je ale s jistými omezeními možno provádět Q-řazení se zařízeními s dotykovým displejem.

Úvodní stránka a administrace mají plnou podporu v mobilních zařízeních díky využití frameworku Bootstrap. Proces vytváření průzkumů či export výsledků by tedy měl být bezproblémový na všech zařízeních.

Kapitola 6

Implementace

V této kapitole je popsána implementace zajímavých částí práce. Zmíněn je i formát dat, ve kterém jsou uchovávány informace o průzkumech, jejich načítání a ukládání. Důležitou částí je podkapitola obsahující popis realizace bezpečnostních prvků, které zvyšují stabilitu aplikace a odolnost proti útokům.

6.1 Použití šablon

Pro některé části této práce jsem se rozhodl použít volně dostupné šablony založené na frameworku Bootstrap. Konkrétně se jedná o úvodní (informační) stránku, stránku s nápovědou a stránky, které se zobrazí po ukončení průzkumu nebo při výskytu některé možné chyby aplikace. Šablona byla také použita v administrátorské části, kde se nachází tvorba a správa průzkumů.

Důvodem jejich použití byla v první řadě jejich propracovanost a jednoduchost nasazení s relativně malou časovou náročností. Velkou roli hrála i jejich připravenost na zobrazení stránek na mobilním zařízení (responsivita). K šablonám jsem přidal vlastní CSS soubory pro drobné úpravy vzhledu a přizpůsobení některých vlastností svým požadavkům.

Pro administraci jsem použil šablonu SB Admin¹ vyvíjenou Davidem Millerem ze společnosti Blackrock Digital. Na ostatní zmíněné stránky byla využita šablona Freelancer² téhož autora.

6.2 Administrace

V této podkapitole jsou popsány detaily fungování administrace.

6.2.1 Přihlašování pomocí služby Google Sign-In

Pro získání přístupu do administrace se uživatel v první řadě musí přihlásit. Rozhodl jsem se nevytvářet přihlašování vlastní, jelikož by to pro zadavatele průzkumů znamenalo vytváření nového účtu a pamatování si dalších přihlašovacích údajů. Díky velkému počtu uživatelů služeb společnosti Google jsem pro přihlašování uživatelů zvolil autentizaci uživatelů pomocí Google účtu. V

¹Bootstrap šablona SB Admin – <https://startbootstrap.com/template-overviews/sb-admin/>

²Bootstrap šablona Freelancer – <https://startbootstrap.com/template-overviews/freelancer/>

Celý proces přihlašování probíhá prostřednictvím Google API. Jako první je potřeba vykreslit přihlašovací tlačítko pomocí příslušné knihovny. Po kliknutí na něj je zobrazeno okno, které žádá uživatele o přihlášení se ke službám Google a povolení, aby má aplikace mohla přistupovat k základním osobním údajům přihlašovaného. Je-li předchozí krok úspěšný, Google API navrátí číselný řetězec, tzv. token, který identifikuje uživatele. Ten je ale ještě z bezpečnostních důvodů potřeba ověřit na serverové části aplikace, a to tak, že se navrácený token zašle na `tokeninfo endpoint`, který jej zkontroluje a nazpět zašle nově ověřené informace o uživateli [5].

Po přihlášení je uživatel přesměrován do administrace.

6.2.2 Vytváření průzkumu

Pro vytvoření průzkumu musí zadavatel vyplnit formulář obsahující následující položky:

- název průzkumu
- název souhlasného sloupce
- název neutrálního sloupce
- název nesouhlasného sloupce
- popis průzkumu (volitelně)
- text průzkumu (tvrzení)

Text průzkumu je možno zadat dvojím způsobem. První možností je napsání nebo překopírování všech tvrzení do připraveného textového pole. Možností druhou je nahrání souboru, který tvrzení obsahuje. Povolené formáty jsou `*.txt` a `*.csv`. Každé jedno tvrzení musí pro odlišení začínat na novém řádku. Pro jejich zpracování bylo potřeba odstranit případné prázdné řádky, aby se v průzkumu neobjevily prázdné kartičky neobsahující žádné tvrzení. Rovněž jsem se rozhodl odstraňovat případné bílé znaky na začátku či konci tvrzení, aby nedocházelo k chybnému zobrazení kartiček. To bylo realizováno na straně klienta s využitím JavaScriptu.

V případě vyplnění všech povinných polí a stisknutí tlačítka „Vytvořit“, jsou data odeslána na server a uložena. Následuje přesměrování uživatele na stránku se správou jeho průzkumů.

6.2.3 Generování odkazů na průzkum

Pro každého respondenta je potřeba vygenerovat unikátní odkaz, ke kterému bude možno přidat dodatečné informace (například jméno respondenta, kterému je odkaz určen). To jsem se rozhodl udělat dvěma odlišnými způsoby.

Tím jednodušším je generování „prázdných“ odkazů, ke kterým se žádné dodatečné informace nevážou. Stačí zadat počet odkazů, které se mají vygenerovat a odeslat je manuálně respondentům.

Druhou cestou je generování odkazů ze seznamu jmen a/nebo e-mailů. Takový seznam pak stačí vložit do připraveného textového pole. Aplikace se v jednotlivých položkách seznamu pokusí rozpoznat e-mailovou adresu a jméno pomocí regulárních výrazů a následně vygeneruje tolik odkazů, kolik je v seznamu položek. Položky seznamu mohou vypadat například takto:

- Petr Novák
- pnovak@pnovak.com
- Petr Novák, pnovak@pnovak.com

Unikátnost samotného odkazu je zajištěna generováním hashe. Pro ten jsem v PHP použil funkci `md5()`³, jejímž vstupem je unikátní identifikátor, který byl vygenerován funkcí `uniqid()`⁴. Pro snížení šance na vznik dvou stejných hashů jsem do jejich vytváření navíc přidal náhodné číslo.

6.2.4 Odesílání e-mailů

Aplikace umožňuje hromadné odesílání e-mailů respondentům, u kterých je známa e-mailová adresa. Pro odeslání stačí vyplnit předmět a text zprávy, do které se místo unikátních odkazů vloží řetězec „%WSLINK%“, který je při odesílání jednotlivých e-mailů vždy příslušným odkazem nahrazen. K dispozici je i možnost pokročilejšího nastavení SMTP serveru.

Odesílat zprávy lze buď respondentům, kterým ještě žádná zpráva odeslána nebyla, nebo respondentům vybraným pomocí zaškrtačacího políčka.

K odesílání byla použita volně dostupná knihovna pro zasílání elektronické pošty PHPMailer⁵, jejímž původním tvůrcem je Brent R. Matzelle.

6.2.5 Exportování výsledků průzkumu

Zadavatel průzkumu má kdykoliv možnost exportovat výsledky průzkumu do `*.csv` souboru. Zahrnutý budou pouze zcela dokončené a roztríděné Q-sety.

Po stisknutí tlačítka „Exportovat výsledky“ se pomocí nastavení příslušných HTTP hlaviček ze serveru zašle uživateli soubor s výsledky.

6.3 Formát dat průzkumu

Text jednotlivých tvrzení bylo potřeba uložit do databáze, za účelem jeho následného načítání respondenty. K tomu jsem použil formát JSON (JavaScript Object Notation)⁶. Jednotlivá tvrzení bylo potřeba pevně svázat s konkrétní kartičkou. Výsledná data obsahující množinu dvojic kartička—tvrzení je pak v databázi uložena následovně:

```
{"card-1":"tvrzení1", "card-2":"tvrzení2", "card-3":"tvrzení3", ...}
```

Formát JSON jsem dále použil pro uložení názvů souhlasného, neutrálního a nesouhlasného sloupce, popisu průzkumu a některých dalších informací. Byť se tato volba může z pohledu návrhu databáze zdát nevhodná, poskytla mi velké množství variability a ulehčení práce.

³PHP manual, funkce `md5()` – <http://php.net/manual/en/function.md5.php>

⁴PHP manual, funkce `uniqid()` – <http://php.net/manual/en/function.uniqid.php>

⁵PHPMailer – <https://github.com/PHPMailer/PHPMailer>

⁶Úvod do JSON – <http://www.json.org/json-cz.html>

6.4 Načítání a ukládání stavu průzkumu

Z pohledu práce s daty průzkumu je databáze rozdělena na dvě tabulky. Jedna obsahuje podrobné informace o průzkumech, texty všech tvrzení a názvy sloupců. Druhá je zaměřena na uchovávání dat respondentů.

Aplikace průběžně ukládá stav roztříděných kartiček, což umožňuje respondentům proces Q-řazení kdykoliv přerušit a pokračovat později a to navíc i z jiného zařízení, jelikož se ukládaný stav váže na unikátní odkaz respondenta, nikoliv na přístroj, ve kterém Q-řazení probíhá.

Pokud respondent právě začíná s Q-řazením, není u jeho odkazu uložen žádný stav a kartičky se načtou pouze do výchozího sloupce v prvním kroku řazení. Je-li některá kartička s tvrzením přesunuta do jednoho ze tří základních sloupců, je pomocí JavaScriptu zachycen aktuální stav, který je technologií AJAX poslán na server k uložení. Pokud již respondent na Q-řazení pracoval, tak se kartičky načtou v závislosti na uloženém stavu do příslušných sloupců. Obdobně to funguje i ve druhém kroku řazení, kde se ale prvotní stav kartiček váže na konečný stav z kroku prvního.

Samotné ukládání probíhá po jakémkoliv přesunu kartičky. To s sebou samozřejmě nese větší zatížení serveru, které ale v současné době není nijak drastické. Při větším nárůstu uživatelů by bylo potřeba ukládání provádět buď v nějakém časovém intervalu nebo po přesunu určitého počtu kartiček.

Formát ukládaného stavu

Při ukládání stavu se na serverovou část aplikace nezasílá text jednotlivých tvrzení, ale pouze identifikátor kartiček, který je s tvrzeními pevně svázán, a název sloupce, ve kterém se kartička nachází. Ukládají-li se například dvě kartičky, které jsou umístěny v neutrálním sloupci, budou odesílaná data vypadat následovně:

```
{"neutral":["card-1", "card-2"]}
```

Použit je opět formát JSON. Hlavním důvodem volby tohoto řešení je jednodušší provedení kontroly konzistence dat a podstatné snížení objemu dat, který je potřeba mezi klientem a serverem přenést.

6.5 Dynamické vykreslování výsledkové mřížky

Ve druhém kroku Q-řazení je nejdůležitějším prvkem výsledková mřížka. Ta musí tvarem odpovídat rozložení podle Gaussovy křivky, které připomíná pyramidu. Mým hlavním cílem v tomto kroku řazení bylo vytvořit algoritmus, který vykreslí výsledkovou mřížku pro libovolný počet kartiček.

Prvním možným řešením bylo odvodit výšku jednotlivých sloupců výsledkové mřížky přímo z matematického vztahu pro Gaussovu křivku. Kvůli vysoké složitosti výpočtu jsem se ale rozhodl pro vlastní řešení.

Na algoritmus jsem měl několik následujících požadavků:

- výsledná mřížka se musí podobat Gaussově křivce
- první a poslední sloupec budou obsahovat pouze jednu kartičku, což vede k výběru jednoznačně nejsouhlasnějšího a nejnesouhlasnějšího tvrzení



Obrázek 6.1: Rozdílné přístupy k vykreslování mřížky. Ve výsledné aplikaci použita implementace vykreslující levou variantu, která obsahuje menší počet citově neutrálních kartiček.

- rozdíl kapacity sousedních sloupců nebude větší než 2, což může snížit podobnost Gaussově křivce, ale může zároveň zvýšit determinismus (obr. 6.1)

Výše uvedené je implementováno pomocí dvou základních funkcí. V první funkci je nejprve získána řádková reprezentace mřížky. Jedná se o pole, jehož položky jsou počty kartiček v jednotlivých řádcích. Jednoduchým příkladem může být pole [1, 3, 5], reprezentující mřížku, která má v nejvrchnějším řádku jednu kartičku, v řádku prostředním dvě a v řádku posledním tři. Tato reprezentace je z hlediska vykreslování mřížky nevhodná a je nutno ji převést na reprezentaci sloupcovou. K tomu slouží druhá funkce, která výše uvedené pole převede na pole následující: [1, 2, 3, 2, 1]. Zde už jednotlivé položky v poli určují zleva maximální počet kartiček v jednotlivých sloupcích.

Další součástí výsledkové mřížky je podbarvení, které slouží k jednoznačnějšímu odlišení citového zabarvení jednotlivých sloupců. K tomu je využito barevného modelu RGBA, kde se prostřednictvím dopočítávání alfa kanálu pro jednotlivé sloupce mění průhlednost barevného pozadí, což má za následek změny v sytosti barvy.

6.6 Bezpečnostní opatření

Při zajišťování bezpečnosti v mé práci jsem se nejprve zaměřil na kontrolu uživatelských vstupů prostřednictvím formulářů, které se kromě administrace nachází také v sekci „Kontakt“ na úvodní stránce (formulář na odeslání zprávy). Ze strany respondentů tyto kontroly odpadají, jelikož se v Q-řazení žádný formulář nenachází.

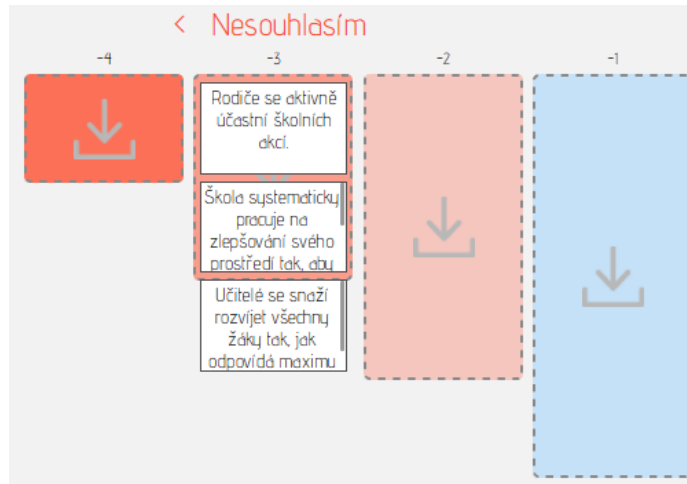
Základním pravidlem je ověřovat všechny informace obdržené od uživatele na serverové části aplikace, jelikož část klientská může být potenciálním útočníkem upravena. Zaměřil jsem se na dva druhy nejběžnějších útoků, kterými jsou SQL Injection a Cross-site Scripting (XSS).

Proti útoku na databázi (SQL Injection) jsem použil PHP funkci `mysqli_real_escape_string()`⁷, která v proměnné překóduje nebezpečné znaky, jako jsou například uvozovky. V kombinaci s ní je potřeba užít vhodný zápis proměnných v SQL dotazech, a to tak, že se uzavřou do apostrofů.

Druhý typ útoku, se kterým je třeba počítat, je Cross-Site Scripting, který funguje na bázi vložení škodlivého JavaScriptového kódu například do názvu sloupců, či popisu průzkumu při jeho vytváření. Ten by se pak mohl u respondenta místo prostého zobrazení interpretovat a spustit. Jako prevenci jsem využil funkci `htmlspecialchars()`⁸, která překóduje znaky jako například ampersandy, různé druhy závorek, uvozovky či apostrofy do

⁷PHP manual, funkce `mysqli_real_escape_string()` – <http://php.net/manual/en/mysqli.real-escape-string.php>

⁸PHP manual, funkce `htmlspecialchars()` – <http://php.net/manual/en/function htmlspecialchars.php>



Obrázek 6.2: Násilné vložení většího počtu kartiček do sloupce.

tzv. HTML entit. To zamezí interpretaci potenciálně škodlivého kódu, který se místo toho zobrazí na stránce jako prostý text.

Z pohledu bezpečnosti bylo rovněž důležité počítat s modifikací klientské části aplikace. Například ve druhém kroku Q-řazení by útočník mohl upravit zdrojový kód tak, že do sloupce určeného pro dvě tvrzení, vloží tvrzení tři (viz obr. 6.2). Takovýto stav by mohl ovlivnit výsledky průzkumu. Násilně může být upraven i identifikátor kartičky, což by mohlo vést na nekonzistenci v systému. Výše zmíněné problémy jsem ošetřil simulovaným vykreslením výsledkové mřížky na serveru a jejím následným porovnáním s příchozími daty od klienta. To je bohužel výpočetně náročná, nicméně nezbytná operace.

Kapitola 7

Testování

Pro ověření této práce a doladění uživatelského rozhraní bylo nutné provést několik testů na uživateli. Testy byly cíleny pouze na nejvytíženější část aplikace, a to na část Q-řazení. Administrace na uživateli testována nebyla.

K testování jsem přistupoval raději kvalitativně, než kvantitativně. Vedl mě k tomu fakt, že provádění testů již na šesti uživateli odhalí odhadem tři čtvrtiny všech problémů [6]. Volil jsem metodu testování použitelnosti (usability testing), což je testování přívětivosti uživatelského rozhraní, které hodnotí, jak uživatelé s aplikací interagují [9].

Výběr testerů

Při výběru testerů jsem se snažil pokrýt různé věkové skupiny a různá pracovní či studijní zaměření. To vedlo na širší spektrum odhalených nedostatků uživatelského rozhraní.

Celkem se testování zúčastnilo 14 osob:

- do 25 let – 6 osob
- 25-40 let – 3 osoby
- 40-60 let – 4 osoby
- nad 60 let – 1 osoba

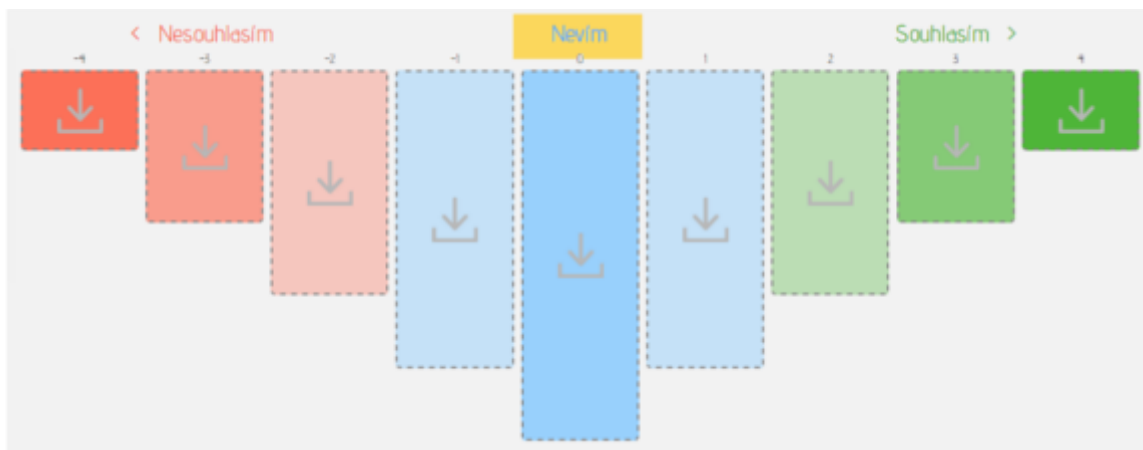
Průběh testování

Před začátkem testování byli testeři požádáni o „přemýšlení nahlas“ a ignorování mé přítomnosti. Poté jim byl poskytnut odkaz pro zahájení Q-řazení. Kromě toho jim nebyly poskytnuty žádné bližší informace o principu fungování programu.

K testování jsem použil dvě různé sady tvrzení v závislosti na preferovaném tématu daného testera, obě o méně než 30 tvrzeních. Větší počet mi přišel pro otestování uživatelského rozhraní zbytečný a časově náročný.

Při průběhu testů jsem přímým pozorováním testerů odhaloval možné nedostatky uživatelského rozhraní. Po ukončení testů vždy následovala diskuze o průběhu Q-řazení a nápadech na případné zlepšení.

Testování bylo prováděno na notebooku se 17" displayem. Testerům byla k dispozici myš i touchpad.



Obrázek 7.1: Původní návrh výsledkové mřížky s popisem neutrální části.

Výsledky testování

Pozitivním zjištěním bylo, že se všem testerům podařilo bez mé pomoci dokončit celý proces Q-řazení, z toho jedenácti bez většího zaváhání. Dobře se osvědčilo umístění nápovědy před samotné řazení a její animované provedení. Nápovědu si před začátkem řazení prošla polovina osob, což považuji za nadprůměr.

Z testování vyplynulo i několik nedostatků, které bylo potřeba odstranit. Někteří testeři se v průběhu řazení vraceli k nápovědě pomocí tlačítka „zpět“ v prohlížeči, což kvůli kešování stránek do paměti způsobovalo chybné načítání dat průzkumu. Řešením bylo přidání následujících HTTP hlaviček na vypnutí této funkcionality:

```
header("Cache-Control: no-store, no-cache, must-revalidate,
max-age=0");
header("Cache-Control: post-check=0, pre-check=0", false);
header("Pragma: no-cache");
```

Drobný problém nastal i u tlačítek pro započítání nebo ukončení průzkumu. Některé byly zobrazovány jako odkaz bez ohraničení, jiné ohraničení měly, ale kliknout se dalo pouze na text uvnitř. Na základě toho došlo k sjednocení všech tlačítek do optimální podoby.

Původní návrh výsledkové mřížky ve druhém kroku obsahoval i popis střední části, jak znázorňuje obrázek 7.1 (popisek je na obrázku zvýrazněn). Ten v několika testech vzbuzoval potřebu vměstnat všechny neutrální kartičky z prvního kroku do modré části mřížky, i když to mnohdy nebylo možné. To vedlo na odstranění tohoto popisku.

Ve druhém kroku dále nastal problém u zvětšování kartiček, jež bylo původně realizováno dvojitým kliknutím. To někdy způsobovalo nechtěné rozřazení kartičky. Proto byl implementován stávající mechanismus zvětšování pomocí podržení kurzoru na příslušné kartičce.

Kapitola 8

Závěr

V rámci této bakalářské práce jsem navrhl a naimplementoval webový nástroj pro Q-řazení. Poté jsem ověřil jeho použitelnost testováním, na základě kterého jsem navíc provedl několik úprav a vylepšení.

Pro úspěšné vytvoření tohoto nástroje bylo potřeba nastudovat principy Q-metodologie, zejména pak fungování Q-řazení. Dále bylo potřeba se seznámit s principy tvorby uživatelských rozhraní a metodami používanými pro jeho testování.

Dalším krokem bylo provedení analýzy již existujících aplikací, které Q-řazení realizují. Z nich jsem vybral několik prvků, které mi přišly pro vlastní řešení zajímavé. Důležité bylo rovněž nalezení chyb a nedostatků těchto aplikací s cílem se jim ve své práci vyhnout.

Následně jsem zvolil vhodné technologie pro implementaci nástroje realizujícího Q-řazení, provedl jeho návrh a na jeho základě jsem nástroj úspěšně naimplementoval. Poté jsem prováděl testy uživatelského rozhraní, díky kterým jsem opravil několik chyb.

Výsledkem je funkční webový nástroj pro testování pomocí Q-řazení s jednoduchým uživatelským rozhraním a designem odpovídajícím dnešním trendům. Součástí nástroje je i administrace, která umožňuje vytváření nových průzkumů, jejich správu, a export výsledků dokončených průzkumů. Vytvořený nástroj je dostupný na adrese <http://websort.oksis.cz>.

Literatura

- [1] Baker, R.: Rachel Baker introduces Q Methodology. Leden 2016.
URL <https://qmethod.org/2016/01/08/rachel-baker-introduces-q-methodology/>
- [2] Brown, T.; Butters, K.; Panda, S.: *HTML5 okamžitě*. Computer Press, 2014, ISBN 9788025142967.
- [3] Coogan, J.; Herrington, N.: Q methodology: an overview. *RESEARCH IN SECONDARY TEACHER EDUCATION Vol.1, No.2.*, 2011.
- [4] Cross, R. M.: Exploring attitudes: the case for Q methodology. *Health Education Research*, ročník 20, č. 2, 2005: str. 206, doi:10.1093/her/cyg121.
- [5] Google Sign-In for Websites. Google, [Online; navštíveno 11. 5. 2017].
URL <https://developers.google.com/identity/sign-in/web/>
- [6] Nielsen, J.: Why You Only Need to Test with 5 Users. Březen 2010.
URL <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>
- [7] Nielsen, J.: How Long Do Users Stay on Web Pages? Září 2011.
URL <https://www.nngroup.com/articles/how-long-do-users-stay-on-web-pages/>
- [8] Rozalia, G. M.: Q FACTOR ANALYSIS (Q-METHODOLOGY) AS DATA ANALYSIS TECHNIQUE. 2008.
- [9] Why You Only Need to Test with 5 Users. Software Testing Class, Únor 2016.
URL <http://www.softwaretestingclass.com/wp-content/uploads/2016/02/Beginner-Guide-To-Software-Testing.pdf>
- [10] Introduction to User-Centered Design. Usability First, [Online; navštíveno 11. 5. 2017].
URL <http://www.usabilityfirst.com/about-usability/introduction-to-user-centered-design>
- [11] Návrh uživatelského rozhraní webové aplikace. Vysoká škola ekonomická v Praze, [Online; navštíveno 11. 5. 2017].
URL <http://gml.vse.cz/data/oppa-webdesign/ui.html>

Přílohy

Příloha A

Obsah CD

Příložené CD obsahuje následující adresáře a soubory:

- `www/` – zdrojové soubory webového nástroje pro testování pomocí Q-sort
- `tz-src/` – zdrojové texty a obrázky technické zprávy pro \LaTeX
- `zprava.pdf` – technická zpráva
- `plakat.pdf` – prezentační plakát
- `video.avi` – prezentační video
- `README.txt` – textový soubor s popisem obsahu CD

Příloha B

Plakát

