Masarykova univerzita
Fakulta informatiky

Platforma pro crowdsourcing na MU

Diplomová práce

Bc. Vít Hovězák

Brno, jaro 2018

Prohlášení

Prohlašuji, že tato diplomová práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Bc. Vít Hovězák

Vedoucí práce: RNDr. Petra Budíková, Ph.D.

Poděkování

Velice rád bych poděkoval všem, kteří mi pro tuto práci ochotně poskytli jakékoli cenné rady. Především paní RNDr. Petře Budíkové, Ph.D., díky které tato práce mohla spatřit světlo světa. V neposlední řadě děkuji také své rodině a přátelům za neustálou podporu.

Shrnutí

Tématem práce je analýza, návrh a implementace prototypu crowdsourcingové platformy pro Masarykovu univerzitu. Teoretická část představuje problematiku crowdsourcingu a porovnává existující marketingové platformy pro zprostředkování práce mezi zadavateli a širokou veřejností. Ve druhé části jsou analyzovány potřeby univerzity na základě srovnání s existujícími platformami, a popsán návrh typů úloh a techniky pro hlídání kvality výsledků. Praktickým výstupem diplomové práce je prototypová implementace platformy.

Klíčová slova

crowdsourcing, spolupráce, testování, kontrola kvality, odměňování, motivace, Amazon Mechanical Turk, MUFIN

Obsah

1 Úvod **1**
1.1 *Proč speciální aplikace pro MU* . . . 2
1.2 *Cíle práce* . . . 3

2 Crowdsourcing **5**
2.1 *Příklady projektů* . . . 6
2.2 *Hlavní principy* . . . 7
2.3 *Problematické aspekty* . . . 11
2.4 *Algoritmy pro kontrolu kvality* . . . 13

3 Existující platformy **17**
3.1 *Amazon Mechanical Turk* . . . 17
3.2 *Clickworker* . . . 23
3.3 *Microworkers* . . . 29
3.4 *Další platformy* . . . 32

4 Crowdsourcing pro MU **34**
4.1 *Analýza potřeb* . . . 36
4.2 *Případy užití* . . . 41

5 Implementace **45**
5.1 *Návrh systému* . . . 46
5.2 *Struktura aplikace* . . . 50
5.3 *Jednotné přihlašování* . . . 54
5.4 *Testování* . . . 55

6 Závěr **57**
6.1 *Naplnění cílů* . . . 58
6.2 *Vize budoucího rozvoje* . . . 59

Bibliografie **60**

A Přílohy **62**
A.1 *Případy užití* . . . 62
A.2 *Obsah přílohy na CD* . . . 79
A.3 *Postup nasazení* . . . 80

A.4 *Struktura dat ve formátu JSON* 82
A.5 *Výsledky experimentů* 83
A.6 *Uživatelské rozhraní* 85

1 Úvod

Všichni se zajisté dokážeme ztotožnit s myšlenkou, obsaženou ve známém českém přísloví „víc hlav víc ví“, a tím pádem nám principy crowdsourcingu nebudou nikterak cizí. Jedná se o metodu, při které využijeme objemnou skupinu lidí pro řešení problému, který by samotný jednotlivec nebyl schopný splnit, případně by trval neúměrně dlouhou dobu. Problém však nemusí být nutně komplikovaný, ale spíše velmi časově náročný, nebo vyžadující širokou škálu nezávislých pohledů.

V dnešní době čím dál častěji narážíme na složité problémy, v jejichž rámci je nutné řešit stále větší množství jednodušších dílčích úkolů a zpracovávat obrovské množství dat. Díky současným technologiím jsme schopni spoustu těchto rutinních operací provádět automatizovaně, nicméně ne vždy je možné lidský faktor nahradit. V takových případech obvykle jeden člověk na vše nestačí. Musíme tedy k provedení práce zaměstnat velké množství lidí, což je základní myšlenkou crowdsourcingu. Vzájemná výpomoc a rozdělování prací nás bude provázet celou touto prací.

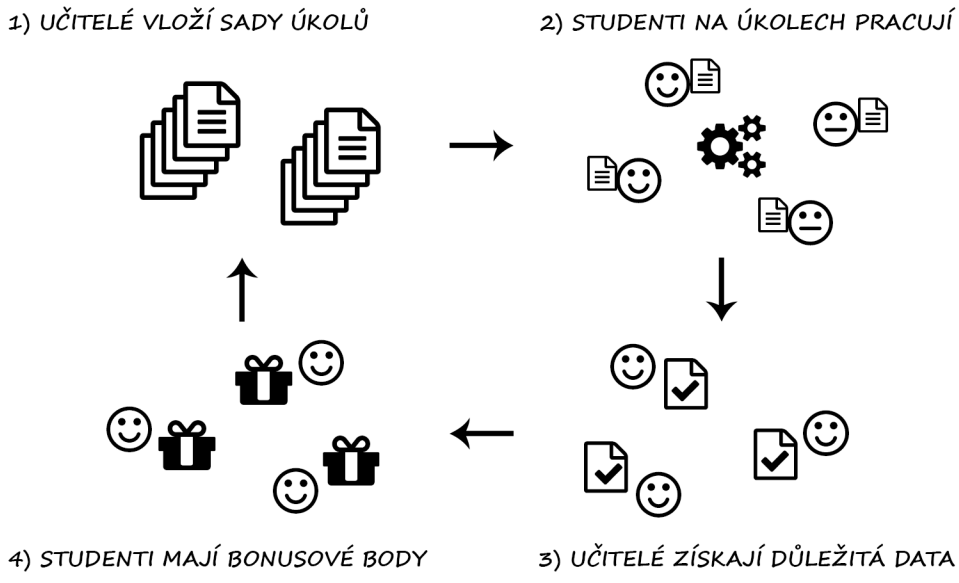
Novotvar crowdsourcing nemá v češtině odpovídající ekvivalent, nicméně je možné jej v jednoduchosti definovat jako „řešení problému s pomocí davu“. Jeho hlavní myšlenkou je využití ochoty lidí podílet se na společném budování hodnot, a to bez závislosti na konkrétním místě, čase nebo na profesním zaměření dobrovolníka. Objevuje se tak možnost třeba i jen částečného přenesení tradičního způsobu práce ze stálých zaměstnanců na mnohdy anonymní jednotlivce z početné skupiny lidí.

Máme tedy šanci využít síly a vědomostí davu k plnění rutinních úkolů, které by za normálních okolností museli zastat jednotlivci. Tito pracovníci by však museli být pro takovou činnost vyčleněni na daleko delší dobu. Výstižné přirovnání „Crowdsourcing je outsourcing na steroidech“ použil pro výše uvedený princip Jeff Howe ve své knize „Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business“ [1].

1.1 Proč speciální aplikace pro MU

V prostředí Masarykovy univerzity existují úlohy a problémy, které jsou pro akademickou půdu specifické, a potřebujeme je řešit. Zároveň se v univerzitním prostředí nachází spousta kandidátů na pracovníky, a to zejména z řad studentů. Existují sice crowdsourcingové platformy, kde je možné zadat úkol a zaplatit za jeho vyřešení, ovšem ne vždy má zadavatel na takové řešení peníze. Musíme tedy hledat jiné možnosti odměňování za provedenou práci, abychom dosáhli dostatečné míry motivace pro zapojení a také udržení studentů.

Takovou motivací mohou být např. bonusové body do studovaných předmětů které by zajistily lepší výsledné známky, případně jiné studijní ohodnocení, odpovídající přínosu odvedené práce. Aby celý proces mohl fungovat, potřebujeme tedy specializované řešení pro Masarykovu univerzitu. Není však podstatné, jestli bude vyvinuta vlastní platforma, nebo adaptace existujícího systému. Systém by měl nabídnout možnost zapojení studentů do probíhajících výzkumů či experimentů, a zároveň vylepšení jejich studijních výsledků.



Obrázek 1.1: Princip aplikace do prostředí MU

Tímto se dostáváme k problematickému bodu crowdsourcingu, který je také jedním ze stěžejních témat této práce – „Jak je při potenciálně vysoké kvantitě získaných výsledků možné kontrolovat jejich kvalitu?“ Je velmi pravděpodobné, že se část respondentů bude snažit minimalizovat úsilí k získání co nejvyššího bodového profitu. Společně s motivací účastníků se jedná o hlavní součásti jakéhokoli systému pro zapojení velkého množství lidí a tvůrce by v jejich možných řešeních měl mít jasno již během navrhování nové platformy.

S tímto vědomím budeme k problému přistupovat i při tvorbě systému pro Masarykovu univerzitu, kde může aplikace, založená na principech crowdsourcingu, přispět k podpoře vyhodnocování experimentů v rámci výzkumu. Podpora různých oblastí bádání je neustále žádoucí, a každá nová efektivní metoda přináší na akademickou půdu další možnosti rozvoje. Často lze však v praxi sledovat situaci, kdy jsou aplikace pro přemíru funkcí a nástrojů velice těžkopádné a složitě použitelné, proto se v tomto případě budeme při její tvorbě snažit rozumně zachovat jednoduché uživatelské rozhraní.

Abychom byli schopni uspokojit požadavky všech uživatelských skupin, musíme si nejdříve uvědomit, kdo bude s naší aplikací pracovat. Na jedné straně to budou zadavatelé, tedy lidé z různých fakult univerzity, kteří budou moci definovat jednotlivé typy úkolů a nastavit motivační odměny. Převážně zde tedy budou učitelé nabízet body do svých předmětů. Obecně však může úkoly zadávat kdokoli z MU, nicméně účastníky si pak musí motivovat sám, jiným způsobem než bonusovými body. Na straně druhé se budou nacházet pracující, tedy studenti vybraných předmětů, případně jiní zájemci o spolupráci, kteří touží po nabízené odměně. V praxi se ale ukazuje, že často stačí pouze dobrý pocit ze společně přidané hodnoty.

1.2 Cíle práce

Pro přehlednější plnění cílů je vhodné si požadované kroky definovat a rozdělit do menších částí:

- Prozkoumat známé crowdsourcingové platformy a pokud existuje, najít nástroj využitelný pro naše účely.

- Na základě existujících platforem a potřeb univerzity zvolit podporované typy úloh a způsob jejich zadávání.
- Vybrat nebo navrhnout techniky pro hlídání kvality výsledků a v případě potřeby je rozdělit podle typu úlohy.
- Navrhnout základní objekty a procesy crowdsourcingové aplikace, případně i možnosti komunikace s externími službami.
- Navrhnout uživatelské prostředí pro zadávání úloh včetně vhodného šablonového systému pro definování prvků aplikace.
- Zvolit technologie pro implementaci, ať už z osvědčených programovacích jazyků a nástrojů, nebo z vhodných alternativ.

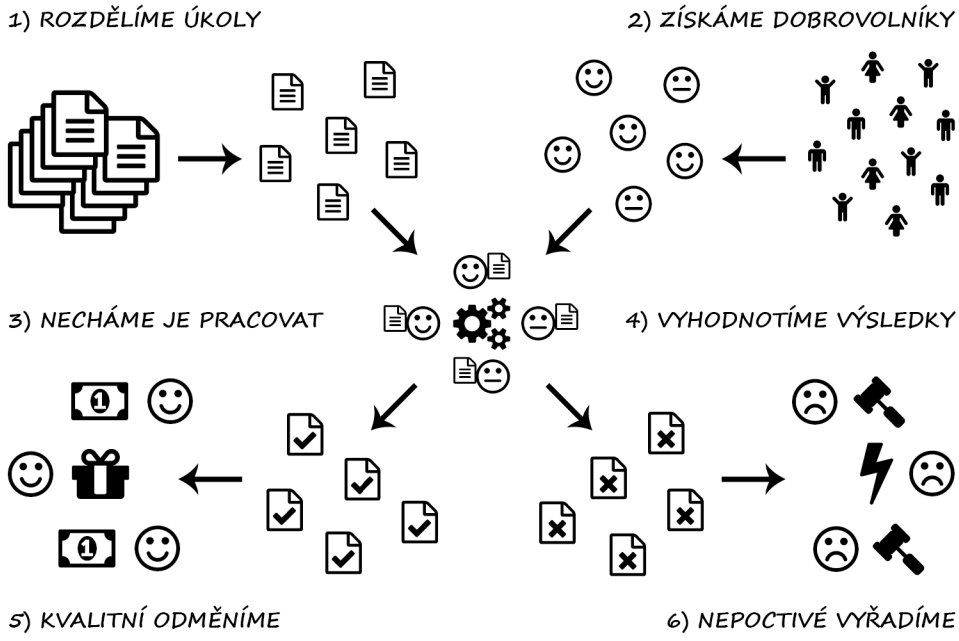
Jednotlivé body jsou rozpracovány v následujících kapitolách. Kapitola 2 popisuje základní principy crowdsourcingu, jeho problematické aspekty, a techniky pro hlídání kvality výsledků. Následující kapitola 3 se podrobně věnuje analýze vybraných existujících crowdsourcingových platforem se zaměřením na jejich funkce, typy úloh a proces zadávání. Kapitola 4 popisuje hlavní cíle platformy pro univerzitní prostředí, způsob jejího využití a definici vhodných typů úloh. Důležité je zde také definovat jednotlivé uživatelské role a motivační systém odměňování.

Kapitola 5.1 se věnuje návrhu výsledné aplikace se všemi procesy či modely, a to včetně rozložení uživatelského prostředí, nebo také externí komunikace. Kapitola 5.2 poté řeší výběr vhodného programovacího jazyka, komponent pro různé vrstvy aplikace a zejména implementaci s pomocí těchto zvolených technologií. V následující kapitole 5.4 ověřujeme funkčnost aplikace pomocí několika experimentů. Nedílnou součástí je také závěrečná kapitola 6, popisující budoucí možnosti rozvoje.

2 Crowdsourcing

Definovat pojem „crowdsourcing“ není snadné. Nejdříve se o to pokusil již v roce 2005 Jeff Howe společně s Markem Robinsonem. Nový způsob spolupráce označili jako „outsourcing to the crowd“, z čehož následně vznikla složenina „crowdsourcing“. První zveřejněná definice se objevila ve článku „The Rise of Crowdsourcing“ [2]:

„Zjednodušeně definováno, crowdsourcing představuje počin firmy nebo instituce, kterým dojde k předání funkce dříve vykonávané jejími zaměstnanci k outsourcingu nedefinovanou (a obecně velkou) sítí lidí ve formě otevřené výzvy. To může mít formu samo organizované produkce (pokud je práce kolektivní), ale často je také prováděna samostatnými jednotlivci. Rozhodující podmínkou je použití otevřené formy výzvy a velká síť potenciálních pracovníků.“ (překlad vlastní)



Obrázek 2.1: Zjednodušený princip crowdsourcingu

2.1 Příklady projektů

Pro vytvoření představy o praktickém využití této techniky nám nejlépe poslouží příklady z prvopočátků crowdsourcingu [3, 4]. Některé poměrně malé projekty se velice rychle vyvinuly do silné a ekonomicky stabilní společnosti, což je potvrzením tohoto „davového" potenciálu. Působivou ukázkou je služba pro výrobu triček a jiného zboží s vlastním potiskem – threadless.com. Registrovaní uživatelé mohou nejen vytvářet vlastní motivy pro tisk, ale také hodnotit ostatní návrhy a tím ovlivňovat, které se dostanou do výroby a které ne. Každý týden je zveřejněno okolo jednoho tisíce návrhů k ohodnocení, po kterém je deset nejúspěšnějších návrhářů finančně odměňováno. Společnost svoji nabídku od roku 2000 neustále rozšiřuje a její uživatelé zůstávají spokojení. Názorně zde vidíme sílu samovolně se utvářejícího trhu, který si poptávku i nabídku reguluje dle aktuálních trendů.

Za zmínku stojí také tržiště pro drobné IT práce – topcoder.com, kde může každý zájemce získat práci od profesionálů z davu. Na druhé straně mohou pracovníci najít své uplatnění díky úspěchu při drobných soutěžích v programování nebo i v grafickém designu. Jiným příkladem může být platforma společnosti Dell – ideastorm.com, která umožňuje vkládat nápady pro vylepšení produktů. Díky hlasování celé komunity jsou některé myšlenky postupně zapracovávány do firemní nabídky. Podobnou strategii zvolila i společnost Starbucks – mystarbucksidea.com a umožňuje tak svým zákazníkům přenést jejich nápady do reality.

Ne vždy ale narazíme pouze na úspěšné projekty, a tak mezi množstvím pokusů o inovaci a prosazení na trhu najdeme i neúspěchy. Hned první známý příklad se objevil ve státní správě – nekomerční platforma „White House Open for Questions" je ukázkou toho, co se stane při nesprávném uchopení takto silného nástroje. Hlasování o nejdůležitějších otázkách veřejnosti se může zdát jako dobrá myšlenka, nicméně již brzy se objevily problémy s jeho zneužitím, kdy neznámý útočník zmanipuloval výsledky tak, aby se do popředí hlasování dostaly dotazy, týkající se legalizace marihuany.

Pokud se podíváme na crowdsourcing z pohledu uživatelem generovaného obsahu, narazíme na internetovou encyklopedii Wikipedia – wikipedia.com a portál pro vkládání videí YouTube – youtube.com. Obě služby dosahují vysoké míry oblíbenosti už od svého založení,

a zejména počet uživatelů YouTube je v dnešní době jedním z nejrychleji rostoucích. Velkou výhodou Wikipedie je snadná forma přispívání – není nutné přidávat nové informace, ale kdokoli může přispět alespoň překladem existujících článků do jiného jazyka. Uživatelská tvorba je okamžitě po vzniku veřejně dostupná, což má výhodu aktuálních dat a flexibility, ale zároveň také nese riziko chybného, nebo nevhodného obsahu. I když můžeme často pochybovat o užitečnosti poskytnutých informací nebo videí, oba portály jsou dnes jen těžko postradatelné.

2.2 Hlavní principy

I přes výrazné výhody a zdánlivou přímočarost crowdsourcingu, se mnoho projektů v této oblasti dostalo do existenčních problémů a neuspělo. Jedná se totiž stále o poměrně nový přístup, který se neustále vyvíjí a nemá pevně daná pravidla. Jeff Howe [1, p. 280] se tedy pokusil stanovit několik základních principů, které sice nejsou vše spásné, ale mohou poskytnout alespoň hrubý základ pro orientaci v této technice.

1. „Volba správného modelu“ Pojem crowdsourcing pod sebou skrývá rozmanitou sadu přístupů a strategií, které mají jedno společné – na všech se musí podílet dav. V ostatních vlastnostech se však mohou výrazně lišit, nicméně základem je vždy stanovení hlavního cíle. Ten nám pomůže při rozhodování a správném výběru crowdsourcingového modelu. U úspěšných projektů se často setkáváme s jejich kombinací, případně s jinými variantami, ale nejčastěji je možné pozorovat následující čtyři modely:

- Kolektivní inteligence („hledání nápadu“) – Důležité je vytvořit takové podmínky, aby se mohly projevit vědomosti celé skupiny. Některé společnosti proto vytvářejí propojení mezi jednotlivci, jiné mají u svých produktů jen jednoduchou možnost textového doporučení na vylepšení.

- Davová tvorba („vytváření produktu“) – V některých případech stačí, pokud lidem umožníme realizovat jejich nápady s rozumnou šancí na úspěch. Rádi se pak budou podílet např. na vy-

tvoření televizní reklamy nebo zvukové nahrávky, případně jen přispět k vylepšení služby pomocí jazykových překladů.

- Hromadné hlasování („řazení výsledků“) – Využíváme zde úsudku velkého množství lidí k organizaci nebo filtrování libovolných dat. Není však nutné nechat skupinu hlasovat přímo, ale např. webové vyhledávače využívají klikání na užitečné výsledky k jejich lepšímu řazení, případně zpravodajské portály nabízející nejpopulárnější články podle jejich čtenosti.

- Crowdfunding („skupinové financování“) – Poslední model využívá potenciálu velkého davu lidí, aby získanými penězi umožnil nahradit velké finanční instituce. Příkladem je portál pro poslech hudby, který začínajícím kapelám nabízí prostor k prezentaci a uživatelé jim dle zájmu mohou přispívat. Po získání odpovídajícího množství financí vydá hudební producent nové album.

2. „Volba správného davu“ Počet lidí s přístupem k internetu tvoří větší polovinu celosvětové populace a neustále stoupá, což nám poskytuje více než 4 miliardy potenciálních uživatelů[1]. Přitom optimální velikost uživatelské základny je podle Alpheuse Binghama[2] pouze okolo pěti tisíc lidí – najít však ty správné uživatele ale není lehké. Pokud tedy vyvíjíme službu pro hodnocení fotbalových trenérů, pak skupina s deseti tisíci lidmi bez zájmu o daný sport nebude pro naše potřeby ideální. Naopak je dobré, abychom hledali v místech s vysokou koncentrací jednotlivců z cílové skupiny.

3. „Nabídka dobré motivace“ Jádrem úspěchu u crowdsourcingového projektu je živá a angažovaná komunita, přičemž zapojení lidí vyžaduje porozumění jejich potřebám a zvolení dobré motivace. Pro někoho je důležitý kontakt s podobně smýšlejícími lidmi, pro jiného možnost rozvoje svých dovedností, ale nesmíme opomenout ani finanční odměny. Jiným způsobem můžou uživatelé vnímat neziskový projekt oproti čistě komerčnímu. Zatímco v prvním případě mnoho z nich daruje svůj čas ochotně, v druhém se bez odpovídající odměny

[1] https://www.internetworldstats.com/stats.htm

[2] Alpheus Bingham je zakladatel crowdsourcingové platformy innocentive.com.

může stát, že nabudou určitý pocit zneužití, což se může velice rychle obrátit proti zadavateli.

4. „Volba mezi davem a zaměstnanci“ Mohlo by se zdát logické, že tedy není nutné platit stálé zaměstnance v případě, kdy nám dav provede práci levněji, nebo dokonce zdarma. Např. společnost Heinz si v praxi ověřila, že tento na první pohled levný způsob propagace může být ve výsledku mnohem dražší[1]. Firma nejprve nechala novou televizní reklamu na kečup vytvořit samotnými uživateli, nicméně vzhledem k množství výsledků se ukázalo, že je nutné na hodnocení najmout specializovanou reklamní agenturu. Její zaměstnanci pak byli natolik vytíženi porušováním autorských práv a nekvalitními či urážlivými pokusy, že se soutěž protahovala a společnost Heinz navíc získala i spoustu negativní reklamy.

5. „Němota davu – princip benevolentního diktátora“ Na jednu stranu jsou lidé schopni výrazné míry sebe organizace (např. rozdělit se podle zájmů a schopností), na stranu druhou dav často vyžadoval pomoc a nasměrování v případě nejasností. Proto je také často nutné vytvořit administrační vrstvu z dobrovolníků, kteří se budou věnovat moderaci a odpovídání na různorodé otázky. V praxi tedy nejúspěšnější projekty vznikají díky všestranné spolupráci mezi tvůrčími jednotlivci a tzv. „benevolentními diktátory“, kteří umí dav efektivně směrovat k lepším výsledkům. Tento princip se osvědčuje např. při komunitním vývoji open source projektů.

6. „Zachování jednoduchosti“ Podobně jako při vytváření softwaru, každou úlohu vhodnou pro crowdsourcing je nutné nejprve rozložit na co nejmenší části. Pokud dosáhneme dobré míry modularity, nejenže tím připravíme snadnější a lákavější podmínky pro dav, ale také zadavatel bude později schopný lépe zasadit klíčové části do jednoho celku. Je důležité zejména zachovat jednoduchou povahu dílčích úkolů – pokud všem umožníme najít si vlastní malou část, kde bude každý moci přispět, výrazně tím zvýšíme pravděpodobnost, že k tomu opravdu dojde. Jako příklad může sloužit projekt threadless.com, kde

[1] https://www.nytimes.com/2007/05/26/business/26content.ready.html

může každý ohodnotit libovolné množství potisků triček, ale může také vytvořit vlastní návrh.

7. „Sturgeonův zákon“ Vzhledem k potenciálnímu objemu davu může být množství příspěvků ohromující. Není tedy možné očekávat, že většina lidí dodá stoprocentní kvalitu. Sturgeonův zákon nám říká, že „z devadesáti procent je výsledek nekvalitní“, což může být v některých případech dokonce i poměrně nízký odhad. Vzhledem k rozvoji technologií je dnes téměř každá znalost nebo schopnost dostupnější, nicméně talent bude vždy vzácnou komoditou. Crowdsourcing se však snaží o nalezení, a hlavně projevení skrytého talentu, což je tématem dalšího bodu.

8. „Využití kvalitních pracovníků“ Existují minimálně dva neocenitelné přínosy crowdsourcingu. Prvním je ten, že všem poskytuje místo pro usilování o úspěch v dovednostech, pro které nutně nemusíme mít nadání, ani potřebné schopnosti. Druhým je fakt, že dav umí spoustu nevydařené a nekvalitní tvorby eliminovat. Automaticky tak vzniká mechanismus, který zabraňuje zahlcení projektů nepřeberným množstvím nepoužitelného materiálu. Díky demokratické volnosti tak mohou samotní pracovníci oddělit dobré výsledky od špatných.

9. „Komunitní volba“ Jako dodatek k principu benevolentního diktátora je dobré podotknout jeden fakt. Je rozhodně dobré snažit se komunitu posunovat správným směrem a poskytovat lidem pomocnou ruku. Nicméně nakonec i tak zjistíme, že názor davu má přednost. Docházíme tedy k závěru, že autorita diktátora v tomto kontextu nemusí být absolutní.

10. „Vycházení vstříc davu“ Crowdsourcing pracuje nejlépe v takovém případě, kdy jednotlivec nebo firma poskytne davu to, co sám potřebuje. Je tedy možné říci, že úspěšný projekt zahrnuje uspokojení nejvyššího stupně Maslowovy pyramidy lidských potřeb[1] [1, s. 288],

[1] Hierarchické uspořádání od nejzákladnějších lidských potřeb po ty individuální. Uspokojením základní potřeby přichází touha po potřebách na vyšším patře pyramidy.

a to pro každého jedince, kterého plánujeme oslovit. Firmám tím pádem nezbývá nic jiného, než změnit způsob nahlížení na vztah se zaměstnanci a poskytnout jim takové podmínky, jaké by oni sami akceptovali.

2.3 Problematické aspekty

Crowdsourcing se za roky své existence vyvinul ze své experimentální podoby do plnohodnotné a uznávané metody pro datově orientované problematiky. Přesto ale narážíme na aspekty, které stále budí pochybnosti a mohou tím pádem část zájemců odradit. Dva z hlavních problematických hledisek jsou motivace a kvalita – ty se navíc výrazně ovlivňují a snadno může dojít k situaci, kdy nedostatečná motivace negativně ovlivní výslednou kvalitu. Dalšími aspekty jsou důvěrnost a komunikace – obojí není snadné správně uchopit. Pokud neopatrně pracujeme s citlivými údaji, případně nedokážeme pochopitelně sdělit zadání, nepotěšíme tím zadavatele ani skupinu pracovníků.

Motivovaný uživatel je jediným, od kterého můžeme očekávat jakékoli výsledky. Nemáme však zaručeno, že jeho přínos bude dosahovat požadované kvality. Existuje hypotéza, podle které zapojení lidé mohou být v zásadě rozděleni do dvou kategorií dle druhu motivace – finanční či pro zábavu [5]. Přitom v druhém případě mohou peníze sloužit jako dodatečný stimulant. Přizpůsobení potřebám jedné z těchto skupin uživatelů má pozitivní vliv na rozpočet zadavatele a následně také na kvalitu odevzdané práce. Zatímco finančně motivovaní lidé jsou zvyklí dodávat více či méně kvalitní výsledky často a pravidelně, u druhé skupiny se přínos objevuje dle jejich možností, ale přistupují k práci zodpovědněji.

Dále je dobré rozlišit motivaci pracovníka a zadavatele. U pracovníků můžeme pozorovat touhu po peněžní či jiné odměně, osobní zapálení pro danou problematiku, případně jen hledání zábavy. Oproti tomu zadavatelé potřebují vylepšit svůj produkt interakcí s cílovou skupinou a nejčastěji tedy chtějí zvýšit zisk z prodeje. Může se jednat i o šanci dostat se do kontaktu s profesionály z oboru, kteří by se k vývoji produktu jiným způsobem nedostali.

Neméně důležitým aspektem je důvěrnost – citlivé údaje jsou v dnešní době často tou nejcennější komoditou, která se ve virtuálním

prostředí pohybuje. Pro vylepšení produktu ale může být nezbytné nechat cizí pracovníky k těmto informacím přistupovat a není žádoucí jejich ochranu podcenit. Snadno tak můžeme dojít do situace, kdy vzniklá finanční ztráta díky úniku důvěrných informací daleko převýší zisk, kvůli kterému jsme snahu o zlepšení produktu vyvíjeli. Řešením může být využití anonymizace poskytovaných dat v co největší míře, případně důkladný výběr spolehlivých a důvěryhodných pracovníků.

To nás přivádí k problematickému aspektu komunikace. Tím je složitost nalezení nejlepšího možného způsobu, kterým můžeme předat úkol lidem, aby byl srozumitelný [6]. Efektivního výsledku (vzhledem k času i finančním úsporám) lze dosáhnout zejména pochopením uvažování cílové skupiny pracovníků. S tímto velice úzce souvisí rozložení celého problému na drobné úkoly, které jsou pak snadno zpracovatelné. Často navíc musíme vyvažovat mezi mírou jednoduchosti a zábavnosti, která má vysoký vliv na motivaci uživatelů. V případě nedostatečné úrovně rozložení a zbytečně komplikovaných instrukcí snižujeme účinnost řešení – to se může projevit nízkým počtem výsledků i jejich sníženou kvalitou.

Kvalita vs. efektivita Ani v případě, kdy poskytneme dobré podmínky ve všech předchozích aspektech, nevyhneme se kontrole kvality. Část uživatelů se bude vždy snažit maximalizovat svůj zisk na úkor řádně odvedené práce [7]. Existuje více různých technik, které vedou k rozlišení dobrých a špatných výsledků, případně k dlouhodobému profilování pracovníků do kategorií poctivých a nepoctivých [8, 9]. Cílem je však vždy přiblížit se ideálnímu poměru mezi kvalitou a efektivitou – ten není nijak stanoven a je specifický pro každý jednotlivý projekt. Např. nekomerční projekty si zřejmě mohou dovolit požadovat vysokou kvalitu na úkor kvantity, zatímco komerční aplikace se vyznačují vyšším tlakem na rozpočet. Obecně se tedy nedá říci, že nadměrná kontrola je zároveň velice drahá. Při šetření financí na kontrolách se ale může stát to, že výsledná cena na jeden přínosný výsledek bude neúměrně vysoká kvůli daleko vyšší míře nekvalitní práce.

Jednou z nejpoužívanějších technik pro kontrolu kvality je proložení běžných úkolů takovými, které jsou již odborně zpracovány

a použijí se jako ověření, jestli uživatel vyplňuje poctivě nebo ne. Pokud takový uživatel ve vyšší míře odpovídá odlišně od vzorově vypracovaného zadání, jedná se s velkou pravděpodobností o podvodného pracovníka. Postup lze aplikovat jen na takové úkoly, kde jsou vstupem pevně definované hodnoty (např. dotazníky či hodnocení relevance). Podobně mohou sloužit nepravidelné manuální kontroly vybraných výsledků – v tomto případě ale můžeme odhalovat nepoctivé pracovníky i v řešení úkolů, které vyžadují vlastní tvorbu (sběr dat nebo jejich přepis).

Další běžnou a efektivní technikou je srovnávání existujících výsledků na úrovni jednoho úkolu, který byl vypracován několika samostatnými pracovníky. Díky rozumné úrovni redundance tak můžeme říci, že většinový názor bude správnou odpovědí a výrazně odlišné odpovědi tak mohou být označeny za chybné. Stejně jako u předchozí techniky jsou poté nepoctiví pracovníci vyřazeni. Podobnou myšlenku lze aplikovat i pomocí techniky s kontrolní skupinou – jen s tím rozdílem, že si uživatelé budou výsledky kontrolovat naprosto vědomě. V praxi to probíhá tak, že je daný úkol vypracován jen jednou a poté je ověřen pracovníkem z kontrolní skupiny. Tento krok je rychlejší, tedy i levnější a uživatelé jsou motivováni možností odhalit podvodníky. Navíc se takto vyhneme vysokému množství opakování stejného zadání.

Všechny předchozí varianty pracují s postupným budováním tzv. skóre pracovníka, které roste či klesá s precizností jeho výsledků. Aktuální hodnota není nikdy pracovníkovi známá, nicméně je možné jí ve vhodnou chvíli využít a poskytnout zpětnou vazbu v podobě varování při určité nízké hodnotě, případně naopak dodatečné poděkování při nadprůměrném skóre. Jednotlivé algoritmy zakládající se na těchto technikách blíže popisuji v následující kapitole.

2.4 Algoritmy pro kontrolu kvality

Existující platformy pro crowdsourcing využívají různých technik, aby zajistily co nejvyšší kvalitu práce dodané zadavatelům. Některé k tomu potřebují asistenci zadavatele a je nutné kontrolní data definovat, jiné využívají automatizovaná řešení s pomocí statistických metod. Většina provozovatelů těchto platforem však konkrétní algoritmy udržuje

v tajnosti, ale lze najít i výjimky (např. Amazon Mechanical Turk [10]). Motivací každého z těchto algoritmů je však minimalizace nákladů se zachováním co nejvyšší míry kvality výsledků.

Pro názornost byly vybrány nejčastěji využívané přístupy, jejichž společným prvkem je určitá míra redundance. Právě vícenásobná práce na jednom zadání je dle Amazonu mezi zadavateli nejžádanější. Problémem však je to, že právě redundance nejvíce zvyšuje výslednou cenu celého řešení. Cílem je tedy najít takovou kontrolní techniku, která umožní s rozumnou přesností odhadnout kvalitu pracovníků, pomocí které následně nepoctivé uživatele rozpoznáme a vyřadíme.

První variantou k odlišení dobrých a špatných výsledků je tedy výpočet většinového názoru (viz algoritmus 1) – konkrétní hodnota je považována za správnou v případě, že se na ní shodne většina pracovníků. Tímto principem můžeme rozlišovat pouze takové typy otázek, na které lze předem definovat diskrétní množinu odpovědí. Kvalita jednotlivých uživatelů je poté vypočítaná jako poměr odpovědí shodných s většinovým názorem a celkovým počtem odpovědí.

Algoritmus 1 The Majority vote algorithm [10]

Vstup: Labels $l[k][n]$ from worker (k) to object o_n
Výstup: Correct labels $T(o_n)$ for each object o_n, Error rate $s[k]$ for each worker (k)

1: **for all** *objects* o_n **do**
2: Count correct labels $T(o_n)$ as majority vote for object o_n
3: **end for**
4: **for all** *workers* (k) **do**
5: Count error rate $s[k]$ from majority votes for each worker (k)
6: **end for**
7: **return** Estimated correct labels $T(o_n)$, Estimated error rates $s[k]$

Odvozená varianta z předchozího využívá myšlenky, že ne všichni pracovníci jsou si po stránce kvality odvedené práce rovni (viz algoritmus 2). Každému uživateli nejprve vypočítáme přesnost jeho odpovědí s pomocí skupiny několika odborně vypracovaných úkolů (kontrolní otázky). Odpověď je v tomto případě považována za správnou, pokud se shoduje s názorem většiny, nicméně váha odpovědí se může výrazně lišit. Nepoctiví pracovníci budou ovlivňovat většinový

názor jen minimálně. Výhodou tohoto přístupu je výrazné snížení míry redundance potřebné pro dosažení určité úrovně kvality, jelikož pracujeme s kvalitnějšími odpověďmi.

Algoritmus 2 The Weighted majority vote algorithm [10]

Vstup: Labels $l[k][n]$ from worker (k) to object o_n, Labels $cl[k][n]$ from worker (k) to control object co_n
Výstup: Correct labels $T(o_n)$ for each object o_n, Error rate $s[k]$ for each worker (k)

```
for all workers (k) do
    Initialize error rate s[k] for worker (k) (assume he is perfect)
    for all control objects co_n do
        Count error rate s[k] for object co_n votes and worker (k)
    end for
end for
for all objects o_n do
    Count correct labels T(o_n) as majority vote for object o_n weighted by worker (k) error rate s[k]
end for
return Estimated correct labels T(o_n), Estimated error rates s[k]
```

Základem pro velkou část algoritmů jsou myšlenky, které přinesli A. P. Dawid a A. M. Skene již v roce 1979 [11]. Jejich práce je založená na statistické metodě maximální věrohodnosti, pomocí které odhadují míru chybovosti konkrétních uživatelů. Zjednodušeně řečeno, algoritmus nejprve získá správné hodnoty pomocí názoru většiny uživatelů a předpokládá bezchybnost každého pracovníka. Poté provádí iterace následujících dvou kroků do dosažení konvergence: 1) odhadne míru chybovosti pracovníků na základě správných hodnot a poté 2) upraví odhad správných hodnot na základě názorů kvalitních pracovníků, seřazených dle míry chybovosti (viz algoritmus 3). Díky tomuto řešení se můžeme vyhnout odborně vypracovaným úkolům (jako v případě předchozího algoritmu) a zároveň můžeme pracovat s rozumnou mírou redundance.

Existují i další zajímavé algoritmy. Jedním z takových je např. algoritmus pro oddělení chybných a záměrně matoucích odpovědí [10]. Ne vždy je totiž možné říci, že chybující pracovníci nejsou užiteční – po-

kud odhalíme, že konkrétní uživatel vždy označuje špatné odpovědi jako správné, lze paradoxně s jeho pomocí indikovat výsledky, které do správných odpovědí určitě nepatří. Dle výše zmíněných zdrojů jsou nejméně užitečnými uživateli takoví, kteří odpovídají zcela náhodně.

Algoritmus 3 The EM algorithm for worker quality estimation [10]

Vstup: Labels $l[k][n]$ from worker (k) to object o_n

Výstup: Confusion matrix $\pi_{ij}^{(k)}$ for each worker (k), Correct labels $T(o_n)$ for each object o_n, Class priors $Pr\{C\}$ for each class C

1: Initialize error rates $\pi_{ij}^{(k)}$ for each worker (k) (e.g., assume each worker is perfect)
2: Initialize correct label for each object $T(o_n)$ (e.g., using majority vote)
3: **while** *not converged* **do**
4: Estimate the correct label $T(o_n)$ for each object, using the labels $l[\cdot][n]$ assigned to o_n by workers, weighting the votes using the error rates $\pi_{ij}^{(k)}$
5: Estimate the error rates $\pi_{ij}^{(k)}$, for each worker (k), using the correct labels $T(o_n)$ and the assigned labels $l[k][n]$
6: Estimate the class priors $Pr\{C\}$, for each class C
7: **end while**
8: **return** Estimated error rates $\pi_{ij}^{(k)}$, Estimated correct labels $T(o_n)$, Estimated class priors $Pr\{C\}$

Každý z uvedených algoritmů lze pro vyšší spolehlivost rozšířit použitím technik zmíněných v předchozí kapitole. Takovým rozšířením může být např. náhodná manuální kontrola výsledků zadavatelem, případně skupinou kontrolních pracovníků. Ti mohou pracovníky s vysokou četností špatných odpovědí vyřazovat a dát tak prostor poctivějším. Vyřazením byť jediného záměrně chybujícího pracovníka lze významně zvýšit kvalitu výsledných odpovědí.

3 Existující platformy

Nástrojů využívajících potenciál crowdsourcingu vzniklo již velké množství a není možné je zmapovat všechny [9]. Můžeme nicméně prozkoumat alespoň několik z takových, které se na trhu vyskytují nejdéle, případně mají nejširší uživatelskou základnu. Podstatné jsou pro nás veškeré funkce platformy, abychom byli schopní zhodnotit její užitečnost a případnou využitelnost pro naše potřeby. Zajímají nás zejména typy podporovaných úkolů – jednotlivé služby se mohou výrazně lišit podrobností rozdělení, případně zaměřením na určité typy úloh. Dále je důležitý samotný proces zadávání a řešení jednotlivých úkolů uživatelem – některé jsou přívětivé a intuitivní, jiné vyžadují dodatečný čas k pochopení a zorientování v prostředí platformy.

3.1 Amazon Mechanical Turk

Pravděpodobně nejznámější platformou pro crowdsourcing je v současné době Amazon Mechanical Turk[1] vyvinutý společností Amazon v roce 2005. Základní jednotkou pro ohodnocení jednotlivých úkolů je jeden HIT[2] – Amazon ji definuje jako „úlohu lidské inteligence“, případně jako „otázku potřebující odpověď“. Jedná se o nejmenší možný úkol, na kterém může libovolný uživatel pracovat, odeslat odpovědi a dostat odpovídající odměnu. Z pohledu plánování rozpočtu se jedná o částky v rozmezí jednoho centu až do několika dolarů, dle náročnosti úkolu – Amazon si poté účtuje provizi 20 procent ze všech plateb a bonusů. Příkladem může být úkol určit kategorii pro obrázek, nalezení jednoduché informace na jiných webech, nebo přeložení fráze do vybraného jazyka.

Pokud tedy vytváříme databázi webových stránek pro marketingovou kampaň a potřebujeme kontaktní informace na specifickou klientelu, můžeme zaměstnat jednoho člověka a nechat ho projít několik stovek nebo tisíců webů – to znamená i několik dní čekání na dokončení práce. Můžeme ale každou webovou adresu vložit na mturk.com jako samostatný HIT se zadáním nalézt kontakt za odměnu něko-

[1] https://www.mturk.com/
[2] Human Intelligence Task

lika centů a úkol může být hotový během jednotek hodin, navíc za podobné či nižší náklady.

Kvalitu výsledků je možné hlídat pomocí tzv. „Mechanical Turk Masters“, což je podmnožina z pracovníků, kteří již prokázali svoji přesnost u specifických úkolů – Amazon využívá statistických metod a algoritmů ke dlouhodobému monitorování práce svých uživatelů [10]. Dále máme možnost pracovníky limitovat výběrem různých kvalifikací – na našich úkolech tak mohou pracovat lidé z konkrétní lokality či pracovníci nepřekračující určitou úroveň chybovosti. Každá kvalifikace je zpoplatněna samostatně a nabídka je široká[1].

Jinou možností je poskytnutí kontrolních odpovědí označovaných jako „Gold Standard HITs“, tedy vzorově vypracovaná zadání se správnými výsledky. Jejich účelem je zajistit sadu řešení, pomocí kterých lze snadno detekovat poctivé řešitele od nepoctivých. Pokud není zadavatel spokojený s libovolným vypracovaným úkolem, může ho odmítnout bez utracení svých prostředků. V takových situacích ale dochází k negativnímu ovlivnění jeho reputace, což působí jako ochrana pro samotné pracovníky, pokud by zadavatel často výsledky odmítal.

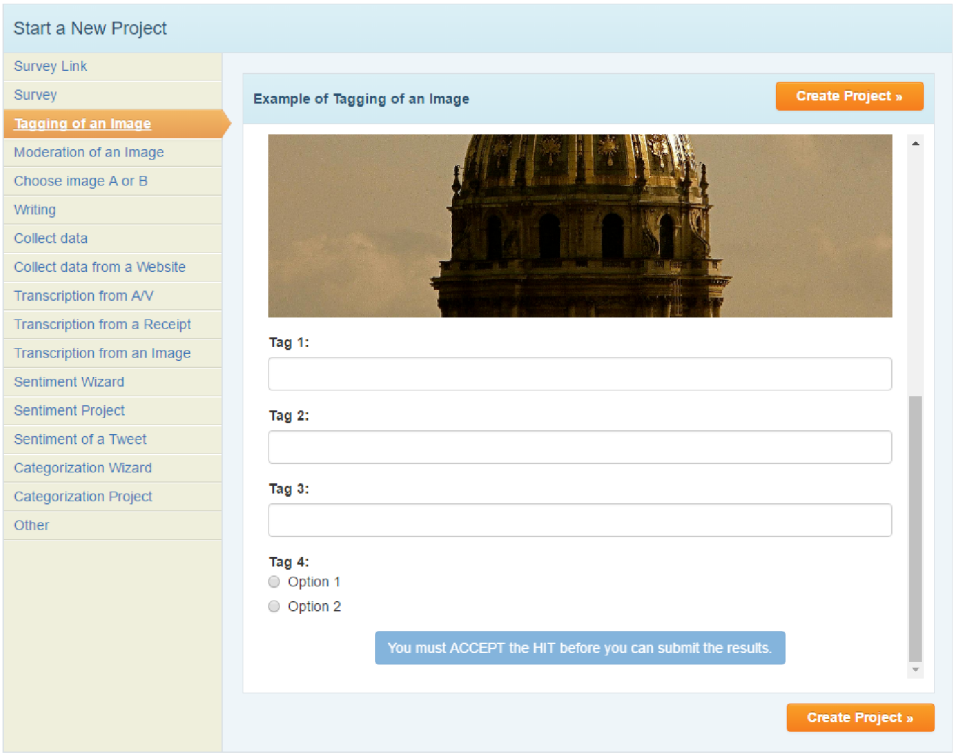
Přehled typů úloh

- Externí dotazník – Součástí každého úkolu je odkaz na dotazník, který může být umístěn na libovolném webu. Po jeho dokončení je pracovníkovi zobrazen jedinečný kód pro zkopírování. Jakmile ho uživatel zadá do systému mturk.com, úkol je považován za vyřešený a odměna je připsána.

- Dotazník – Sada otázek v podobě formuláře, ve kterém je možné odpověď definovat textovým polem pro volnou odpověď či výběrem z několika možností (jednoduchý nebo vícenásobný výběr).

- Klíčová slova k obrázku – Vstupním prvkem je obrázek, ke kterému pracovník vypíše definované množství jednoslovných klíčových slov. Slova musí popisovat obrázek, jeho obsah nebo jiný relevantní kontext. Je možné definovat i výběr z několika možností.

[1] https://requester.mturk.com/pricing/



Obrázek 3.1: Výběr typu nového projektu v rozhraní Mechanical Turk

- Moderace obrázku – Uživateli je zobrazena sada několika obrázků, ve které má za úkol označit všechny, splňující zadaná kritéria (např. nahota, násilí nebo ofenzivní gesta).

- Výběr obrázku A/B – Ze dvou zobrazených obrázků (např. varianty loga) je nutné vybrat jednu, která více splňuje požadované vlastnosti.

- Psaní – Tvůrčí typ úkolu, jehož vstupem může být jen adresa webu a požadavky na výsledný text. Pracovník si stránky nejprve prohlédne a poté sepíše krátký popis jejich obsahu, nabízeného produktu nebo služeb. Požadován je originální text s délkou v určitém rozsahu.

- Sběr dat – Na základě poskytnutých údajů musí uživatel dohledat některé další. Např. vyplnit adresu webových stránek se znalostí názvu a adresy firmy.
- Sběr dat z webu – Opačný princip než v předchozím případě. Pracovníkovi je poskytnuta pouze adresa webových stránek, ve kterých má za úkol dohledat požadované informace (název společnosti, adresu, telefonní kontakty a další).
- Přepis z audia/videa – Uživateli je poskytnuta zvuková nebo video nahrávka, kterou si musí poslechnout a následně zaznamenat veškeré mluvené slovo. Na základě instrukcí je nutné např. vynechat tzv. „slovní berličky“ či rozlišovat jednotlivé mluvčí. Délka záznamu je často určujícím kritériem pro stanovení výše odměny.
- Přepis účtenky – Vstupem je oskenovaná nebo vyfocená účtenka, ze které musí pracovník vyčíst a zapsat jednotlivé položky s cenami, mezisoučet, vypočítanou daň a celkovou částku. Pokud nejsou údaje čitelné, je možné tuto skutečnost zvolit bez vyplňování ostatních údajů.
- Přepis obrázku – Podobný princip jako v předchozím případě, jen se jedná o jeho obecnější variantu. Vstupem je opět obrázek, nicméně uživatel poté přepisuje požadované údaje dle instrukcí.
- Sentiment – Podrobný nástroj pro vytváření úkolů spojených s hodnocením citového obsahu sdělení. Zadavatel může nechat pracovníky ohodnotit např. komentáře ke svému produktu, a to na škále od silně negativních, přes neutrální, až po silně pozitivní. Získá tak dodatečnou informaci, se kterou může dále pracovat. Amazon nabízí i rozšíření v podobě hodnocení sentimentu zpráv na sociální síti Twitter[1].
- Kategorizace – Velice podobný princip jako u přiřazování klíčových slov k obrázku. Uživatel má k dispozici předem definované kategorie, ze kterých vybere vždy jednu, nejlépe vystihující zobrazený obrázek.

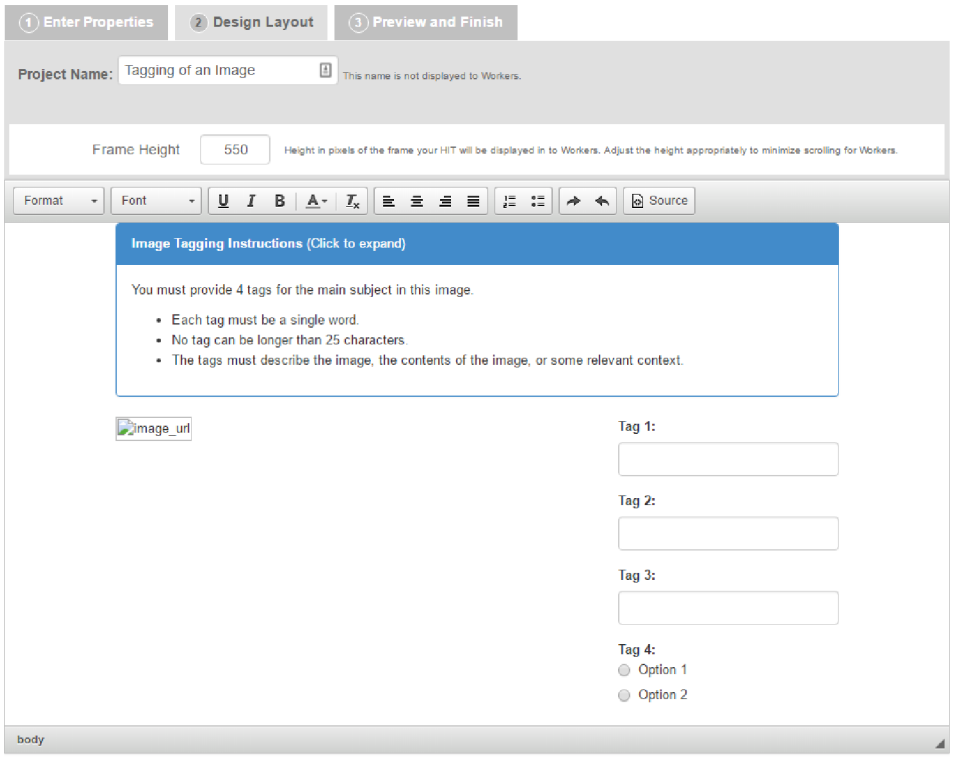
[1] https://twitter.com/

Proces zadávání Uživatelské prostředí je přívětivé a intuitivní, navíc bez nadbytečných kroků. Z pohledu zadavatele si nejprve zvolíme jeden z velkého množství typů projektů (viz výše). Ke každému typu Amazon zobrazuje příklad využití (viz obrázek 3.1) s doplňujícím komentářem a ihned po výběru následuje zadání veškerých atributů projektu:

- Administrační název – Popis viditelný jen pro naše potřeby.
- Informace pro pracovníky – Název projektu, krátký popis a klíčová slova.
- Nastavení pro jeden HIT – Výše odměny za splnění jednoho úkolu, počet uživatelů pracujících na stejném úkolu (míra redundance), maximální povolený čas na jeho splnění, nastavení doby expirace (jak dlouho bude HIT nabízen) a doby pro automatické schválení (čas, který má zadavatel pro odmítnutí splněného úkolu).
- Požadavky na pracovníka – Volba uživatelů pouze ze skupiny „Masters“, výběr až pěti dodatečných kvalifikací, označení projektu zahrnujícího obsah pro dospělé a nastavení viditelnosti úkolů (u veřejného si popis i náhled zobrazí všichni, u soukromého vidí náhled pouze pracovníci splňující kvalifikace a u skrytého vidí popis i náhled jen uživatelé splňující daná kritéria).

Dalším krokem je již vytváření šablony, které jsou řešeny formou WYSIWYG[1] editoru. V něm je možné definovat rozložení prvků pro všechny úkoly, včetně instruktážního textu (viz obrázek 3.2) – do této formy budou později automatizovaně vkládány jednotlivé úkoly. Ty jsou v šabloně vloženy na místa, která jsou označena speciální sekvencí znaků. Po prohlédnutí výsledné formy je projekt připraven a je možné importovat sadu úkolů k řešení. Některé typy úkolů umožňují následně vložit i kontrolní odpovědi označované jako „Gold Standard HITs“.

[1] Akronym věty „What you see is what you get“ („co vidíš, to dostaneš“).



Obrázek 3.2: Editace šablony projektu v rozhraní Mechanical Turk

Proces řešení Pracovník má situaci ještě jednodušší. Stačí přejít na stránku s výpisem jednotlivých úkolů (viz obrázek 3.3), dle vyhovujících kritérií vyhledat libovolný úkol ze všech dostupných (např. podle popisu, výše odměny či klíčových slov) a jedním kliknutím zobrazit náhled konkrétního zadání. Pokud forma vyhovuje a uživatel si je jistý, že rozumí zadání, může ihned začít řešit úkoly. V případě, že jeho vypracované úkoly budou schváleny, k uživatelskému účtu je připsána odpovídající odměna.

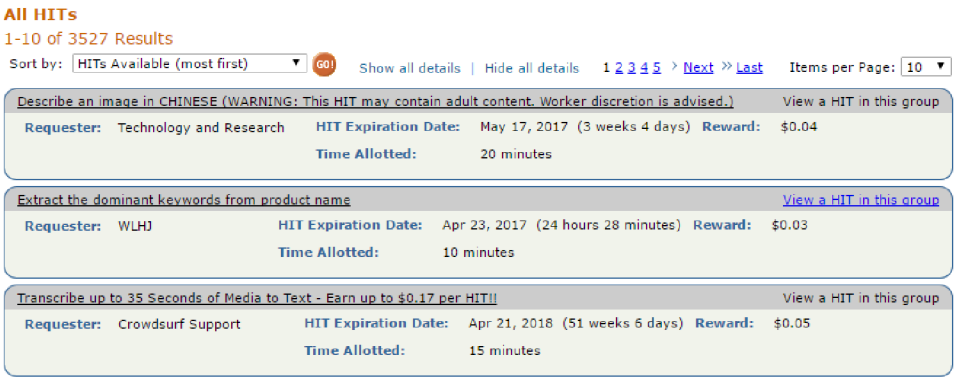
Výhody

- Více než půl milionu pracovníků z nejméně 190 zemí světa.
- Možnost škálování počtu pracovníků v reálném čase.

- Pokročilé API pro komunikaci s externími systémy.

Nedostatky

- Zastaralý design (odpovídající době vzniku).



Obrázek 3.3: Nabídka několika úkolů v rozhraní Mechanical Turk

3.2 Clickworker

Projekt Clickworker[1], stejně jako v případě Amazonu, vznikl v roce 2005 a je tedy jedním z nejstarších svého druhu, nicméně první klienty získal až v roce 2007. V současné době služba nabízí dvě úrovně zadavatelů – základní účet zdarma s omezenou funkcionalitou a bez API (za plnění objednávek je nutná platba předem), případně tarif na míru s osobním přístupem a množstvím pokročilých funkcí (s fakturací až po dodání práce). Jednotkou pro pracovní úkon je jeden „microjob“, na jehož základě probíhá výpočet ceny. Clickworker udává, že geografické složení pracovníků je téměř na čtvrtiny rozděleno mezi Německo, USA, poté společně ostatní státy Evropy a poslední čtvrtinou jsou pracovníci z ostatních států světa.

Základní typy úkolů jsou podobné těm poskytovaným Amazonem, nicméně zajímavým rozšířením jsou úlohy pro trénink umělé

[1] https://www.clickworker.com/

inteligence a mobilní crowdsourcing. Pomocí nich můžeme rozvíjet libovolný systém vyžadující strojové učení – pracovníci mohou např. vytvářet specifické zvukové záznamy (rozpoznávání mluveného slova), konkrétně zaměřené fotografie (odhadování emocí z výrazu ve tváři), ručně psané texty s předem daným obsahem a jiné. V případě mobilního crowdsourcingu zákazníci např. hledají nebo pořizují obrázky restaurací a jejich menu, analyzují pozici na trhu u daných produktů, nedostatečnou dostupnost některých výrobků, nebo může být užitečný pro tzv. „mystery shopping“[1].

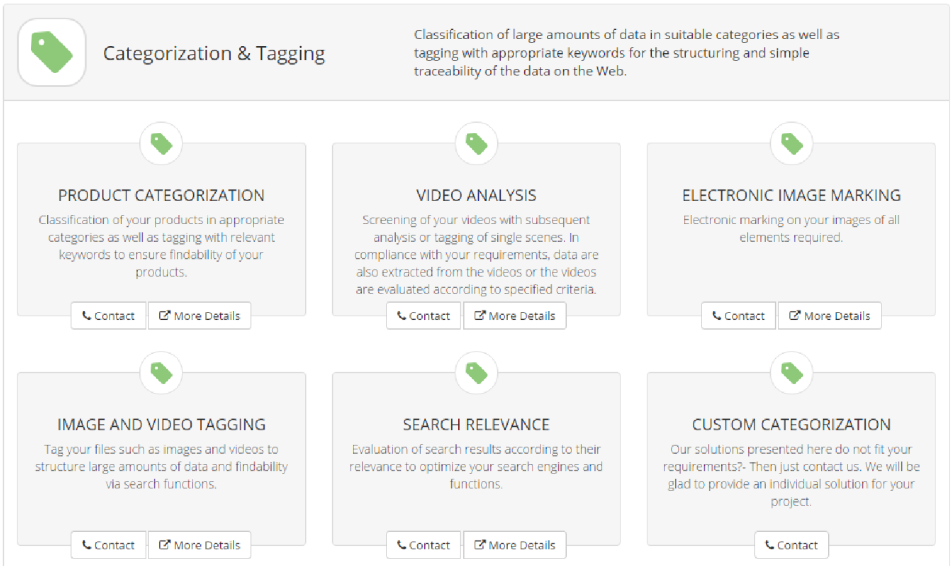
Kontrola kvality probíhá v první fázi striktněji než u Amazonu, ale později je skóre pracovníka upravováno podobným způsobem. Ihned po registraci musí totiž uživatel poskytnout informace o svých znalostech či zkušenostech a dále projít tréninkem, díky kterému získá kvalifikace pro plnění konkrétních úkolů. Nelze tedy začít zpracovávat úlohy bez odpovídajících kvalifikací. Po tréninku však může pracovat na dostupných zadáních a v případě spokojenosti s jeho výsledky jsou jeho kvalifikace postupně zvyšovány. Současně probíhá i automatizovaná analýza kvality pomocí různých metrik a statistik, které však nejsou zadavatelem ovlivnitelné. V případě detekce plagiátu je výsledek vyřazen a neproplacen. Každý typ úlohy je ale v kontrolních technikách odlišný, takže kromě výše zmíněných poskytuje Clickworker také kontrolu pomocí většinové volby (vícenásobné řešení), případně audity pracovníkem s vyšší kvalifikací.

Přehled typů úloh

- Text – Nejpočetnější skupina různých typů úkolů, které zajišťují vytvoření požadovaných textů na základě zadavatelem dodaných informací a s možností překladu do různých jazyků.
  - Obecný text – Umožňuje vytváření obsahu pro brožury či webové stránky, ale bez požadavků na klíčová slova.
  - SEO[2] text – Unikátní obsah pro webové stránky, ve kterém požadujeme rovnoměrné použití klíčových slov.

[1] Mystery shopping („tajné nakupování“) je metoda pro výzkum trhu, která pro zkoumání kvality produktů či služeb využívá fiktivní nakupující.

[2] Search Engine Optimization („optimalizace pro vyhledávače“)



Obrázek 3.4: Výběr typu nového projektu v rozhraní Clickworker

  - Popis produktu nebo kategorie – Texty jsou vhodné jako obsah prodejních webů, kde vyžadujeme srozumitelné předání informací zákazníkům.
  - Poradce nebo glosář – Vysoce kvalitní a obsáhlý text sloužící jako návod, případně naopak krátký a výstižný popis konkrétního pojmu.
  - Popis destinace – Informačně a často i emocionálně atraktivní text o dané zemi, regionu či městě.
  - Profil společnosti – Přesvědčivé texty pro prezentaci firmy.
  - Blogový článek – Profesionální a dobře strukturovaný článek pro konkrétní cílovou skupinu návštěvníků.

- Dotazník – Standardní způsob získávání informací od uživatelů. Clickworker nabízí vlastní nástroj pro generování dotazníků, ale také integraci externích služeb. Podstatné je zde nastavení správné cílové skupiny pracovníků tak, aby výsledky odpovídaly našim potřebám (např. věk, pohlaví, země).

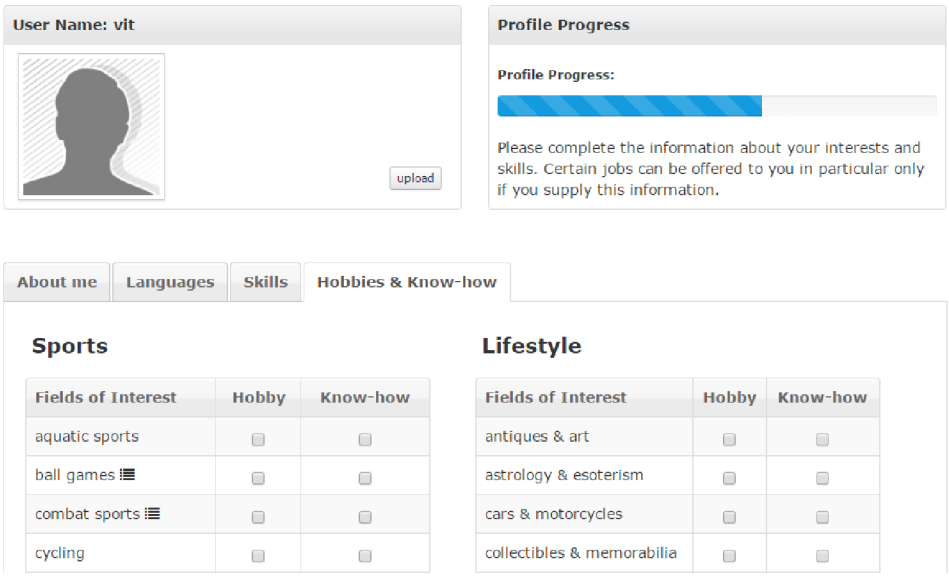
- Webový výzkum – Součástí této skupiny úloh je získávání a ověřování informací z různých webových stránek. Může se jednat o sběr dat (např. kontaktní adresy či pracovní doba), ale také o jejich validaci oproti zadavatelem poskytnutým informacím. Důležitou možnosti je také získávání informací o konkurenci nebo o potenciálních klientech.

- Kategorizace a štítkování – Klasifikaci velkého objemu dat a přiřazování klíčových slov je možné využít různými způsoby. Společným cílem je zde usnadnění vyhledání daného produktu či jiných objektů. Clickworker nabízí také analýzu videí, zvukových záznamů a obrázků, ke kterým následně pracovníci přiřazují odpovídající kategorie. V případě videí je navíc možné samostatně kategorizovat i jeho konkrétní úseky a získávat doplňující informace. Podstatnou funkcí je také posuzování relevance výsledků vyhledávání dle klíčových slov.

- Produktová data – Díky této skupině úkolů má zadavatel možnost udržovat údaje o svých produktech aktuální. Je možné zadat pracovníkům extrakci kvantitativních dat z informací o produktu a tím zajistit jeho vyšší dostupnost při vyhledávání.

- Umělá inteligence – Tato skupina je jednou z nejvíce inovativních, jelikož umožňuje zdokonalovat systémy založené na strojovém učení. Zadavatel může zadat požadavek na vytváření a upravování dat pro trénink těchto systémů, jako i validaci jejich výsledků, které není možné kontrolovat automatizovaně. Ke každému z takových případů je přistupováno individuálně, což zaručuje dobrou úroveň přesnosti.

- Mobilní crowdsourcing – Poslední možností jsou úkoly využívající chytré telefony. Snadno tak lze monitorovat silná a slabá místa při prodeji konkrétních produktů přes tato zařízení, případně nechat pracovníky ověřovat kontaktní informace, navigační systémy či jiný typ dat. Dále je možné nechat uživatele tato data pořizovat, včetně fotografií nebo údajů o poloze.



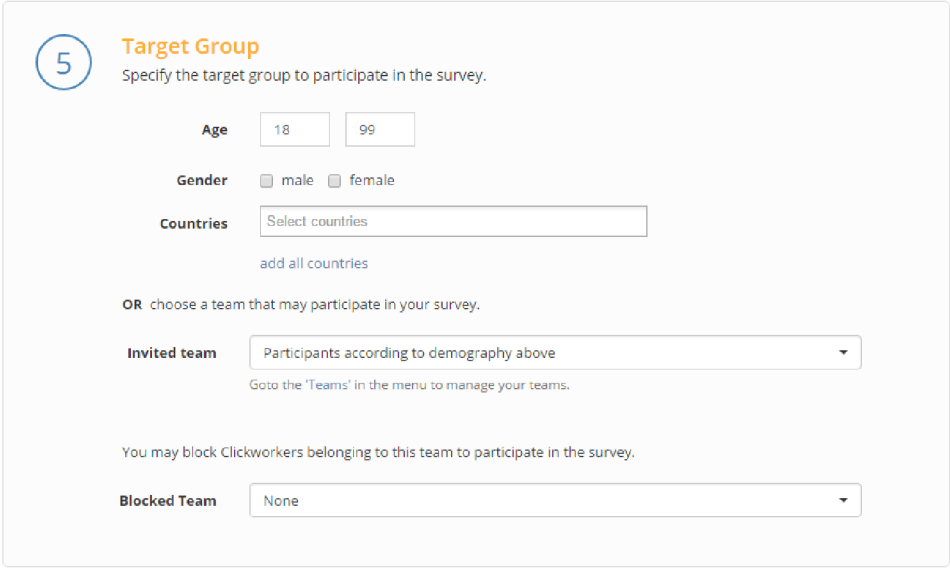
Obrázek 3.5: Editace profilu pracovníka v rozhraní Clickworker

Proces zadávání Platforma je ve srovnání s Amazonem na daleko lepší úrovni z pohledu uživatelského rozhraní. Prostředí je intuitivní a snadno ovladatelné, čemuž pomáhá i responzivní design. Pokud si zadavatel vybere z mnoha typů úloh (viz obrázek 3.4), následuje zadání všech informací o objednávce projektu:

- Název projektu pro pracovníky a interní název pro zadavatele.
- Nastavení dle typu úkolu – Pro tvorbu textů se jedná o vyžadovaný jazyk, délku textu, úroveň kvality (určující parametr pro výpočet ceny) a požadavku na korekturu vysoce kvalifikovanými uživateli. U překladů zvolíme podobně jazyk a úroveň kvality, naopak v případě dotazníků navíc nastavujeme i počet účastníků a cenu za jeho vyplnění.
- Instrukce – Textová informace pro pracovníky, vysvětlující účel úkolu, dodatečné podmínky a postup jeho plnění.

- Cílová skupina – Nastavení určující věkové rozpětí, pohlaví, země pobytu, případně další specifické požadavky na pracovníka (viz obrázek 3.6).

- Doba trvání – Některé typy úkolů mohou být pro vyplnění omezeny datem.



Obrázek 3.6: Nastavení cílové skupiny v rozhraní Clickworker

Dalším krokem je již import dat dle vyžadované struktury a náhled zkompletované objednávky projektu. Postup je již podobný jako v případě Amazonu.

Proces řešení Rozhraní pro pracovníky je po grafické stránce starší, nicméně stále velice dobře přehledné. Pro plnění úkolů je nutné se registrovat a řádně vyplnit profil pracovníka (viz obrázek 3.5) – zejména jazykové znalosti, zkušenosti a osobní zájmy. Poté je již možné navštívit stránku s aktuální nabídkou, která je určena i pro nejnižší kvalifikaci. Pokud jsou vyžadovány konkrétní znalosti či zájmy, nabídka se nezobrazuje vůbec. V rámci testování bylo možné některé úkoly začít řešit

ihned, nicméně pro jiné je nejdříve nutné splnit tréninkové úkoly. Po vypracování úlohy je odměna přičtena zpravidla do 24 hodin.

Výhody

- Rozsáhlá uživatelská základna čítající 800 tisíc pracovníků.
- Moderní a intuitivní design uživatelského rozhraní.
- Přehledná struktura hlavních skupin úkolů a jejich podtypů.

Nedostatky

- Část úkolů není možné definovat samostatně a je nutné kontaktovat poskytovatele služby. Jen několik úloh je možné zadat pomocí tzv. „samoobsluhy" – např. vytváření textů, jejich překlady či dotazníky.

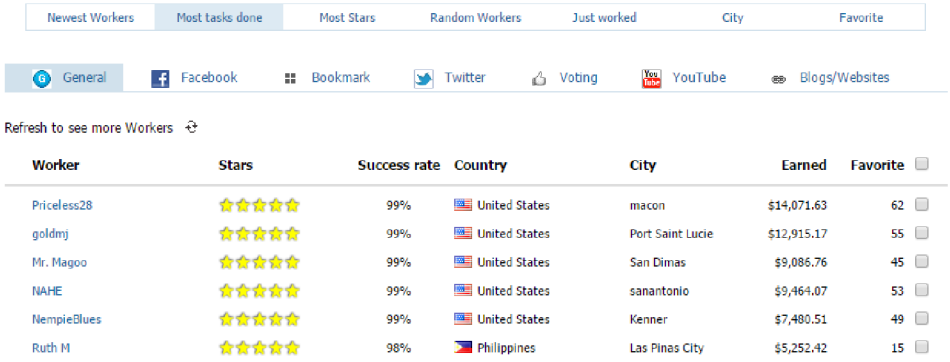
3.3 Microworkers

Portál Microworkers[1] vznikl v roce 2009 a je jedním z nejobsáhlejších z pohledu možných typů úloh (v tomto srovnání vyzdvihuji pouze takové, které jsou odlišné od ostatních služeb). Z vybraných platforem pro crowdsourcing je zároveň i příkladem s největší uživatelskou základnou – za to pravděpodobně vděčí hlavně široké škále inovativních typů úkolů. Je možné jej využít např. pro hodnocení kvality videa či obrázků, jednoduché A/B testování či validaci vhodnosti reklamních bannerů na webových stránkách. Zajímavou možností je také testování mobilních aplikací, které je i nadprůměrně finančně hodnoceno. Jednotlivé projekty (kampaně) lze také pozastavovat, případně po dokončení opakovat se shodným nastavením.

Jako jedna z mála platforem také poskytuje hodnocení kvality jednotlivým uživatelům a různé statistické žebříčky (viz obrázek 3.7), které mohou kladně ovlivnit motivaci pracovníků. Díky veřejným hodnocením uživatelů je také možné jednotlivé pracovníky zaměstnat

[1] https://www.microworkers.com/

selektivně, případně odměňovat ty nejproduktivnější. Kvalitu lze kontrolovat několika různými způsoby – ať už poskytnutím předvyplněných kontrolních úkolů, případně plně automatizovanými metodami. Pro některé typy úkolů lze od uživatele vyžadovat vložení snímku obrazovky či jiného potvrzení o splnění úlohy.



Obrázek 3.7: Přehled nejlepších pracovníků v rozhraní Microworkers

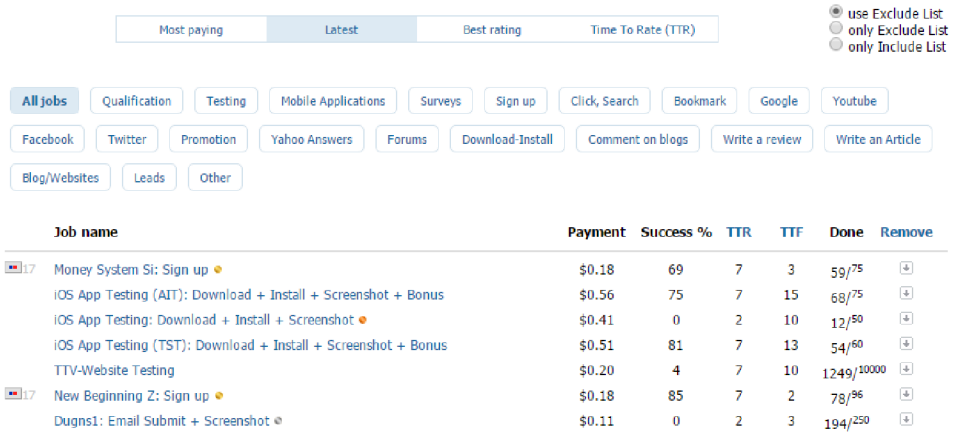
Přehled typů úloh

- Řazení událostí – Pracovník má za úkol chronologicky seřadit dané události na základě obrázku nebo textu.
- Kvalita obrázků nebo videa – Hodnocení kvality konkrétního snímku či různých formátů videa.
- Srovnání produktů – Uživatel na základě parametrů konkrétního zboží nebo služby hodnotí, jestli jsou identické, případně který produkt je lepší.
- A/B testování – Využití klasické marketingové metody např. pro výběr vhodnějšího designu nebo reklamního sdělení.
- Monitoring reklam – Pracovník hodnotí, jestli jsou reklamy pro danou stránku vhodné, příliš rušivé, nebo v nějakém ohledu závadné.

- Testování mobilní aplikace – Hodnocení aplikace na základě daných otázek. V tomto případě musí uživatel často poskytnout i snímek obrazovky s danou aplikací.

Proces zadávání Uživatelské prostředí je oproti jiným platformám shodné pro pracovníka i zadavatele – v hlavní navigaci jsou viditelné volby pro vypracovávání úkolů, ale také pro správu kampaní. Nejdříve je nutné zvolit z nepřeberného množství existujících typů úloh a poté vyplníme následující nastavení nové kampaně:

- Název kampaně viditelný pro všechny pracovníky.
- Cílení projektu – Pomocí zaškrtávacích polí lze zvolit různé kombinace světových regionů a jednotlivých států, případně pracovníky z některých států vyloučit.
- Kategorie – Typ úlohy lze upřesnit pomocí dvouvrstvé kategorizace, díky které Microworkers předvyplní doporučené nastavení počtu pracovníků a odměn.
- Orientační trvání jednoho úkolu – Informační hodnota, která pracovníkům poskytne představu o výhodnosti nabídky.
- Dostupné pozice – Počet pracovníků, kteří se tohoto projektu mohou účastnit.
- Odměna – Kolik uživatel získá za vypracování jedné úlohy.
- Rychlost – Hodnota od 1 do 1000, která umožní regulovat počet současně pracujících uživatelů.
- Instrukce ke zpracování – Textové pole s možností vložení souboru.
- Důkaz – Volba zadavatele, jestli pro splnění úkolu požaduje vložení souboru (např. snímek obrazovky).
- Čas pro splnění – Počet dnů, které má pracovník k dispozici pro vypracování všech úloh.
- Automatické schvalování – Možnost urychlení procesu schvalování vypracovaných úkolů.



Obrázek 3.8: Nabídka několika úkolů v rozhraní Microworkers

Proces řešení Pracovník si může zvolit z velkého počtu probíhajících kampaní takovou, která mu vyhovuje. Projekty lze filtrovat podle kategorií, nebo také řadit podle hodnocení nebo výše odměny (viz obrázek 3.8). Pro plnění některých úkolů (zejména pro testování mobilních aplikací) je nutné ověření telefonního čísla zaslaným kódem.

Výhody

- Uživatelská základna čítající téměř 1 milion pracovníků.
- Široké možnosti nastavení kampaně a šablon.

Nedostatky

- Zastaralý design, nicméně intuitivní uživatelské rozhraní.

3.4 Další platformy

K dispozici je jen velice málo nekomerčních platforem. V průběhu analýzy existujících možností nebylo k dispozici žádné řešení, které by vyhovovalo všem stanoveným požadavkům pro univerzitní prostředí, nebo alespoň jejich větší části. Mezi hlavní nedostatky patří absence propojení s externími systémy (např. jednotné přihlašování uživatelů),

nevyhovující formát importovaných či exportovaných dat, případně také nedostatečné kontrolní mechanismy.

V posledních letech však dosáhl výrazného vývoje nekomerční projekt s názvem PYBOSSA[1]. Jedná se o modulární crowdsourcingovou platformu s otevřeným zdrojovým kódem, která je založená na programovacím jazyce Python a umožňuje široké možnosti využití. Tento nástroj lze již vnímat jako možnou alternativu k implementaci vlastní platformy. Nevýhodou může být vysoká složitost konfigurace a uživatelského prostředí, které mohou potenciální zadavatele či respondenty odradit.

Existuje i mnoho dalších nástrojů pro crowdsourcing, nicméně pro naše účely již nepřináší více funkcionality než výše zmíněné. Platformy se odlišují zejména množstvím aktivních uživatelů, což je jeden z nejpodstatnějších faktorů. Velká část vznikla v posledních letech a disponuje moderním designem a přehledným uživatelským prostředím, nicméně po stránce základny pracovníků je nelze srovnávat s předchozími příklady. Jako zdařilé příklady mohu uvést alespoň následující:

- CloudFactory[2]
  - Poskytuje široké možnosti pro strojové učení – Pracovníci mohou analyzovat přesnost nástrojů pro detekci obličeje nebo rozpoznávání psaného textu.
- OneSpace[3]
- CrowdFlower[4]

[1] http://pybossa.com/
[2] https://www.cloudfactory.com/
[3] https://www.onespace.com/
[4] https://www.crowdflower.com/

4 Crowdsourcing pro MU

V předchozí kapitole jsme se seznámili s možnostmi současných crowdsourcingových platforem. Nebyl však nalezen takový nástroj, který by splňoval všechny specifické požadavky prostředí Masarykovy univerzity a zároveň byl ekonomicky příznivým řešením pro akademické účely. Vyhovující řešení tedy musíme navrhnout a implementovat na základě vlastních potřeb.

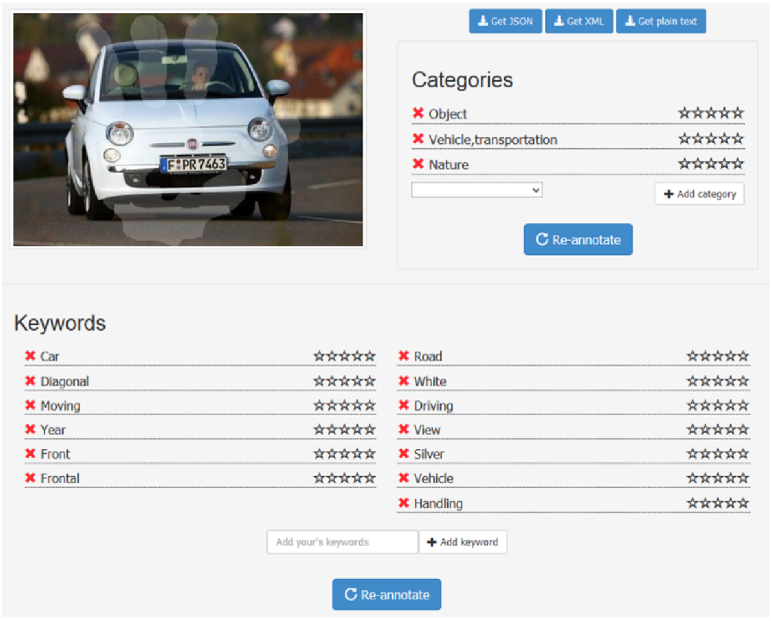
Hlavním cílem je vytvořit nástroj, který umožní dlouhodobě podporovat různé oblasti výzkumu na MU, jenž se nacházejí zejména ve fázi vyhodnocování experimentů. Přitom bude mít natolik jednoduché rozhraní pro zadávání úloh, aby svojí složitostí nebránil využívání v co největším okruhu oborů. Ať už se jedná o koncový produkt či prototyp, testování je důležitou součástí vývoje softwaru – v tomto případě se budeme soustředit zvláště na situace, kdy nízký počet testerů nestačí, a hodí se nám využít početnou skupinu dobrovolníků.

Aplikaci principů crowdsourcingu do prostředí univerzity lze demonstrovat na softwaru MUFIN Image Annotation[1]. Ten slouží pro generování anotací [12] (klíčových slov) souborům multimediálních dat na základě sémantické analýzy existujících anotací. Využívá k tomu vyhledávací software MUFIN Image Search [13], který dokáže prohledat obsah více než padesáti milionů obrázků, a to v čase kratším než půl vteřiny. Ke konkrétnímu obrázku tak MUFIN Image Annotation získá množinu klíčových slov, která jsou v databázi vázaná na vizuálně podobné obrázky. Výsledkem algoritmu je následně podmnožina relevantních anotací, získaná pravděpodobnostní analýzou.

Aby takového výsledku mohlo být dosaženo s co nejvyšší mírou přesnosti, neobejdeme se bez lidského faktoru. Nástroj pro generování anotací je výzkumným prototypem – pro jeho další rozvoj a zvyšování spolehlivosti (generování většího procenta relevantních dat) potřebujeme testovat kvalitu výsledných klíčových slov – tu jsme však schopní ověřit jen díky dobrovolníkům z řad uživatelů. Zde tedy může být crowdsourcing výhodným pomocníkem, který zajistí manuální testování výsledků tohoto softwaru.

[1] http://disa.fi.muni.cz/prototype-applications/image-annotation/

Korekce generování anotací, tím pádem i výsledků vyhledávání, probíhá převážně manuálně. Nejdříve je nutné k náhodně zvolenému vzorku obrázků nechat systém vygenerovat klíčová slova a kategorie (viz obrázek 4.1). Poté už je největší část práce na člověku. Anotace je nutné ohodnotit dle jejich relevance ke konkrétnímu obrázku. V tomto případě zvolil tvůrce hodnocení na škále od žádné (nesouvisející klíčové slovo) do pěti hvězdiček (zcela relevantní anotace).



Obrázek 4.1: Náhled uživatelského prostředí softwaru MUFIN

Takto uchopený a na malé části rozdělený problém je nyní možné rozdat mezi studenty a nechat je generovaná slova ohodnotit. Vzhledem k důležitosti co nejsprávnějších výsledků je často nutné provádět jejich kontrolu. K tomu lze využít více technik, které byly představeny v předchozích kapitolách. Výsledkem práce studentů je poté sada relevantních anotací, na jejich základě je již tvůrce schopný přesnost programu postupně vylepšovat.

4.1 Analýza potřeb

Pro analýzu potřeb byly zvoleny dvě základní metody. Tou první je přímá konzultace s výzkumným pracovníkem Fakulty informatiky (RNDr. Petra Budíková, Ph.D.), během které byly zkoumány specifické požadavky univerzity a diskutována jejich použitelnost. Druhá metoda vychází z průzkumu existujících platforem a jejich možností. V kapitole 3 již byly prozkoumány různé typy úloh, takže na základě jejich společných jmenovatelů či zásadních rozdílů lze definovat takovou sadu požadavků a případů užití, které budou vyhovovat uživatelům v co největší míře.

Potřeby je možné rozdělit do několika skupin, které obsahují základní kritéria pro vytvoření crowdsourcingové platformy. Nejdříve zjistíme, které typy úloh se hodí pro univerzitní prostředí a které naopak ne – některé z nich je možné testovat automatizovaně, jiné lze kontrolovat jen manuálně. Poté se zaměříme na uživatelské role – kdo bude se systémem pracovat a jaké bude mít možnosti. Dále nás zajímá, jaké kontrolní mechanismy a formy odměňování bude systém nabízet. Neméně důležité budou dodatečné funkce, které jsou specifické pro Masarykovu univerzitu. Na základě těchto skupin jsme již schopní specifikovat konkrétní požadavky a případy užití. V následující kapitole 5.1 z těchto požadavků vytvoříme návrh výsledné aplikace, ve kterém budou definované požadavky zohledněny.

Definice úkolů Na základě konzultace a průzkumu existujících platforem bylo nalezeno několik různých typů úkolů, které by se mohly v univerzitním prostředí hodit. Jejich seznam je doplněn krátkým popisem zamýšlených scénářů. Výpis nemusí být v této fázi úplný, proto je hlavním požadavkem také snadná rozšiřitelnost aplikace. Pro řešení různých situací zvažuje zadavatel následující typy úkolů:

- Kategorizace – Uživatel má k dispozici sadu předem definovaných kategorií, kde u každé ohodnotí, do jaké míry je relevantní k zobrazenému obrázku.
- Moderace – Pracovníkovi je zobrazena sada několika obrázků, ve které má za úkol označit takové, které nesplňují zadané kritérium.

- Testování kategorií a štítků – Z různých systémů máme k dispozici generované kategorie a štítky, ke kterým potřebujeme zjistit úroveň relevance. Uživatel má tedy za úkol každé generované slovo ohodnotit pomocí definované škály hodnot.

- A/B testování – Jednoduchá volba mezi dvěma variantami obrázku dle zadaného kritéria.

- Štítkování – Úkolem pracovníka je vepsat definovaný počet klíčových slov, která nejvíce vystihují zobrazený obrázek.

- Přepis z obrázku – Uživatel textovou formou zaznamená požadované informace poskytnuté formou obrázku.

- Dotazník – Sada otázek v podobě formuláře, ve kterém je možné definovat různé typy otázek i odpovědí (textové pole či výběrem z několika možností).

- Sběr dat – vyhledání informací z webových stránek, nebo naopak na základě poskytnutých informací najít odpovídající webové stránky.

Tyto potřeby bychom mohli rozdělit do skupin dle možností kontroly výsledků a dle jejich využití. V případě „Kategorizace“ a „Testování kategorií a štítků“ se jedná o velice podobné, široce využitelné situace. Máme k dispozici seznam předem známých klíčových slov či kategorií, které jsou totožné pro všechny uživatele. Zároveň může pracovník k těmto slovům přiřadit pouze předem definované hodnoty (procentuální míra relevance k danému objektu, případně jiná sada daných hodnot). Díky tomu můžeme testovat vypracované výsledky automatizovaně. Podobně tomu je u „Moderace“ a „A/B testování“, kdy pracujeme s více objekty na úrovni jednoho úkolu, nicméně máme stále jasně definovanou sadu možných hodnot.

Úkoly „Štítkování“, „Přepis z obrázku“, „Dotazník“ a „Sběr dat“ jsou příkladem užitečných funkcí, které automatickými testy kontrolovat nelze (hodnoty jsou většinou individuální u každého řešitele), pro naši platformu se však hodí také. Tímto způsobem lze např. zpracovávat vizuální data, která jsou pro software stále těžce rozeznatelná. Objektem k porovnání nemusí být vždy jen obrázek – může se jednat

např. o textovou informaci, video, případně zvukovou nahrávku. Jako kontrolu můžeme provádět manuální revizi náhodných výsledků.

Analýzou těchto typů úkolů docházíme k závěru, že jsou principiálně velice podobné a je možné na ně nahlížet i jiným způsobem. Tím je právě počet objektů, se kterými uživatel pracuje. Můžeme tedy zobecnit požadované typy následovně:

- Obecný úkol – Obsahuje vždy jen jeden objekt, ke kterému uživatel odpovídá na libovolné množství otázek. Tímto typem lze obsáhnout kategorizaci, testování štítků, ale také všechny formy přepisu informací a sběru dat. Typy otázek lze kombinovat, což umožní k jednomu objektu přiřadit testovatelné i volné odpovědi.

- Moderace – Uživatel má k dispozici neomezené množství objektů, ze kterých vybírá jeden nebo více možností. Způsob výběru je uživateli zdělen v instrukcích k danému úkolu. Typickým příkladem je výběr takových obrázků, které jsou relevantní k určitému tématu, případně zobrazují nevhodný obsah.

- A/B testování – Speciální případ moderace, který obsahuje přesně dva objekty a uživatel zvolí vždy jen jeden. Požadován může být například výběr vhodnější varianty obrázku dle zadaných kritérií.

Uživatelské skupiny Pro obsluhu platformy se budou univerzitě hodit jen tři uživatelské role – zadavatelé, pracovníci a administrátoři. Jelikož je uživatel autentizován svým univerzitním identifikátorem a není tím pádem pro jednu osobu možné mít více účtů současně, je nutné mít možnost tyto role kombinovat (zadavatel může být současně administrátorem). Tato potřeba je důležitým parametrem výsledné aplikace a podobný přístup lze nalézt u většiny existujících platforem.

Zadavatelé jsou lidé z Masarykovy univerzity, kteří potřebují definovat projekty a úkoly k řešení. Jednat se bude především o vyučující předmětů, kteří z různých důvodů potřebují testovat data z oblasti jejich výzkumu. Zadavatel by měl mít možnost vytvořit projekt a nastavit motivační bodové hodnocení do jím vyučovaného předmětu. Zároveň se hodí nějakým způsobem omezit skupinu pracovníků tak,

aby byli k řešení připuštěni pouze definovaní studenti. Zadávat úkoly může také kdokoli z univerzity. Účastníky projektů si však v tom případě musí motivovat jiným způsobem, než jsou body do předmětů.

Pracovníkem může být obecně kdokoli s aktivním univerzitním účtem, kdo bude mít o zpracování úkolů projektu zájem. Zejména se však budou moci zapojit studenti vybraných předmětů. Pracovník si zvolí libovolný, jemu dostupný projekt a může ihned začít řešit úkoly. Po ukončení projektu mu jsou uděleny body do souvisejícího předmětu.

Administrátoři jsou privilegovanou skupinou, která má možnost upravovat veškeré informace v systému – např. projekty, uživatele, nebo i jednotlivé úkoly. Mohou také uživatelům (pracovníkům) přidělovat oprávnění (např. stát se zadavatelem projektů).

Systém odměňování Jednou ze základních potřeb zadavatele je také definování odměny za splněné úkoly. Tím budou bonusové body do předmětu, pro který je projekt vytvořen. Tyto body je možné udělovat v univerzitním informačním systému manuálně či také pomocí hromadného importu. Studenti tak mohou získat drobnou výhodu při průchodu předmětem, případně lepší výslednou známku. Ostatním pracovníkům může stačit i jen dobrý pocit z pomoci ve výzkumu, případně určí zadavatel jinou formu odměny a specifikuje ji v popisu projektu.

Zadavateli se hodí i možnost omezení výsledné odměny pro jednoho uživatele – s definováním počtu bodů za jeden splněný úkol bude vhodné mít zároveň možnost nastavit i horní limit, aby nedošlo k vyčerpání dostupných bodů rychlejší skupinou studentů.

Kontrolní mechanismy Techniky pro odhalování správnosti poskytnutých výsledků byly analyzovány v kapitole 2.4. Z konzultovaných požadavků univerzity vyplývá, že je pro naše potřeby využitelný zejména výpočet většinového názoru (algoritmus 1), ale také odvozená varianta s kontrolními otázkami (algoritmus 2). Pro zajištění kvalitnějších výsledků je dále požadována možnost manuální kontroly odeslaných výsledků a selektivní vyřazování nepoctivých pracovníků.

Zadavatel by si tedy mohl definovat požadovanou úroveň kvality výběrem mezi prostou kontrolou (většinový názor) a algoritmem

s použitím kontrolních otázek. Dále je důležité specifikovat požadovaný počet řešitelů na jeden úkol (úroveň redundance) – ten je přímo úměrný přesnosti výsledků. Vzhledem k relativně malému množství potenciálních respondentů bude obecně vhodnější zpracovávat více úkolů, než využívat vysokou míru redundance.

Diskutovaným tématem bylo také skóre pracovníků ve vztahu k jednotlivým projektům. První možností je vnímat uživatele jako bezchybného na počátku každého nového projektu a vyhodnocovat jej nezávisle na ostatních projektech. Důsledkem této varianty se nepoctiví pracovníci dostávají do výhody oproti poctivým, protože s každým projektem začínají na stejné úrovni. Druhou a preferovanou možností je přepočítávání skóre s každým novým projektem (počáteční úroveň uživatelů se bude lišit na základě jejich historie). Poctiví pracovníci tak zajistí vyšší pravděpodobnost kvalitních výsledků a nepoctiví mají vždy šanci své skóre zlepšit.

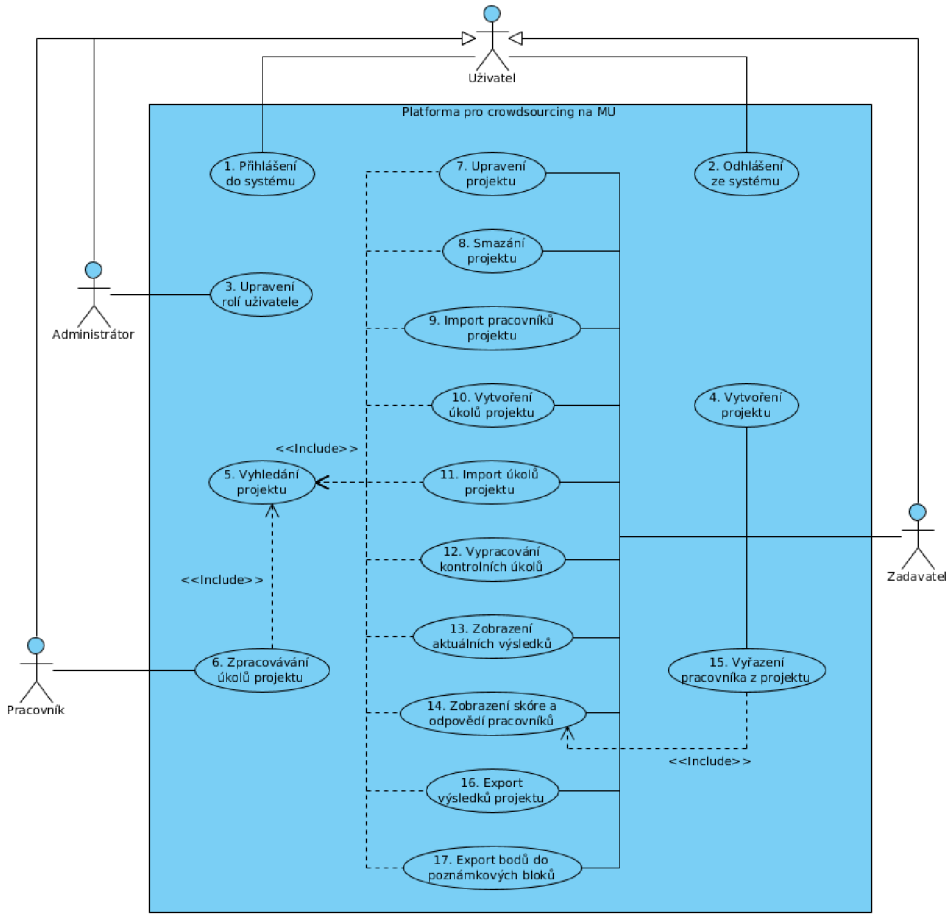
Specifické funkce Oproti ostatním platformám potřebujeme možnost snadné autentizace uživatelů přes systém Masarykovy univerzity. Zadavatel ani pracovník tak nebude již z počátku demotivován nutnou registrací a použitím dalších hesel. Pro takové případy má univerzita k dispozici systém jednotného přihlášení, který lze do nové aplikace integrovat.

Pro usnadnění udělování bonusových bodů do předmětu se nám hodí i komunikace s aplikací „Poznámkové bloky“, která je součástí informačního systému. Jelikož tento nástroj disponuje možností exportu a importu dat[1] ve formátu TSV[2], můžeme jej využít a jednotlivým uživatelům tak připsat body za konkrétní projekt přímo do vybraného předmětu.

[1] https://is.muni.cz/napoveda/ucitel/bloky#b_import
[2] Tabulátorem oddělené hodnoty

4.2 Případy užití

Na základě předchozí analýzy je již možné specifikovat požadované případy užití (viz obrázek 4.2). Jejich kompletní přehled je obsažen v kapitole A.1 přílohy. V této sekci zformulujeme princip netriviálních případů.



Obrázek 4.2: Diagram případů užití platformy

Aktéři Seznam aktérů je odvozen z uživatelských skupin.

- Uživatel – generalizace speciálních případů
- Administrátor – uživatelská role pro správu uživatelů
- Zadavatel – uživatelská role pro správu projektů
- Pracovník – uživatelská role pro zpracovávání úkolů

UC_4 – Vytvoření projektu Před vložení jednotlivých úkolů musí zadavatel vytvořit projekt, v rámci kterého probíhají veškeré výpočty. Ten slouží k definování obecných informací a instrukcí k vypracovávání úkolů. Nastavením projektu lze určit zejména omezení pro jeho trvání (datum začátku a konce řešení), výši odměny (počet bodů za jeden splněný úkol), ale také použitý kontrolní algoritmus. Zadavatel nastaví požadované vlastnosti pomocí formuláře a po jeho uložení lze vytvářet nebo importovat jednotlivé úkoly.

UC_6 – Zpracovávání úkolů projektu Jakmile existují v projektu nezpracované úkoly a lze je v daný moment řešit, pracovník může začít s jejich vypracováváním. Nejdříve jsou uživateli předloženy kontrolní otázky (pokud existují), aby bylo možné stanovit jeho skóre (úroveň přesnosti). Poté již uživatel zpracovává úkoly v pořadí daném aplikací. Diskutováno bylo zpracovávání úkolů takovým způsobem, aby se co nejdříve naplnila nastavená redundance (pro zajištění vysoké kvality výsledků), nicméně jako vhodnější byl zvolený opačný přístup. Tím je snaha o co nejrychlejší získání alespoň jednoho zpracování odpovědí ke každému úkolu. Uživatel tedy zpracovává nejdříve ty úkoly, které mají nejmenší počet odeslaných řešení a postupným přidávání dalších řešení dochází k zpřesňování výsledků.

UC_9 – Import pracovníků projektu Zadavatelé projektů mohou chtít omezit řešení úkolů pouze pro konkrétní skupinu pracovníků (např. pro studenty některého z předmětů). Univerzitní informační systém poskytuje možnost exportu seznamu všech studentů daného předmětu. Tento formát lze použít pro přímý import do nastavení projektu, kde bude seznam přiřazen ke kódu předmětu a bude později

sloužit pro následný export bodů do poznámkového bloku. Zároveň může zadavatel vyučovat několik předmětů současně, takže jsou seznamy studentů oddělené.

UC_10 – Vytvoření úkolů projektu Úkoly projektu je možné definovat také manuálně pomocí formuláře. Každý úkol se skládá z jednotlivých otázek a pracovníci vkládají své odpovědi. Otázky je možné rozlišit na globální (definované na úrovni projektu) a úkolové (specifické pro konkrétní úkol). Globální otázky jsou následně používané pro každý úkol z projektu. Může se jednat např. o obecné kategorizace, do kterých jsou zařazovány objekty z jednotlivých úkolů. Naopak úkolové otázky jsou přímo vztažené k objektu jednoho úkolu. Zadavatel může k úkolům dle potřeby nahrávat soubory z disku (tím může být obrázek, video, případně zvuková nahrávka).

UC_11 – Import úkolů projektu Pro větší objem dat je vhodnější využít zjednodušenou formu vytváření úkolů. Tím je import pomocí formátu dat JSON[1]. Zadavatel tak může v jednom kroku vložit definice úkolů (včetně obou typů otázek), nahrát soubory z disku, ale také vložit kontrolní úkoly. Přesná struktura vstupních dat je popsána v kapitole A.4 přílohy.

UC_12 – Vypracování kontrolních úkolů Využití kontrolních úkolů bylo shledáno jako jedna z možností pro upřesnění výpočtu správných odpovědí. Pokud tedy zadavatel zvolí algoritmus s využitím kontrolních otázek, je možné vzorově vypracovat neomezené množství úkolů. Ty jsou poté řešeny každým uživatelem přednostně. Přesnost odpovědí každého pracovníka je následně použita jako váha jeho dalších odpovědí. Uživatelé se špatným skóre z kontrolních otázek tak budou ovlivňovat výsledky jen minimálně.

UC_15 – Vyřazení pracovníka z projektu Po výpočtu kvality odpovědí zúčastněným pracovníků má zadavatel možnost prohlížet si aktuální skóre i vypracované úkoly pracovníků. Pokud některý zúčastněný uživatel nesplňuje představy zadavatele špatnou kvalitou

[1] JavaScript Object Notation

odpovědí, je možné jej vyřadit z projektu. Tím dojde k označení všech jeho odpovědí jako vyřazené a tyto odpovědi dále nebudou do výpočtu zahrnuty.

UC_16 – Export výsledků projektu Ukončení projektu a export vypočítaných výsledků probíhá dle potřeby zadavatele. Ten má možnost projekt předčasně ukončit, ale také prodloužit jeho platnost v závislosti na tom, kolik vypracovaných úkolů uživatelé poskytli. V případě dostatečného množství odpovědí lze zahájit export výsledků do formátu JSON. Na základě volby kontrolního algoritmu systém prochází všechny úkoly a pro každou odpověď vypočítá procentuální pravděpodobnost, že daná možnost je správná.

5 Implementace

Platforma bude implementována formou webové aplikace, což eliminuje nutnost instalace jakéhokoli dodatečného softwaru na klientská zařízení – uživatelům stačí libovolný webový prohlížeč a mohou systém využít i na jakémkoli mobilním zařízení. Díky tomu, že je software nasazen na serveru, je také usnadněn budoucí rozvoj a není nutné současně spravovat více verzí aplikace.

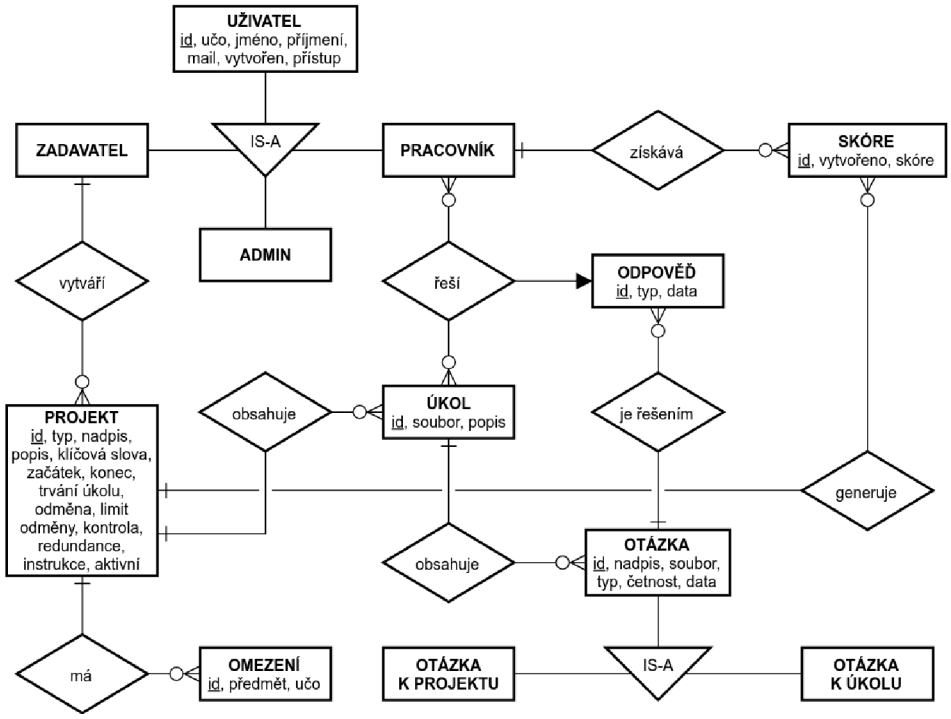
V předchozí kapitole byly analyzovány funkční požadavky. Tedy veškerá interakce uživatele s aplikací. Naopak nefunkční požadavky jsou založené např. na požadované architektuře či na kvalitativních aspektech (např. výkon či design). Mezi takové lze pro naši implementaci navrhnout následující:

1. Aplikace a veškeré její funkce jsou dostupné přes webové rozhraní pro desktopová i mobilní zařízení (responzivní design).
2. Pro ukládání informací aplikace využije databázový systém.
3. Architektura aplikace respektuje ověřené návrhové vzory používané u podobných projektů – např. členění do samostatných vrstev (persistentní, servisní, webová, případně další dle potřeby).
4. Webová vrstva využívá tříúrovňový koncept MVC[1]
5. Aplikace podporuje lokalizaci, internacionalizaci a logování.
6. Pro přihlášení uživatelů pomocí UČO[2] a sekundárního hesla je integrován systém jednotného přihlášení MU.
7. Služba je dostupná nepřetržitě, s výjimkou nevyhnutelných odstávek.
8. Jsou dodržovány základní principy bezpečnosti s kontrolou oprávnění a ochranou proti běžným útokům.
9. Aplikace umožňuje její škálování a budoucí rozšiřitelnost.

[1] Model-view-controller („datový model, uživatelské rozhraní, řídící logika“)
[2] Univerzitní číslo osoby

5.1 Návrh systému

Na základě požadavků ze strany zadavatele byl navržen zjednodušený entitně relační model plánovaného systému (viz obrázek A.6). Ten obsahuje několik základních objektů, které mezi sebou mají požadované vztahy.



Obrázek 5.1: Návrh entitně vztahového modelu platformy

Základní entitou je v tomto případě „**Uživatel**“, od kterého se odvíjí všechny ostatní relace. Jedná se o nadřazenou entitu (generalizaci) pro role zadavatele, pracovníka i administrátora a může nabývat všech specializací současně (nejméně však jedné z nich). V běžném případě bude uživatel mít pouze jednu roli, nicméně např. někteří zadavatelé můžou mít zájem i o roli pracovníka. Uživatel obsahuje společné atributy pro všechny specializované entity: základní identifikaci osoby (UČO, jméno, příjmení, e-mailovou adresu) a časová razítka pohybu

v systému (první přihlášení a posledního přístupu). Specializované role neobsahují žádné dodatečné atributy oproti uživateli.

„**Administrátor**“ je speciálním uživatelem, který dokáže obsluhovat všechny funkce systému a není tím pádem limitován žádným ověřováním na úrovni oprávnění. Má na starosti zejména správu uživatelů, např. přidělování uživatelských rolí. První přihlášený uživatel v systému bude vždy automaticky administrátorem.

„**Zadavatel**“ je další specializovanou rolí uživatele, která se stará o vytváření projektů a úkolů s otázkami. V námi navrhované platformě je důležitý pro veškerou organizaci projektů a řízení jejich vyhodnocování. Zadavatel má přímou relaci na entitu „Projekt“ a může jich vytvořit neomezené množství (0 až N).

„**Pracovník**“ je posledním speciálním případem uživatele a plní funkci řešitele úkolů vytvořených zadavatelem. Za svoji činnost mu je průběžně vypočítáváno skóre (0 až N), které je uchováváno včetně minulých stavů, aby bylo možné vypočítávat průměr ze všech projektů. Hlavní činností pracovníka je plnění úkolů projektu (0 až N), kdy vyřešením úkolu vznikají entity odpovědí. Na základě vyřešení jednoho úkolu (sady M otázek) tedy z relace „řeší“ vznikne sada M odpovědí.

„**Projekt**“ obsahuje nastavení, která specifikuje zadavatel. Jedná se o tyto atributy: typ (např. „Obecný projekt“), nadpis, popis, klíčová slova (pro vyhledávání mezi ostatními projekty), datum začátku a konce řešení (kdy je možné řešit úkoly pracovníkem), orientační trvání jednoho úkolu (pro představu o náročnosti), odměna (počet bodů získaných za jeden vypracovaný úkol), limit odměny (maximální počet bodů, které může jeden pracovník získat), kontrolní algoritmus (např. „většinový názor“), redundance (počet pracovníků na 1 totožný úkol), instrukce pro vypracování (textový popis se zadáním projektu), aktivní (jestli je projekt možné řešit). Entita má vazbu na úkoly (0 až N) – maximální počet vypracovaných řešení je tedy roven počtu dostupných úkolů násobeného redundancí. Další vazba je k entitě „Omezení“ (0 až N) – tu lze využít v případě, kdy bude chtít zadavatel limitovat účast v projektu na konkrétní seznam pracovníků. Dále vazba k entitě „Skóre“ (0 až N) vzniká v případě, kdy je projekt vyhodnocen a pracovníkům je vypočítána jejich přesnost.

„**Omezení**“ má atributy kód předmětu a UČO a je vázáno k projektu (1). Vazba na konkrétního pracovníka zde není z toho důvodu,

že jeho entita v systému zatím nemusí existovat. Zadavatel takto může vytvořit několik samostatných seznamů (studenti daných předmětů) a tím omezit účast na projektu pouze těmto pracovníkům.

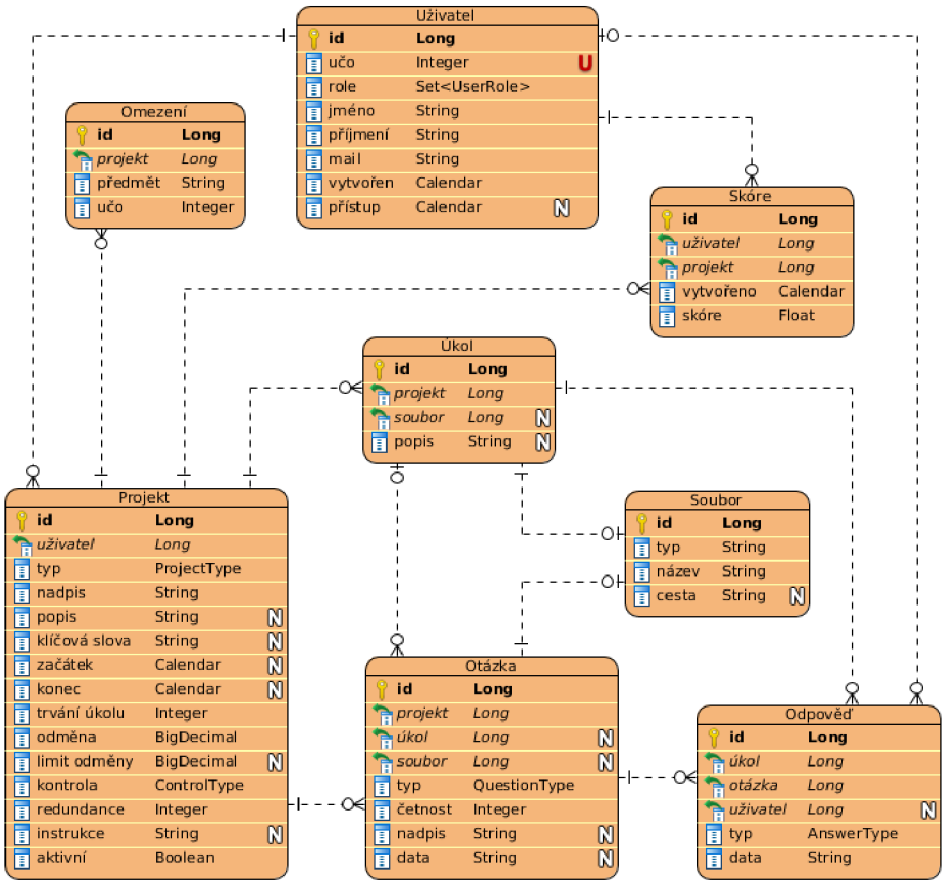
„**Úkol**“ je nosnou entitou pro všechny obsažené otázky. Obsahuje vazbu na projekt (1) a je možné ho řešit pracovníkem. Entita má pouze tyto atributy: soubor (objekt, ke kterému uživatel odpovídá na otázky) a popis (dodatečná informace k úkolu). Jeden konkrétní úkol obsahuje neomezené množství otázek (0 až N) – některé jsou globální (projektové) a jiné specifický pro daný objekt. Uživatel po zodpovězení otázek generuje odpovědi (ve stejném množství, jako bylo otázek).

„**Otázka**“ je nadřazenou (generalizace) entitou pro dva samostatné typy. Může se jednat o „Otázku k projektu“ (globální), ale také o specifickou „Otázku k úkolu“. Důvodem k tomuto typu vazeb je splnění požadavků univerzity – některé otázky totiž díky tomu mohou být zadané globálně pro projekt (zobrazí se shodné u každého úkolu) a ostatní mohou být specifické pro daný úkol. Můžeme tak např. definovat obecné kategorie shodné pro všechny úkoly, ale také kategorie specifické pro daný úkol. Entita má vazbu na odpovědi (0 až N). Atributy jsou následující: nadpis (např. „Vyberte odpovídající kategorie“), typ (např. „Výběr s více hodnotami“), četnost (kolik hodnot uživatel vybírá), data (sada možností, např. „Sport“, „Kultura“, „Architektura“) a soubor (pokud je pro daný typ projektu vyžadován – např. moderace obrázků).

„**Odpověď**“ je vytvořena pokaždé, když pracovník zpracuje úkol – poté je pro každou otázku v úkolu vytvořena odpovídající odpověď. Tímto zajistíme zpracovatelnou strukturu dat, které nám pomohou efektivně provádět výpočty výsledků. Atributy jsou: typ odpovědi (např. „uživatelská“ či „vyřazená“) a data (hodnoty zvolených odpovědí).

Poslední součástí modelu je „**Skóre**“. Tato entita vzniká na základě výpočtu úspěšnosti pracovníka v konkrétním projektu. Obsaženými atributy jsou pouze datum vytvoření a hodnota skóre (0 až 1) – ta udává procentuální kvalitu s maximem 100 %.

Dalším důležitým modelem je návrh databázové struktury. Entitně relační model byl do této formy převeden v souladu s navrženou funkčností, aby zachoval efektivní práci s výpočty. Oproti předchozímu modelu zejména převádí dědičnost uživatele na množinu rolí. Důležité je také zmínit, že oba typy otázky jsou sjednoceny pod jednu entitu a liší se jen obsaženou či chybějící hodnotou v referenci na úkol. Poslední změnou je přidání samostatné entity pro soubor, což usnadní práci s médii.



Obrázek 5.2: Návrh datového modelu platformy

V rámci modelu jsou využité následující číselníky:

- UserRole (uživatelská role): Administrátor, Zadavatel, Pracovník
- ProjectType (typ projektu): Obecný (jeden objekt a různé typy otázek), Moderace (výběr z několika objektů), A/B testování (dva objekty)
- ControlType (kontrolní algoritmus): Většinový hlas, Kontrolní otázky
- QuestionType (typ otázky): Ano/Ne (dvě možné hodnoty), Procentuální hodnocení (několik kroků), Výběrový box (volba z několika hodnot), Seznam možností (označení jedné nebo více voleb), Textové pole (jednořádkový vstup), Textová oblast (víceřádkový vstup)
- AnswerType (typ odpovědi): Pracovník (vložená pracovníkem), Kontrolní (vložená zadavatelem), Výsledná (vypočítaná odpověď), Vyřazená (vložená pracovníkem a vyřazená zadavatelem)

Uživatelské rozhraní Uživatelské rozhraní bylo navrhováno se snahou co největší flexibility a jednoduchosti. Využívá responzivního designu, takže se prostředí přizpůsobuje velikosti displeje a je tedy využitelné na mnoha různých typech zařízení. Obrázky prezentující jednotlivé části navrženého rozhraní jsou umístěny v kapitole A.6 přílohy.

5.2 Struktura aplikace

Pro implementaci a budoucí rozvoj crowdsourcingové platformy byl zvolen objektově orientovaný programovací jazyk Java, který je jedním z nejpoužívanějších na světě. V úvahu přicházel i skriptovací jazyk PHP – a to hlavně z důvodu snadné použitelnosti, případně další obecně využitelné jazyky Ruby nebo Python.

Platforma je implementována formou webové aplikace, což eliminuje nutnost instalace jakéhokoli dodatečného softwaru na klientská zařízení – uživatelům stačí libovolný webový prohlížeč a mohou systém využít i na jakémkoli mobilním zařízení. Díky tomu, že je software

nasazen na serveru, je také usnadněn budoucí rozvoj a není nutné současně spravovat více verzí aplikace.

Během vývoje byly využity nástroje usnadňující programování:

- Git – Systém pro správu verzí zdrojových souborů.
- Apache Maven – Nástroj pro správu použitých balíčků a pro automatizaci procesů kompilace a nasazení.
- IntelliJ IDEA – Vývojové prostředí s validací zdrojového kódu a s dalšími podpůrnými nástroji.

Některé části zdrojového kódu aplikace byly strojově generovány, nicméně většinu z nich bylo nutné během vývoje upravovat manuálně. Nejvíce takového kódu obsahují balíky „cz.muni.fi.muc.entity“ a „cz.muni.fi.muc.dto“. Přehled generovaných částí:

- Metody atributů getNazevAtributu(), setNazevAtributu()
- Pomocné metody equals(), hashCode() a toString()

Projekt je rozdělen do čtyř samostatných modulů se vzájemnými závislostmi – jednotlivé vrstvy tak mohou manipulovat pouze s takovými komponentami a vykonávat jen takové funkce, které jsou v jejich kompetenci. Základní závislostí celého systému je řídící komponenta „Spring IO platform“, která se stará o definování správných verzí většiny použitých závislostí. Díky ní je možné přecházet na vyšší verze bez komplikovaného testování jejich různých kombinací.

Nejspodnějším modulem je „**perzistentní vrstva**“. Ta manipuluje přímo s databází a stará se o její konfiguraci. Hlavní součástí jsou třídy všech systémových entit, které definují jednotlivé databázové tabulky. Obslužnou částí jsou tzv. DAO[1] třídy, poskytující veškerá databázová volání – vytváření instancí entit, jejich editace, mazání a získávání na základě specifických kritérií. Důležitým prvkem vrstvy jsou testy komponent, zajišťující ověřování základní funkčnosti aplikace. Poslední částí jsou tzv. číselníky, které nám poskytují typové hodnoty pro atributy některých entit.

[1] Data access object

- Spring Data JPA – Konfigurační nástroj pro přístup k databázi.
- PostgreSQL – Poskytuje databázi, která nám umožňuje flexibilní operace s daty.
- Hibernate Validator – Specifikuje pravidla pro validaci atributů entit.
- Java Persistence – API pro řízení objektově relačního mapování.
- Spring Test a TestNG – Nástroje pro testování komponent.
- H2 Database Engine – Databáze používaná pro spouštění testů v operační paměti.

Mezičlánkem je „**API vrstva**“, která poskytuje třídy užitečné v nadřazených vrstvách. Jedná se zejména o komponenty DTO[1], které jsou lépe využitelné pro specifické situace, než samotné entity. Díky nim můžeme upřesňovat různá validační kritéria pro konkrétní práce s entitami. Nalezneme zde také pomocné editační třídy, převádějící formulářové hodnoty na objekty a opačně. Součástí jsou také definice samotných formulářů nebo třídy pro JSON[2] mapování.

Nadřazenou částí je poté „**servisní vrstva**“. Poskytuje konfiguraci služeb pro využití v poslední vrstvě a využívá DAO komponenty pro manipulaci s entitami. Jako součást crowdsourcingové platformy jsou navíc implementovány tzv. „fasádní služby“, sloužící pro převod entit na přechodné objekty (DTO) a zpět. V této vrstvě je implementována i služba pro předávání dodatečných informací ze systému jednotného přihlášení MU (třída SAMLUserDetailsServiceImpl).

- Spring Security SAML – Umožňuje komunikaci mezi poskytovatelem identit a naší implementací služby.
- Dozer – Nástroj pro mapování přechodných tříd na entity.
- Jackson Databind – Nástroj usnadňující manipulaci s různými formáty pro výměnu dat.

[1] Data transfer object
[2] JavaScript Object Notation

- Opencsv – Jednoduchá knihovna pro zpracování dat ve formátu CSV.

Poslední a zároveň nejobsáhlejší částí je „**MVC vrstva**“, starající se celou aplikační logiku – zajišťuje předávání dat (poskytovaných servisní vrstvou) pro zpracování a případné zobrazení uživateli. Pro svoji obsáhlost je vrstva dále rozdělena na kontrolní třídy a pohledové šablony. Veškeré logické operace probíhají výhradně v kontrolních třídách a výsledná data jsou předána pro zobrazení do specifických šablon. Vrstva také obsahuje nejobsáhlejší část konfigurace celé platformy, řídí mapování požadavků pro zpracování jednotlivými kontrolními třídami, ale řeší také zabezpečení aplikace.

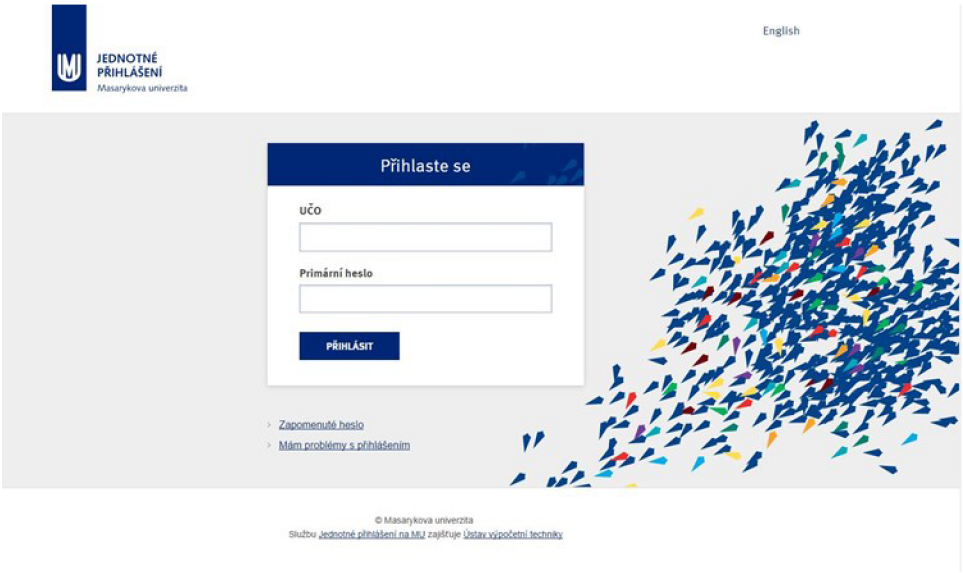
- Spring Web MVC – Umožňuje mapování zdrojů aplikace na požadavky zasílané klientem.
- Spring Security Taglibs – Veškeré bezpečnostní nástroje.
- Apache Tiles – Komponenta pro centrální definování hierarchie šablonového systému.
- JavaServer Pages Standard Tag Library – Poskytuje sady elementů pro zpracování dat v šablonách.
- SLF4J – Komponenta pro logování.

Všechny výše uvedené technologie pracují na straně serveru, nicméně využity byly i nástroje pracující na straně klienta (ve webovém prohlížeči).

- Bootstrap – Populární sada stylů a skriptů pro vytváření moderních responzivních projektů se snadnou rozšiřitelností.
- CKEditor – Jednoduchý WYSIWYG editor pro snadnou práci s textem.
- Bootstrap Star Rating – Knihovna poskytující nástroj pro hlasování pomocí volby z několika hvězdiček.

5.3 Jednotné přihlašování

Pro snadný přístup stávajících uživatelů univerzitního systému bylo možné využít autentizační službu „Jednotné přihlášení na MU“[1], kterou zajišťuje Ústav výpočetní techniky Masarykovy univerzity. Ten našemu projektu vyšel vstříc a na základě žádosti službu aktivoval pro využití v crowdsourcingové platformě.



Obrázek 5.3: Jednotné přihlášení na MU

Princip fungování je na bázi propojení univerzity jako poskytovatele identit a naší platformy jako poskytovatele služby (zkráceně IdP a SP[2]) – crowdsourcingová aplikace tedy nemusí spravovat uživatelské účty svépomocí, ale místo toho dostává základní informace o autentizované osobě ze strany MU. Součástí nastavení je také zasílání sady několika dodatečných údajů, které lze s poskytovatelem identit dohodnout na míru aplikace. V našem případě se jedná o UČO jako hlavní identifikátor, křestní jméno, příjmení a e-mailovou adresu dotyčné osoby. Tyto atributy slouží pro personalizaci služby a jsou vždy

[1] https://it.muni.cz/sluzby/jednotne-prihlaseni-na-muni
[2] Identity Provider, Service Provider

aktuální (po každém úspěšném přihlášení dojde k jejich obnovení i v našem systému). Výhodou tohoto přístupu je fakt, že poskytovatel služby nedochází do kontaktu s heslem uživatele a nemá tím pádem přístup do informačního systému MU.

Z pohledu serverové architektury stojí za poskytováním jednotného přihlašování komponenta s otevřeným zdrojovým kódem Shibboleth a pro samotnou komunikaci mezi IdP a SP slouží standard SAML[1], který je mechanismem pro výměnu autentizačních a autorizačních dat. Aplikační část obsluhuje balíček Spring Security SAML, který je nakonfigurován na míru naší platformy.

5.4 Testování

Pro testování výsledné aplikace a implementovaných algoritmů byly zvoleny dva přístupy. Prvním je automatizované testování komponent na úrovni perzistentní vrstvy, které probíhá vždy při sestavování aplikace. Bylo implementováno více než 100 samostatných testů, ověřující očekávané chování aplikace. Druhým přístupem je využití experimentů ve skutečných podmínkách. Základem byla sada reálných dat generovaná softwarem MUFIN Image Annotation[2] a skupina pěti dobrovolníků z řad studentů univerzity.

Vstupní data jsou složena z dvaceti obrázků, u kterých je přiřazeno vždy padesát kandidátních klíčových slov. Takové množství by nebylo snadné testovat v kompletní podobě – proto bylo zvoleno jen 10 obrázků, vždy s 5 klíčovými slovy, u kterých bylo možné snadno odhalit, jestli jsou správná nebo ne. Těchto 10 pokusných úkolů po pěti klíčových slovech (celkem tedy 50 hodnocených slov) bylo předáno k vypracování všemi dobrovolníky.

Jeden z pracovníků dostal instrukci vybírat hodnocení naprosto náhodně, abychom tak simulovali nepoctivého pracovníka. Ostatní měli za úkol hodnotit s co nejvyšší přesností, nicméně chyby byly přípustné. Hodnocení probíhalo metodou deseti hvězdiček (čím více vybraných, tím vyšší přesnost dané klíčové slovo má). Hodnotícím algoritmem byla „většinová volba“, tedy varianta kontroly bez vzorově vypracovaných otázek.

[1] Security Assertion Markup Language

[2] http://disa.fi.muni.cz/prototype-applications/image-annotation/

Výsledná data byla analyzována a exportována do podoby přehledných tabulek a jsou dostupná v kapitole A.5 přílohy. V každé z nich se nachází tato data:

- Počáteční a výsledné skóre každého pracovníka (počáteční je vždy 100 procent, výsledné v závislosti na přesnosti práce).
- Hodnocení každého klíčového slova od každého z pracovníků. Pro přehlednost jsou jednotlivá klíčová slova v samostatných tabulkách. Každé jednotlivé hodnocení je doplněno průměrem všech pracovníků a také odvozenou správností odpovědi (1 pro správné a 0 pro chybné).

První experiment (viz obrázek A.1) dopadl očekávaným způsobem. Nepoctivý pracovník byl odhalen a jeho skóre nastaveno na výrazně nižší hodnotu, než mají ostatní účastníci. Druhý experiment (viz obrázek A.2) byl vstupními podmínkami téměř totožný – rozdíl byl pouze v tom, že nepoctiví pracovníci byli tentokrát dva ze tří. Opět se potvrdilo stejné zjištění jako v prvním případě a oba nepoctiví pracovníci byli odhaleni.

Z těchto dat vyplývá, že systém je vhodný pro sběr výzkumných údajů. Dle zpětné vazby od pokusných osob se aplikace používá bez potíží a podobný způsob získávání bonusových bodů do předmětů by v budoucnu rádi využívali.

6 Závěr

Cílem práce bylo především analyzovat, navrhnout a implementovat prototyp crowdsourcingové platformy pro Masarykovu univerzitu. Nejdříve tedy byla představena samotná problematika crowdsourcingu s hlavními principy a důležitými aspekty. Poté proběhlo porovnání vybraných existujících marketingových nástrojů pro zprostředkování práce mezi zadavateli a širokou veřejností. Pro důkladné seznámení s principy crowdsourcingu posloužil široký výběr dostupných informačních zdrojů, z nichž velká část obsahovala myšlenky s přesahem i do jiných problematik. Z tohoto průzkumu vyplynulo, že současný trh nenabízí žádný vhodný nástroj, který by byl snadno a za vyhovujících podmínek použitelný v univerzitním prostředí.

Důležitou součástí teoretické kapitoly je průzkum algoritmů pro kontrolu kvality získaných výsledků a pro dlouhodobé sledování poctivosti uživatelů, který přinesl nejpoužívanější přístupy řešící tuto problematiku. Většina zmíněných postupů byla využita v nově navrhované univerzitní platformě.

Na základě srovnání s existujícími nástroji byly ve druhé části práce analyzovány potřeby univerzity a byla definována sada obecných požadavků. Z této analýzy vzešel návrh typů úloh a použitých technik pro hlídání kvality výsledků. Současně s tímto krokem bylo nutné specifikovat uživatelské role a také systém odměňování, který je pro univerzitní prostředí vyřešen možností udělování bonusových bodů do předmětů.

Praktickým výstupem diplomové práce je prototypová implementace platformy, u které bylo nutné navrhnout jednotlivé vrstvy webové aplikace a jednoduché uživatelské prostředí. Pro snadnější plnění této části bylo přínosné nejdříve definovat nejdůležitější případy užití. Součástí implementace je využití systému jednotného přihlašování, které Masarykova univerzita poskytuje a není tedy nutné vyžadovat po uživatelích nadbytečný proces registrace. Po implementaci všech součástí bylo provedeno testování pomocí navržených scénářů, které ověřily její funkčnost.

6.1 Naplnění cílů

V úvodní kapitole byly stanoveny dílčí úkoly, které jsou pro tuto práci důležité. Závěrem tedy můžeme plnění jednotlivých cílů zhodnotit:

- Prozkoumat známé crowdsourcingové platformy a pokud existuje, najít nástroj využitelný pro naše účely.

Podobných komerčních platforem existuje mnoho, nicméně prozkoumány byly takové, které svou uživatelskou základnou tvoří podstatnou část dnešního trhu. Jedná se především o systémy s dlouhou tradicí, nicméně v současné době se rozvíjí také mnoho modernějších aplikací – ty zatím neobsahují významné inovace oproti svým předchůdcům. Mezi volně dostupnými nástroji však nebylo nalezeno takové řešení, které by bylo vhodné použít jako základ pro novou univerzitní aplikaci.

- Na základě existujících platforem a potřeb univerzity zvolit podporované typy úloh a způsob jejich zadávání.

Díky již existujícím platformám nebylo obtížné zvolit takové úlohy a způsoby zadávání, které vyhovují potřebám univerzity. Sada podporovaných úloh nyní obsahuje základní nástroje a samotná aplikace je implementována s ohledem na budoucí rozšiřitelnost.

- Vybrat nebo navrhnout techniky pro hlídání kvality výsledků a v případě potřeby je rozdělit podle typu úlohy.

Zvoleny byly takové techniky, které jsou využitelné pro všechny typy úkolů s pevně definovanou sadou možností. Pro ostatní úlohy je možné kontrolu kvality provádět manuálně, např. náhodně zadavatelem.

- Navrhnout základní objekty a procesy crowdsourcingové aplikace, případně i možnosti komunikace s externími službami.

Platforma pracuje s hodnocením textu, obrázků, videí a audio nahrávek. Díky univerzální implementaci je možné ji rozšířit pro podporu libovolného dalšího multimediálního obsahu. Také komunikace s dalšími službami probíhá formou předávání dat v platformě nezávislém formátu JSON, což umožňuje snadné definování úkolů k projektům, případně import výsledků do jiných systémů.

- Navrhnout uživatelské prostředí pro zadávání úloh včetně vhodného šablonového systému pro definování prvků aplikace.

Nástroj lze snadno ovládat bez speciálních znalostí – díky využití moderního a často využívaného šablonového systému je uživatelské prostředí jednoduché a intuitivní. V případě potřeby lze tedy např. použít odlišný grafický design, aniž by bylo nutné zasahovat do ostatních vrstev systému.

- Zvolit technologie pro implementaci, ať už z osvědčených programovacích jazyků a nástrojů, nebo z vhodných alternativ.

Všechny součásti byly zvoleny tak, aby co nejvíce respektovaly standardní vývojové postupy a nevybočovaly do nestabilních oblastí. Další rozvoj aplikace tím pádem může probíhat bez speciálních znalostí a bez závislosti na nepodporovaných komponentách.

6.2 Vize budoucího rozvoje

Samotná platforma je již pro běžné využití zcela připravena, nicméně lze najít možnosti dalšího rozšíření. Z pohledu zadavatele mohou být v rámci budoucího rozšíření požadované statistické přehledy, případně možnosti hromadného rozesílání e-mailů (všem účastníkům projektu, těm kteří získali body, případně vyřazeným). Dále je možné přidat automatizované upozorňování na účty s výrazně horší kvalitou odpovědí a také podrobnější informace o aktuálním průběhu a plnění probíhajících projektů.

Ze strany pracovníků se může jednat o přehlednou statistiku získaných bodů, případně výpis bodů čekajících na udělení. Administrátoři by mohli mít k dispozici např. dlouhodobé přehledy za určitá období (např. počet uživatelů podle rolí, probíhající a vyřešené úkoly, získané body nebo průměrná doba řešení úkolu).

Bibliografie

1. HOWE, Jeff. *Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business*. 1. vyd. New York, NY, USA: Three Rivers Press, 2008. ISBN 978-0-307-39621-1.
2. HOWE, Jeff. The rise of crowdsourcing. *Wired magazine*. 2006, roč. 14, č. 6, s. 1–4.
3. DOAN, Anhai; RAMAKRISHNAN, Raghu; HALEVY, Alon Y. Crowdsourcing Systems on the World-Wide Web. *Commun. ACM*. 2011, roč. 54, č. 4, s. 86–96. ISSN 0001-0782. Dostupné z DOI: 10.1145/1924421.1924442.
4. SAXTON, Gregory D.; OH, Onook; KISHORE, Rajiv. Rules of Crowdsourcing: Models, Issues, and Systems of Control. *Information Systems Management*. 2013, roč. 30, č. 1, s. 2–20. Dostupné z DOI: 10.1080/10580530.2013.739883.
5. EICKHOFF, Carsten; HARRIS, Christopher G.; VRIES, Arjen P. de; SRINIVASAN, Padmini. Quality Through Flow and Immersion: Gamifying Crowdsourced Relevance Assessments. In: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Portland, Oregon, USA: ACM, 2012, s. 871–880. SIGIR '12. ISBN 978-1-4503-1472-5. Dostupné z DOI: 10.1145/2348283.2348400.
6. KOSINSKI, Michal; BACHRACH, Yoram; KASNECI, Gjergji; VAN-GAEL, Jurgen; GRAEPEL, Thore. Crowd IQ: Measuring the Intelligence of Crowdsourcing Platforms. In: *Proceedings of the 4th Annual ACM Web Science Conference*. Evanston, Illinois: ACM, 2012, s. 151–160. WebSci '12. ISBN 978-1-4503-1228-8. Dostupné z DOI: 10.1145/2380718.2380739.
7. EICKHOFF, Carsten; VRIES, Arjen de. How crowdsourcable is your task. In: *Proceedings of the workshop on crowdsourcing for search and data mining (CSDM) at the fourth ACM international conference on web search and data mining (WSDM)*. 2011, s. 11–14.

8. QUINN, Alexander J.; BEDERSON, Benjamin B. Human Computation: A Survey and Taxonomy of a Growing Field. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Vancouver, BC, Canada: ACM, 2011, s. 1403–1412. CHI '11. ISBN 978-1-4503-0228-9. Dostupné z DOI: 10.1145/1978942.1979148.

9. YUEN, M. C.; KING, I.; LEUNG, K. S. A Survey of Crowdsourcing Systems. In: *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*. 2011, s. 766–773. Dostupné z DOI: 10.1109/PASSAT/SocialCom.2011.203.

10. IPEIROTIS, Panagiotis G.; PROVOST, Foster; WANG, Jing. Quality Management on Amazon Mechanical Turk. In: *Proceedings of the ACM SIGKDD Workshop on Human Computation*. Washington DC: ACM, 2010, s. 64–67. HCOMP '10. ISBN 978-1-4503-0222-7. Dostupné z DOI: 10.1145/1837885.1837906.

11. DAWID, A. P.; SKENE, A. M. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*. 1979, roč. 28, č. 1, s. 20–28.

12. BUDÍKOVÁ, Petra; BATKO, Michal; BOTOREK, Jan; ZEZULA, Pavel. Search-based image annotation: Extracting semantics from similar images. In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 6th International Conference of the CLEF Association, CLEF 2015*. Toulouse, France: Springer, 2015, s. 327–339. ISBN 978-3-319-24026-8. Dostupné z DOI: 10.1007/978-3-319-24027-5_36.

13. BUDÍKOVÁ, Petra; BATKO, Michal; ZEZULA, Pavel. Evaluation Platform for Content-based Image Retrieval Systems. In: *International Conference on Theory and Practice of Digital Libraries 2011, LNCS 6966*. Berlin: Springer, 2011, s. 130–142. ISBN 978-3-642-24468-1. Dostupné z DOI: 10.1007/978-3-642-24469-8_15.

A Přílohy

A.1 Případy užití

ID	UC_1
Název	Přihlášení do systému
Popis	Slouží k identifikaci uživatele v rámci systému.
Aktéři	Uživatel
Vstupní podmínky	Uživatel má aktivní účet v externím IS MUNI (Informační systém Masarykovy univerzity).
Základní scénář	1. Uživatel zvolí v menu „Přihlásit“. 2. Systém uživatele přesměruje do externího systému Jednotného přihlášení MU. 3. V rámci externího systému uživatel zadá své UČO (Univerzitní číslo osoby) a sekundární heslo. 4. Externí systém uživatele ověří a přesměruje zpět. 5. POKUD uživatel v systému zatím neexistuje: 1. Systém vytvoří nového uživatele s rolí pracovníka. 6. POKUD v systému neexistuje žádný administrátor: 1. Systém přiřadí uživateli všechny role. 7. Uživatel je informován o výsledku operace.
Výstupní podmínky	Uživatel je přihlášen a přesměrován do svého uživatelského účtu.
Alternativní scénář	V případě neplatných uživatelských údajů je zobrazena informace o neúspěšném pokusu a je možné jej opakovat.

Tabulka A.1: Případ užití UC_1

ID	UC_2
Název	Odhlášení ze systému
Popis	Slouží k odhlášení uživatele v rámci systému.
Aktéři	Uživatel
Vstupní podmínky	Uživatel je přihlášen v rámci systému.
Základní scénář	1. Uživatel zvolí v menu „Odhlásit“. 2. Systém uživatele přesměruje do externího systému Jednotného přihlášení MU. 3. Externí systém uživatele odhlásí a přesměruje zpět. 4. Uživatel je informován o výsledku operace.
Výstupní podmínky	Uživatel je odhlášen a přesměrován na úvodní stránku systému.
Alternativní scénář	–

Tabulka A.2: Případ užití UC_2

ID	UC_3
Název	Upravení rolí uživatele
Popis	Slouží k přiřazení nebo odebrání rolí vybranému uživateli.
Aktéři	Administrátor
Vstupní podmínky	Uživatel je přihlášen v rámci systému a má roli administrátora.
Základní scénář	1. Uživatel zvolí v menu „Uživatelé“. 2. Vyhledá uživatele podle jména nebo e-mailové adresy. 3. Uživatel zvolí volbu „Upravit“ u vybraného uživatele. 4. Zvolí jednu či více rolí v uživatelském formuláři. 5. Odesláním formuláře potvrdí upravení rolí. 6. Systém uživateli nastaví vybrané role. 7. Uživatel je informován o výsledku operace.
Výstupní podmínky	Zvolený uživatel získal vybrané role.
Alternativní scénář	V případě že uživatel v systému ještě neexistuje, je možné jej nejdříve vytvořit použitím volby „Vytvořit uživatele“.

Tabulka A.3: Případ užití UC_3

ID	UC_4
Název	Vytvoření projektu
Popis	Slouží k přidání nového projektu do systému.
Aktéři	Zadavatel
Vstupní podmínky	Uživatel je přihlášen v rámci systému a má roli zadavatele.
Základní scénář	1. Uživatel zvolí v menu „Projekty“. 2. Zvolí volbu „Vytvořit projekt“. 3. Vyplní požadované položky formuláře: 1. Typ projektu (obecný, moderace, A/B testování). 2. Název, popis, klíčová slova pro vyhledávání. 3. Datum začátku a konce možnosti řešení. 4. Odměna za jeden úkol a limit pro celkovou odměnu. 5. Použitý kontrolní algoritmus (většinový názor, případně vážený většinový názor s využitím kontrolních otázek) a míra redundance (počet pracovníků na 1 totožný úkol). 6. Orientační čas trvání jednoho úkolu. 7. Instrukce k vypracování úkolů. 4. Odesláním formuláře potvrdí vytvoření projektu. 5. Uživatel je informován o výsledku operace.
Výstupní podmínky	Nový projekt je vytvořen v systému a uživatel je přesměrován na stránku s detailem projektu.
Alternativní scénář	V případě neúplně vyplněného formuláře je zobrazena informace o neúspěšném pokusu a je možné jej opakovat.

Tabulka A.4: Případ užití UC_4

ID	UC_5
Název	Vyhledání projektu
Popis	Slouží k vyhledání projektu dle zadaných parametrů.
Aktéři	Zadavatel, Pracovník
Vstupní podmínky	Uživatel je přihlášen v rámci systému a má roli zadavatele nebo pracovníka.

Základní scénář	1. Uživatel zvolí v menu „Projekty“.
	2. POKUD má uživatel roli zadavatele:
	  1. Systém zobrazí výpis projektů vytvořené tímto zadavatelem.
	3. POKUD má uživatel roli pracovníka:
	  1. Systém zobrazí výpis všech projektů, které tento pracovník může řešit, případně je v minulosti řešil.
	  2. U každého projektu systém vypíše informaci o počtu bodů získaných pracovníkem a také o maximálním počtu bodů, které lze získat („získáno X bodů z Y možných“). V případě vyřazeného pracovníka systém vypíše „vyřazen z projektu“.
	4. POKUD má uživatel roli administrátora:
	  1. Systém zobrazí výpis všech existujících projektů.
	5. Systém poskytne možnost vyhledávání dle různých parametrů (název, popis, klíčová slova).
	6. POKUD uživatel použije vyhledávání:
	  1. Systém zobrazí výpis jen vyhovujících projektů.
	7. Uživatel zvolí některý z dostupných projektů.

Výstupní podmínky	Uživatel je přesměrován na stránku s detailem projektu.
Alternativní scénář	–

Tabulka A.5: Případ užití UC_5

ID	UC_6
Název	Zpracovávání úkolů projektu
Popis	Slouží k práci na jednotlivých úkolech konkrétního projektu.
Aktéři	Pracovník
Vstupní podmínky	Uživatel je přihlášen v rámci systému a má roli pracovníka.
Základní scénář	1. ZAHRNUJE UC_5 2. POKUD je projekt aktivní, obsahuje úkoly k řešení a není omezen pro konkrétní pracovníky, případně omezen je a uživatel se mezi nimi vyskytuje, a zároveň uživatel není z projektu vyřazen: 1. Uživatel zvolí volbu „Začít pracovat na projektu“. 2. DOKUD uživatel nezvolí „Ukončit práci na projektu“, případně dokud jsou dostupné neřešené úkoly: 1. POKUD v projektu existují kontrolní úkoly, které uživatel ještě nevyřešil: 1. Systém zobrazí náhodný neřešený kontrolní úkol. 2. JINAK: 1. Systém zobrazí náhodný úkol s nejmenším počtem již odeslaných řešení. 3. Uživatel odpoví na všechny položené otázky. 4. Odesláním formuláře potvrdí vyřešení úkolu.
Výstupní podmínky	Systém obsahuje odeslané odpovědi na položené otázky a uživatel je přesměrován na stránku s detailem projektu.
Alternativní scénář	V případě neúplně vyplněného formuláře je zobrazena informace o neúspěšném pokusu a je možné jej opakovat.

Tabulka A.6: Případ užití UC_6

ID	UC_7
Název	Upravení projektu
Popis	Slouží k upravení existujícího projektu v rámci systému.
Aktéři	Zadavatel
Vstupní podmínky	Uživatel je přihlášen v rámci systému a má roli zadavatele.
Základní scénář	1. ZAHRNUJE UC_5 2. Uživatel zvolí volbu „Upravit projekt“. 3. Upraví požadované položky formuláře. 4. POKUD je nutné projekt ukončit: 1. Uživatel upraví datum konce možnosti řešení, případně označí projekt za neaktivní. 5. Odesláním formuláře potvrdí upravení projektu. 6. Uživatel je informován o výsledku operace.
Výstupní podmínky	Existující projekt je upraven v systému a uživatel je přesměrován na stránku s detailem projektu.
Alternativní scénář	V případě neúplně vyplněného formuláře je zobrazena informace o neúspěšném pokusu a je možné jej opakovat.

Tabulka A.7: Případ užití UC_7

ID	UC_8
Název	Smazání projektu
Popis	Slouží ke smazání existujícího projektu v rámci systému.
Aktéři	Zadavatel
Vstupní podmínky	Uživatel je přihlášen v rámci systému a má roli zadavatele.
Základní scénář	1. ZAHRNUJE UC_5 2. Uživatel zvolí volbu „Smazat projekt“. 3. Systém zobrazí varování o nevratné operaci. 4. Uživatel odesláním formuláře potvrdí smazání projektu. 5. Systém smaže projekt a také všechny jeho úkoly. 6. Uživatel je informován o výsledku operace.
Výstupní podmínky	Existující projekt je smazán v systému a uživatel je přesměrován na přehled všech projektů.
Alternativní scénář	–

Tabulka A.8: Případ užití UC_8

ID	UC_9
Název	Import pracovníků projektu
Popis	Slouží k omezení účasti řešitelů konkrétního projektu.
Aktéři	Zadavatel
Vstupní podmínky	Uživatel je přihlášen v rámci systému a má roli zadavatele.
Základní scénář	1. ZAHRNUJE UC_5 2. Uživatel zvolí volbu „Importovat studenty“. 3. Vyplní požadované položky formuláře: 1. Kód předmětu, pro který pracovníky importuje. 2. Seznam pracovníků ve formátu TSV exportovaný z aplikace poznámkového bloku v IS MUNI. 4. Odesláním formuláře potvrdí import pracovníků. 5. Uživatel je informován o výsledku operace.
Výstupní podmínky	Povolení pracovníci jsou přiřazení k projektu a uživatel je přesměrován na stránku s detailem projektu.
Alternativní scénář	V případě neúplně vyplněného formuláře je zobrazena informace o neúspěšném pokusu a je možné jej opakovat.

Tabulka A.9: Případ užití UC_9

ID	UC_10
Název	Vytvoření úkolů projektu
Popis	Slouží k manuálnímu vytvoření sady úkolů v rámci projektu.
Aktéři	Zadavatel
Vstupní podmínky	Uživatel je přihlášen v rámci systému a má roli zadavatele.
Základní scénář	1. ZAHRNUJE UC_5 2. Uživatel zvolí volbu „Vytvořit úkoly“. 3. DOKUD uživatel nezvolí „Ukončit vytváření úkolů“: 1. Vyplní požadované položky formuláře: 1. Soubor z disku, pokud jsou pro úkol vyžadován. 2. Volitelný textový popis úkolu. 3. Neomezené množství otázek k tomuto úkolu (pro každou otázku uživatel vyplní také její typ, četnost odpovědí a dostupné možnosti, případně také vybere soubor z disku). 2. Odesláním formuláře potvrdí vytvoření úkolu. 4. POKUD projekt vyžaduje využití globálních otázek: 1. Uživatel zvolí volbu „Vytvořit globální otázky“. 2. Vyplní požadované položky formuláře: 1. Neomezené množství otázek k tomuto projektu (pro každou otázku uživatel vyplní také její typ, četnost odpovědí a dostupné možnosti, případně také vybere soubor z disku). 3. Odesláním formuláře potvrdí vytvoření otázek.
Výstupní podmínky	Nové úkoly jsou vytvořeny v systému a uživatel je přesměrován na stránku s detailem projektu.
Alternativní scénář	V případě neúplně vyplněného formuláře je zobrazena informace o neúspěšném pokusu a je možné jej opakovat.

Tabulka A.10: Případ užití UC_10

ID	UC_11
Název	Import úkolů projektu
Popis	Slouží k hromadnému vytvoření sady úkolů v rámci projektu.
Aktéři	Zadavatel
Vstupní podmínky	Uživatel je přihlášen v rámci systému a má roli zadavatele.
Základní scénář	1. ZAHRNUJE UC_5 2. Uživatel zvolí volbu „Importovat úkoly“. 3. Vyplní požadované položky formuláře: 1. Seznam úkolů ve formátu JSON. 2. Soubory z disku, pokud jsou pro úkoly vyžadovány. 4. Odesláním formuláře potvrdí import úkolů. 5. Uživatel je informován o výsledku operace.
Výstupní podmínky	Nové úkoly a globální otázky jsou vytvořeny v systému a uživatel je přesměrován na stránku s detailem projektu.
Alternativní scénář	V případě neúplně vyplněného formuláře je zobrazena informace o neúspěšném pokusu a je možné jej opakovat.

Tabulka A.11: Případ užití UC_11

ID	UC_12
Název	Vypracování kontrolních úkolů
Popis	Slouží ke vložení kontrolních odpovědí pro některé úkoly.
Aktéři	Zadavatel
Vstupní podmínky	Uživatel je přihlášen v rámci systému a má roli zadavatele.
Základní scénář	1. ZAHRNUJE UC_5 2. POKUD kontrolní algoritmus projektu povoluje využití kontrolních úkolů: 1. Uživatel zvolí volbu „Řešit kontrolní úkoly“. 2. DOKUD uživatel nezvolí „Ukončit řešení kontrolních otázek“, případně dokud jsou dostupné úkoly: 1. Systém zobrazí náhodný neřešený úkol. 2. Uživatel odpoví na všechny položené otázky. 3. Odesláním formuláře potvrdí vyřešení úkolu.
Výstupní podmínky	Systém obsahuje odeslané odpovědi na kontrolní otázky a uživatel je přesměrován na stránku s detailem projektu.
Alternativní scénář	V případě neúplně vyplněného formuláře je zobrazena informace o neúspěšném pokusu a je možné jej opakovat.

Tabulka A.12: Případ užití UC_12

ID	UC_13
Název	Zobrazení aktuálních výsledků
Popis	Slouží k prohlížení aktuálních výsledků řešených úkolů.
Aktéři	Zadavatel
Vstupní podmínky	Uživatel je přihlášen v rámci systému a má roli zadavatele.
Základní scénář	1. ZAHRNUJE UC_5 2. Uživatel zvolí volbu „Zobrazit výsledky“. 3. Systém vypočítá výsledky těch úkolů, u kterých počet řešitelů úkolu dosáhla redundance požadované v nastavení projektu. Zároveň systém aktualizuje skóre zúčastněným pracovníkům na základě odeslaných odpovědí. 4. Systém zobrazí výpis všech úkolů projektu seřazených od nejvyššího množství odpovědí po nejmenší. 5. Uživatel zvolí některý z úkolů. 6. Systém zobrazí detailní stránku úkolu s aktuálními výsledky odeslaných odpovědí.
Výstupní podmínky	–
Alternativní scénář	–

Tabulka A.13: Případ užití UC_13

ID	UC_14
Název	Zobrazení skóre a odpovědí pracovníků
Popis	Slouží k prohlížení aktuálního skóre pracovníků a jejich vypracovaných úkolů v rámci projektu.
Aktéři	Zadavatel
Vstupní podmínky	Uživatel je přihlášen v rámci systému a má roli zadavatele.
Základní scénář	1. ZAHRNUJE UC_5 2. Uživatel zvolí volbu „Zobrazit pracovníky“. 3. Systém vypočítá výsledky těch úkolů, u kterých počet řešitelů úkolu dosáhla redundance požadované v nastavení projektu. Zároveň systém aktualizuje skóre zúčastněným pracovníkům na základě odeslaných odpovědí. 4. Systém zobrazí výpis všech pracovníků a jejich aktuálních hodnocení seřazených od nejnižšího skóre po nejvyšší. 5. Uživatel může prohlížet výsledky odeslané konkrétním pracovníkem v rámci projektu.
Výstupní podmínky	–
Alternativní scénář	–

Tabulka A.14: Případ užití UC_14

ID	UC_15
Název	Vyřazení pracovníka z projektu
Popis	Slouží k vyřazení pracovníků s nevyhovující kvalitou odpovědí.
Aktéři	Zadavatel
Vstupní podmínky	Uživatel je přihlášen v rámci systému a má roli zadavatele.
Základní scénář	1. ZAHRNUJE UC_14 2. Uživatel zvolí volbu „Vyřadit pracovníka z projektu“ u konkrétního pracovníka, který má příliš nízké skóre, případně jsou jeho odpovědi nevyhovující. 3. Systém zobrazí varování o prováděné operaci. 4. Uživatel odesláním formuláře potvrdí vyřazení pracovníka. 5. Systém označí odpovědi daného pracovníka v rámci projektu jako vyřazené. 6. Uživatel je informován o výsledku operace.
Výstupní podmínky	Odpovědi pracovníka jsou vyřazeny z projektu a uživatel je přesměrován na stránku s výpisem pracovníků.
Alternativní scénář	–

Tabulka A.15: Případ užití UC_15

ID	UC_16
Název	Export výsledků projektu
Popis	Slouží k exportu výsledku řešených úkolů pro konkrétní projekt.
Aktéři	Zadavatel
Vstupní podmínky	Uživatel je přihlášen v rámci systému a má roli zadavatele.
Základní scénář	1. ZAHRNUJE UC_5 2. Uživatel zvolí volbu „Exportovat výsledky“. 3. Systém vypočítá výsledky těch úkolů, u kterých počet řešitelů úkolu dosáhla redundance požadované v nastavení projektu. Zároveň systém aktualizuje skóre zúčastněným pracovníkům na základě odeslaných odpovědí. 4. Systém zobrazí seznam úkolů a výsledných odpovědí ve formátu JSON. 5. Uživatel je informován o výsledku operace.
Výstupní podmínky	–
Alternativní scénář	–

Tabulka A.16: Případ užití UC_16

ID	UC_17
Název	Export bodů do poznámkových bloků
Popis	Slouží k exportu bodů do poznámkových bloků pro řešitele projektu.
Aktéři	Zadavatel
Vstupní podmínky	Uživatel je přihlášen v rámci systému a má roli zadavatele.
Základní scénář	1. ZAHRNUJE UC_5 2. Uživatel zvolí volbu „Exportovat body“. 3. POKUD je projekt omezený pro konkrétní předměty a pracovníky: 1. PRO KAŽDÝ předmět: 1. Systém zobrazí seznam pracovníků přiřazených k danému předmětu a získaných bodů ve formátu TSV, který je importovatelný do aplikace poznámkového bloku v IS MUNI. 4. JINAK: 1. Systém zobrazí seznam pracovníků a získaných bodů ve formátu TSV, který je importovatelný do aplikace poznámkového bloku v IS MUNI. 5. Uživatel je informován o výsledku operace.
Výstupní podmínky	–
Alternativní scénář	–

Tabulka A.17: Případ užití UC_17

A.2 Obsah přílohy na CD

Součástí práce jsou následující elektronické přílohy:

- Elektronická verze práce v přenosném formátu PDF.
  - soubor „thesis.pdf“
- Zdrojový soubor s celou prací ve formátu TEX.
  - adresář „tex“
- Zdrojový kód webové aplikace ve formě Java tříd.
  - adresář „app“
- Kompilovaná verze aplikace ve formátu WAR.
  - soubor „app.war“
- Zkušební sady úkolů ve formátu JSON.
  - soubor „tasks-general.txt“
  - soubor „tasks-moderation.txt“
  - soubor „tasks-testing.txt“
- Zkušební sada médií ve formátech JPG, MP3 a MP4.
  - adresář „media“
- Zkušební export z poznámkového bloku ve formátu TSV.
  - soubor students.txt“

A.3 Postup nasazení

Požadavky pro spuštění aplikace:

- Zařízení je připojeno k internetové síti.
- Instalovaný balíček Java Development Kit (JDK) ve verzi 8.
  - http://www.oracle.com/technetwork/java/javase/downloads/index.html
  - Ověřit lze příkazem „java -version“.
- Instalovaný nástroj Apache Maven ve verzi 3.
  - https://maven.apache.org/download.cgi
  - Ověřit lze příkazem „mvn --version“.
- Instalovaný databázový systém PostgreSQL ve verzi 9.
  - https://www.postgresql.org/download/
  - Ověřit lze příkazem „psql --version“.
- Vytvořená prázdná PostgreSQL databáze s názvem „muc“.
  - Uživatel i heslo nastaveny shodně na „postgres“.

Postup pro spuštění webové aplikace:

1. Z přiloženého CD zkopírovat adresář „app“ na pevný disk.
2. V tomto adresáři postupně vykonáme následující příkazy:
   - „mvn clean“
   - „mvn install“
3. Přejít do adresáře „app/muc-mvc“ a spustit příkaz:
   - „mvn tomcat7:run“
4. Aplikace je dostupná na adrese:
   - http://localhost:8080/muc/
5. Pokud je port 8080 již obsazený, lze jej změnit editací souboru „app/muc-mvc/pom.xml“ na řádku č. 88.

Postup pro testování aplikace s exportem do poznámkového bloku:

1. V aplikaci kliknout na odkaz „Přihlásit“ – dojde k přesměrování na vývojovou verzi systému jednotné autentizace MU.
2. Přihlásit se pomocí vašeho UČO a hesla – dojde k přesměrování zpět do aplikace s předáním základních informací o uživateli.
    - Pokud v systému neexistuje žádný administrátor, první přihlášený uživatel dostane tuto roli automaticky.
3. Vytvořit projekt: „Projekty“ -> „Vytvořit projekt“.
    - Zadat následující hodnoty: typ „Obecný“, název „Kategorizace médií“, kontrola „Výpočet skóre z kontrolních otázek“.
    - Kliknout na „Vytvořit projekt“.
4. Po vytvoření a přesměrování na stránku projektu kliknout na „Importovat úkoly“.
5. Vložit JSON data ze souboru „tasks-general.txt“, zvolit všechny soubory ze složky „media“ a kliknout na „Importovat úkoly“.
6. Vyplněním celého formuláře jednou či vícekrát zpracovat několik úkolů.
7. Kliknout na „Zpět k projektu“ a poté na „Generovat body“.
8. Systém vypíše tabulátorem oddělený text vhodný k přímému vložení do aplikace poznámkového bloku v IS MUNI.
    - Dle množství zpracovaných úkolů je na konci výpisu viditelná informace o počtu splněných úkolů a udělených bodech.

A.4 Struktura dat ve formátu JSON

Pro import a export úkolů projektu je použita shodná struktura dat. Definovaná je třídou „ProjectJson“ v API vrstvě aplikace. Základní JSON objekt nese pouze 2 klíče: „global“ a „tasks“. Pod prvním z nich se nachází seznam globálních otázek (objekty typu „QuestionJson“), druhý obsahuje seznam úkolů (objekty typu „TaskJson“). Ve třídě úkolu se nachází klíče „file“ (název importovaného souboru), „description“ (textový popis úkolu) a „questions“ (seznam znovu použitého objektu typu „QuestionJson“, který v tomto případě obsahuje otázky specifické pro daný úkol).

Objekt otázky obsahuje klíče „type“ (číselník „QuestionType“), „cardinality“ (počet možností, které lze v daném úkolu zvolit jako správné), „title“ (nadpis úkolu), „file“ (název importovaného souboru) a „data“ (pole s textovými hodnotami – každá reprezentuje jednu možnou volbu uživatele). Příkladem použití jsou 3 zkušební sady úkolů v příloze na CD.

```
public class ProjectJson {
  private List<QuestionJson> global;
  private List<TaskJson> tasks;
}
public class TaskJson {
  private String file;
  private String description;
  private List<QuestionJson> questions;
}
public class QuestionJson {
  private String type;
  private Integer cardinality;
  private String title;
  private String file;
  private String[] data;
}
```

A.5 Výsledky experimentů

Jméno pracovníka	Miluška	Jiří	Lenča	Michal	Marťa				
Počáteční skóre	100	100	100	100	100				

Slovo 1	Obrázek 1	Obrázek 2	Obrázek 3	Obrázek 4	Obrázek 5	Obrázek 6	Obrázek 7	Obrázek 8	Obrázek 9	Obrázek 10
Miluška	0.9	0	0.7	0	0.9	0	0.8	0.1	0.7	0.1
Jiří	0.7	0.2	1	0.2	1	0.2	0.3	0	1	0.2
Lenča	0.9	0.1	0.5	0	0.6	0.9	0.3	0.8	0.8	0.8
Michal	1	0.1	0.5	0.4	0.4	0	0.8	0.2	0.5	0
Marťa	0.9	0	0.7	0.4	0.8	0	0.5	0	0.5	0
Průměr	**0.88**	**0.08**	**0.68**	**0.2**	**0.74**	**0.22**	**0.54**	**0.22**	**0.7**	**0.22**
Správně	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**

Slovo 2	Obrázek 1	Obrázek 2	Obrázek 3	Obrázek 4	Obrázek 5	Obrázek 6	Obrázek 7	Obrázek 8	Obrázek 9	Obrázek 10
Miluška	0.8	0.1	0.1	0.1	0.8	0.2	0.7	0.2	0.6	0.3
Jiří	0.5	0	0.5	0.2	0.7	0.3	1	0	0.7	0.3
Lenča	0.2	0.1	0.9	0.6	0.8	0	0.2	0.5	0.6	0.6
Michal	0.8	0	0.8	0.3	0.5	0.3	0.6	0.2	1	0.3
Marťa	0.7	0	0.6	0.4	1	0	0.8	0.4	0.2	0.2
Průměr	**0.6**	**0.04**	**0.58**	**0.32**	**0.76**	**0.16**	**0.66**	**0.26**	**0.62**	**0.34**
Správně	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**

Slovo 3	Obrázek 1	Obrázek 2	Obrázek 3	Obrázek 4	Obrázek 5	Obrázek 6	Obrázek 7	Obrázek 8	Obrázek 9	Obrázek 10
Miluška	0.7	0.4	0.7	0.4	0.7	0.4	0.6	0.2	0.6	0.1
Jiří	1	0.3	0.6	0.1	0.8	0	0.5	0.6	1	0.2
Lenča	0.3	0	0.1	0.5	0.9	0.6	1	0.9	0.6	0.8
Michal	0.7	0	0.5	0.2	0.8	0.2	0.9	0.2	0.9	0.4
Marťa	0.8	0.3	0.5	0.7	0.5	0	0.7	0.2	0.8	0
Průměr	**0.7**	**0.2**	**0.48**	**0.38**	**0.74**	**0.24**	**0.74**	**0.42**	**0.78**	**0.3**
Správně	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**0**

Slovo 4	Obrázek 1	Obrázek 2	Obrázek 3	Obrázek 4	Obrázek 5	Obrázek 6	Obrázek 7	Obrázek 8	Obrázek 9	Obrázek 10
Miluška	0.6	0	0.8	0.1	0.7	0.2	0.5	0.1	0.9	0
Jiří	0.7	0.4	0.5	0.1	0.8	0.2	0.5	0.3	1	0.4
Lenča	0.7	0.2	0.5	0.4	0.8	0.9	0.9	0.1	0	0.4
Michal	0.9	0.4	0.2	0	1	0.3	0.5	0	0.8	0.4
Marťa	0.8	0.1	0.5	0.2	0.7	0.4	0.8	0.1	0.7	0.3
Průměr	**0.74**	**0.22**	**0.5**	**0.16**	**0.8**	**0.4**	**0.64**	**0.12**	**0.68**	**0.3**
Správně	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**

Slovo 5	Obrázek 1	Obrázek 2	Obrázek 3	Obrázek 4	Obrázek 5	Obrázek 6	Obrázek 7	Obrázek 8	Obrázek 9	Obrázek 10
Miluška	0.8	0	0.9	0.1	0.6	0.4	0.5	0	0.6	0
Jiří	0.9	0	0.6	0.2	0.9	0.3	0.9	0	0.7	0.3
Lenča	1	0.8	1	0.9	0.9	0	0.9	0.2	0.4	0
Michal	0.8	0.4	1	0.4	0.7	0.3	0.3	0.2	0.9	0.4
Marťa	0.9	0.4	0.5	0.2	0.9	0.2	0.9	0	0.7	0.4
Průměr	**0.88**	**0.32**	**0.8**	**0.36**	**0.8**	**0.24**	**0.7**	**0.08**	**0.66**	**0.22**
Správně	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**

Jméno pracovníka	Miluška	Jiří	Lenča	Michal	Marťa				
Výsledné skóre	96	94	62	92	94				

Obrázek A.1: Experiment s jedním nepoctivým pracovníkem.

Jméno pracovníka	Miluška	Jiří	Lenča	Michal	Marťa				
Počáteční skóre	100	100	100	100	100				

Slovo 1	Obrázek 1	Obrázek 2	Obrázek 3	Obrázek 4	Obrázek 5	Obrázek 6	Obrázek 7	Obrázek 8	Obrázek 9	Obrázek 10
Miluška	0.7	0.2	0.4	0.4	0.7	0	0.8	0	0.5	0.2
Jiří	0.7	0	0.7	0.2	0.6	0.3	0.7	0.2	0.8	0.3
Lenča	0.8	0	0	1	0.2	0.5	0.1	0.5	0.2	0.6
Michal	0.5	0.2	0.7	0.4	1	0.4	0.7	0	0.6	0.1
Marťa	0.8	0.1	0.8	0.2	0	0	0.3	0.9	0.3	0.6
Průměr	**0.7**	**0.1**	**0.52**	**0.44**	**0.5**	**0.24**	**0.52**	**0.32**	**0.48**	**0.36**
Správně	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**0**

Slovo 2	Obrázek 1	Obrázek 2	Obrázek 3	Obrázek 4	Obrázek 5	Obrázek 6	Obrázek 7	Obrázek 8	Obrázek 9	Obrázek 10
Miluška	0.7	0.3	0.7	0.2	0.5	0.6	0.9	0.4	0.9	0.1
Jiří	0.6	0	0.6	0.4	0.8	0	0.8	0.3	0.9	0.3
Lenča	0.5	0	0.9	0.9	0.8	0.8	0.9	0.7	0.2	0.4
Michal	0.9	0.2	0.8	0.4	0.6	0.4	0.7	0.4	0.9	0
Marťa	1	0.6	0.1	0.5	0.3	0.2	0.1	0.9	0.6	0.1
Průměr	**0.74**	**0.22**	**0.62**	**0.48**	**0.6**	**0.4**	**0.68**	**0.54**	**0.7**	**0.18**
Správně	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**0**

Slovo 3	Obrázek 1	Obrázek 2	Obrázek 3	Obrázek 4	Obrázek 5	Obrázek 6	Obrázek 7	Obrázek 8	Obrázek 9	Obrázek 10
Miluška	0.9	0.1	1	0.4	0.7	0.2	1	0.4	0.6	0.1
Jiří	0.9	0.3	0.7	0.2	0.5	0.3	1	0.4	0.6	0
Lenča	0.2	0.5	0.6	0	0	0	0.6	0.8	0.7	0
Michal	0.5	0.2	1	0.4	1	0.2	0.4	0	1	0.4
Marťa	0	0.7	0.1	0	1	0.1	0.1	1	0.4	0.1
Průměr	**0.5**	**0.36**	**0.68**	**0.2**	**0.64**	**0.16**	**0.62**	**0.52**	**0.66**	**0.12**
Správně	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**0**

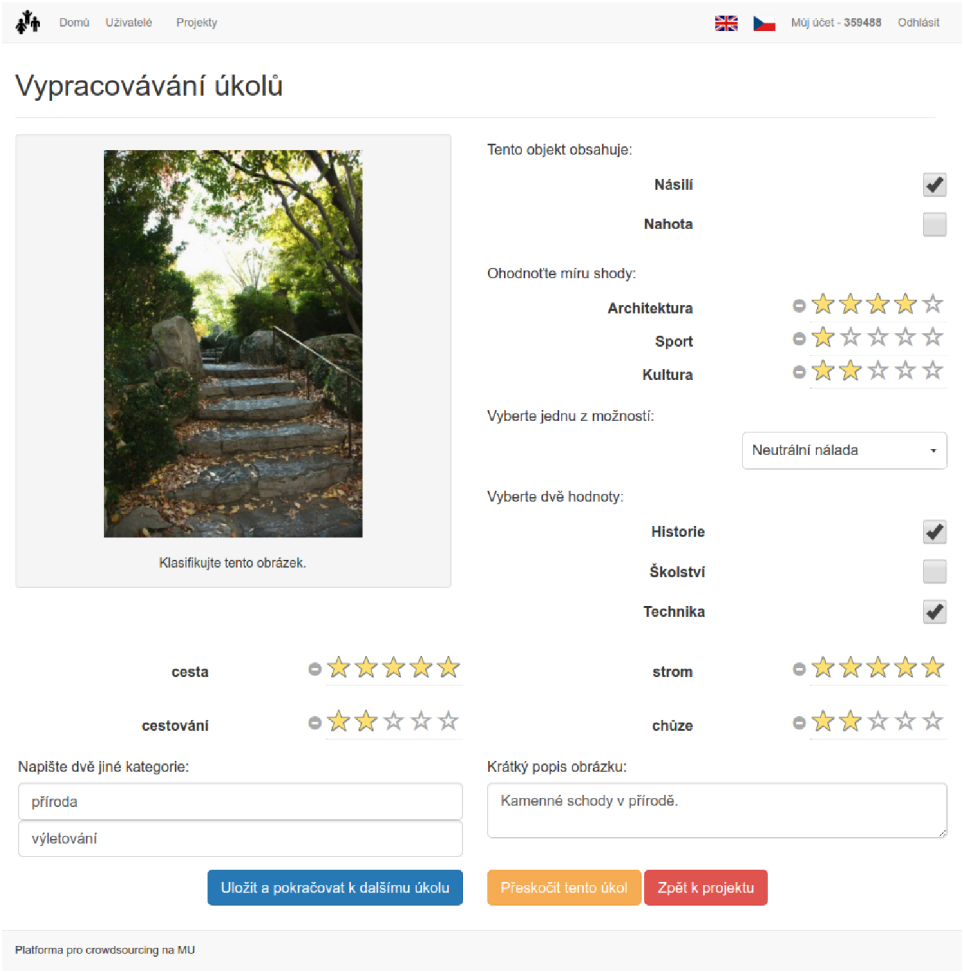
Slovo 4	Obrázek 1	Obrázek 2	Obrázek 3	Obrázek 4	Obrázek 5	Obrázek 6	Obrázek 7	Obrázek 8	Obrázek 9	Obrázek 10
Miluška	1	0.2	0.8	0	0.8	0.3	0.5	0.4	0.8	0.2
Jiří	0.9	0.4	0.9	0.2	1	0.2	0.9	0.1	0.3	0.4
Lenča	0.9	0.6	0.9	0	0.2	0.6	0.8	0.4	1	0.2
Michal	0.5	0.3	0.9	0	0.6	0.3	0.6	0.4	0.8	0.3
Marťa	0.9	0.2	0.8	0.7	0.5	1	0.6	0.5	0.7	0.5
Průměr	**0.84**	**0.34**	**0.86**	**0.18**	**0.62**	**0.48**	**0.68**	**0.36**	**0.72**	**0.32**
Správně	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**

Slovo 5	Obrázek 1	Obrázek 2	Obrázek 3	Obrázek 4	Obrázek 5	Obrázek 6	Obrázek 7	Obrázek 8	Obrázek 9	Obrázek 10
Miluška	0.8	0.2	1	0.1	0.6	0.2	0.6	0.2	0.7	0
Jiří	1	0.3	0.9	0.1	0.7	0.3	0.7	0.1	1	0
Lenča	0.3	0.6	0.3	0.2	0.2	0	0.6	0.5	0.9	0.5
Michal	0.5	0.3	0.5	0.1	0.5	1	0.7	0	0.6	0.3
Marťa	0.5	0.2	0	0	0.9	0.5	0.4	0.9	0.7	0.2
Průměr	**0.62**	**0.32**	**0.54**	**0.1**	**0.58**	**0.4**	**0.6**	**0.34**	**0.78**	**0.2**
Správně	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**

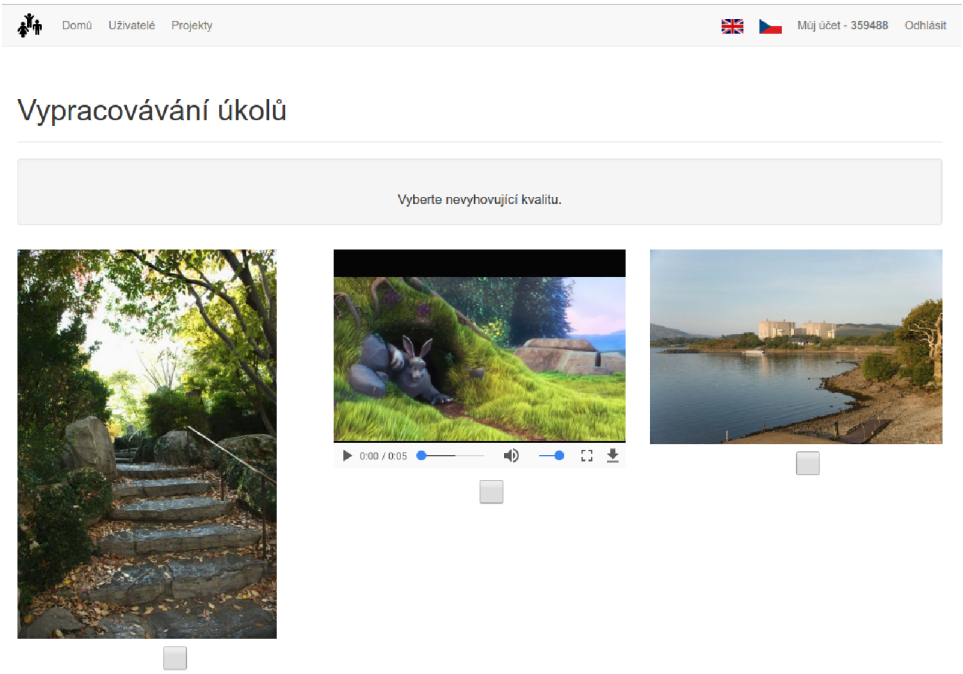
Jméno pracovníka	Miluška	Jiří	Lenča	Michal	Marťa				
Výsledné skóre	90	92	56	90	56				

Obrázek A.2: Experiment se dvěma nepoctivými pracovníky.

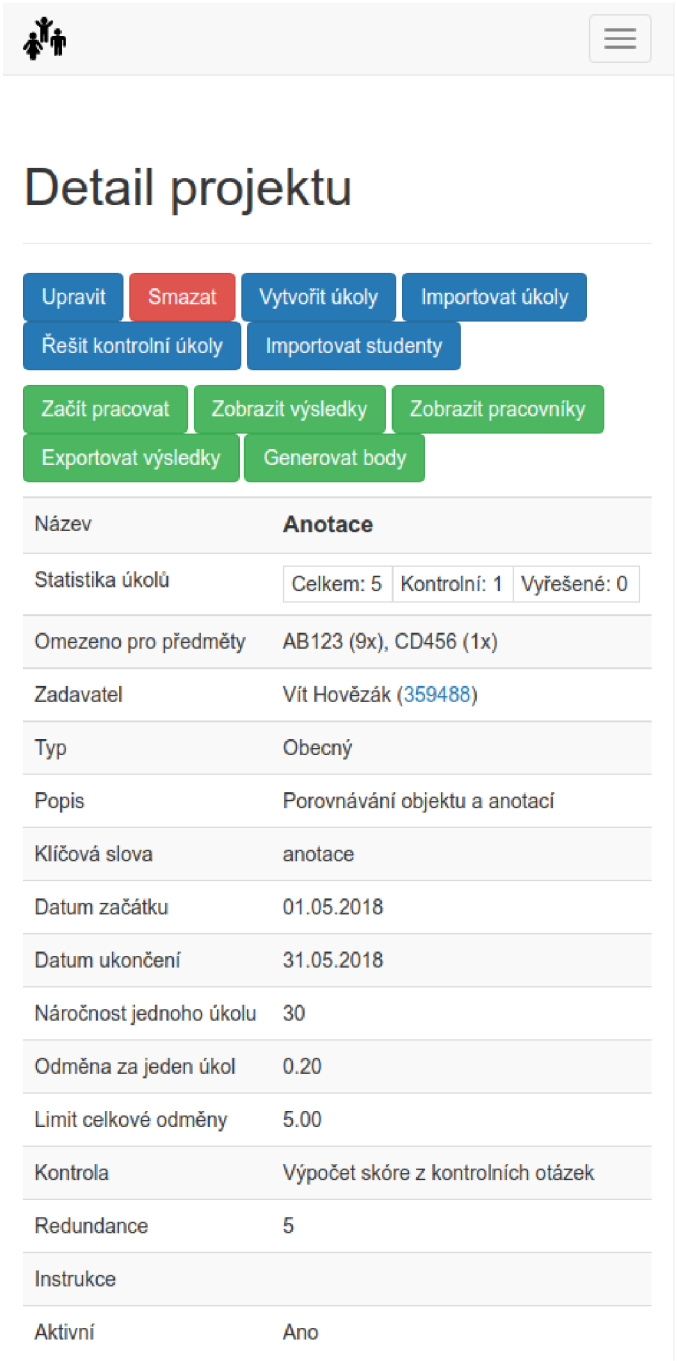
A.6 Uživatelské rozhraní



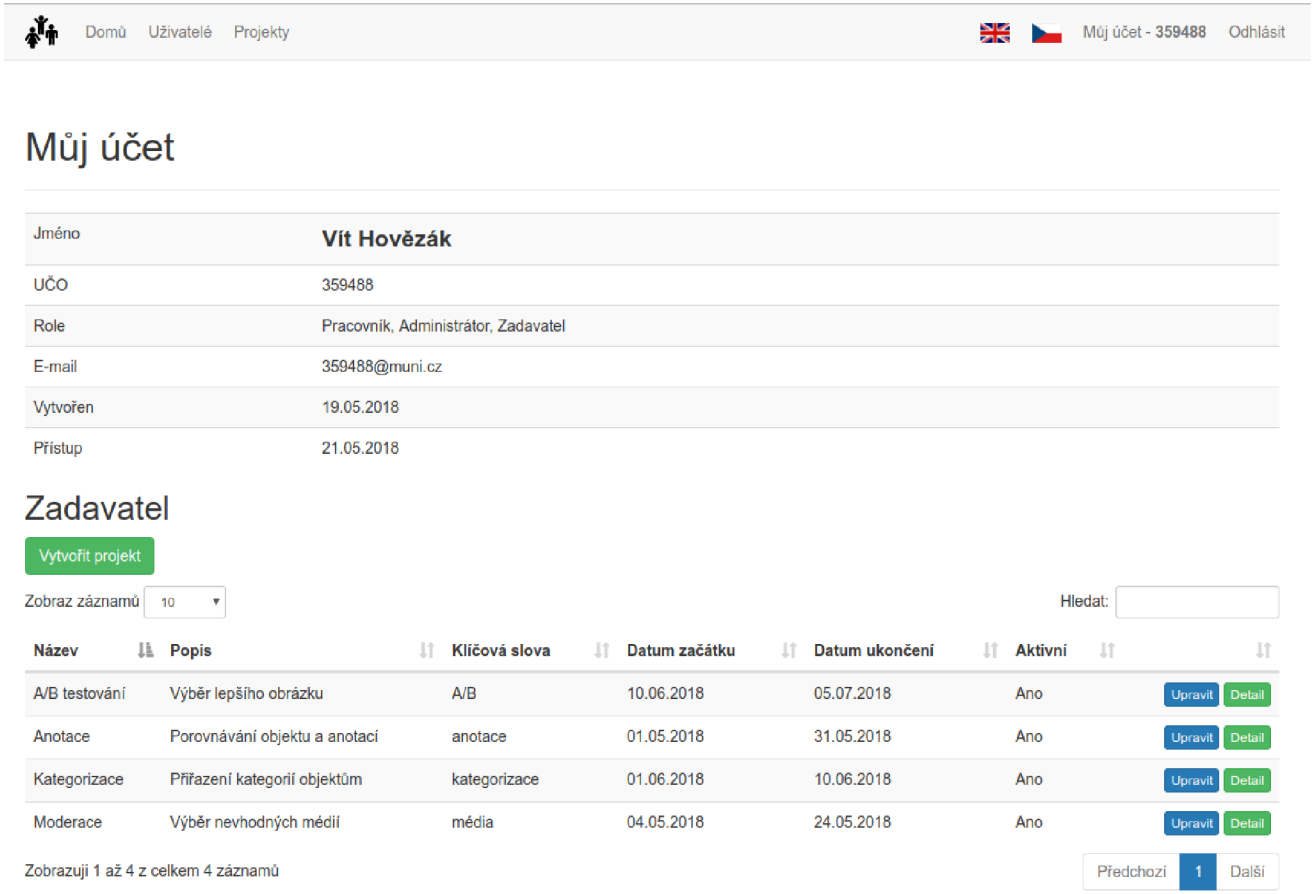
Obrázek A.3: Ukázka řešení úkolu v projektu



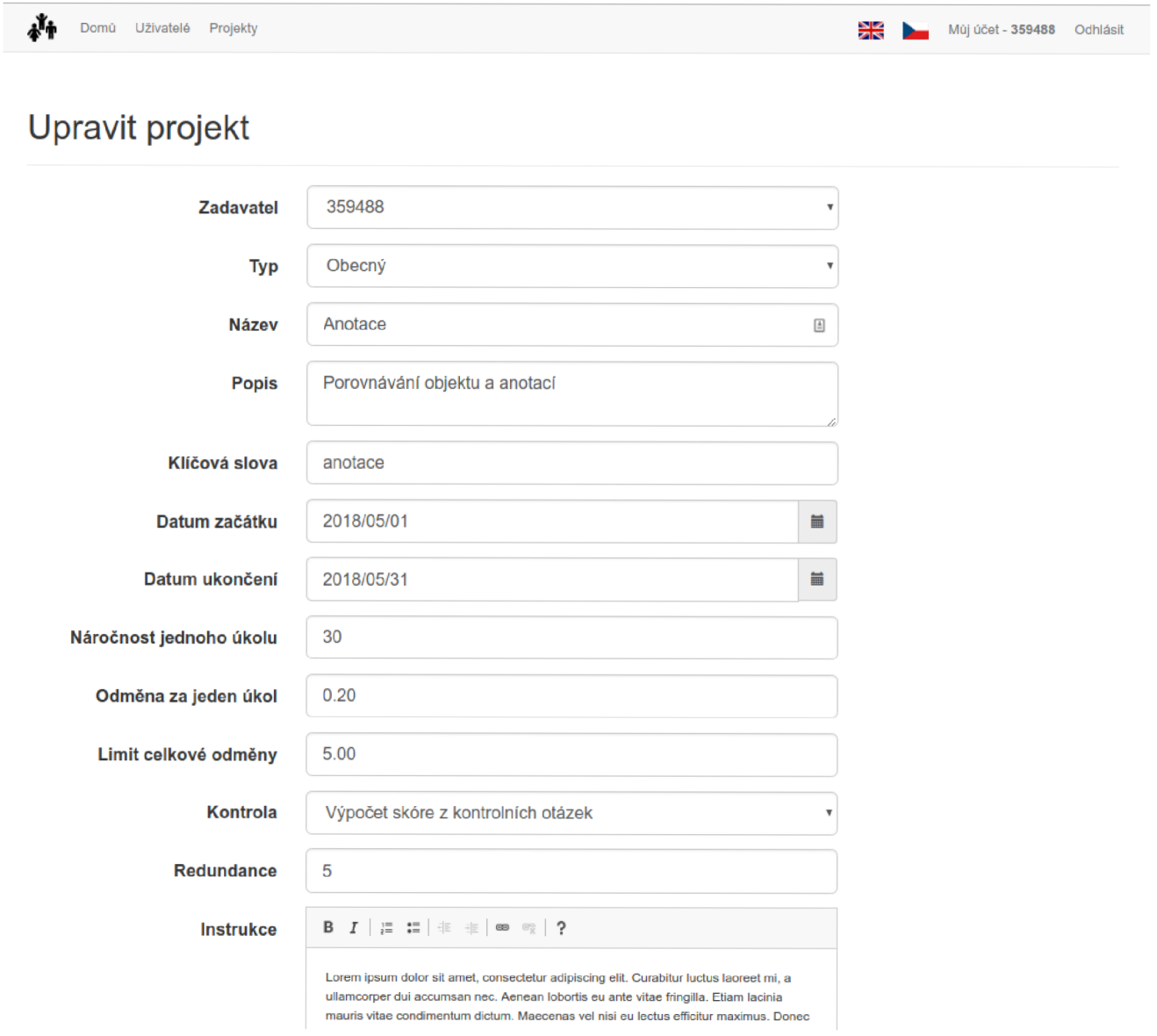
Obrázek A.4: Ukázka řešení projektu typu moderace



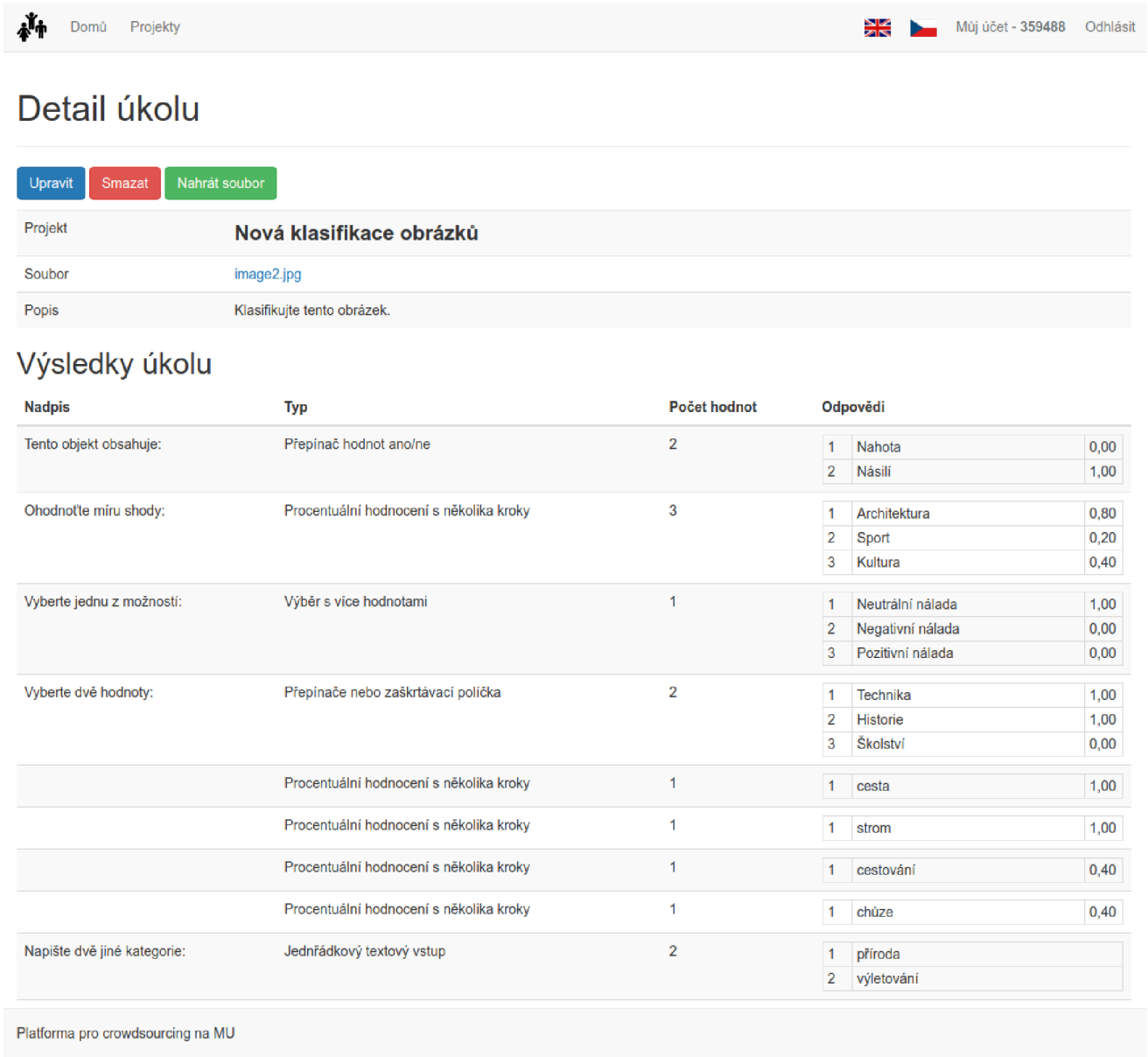
Obrázek A.5: Náhled projektu v mobilním rozložení



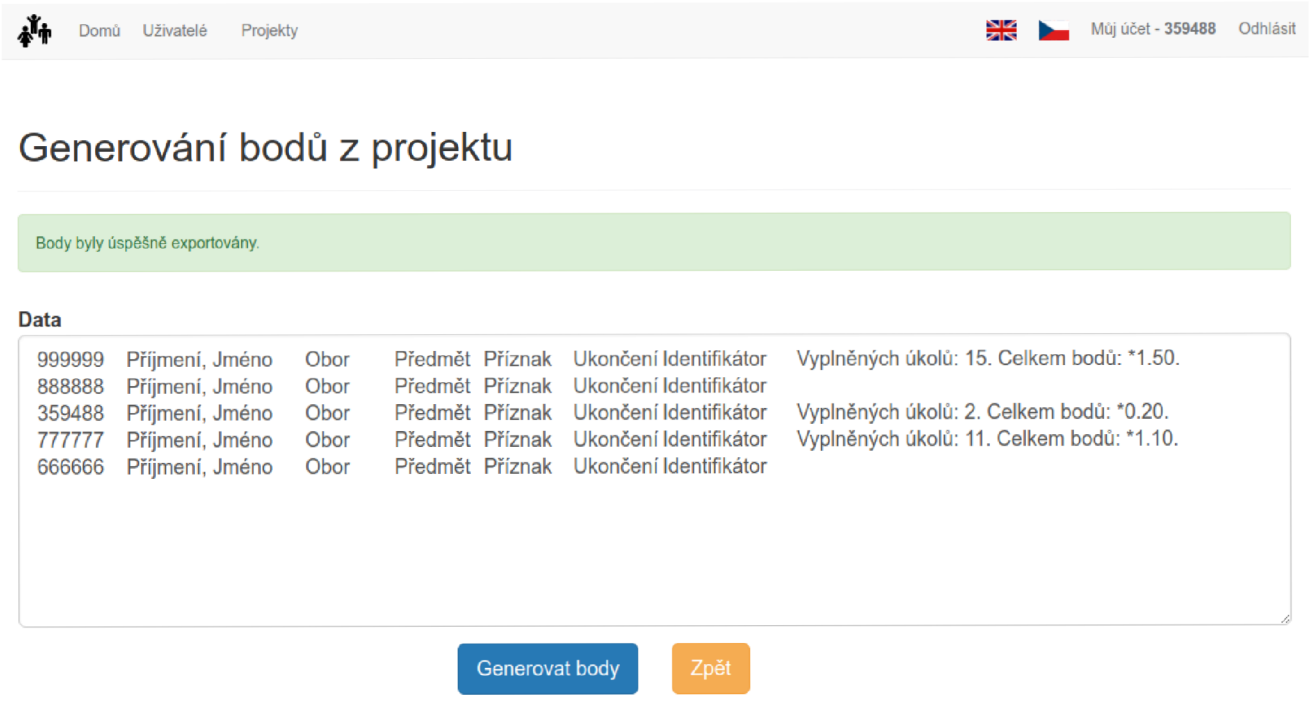
Obrázek A.6: Profil zadavatele s přehledem projektů



Obrázek A.7: Uživatelské prostředí editace projektu



Obrázek A.8: Uživatelské prostředí s výsledky úkolu



Obrázek A.9: Uživatelské prostředí s exportem bodů do poznámkového bloku