

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava 4

Pinta.sk

Digital Sweatshop

Dokumentácia k inžinierskemu dielu

Študijný program: Informačné systémy, Softvérové inžinierstvo

Ročník: 1.

Predmet: Tímový projekt I.

Vedúci projektu: Ing. Dušan Zeleník

Členovia tímu: Bc. Filip Bednárík

Bc. Róbert Černý

Bc. Marek Lenčes

Bc. Miroslav Molnár

Bc. Patrik Štrba

Bc. Martin Toma

Bc. Miroslav Vojtuš

Ak. rok: 2013/2014

Obsah

1	Úvod	4
2	Ciele projektu na zimný semester	5
3	Opisy príbehov	6
3.1	Pred prvým šprintom	6
3.1.1	Analýza existujúcich riešení	6
3.1.2	Výber a inštalácia podporných nástrojov	6
3.1.3	Vytvorenie webového sídla	8
3.1.4	Lean Canvas	9
3.2	Šprint #1	10
3.2.1	Vytvorenie základných rozhraní aplikácie	10
3.2.2	Rozbitie prvej User Story	10
3.2.3	Návrh rozhrania webovej aplikácie	11
3.2.4	Autorizácia a autentifikácia používateľa v systéme	12
3.2.5	Upload súborov ZIP a ich spracovanie	12
3.2.6	Definovanie typov úloh, vytvorenie štruktúry	14
3.3	Šprint #2	16
3.3.1	Zjednotenie Git repozitára	16
3.3.2	Dokončenie detailov	16
3.4	Šprint #3	17
3.4.1	Usporiadanie produktového backlogu	17
3.4.2	Vytvorenie metodík	17
3.4.3	Revízia doterajších zápisníc	17
3.4.4	Vytvorenie formulára pre vyplnenie profilu	17
3.4.5	Nasadenie produkčného servera	17
3.4.6	Získanie informácií prostredníctvom IP používateľa	17
3.5	Šprint #4	19
3.5.1	Vytvorenie filtrov a kreditového systému	19
3.5.2	Vytvorenie referral systému	20
3.5.3	Vytvorenie profilovej stránky	20
3.5.4	Integrácia so sociálnou sieťou Facebook	21
3.5.5	Integrácia so sociálnou sieťou LinkedIn	22
3.5.6	Vytvorenie achievement systému	23
3.6	Šprint #5	25
3.6.1	Vytváranie dokumentácie	25
3.6.2	Facebook lajky a komentáre	25
3.6.3	Integrácia s Google účtom	26
3.6.4	Integrácia s Twitter účtom	26
3.7	Šprint #6	27
3.7.1	Vytvorenie viackrokových formulárov	27
3.7.2	Rebranding - web tímu, dizajn	28
3.7.3	Zlepšená organizácia profilu	28
3.8	Šprint #7	32
3.8.1	Zarovnanie dizajnu produktovej aplikácie s novou identitou tímu	32
3.8.2	Survey task	32
3.8.3	Vytvorenie grafických návodov pre landing page	32
3.8.4	Integrácia so sociálnou sieťou Twitter	32

3.9	Šprint #8	34
3.9.1	Zmena dizajnu badges	34
3.9.2	Vytváranie záznamov z externej aplikácie	35
3.9.3	Úloha s pridávaním tzv. Facebook lajkov	36
3.9.4	Intenacionalizácia (SK, ENG)	38
3.9.5	Zobrazovanie výsledkov	39
4	Celkový pohľad	41
4.1	Opis prototypu	41
4.2	Funkcionálne požiadavky	42
4.3	Prípady použitia	42
4.4	Dátový model	44
5	Používateľská príručka	45
5.1	Prihlásenie do systému	46
5.2	Registrácia do systému	46
5.3	Opätovné zaslanie aktivačného emailu	47
5.4	Sekcia pre pracovníka	47
5.4.1	Profil	48
5.4.2	Badges	48
5.4.3	All tasks	49
5.5	Sekcia pre zadávateľa	49
5.5.1	Profil	50
5.5.2	Badges	50
5.5.3	Tasks	50

1 Úvod

Účel dokumentu

Tento dokument vznikol na predmete tímový projekt iniciatívou členov tímu 11. Slúži na dokumentovanie projektu s názvom Digital Sweatshop.

Štruktúra dokumentu

Štruktúra dokumentu je prispôsobená agilnému, prírastkovému a iteratívne projektovému manažmentu (SCRUM). V jednotlivých kapitolách dokumentu je charakterizovaný konkrétny používateľský príbeh spoločne s jeho špecifikáciou, implementáciou a testovaním.

Zadanie projektu

Známy Digital Sweatshop je napríklad AMT. Spravme vlastný "pracovný tábor", kde objednávateľ zadá prácu a pracant ju za odmenu vykoná. Od experimentov za pívá, cez marketingové prieskumy za zľavy, až po domáce úlohy za body tak zmeňme model mikropráce na Webe.

Prečo si mám vybrať tento projekt:

Okrem toho, že sa veľa naučíš, vytvoríš niečo, čo už teraz má svojich cieľových zákazníkov. Už prvý prototyp budú využívať chtiví experimentujúci skupiny PeWe a k vypusteniu na svetlo sveta to už bude len krôčik.

Napríklad:

pracuješ na svojej diplomovke, kde je nutné vašu metódu nejako overiť. Vytvoríš svoj projekt, napríklad stránku, ktorá zobrazuje výstupy tvojej metódy používateľovi. Ako to overiť, ako to chceš stihnúť? Nuž, na poslednú chvíľu zoženieš 20 ľudí, ktorí ti niečo poklikajú a aj tak to bude málo. Preto vytvoríme tento SweatShop. Zadáš úlohu, opíšeš misiu, nastaviš odmenu, použiješ jednoduché API, pridáš ešte zopár otázok, na ktoré chceš odpoveď a čakáš. Misia sa zobrazí všetkým čakajúcim pracantom a brigáda môže začať. Po ukončení dostaneš výsledky a pracanti svoju odmenu.

Idea je tá istá s akou pracujú aj iné projekty pre mikroprácu. Zaujímavá obmena je v spôsobe odmeňovania. Kto by už robil za pár centov nudnú prácu. Vytvoríme platformu, kde formou hry vyhodnotíme a odmeníme zúčastnených pracantov. Záleží na zadávateľovi, či dá výhercovi čokoládu, alebo víno. Tento model v reále už funguje minimálne v skupine PeWe, kde sú experimentujúci odmeňovaní podobnými darmi. Poďme túto realitu zastrešiť systémom, ktorý sa o mnohé postará. Napríklad aj o automatickú kontrolu, ktorú zadávateľ môže definovať. Nepôjde preto o prácu, ktorú možno oklamať a podmienky pre každého tak zostanú rovnaké.

Základnou náplňou projektu je prostredníctvom webovej aplikácie poskytnúť istý druh rozhrania medzi dvoma skupinami ľudí, z ktorých jedna skupina potrebuje získať určitú formu spätnej väzby a druhá skupina ľudí si chce privyrobiť a v neposlednom rade aj trochu pomôcť. Využitie tohto rozhrania je tak veľké, že môžeme pokryť až veľmi širokú oblasť. Od akademickej pôdy pri rôznych formách úloh, od dotazníkov cez jednoduché úlohy až po zložitejšie overovanie hypotéz

v rámci bakalárskych, diplomových či dizertačných prác.

Ďalšou veľkou oblasťou je marketing. Nahradenie zastaraných marketingových techník a aktivít

(telefónny marketing) za priame oslovenie cieľovej skupiny prostredníctvom nášho rozhrania. Možnosti využitia tohto rozhrania sú teda naozaj skoro neobmedzené.

2 Ciele projektu na zimný semester

Základným cieľom projektu je vytvoriť aktívne používanú webovú či mobilnú aplikáciu, ktorej hlavným cieľom je sprostredkovať mikroprácu. Chceme dosiahnuť spokojnosť na oboch stranách (zadávateľ aj pracovník). Zadávateľovi chceme ponúknuť možnosť získať kvalitnú spätnú väzbu tým že efektívne nájdeme správnych kandidátov na danú úlohu. Pracovníkovi chceme dať možnosť si privyrobiť a motivovať ho systémom reputácie a odmeňovania.

Projekt má potenciál poskytnúť rozhranie využiteľné v rôznych oblastiach. Hlavným cieľom tohto projektu je však pokrytie akademickej a marketingovej oblasti. V budúcnosti sa však určite zameriame aj na iné oblasti.

3 Opisy príbehov

3.1 Pred prvým šprintom

3.1.1 Analýza existujúcich riešení

Autor: Bc. Marek Lenčes

Amazon Mechanical Turk

Podobným riešením pre ponuku mikropráce je systém Amazon Mechanical Turk (MTurk). Obsahuje úlohy, ktoré sa nazývajú HITy (Human intelligence task), ktoré sú jednoduchými úlohami ako napríklad vyhľadanie informácií o firme na webe, tagovanie obrázkov, alebo písanie recenzií. MTurk sa momentálne nachádza v beta verzii a je dostupný len pre zadávateľov na území USA, no pracovníci môžu byť z celého sveta. Zadávatel môže vyžadovať od pracovníka špeciálnu kvalifikáciu na vykonanie úlohy, ak je to potrebné. Pracovníka môže zadávateľ aj odmietnuť, a tým sa pracovníkovi zhorší reputácia v rámci systému, a bude mať problém získať úlohu na splnenie. Zadávatelia vyplácajú 10% z úspešne vykonaných úloh, prevádzkovateľovi systému, ktorým je spoločnosť Amazon.

The screenshot shows the Amazon Mechanical Turk website. At the top, there's a navigation bar with 'Your Account', 'HITS', and 'Qualifications' tabs. Below that, a banner states 'Mechanical Turk is a marketplace for work.' and 'We give businesses and developers access to an on-demand, scalable workforce. Workers select from thousands of tasks and work whenever it's convenient. 443,562 HITs available. View them now.' The main content is split into two columns. The left column is titled 'Make Money by working on HITs' and describes HITs as individual tasks. It lists benefits for workers: 'Can work from home', 'Choose your own work hours', and 'Get paid for doing good work'. It includes a flowchart: 'Find an interesting task' (with a magnifying glass icon) -> 'Work' (with a gear icon) -> 'Earn money' (with a dollar sign icon). A 'Find HITs Now' button is at the bottom. The right column is titled 'Get Results from Mechanical Turk Workers' and describes asking workers to complete HITs. It lists benefits for requesters: 'Have access to a global, on-demand, 24 x 7 workforce', 'Get thousands of HITs completed in minutes', and 'Pay only when you're satisfied with the results'. It includes a flowchart: 'Fund your account' (with a plus sign icon) -> 'Load your tasks' (with a gear icon) -> 'Get results' (with a star icon). A 'Get Started' button is at the bottom.

Obr. 1: Amazon MTurk

3.1.2 Výber a inštalácia podporných nástrojov

Autor: Bc. Patrik Štrba, Bc. Martin Toma

Ruby on Rails

Jedným z prvých spoločných rozhodnutí tímu bol výber základných technológií. Vzhľadom na špecifiká projektu (webová aplikácia) a pomerne „start-up“ like charakter sme po miernom výbere zúžili možné technológie na tieto tri:

- Groovy (JVM based)
- Ruby (Ruby on Rails)

- Python (Django)

Naším prvým rozhodnutím bol jazyk Groovy založený na jazyku JAVA. Prvé dva šprinty sme venovali aktivitám, ktoré rátali s týmto jazykom. Viaceré problémy nás však donútili prehodnotiť tento výber a rozhodnúť sa, či nám rýchla zmena technológií v súčte neušetrí mnoho času v porovnaní s tým, keby sme riešili problémy našej pôvodne vybranej technológií.

Rozhodli sme sa kým bol ešte čas zmeniť technológiu na projektovým vedúcim odporúčaný framework Ruby on Rails, v ktorom momentálne vyvíjame. Konkrétne používame verziu Ruby 2.0 a verziu Rails 4.0. Mierne problémy nám spôsobuje pomerne čerstvo vydaná verzia ale chceli sme vyvíjať v tom najnovšom a najmodernejšom prostredí aké je k dispozícii.

JetBrains RubyMine

Pri výbere nástroja na vývoj sme museli vybrať nástroj, ktorý spĺňa nasledujúce podmienky:

- musí byť zadarmo (alebo školská licencia)
- multiplatformové
- dobrá podpora jazyka ruby
- dobrá podpora webového frameworku ruby on rails
- aktívny vývoj

Na základe týchto podmienok nám ostal len nástroj RubyMine od spoločnosti JetBrains.

Atlassian JIRA

Manažment úloh je proces, ktorý taktiež vyžaduje softvérovú podporu. Pri výbere nástroja na tento účel sme požadovali tieto vlastnosti:

- podpora agilných metód (konkrétne SCRUM),
- jednoduchosť a intuitívnosť,
- možnosť externého hostingu.

Z týchto kritérií nám zostalo viacero kandidátov, no v konečnom dôsledku sme sa rozhodli pre JIRU od spoločnosti Atlassian. Tento produkt je v oblasti manažmentu úloh momentálne svetovou jednotkou a to bol rovnako dôvod nášho výberu. Momentálne teda používame JIRU vo verzii 5.1.6, s rozšírením Grasshopper pre podporu agilnej metódy SCRUM.

Bitbucket

V tejto oblasti existujú dve populárne služby na zdieľanie repozitárov a to:

- Bitbucket
- Github

Služba musela spĺňať nasledovné podmienky:

- musí byť zadarmo (minimálne pre 7 ľudí)
- možnosť vytvorenia privátneho repozitára
- musí podporovať správu oprávnení

Tieto podmienky spĺňajú obe služby a preto sme sa spoločne rozhodli pre github. Avšak neskôr pre problémy pri integrácii s nástrojom JIRA sme sa rozhodli použiť Bitbucket, ktorý poskytuje rovnaká firma ako nástroj JIRA.

Jenkins

Pri procese nasadenia sme museli vybrať nástroj, ktorý spĺňa nasledujúce podmienky:

- musí byť zadarmo
- podpora jazyka ruby
- jednoduchá správa
- podpora verziovacieho nástroja git
- dostupný pre unix/linux systémy
- podpora v službe Bitbucket

Na základe týchto podmienok a skúsenosti sme vybrali nástroj Jenkins, ktorý po pridaní zmien do repozitára (vetvy master) spustí proces nasadenia a aktualizuje produkčnú verziu projektu.

3.1.3 Vytvorenie webového sídla

Autor: Bc. Róbert Černý, Bc. Marek Lenčoš, Bc. Miroslav Molnár

Medzi primárne ciele nášho tímu pre prvý šprint patrí vytvorenie webovej stránky tímu 11. Slúži na prezentovanie vykonanej práce tímu, ako aj poskytnutie dôležitých informačných zdrojov o postupe a aktuálnom stave vyvíjaného projektu. Spoločne sme určili tieto požiadavky, ktoré webová forma prezentácie nášho tímu bude spĺňať:

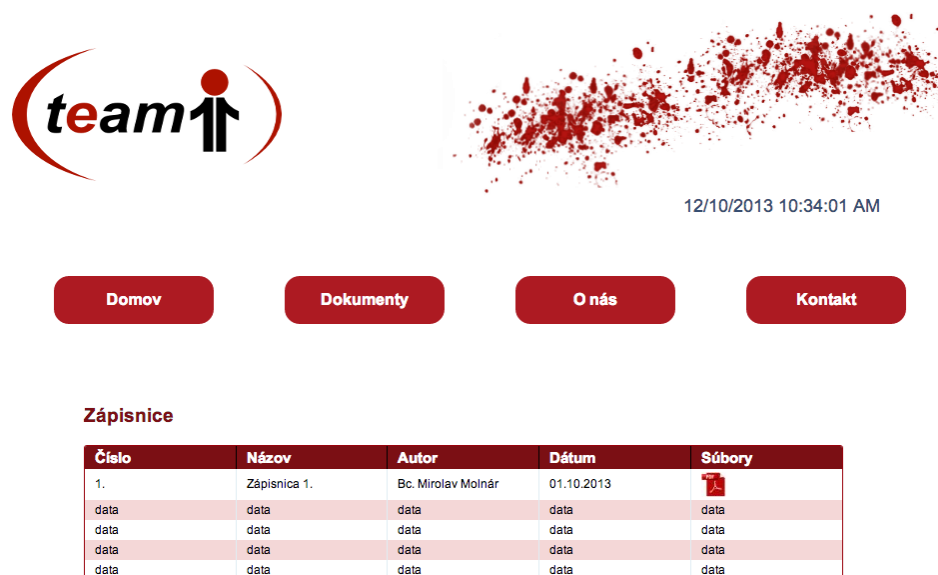
- ľahká dostupnosť a kvalita informačných zdrojov (dokumenty, plagát)
- kvalita nad kvantitou
- intuitívna navigácia na stránke
- responzivnosť
- minimalistický dizajn
- dizajn prispôbený nášmu zadaniu (logo, motívy)

Na stránke sa budú nachádzať informácie o projekte Digital Sweatshop, ako aj motivácia a potenciál tohto projektu. Zodpovieme na základné otázky ako:

- Čo to je Digital Sweatshop ?
- Prečo to robíme ?
- Ako to funguje ?
- Kto sú členovia tímu 11 ?

Návštevník si môže dôkladne prezrieť zostavenie tímu a popis jednotlivých členov. Každý člen má pri svojom popise aj aktuálnu fotku. Pre možnosť komunikácie je v popise člena tímu uvedený aj jeho emailový kontakt.

Samozrejmosťou je sekcia dokumentov, kde budú voľne dostupné dokumenty na stiahnutie. Ich cieľom je zaznamenať postup, aký sme pre plnenie tohto projektu zvolili. Prvý prototyp spĺňa nutné podmienky avšak, všetci sme si vedomí, že po bližšej špecifikácii, vytvorení loga a príznačného názvu pre náš projekt, dáme našej webovej stránke iný rozmer. Náhľad na prvú verziu nášho webového sídla po prvom šprinte je zobrazený nižšie.



Obr. 2: Prototyp tímového webu

3.1.4 Lean Canvas

Autor: Všetci členovia

Academics and Marketing

PROBLEM Obtaining high-quality product feedback or feedback from specific group.	SOLUTION Clients are posting tasks and users are solving them.	UNIQUE VALUE PROPOSITION Very user/friendly (intuitive) UI. Focus also on Slovak market. User motivation program Game-like features.	UNFAIR ADVANTAGE Something that can't be easily copied or bought 9	CUSTOMER SEGMENTS Academics and Science communities Marketing and PR
	EXISTING ALTERNATIVES Telemarketing Surveys Public interviews		KEY METRICS Number of posted tasks. Number of processed tasks. Number of registered users/clients. Number of active users/clients.	
COST STRUCTURE Maintance Hosting and scaling costs		REVENUE STREAMS From clients (posting tasks) From both (small part from task revenue)		

Obr. 3: Lean Canvas

3.2 Šprint #1

3.2.1 Vytvorenie základných rozhraní aplikácie

Registrácia a prihlásenie

Autor: Bc. Patrik Štrba, Bc. Martin Toma

Ako skoro každá webová aplikácia, tak aj tá naša vyžaduje funkcionality registrácie a následného prihlásenia používateľov. Preto sme sa už v druhom šprinte rozhodli implementovať základ tejto funkcionality. Navrhli a implementovali sme teda základné možnosti registrácie s použitím jednoduchej metódy šifrovania hesla v databáze. Po šprinte sme teda boli schopní simulovať registráciu a následné prihlásenie do aplikácie. Zhodli sme sa, že neskôr bude asi potrebné implementovať sofistikovanejšie metódy šifrovania, overenie registrácie prostredníctvom emailu a možnosti obnovenia strateného hesla. To sme však v danej dobe neriešili.

Vytvorenie jednoduchej úlohy

Autor: Bc. Róbert Černý, Bc. Filip Bednárík

Na pohľad veľmi jednoduchá úloha avšak pri jej implementácii je nutné vytvoriť dátový model a prispôbiť ho požiadavkám na systém. Vytváranie a plnenie úloh v našej aplikácii tvorí jadro systému, preto je dôležité, aby sa logika poriadne premyslela. V druhom šprinte sme implementovali základný systém správy úloh a podúloh, čo je jadrom nášho systému. Na začiatok sme vytvorili model Task a Subtask, ich prepojenie a jednoduchý CRUD, ktorý nám umožňuje spravovať záznamy v databáze. Pomaly sme sa začali rozhodovať aké vlastnosti a atribúty budú niesť úlohy a podúlohy a tiež sme sa rozhodli, že riešiť sa budú podúlohy, ktoré úlohy budú agregovať. Identifikovali sme základné vlastnosti úlohy a to sú: Názov, Popis, Kategória/Typ a počet ľudí, ktorý budú túto úlohu môcť riešiť.

3.2.2 Rozbitie prvej User Story

Autor: Bc. Filip Bednárík

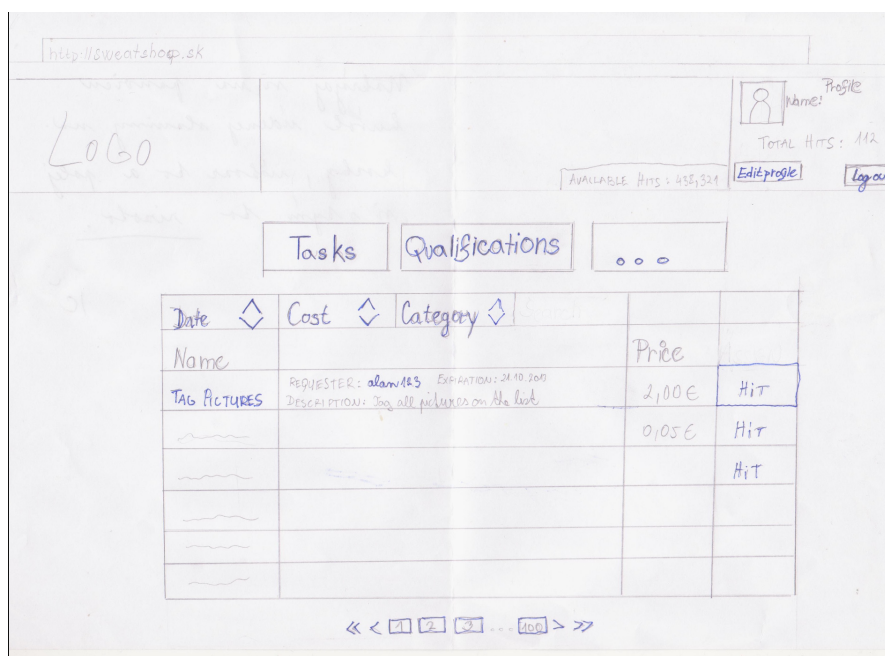
Našou primárnym cieľom je spraviť aplikáciu, ktorá bude využiteľná pre našich kolegov v akademickej doméne, kde je veľký dopyt po takomto systéme, ktorý poskytuje overenie rôznych počítačových algoritmov človekom. Najzákladnejším user story je, že zadávateľ nahrá do systému obrázku a pracovníci mu pri riešení úlohy napíšu aké objekty sa na obrázku nachádzajú. V tomto šprinte sme integrovali pridávanie a rozbalovanie zip súborov s obrázkami a vytváranie úloh a podúloh na základe týchto obrázkov. Z pohľadu zadávateľa sme teda upravili základný CRUD tak aby vytváranie úlohy obsahovalo tlačidlo vybrať súbor a po nahratí zip súboru sa tento rozbalil a vytvorili sa podúlohy pre každý jeden obrázok. Z pohľadu pracovníka sme navrhli obrazovku v ktorej vidia pracovníci všetky dostupné úlohy a môžu si zvoliť úlohu, ktorú budú riešiť. Pri riešení úlohy prechádzajú všetky podúlohy danej úlohy a postupne vytvárajú nové riešenia pre danú podúlohu. Pri dokončení všetkých podúloh sa označí úloha za ukončenú. Pri odídení zo stránky riešenia a návrate na stránku zoznamu všetkých úloh má pracovník možnosť sa vrátiť k riešeniu úlohy a to tak, že mu systém ponúkne vyriešiť poslednú nevyriešenú podúlohu danej úlohy.

3.2.3 Návrh rozhrania webovej aplikácie

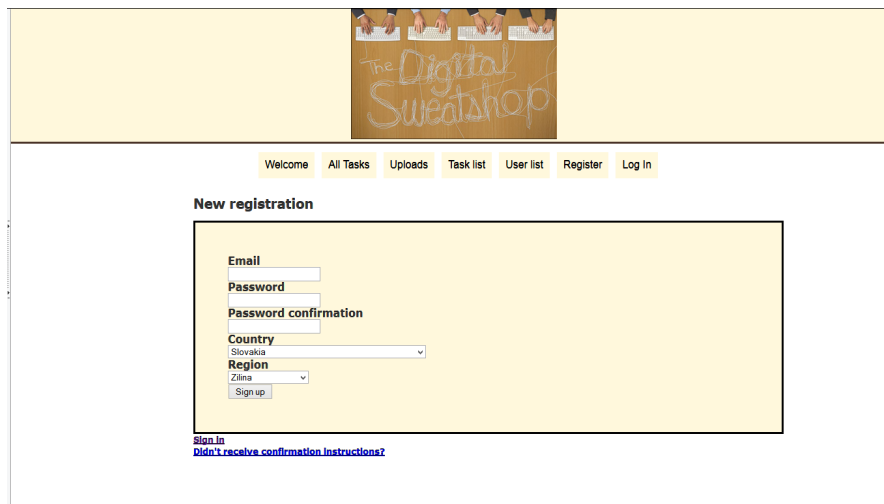
Autor: Bc. Róbert Černý

Webové rozhranie aplikácie je momentálne zložené z jednoduchých HTML súborov bez štýlovania. Vytvorením prvého návrhu dizajnu stránky zjednodušíme a sprehladníme proces vývoja ďalších úloh. Cieľom je vytvoriť jednotné grafické rozhranie, ktorého princípy sa použijú v celej aplikácii. Základnými požiadavkami je vytvorenie navigačného panela, pre jednoduchšie zapracovávanie privilégií používateľov ako aj pre orientáciu v aplikácii. Dôležitým aspektom je aj zobrazovanie tabuliek v jednotnom formáte, nakoľko sa v systéme často objavujú. Prvá verzia UI slúži iba na demonštrovanie implementovaných používateľských príbehov a v budúcnosti sa prepracuje z rôznych aspektov.

Low fidelity návrh UI



Obr. 4: Prvý návrh grafického používateľského rozhrania aplikácie



Obr. 5: Prvý návrh grafického používateľského rozhrania aplikácie

3.2.4 Autorizácia a autentifikácia používateľa v systéme

Autor: Bc. Patrik Štrba, Bc. Martin Toma

Neskôr sme neboli dostatočne spokojný s implementáciou registrácie a prihlásenia, tak sme sa po dohovore dohodli použiť radšej už vytvorený gem, ktorý zjednoduší implementáciu a pridá ďalšiu funkcionálnu potrebnú do budúcnosti.

V tomto šprinte sme použili gem *devise*, ktorý rieši problém autentifikácie a zároveň umožňuje nastaviť, ktorá funkcionálna sa má použiť. Tento gem obsahuje teda autentifikáciu, ale aj registráciu a funkcionálnu okolo toho, napríklad potvrdenie účtu pomocou emailovej adresy alebo zabudnuté heslo.

Pre korektné fungovanie tohto gemu sme museli najskôr vytvoriť model používateľa, ktorý sme neskôr spojili s funkcionálnou gemu.

Po dokončení autentifikácie sme prešli na analýzu autorizácie. Pre autorizáciu sme zvolili gem *canon*, ktorý spolu s gemom *devise* zabezpečujú autentifikáciu a autorizáciu. Avšak sme posúdili, že autorizáciu ešte nie je potrebné implementovať do systému vzhľadom na malý počet používateľských rolí a malého vytvorenej funkcionality.

3.2.5 Upload súborov ZIP a ich spracovanie

Autor: Bc. Miroslav Vojtuš

Analýza

Ako jednu zo základných úloh sme identifikovali tagovanie obrázkov. Pre získanie obrázkov je nutné zvážiť dostupné možnosti. V rámci analýzy sme zvažovali:

- jednotlivé odovzdanie http protokolom - po jednom obrázku
- implementáciou rest API alebo WS klientom
- simulované hromadné odovzdanie - postupné vyberanie obrázkov po jednom a odoslanie naraz
- zoskupenie obrázkov do ZIP archívu s pevne stanovenou štruktúrou obsahu

Špecifikácia požiadaviek

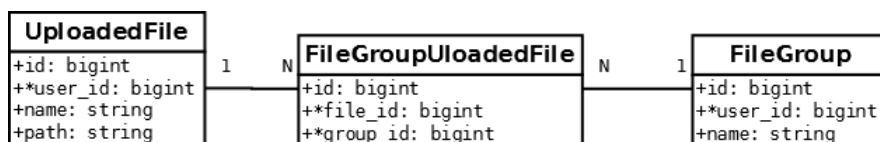
Po funkcionálnej stránke je odovzdanie jednoduchý transport súboru, ktorého obsah sa vyhodnotí a rozbalí na strane serveru a informuje používateľa o stave importu.

Nefunkcionálne požiadavky sme identifikovali nasledovne:

- stabilita prenosu
- stabilita importu
- udržateľnosť indexov súborov pre spätné operácie(CRUD)

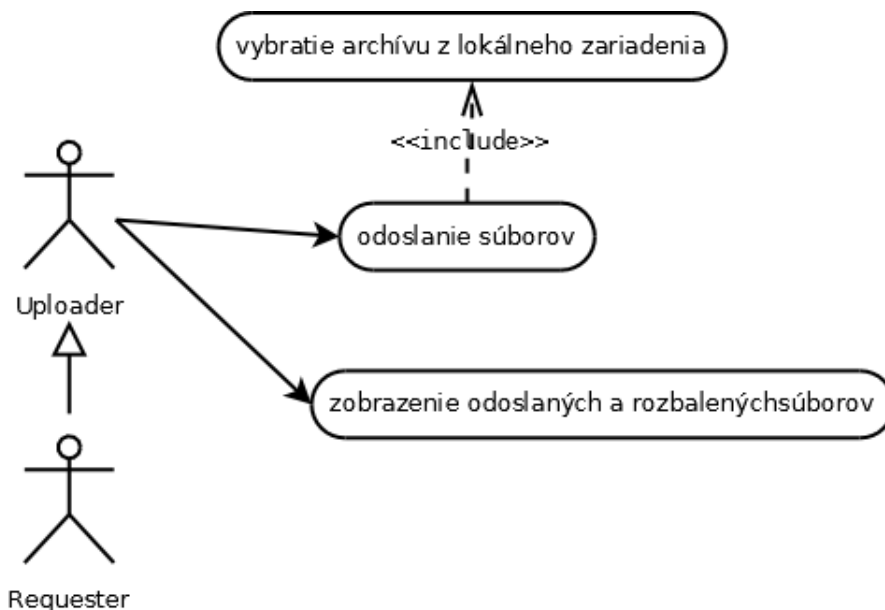
Návrh

Dátová vrstva je opísaná nasledujúcim obrázkom. V entite UploadedFile je path relatívna cesta k súboru na disku. Entita FileGroup zastrešuje zoskupenie súborov a nahrádza pojem archív, vo výsledku súbory už nie sú v jednom archíve ale len zoskupené.



Obr. 6: Dátový model

Príbeh je opísaný UC diagramom na nasledujúcom obrázku. Zadávatel vystupujúci v role odosielateľa má možnosť odoslať súbor vybratím z lokálneho zariadenia a zobraziť si už odoslané súbory.



Obr. 7: Model prípadu použitia

Implementácia a testovanie

Príbeh je implementovaný ako jednoduchý Ruby skript naviazaný na Ruby on Rails programový rámec, kde je hlavnú funkcionality implementovaný v modeli FileUpload.

V aktuálnej verzii je validita rozbaleného archívu zabezpečená jednoduchou validáciou pomocou rozpoznávania názvov súborov a odvodenia ich typu(MIME-TYPE). Dátový obsah jednotlivých súborov nie je analyzovaný, nakoľko záleží na zobrazači ako ošetrí bezpečnosť zobrazených dát z pohľadu potenciálnych útokov. V prípade obrázkov nehrozí žiadne vážnejšie riziko okrem odosielania nežiadúceho alebo zakázaného obsahu. V takomto prípade je nutné manuálna kontrola.

K súborom sa dá pristupovať jednotlivo pomocou abstraktnej cesty odvodenej z identifikátora súboru: '/file/:id' (napr. <http://www.pinta.sk/file/1>). Fyzická cesta k súborom je blokováná pre verejnosť - len pre privátny prístup.

Testovanie je zabezpečené niekoľkými jednotkovými testami na úrovni odosiela-
nia súborov pre základne hraničné prípady ako:

- prázdny archív
- neznámy MIME-TYPE
- viac ako N súborov, kde N je konštanta definovaná konfiguráciou
- archív s podzložkami
- nevalidný binárny súbor s príponou .zip - simulácia útoku

3.2.6 Definovanie typov úloh, vytvorenie štruktúry

Autor: Bc. Marek Lenčes, Bc. Miroslav Molnár

Bol vytvorený prieskum aktuálne najpopulárnejších úloh na konkurenčnom serveri Amazon MTurk, ktorý taktiež slúži na to aby používatelia mohli vytvárať a vypracovávať úlohy. MTurk je komerčný a ak si chce používateľ nejakú úlohu vytvoriť na serveri, musí im za vytvorenie zaplatiť. MTurk taktiež nieje rozšírený na viaceré krajiny ako je napríklad Slovensko. Po prejdení dostupných úloh sme vypísali príklady tých najpoužívanejších a vytvorili sme kategórie do ktorých sme ich zaradili. Spísali sme pre úlohy potrebné vstupy a výstupy.

Kategórie úloh:

Vyberanie dát z obrázka

Používateľ z obrázku vyčítava údaje, ktoré vyžaduje zadávateľ.

Príklady:

Zapíš informácie z pokladničného bloku(mena a ceny produktov).

Vstupy: obrázok

Výstupy: textové informácie

Tagovanie obrázkov niekoľkými slovami

Vstupy: obrázok

Výstupy: textový dokument s konkrétnym počtom slov

Survey (prieskum): vyplnenie dotazníku

Príklady:

Vypracuj dotazník.

Vstupy: všeobecné informácie o sebe

Výstupy: vyplnený dotazník

Nájdenie informácií na webe: nájdenie informácií zadaných zadávateľom.

Príklady:

Nájd kontaktnú osobu a jej email adresu pre danú URL.

Vstupy: odkaz URL

Výstupy: kontaktná osoba a údaje o nej.

Nájd najrelevantnejší výsledok pre každé poskytnuté kľúčové slovo v internetovom vyhľadávači google.com.

Vstupy: kľúčové slová

Výstupy: množina najrelevantnejších slov.

Nájd webové sídlo (URL) firmy.

Vstupy: názov firmy

Výstupy: odkaz URL

kategorizácia, sortovanie, zaradenie dát: zaradte dáta do zadávateľom určených kategórií.

Príklady:

Kategorizácia obrázkov podľa kritérií (odfotený vnútri, vonku, doma, v robote).

Vstupy: obrázky

Výstupy: výberové tlačidlá s viac možnosťami pre obrázok

vytváranie textu, upraví textu: vytvorenie dokumentu podľa zadávateľom zadaných požiadaviek.

Príklady:

Oprav a prepíš text z PDF

Vstupy: pdf

Výstupy: textový súbor

Prelož zadaný text do iného jazyka

Vstupy: text, PDF, obrázok,..

Výstupy: text

Napísať recenziu na produkt, vytvoriť dokument s požadovanými vlastnosťami a informáciami.

Vstupy: text popis

Výstupy: text dokument

Práca s médiami a spracovaní informácií z nich

Príklady:

Prepíš audio nahrávku do textu

Vstupy: URL odkaz na audio nahrávku

Výstupy: text

Vytvor titulky na daný film.

Vstupy: URL na daný film uložený na

Výstupy: súbor s titulkami

3.3 Šprint #2

3.3.1 Zjednotenie Git repozitára

Autor: Bc. Patrik Štrba

V tomto šprinte sme menili architektúru tímového repozitára. Doteraz sme používali všetci jeden spoločný repozitár, ktorý sme upravili z dôvodov bezpečnosti, konzistencie a jednoduchšej údržby. Následne sme prešli na vlastné repozitáre a funkcionality sa pridávala do tímového repozitára pomocou pull-requestov. Týmto sme zabezpečili transparentnosť nad vytvorenýnou funkcionalitou a dosiahli väčšiu kontrolu pridanej funkcionality.

Taktiež je táto nová architektúra vhodnejšia pre tvorbu code-review, kedy si môže každý člen z tímu pozrieť pull-request a schváliť ho, prípadne k nemu zabechať komentáre.

Pri schvalovaní pull-requestu sme zvolili taktiku, kedy je potrebné pre schválenie pull-requestu vyjadrený súhlas minimálne dvomi členmi a to manažér podpory vývoja a vedúci tímu.

3.3.2 Dokončenie detailov

Autor: Všetci členovia

Nakoľko sa v predchádzajúcom šprinte naplánovalo viac netriviálnych používateľských príbehov, bude potrebné implementácie upraviť. Každý člen tímu pracoval na vlastnej vetve, preto bude tento proces jednoduchší. Tieto úpravy sú nutné z pohľadu konzistentnosti a spolupráce jednotlivých funkcií navzájom.

3.4 Šprint #3

3.4.1 Usporiadanie produktového backlogu

Autor: Bc. Patrik Štrba, Bc. Martin Toma

V piatom šprinte po implementácii niekoľkých základných funkcionalít sme sa zhodli, že by sme potrebovať upratať produktový backlog aby sme vytvorili lepší základ do ďalších šprintov. Stretli sme sa preto špeciálne v neformálnom prostredí s projektovým vedúcim a prebehol brainstorming o možnostiach a smerovaní projektu. Výsledkom bola identifikácia nových features a upratanie existujúcich. Takisto sme sformulovali základné prípady použitia, ktoré je potrebné mať stále v hlavne pri vývoji celej aplikácie.

3.4.2 Vytvorenie metodík

Autor: Všetci členovia

V tíme zastáva každý člen svoju rolu v rámci riadenia. Jednou z jeho povinností je navrhnúť metodiku (jednoznačnú, záväznú, potrebnú), ktorú budú členovia tímu brať ako návod na to, ako správne fungovať. Výstupom sú teda dokumenty obsahujúce pravidlá, ktoré jednoznačne vysvetľujú čo sa bude robiť akým spôsobom a kto je za ne zodpovedný.

Očakávame, že používanie metodík zlepší naše výsledky, nakoľko bude riadenie oveľa jasnejšie a bude mať písomnú formu, ktorá bude prístupná celému tímu.

3.4.3 Revízia doterajších zápisníc

Autor: Bc. Marek Lenčes

Zistili sme drobné nedostatky pri vypracovávaní zápisníc, preto sme sa na ne viac zamerali, a zapísali všetky dôležité údaje, ktoré sme na stretnutiach preberali. Tiež sme jasne definovali, kto mal v ktorý šprint pridelenú akú úlohu.

3.4.4 Vytvorenie formulára pre vyplnenie profilu

Autor: Bc. Miroslav Molnár

Pre detegovanie vlastností a schopností pracovníka sme sa dohodli, že nám najlepšie pomôže profil. Zadávateľ pomocou týchto informácií môže pre svoju úlohu vybrať pracovníka s potrebnými vlastnosťami a schopnosťami. Z tohto pohľadu profil slúži ako jeden z faktorov vplývajúci na zvýšenie kvality práce v našom systéme.

3.4.5 Nasadenie produkčného servera

Autor: Bc. Patrik Štrba

V tomto šprinte sme dostali priradené časti subdomén a preto sme mohli vytvoriť už funkčnú verziu projektu prezentovateľnú aj ostatným členom v tíme a verejnosti.

3.4.6 Získanie informácií prostredníctvom IP používateľa

Autor: Bc. Miroslav Vojtuš

Analýza

Pomocou predvyplnenia základných informácií môžeme používateľovi uľahčiť registráciu. Z IP adresy vieme získať odhadom krajinu a kraj používateľa. Existujú dostupné implementácie ktoré sa líšia v presnosti odhadu. Máme možnosť využiť externé API rozhrania ktoré ponúka napríklad služba IP info db alebo GeoIP od MaxMind Inc., prípadne vykonať vyhľadávanie nad lokálnou db pomocou vlastných prostriedkov.

Špecifikácia

Funkcionálne:

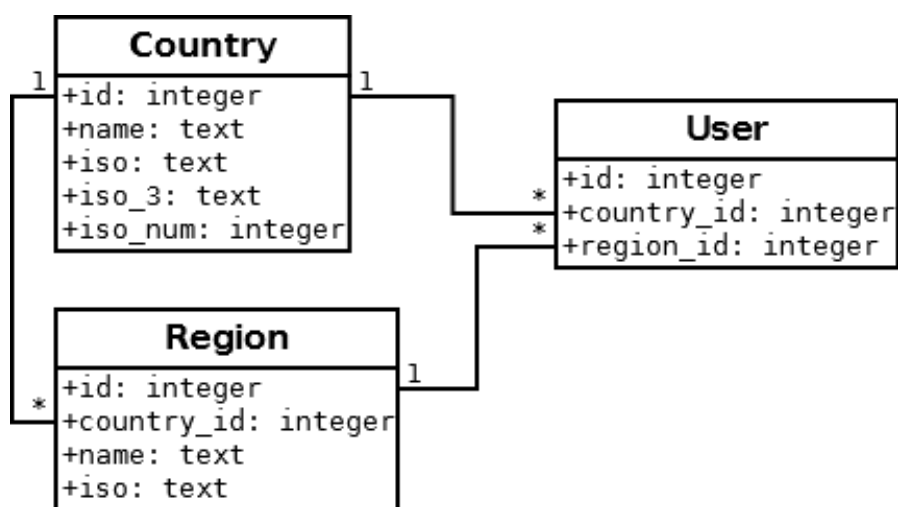
- predvyplnenie polí “country” a “region” automaticky na základe IP

Nefunkcionálne:

- spoľahlivosť - maximalizovať dostupnosť služby
- použiteľnosť - informácie musia byť vyplnené v reálnom čase

Návrh

Dátový model:



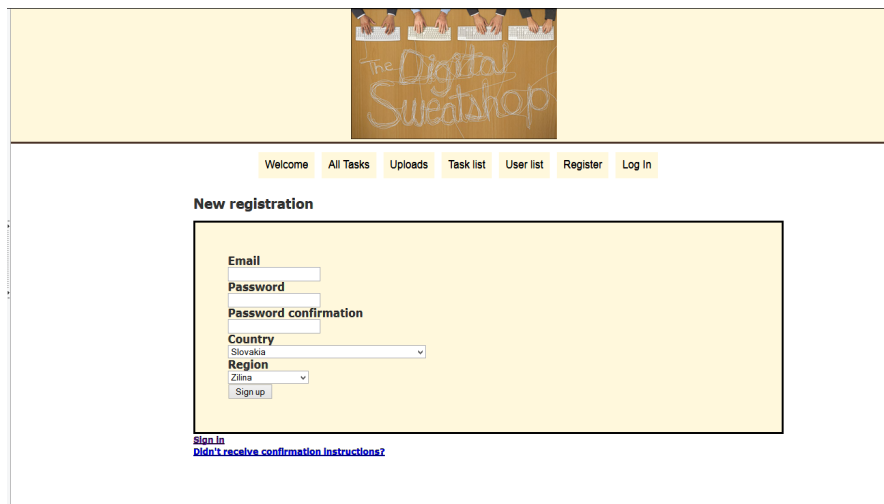
Obr. 8: Dátový model

Implementácia a testovanie

V implementácii sme sa rozhodli využiť lokálnu databázu nad ktorou vykonáme prehľadávanie pomocou rails gemu *geoip*.

Všetky dátové entity sú ukladané v lokálnej databáze. Gem *geoip* vykonáva operácie nad vlastnou lokálnou databázou. Obsah tejto je synchronizovaný pri zavádzaní s databázou nášho systému. Používateľské rozhranie: Pri registrácii má používateľ dopredne vyplnené polia získateľné pomocou služby GeoIP - Country, Region.

Nakoľko je na implementáciu použitý externý gem, nie sú vykonané žiadne automatické testy. Integrovaný test bol vykonaný manuálne pre overenie kvality použitej implementácie. Polia sú v lokálnych podmienkach vyplnené v reálnom čase do 0,5 sekundy. Odhad je presný na krajinu, región je pre slovenskú republiku nepresný.



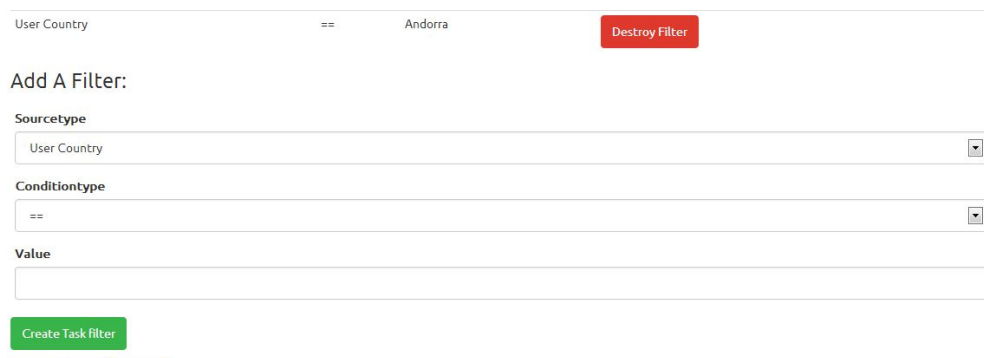
Obr. 9: Používateľské rozhranie

3.5 Šprint #4

3.5.1 Vytvorenie filtrov a kreditového systému

Autor: Bc. Filip Bednárík

Ak chce zadávateľ určiť pre koho sú úlohy určené potrebuje filtrovať pracovníkov, ktorí môžu úlohu ktorú zadal riešiť. Preto sme sa rozhodli v tomto šprinte pridať funkciu správy filtrov pre zadanú úlohu. Filtre je možné zadávať až po vytvorení úlohy. Jednoducho v grafickom rozhraní. Jednotlivé filtre sa skladajú z troch častí: SourceType, ConditionType a Value.



Obr. 10: Pridávanie filtrov

Source type je nejaká vlastnosť používateľa alebo niečo čím sa odlišuje od ostatných. Momentálne sú implementované základné typy zdroja filtra a to sú:

- User Country (Krajina používateľa)
- User Solved Tasks (Množina všetkých používateľom vytvorených úloh)

ConditionType je operátor podmienky. Momentálne sú implementované nasledujúce základné typy podmienok:

- == (Zdroj sa musí rovnať hodnote)
- != (Zdroj sa nesmie rovnať hodnote)

- > (Zdroj musí byť väčší ako hodnota)
- < (Zdroj musí byť menší ako hodnota)
- >= (Zdroj musí byť väčší alebo rovný ako hodnota)
- <= (Zdroj musí byť menší alebo rovný ako hodnota)
- includes (Zdroj ako množina musí obsahovať hodnotu)

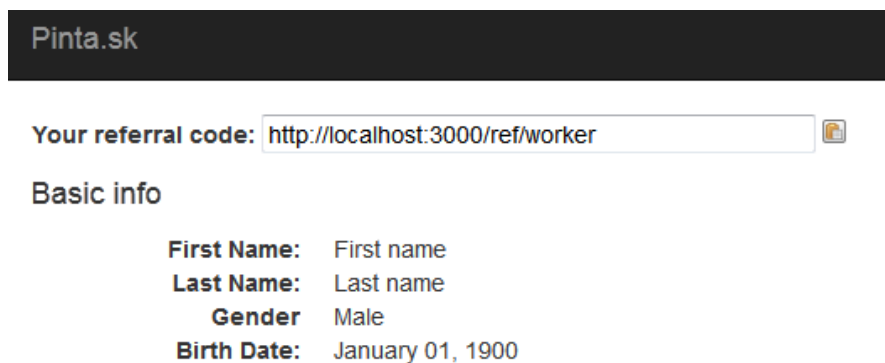
Value je akákoľvek platná hodnota.

Jednou zo základných funkcií našej aplikácie je kreditný systém. Vďaka kreditnému systému totiž môžeme odmeňovať pracovníkov a spravovať zdroje zadávateľov. Preto sme v tomto šprinte implementovali základný kreditný systém, ktorý zahrňuje počet kreditov pre každého používateľa, výšku odmeny za vyriešenie jednej podúlohy a počet kreditov, ktoré je ochotný zadávateľ minúť na jednu úlohu. Pri vyriešení podúlohy pracovníkom sa odpočíta odmena za vyriešenie podúlohy z celkového rozpočtu pre danú úlohu a tieto kredity sa pripočítajú na pracovníkov účet. Museli sme upraviť aj zobrazenie zoznamu úloh pre pracovníkov aby sme zamedzili riešeniu úloh, ktoré už nemajú rozpočet.

3.5.2 Vytvorenie referral systému

Autor: Bc. Marek Lenčes

Po aplikovaní kreditného systému sme zvažili možnosť využitia referral odkazov pre získanie čo najväčšieho množstva nových pracovníkov, ale aj zadávateľov. Vďaka referral odkazom je možné nie len získať nových pracovníkov, ale tých ktorí tento referral odkaz poskytnú, môžeme odmeniť nejakou výškou kreditu. Referral odkaz v profile používateľa je znázornený na Obrázku 11



Obr. 11: Referral odkazy

3.5.3 Vytvorenie profilovej stránky

Autor: Bc. Miroslav Molnár

Profil je jednou z najdôležitejších častí celého nášho systému. Zastrešuje všetky informácie o používateľoch. Tieto informácie sa následne používajú pri identifikácii cieľovej skupiny pracovníkov pre konkrétne zadanie. Vďaka kvalitným informáciám z profilu sa znižuje riziko zlého vypracovania úlohy a tak sa systém stáva efektívnejší a stabilnejší. To je cieľom našej aplikácie, poskytnúť používateľom kvalitné služby.

Analýza

Pre profil nie sú vhodné všetky informácie o užívateľovi. Boli detegované a vybraté hlavné dáta do profilu:

- Firstname - meno užívateľa
- Lastname - priezvisko užívateľa
- Birth_date- dátum narodenia užívateľa
- Gender - pohlavie užívateľa
- User_description- textový popis užívateľa
- Twitter_pin- pin užívateľa na užívateľský účet užívateľa na twitter.

Boli detekované aj ďalšie dáta ktoré sú vhodné pre ich umiestnenie do profilu:

- Language_skills – jazykové znalosti
- Computer_skills - počítačové znalosti

Špecifikácia požiadaviek

Vytvorte profil užívateľa s dátami:

- meno
- priezvisko
- dátum narodenia
- pohlavie

Doimplementujte výpočet:

- Vek používateľa
- Počet spravených úloh
- Percentuálne hodnotenie spravených úloh

Návrh

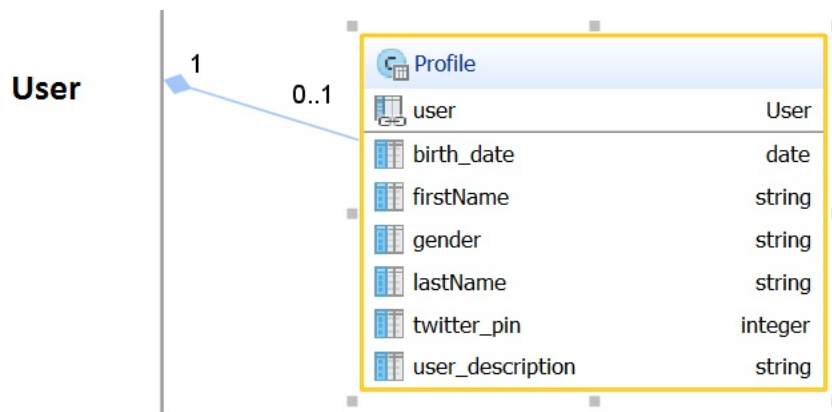
Bol navrhnutý dátový model popísaný na obrázku nižšie, ktorý zobrazuje aktuálne dáta v databáze.

Implementácia a testovanie

Profil je jednoznačne prepojený s používateľom. Pri vytváraní nového používateľa sa vytvára aj jeho profil s predvolenými dátami. Po zmazení používateľa sa zmaže aj profil na neho naviazaný.

3.5.4 Integrácia so sociálnou sieťou Facebook

Autor: Bc. Patrik Štrba



Obr. 12:

Analýza

V dnešnom svete existuje veľké množstvo sociálnych sietí. Sociálne siete sú veľkým zdrojom dát, ale aj veľmi vhodné na propagáciu a zviditeľnenie aplikácie, či jej produktov.

Špecifikácia požiadaviek

Potreba integrácie so sociálnymi sieťami vzniká ako potreba integrácie sociálnych prvkov a prilákanie nových potenciálnych používateľov touto cestou.

Návrh

Aplikácia bude schopná komunikovať cez rozhranie so sociálnou sieťou facebook, kde bude použitý autorizačný mechanizmus OAuth. Pomocou tohto autorizačného mechanizmu spárujeme používateľa aby sme mohli získať o ňom údaje, ktoré budeme potrebovať v aplikácií. Používateľ bude mať taktiež možnosť sa zapojiť do verejnej diskusie, ktorá bude fungovať pre všetkých, ktorý majú vytvorený účet na tejto sociálnej sieti. Používateľ bude taktiež mať možnosť vyjadriť jeho spokojnosť s danou úlohou, prípadne s inou aktivitou či produktom.

Implementácia a testovanie

Funkcionalita je implementovaná pomocou gemu Koala. Tento gem zabezpečuje komunikácie a tak je možné abstrahovať a zjednodušiť prácu pri integrácií. V rámci implementácie sa implementovala funkcionalita na komunikáciu v diskusiách, ktoré boli pridelené jednotlivým úlohám. Taktiež bola implementovaná možnosť označovať jednotlivé úlohy používateľom, že sa im páčia.

3.5.5 Integrácia so sociálnou sieťou LinkedIn

Autor: Bc. Martin Toma

Analýza

V našom projekte je veľmi dôležité pracovať s relevantnými základnými údajmi o používateľoch. V aplikácia bude možnosť definovať úlohy pre špecifické pohlavie, vek, vzdelanie, schopnosti a znalosti. Preto sme hľadali spôsob ako získať tieto

údaje. Dôležité však je aby používateľ nemusel pracne vyplňovať tieto údaje. Sociálna sieť LinkedIn momentálne zažíva vrchol popularity. Automaticky sa sňou spája pracovný život ľudí a teda formálne a relevantné údaje. Integráciou s touto sociálnou sieťou teda používateľovi ušetríme čas a zároveň my získame relevantné údaje.

Špecifikácia požiadaviek

Požiadavkou je aby sme boli schopní základné informácie ako :

- meno a priezvisko,
- dátum narodenia (vek),
- motto,
- zoznam znalostí a pracovné skúsenosti,
- vzdelanie.

Ďalšou požiadavkou je dať používateľovi možnosť vybrať si, či chce aktualizovať konkrétny atribút vo svojom profile.

Návrh

Pomocou LinkedIn API a s využitím autorizačného mechanizmu OAuth spolu s externým gemom LinkedIn zabezpečíme prístup k dátam jednotlivých používateľov. Na tento účel bola na sociálnej sieti LinkedIn vytvorená aplikácia Pinta.sk, s nastavením potrebných oprávnení.

Implementácia a testovanie

V samotnej aplikácii je funkcionality implementovaná pomocou `linkedin_controller` modulu, ktorý má za úlohu komunikovať s LinkedIn API. Súčasťou návrhu je dátový model pre spomínané entity. Je ho možné vidieť na obrázku x. kde je zobrazený aktuálny celkový dátový model aplikácie. Implementovala sa teda funkcionality prepojenia používateľského účtu našej aplikácie s existujúcim účtom na sociálnej sieti LinkedIn. Ďalej bola implementovaná funkcionality aktualizovania profilu pomocou týchto dát. Funkcionality výberu špecifických atribútov sa z časového dôvodu počas šprintu odobrala z backlogu a bude implementovaná v najbližších šprintoch.

3.5.6 Vytvorenie achievement systému

Autor: Bc. Miroslav Vojtuš

Analýza

Za účelom motivácie a zároveň sledovania stavu používateľa sme zvažili ako vhodnú metódu pridelenia statusov na základe dosiahnutých úspechov, vo hernom svete sa nazýva takýto status achievement.

Pre udržanie používateľa v systéme a zabezpečenie pocitu uspokojenia dostáva priebežne po vykonaní akcií nové statusy, za ktoré môže byť v budúcnosti odmenený aj vecne. V systéme môžeme požívať tieto statusy na filtrovanie pracovníkov.

Taktiež slúži na vedenie používateľa systémom. Ukazuje možnosti, ktoré systém ponúka. Časom sa môže báza achievement-ov rozširovať. Základne achivementy sú opísané v nasledujúcej tabulke:

názov	podmienka	viacnásobný?
apprentice-requester	viac ako 1 zadaná úloha	NIE
journeyman-requester	viac ako 10 zadaných úloh	NIE
expert-requester	viac ako 100 zadaných úloh	NIE
artisan-requester	viac ako 1000 zadaných úloh	NIE
apprentice-worker	viac ako 1 splnená úloha	NIE
journeyman-worker	viac ako 10 splnených úloh	NIE
expert-worker	viac ako 100 splnených úloh	NIE
artisan-worker	viac ako 1000 splnených úloh	NIE
identified-user	vyplnené základné polia v profile	NIE

Špecifikácia požiadaviek

Funkcionálne:

- zobrazenie zoznamu dostupných achievement-ov
- zobrazenie o získaní nového achievement-u

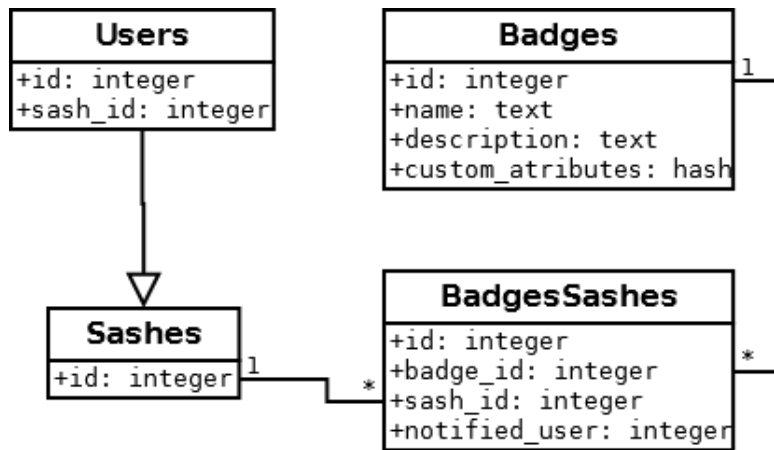
Nefunkcionálne:

- rozšíriteľnosť - musí byť jednoduché pridať nový achievement
- rýchlosť - vyhodnotenie achievement-ov nesmie zásadne ovplyvňovať chod systému
- použiteľnosť - nie je to zásadná súčasť systému a achievement-y by nemali používateľa rozptyľovať od úloh. Zobrazenie informácie musí byť prívetivou cestou.

Implementácia a testovanie

Pre implementáciu bol použitý rails gem merit. Ktorý poskytuje základnú funkcionálnu získavania achievement-ov pod názvom badge. Zároveň ponúka možnosti pridelovania bodov prípadne iných motivačných bonusov.

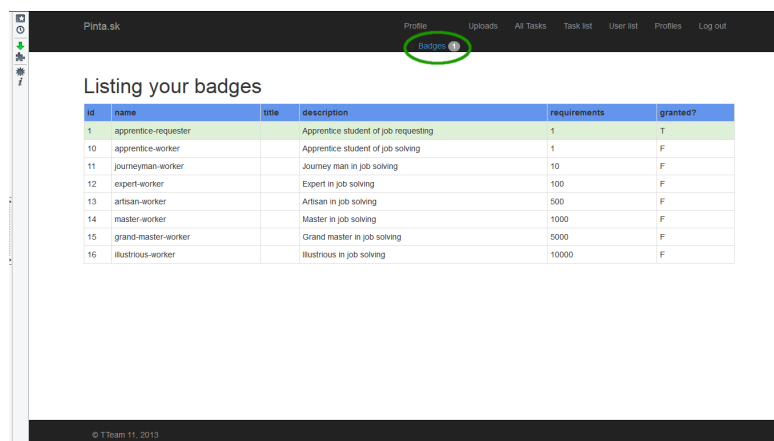
Dátový model pre achievement-y:



Obr. 13: Dátový model pre achievmenty

Všetky dátové entity okrem Badges sú uložené v databáze. Badges sú definované priamo v zdrojových kódach. Logika pridelenia achievmentu je viazaná na Badges.name. Nakolko je ruby interpretovaný jazyk, nie je problém pridať nový achievment spolu s logikou za behu systému.

Používateľské rozhranie:



Obr. 14: Používateľské rozhranie

Používateľské rozhranie bolo implementované s ohľadom na požiadavku použiteľnosti. Informácia o novom achievment-e je zobrazená ako jednoduché počítadlo pri navigačnom uzle prislúchajúcom achievment-om. Po rozkliknutí používateľ vidí stav dsotupných achivement-ov.

3.6 Šprint #5

3.6.1 Vytváranie dokumentácie

Autor: Všetci členovia

Všetci členovia sa podieľali na vytváraní a dopĺňaní dokumentácie k riadeniu, aj k inžinierskemu dielu.

3.6.2 Facebook lajky a komentáre

Autor: Bc. Patrik Štrba

V tomto šprinte bolo potrebné spraviť integráciu sociálnych prvkov pre náš projekt. Na základe tímového stretnutia sme sa rozhodli integrovať prvky zo sociálnej siete facebook a to konkrétne tlačidlo “páči sa mi to” a komentáre. Najskôr bolo potrebné vytvoriť aplikáciu na facebooku, pomocou ktorej sa následne integrovali prvky na stránku. Následne sme tieto prvky sme analyzovali a navrhli dizajn a štruktúru čo by bolo pre nás výhodné. Následne sme tieto prvky integrovali do projektu takým spôsobom, aby bolo možné tieto prvky používať aj v budúcnosti jednoduchším spôsobom.

3.6.3 Integrácia s Google účtom

Autor: Bc. Marek Lenčes

Analýza

Dnes je už vo všetkých významných webových aplikáciách možnosť integrácie účtu webovej aplikácie s inými účtami, napríklad sociálnymi sieťami, z ktorých je možné získať informácie o používateľovi a zjednodušiť mu tak vyplnenie jeho profilu. Významné postavenie na internete majú služby firmy Google, ktoré sú využívané veľkým množstvom používateľov, a sú teda vhodné na získanie údajov o prihlásenom používateľovi.

Špecifikácia požiadaviek

Keďže primárnym cieľom integrácie našej aplikácie s účtom Google je získanie údajov, musíme sa zamerať na údaje, ktoré nás zaujímajú v našej aplikácii, predovšetkým su to osobné údaje o používateľovi, meno, dátum narodenia, profilová fotka. Neskôr možno využijeme ďalšie informácie, ktoré služby Google ponúkajú.

Návrh

Integráciu s účtom Google bude potrebné vytvoriť pomocou nejakého rozhrania, ktoré ďalej umožní získavanie údajov zo služieb Google.

Implementácia a testovanie

Implementácia je vykonaná pomocou rozhrania OAuth, ktoré zabezpečí spárovanie účtu našej webovej aplikácie a účtu Google. Ďalej je možné získať želané údaje, ktoré využijeme pri naplnení profilu používateľa.

3.6.4 Integrácia s Twitter účtom

Autor: Bc. Miroslav Molnár Pre získanie ďalších informácií o používateľovi a možnosť v budúcnosti komunikovať zo sociálnou sieťou, sa navrhla integrácia s používateľským účtom na Twitteri. Ďalšou výhodou je možnosť prihlasovať používateľov cez ich existujúce konto na Twitteri, posielanie správ na sociálnu sieť, alebo sledovať činnosť používateľa. Prepojenie s touto sociálnou sieťou nám umožňuje rýchlejšie rozšírenie našej aplikácie medzi súčasnou generáciou mladých ľudí. Čím viac používateľov našu aplikáciu bude používať, tým bude použiteľnejšia pre všetkých používateľov.

3.7 Šprint #6

3.7.1 Vytvorenie viackrokových formulárov

Autor: Bc. Filip Bednárík

Pri type úlohy Survey (dotazník) sme prišli na to, že potrebujeme viac krokov na to aby sme korektné vedeli zobrazit grafické rozhranie pridávania podúloh pre zadávateľa. Preto sme v tomto šprinte implementovali tzv. Wizard formy a tzv. Nested formy.

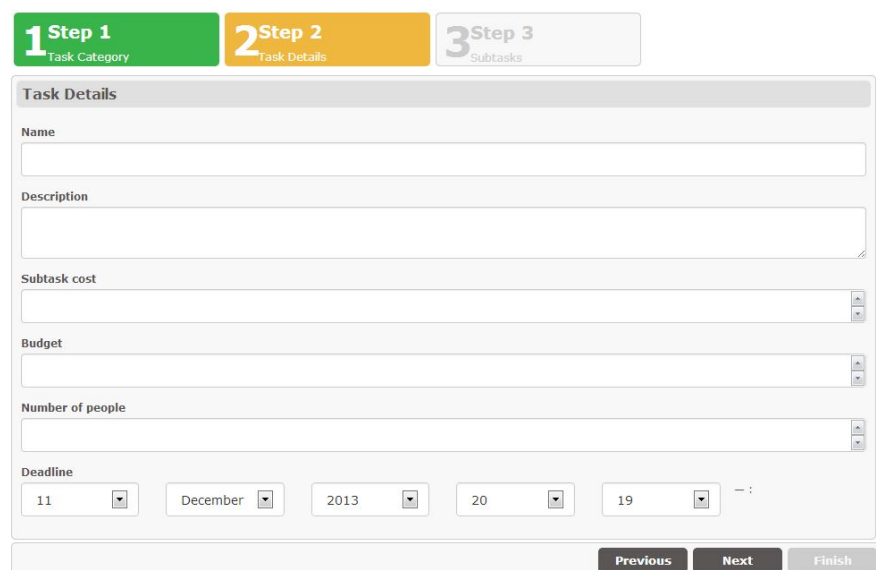
Pri každom kroku sa posielajú údaje z formulára na server bez znovunačítania stránky prehliadačom, čo nám dovoľuje použiť pekný efekt posunu formuláru doľava a tiež spríjemňuje prácu s formulárom zadávateľovi. Pri každom načítaní kroku sa berú do úvahy dáta prijaté pri kliknutí na tlačidlo “Next”.

Najprv zadávateľ vyberie kategóriu úlohy podľa čoho sa mu neskôr v kroku 3 zobrazí korektný formulár pre vytvorenie podúloh pre daný typ úlohy.



Obr. 15: Formulár zadávania úlohy

V kroku 2 zadávateľ zadá základné informácie o úlohe. Zadávateľ sa môže posúvať vo formulári aj klávesami doľava a doprava.



Obr. 16: Formulár zadávania úlohy

Tretí krok bol najťažší z implementačného hľadiska. Ide totiž o vhniedený for-

mulár obsahujúci n záznamov do tabuliek podúloh viažucich sa na hlavnú entitu úlohy. Zadávatel má možnosť bez znovunačítania stránky jednoducho pridávať nové podúlohy, mazať ich alebo ich upravovať.

The screenshot shows a three-step process for task management. Step 1 is 'Task Category', Step 2 is 'Task Details', and Step 3 is 'Subtasks'. The 'Subtasks' section is active and contains a 'Questions' table with two rows of input fields and 'Remove' buttons. Below the table is an 'Add Question' button and navigation buttons 'Previous', 'Next', and 'Finish'.

Obr. 17: Formulár zadávania úlohy

3.7.2 Rebranding - web tímu, dizajn

Autor: Bc. Martin Toma, Bc. Marek Lenčes

V rámci rebrandingu sme vytvorili nové logo, ktoré zároveň reprezentuje naše nové pomenovanie Pinta.sk. S tým súvisí aj rekonštrukcia tímového webu, ktorý už nevyhovoval novým požiadavkám. Webu sme dali nový dynamický responzívny dizajn, založený na šablóne Bootstrap, ukážku nového webu môžeme vidieť na obr



Obr. 18:

Rovnako sme navrhli a vytvorili nový obsah, ktorý prezentuje náš projekt. Snažili sme sa použiť pokročilejšie copywriting techniky spojené s technikami SEO. Výsledkom je teda nová identita produktu a rovnako aj tímu 11, ktorý od tohto šprintu nesie oficiálne meno Pinta Group a produkt Pinta.sk.

3.7.3 Zlepšená organizácia profilu

Autor: Bc. Miroslav Vojtuš

Analýza

Aktuálna verzia profilu je postačujúca pre vývojové účely. Avšak jej vzhľad a štruktúra nezodpovedá doterajšiemu designu. Je potrebné doplniť možnosť manuálne vyplňať polia v profile, používateľsky prívetivou cestou.

S pohľadu použiteľnosti existujú dve základné cesty:

- synchronný prístup
 - každá akcia jeden kompletný request s návratom kompletného pohľadu
 - ľahké na implementáciu - rails ponúka generátory CRUD operácií
- asynchrónny prístup
 - každá akcia jeden čiastočný dopyt na server s návratom len čístej obnovenej pohľadu
 - náročnejšia implementácia, vyžadujúca viac vlastného kódu
 - ťažšie z pohľadu udržateľnosti

Vzhľadom k tomu, že vyvíjame moderný webový systém, najvhodnejšou variantou je asynchrónny prístup, ktorý vša nesie nevýhodu ťažšej implementácie, náročnejšej na údržbu.

Špecifikácia

Funkcionálne požiadavky:

- zobrazenie profilu v štruktúrovanej forme
- editovanie jednotlivých kategórií
- pridanie novej položky do dynamických kategórií (zoznam)
- odstránenie položky z dynamických štruktúr
- iné operácie ako je čítanie povoliť len majiteľovi profilu

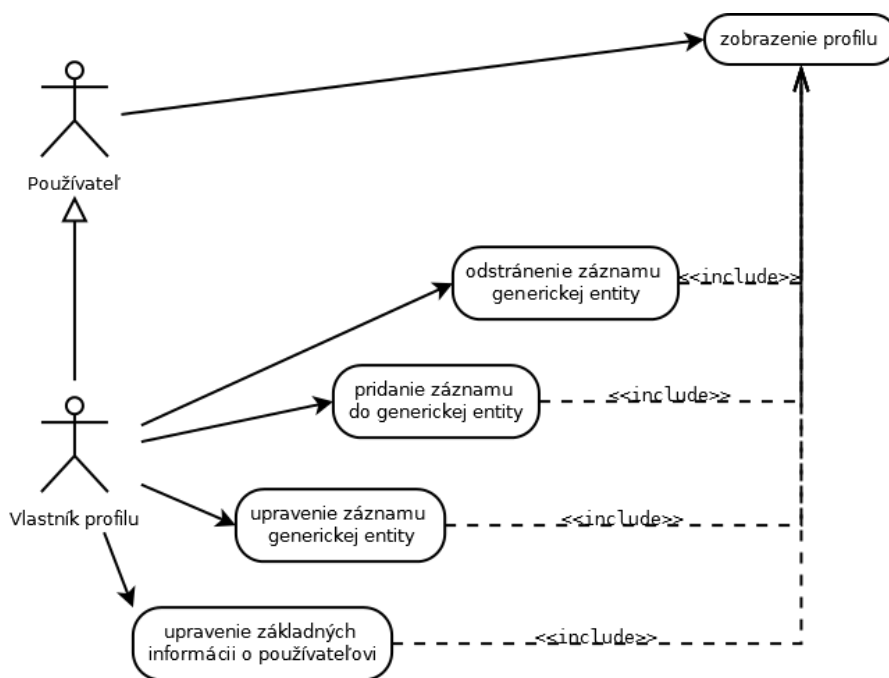
Nefunkcionálne požiadavky:

- grafické prevedenie zladené so štýlom systému
- dostupnosť - minimalizovať čas odozvy systému medzi akciami
- použiteľnosť - jednotlivé akcie musia byť jasne viditeľné

Návrh

Kategórie zostanú zachované podľa dátových entít:

- všeobecné informácie (and. Basic info)
- vzdelanie (and. Education)
- skúsenosti (and. Skills)
- jazyky (and. Languages)



Obr. 19: Diagram prípadov použitia

- zamestnania (ang. Positions)

Dátový model nie je ovplyvnený, zachovaná štruktúra informácií.

Diagram prípadov použitia: Diagram opisuje základné funkcionálne prvky profilu. Generická entita je jedna z dopredne určených entít s dynamickým obsahom. Základné sú odvodené vyššie. Úpravy profilu vyžadujú zobrazenie profilu. Profil si môže zobraziť ktokoľvek ale editovať ho môže len vlastník profilu.

Pre zabezpečenie požiadavky štruktúrovanosti je profil rozdelený do dvoch stĺpcov. Jednotlivé kategórie majú vlastný panel s názvom a informačným obsahom. Podľa typu uložených informácií sa tlačítko s akciou pre úpravu bude nachádzať, buď pri celej kategórii alebo jednotlivej položke. Týmto je zabezpečená požiadavka použiteľnosti - v prípade veľkého množstva záznamov používateľ edituje len ten, ktorý potrebuje.

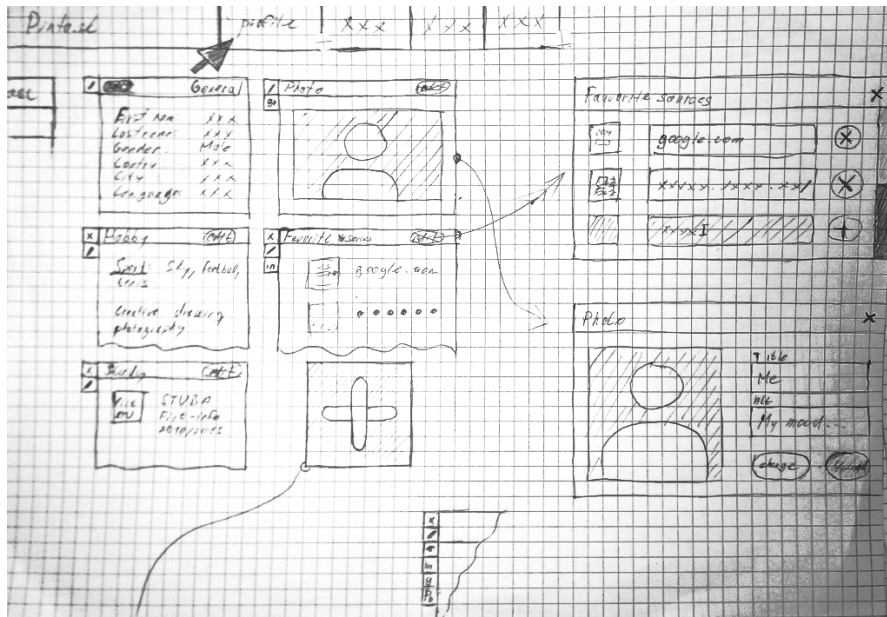
Low fidelity návrh rozloženia funkcionálnych prvkov:

Implementácia a testovanie

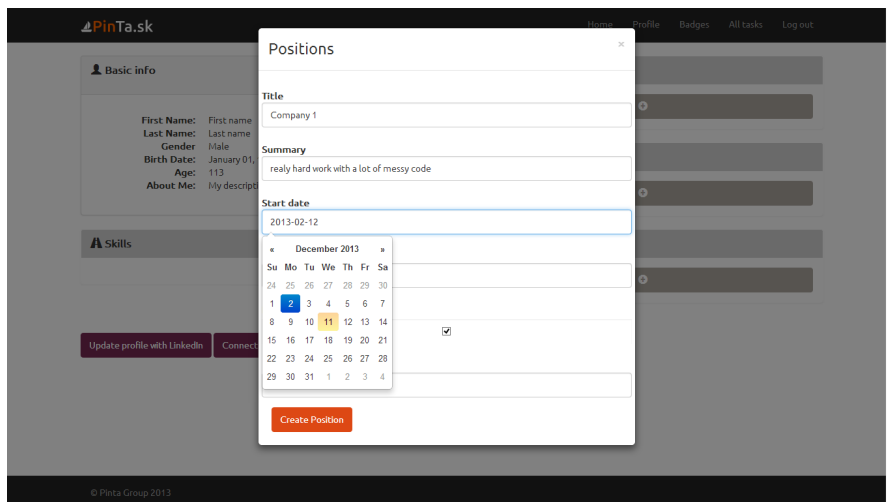
V implementácii je kladený dôraz na použiteľnosť a dostupnosť. Zvolili sme asynchrónnu metódu. Základná štruktúra zodpovedá pôvodnému low fidelity návrhu. Z dôvodu vyššej komplexity niektorých prvkov a nízkej priority boli v implementácii opomenuté. Zamerali sme sa na hlavné funkcionálne prvky a ostatné sme odsunuli do neskorších fáz vývoja.

Rozhranie modálneho okna je zobrazené na nasledujúcom obrázku. Obsah je získaný asynchrónnou požiadavkou na server. Formulár je štandardný pre rails-ovské aplikácie s podporou asynchrónneho odovzdania dát. Modálne okno je ako komponent abstrahované pre ďalšie použitie v aplikácii.

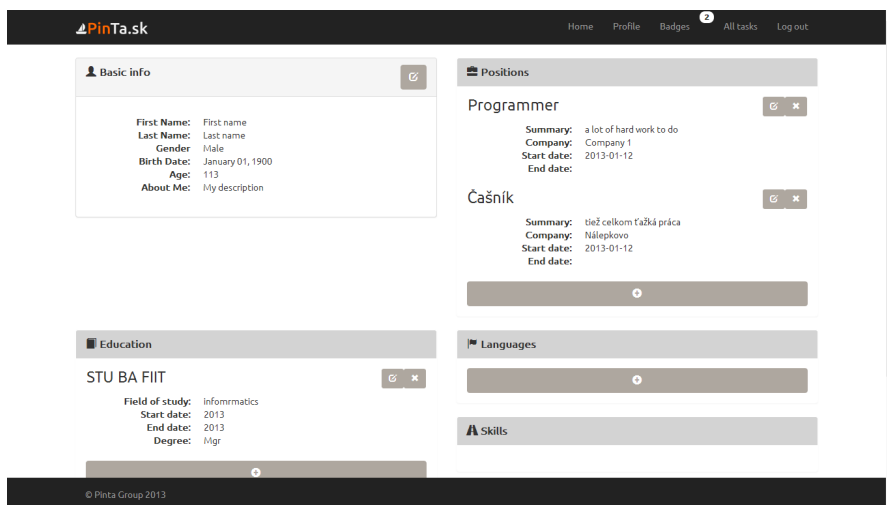
Výsledná štruktúra profilu je zobrazená na nasledujúcom obrázku. Rozdiel medzi pohľadom na cudzí profil a vlastný je v absencii ovládacích prvkov pre pridanie a úpravu odstránenie informácií.



Obr. 20: Low fidelity návrh



Obr. 21: Modálne okno



Obr. 22: Návrh nového profilu

Na základe požiadavky zachovať štýl celého systému, boli použité triedy štýlov definované použitým GUI frameworkom Bootstrap.

Z dôvodu náročnosti automatizovaného testovania asynchrónneho vykonávania boli jednotlivé prvky testované len manuálne.

3.8 Šprint #7

3.8.1 Zarovnanie dizajnu produktovej aplikácie s novou identitou tímu

Autor: Bc. Martin Toma

V minulom týždni prebehol rebranding a navrhli sme produktové logo, tímové meno a hlavné produktové farby. Keďže aplikácia mala v tej dobe iba mierne upravený základný bootstrap dizajn, rozhodli sme sa zaviesť miernu konzistenciu aj v tomto ohľade.

Vytvoril sa teda nový CSS dizajn a použili sa pokročilejšie bootstrap techniky a hlavne sa aplikovali farby a fonty navrhnuté v minulom šprinte pri rebrandingu.

3.8.2 Survey task

Autor: Bc. Filip Bednárík

Ďalším user story, ktorý sme riešili bol user story v ktorom zadávateľ vyžadoval od pracovníkov vyplňovať dotazníky. Preto sme v tomto šprinte implementovali novú kategóriu úloh a tou je Survey (dotazník). Zmenili sme atribúty podúloh a to tak, že sme pridali typ vstupu a typ výstupu. Typ vstupu môže byť v súčasnosti:

- Obrázok z URL
- Text
- Facebook stránka

Dostupné typy výstupu:

- Text
- Číslo
- Možnosti (oddelené čiarkou)

Zadávateľ má možnosť vyskladať si vlastný dotazník z týchto elementárnych otázok. V ďalších šprintoch som sa venoval zmene GUI CSS atď. Searchbox som spravil aby fungoval AJAXovo spolu s willpaginate.

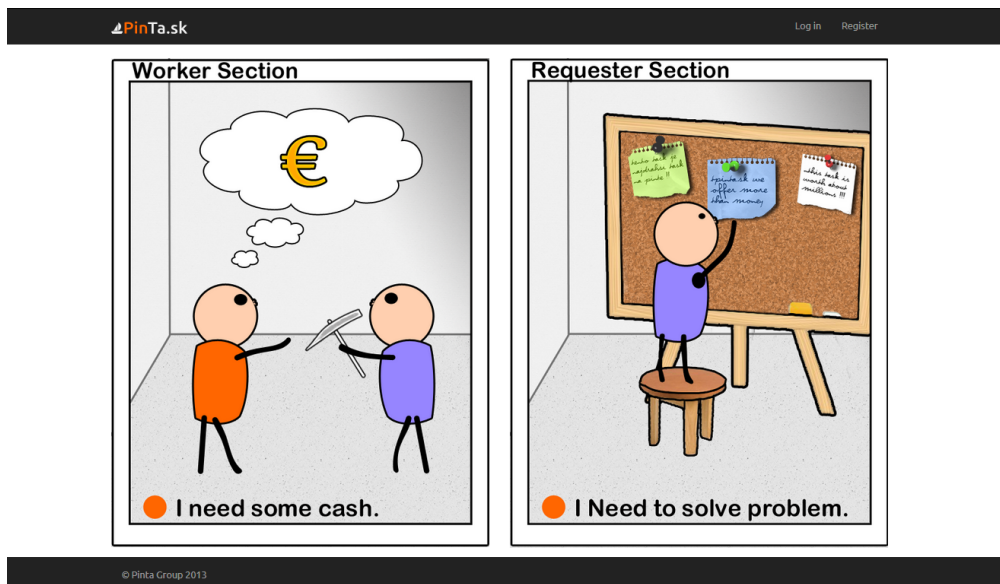
3.8.3 Vytvorenie grafických návodov pre landing page

Autor: Bc. Marek Lenčes

Na úvodnej stránke aplikácie sa pri návšteve objaví rozcestník pre pracovníka, alebo pre zadávateľa úlohy, podľa výberu sa mu zobrazí jednoduchý obrázkový návod, ako asi funguje vypracovanie/zadanie úloh v systéme Pinta.sk.

3.8.4 Integrácia so sociálnou sieťou Twitter

Autor: Bc. Miroslav Molnár



Obr. 23: Výber sekcie na landing page

Analýza

Pre možnosť získania viacerých zdrojov informácií o používateľovi nášho systému sme vytvorili integráciu so sociálnou sieťou Twitter. Druhým významom je biznis logika, pomocou ktorej sa snažíme získať väčšie množstvo používateľov. Čím viac používateľov systém bude používať, tým bude väčšia pravdepodobnosť získania kvalitnejších výsledkov a väčšie množstvo úloh.

Špecifikácia požadaviek

Vytvorenie integrácie používateľa s jeho účtom na sociálnej sieti Twitter. Používateľ v profile pomocou tlačidla bude môcť sa zintegrovat' so sociálnou sieťou. Po úspešnom prepojení s účtom na Twittri, bude môcť používateľ pokračovať v činnosti na webovej stránke projektu.

Návrh

Po dôkladnom preštudovaní možností prepojenia so sociálnou sieťou treba vybrať tú najbezpečnejšiu pre používateľa a zároveň najefektívnejšiu pre náš projekt. Je potrebné dbať na obmedzujúcu podmienku času, vyhradeného na vývoj. Vytvorené tlačidlo v profile webovej aplikácie projektu po stlačení presmeruje používateľa na stránku sociálnej siete Twitter. Tu sa prihlási pomocou svojich prihlasovacích údajov, ktoré má na sociálnej sieti. Po úspešnej registrácii, bude používateľ presmerovaný na webovú stránku v profile, kde bude uvedomený o úspešnej, alebo neúspešnej integrácii s účtom.

Pre umožnenie integrácie používateľa s našou webovou stránkou projektu je nevyhnutné na stránke Twittru <https://dev.twitter.com/>, registrovať našu aplikáciu. Pre registráciu je nevyhnutné mať účet na sociálnej sieti Twitter. Po úspešnej registrácii aplikácie, budú vygenerované špeciálne hash kľúče slúžiace na identifikáciu pri komunikácii s Twittrom

Implementácia a testovanie

Do profilu bolo doimplementované tlačidlo s funkcionalitou pre komunikáciu s Twittrom. Zvolili sa OAuth a Twitter gemy. Pomocou týchto gemov sa zaobstaráva komunikácia so sociálnou sieťou. Pre naviazanie komunikácie sa využívajú hash kľúče vygenerované pri registrácii našej aplikácii na stránke Twitter. Po naviazaní komunikácie nám Twitter pošle request token, z ktorého si získame dáta potrebné na presmerovanie používateľa z našej stránky na stránku Twitter. Po úspešnom prihlásení presmeruje používateľa na našu stránku. V parametri URL, server Twittru, odovzdá vygenerované hash kľúče integrujúceho používateľa. Tieto kľúče uložíme do tabuľky profilu používateľa. Bola implementovaná metóda `show_twitter_profile` v kontroleri profilu, ktorá využije hash kľúče používateľa už integrovaného s účtom na Twittri a získa dáta z jeho profilu. Funkcia má ukazovať správnu komunikáciu so sociálnou sieťou a spracovanie prijatých údajov.

3.9 Šprint #8

3.9.1 Zmena dizajnu badges

Autor: Bc. Miroslav Molnár

Analýza

Jednou z možností ako motivovať pracovníka a zadávateľa ku kvalitnejšej práci v našom systéme je dostávanie badges. Badges sa dostávajú za špeciálne procesy splnené v systéme. Takýto štýl motivácie je veľmi prínosný a lákavý pre používateľov. Pre jeho zlepšenie bola navrhnutá jeho grafická úprava.

Špecifikácia požiadaviek

Navrhňte lepšiu grafickú reprezentáciu badges v systéme. Použiť už implementovaný zdrojový kód. Navrhňte ďalšie možné typy badges.

Návrh

Pre reprezentáciu badges bola navrhnutá grafická podoba v tvare guľatých lesklých odznakov, farebne rozlíšených pre pracovníka a zadávateľa. Pri nedosiahnutí badges sa zobrazia tieto grafické podobizne čiernobielo Obr.1 V pôvodnom návrhu bola tabuľka so všetkými badges a informáciami o nich. Tento typ bol pre používateľa hneď po prvom zobrazení veľmi chaotický. Prílišné množstvo informácií spôsobovalo negatívnu dezorientáciu.

Z tohto dôvodu bol význam dát prehodnotený a nepotrebné informácie sa nezobrazujú používateľovi, ako je napríklad ID badges. Ostatné informácie sa zobrazia až keď si ich používateľ vyžiada. Týmto prístupom sa stalo grafické rozhranie intuitívnejšie a príjemnejšie pre používateľa. Príklad už upraveného užívateľského rozhrania môžeme vidieť na Obrázku 25

Celková reprezentácia dát ostala v tabuľke. Bola však vytvorená nová šablóna „.css“ súboru, ktorá nemá zbytočné čiary a tak je voľnejšia. Pre lepšiu čitateľnosť sa riadky farebne odlišujú. V tabuľke sa zobrazuje iba názov badges, pre bližšie informácie používateľ musí kliknúť na tlačidlo nachádzajúce sa napravo. Po stlačení tlačidla sa zobrazí modálne okno so všetkými informáciami o konkrétnom badge.



Obr. 24: Zobrazenie grafickej reprezentácie badges

Implementácia

Reprezentácia obrázkov bola nakreslená a zmodifikovaná do rôznych farieb. Modálne okno bolo prevzaté z profilu a upravené pre potreby použitia. Boli doimplementované potrebné preklady jazykov. Bol doplnený parameter worker typu boolean pre detekciu či sa jedná o badge pre zadávateľa, alebo pracovníka. Výsledná schéma bola otestovaná.

3.9.2 Vytváranie záznamov z externej aplikácie

Autor: Bc. Filip Bednárík

Analýza

Pri user story, v ktorom zadávateľ vytvorí úlohu, kde vyžaduje od pracovníka nainštalovanie nejakej aplikácie, ktorú vyvinul a jej používanie je vhodné aby mal možnosť zadávateľ sledovať pracovníkovu aktivitu v jeho aplikácii, odmeňovať ho za používanie, prípadne mu prístupňovať nové úlohy na základe jeho aktivity.

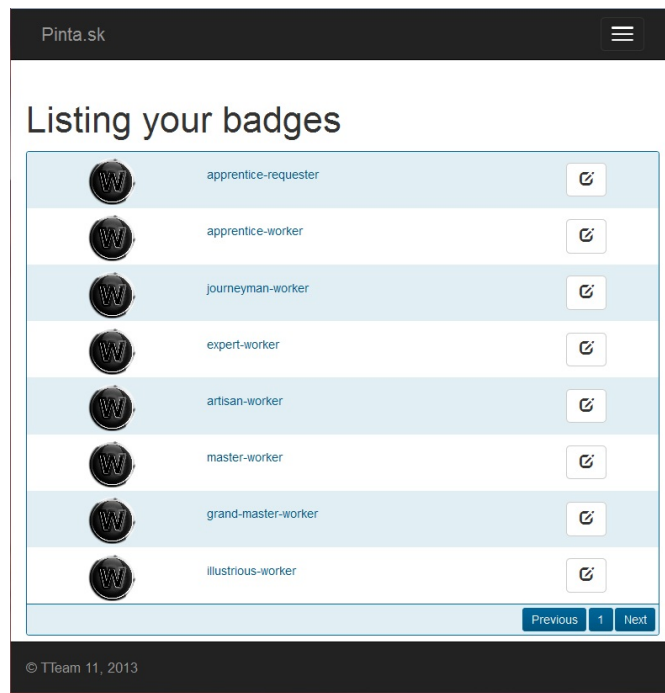
Špecifikácia požiadaviek

Pre tento účel potrebujeme API pre logovanie (vytváranie záznamov) z externej aplikácie do nášho systému. Logovanie musí prebiehať čo najjednoduchšie zavolaním URL s parametrami. V parametroch musí byť identifikovateľný zadávateľ prípadne úloha ku ktorej sa záznam vzťahuje, pracovník, ktorý používa danú aplikáciu, nejaká značka podľa ktorej si zadávateľ vie filtrovať záznamy a nakoniec samotná hodnota záznamu.

Návrh

Navrhli sme jednoduchý CRUD, kde môže zadávateľ svoje záznamy spravovať. Taktiež sme navrhli formát adresy ktorú zadávateľ zavolá zo svojej externej aplikácie pre vytvorenie záznamu.

`http://www.pinta.sk/log/*API-KEY*/*USER-TAG*/*TAG*/*VALUE*`



Obr. 25: Zobrazenie stránky s badges

Listing Logs - API Key: 31b8face-08c7-4845-9ef7-b01c51816c0d

Tag	Value	User Tag	Actions
tag1	value	drndos	Edit Destroy
TaG	sadhsadhasdsad	drndos	Edit Destroy
TaG	sadhsadhasdsad	drndos	Edit Destroy
TaG	sadhsadhasdsad	drndos	Edit Destroy

New Log

Obr. 26: CRUD

- ***API-KEY*** je jedinečný kľúč zadávateľa do nášho systému, ktorý sa mu vygeneruje akonáhle prisúpy na stránku záznamov.
- ***USER-TAG*** je identifikátor používateľa systému
- ***TAG*** je značka, podľa ktorej si vie zadávateľ záznamy filtrovať
- ***VALUE*** je samotná hodnota záznamu

Implementácia a testovanie

Pri implementácii sme využili generátor RoR pre CRUD. Nastavili sme routes tak aby pri navštívení URL daného formátu sa vyvolal controller pre vytvorenie nového záznamu z URL.

Pri generovaní API sme využili generátor jedinečného kľúča poskytovaný frameworkom.

3.9.3 Úloha s pridávaním tzv. Facebook lajkov

Autor: Bc. Róbert Černý

Analýza

Jednou z najnavštevovanejších sociálnych sietí v súčasnosti je Facebook. Okrem regulárnych používateľov sa na tejto sieti nachádzajú aj stránky, ktoré prezentujú spoločnosti, významné osobnosti, médiá a iné skupiny, ktoré bojujú o zviditeľnenie a popularitu. Tá sa meria počtom hlasov (tzv. lajk). Automatizovanie procesu pridávania hlasov k želaným stránkam na facebooku je ďalšou možnosťou ktorú náš systém poskytne. Nakoľko je Facebook rozšírený, predpokladá sa, že táto úloha bude veľmi obľúbená nakoľko je nenáročná a jednoduchá.

Špecifikácia požiadaviek

Vytvorenie novej kategórie úlohy Facebook lajk, prostredníctvom ktorej automatizujeme proces hlasovania na sociálnej sieti Facebook. Tento proces vyžaduje od pracovníka iba jednoduché kliknutie na tlačidlo lajku konkrétnej stránky.

Návrh

Zadávateľ úlohy poskytne zoznam URL stránok pre lajkovanie. Zobrazenie úlohy pre pracovníka je veľmi jednoduché. Jednotlivé URL sú zobrazené v prehľadnej tabuľke pričom je ku každej priradené tlačidlo pre hlasovanie. Pracovník vždy vie, či za danú stránku už hlasoval. Táto úloha prebieha výlučne na našej stránke, nieje teda potrebné navštíviť jednotlivu každú URL zo zoznamu. Pre implementovanie tejto úlohy je nutné priradiť pracovníkovi autorizačný token, aby bolo jasné, kto hlasuje. Túto funkcionality rieši integrácia so sociálnou sieťou Facebook (kapitola ...). Po obdržaní autorizácie s príslušnými privilégiami máme možnosť používať Graph API, ktoré poskytuje facebook. Pomocou tohto API dokážeme dopytovať servery sociálnej siete a získavať tak dáta o pracovníkovi, ktorý práve hlasuje. Pre optimalizáciu a rýchlosť budeme využívať FQL (Facebook Query Language). Úloha skončí úspešne, ak bude každá zo stránok lajknutá.

Implementácia a testovanie

Pre úlohu vytvoríme možnosť novej kategórie “facebook_like”. Pri implementácii formulára pre novú úlohu vytvoríme možnosť pre vkladanie jednotlivých URL, čo bude jediná špecifická povinnosť zadávateľa úlohy. Príkladom takejto URL je: <https://www.facebook.com/amazing.weird>.

Jednotlivé tlačidlá pre lajkovanie je možné implementovať pomocou jednoduchého HTML kódu kde premenná “url” reprezentuje URL stránky.

```
1 <div class="fb-like "  
2   data-href="<%= url %>"  
3   data-layout="button_count "  
4   data-action="like "  
5   data-show-faces="false "  
6   data-share="false ">  
7 </div>
```

Metóda “pages_liked?” kontroluje, či boli všetky stránky lajknuté.

Výsledkom bude nová inštancia triedy “solution”, ktorá má v atribúte “data” celkový počet lajknutých stránok nakoľko iný výstup úloha nepožaduje.

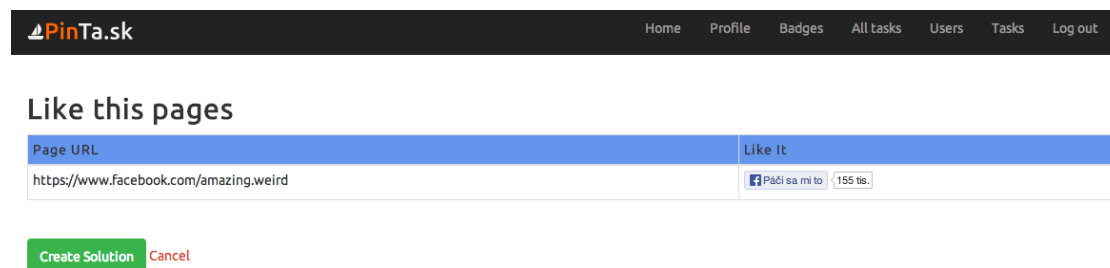
Chyby ktoré môžu v tomto procese nastať zachytávame pomocou výnimiek, ktoré ošetrujeme. Medzi ne patria:

1. pracovník nieje prepojený s facebookom, teda nemá autorizačný token
2. niektorá zo stránok neexistuje
3. pracovník nelajkol všetky stránky

Takýmto jednoduchým dopytom zistíme, či pracovník lajkol stránku.

```
SELECT page_id FROM page_fan WHERE uid=me() AND page_id=page_act['id']
```

Vyberáme teda identifikátor stránky z tabuľky “page_fan” ktorá patrí aktuálnemu pracovníkovi. Musí platiť podmienka, že identifikátor stránky z tabuľky sa zhoduje s identifikátorom stránky z úlohy. V prípade, že pracovník lajkol stránku, výsledný dopyt vráti hodnotu identifikátora stránky. Tento proces opakujeme pre každú stránku z čoho vieme určiť správnosť vypracovania úlohy.



Obr. 27: Lajkovanie stránok

3.9.4 Intenacionalizácia (SK, ENG)

Autor: Bc. Róbert Černý

Analýza

Žiadna webová aplikácia používaná medzinárodne sa nezaobíde bez jazykových prekladov. Identifikovali sme požiadavku pre implementáciu frameworku, ktorý bude poskytovať možnosť jednoduchým spôsobom meniť jazyk aplikácie. Pre účely testovania sme sa rozhodli pre jazyky slovenčina a angličtina.

Špecifikácia

Základnou požiadavkou bolo, sprístupniť používateľovi možnosť meniť jazyk aplikácie. Táto funkcionality sa bude nachádzať priamo na úvodnej stránke po prihlásení. Kliknutím na zástavku sa automaticky zmení jazyk bez nutnosti akéhokoľvek iného úkonu. Požiadavkou samotného vývojového tímu bolo zdefinovať čo najjednoduchší spôsob ako pri programovaní definovať reťazec znakov určený na preklad. Najefektívnejším spôsobom bude používať volanie metódy. Príkladom definovania takéhoto reťazca je: `f('tento text sa prekladá')`. Takýmto spôsobom ušetríme čas a zefektívime proces vývoja.

Implementácia a testovanie

Súčasťou Ruby on Rails je API L18n, ktoré nám dovoľuje jednoducho a efektívne meniť jazyk. Jazyk je vhodné nastaviť v súbore `application_controller.rb` z ktorého dedia všetky ostatné kontroleri. Samotný jazykový preklad sa nachádza v súboroch `en.yml` a `sk.yml` so špecifickou syntaxou. Výraz "Description" by sme zadefinovali takto:

V súbore `en.yml` pridali:

```
1 description: Description
2           sk.yml pridali
3 description: Popis
```

V zdrojovom kóde stačí je možné použiť túto syntax a jazykové API automaticky substituujú výraz za správnu hodnotu, podľa nastaveného jazyka `<%=t "description"%>`

Zmena jazyka prebieha pomocou kontrolera `locale.rb`, ktorý umožňuje zmenu jazyka po kliknutí na zástavku.



Obr. 28:



Obr. 29:

3.9.5 Zobrazovanie výsledkov

Autor: Bc. Róbert Černý

Analýza

Zobrazovanie a vyhodnocovanie výsledkov pracovníkov je dôležitou súčasťou systému pre sprostredkovanie práce. Zadávateľia úloh prichádzajú do systému aby obdržali relevantné výstupy od pracovníkov. Tieto údaje budú prístupné zadávateľovi úlohy, avšak okrem toho je potrebné ich štatisticky vyhodnocovať. Zadávateľovi tak pribudne možnosť prezerat výsledky svojich úloh. Prvou úlohou je vytvoriť stránku s diagramom výsledkov pre úlohu dotazník, ktorá bude prístupná zadávateľovi danej úlohy.

Špecifikácia

Pre dotazník sa vyhodnotí každá otázka oddelene. Znamená to, že na stránke pre vyhodnotenie bude rovnaký počet diagramov ako bolo otázok. Zvolíme koláčový diagram, ktorý je na tento typ výsledkov najvhodnejší. Názov diagramu bude znenie otázky a jednotlivé odpovede budú zobrazené v legende. Každú odpoveď bude znázorňovať jedna farba. Po kliknutí na úsek v diagrame sa nám zobrazia informácie o počte hlasov a percentuálnom podiele, ktoré daná odpoveď získala. Takýmto spôsobom vytvoríme jednoduchý framework pre zobrazovanie výsledkov.

Implementácia

Pri implementácii bol použitý gem Highchart, ktorý ponúka diagramy rôznych typov v graficky zaujímavom prevedení. Za účelom zobrazovania výsledkov bol vytvorený kontroler, ktorý sa bude v budúcnosti rozširovať o ďalšie štatistické zobrazenia prípadne filtrovanie výsledkov podľa atribútov pracovníkov. Diagramy je možné definovať pomocou jQuery takto:

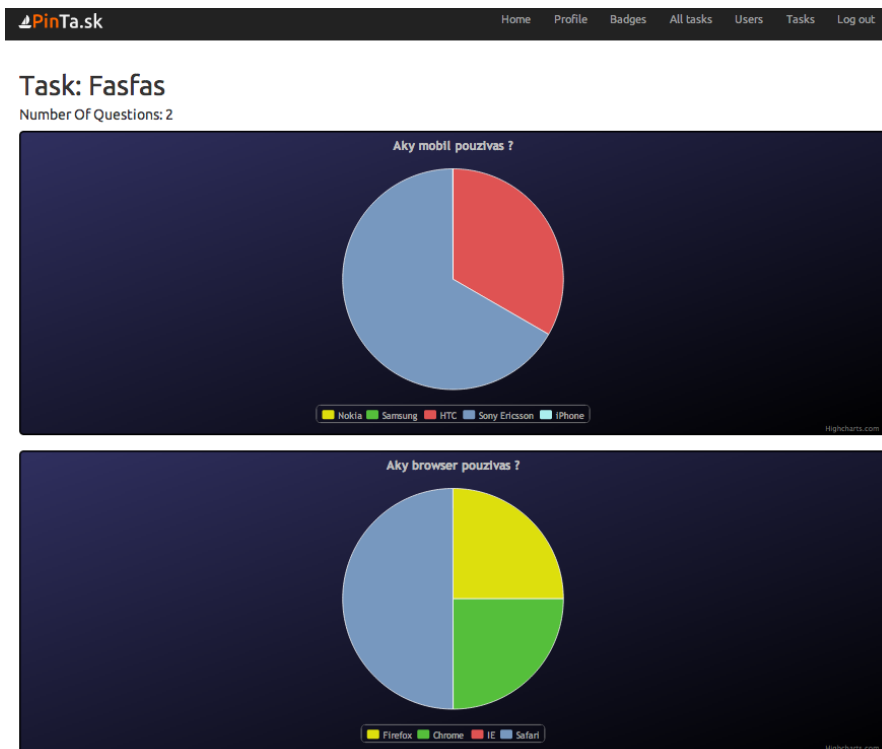
```
1 <script >
2     $(function () {
3         var chart;
4         $(document).ready(function () {
5             // Build the chart
6             $('#<%= element.__id__ %>').highcharts({
7                 chart: {},
8                 title: {},
9                 tooltip: {},
10                plotOptions: {
11                    pie: {}
12                },
13                series: [
14                    {
15                        type : ""
16                        name: ""
17                            data: []
18                    }
19                ]
20            });
21        });
22    });
23 </script >
```

V tomto skripte je potrebné doplniť typ diagramu, ktorý chceme zobraziť, znenie otázky a dataset obsahujúci dvojicu hodnôt - (odpoveď,počet). Tento dataset sa naplňa v samotnom kontrolery, kde sa pre každú podúlohu danej úlohy vytvorí pomocný dataset s informáciami o znení otázky, možných odpovediach, pričom sa pre každú odpoveď pomocou dopytu vyhľadajú hlasy a uchová sa ich počet. Výsledný dataset vyzerá nasledovne:

```
„question“=> „Aky mobil pouzivas ?“, „answers“=>[„Nokia“, „Samsung“, „HTC“, „Sony Ericsson“, „ iPhone“], „Nokia“=>0, „Samsung“=>0, „HTC“=>1, „Sony Ericsson“=>2, „ iPhone“=>0
```

```
„question“=> „Aky browser pouzivas ?“, „answers“=>[„Firefox“, „Chrome“, „IE“, „Safari“], „Firefox“=>1, „Chrome“=>1, „IE“=>0, „Safari“=>2
```

Po otvorení stránky pre výsledky úlohy sa zobrazia tieto diagramy. Údaje korešpondujú s datasetom, ktorý je spomínaný vyššie. Tento prístup zabezpečuje jednoduché pridávanie nových diagramov a štatistických zobrazení.



Obr. 30:

4 Celkový pohľad

4.1 Opis prototypu

Súčasný stav aplikácie pracuje v testovacom režime. Beta verzia aplikácie sa nachádza na www.pinta.sk a je prístupná verejnosti. Webová aplikácia má plne upravené grafické rozhranie, ktoré je prispôbené téme lode, ktorá vyplýva z názvu projektu pint. Po príchode na stránku je používateľ informovaný o účele webovej aplikácii. Aplikácia umožňuje prihlásiť sa pod testovacími heslami vytvorených, ako pre zadávateľa, tak pre pracovníka. Aplikácia pracuje v dvoch základných módoch:

- mód určený pre zadávateľa,
- mód určený pre pracovníka.

Funkcionalita módov sa líši. V móde zadávateľ používateľ môže vytvoriť novú úlohu pre pracovníka. V aplikácii sa môžu aktuálne vytvoriť nasledovné úlohy: detekcia informácií z obrázka - používateľ môže nahrať konkrétny obrázok, alebo dokáže nahrať archivačný súbor, ktorý je aplikácia schopná rozbaľiť a vybrať z neho obrázky.

Dynamické vytváranie testovacích formulárov: používateľ je schopný vytvoriť otázku a k nej odpovede. Takýmto spôsobom sa dajú vytvoriť celé testovacie zadania. Pracovník s testovacím formulárom pracuje tak, že poznačuje správne odpovede k zadaným otázkam.

Pridanie like na facebook- používateľ vloží url na facebook, pomocou ktorej pracovník sa dostane na stránku a potvrdí like.

Po vybratí typu úlohy aplikácia umožňuje zadať aj obmedzujúce podmienky pre pracovníka. Vytvorenú úlohu môžu vykonať iba pracovníci, ktorí tieto podmienky spĺňajú. Zadávatel ďalej môže určiť a odmenu pracovníka za vypracovanie úlohu.

Pracovník je takto motivovaný k lepšej a efektívnejšej práci v našej aplikácii. Aplikácia umožňuje zadávateľovi taktiež pohľad na aktuálny stav úlohy. Výsledky úlohy dokáže reprezentovať pomocou grafov.

V móde pracovník, používateľ nemá možnosť vytvárať úlohy, ale rovnako ako zadávateľ môže si úlohy prezerat. Pracovník po začatí vypracovávania úlohy môže úlohu prerušiť. Pracovníkovi sa zobrazujú iba tie úlohy, pri ktorých spĺňa podmienky zadávateľa.

Okrem odmiern za vypracovanie úlohy sa nachádza v aplikácii aj odmeňovací systém badges, ktorý oceňuje pracovníkov za špeciálne procesy splnené v systéme. Je to ďalší proces motivácie používateľov a je ním sledovaná väčšia chuť pracovať a pracovať kvalitne na svojich úlohách.

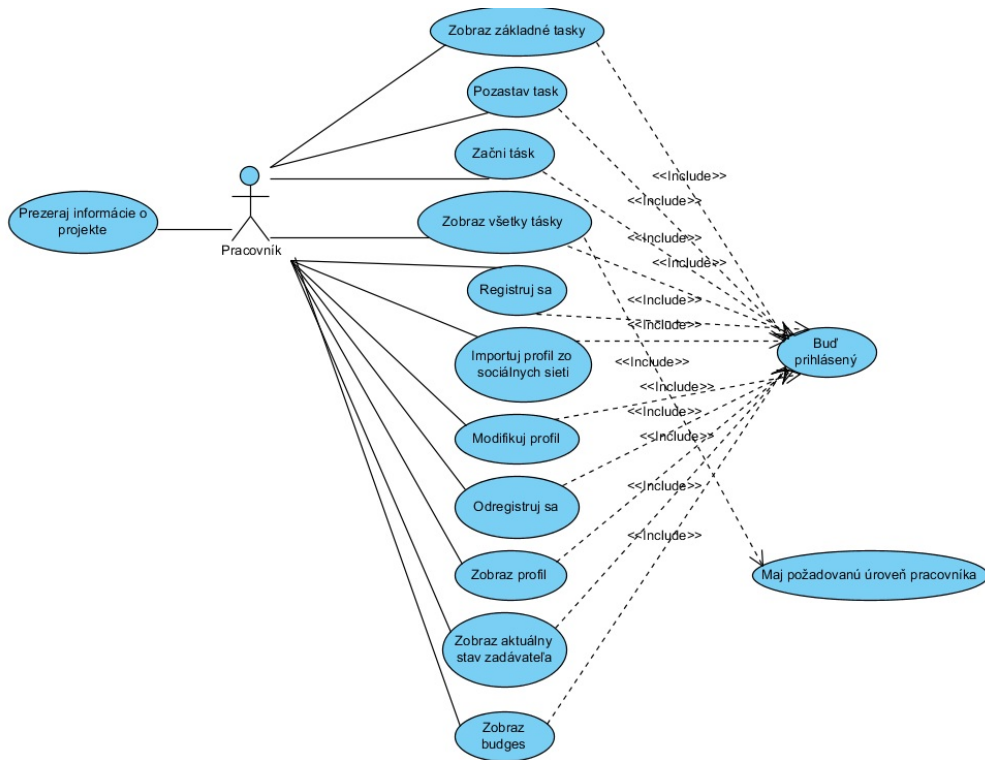
Aplikácia umožňuje používateľom pripájať sa na sociálne siete Twitter, Facebook, LinkedIn, Google. Prepojenie so sociálnymi sieťami aplikácia využíva na doplnenie profilu. Profil si používateľ vie sám upravovať a meniť podľa potreby. Je možné vyplniť krajinu, región, meno, priezvisko, prezývku, email, dátum narodenia, z ktorého sa generuje vek používateľa, jazykové zručnosti, dosiahnuté vzdelanie. Plne vyplnený profil je pre pracovníka výhodou, lebo zadávateľ môže zadať obmedzujúce podmienky na hodnoty z profilu.

4.2 Funkcionálne požiadavky

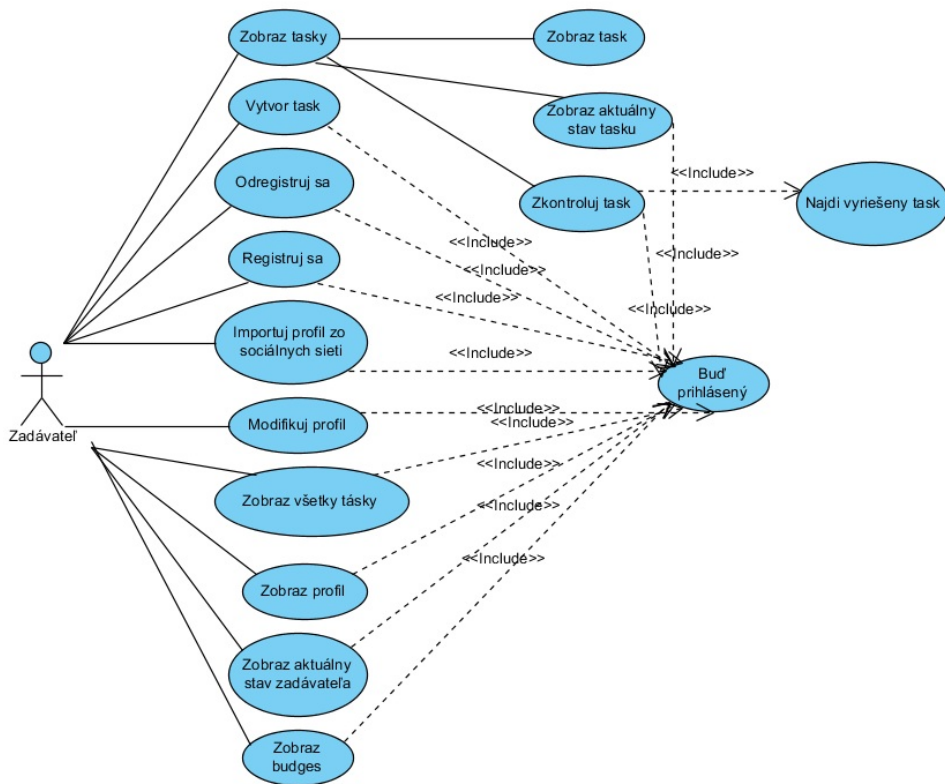
Z funkcionálneho hľadiska sa nám podarilo naimplementovať webovú aplikáciu umožňujúcu registrovať a následne sa prihlásiť používateľom. Aplikácia umožňuje vytvárať rôzne typy úloh s rôznymi nastaveniami nárokov na vypracovanie, ale aj na pracovníka. Je možné taktiež zobrazovať grafickú podobu výsledkov. Na strane pracovníka je možné úlohy prerušovať a ľubovoľne meniť poradie ich vypracovania. Jednou z významných funkcionalít je práca a profilom, ktorý si môže používateľ vyplňať a upravovať. Do profilu je možné načítavať informácie z profilov na sociálnych sieťach po integrácii s účtom. V profile je možné nastavovať najrôznejšie veci od mena, priezviska až po vzdelanie a pracovnú kvalifikáciu.

4.3 Prípady použitia

V systéme máme dve hlavné roly a to zadávateľ a pracovník. Pracovník vypracováva úlohy zadané zadávateľom. Je mu umožnené prezerat si zadania základných úloh. Po registrácii sa môže prihlásiť do systému. Po prihlásení do systému pracovník môže úlohy, ktoré sa mu zobrazia aj vypracovať. Taktiež si môže prezerat a dopĺňať profil a tým zvyšovať svoje šance na zobrazenie lepších úloh. Úlohy môžu byť zadané s rôznymi požiadavkami. Jedným z tipov požiadaviek môžu byť špecifické hodnoty z profilu. Pracovník môže taktiež úlohu prerušiť. Je možné si prezerat aj zatiaľ dosiahnuté výsledky ako môžeme vidieť na Obrázku 31



Obr. 31: Zobrazenie prípadu použitia pracovníka



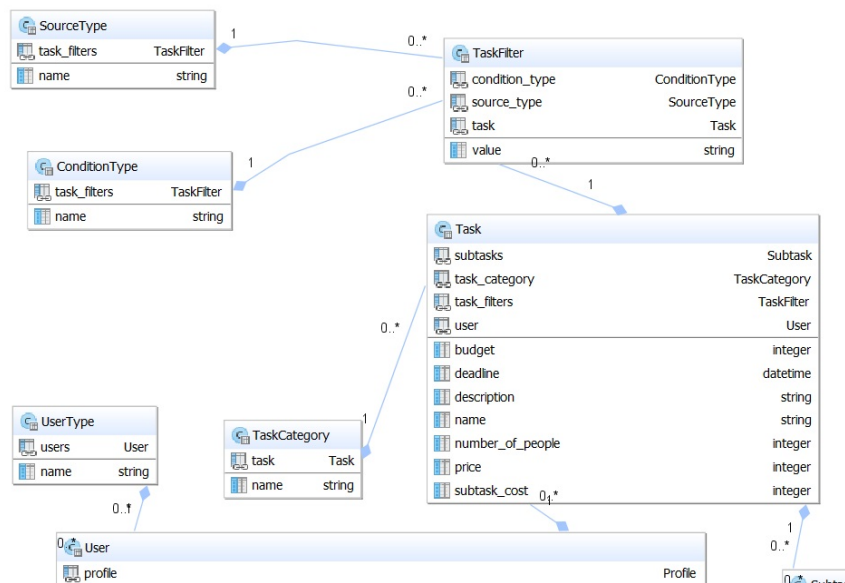
Obr. 32: Zobrazenie prípadu použitia zadávateľa

Zadávateľovi, a pracovníkovi sú pred prihlásením zobrazené tie isté informácie a majú možnosť pracovať s rovnakou funkcionalitou v aplikácii. Zmena nastáva

až po prihlásení, kedy Zadávateľ má iné možnosti práce z aplikáciou a je schopný vytvárať úlohy. Týmto úlohám môže priradiť rôzne kritéria na vlastnosti pracovníka. Následne sa úlohy zobrazia iba pracovníkovi, ktorý tieto kritéria spĺňajú. Ako môžeme vidieť zobrazené na prípade použitia obr 2. .Zadávateľ si taktiež môže vytvorené úlohy prezerať .

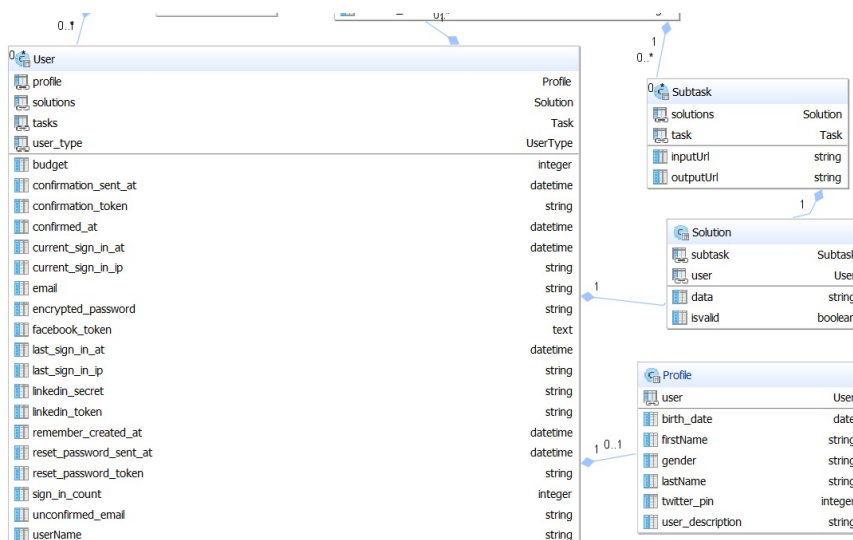
4.4 Dátový model

Dôležitými dátovými štruktúrami sú Task, Subtask a Solution. Tieto dátové štruktúry uchovávajú všetky úlohy vytvorené zadávateľmi v systéme a ich stavy. Sú dôležité pre zadávateľa, ale aj pracovníka. Obaja však majú rôzny pohľad na informácie v týchto dátových štruktúrach. Hlavnou dátovou štruktúrou v projekte je



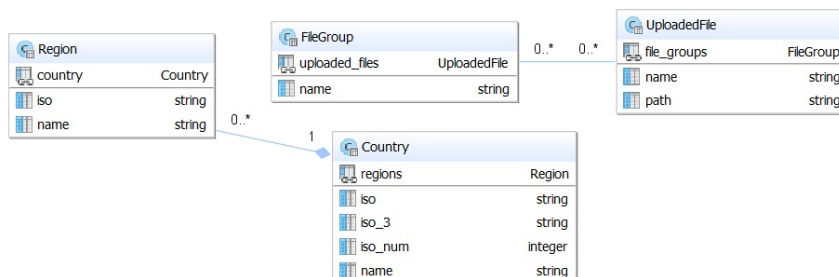
Obr. 33: Dátový model zobrazujúci štruktúru Task, Subtask a Solution.

User. User obsahuje informácie o používateľovi. Používateľom je zadávateľ, ale aj pracovník. Dátová štruktúra slúži na ukladanie informácií pre prihlásenie, registráciu a iných stavov používateľa v systéme. Tabuľka je prepojená s ďalšími dátovými štruktúrami, ktoré slúžia na ukladanie informácií o používateľoch.



Obr. 34: Datový model zobrazujúci štruktúru User.

Jednou z takýchto dátových štruktúr je Profile, ktorý v sebe uchováva doplňujúce informácie dôležité pre uploadera. Profile je neoddeliteľnou súčasťou User. Po vytvorení User dátovej štruktúry sa automaticky vytvorí aj Profile dátová štruktúra. Pri zmazení User sa zmaže aj Profile. Opačne to neplatí. Pre uloženie regiónu a krajiny slúžia dátové štruktúry Region a Country.



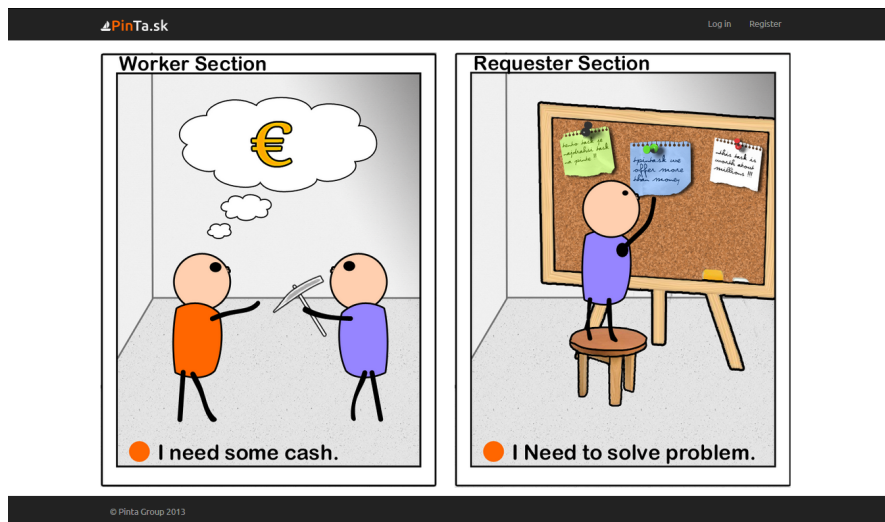
Obr. 35: Datový model zobrazujúci datovú štruktúru Country, Region, Uploaded-File.

5 Používateľská príručka

Úvodná stránka používateľovi ponúka na výber dve sekcie. Vľavo je sekcia pre “zadávateľa” a vpravo pre “pracovníka”. Každá rola má čiastočne upravené dostupné funkcie.

Role:

- Zadávateľ - zadáva úlohy, ktoré riešia pracovníci
- Pracovník - má možnosť vypracovať úlohy zadane zadávateľom



Obr. 36: .

Hlavné menu pre prihlásením ponúka možnosti:

- Log in/Sign in (prihlásenie)
- Register/(registrácia)

5.1 Prihlásenie do systému

Na úvodnej stránke si zvolte ponuku pre prihlásenie (Log in)

Pre prihlásenie je potrebný email a heslo používateľa, s ktorými sa registroval. Nezabudnite pred prihlásením potvrdiť aktiváciu prostredníctvom odkazu, ktorý vám prišiel do emailu. V prípade že vám neprišiel, využite možnosť opätovného zaslania (odkaz “Didn’t received confirmation instructions?”).

V prípade, že ste zabudli prihlasovacie heslo. Môžete využiť možnosť obnovy hesla (odkaz “Forgot your password?”).

Sign In

Email

Password

Remember me

[Sign up](#)
[Forgot your password?](#)
[Didn't receive confirmation instructions?](#)

Obr. 37: .

5.2 Registrácia do systému

Na úvodnej stránke si zvolte ponuku pre registráciu (Register)

Pre úspešnú registráciu je nutné vyplniť všetky polia vrátane krajiny a regiónu.

- V poli email musí byť vyplnená platná emailová adresa.
- heslá sa musia zhodovať

- krajina aj region musí byť vyplnený - systém automaticky tieto údaje odhadne z vašej IP adresy, tieto údaje sú použité na ciele úlohy pre určité lokality.

Sign Up

Email
worker@example.com

Password
••••••••

Password confirmation
Your password again

Country
Slovakia

Region
▼

Sign up

Sign in
Didn't receive confirmation instructions?

Obr. 38: .

5.3 Opätovné zaslanie aktivačného emailu

V prípade, že vám neprišiel email s aktivačnými inštrukciami. Môžete požiadať o znovuposlanie cez formulár prístupný cez pohľad prihlásenie a odkaz “Didn't receive confirmation instructions?”.

Resend confirmation instructions

Email

Resend confirmation instructions

Sign in
Sign up
Forgot your password?

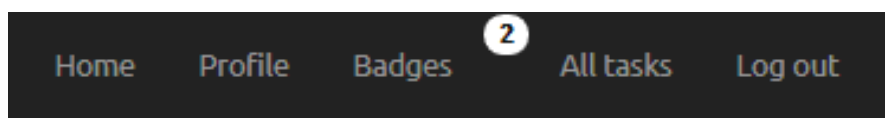
Obr. 39: .

5.4 Sekcia pre pracovníka

Hlavné menu ponúka nasledovné možnosti:

- Home - domovská stránka so základnými informáciami o používaní systému a inými detailmi podľa aktuálnej ponuky (v aktuálnej verzii novinky a základne informácie, štatistiky vypracovaných úloh)
- Profile - profil používateľa. Informácie o používateľovi, ktoré je možné prepojiť so sociálnymi službami a tým získať ďalšie výhody. Tieto informácie slúžia na ciele zadávanie úloh.

- Badges - achievement-y alebo úspechy, ktoré používateľ dosiahol/môže dosiahnúť , číslo pri odkaze značí počet nových achievement-ov
- All Tasks - zoznam riešiteľných úloh



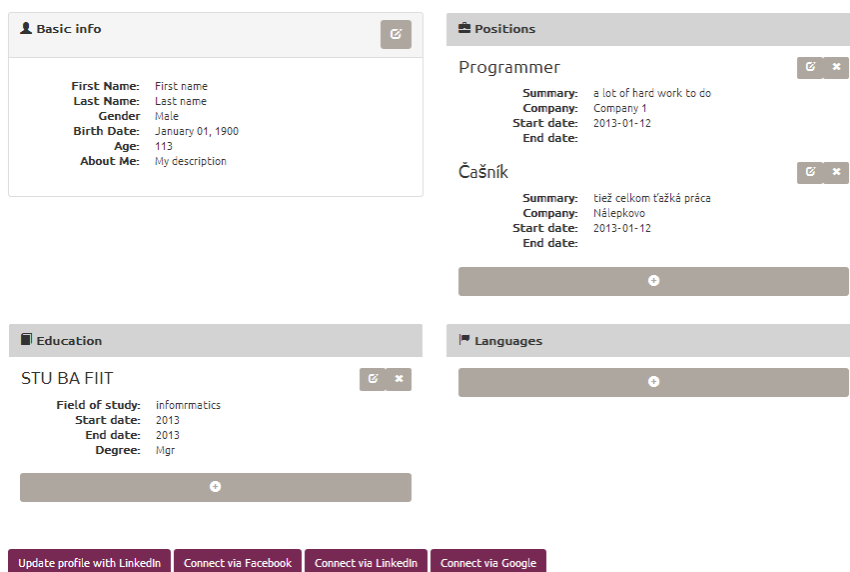
Obr. 40: .

5.4.1 Profil

Pohľad s informáciami o používateľovi. Všetky údaje môžu byť upravené pomocou príslušného ovládacieho prvku:

- zmena/upravenie záznamu
- pridanie záznamu
- odstránenie záznamu

Tlačidlá “Connect via xxxx” slúžia na prepojenie účtu s príslušnou sociálnou službou. Po prepojení sa tlačidlo zmení na “Update profile with xxxx”, po kliknutí obnoví informácie v profile pomocou dát z príslušnej služby.



Obr. 41: .

5.4.2 Badges

Zoznam dostupných achievement-ov (úspechov) dosiahnutých používaním systému. V budúcnosti budú pužité na ciele zadávanie úloh.

Listing your badges

id	name	title	description	requirements	granted?
1	apprentice-requester		Apprentice student of job requesting	1	F
10	apprentice-worker		Apprentice student of job solving	1	F
11	journeyman-worker		Journey man in job solving	10	F
12	expert-worker		Expert in job solving	100	F
13	artisan-worker		Artisan in job solving	500	F
14	master-worker		Master in job solving	1000	F
15	grand-master-worker		Grand master in job solving	5000	F
16	illustrious-worker		Illustrious in job solving	10000	F
20	first-language		First language defined.	1	F
21	first-education		First education defined.	1	T
22	first-position		First position defined.	1	T
23	first-skill		First skill defined.	1	F

Obr. 42: .

5.4.3 All tasks

Pohľad pre zobrazenie úloh dostupných na riešenie.

Každá riešiteľná úloha má v slípce action tlačidlo “solve”, ktoré sprístupni proces riešenia úlohy. Každá úloha je sprevádzana sprievodným textom, ktorý slúži ako príručka.

Title

Available Task: 10

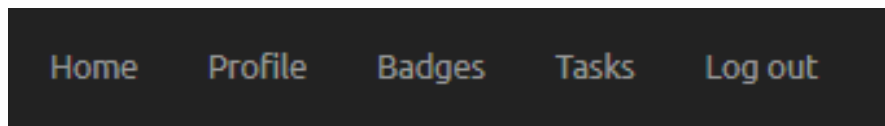
id	Name	Description	Number Of People	Subtask Cost	Budget	Deadline	Task Category	User	Created At	Action
1	test		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve
2	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve
3	test		100	1	1		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve
4	test2		100	1	2		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve
5	test		100	1	0		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Closed
6	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve
7	test		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve
8	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve
9	test		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve
10	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve

Obr. 43: .

5.5 Sekcia pre zadávateľa

Hlavné menu ponúka nasledovné možnosti:

- Home - domovská stránka so základnými informáciami o používaní systému a inými detailmi podľa aktuálnej ponuky (v aktuálnej verzii novinky a základne informácie, štatistiky zadaných úloh a výsledkov)
- Profile - profil používateľa. Informácie o používateľovi, ktoré je možné prepojiť so sociálnymi službami a tým získať ďalšie výhody
- Badges - achievement-y alebo úspechy, ktoré používateľ dosiahol/môže dosiahnuť, číslo pri odkaze značí počet nových achievement-ov
- Tasks - zoznam zadaných úloh
- Logout - odhlásenie zo systému



Obr. 44: .

5.5.1 Profil

V aktuálnej verzii totožný s profilom pracovníka

5.5.2 Badges

V aktuálnej verzii totožný s pohľadom pre achievement-y pracovníka

5.5.3 Tasks

Zoznam úloh v riešení. Každý task má v stĺpci action nasledovné tlačidlá:

- edit - upravenie úlohy - funkcionálna analogická k funkcii vytvorenia úlohy
- results - pohľad na výsledky vo forme grafov a štatistík
- delete - odstránenie úlohy zo zoznamu riešených

Funkcionálna vytvorenia úlohy (tlačidlo “create new task”), po rozkliknutí zobrazí postupnosť krokov potrebných k vytvoreniu úlohy.

Your current tasks list

Create new task

Id	Name	Description	People count	Subtask cost	Budget	Deadline	Category	User	Created	Action
1	test		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
2	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
3	test		100	1	1		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
4	test2		100	1	2		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
5	test		100	1	0		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
6	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
7	test		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
8	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
9	test		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
10	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete

Obr. 45: .

Postup vytvorenia úlohy

1. krok - Z ponuky vyberte typ úlohy ktorú chcete zadať.

New Task

The screenshot shows the 'New Task' form at Step 1, 'Task Category'. At the top, there are three step indicators: '1 Step 1 Task Category' (highlighted in orange), '2 Step 2 Task Details', and '3 Step 3 Subtasks'. The main form area is titled 'Task Category' and contains a 'Task category' dropdown menu with 'Default' selected. At the bottom right, there are three buttons: 'Previous', 'Next', and 'Finish'.

Obr. 46: .

2. krok - Vyplňte informácie o úlohe

New Task

The screenshot shows the 'New Task' form at Step 2, 'Task Details'. The step indicators at the top are '1 Step 1 Task Category', '2 Step 2 Task Details' (highlighted in orange), and '3 Step 3 Subtasks'. The main form area is titled 'Task Details' and contains several input fields: 'Name', 'Description', 'Subtask cost', 'Budget', and 'Number of people'. The 'Deadline' field is a date picker showing '1', 'December', '2013', '18', and '02'. At the bottom right, there are three buttons: 'Previous', 'Next', and 'Finish'.

Obr. 47: .

3. krok - Podľa typu úlohy vyplnete požadované polia a odovzdajte úlohu

New Task

1 Step 1
Task Category

2 Step 2
Task Details

3 Step 3
Subtasks

Subtasks

Questions

Image from TextBox Remove

Image from TextBox Remove

Add Question

Previous Next Finish

Obr. 48: .

New Task

1 Step 1
Task Category

2 Step 2
Task Details

3 Step 3
Subtasks

Subtasks

Facebook URLs

Add Link

Previous Next Finish

Obr. 49: .

4. krok - odovzdanie úlohy na riešenie a obmedzenie viditeľnosti úloha je odovzdaná od ukončenia kroku 3. V záverečnej fáze je možno dodatočne upraviť viditeľnosť úlohy len pre špecifické skupiny používateľov pomocou dostupných filtrov. Filtrovateľné atribúty(source type):

- krajina pôvodu
- počet vyriešených úloh

Operátory:

- == - rovnosť
- <= - menší alebo rovný
- >= - väčší alebo rovný
- < - menší
- > - väčší
- includes - zahrňa

Value (hodnota) : druhý operand zvolenej operácie filtra
Stlačením tlačidla “create filter” bude filter vytvorený a aplikovaný.

Task was successfully created.

Name: Task
Description: Description
Number of people: 10
Deadline: 2013-12-11 20:42:00 UTC
Task category: Survey
User: Jason Derulo

:

Source type

User Country

Condition type

==

Value

Create Task filter

Back

Edit

Delete

Obr. 50: .