



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

WEBOVÝ PORTÁL PRO SPOLUPRÁCI STUDENTŮ VUT

WEB PORTAL FOR COOPERATION OF BUT STUDENTS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ SVETLÍK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÍTĚZSLAV BERAN, Ph.D.

BRNO 2019

Zadání bakalářské práce



21237

Student: **Svetlík Tomáš**
Program: Informační technologie
Název: **Webový portál pro spolupráci studentů VUT**
Web Portal for Cooperation of BUT Students
Kategorie: Uživatelská rozhraní

Zadání:

1. Prostudujte nástroje pro tvorbu moderních webových aplikací a metodami UX.
2. Navrhněte informační systém pro podporu spolupráce studentů VUT při řešení týmových projektů. Systém musí umožnit zadávání projektů a jejich parametrů, včetně požadavků na řešitelský tým a kompetence členů týmu. Dále musí umožnit efektivní filtraci projektů a vyhledávání.
3. Implementujte navržený systém s využitím relevantních dostupných technologií. Zaměřte se na kvalitu rozhraní různých částí systému.
4. Vyhodnoťte vlastnosti výsledného systému na základě experimentů s reálnými uživateli.
5. Prezentujte klíčové vlastnosti řešení formou plakátu a krátkého videa.

Literatura:

- Brian Burke. *Gamify: how gamification motivates people to do extraordinary things*. Brookline, MA: Garthner, Inc., 2014. ISBN 9781937134853.
- Semmy Purewal. *Learning Web App Development: Build Quickly with Proven JavaScript Techniques*. O'Reilly Media, Inc., 2014. ISBN: 9781449370190.
- Steve Krug. *Don't make me think, revisited: a common sense approach to web usability*. San Francisco: New Riders, ISBN 978-0321965516.
- Dále dle pokynu vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2 a částečně bod 3.

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Beran Vítězslav, Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2018

Datum odevzdání: 15. května 2019

Datum schválení: 1. listopadu 2018

Abstrakt

Táto bakalárska práca sa venuje procesu tvorby moderných webových aplikácií s využitím metód UX. Cieľom je navrhnúť webový portál pre podporu spolupráce študentov a overiť vytvorený návrh pomocou testovacej verzie portálu. Podpora spolupráce sa zameriava na proces hľadania spolupracovníkov ku projektu. Systém umožňuje užívateľom si vytvoriť vlastný projekt alebo vyjadriť záujem na už existujúcu pozíciu. Výsledný webový portál je implementovaný s použitím moderných webových technológií ako framework Laravel a knižnica React. Výstupom práce je návrh, implementovaný prototyp a vyhodnotenie navrhutej aplikácie, na základe užívateľského testovania.

Abstract

This bachelor thesis deal with the process of creating modern web applications with using UX methods. The aim is to design a web portal to support students cooperation and verify the design using a test version. Collaborative support focuses on the process of looking for project collaborators. The system allows users to create their own project or to be interested in an existing positions. The web portal was created with modern web technologies such as Laravel or React library. The output of the thesis is the design, prototype and evaluation of the application, based on user testing.

Klíčové slová

Webový portál, Spolupráca študentov, Webové technológie, React, Laravel, MySQL, Bootstrap, UX, Metody UX, Užívateľské testovanie

Keywords

Web portal, Student collaboration, Web technologies, React, Laravel, MySQL, Bootstrap, UX, UX methods, User testing

Citácia

SVETLÍK, Tomáš. *Webový portál pro spolupráci studentů VUT*. Brno, 2019. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Vítězslav Beran, Ph.D.

Webový portál pro spolupráci studentů VUT

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pana Ing. Vítězslava Berana Ph.D.. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Tomáš Svetlík
15. mája 2019

Podakovanie

Chcel by som poďakovať vedúcemu práce Ing. Vítězslavu Beranovi Ph.D za čas a rady, ktoré mi venoval počas priebehu tvorby tejto práce. Poďakovať by som sa chcel aj všetkým, ktorí sa zúčastnili jednotlivých fáz testovania a získavania dát. V neposlednom rade ďakujem aj kamarátom a rodine za podporu.

Obsah

1	Úvod	3
2	Vývoj moderných webových aplikácií	4
2.1	Klientská časť	4
2.2	Serverová časť	7
2.3	Komunikácia	9
3	UX pri tvorbe webovej aplikácie	10
3.1	User Experience	10
3.2	Výskumné metódy	12
3.3	Analytické metódy	13
3.4	Návrhárske metódy	14
3.5	Metódy testovania a validácie	14
4	Analýza a návrh riešenia	16
4.1	Cieľ práce	16
4.2	Cielový užívateľia	16
4.3	Existujúce riešenia	17
4.4	Špecifikácia požiadaviek	18
4.5	Prípady použitia	21
4.6	Návrh grafického užívateľského rozhrania	22
4.7	Dátový model	26
5	Implementácia	29
5.1	Použité technológie	29
5.2	Použité cudzie časti	29
5.3	Adresárová štruktúra	31
5.4	Dátové modely	32
5.5	Kontroléry	32
5.6	Pohľad	33
6	Užívateľské testovanie a vyhodnotenie testovania	35
6.1	Prvá fáza	35
6.2	Druhá fáza	36
6.3	Výsledky testovania	36
6.4	Zhrnutie testovania	37
6.5	Možné vylepšenia	38

7 Záver	39
Literatúra	40
A Obsah priloženého pamäťového média	42
B Persony	43
C Zadania úloh 2.fázy testovania	45

Kapitola 1

Úvod

Študenti často prichádzajú s rôznymi nápadmi a projektami, ktoré by chceli zrealizovať. K uskutočneniu nápadu im veľakrát nestačia vlastné znalosti a skúsenosti. Vyhľadávajú preto niekoho, s kým by mohli spolupracovať a kto by im mohol pomôcť s realizáciou. Dnes ale neexistuje veľa spôsobov ako by študent mohol nájsť spolupracovníkov. Moja práca prináša riešenie, ktoré celý proces nájdenia spoločníkov zjednoduší a zprehľadní.

Náplňou a cieľom tejto bakalárskej práce je navrhnúť webový portál pre spoluprácu študentov a svoj návrh overiť pomocou testovacej verzie. Overovanie bude prebiehať formou užívateľského testovania. Testovaním získame spätnú väzbu od užívateľov a návrh budeme môcť zhodnotiť. Návrh a vývoj aplikácie sa zameriava na použiteľnosť a užívateľskú prívetivosť. Z toho dôvodu počas práce používam niekoľko metód UX.

Hlavnou zložkou systému budú projekty. Užívatelia si v systéme budú môcť vytvoriť projekt a vyjadrovať záujem o pozíciu na už existujúcich projektoch. Zo všetkých záujemcov si autor projektu bude môcť vybrať s kým chce spolupracovať a vymeniť si kontakt, prípadne sa dohodnúť na stretnutí. Systém tiež bude poskytovať profily projektov s informáciami a zoznam projektov s možnosťou vyhľadávania a filtrovania.

Táto práca sa skladá z niekoľkých dielov. Teoretickej časti práce sa venujú dve kapitoly. Prvá z nich sa zaoberá technológiami pre vývoj moderných webových aplikácií. Zameraná je hlavne na technológie, ktoré sú použité počas implementácie prototypu. Ďalšia z teoretických kapitol definuje čo znamená UX a poukazuje na metódy UX, ktoré sa často používajú pri vývoji webových aplikácií v jednotlivých fázach vývoja. V nasledujúcich kapitolách prejdem k vývoju webového portálu. Štvrtá kapitola sa zameriava na získanie dát, ich analýzu a návrh riešenia. Postupne v nej budem určovať ciele, cieľových užívateľov, pozriem sa na existujúce riešenia, špecifikujem požiadavky a uvediem prípady použitia. V rámci tejto kapitoly sa tiež pozrieme na návrh grafického rozhrania a dátového modelu. Nasledovať bude kapitola venovaná implementácií, ktorá predstavuje všetky použité technológie. Ďalej tiež popisuje štruktúru aplikácie a jednotlivé implementované časti. Záverečná kapitola približuje vykonané užívateľské testovanie a zmieňuje sa o možných rozšíreniach alebo vylepšeniach navrhnutého webového portálu.

Kapitola 2

Vývoj moderných webových aplikácií

Webová aplikácia je počítačový program, ktorý je uložený na vzdialenom servere a prostredníctvom internetu je poskytovaný užívateľom. Užívatelia k nej prístupujú prostredníctvom webových prehliadačov. Jej inštalácia do užívateľovho zariadenia nie je potrebná. Práve to bol jeden z dôvodov, prečo sa webové aplikácie stali po rozšírení internetu veľmi populárne.

Vytváranie webových aplikácií je častokrát zložitá úloha, pretože potrebujeme použiť viacero programovacích jazykov. Webová aplikácia pozostáva na princípe klient-server architektúry. To znamená, že môžeme jej vývoj rozdeliť na dve časti. Tieto časti sa nazývajú klientská a serverová.

V tejto kapitole priblížim technológie pre vývoj jednotlivých častí webovej aplikácie. V prvej podkapitole sa pozriem na klientskú časť a v druhej prejdem tvorbu serverovej časti. Vybrané sú hlavne technológie, ktoré budú použité pri implementácii. Spomenuté sú však aj niektoré iné technológie, ktoré sa používajú veľmi často.

2.1 Klientská časť

Klientská časť webovej aplikácie beží na strane klienta, teda najčastejšie vo webovom prehliadači. Jej úlohou je komunikácia s užívateľom a so serverom. Užívateľ si určitým úkonom, napríklad kliknutím na odkaz, vyžiada informácie, ktoré chce dostať. Klientská časť toto kliknutie zachytí a pošle požiadavku na server. Dáta vrátené zo serveru sú potom zobrazené užívateľovi. Na implementáciu frontendu, ako sa klientskej časti hovorí, sa používajú najmä značkovací jazyk HTML, kaskádové štýly (CSS) a skriptovací jazyk Javascript.

HTML¹

HTML, čo je skratka pre hypertextový značkovací jazyk (HyperText Markup Language), je značkovací jazyk ktorý sa používa na tvorbu webových stránok a aplikácií. Definuje štruktúru a obsah webovej aplikácie. Pôvodne sa pomocou neho určoval aj vzhľad výsledného dokumentu ale ten sa už zvykne definovať pomocou CSS.

Základným stavebným blokom HTML dokumentu je element. Elementy sa do seba zoradujú a vytvárajú hierarchickú štruktúru. V dokumente sú značené pomocou HTML tagov. Rozsah elementu je zvyčajne určený párom „start“ tag a „end“ tag. Ak element pozostáva

¹<https://en.wikipedia.org/wiki/HTML>

z určitého textového obsahu, text je vložený medzi tagmi. „Start“ tag môže v sebe obsahovať atribúty, ktoré sú nositeľom určitej informácie. Väčšina atribútov je párových (meno, hodnota).

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>My Web App</title>
5   </head>
6   <body>
7     <h1>Hello, World!</h1>
8   </body>
9 </html>
```

Výpis 2.1: Základná kostra HTML dokumentu s tagom h1, ktorý vypisuje nadpis „Hello, World!“

CSS²

Na popis štýlu HTML dokumentu sa používa jazyk CSS, skratka pre kaskádové štýly (Cascading Style Sheets). Využíva sa aj na prispôbenie prezentovaného rozhranie na rôzne veľkosti zariadení. CSS je nezávislé od HTML, čo umožňuje oddelenie definície obsahu a definície vzhľadu. Toto oddelenie prinieslo zjednodušenie a zprehľadnenie kódu. Definovanie CSS v samostatnom súbore umožňuje jednoduchú zmenu celého štýlu alebo možnosť viacerým HTML dokumentom zdieľať rovnaký vzhľad.

Tento jazyk sa dá použiť aj s inými značkovacími jazykmi ako napríklad XHTML, SVG a XML.

Pomocou CSS sa HTML elementu priradujú pravidlá. Vďaka týmto pravidlám sa formuje umiestnenie v dokumente, veľkosť, farba, štýl, ohraničenie a mnoho ďalších vlastností elementu.

```
1 h1 {
2   color: red;
3   text-align: center;
4   font-size: 32px;
5 }
```

Výpis 2.2: Definovanie štýlu pre element h1

Javascript

Jazyky HTML a CSS nám vytvárajú iba statickú verziu stránky. Na pridanie interaktívnych prvkov nám slúži javascript, ktorý sa stal práve vďaka tomu nevyhnutnou súčasťou webových aplikácií.

Je to dynamický, interpretovaný a objektovo orientovaný programovací jazyk. Pôvodne implementovaný len na strane klienta vo webových prehliadačoch, ale dnes sa už používa aj na serverovej strane. Javascript je podporovaný všetkými webovými prehliadačmi, avšak na odlišných prehliadačoch sú jeho rôzne verzie, čo spôsobuje problémy. Práve kvôli tomu sa veľmi často začala používať knižnica jQuery.

Javascriptová knižnica jQuery značne zjednodušuje programovanie javascriptu. Uľahčuje interakciu so stránkou, reagovanie na činnosť užívateľa, načítavanie a zobrazovanie nového

²<https://www.w3.org/standards/webdesign/htmlcss>

obsahu. Knižnica jQuery sa však z vývoja webových aplikácií pomaly vytráca a nahradzujú ju javascriptové frameworky. Tieto frameworky sú efektívnejšie a vytvárajú prehľadnejší kód. Dnes najpoužívanějšími nástrojmi sú Angular, React, Vue.js, Node.js a niekoľko ďalších [2]. Ja som sa rozhodol pre svoju aplikáciu využiť React.

React³

React v skutočnosti nie je framework, ale javascriptová knižnica pre tvorbu užívateľského rozhrania. Bola vyvinutá spoločnosťou Facebook v roku 2011 a v roku 2013 bola táto knižnica slobodne zverejnená. Je to pomerne mladý nástroj, ktorý sa však hneď stal veľmi obľúbený medzi vývojármi. Jedným z dôvodov, prečo sa stal populárnym je jednoduchá výučba.

S Reactom je nenáročné vytvárať interaktívne a znovupoužiteľné prvky užívateľského prostredia. Každá React aplikácia má minimálne jeden koreňový uzol, ktorého obsah je definovaný pomocou Reactu.

Srdcom všetkých React aplikácií sú komponenty. Komponent je nezávislá časť užívateľského rozhrania. Každá React aplikácia obsahuje minimálne jeden komponent, ktorý nazývame koreňový komponent. Tento komponent reprezentuje pozadie aplikácie a obsahuje ostatné podkomponenty. Spojením komponentov nám vznikne výsledná podoba stránky aplikácie. Komponenty sú implementované podobne ako javascriptové triedy ale sú písané v JSX. JSX (Javascript XML) je rozšírenie XML/HTML do jazyka Javascript. Komponenty obsahujú dáta (stav komponentu) a metódu *render()*. Dáta sa dajú odovzdávať z komponentu do podkomponentu. Výstupom metódy *render()* je React element, ktorý nám vykresluje obsah a vzhľad komponentu. React element je jednoduchý objekt reprezentujúci element z DOM.

```
1 class HelloComponent extends React.Component {
2
3   render() {
4     return (
5       <div>
6         Hello World
7       </div>
8     );
9   }
10 }
```

Výpis 2.3: Ukážka jednoduchého React komponentu, ktorý vypíše „Hello World“

React drží v pamäti určitú reprezentáciu DOM (Document Object Model), ktorá sa nazýva virtuálny DOM. Vo virtuálnom DOME sa zaznamenávajú zmeny v komponentoch. Vďaka virtual DOM konceptu je React schopný znovu vykresliť iba tie komponenty, v ktorých bola zaznamenaná zmena. Zmena sa zisťuje porovnaním virtuálneho DOMu a pôvodného DOMu. Týmto sa zvyšuje rýchlosť načítania a znižujú sa žiadosti o prenos dát.

React sa stará iba o vykreslenie pohľadu a uisťuje, že pohľad je synchronizovaný zo stavom systému. K niektorým ďalším činnostiam ako je napríklad smerovanie alebo volanie služieb HTTP potrebuje využiť ďalšie knižnice. Celá táto časť o Reacte čerpala okrem oficiálnej dokumentácie Reactu aj zo zdroju [4].

³<https://reactjs.org/docs/>

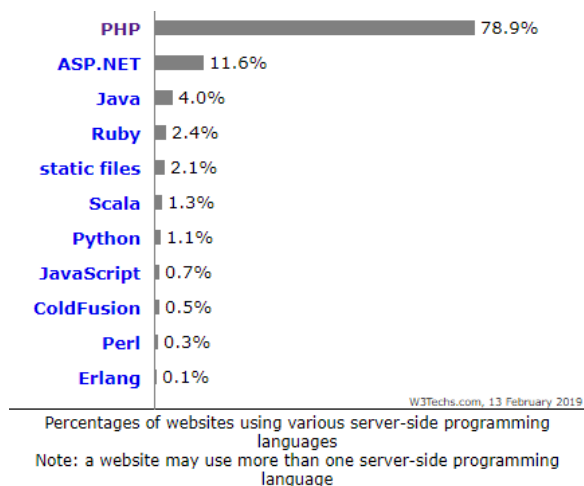
Bootstrap ⁴

Bootstrap je slobodná knižnica pre vývoj klientskej časti aplikácie pomocou HTML, CSS a Javascriptu. Táto knižnica obsahuje množstvo preddefinovaných šablón pre tvorbu komponentov na web, ako tlačidlá, formuláre, zoznamy, menu atď. Využitím týchto komponentov si uľahčíme vývoj.

Bootstrap tiež poskytuje možnosť vytvorenia responzívneho rozhrania, ktoré je v dnešnej dobe veľmi výhodné a potrebné. Pomocou responzívneho rozhrania vieme zabezpečiť úpravu vzhľadu webovej aplikácie pri zmene veľosti zariadenia. Webová aplikácia teda ostane užívateľsky priateľná aj v mobilných zariadeniach.

2.2 Serverová časť

Serverová časť beží na serveri a zabezpečuje funkcionality webovej aplikácie. Úlohou serverovej aplikácie je na základe požiadavky od webového prehliadača odoslať stránku alebo dáta. Dáta sú získané z databázy. Pre tvorbu databázy sa najčastejšie používa jazyk SQL. Technológií pre tvorbu serverovej aplikácie je veľké množstvo. V podstate každý kód, ktorý je spustiteľný na počítači a umožňuje posielat odpovede na požiadavky od klienta, môže bežať na servery. Najčastejšie sa používajú jazyky alebo frameworky jazykov PHP, Ruby, Python, C, Java a Javascript.



Obr. 2.1: Prehľad najpoužívanejších technológií pre serverovú časť [3].

PHP ⁵

Najpopulárnejším jazykom na strane servera je PHP (HyperText Preprocessor). Je to slobodný skriptovací jazyk, ktorý bol navrhnutý špeciálne na tvorbu webu. Je to nástroj pre tvorbu dynamických a interaktívnych webových aplikácií. Jazyk má jednoduchú syntax, ktorá je odvodená z jazykov C, Java a Perl. Program v PHP sa môže priamo vkladať do HTML stránky medzi značky `<?php` a `?>`.

⁴<https://getbootstrap.com/>

⁵<https://www.php.net>

PHP dokáže pomerne jednoducho spolupracovať s relačnými databázami. Svoju atraktivitu PHP udržuje najmä vďaka frameworkom. Tieto frameworky uľahčujú a urýchľujú vývoj webovej aplikácie. Poskytujú nám rôzne výhody, ako napríklad udržiavateľnejší a znovuvyužitelný kód, zaisťujú rozdelenie prezentačnej časti od logiky a presadzujú moderné vývojové metódy [9]. Pre realizáciu aplikácie som sa rozhodol využiť framework Laravel.

Laravel⁶

Laravel je slobodný PHP framework, ktorý patrí medzi najobľúbenejšie frameworky. Umožňuje rýchly vývoj webovej aplikácie a poskytuje prehľadný a udržiavateľný kód. Vybral som si ho, pretože poskytuje prepracovanú dokumentáciu a má veľa schopností na uľahčenie vývoja. Takýmito schopnosťami je napríklad riešenie bezpečnosti webu, autentifikácia či systém notifikácií. Do Laravelu je možné pridať množstvo rozšírení a pomocných nástrojov. Laravel je založený na architektonickom vzore MVC (Model View Controller).

MVC

Jedným z najdôležitejších konceptov pri vývoji webových aplikácií je návrhový vzor Model-View-Controller (skratka MVC). Je to prístup k aplikačnej architektúre, ktorý sa zväčša venuje návrhu databázovo riadenej aplikácie [14]. Tento vzor sa považuje za najlepší spôsob ako programovať webové aplikácie, a preto je súčasťou mnohých frameworkov na tvorbu aplikácií.

Architektúra rozdeľuje aplikáciu do troch komponentov: dátový model, užívateľské rozhranie a riadiacu logiku. Modifikácie jednej z týchto častí má minimálny vplyv na zvyšné časti.

- **Model (model)** – Ústrednou zložkou vzoru MVC je Model, ktorý reprezentuje dáta a zaisťuje správu dát v aplikácií. Na starosti má tiež aplikačnú logiku.
- **View (pohľad)** – Táto časť predstavuje užívateľské rozhranie, ktoré je vo forme klientskej aplikácie. Stará sa o sprostredkovanie získaných dát z radičov. Dáta sa sprostredkujú užívateľom v čo najpriateľnejšej forme.
- **Controller (radič)** – HTTP požiadavky z webového prehliadača sú presmerované do radičov. Tieto radiče nám zabezpečujú riadiacu logiku. Prijaté požiadavky sa zmenia na akcie, ktoré zvyčajne koordinujú databázové akcie na modeli. Po vykonaní akcie sa zasiela odpoveď.

MySQL ⁷

MySQL je relačný databázový systém, ktorý vychádza z programovacieho jazyka SQL. Je rýchly, jednoduchý a slobodný systém, vďaka čomu je to veľmi obľúbený prostriedok pre tvorbu databáz. Podporovaný je na viacerých platformách (Windows, Linux ...) a prácu s týmto systémom podporuje mnoho programovacích jazykov. Dáta sú uložené v tabuľkách, ktoré vytvárajú databázu. V riadkoch tabuľky sa ukladajú jednotlivé informácie a stĺpce určujú dátový typ.

⁶<https://laravel.com/docs/5.8>

⁷<https://www.mysql.com/>

Komunikácia s databázou prebieha pomocou dotazov, ktoré sú písané v programovacom jazyku SQL. Pre jednoduchšiu správu databázi boli vytvorené nástroje ako phpMyAdmin⁸ alebo Adminer⁹. Tieto nástroje umožňujú zálohovanie, vytváranie tabuliek, pridávanie, edíciu a mazanie záznamov z tabuliek.

2.3 Komunikácia

Táto časť sa zaoberá komunikáciou medzi klientskou a serverovou časťou. Túto komunikáciu má na starosti sada technológií, ktorú si môžeme predstaviť ako most medzi klientom a serverom [14]. Pozriem sa teda bližšie na technológie, ktoré vedú ku jednoduchšej výmene informácií.

JSON ¹⁰

JSON (JavaScript Object Notation) je jedným z najpoužívanejších dátových formátov pre výmenu dát. Využíva sa na prenos krátkych štruktúrovaných dát medzi serverom a klientom. Pre človeka je tento formát ľahko čitateľný a zapisovateľný. To isté platí aj pre stroje, ktoré s ním dokážu jednoducho pracovať. Je založený na javascriptovom zápise reťazcových objektov, ale je kompletne nezávislý od programovacieho jazyka. JSON pozostáva z dvoch štruktúr. Jedna reprezentuje informáciu vo forme páru meno a hodnota, druhá vytvára usporiadaný zoznam hodnôt. Meno a hodnota sú v zápise oddelené dvojbodkou. Jednotlivé informácie sú oddelené čiarkami. Hodnota ukladaná do JSON môže byť číslo, textový reťazec, logická hodnota, null, objekt alebo pole.

AJAX

AJAX (Asynchronous JavaScript And XML) je sada vzájomne súvisiacich webových technológií. Tieto technológie umožňujú získavanie dát zo servera asynchrónne (na pozadí), bez obnovovania stránky. K prenosu dát využíva formát JSON. Aplikácie s použitím AJAXu prinášajú užívateľsky prívetivejšie prostredie.

⁸<https://www.phpmyadmin.net/>

⁹<https://www.adminer.org/sk/>

¹⁰<https://www.json.org/>

Kapitola 3

UX pri tvorbe webovej aplikácie

Na začiatku tejto kapitoly sa zameriavam na to, čo vlastne znamená výraz User Experience (UX) a čo ho ovplyvňuje. V ďalších častiach kapitoly sa budem venovať metódam UX. V oblasti UX sa nachádza veľké množstvo rôznych metód, ktoré sa aplikujú v rôznych fázach vývoja projektu. Postupne predstavím najčastejšie metódy vo fáze výskumu a nasledovať budú metódy pri analýze informácií. V ďalších podkapitolách zase priblížim metódy využívané počas návrhu produktu a tiež predstavím niektoré metódy využívané pri testovaní.

3.1 User Experience

Výraz User Experience (UX) môžeme preložiť ako užívateľská skúsenosť alebo užívateľský zážitok (ďalej v texte bude využívaná anglická verzia alebo skratka, keďže sa to v obore neprekladá). Tento termín prvýkrát použil Don Norman v roku 1988 vo svojej publikácii *The Design of Everyday Things* [12]. V emailovej komunikácii s redaktorom Petrom Merholzom to okomentoval takto: „Vymyslel som tento termín, pretože rozhranie a použiteľnosť mi pripadali príliš úzke. Chcel som pokryť všetky aspekty užívateľských skúseností so systémom, vrátane priemyselného dizajnu, grafiky, rozhrania, fyzickej interakcie a manuálu.“ [8]

Postupom času sa User Experience stala dôležitou súčasťou vývoja systémov a zariadení. Začalo ju používať čoraz viac a viac spoločností. Vznikalo aj mnoho expertov na UX. V praxi sa aj preto vyskytuje veľa definícií, ktoré určujú význam pojmu User Experience. Don Norman a Jacob Nielsen tento pojem definujú nasledovne: „UX zahŕňa všetky aspekty interakcie koncového užívateľa so spoločnosťou, jej službami a produktami.“ [13] Z definície vyplýva, že sa UX zameriava na celkovú skúsenosť, ktorú používateľ získava pri práci s konkrétnym systémom alebo zariadením. Túto skúsenosť vytvára mnoho faktorov, ktoré na užívateľa pôsobia.

Faktory ovplyvňujúce User Experience

UX sa veľmi často zamieňa s použiteľnosťou (usability), ktorá opisuje ako ľahko sa má produkt používať. UX sa ale neobmedzuje iba na použiteľnosť. UX ako disciplína začala s použiteľnosťou, avšak postupne sa vyvíjala a rástla. Aby bolo možné dodávať úspešné produkty, je dôležité upriamiť pozornosť na všetky stránky UX.

Peter Morville, priekopník UX, určil 7 aspektov ktoré ovplyvňujú UX [10]. Tieto faktory sú predstavené ako plást medu, ktorý je ukázaný na obrázku 3.1.



Obr. 3.1: User Experience Honeycomb [10].

- **Užitočné (Useful)** – Produkt musí byť užitočný, teda musí naplniť potreby užívateľa.
- **Použiteľné (Usable)** – Veľmi dôležitá je jednoduchosť používania. Užívateľ musí efektívne a ľahko dosiahnuť svoj cieľ.
- **Cenné (Valuable)** – Produkt musí priniesť určitú hodnotu pre autora i pre užívateľa.
- **Žiaduce (Desirable)** – Žiadosť je identifikovaná prostredníctvom dizajnu. Celkový dizajn ako fotky, farby, fonty, logá, elementy je prostriedkom na vyvolanie emócie.
- **Zistiteľné (Findable)** – Užívatelia musia nájsť obsah, ktorý potrebujú a k tomu musí byť prispôbený dizajn a štruktúra produktu.
- **Prístupné (Accessible)** – Na svete je mnoho ľudí s určitým postihnutým a preto musí byť produkt dostupný aj takýmto užívateľom.
- **Dôveryhodné (Credible)** – Používateľ musí byť schopný dôverovať produktu. Produkt musí nielen urobiť svoju prácu, ale aj poskytovať presné a vhodné informácie a vykonávať prácu v rozumnom čase.

Disciplíny UX

Táto časť vychádza zo zdroju [5]. UX je stále pomerne mladé a naďalej sa rozvíjajúce odvetvie, a preto je ťažké konkrétne definovať disciplíny. Keď sa však bližšie pozrieme do praxe, na základe pracovných miest v UX obore by sme vedeli určiť niekoľko kategórií:

- **Interakčný dizajn a informačná architektúra**

Táto kategória zahŕňa návrh štruktúry systému a organizáciu obsahu, ktorý tento systém poskytuje. Zaoberá sa tiež interakciou, teda vzájomným pôsobením, medzi používateľom a produktom. Cielí na pohyb užívateľa v systéme a ako sa užívateľ dostane k informáciám, ktoré poskytuje daný systém.

- **Vizuálny dizajn**

Zameriava sa na vizuálnu stránku produktu. Navrhuje vzhľad produktu a všetky s tým spojené veci, ako napríklad použité farby, typografiu, grafické prvky. Zakladá sa na emocionálnom pôsobení týchto prvkov.

- **Užívateľský výskum**

Snaží sa o pochopenie správania používateľov a analýzu užívateľských potrieb. Výskum väčšinou zahŕňa celý proces od prvotného zistenie potrieb užívateľa, cez validáciu produktu, až po užívateľské testovanie. Jeho účelom je zlepšenie použiteľnosti produktu.

- **Obsahová stratégia**

Venuje sa obsahovej stránke produktu. Vytvára a spravuje text, obrázky, audio a video, ktoré sa vyskytujú v produkte.

Všetky tieto kategórie spolu častokrát veľmi súvisia a pracovníci, ktorí v týchto oblastiach pracujú, medzi sebou úzko spolupracujú.

3.2 Výskumné metódy

Výskum je veľmi dôležitou časťou tvorby produktu. Pri realizácii potrebujeme dobre poznať problém, ktorým sa produkt zaoberá, a užívateľov, ktorí budú produkt používať. Výskum môže slúžiť aj na objavovanie a naberanie inšpirácie. Pri získavaní informácií môžeme aplikovať rôzne metódy. Výstup týchto metód ďalej ovplyvňuje priebeh realizácie. Táto sekcia čerpá informácie zo zdrojov [15] a [1].

Výskum od stolu

Začiatkom takmer každého výskumu je výskum od stolu (Desk research). Ide o takzvaný sekundárny výskum, čo znamená že získavame už existujúce dáta alebo spoznávame podobné existujúce produkty.

Úvodom tohto výskumu je jasná definícia i špecifikácia problému a vytýčenie si vecí, ktoré ma zaujímajú. Potom prehľadávam dostupný materiál a zapisujem si z neho dôležité poznatky. Materiál na tento výskum je k dispozícii v knižniciach, na internete alebo nám ho môže poskytnúť organizácia zainteresovaná vo výskume. Samotný zdroj poznatkov by však mal byť vždy dôkladne analyzovaný, aby sme overili jeho aktuálnosť a spoľahlivosť. Zozbierané dáta sa analyzujú a sú zhrnuté do výslednej správy, ktorá obsahuje najdôležitejšie poznatky. Pomocou výskumu od stolu je možné tiež zmapovať trendy v obore alebo konkurenciu.

Sekundárny výskum je lacnejší a aj rýchlejší než primárny výskum, pri ktorom sa údaje zbierajú priamo od užívateľov.

Rozhovor

Rozhovor (Interview) je kvalitatívna výskumná metóda na zbieranie dát. Vďaka nemu vieme získať detailnejší užívateľský pohľad na problém a lepší pohľad na užívateľa. Rozhovor môže mať formu telefonického hovoru, ale výhodnejšie je osobné stretnutie, pri ktorom je možné sledovať reč tela. Výskumník by mal mať dopredu pripravené otázky alebo scenár rozhovoru. Nepriaznivá stránka rozhovoru je, že častokrát zabere množstvo času.

Dotazník

Dotazník poskytuje rýchly zber veľkého množstva informácií, preto ho radíme ako kvantitatívnu metódu zbierania dát. S jeho pomocou vieme širšie charakterizovať cieľovú skupinu a spoznať názory a postoje ľudí. Dotazník môže mať papierovú alebo online formu. Pri tejto metóde je veľmi dôležité si vytvoriť užitočné otázky aby sme vo výsledku dostali prospešné informácie.

Focus group

Focus group je moderovaná diskusia cieľových užívateľov, ktorá využíva skupinovú dynamiku. Vďaka nej získame od užívateľov ich postoje, túžby a názory. Debata, ktorá vznikne v skupine, môže ukázať hlbšiu príčinu problému alebo úspechu, o ktorej sme nevedeli. Dôležitou súčasťou focus group je skúsený moderátor, ktorý usmerňuje debatu.

3.3 Analytické metódy

Analytické metódy sa zameriavajú na dáta získané pri výskume. Množstvo získaných informácií je potrebné roztriediť podľa toho, čo definujú. Z roztriedených informácií si potom vyberáme len tie poznatky, ktoré sú prospešné pre vývoj. Táto časť vychádza z informácií získaných zo zdrojov [1] a [15].

Priority

Kľúčovou analytickou metódou je stanovenie si priorít. Problémy a požiadavky musia byť rozdelené podľa dôležitosti. Nie je možné vynakladať všetkým problémom rovnaký čas a úsilie.

Užívateľský scenár

Užívateľský scenár (User flow), je v podstate popis cesty. Túto cestu užívateľ musí prekonať, keď chce vyriešiť určitý problém. Na vytvorenej ceste sa následne analyzuje, či sú všetky jej kroky potrebné, logické, viditeľné a či užívateľ chápe prečo ich vykonáva.

Strom problémov

K špecifikácií problému a jeho rozdelení na podproblémy môžeme využiť analytickú metódu strom problémov. Venuje sa identifikovaniu príčin a dôsledkov problému. Keď je problém bližšie určený, tak sa snažíme nájsť možné riešenie.

Persony

Persony sú modelovaný užívatelia produktu. Vytvárajú sa na základe získaných informácií o cieľovej skupine užívateľov. Slúžia na lepšie predstavenie skutočného užívateľa pri návrhu produktu. Predstavovanie si užívateľa pomáha pri posudzovaní cieľov, želaní a usmerňovaní rozhodovania, aby bol výsledný produkt pre neho lepšie prispôbený.

3.4 Návrhárske metódy

Na základe získaných informácií z výskumu je možné vytvoriť návrh produktu. Návrh predstavuje funkčnosť, použiteľnosť a vizuálnu stránku produktu. Pri tvorbe návrhu pomáhajú rôzne metódy vývoja riešenia a prototypovania. Informácie o jednotlivých metódach som získal zo zdrojov [1] a [15].

Sketching

Najjednoduchšou metódou pre navrhovanie je sketching, teda skicovanie. Všetko, čo k tomu potrebujeme, je iba papier a pero. Sketching umožňuje dizajnérovi vyskúšať si mnoho nápadov predtým, než sa rozhodne pre jeden konkrétny. Keď je nápad zakreslený, dá sa nad ním lepšie premýšľať, preskúmať ho a objavovať potenciálne vylepšenia. Prípadné zdielanie vytvoreného sketchu s niekým, pomáha pri získavaní spätnej väzby na návrhársky koncept.

Wireframe

Wireframe zobrazuje základné rozloženie elementov a obsahu na webovej stránke. Ukazuje nám, ako budú jednotlivé elementy spolu fungovať. Nazýva sa aj ako „drátený model“, pretože neobsahuje grafické prvky a najčastejšie ide iba o čiernobiely návrh.

Mockup

K vizualizácií a testovaniu produktu nám slúži mockup. V porovnaní s wireframe už zobrazuje celkovú grafiku a dizajn. Mockup používajú dizajnéri hlavne k získaniu spätnej väzby od užívateľa na vzhľad. V prípade, že mockup poskytuje aj jednoduchú funkčnosť, môžeme hovoriť o prototypu.

Card Sorting

Pokiaľ potrebujeme zorganizovať väčšie množstvo informácií do výslednej štruktúry, pomôže nám s tým metóda card sorting. Táto metóda pomáha vytvoriť alebo vylepšiť informačnú architektúru webovej aplikácie s pomocou užívateľov. Užívateľom sú poskytnuté kartičky s pojmy, ktoré má roztriediť do kategórií. Pri triedení je užívateľ sledovaný, komunikuje sa s ním a snažíme sa pochopiť jeho premýšľanie nad informáciami.

3.5 Metódy testovania a validácie

V priebehu celého vývoja by sa malo uskutočňovať testovanie. Jeho úlohou je určiť defekty toho čo testujeme. Otestované môže byť všetko, napríklad požiadavky, návrh, prototyp a finálny produkt. Testovanie je uskutočňované rôznymi metódami podľa toho, čo, s kým a ako testujeme. Nasledujúca podkapitola čerpá zo zdrojov [1] a [15].

Testovanie atraktivity vzhľadu

Pri testovaní atraktivity vzhľadu ide hlavne o zistenie toho, ako produkt pôsobí na užívateľa. Zameriava sa na to, či užívateľ hneď pochopí zmysel stránky. Toto testovanie sa dá vykonávať už na prvých návrhoch, aby sa rýchlo overila navrhnutá štruktúra. Jedná sa o nemonerovanú metódu.

Užívateľské testovanie

Jedna z najdôležitejších a najčastejšie používaných metód pri testovaní je užívateľské testovanie. Zisťuje sa s ním, ako užívateľ používa vytvorený produkt. Sledovanie je cielené na to, či užívateľ natrafí na nejaký problém, či niečomu nechápe alebo či mu niečo chýba. Užívateľ takto môže objaviť problém, ktorý bol doteraz skrytý. Toto testovanie je časovo náročnejšie, ale účinné. Je výhodné ho nasadiť už pri vývoji aplikácie, aby bol vývoj usmerňovaný správnym smerom. Užívateľské testovanie nám v konečnom dôsledku môže ušetriť veľa času, financií a môže spraviť produkt úspešnejším.

A/B testovanie

A/B testovanie je možné aplikovať v prípade, že sú k dispozícii dve verzie riešenia problému. Varianty sú testované na užívateľoch a počas testovania sa sledujú zvolené ciele. Následne sa verzie vedú porovnať a vyhodnotiť, ktorú je výhodnejšie použiť.

Eye tracking

Ďalšou formou testovania je Eye tracking, teda sledovanie pohybu očí. Špeciálne zariadenie meria, kde presne užívateľ pozerá pri plnení úloh. Výstupom metódy sú mapy, ktoré znázorňujú kam užívateľ najčastejšie upiera svoj zrak. Tento spôsob testovania je však finančne nákladnejší, pretože je potrebná špeciálna technika.

Kapitola 4

Analýza a návrh riešenia

Aké problémy by mal nový systém riešiť? Kto bude so systémom pracovať? Aké funkcie bude systém ponúkať? To sú niektoré z otázok, ktorých odpovede musíme spoznať pred začiatkom návrhu aplikácie. Zo zistených odpovedí sa následne zostavia požiadavky na systém. Po získaní a analyzovaní informácií sa môže pristúpiť k návrhu systému.

Táto kapitola sa venuje analýze informácií a návrhu riešenia. V prvej podkapitole predstavím cieľ práce. Nasledovať bude definovanie cieľových užívateľov. Tretia podkapitola bude venovaná existujúcim riešeniam. Ďalšia časť bude špecifikovať požiadavky, ktoré by mal systém spĺňať. V piatej podkapitole predstavím prípady použitia vyvíjanej aplikácie. V záverečných dvoch častiach sa budem venovať návrhu grafického rozhrania a dátového modelu.

4.1 Cieľ práce

Cieľom tejto práce je vytvoriť návrh systému, ktorý by študentom zjednodušil iniciovanie spolupráce na projektoch. Návrh systému bude overený, testovaný a prezentovaný vo forme prototypu webového portálu.

Dôraz práce sa kladie na užívateľskú spokojnosť pri práci so systémom. Zameriavam sa na jednoduchosť používania a aby sa v ňom užívateľ vyznal. Užívateľovi musí byť bez zbytočného premýšľania jasné, čo má urobiť v systéme, aby dosiahol želaného výsledku. Podstatná je aj prehľadnosť a prezentácia všetkých poskytovaných informácií. Priaznivú užívateľskú spokojnosť sa snažím doceliť využitím rady UX metód počas vývoja aplikácie.

Hlavnou úlohou systému, ktorý mám navrhnúť, je vytváranie medziodborových tímov. Systém sa zameriava na ulahčenie procesu nájdenia spolupracovníkov a na vyhľadávanie projektu. Portál by mal poskytovať základné informácie o projektoch, ktoré budú zjednodušovať rozhodovanie užívateľa podieľať sa na projekte.

Námet tejto práce je čerpaný z bakalárskej práce Nely Klímovej, ktorá sa na fakulte výtvarných umení venovala téme študentskej spolupráce [6]. Z tejto práce som prebral aj názov aplikácie UNI.TEAM.

4.2 Cieľový užívateľia

Ako ďalšie som si definoval skupinu cieľových užívateľov, ktorí budú navrhovaný systém používať. Webový portál je primárne určený študentom univerzity. Táto skupina je vekového rozhrania približne 18 až 26 rokov. Môžeme u nej počítať minimálne so základnými znalos-

tami používania informačných technológií. U cieľovej skupiny je veľmi populárne používanie smartphonov, tabletov a ďalších podobných zariadení s rôznou veľkosťou obrazovky. Vytváraný portál by mal preto určite obsahovať responzívny dizajn, aby sa zachovala užívateľská priateľnosť aj pre tieto zariadenia.

Typickým užívateľom je študent s nápadom na projekt, ktorý by ho rád zrealizoval. Tento študent potrebuje ľudí, ktorý by mu pomohli, a preto príde na webový portál. Na portáli si vytvorí svoj projekt a čaká, kým sa mu niekto ozve. Ďalším typickým užívateľom je študent, ktorý má záujem o prácu na projekte. Na portáli si vyhľadáva projekt, ktorý ponúka pre neho zaujímavú pozíciu a vyjadrí záujem o túto pozíciu.

Malú časť užívateľov môžu tvoriť aj učitelia alebo iná osoba, ktorá však musí mať povolenie k používaniu. Táto skupina je ale minimálneho zastúpenia, a preto sa prioritne zameriavam na študentov.

Na základe definície a analýzy cieľových užívateľov som vyrobil 2 osoby typických užívateľov. Persóny uvádzam v prílohe B tejto práce.

4.3 Existujúce riešenia

Pre úspešný návrh systému je výhodné získať poznatky o tom, ako študenti iniciujú spoluprácu medzi sebou v súčasnosti. Týmito spôsobmi sú najčastejšie facebookové skupiny jednotlivých fakúlt alebo internátov a tiež fórum univerzity. Študenti uverejnia oznam, že hľadajú spolupracovníkov s určitými schopnosťami. Záujemcovia sa im ozývajú pomocou komentárov alebo osobných správ. Tieto metódy sú však veľmi nepraktické. Vo facebookových skupinách je veľké množstvo informácií. Statusy s ponukami práce na projekte sa veľmi rýchlo stratia z dohľadu. Študent, ktorý ponuku vytvoril, nemá jednoduchý prehľad kto všetko má záujem. Univerzitné fóra už zase majú svoje najlepšie roky za sebou. Sú to centrá množstva informácií, ktoré sa často stávajú neprehľadné. Množstvo študentov o fórach nich ani nevie.

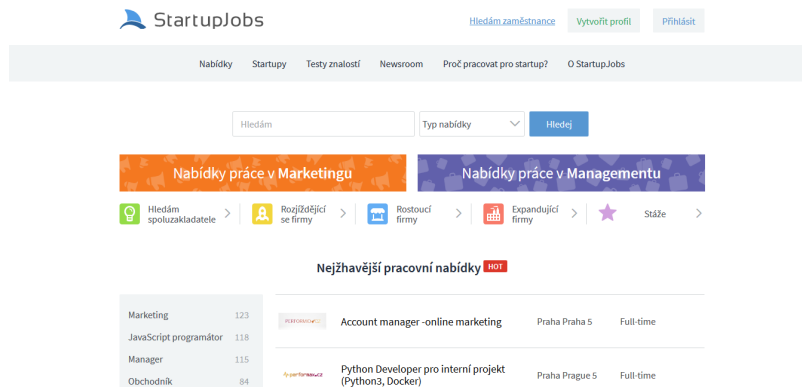
Skúsil som tiež vyhľadať či neexistuje nejaký systém, ktorý sa snaží tiež riešiť daný problém. Takýto systém by som mohol použiť na získanie inšpirácie a dôležitých informácií, čo spraviť lepšie. Rovnako zameraný portál som ale pri hľadaní nenašiel. Existuje však mnoho portálov, ktoré sa zameriavajú na hľadanie zamestnancov alebo členov startupových tímov. Táto idea je dosť podobná so zámerom, ktorý má mnou vytváraná aplikácia. Rozhodol som sa preto využiť podobné portály pre čerpanie inšpirácie a predstavy. Takýmito webovými portálmi sú napríklad startupjobs.cz alebo founder2be.com.

Pri analýze podobných existujúcich portálov som sa zameril na základné prvky, ktorými tieto portály disponujú. Takými prvkami boli vyhľadávacie elementy, náhľady ponúk, profily ponúk.

startupjobs.cz¹

StartupJobs je webový portál, ktorý pomáha s hľadaním zamestnancov do startupov. Design aplikácie je v jednoduchom a peknom prevedení. Webová aplikácia poskytuje zoznam so startupmi alebo pracovnými ponukami. Z pohľadu UX sa mi nepozdávajú náhľady v zoznamoch. V náhlade je minimum informácií. Pre veľa podstatných informácií musím otvárať profily, čo spôsobuje veľa zbytočného klikania.

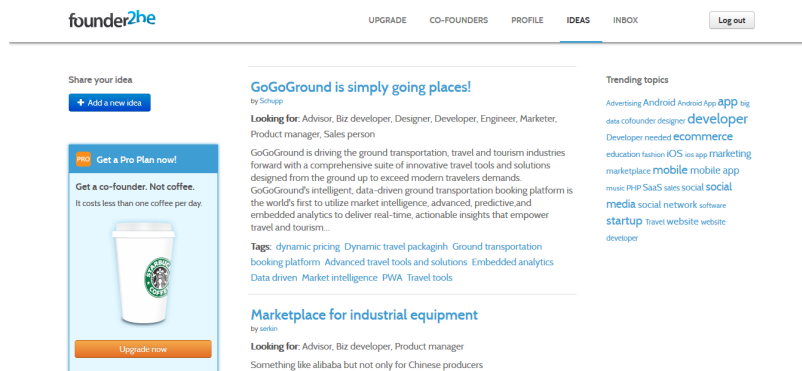
¹<https://www.startupjobs.cz/>



Obr. 4.1: Snímka webovej aplikácie startupjobs.cz

founder2be.com²

Founder2be je webový portál, v ktorom si užívateľ môže vytvoriť nápad a hľadať naň pracovníkov. Dizajnovo je aplikácia veľmi jednoduchá a prehľadná. Pri prezeraní som však našiel niekoľko nedostatkov, ktorým by som sa chcel vyvarovať. Pri zozname nápadov som nenašiel nejaký spôsob filtrácie a panel na vyhľadanie je schovaný až nižšie na stránke, čo si užívatelia ani nemusia všimnúť. Páči sa mi, že hneď v náhlade projektu je popis nápadu. Viem sa teda hneď rozhodnúť či ma nápad zaujal. Veľké mínus ale vidím aj v tom, že táto stránka neposkytuje nejakú formu kontaktovania autora. Profil nápadu neposkytuje ani žiadny kontakt, iba link na web nápadu.



Obr. 4.2: Snímka webovej aplikácie founder2be.com

4.4 Špecifikácia požiadaviek

Pre správny návrh systému si musím určiť základné požiadavky a priority, ktoré systém musí spĺňať. Na základe sledovania riešenia problému spolupráce v súčasnosti, analýzy existujúcich riešení a rozhovorov s potenciálnymi užívateľmi som dospel k nasledujúcim požiadavkám a poznatkom.

²<https://www.founder2be.com/>

Správa projektu a vyjadrenie záujmu o pozíciu

Užívateľ, ktorý má nejaký nápad na projekt, ho potrebuje uverejniť. Musí byť schopný vytvoriť projekt, čo znamená do systému zadať údaje o projekte, ktorými sú názov, popis, pozície a plánovaný začiatok. O každej pozícií musí byť tiež evidovaný názov, typ a popis. Tieto údaje by malo byť možné autorom projektu kedykoľvek zmeniť. Keďže hlavným cieľom je nájsť pracovníka na pozíciu, každá voľná pozícia musí poskytovať zoznam prihlásených záujemcov a možnosť výberu z nich. Aby mohol užívateľ získať informácie o projekte, tak musí systém obsahovať profil projektu.

Keď užívateľa zaujme určitý projekt a pozícia na ňom, bude môcť vyjadriť záujem o pozíciu. Vyjadrenie záujmu znamená, že dotýčny užívateľ chce pracovať na danej pozícií. Následne bude musieť užívateľ počkať, či si ho autor projektu vyberie.

Jednotlivé projekty môžu byť v rôznych fázach a preto bude systém definovať niekoľko stavov projektu.

- **Neschválený** – Stav, keď je projekt vytvorený, ale je ešte neschválený užívateľom.
- **Aktívny** – Projekt bude v tomto stave keď bude schválený, bude mať voľné pozície a zároveň nevypršala doba, kedy sa naplánoval začiatok práce. Užívateľia sa budú môcť hlásiť na pozície takéhoto projektu.
- **Bežiaci** – Stav keď má projekt obsadené všetky pozície, čiže sa reálne začína práca na projekte.
- **Expirovaný** – Projekt sa dostane do tohto stavu, keď sa do zadaného začiatočného dátumu práce nestihnú naplniť všetky pozície. V tomto stave sa bude dať projekt vymazať alebo posunúť dátum začiatku prác a hľadať ďalej.
- **Skončený** – Tento stav znamená, že je ukončená práca na projekte.

Zmenu medzi bežiacim a skončeným stavom by mal vykonávať užívateľ. Zvyšné zmeny budú automaticky riadené systémom.

Vyhľadávanie projektu a vyjadrenie záujmu o pozíciu

Podstatnou požiadavkou je možnosť vyhľadať si projekt. Systém preto potrebuje obsahovať katalóg projektov, ktorý sa bude dať rôzne filtrovať a vyhľadávať. Vyhľadávanie by malo prebiehať minimálne podľa názvu alebo časti názvu projektu. Keďže projekt primárne hľadáme kvôli pozíciám, určite by pri filtrovaní nemal chýbať výber projektov s určitou pozíciou. Pre prehľadnosť svojich projektov by aplikácia mala poskytovať aj zoznam vlastných projektov a zoznam projektov o ktoré mám záujem.

Vyhľadávanie osoby a odporúčenie

V prípade, že autor projektu nemá žiadnych záujemcov alebo potrebuje rýchlo nájsť záujemcu, tak systém bude poskytovať možnosť odporúčenia svojej pozície iným užívateľom. Všetci užívateľia sa budú nachádzať v zozname, v ktorom sa bude dať vyhľadávať a filtrovať.

Komunikácia medzi záujemcom a autorom

Medzi záujemcom a autorom projektu musí prebiehať aj určitá forma komunikácie. Táto komunikácia by mala užívateľovi uľahčiť výber jedného zo záujemcov o pozíciu. Jej zmyslom je iba výmena základných informácií o záujme a nie akási sociálna sieť na pravidelné komunikovanie. Ďalšia komunikácia o práci na projekte a iné, by už mala prebiehať pomocou mailu, stretnutia alebo sociálnych sietí. Jednou z možností ako zaistiť aby komunikácia slúžila iba k záujmu o pozíciu, je možnosť komunikovania iba medzi autorom a záujemcom. Každá komunikácia by tak mala byť započatá záujmom o pozíciu. Aby mal užívateľ jednoduchý prehľad, ktorá správa sa zaoberá akou pozíciou, tak by sa jednotlivé správy mali viazať na pozíciu a na projekt. Záujemcovia budú často posielať svoje životopisy, a preto by mal systém poskytovať jednoduchú formu priradovania životopisov ku správe.

Dozor

Dostal som tiež požiadavku, aby sa do systému mohol registrovať aj neštudent univerzity. Neštudenti by však mali byť niekym schválení. Preto by mal mať systém rolu takzvaného supervízora, ktorá by sa starala o schvalovacie procesy. Okrem študentov mimo univerzity by mal supervízor schvalovať aj každý pridaný projekt. Toto opatrenie by mal systém obsahovať, aby bola prevádzaná kontrola a dohľad nad projektami. Cieľom kontroly bude zamedziť obsahu nevhodných a vulgárnych projektov. Pokiaľ by supervízor projekt neschválil, tak by sa vymazal zo systému.

Prehľadnosť zmien na projekte

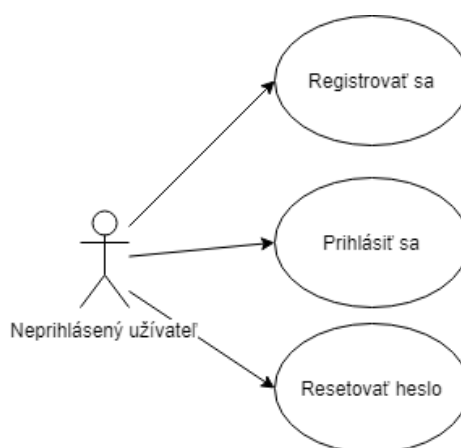
Aby mali užívatelia aktuálny a rýchly prehľad, tak bude webový portál poskytovať upozornenia. Upozornenia by mali užívateľov oznamovať udalosti alebo zmeny, ktoré boli vykonané. Systém by mal poskytovať nasledujúce notifikácie.

- **Schválený projekt** – Notifikácia oznamuje, že projekt bol schválený.
- **Neschválený projekt** – Notifikácia oznamuje, že projekt nebol schválený.
- **Potvrdenie Osoby** – Oznamuje, že nejaká osoba mimo univerzity žiada o potvrdenie registrácie. Notifikácia pre supervízora.
- **Schválenie Osoby** – Oznamuje, že osoba mimo univerzity, ktorá žiadala o potvrdenie registrácie, bola schválená. Táto notifikácia by mala byť danej osobe zaslaná na mail.
- **Potvrdenie projektu** – Oznámenie o tom, že užívateľ žiada o schválenie projektu. Notifikácia pre supervízora.
- **Výber** – Táto notifikácia ohlasuje, že bol užívateľ vybraný na pozíciu.
- **Zrušenie výberu** – Notifikácia, že bol zrušený výber užívateľa na pozíciu.
- **Záujem** – Ohlásenie vytvorenia záujmu na pozíciu z projektu.
- **Zrušenie záujmu** – Oznámenie zrušenia záujmu na pozíciu.
- **Správa** – Oznámenie o doručenej správe.
- **Odporúčenie** – Oznamuje, že nejaký užívateľ mi odporučil svoju pozíciu.
- **Expirácia projektu** – Oznámenie o tom, že sa projekt dal do stavu expirácie.

4.5 Prípady použitia

Užívatelia, ktorý navštívia webový portál môžu byť prihlásený alebo neprihlásený. Prihlásených užívateľov môžeme rozdeliť do troch užívateľských rolí. Na základe rolí je určený rozsah ich činností, ktoré môžu v systéme robiť.

Neprihlásený užívateľ



Obr. 4.3: Diagram prípadu užitia pre neprihláseného užívateľa

Každý užívateľ, ktorý chce využívať aplikáciu musí byť registrovaný. Neregistrovaný užívateľ môže jedine vykonať registráciu. Neprihlásený užívateľ, ktorý je už registrovaný má možnosť prihlásenia alebo resetovania hesla. Ku ďalším činnostiam sa užívateľ dostane až po prihlásení.

Prihlásený užívateľ - bežný užívateľ

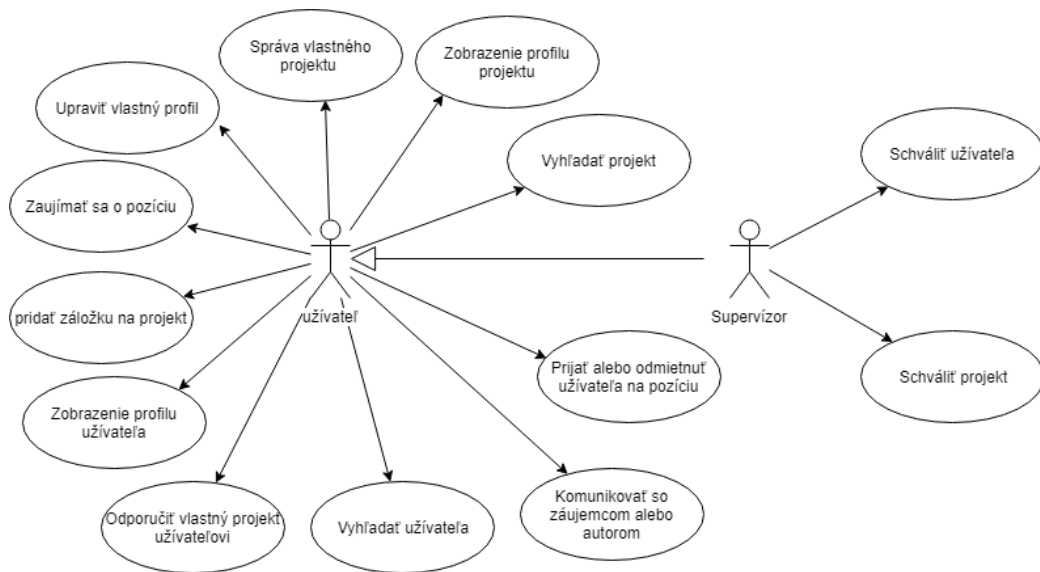
Každý prihlásený užívateľ uvidí zoznam všetkých projektov. V zozname je umožnené vyhľadávanie a filtrovanie. Taktiež má užívateľ právo vidieť profil vybraného projektu. Na voľnú pozíciu, ktorá užívateľa zaujme vie vyjadriť svoj záujem.

Prihlásenému užívateľovi je umožnené vytvoriť nový vlastný projekt. Tento projekt potom môže upravovať alebo odstrániť. Autor projektu si vie zobrazíť všetkých záujemcov o pozície a vie tiež pridať jedného záujemcu na vybranú pozíciu. Autor so záujemcami majú schopnosť komunikovať medzi sebou.

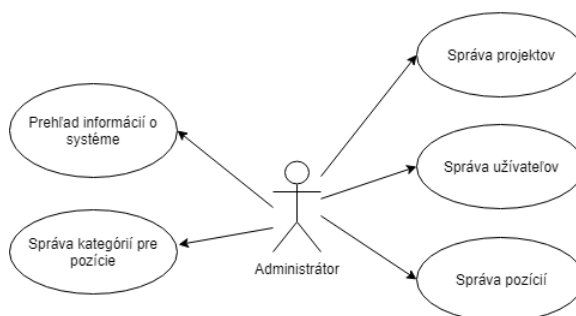
Užívateľ, ktorý je prihlásený do systému, uvidí aj zoznam všetkých užívateľov a profily jednotlivých užívateľov. Vlastný profil si užívateľ môže upravovať. Každému užívateľovi môže autor projektu poslať odporúčanie na pozíciu z vlastného projektu.

Prihlásený užívateľ - supervízor

Užívateľ prihlásený ako supervízor môže vykonávať všetky činnosti presne ako bežný užívateľ. Navyše má akúsi dozornú moc v prípade ak je vybraný ako supervízor pre projekt alebo užívateľa mimo univerzity. V takomto prípade schvaľuje projekt alebo registráciu užívateľa.



Obr. 4.4: Diagram prípadu užívania pre bežného užívateľa a supervízora



Obr. 4.5: Diagram prípadu užívania pre administrátora

Prihlásený užívateľ - administrátor

Administrátor nepristupuje k systému ako bežný užívateľ. Pre systém má pozíciu správcu. Disponuje základnými štatistikami o systéme. Dokáže spravovať akékoľvek informácie o projektoch, pozíciách či užívateľoch. Pri užívateľoch tiež určuje ich rolu v systéme. Ďalšou z jeho činností je vytváranie, upravovanie alebo odstraňovanie kategórií, do ktorých sa triedia pozície.

4.6 Návrh grafického užívateľského rozhrania

Grafické rozhranie je časť aplikácie s ktorou priamo pracuje užívateľ. Pomocou rozhrania sa ovláda celá aplikácia a pristupuje sa k uloženým dátam. Môžeme povedať, že táto časť najviac ovplyvňuje užívateľský zážitok. Práve preto, by rozhranie malo byť prispôbené činnosti užívateľa. Jacob Nielsen počas výskumu zistil, že až 79% testovacích užívateľov si web nečíta po slovách, ale stránku len skenujú [11]. Pre dobrý užívateľský zážitok by stránka mala byť prispôbená na to, že ju užívateľ bude len skenovať. Toto prispôbenie pozostáva napríklad z rôznych úrovní nadpisov, zvýraznenia kľúčových informácií a členenia

textu do odstavcov. Všetky tieto vlastnosti som sa snažil zakomponovať aj do návrhu mojej aplikácie.

Skicovanie

Priebeh návrhu grafického rozhrania vytvárajúcej aplikácie som začal skicovaním. Ako som už uviedol v časti 3.4 je to najjednoduchšia metóda návrhu a podporuje v rozvíjaní predstavivosti. Na papier som si vytvoril náčrt základných častí rozhrania. Tým som si ujasnil štruktúru a základné prvky. Načrtnuté boli rôzne verzie systému. Z týchto verzií som si vybral jeden, ktorý sa mi nakoniec najviac pozdával.

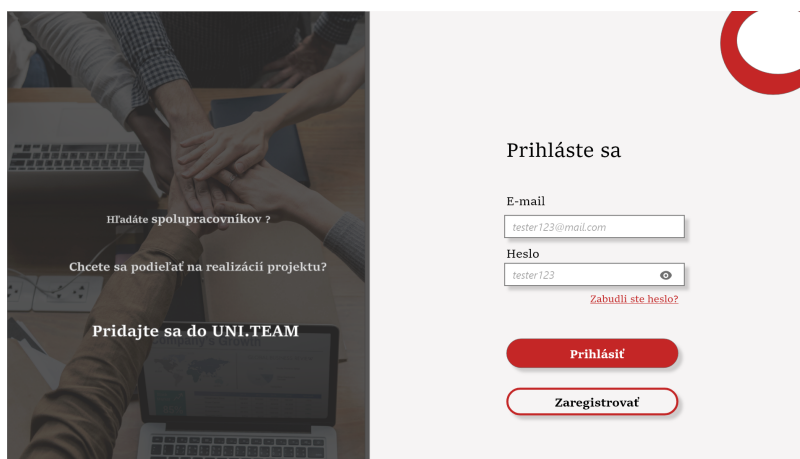
Mock up

Na základe náčrtu som si vytvoril mock up aplikácie. Mock up som tvoril pomocou programu Adobe XD³. Do mock upu som pridal aj jednoduchú funkčnosť na preklikanie stránok, takže výstupom návrhu grafického rozhrania bol jednoduchý prototyp. Prototyp bol vytvorený v slovenčine ale výsledná aplikácia bude v angličtine. Teraz si ukážeme a opíšeme jednotlivé časti navrhnutého rozhrania.

Úvodná stránka

Užívateľa na úvodnej stránke nechceme otravovať príliš veľa informáciami a grafickými prvkami. Pri návrhu som sa práve preto snažil o jednoduchosť stránky. Úvodná stránka musí ozrejmiť čo to je a na čo to je. Vysvetlenie pomocou dlhého textu užívateľa skôr odoženie, a preto som zvolil iba krátke vety. Tieto vety naznačujú funkciu aplikácie. Stránka je doplnená obrázkom na ktorom je podanie rúk. Tento symbol dotvára celkové zmysľovanie užívateľa čomu sa aplikácia venuje.

Ďalšou úlohou úvodnej stránky je naviesť užívateľa na prihlásenie alebo registráciu do aplikácie. Na pravej strane stránky je umiestnený prihlasovací formulár a potvrdzovacie tlačidlo. Pod prihlasovacím tlačidlom je umiestnené tlačidlo, ktoré užívateľa presmeruje na stránku registrácie.



Obr. 4.6: Mock up úvodnej stránky aplikácie

³<https://www.adobe.com/sk/products/xd.html>

Menu

V hornej časti je umiestnené navigačné menu. Menu sa bude zobrazovať na každej stránke po prihlásení. Pri posúvaní dole, sa bude panel skrývať. Menu sa objaví vždy na vrchu stránky alebo pri pohybe nahor. Navigačné menu sa skladá z niekoľkých prvkov. Na ľavej časti menu je názov UNI.TEAM odkazujúci na hlavnú stránku po prihlásení, ktorou je zoznam projektov. Neďaleko názvu sú odkazy na jednotlivé základné stránky aplikácie. Na pravej strane menu je rozbaľovacie menu označené menom prihláseného užívateľa. Odkazy v rozbaľovacom menu mieria na stránky, ktoré sú zamerané na komunikáciu a úpravu profilu užívateľa. Po ľavej strane rozbaľovacieho menu sa vyskytujú notifikačné ikony.

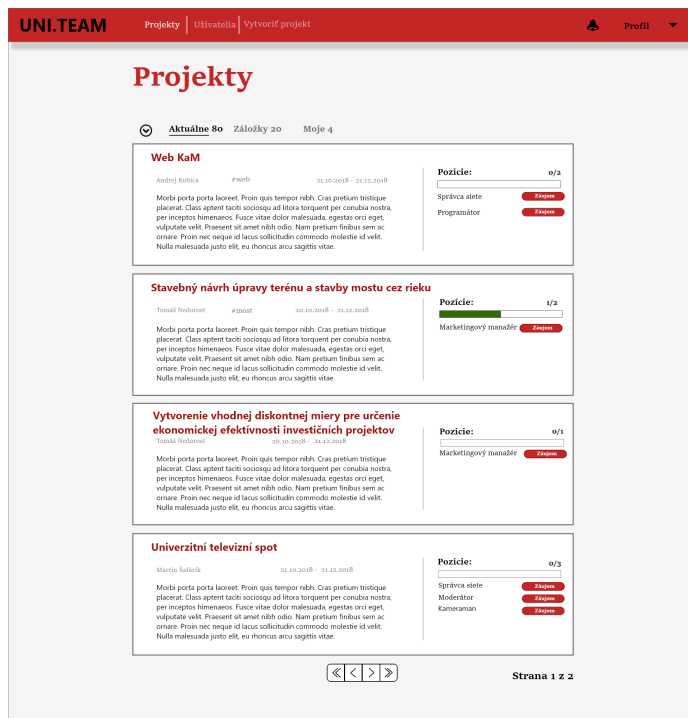


Obr. 4.7: Navigačného menu vybrané z Mockupu aplikácie

Zoznamy

Zoznamy, či už projektov alebo užívateľov sú rozdelené na 2 časti. Prvou je filtrovacia časť, ktorá určuje aké projekty sa budú zobrazovať v obsahovej časti. Hlavné prepínanie medzi aktuálnymi, vlastnými, založenými prípadne supervízorskými projektami alebo všetkými a supervízorskými užívateľmi je dostupné stále. Po kliknutí na ikonu šípky sa nám zobrazí panel s vyhľadávacím oknom a elementy pre výber zoradovania a filtrovania.

Pod prvkami filtrovania nasleduje obsahová časť, zložená z náhľadov na projekty alebo užívateľov.



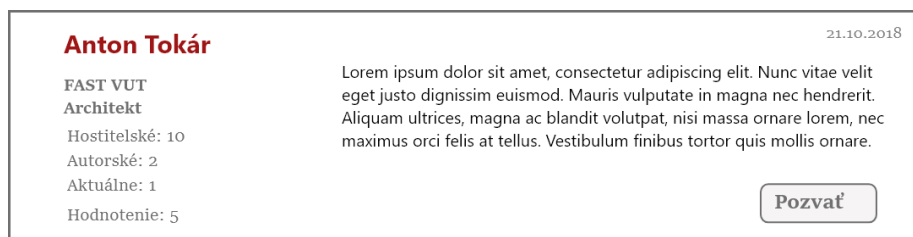
Obr. 4.8: Pohľad na zoznam projektov z Mockupu

Náhľad na projekt je tvorený jednoduchým panelom, so všetkými dôležitými informáciami o projekte. Cieľom pri návrhu náhľadu bolo dosiahnuť, aby sa užívateľ vedel rozhodnúť, či ho ten projekt zaujal aj bez zobrazovania profilu projektu. Náhľad zobrazuje na pravej strane aj voľné pozície na projekt. Z náhľadu sa dá priamo prihlásiť na voľnú pozíciu pomocou tlačidla. Po vyjadrení záujmu na pozíciu, bude tlačidlo signalizovať, že už bol záujem prejavý. Prípadne sa dá záujem aj zrušiť. Voľné pozície sú doplnené grafickým znázornením pomeru voľných a obsadených pozícií.



Obr. 4.9: Náhľad na projekt z Mockupu

Náhľad na užívateľov sa zobrazuje tiež ako jednoduchý panel so základnými informáciami o užívateľovi. Medzi informáciami sú aj počet vytvorených projektov a počet projektov na ktorých pracoval. Jediné tlačidlo, ktoré obsahuje náhľad slúži k vytvoreniu odporúčania. Po kliknutí na toto tlačidlo sa zobrazí pop-up okno, v ktorom si užívateľ bude mať vybrať svoj projekt a pozíciu, ktorú chce odporučiť.



Obr. 4.10: Náhľad na užívateľa z Mockupu

Notifikácie

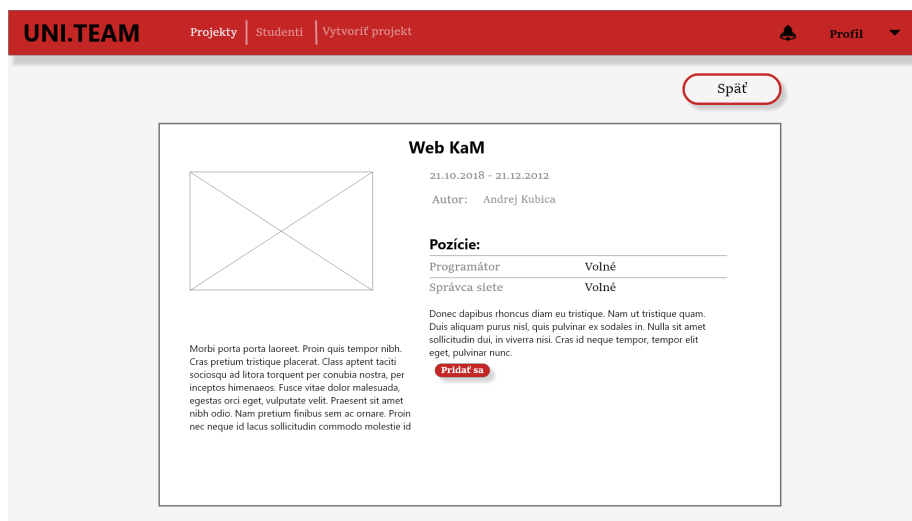
Notifikácie sa zobrazia po kliknutí na notifikačnú ikonu v menu. Rola supervízora obsahuje aj špeciálne supervízorské notifikácie. Bežné notifikácie budú rozdelené podľa projektov a supervízorské notifikácie podľa užívateľov. Notifikácie sú umiestnené po ľavej strane vo dvoch stĺpcoch. Prvý stĺpec zobrazuje projekty alebo užívateľov na ktoré mám notifikácie. Po kliknutí na jeden z prvkov v prvom stĺpci sa v pravom stĺpci zobrazia notifikácie. Notifikácia je navrhnutá ako jednoduché textové vysvetlenie, prípadne obsahuje aj tlačidlo k určitej akcii. Po kliknutí mimo dva notifikačné stĺpce sa notifikácie zatvoria.

Profil projektu

Profil projektu je tvorený jedným bielym elementom. Vo vrchnej časti sa nachádza názov projektu. Pod názvom sa profil delí na dve časti. Na ľavej strane je umiestnená titulná fotografia a pod ňou opis projektu. Pravú časť tvorí stav, dátumy vytvorenia a ukončenia

a meno autora. Pod menom autora sú vypísané názvy pozícií. V riadku za menom pozície je element, ktorý znázorňuje či je pozícia voľná.

V prípade, že ide o profil vlastného projektu, v pravom hornom rohu sú umiestnené ikony pre úpravu alebo zmazanie projektu. Tieto ikony sú umiestnené aj v riadku pri pozíciách. Nad prvou pozíciou je tiež umiestnená ikona plus pre pridanie pozície. Po kliknutí na akúkoľvek ikonu sa zobrazí pop-up okno buď s potvrdením zmazania alebo s formulárom pre úpravu či pridanie.



Obr. 4.11: Zobrazenie profilu cudzieho projektu z Mockupu

Profil užívateľa

Profil užívateľa pozostáva z informácií o užívateľovi a prehľade jeho projektov. Prehľad projektov je situovaný na pravej strane a skladá sa z náhľadov jednotlivých projektov, ktoré užívateľ vytvoril. Informácie o užívateľovi sú umiestnené v ľavej časti stránky na paneli. Tento panel obsahuje profilovú fotografiu užívateľa, jeho meno, email, fakultu, pozíciu a popis. Pod informáciami je tlačidlo pre odporúčenie na vlastný projekt, ktoré funguje rovnako ako pri náhľade.

Vytvorenie projektu

Grafické rozhranie vytvorenia projektu som navrhol ako jednoduchý formulárom. Vo formulári vyplňujem informácie o projekte. V spodnej časti je tlačidlo na pridanie pozície do projektu. Po stlačení na toto tlačidlo sa objaví pop-up okno s formulárom na vyplnenie údajov o pozícii. Na potvrdenie formuláru vytvorenia projektu slúži tlačidlo „Vytvoríť“.

4.7 Dátový model

Súčasťou fungujúceho webového portálu je aj databáza, v ktorej sú uschované dôležité dáta. Jednou z častí návrhu aplikácie je proces návrhu štruktúry a usporiadania dát. Jeho podstatou je vymedzenie dát, ktoré potrebujem ukladať. Pri vymedzovaní dát som vychádzal z predchádzajúcej špecifikácie systému. Z vybraných dát som vytvoril dátový model. Tento

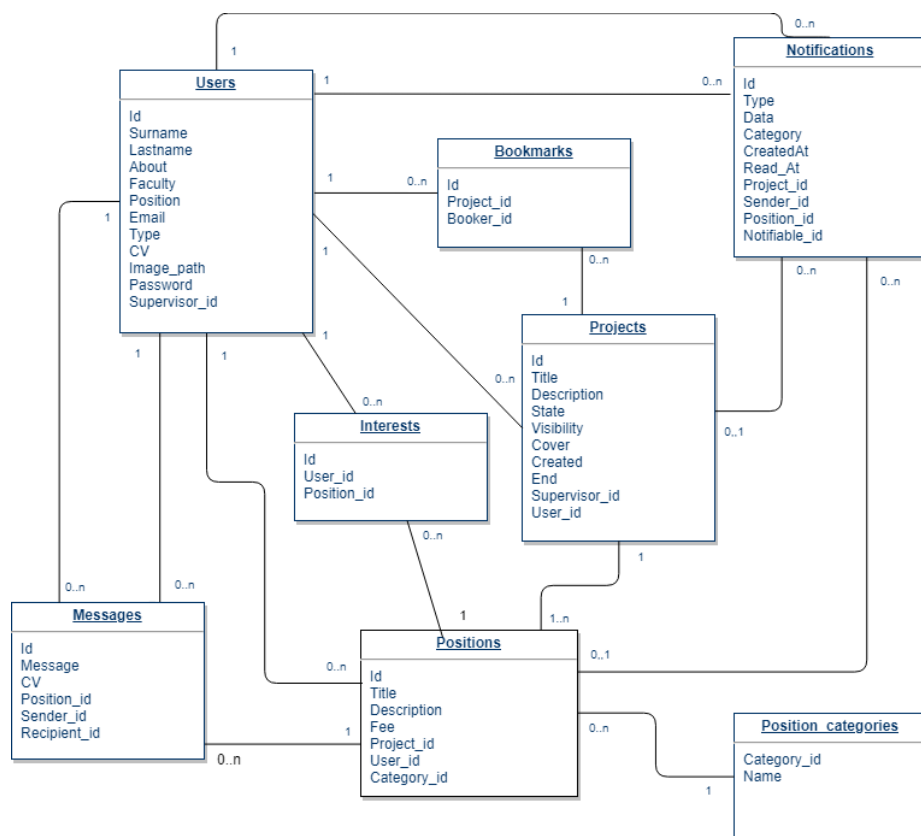
model je prezentovaný formou ER diagramu na obrázku 4.12, ktorý pozostáva z entitných množín a vzájomných vzťahov medzi nimi. Dátový model popisuje formát a štruktúru dát v systéme.

Popis entitných množín

- **Users** – Informácie o užívateľovi popisuje tabuľka Users. Užívateľ je identifikovateľný na základe id a emailu. O užívateľovi budem ukladať všetky údaje, ktoré vyplňa pri registrácii (meno, heslo, typ, email). Okrem týchto údajov si užívateľ v profile môže nastaviť aj ďalšie vlastnosti, ktoré tiež ukladám v tabuľke. Sú to fakulta, informácie o sebe a pozícia. Užívateľ má možnosť si nahrať profilovú fotku alebo životopis.
- **Projects** – V tejto tabuľke sa popisujú dáta objektu projekt. Projekt je jednoznačne identifikovateľný na základe id. O projekte budem evidovať názov, popis, autora pomocou jeho id a id supervízora. Pre projekt budem vedieť nahrať titulnú fotografiu. Táto tabuľka navyše obsahuje atribút state, ktorý definuje stav v ktorom sa projekt nachádza.
- **Positions** – Popisuje údaje o jednotlivých pozíciách z projektu. Obsahuje atribúty id čísla, názov, popis, odmena, id kategórie a id projektu, ktorého je pozícia súčasťou. Ďalej zaznamenáva id užívateľa, ktorý je na danú pozíciu vybraný. Pokiaľ ešte nie je nikto vybraný bude nastavená predvoľená hodnota 0.
- **Position Categories** – Táto tabuľka popisuje kategórie pozícií. Jediným atribútom je názov kategórie. Identifikovateľná je pomocou id.
- **Interests** – Väzobná entitná množina Interest reprezentuje záujem o projekt. Obsahuje identifikáciu projektu a užívateľa, ktorý sa o ten projekt zaujíma.
- **Notifications** – Popisuje informácie o notifikáciách. Atribútmi sú typ notifikácie, id notifikovaného užívateľa a dáta. Ďalej určuje id projektu a id pozície na ktoré sa notifikácia viaže, id užívateľa, kvôli ktorému bola vytvorená notifikácia a kategóriu. Kategória definuje 2 skupiny notifikácií, supervízorská a normálna notifikácia
- **Messages** – Táto entitná množina reprezentuje objekt správy. Vlastnosti, ktoré obsahuje sú id odosielateľa a prijímateľa, obsah správy a id pozície, na ktorú sa správa viaže. Navyše obsahuje aj atribút cv, ktorý nám hovorí o tom či je ako príloha poslaný životopis odosielateľa.
- **Bookmarks** – Definuje záložku na projekte. Atribútmi sú id užívateľa, ktorý si záložku vytvoril a id projektu, ktorý bol záložkou označený.

Popis vzťahov medzi entitnými množinami

Vzťahy medzi entitnými množinami sú nasledovné. Užívateľ môže vytvoriť projekt alebo prejavíť záujem o pozíciu. Projekt obsahuje určitý počet pozícií. Pozícia patrí do určitej kategórie. Na pozíciu môže byť vytvorených niekoľko záujmov alebo môže mať pozícia už priradeného pracovníka. Užívateľ tiež môže vytvoriť záložku, ktorá sa viaže na projekt. Správy sú vytvorené užívateľom a sú poslané inému užívateľovi. Každá správa sa musí viazať na určitú pozíciu. Notifikácie patria jednému užívateľovi. Notifikácia je reakciou na činnosť druhého užívateľa. Dá sa povedať, že druhý užívateľ je tvorca notifikácie. Notifikácie oznamujú udalosť na projekte alebo pozícií, preto sa viažu buď na pozíciu alebo na projekt.



Obr. 4.12: Dátový model

Kapitola 5

Implementácia

Keď som mal hotový návrh, pokračoval som implementáciou testovacej verzie aplikácie. V tejto kapitole si spravíme prehľad o tom, aké technológie som použil pri implementácii systému. Ďalej sa pozrieme tiež na adresárovú štruktúru výslednej aplikácie a povieme si niečo o jednotlivých častiach implementácie.

5.1 Použité technológie

Systém bol vytváraný vo vývojovom prostredí Visual Studio Code¹. Toto prostredie má podporu pre bežné webové technológie. Jeho veľkou výhodou je poskytnutie jednoduchšej práce s Javascriptom, ako napríklad automatické dopĺňanie kódu a farebné odlišovanie kódu.

Pre plnohodnotnú prácu s databázou som používal vývojové prostredie XAMPP². Toto prostredie poskytuje MySQL databázový systém a systém phpMyAdmin pre jeho jednoduchú správu. XAMPP bol tiež používaný na vytvorenie webového serveru pre vývoj a testovanie aplikácie. Ďalším nástrojom, ktorý som využíval pri implementácii, bol internetový prehliadač Chrome. Používal som ho pre zobrazovanie implementovanej aplikácie.

Frontendová časť aplikácie bola implementovaná pomocou HTML, CSS a Javascriptovej knižnice React. Pre zjednodušenie definovania štýlu a tvorbu responzívneho dizajnu bol použitý framework Bootstrap.

Na programovanie backendovej časti bol využitý jazyk PHP. Nepoužil som čisté PHP, ale jeho framework Laravel. Vďaka tomuto frameworku bol projekt implementovaný na princípe návrhového vzoru MVC. Laravel mi takisto poskytol vstavané prvky, ako napríklad princíp notifikácií, autentifikáciu a jednoduchú prácu s databázou. Implementačné technológie sú bližšie popísané v kapitole 2.

Výsledný webový portál je umiestnený na školskom webovom serveri. Je dostupný na adrese: <http://www.stud.fit.vutbr.cz/~xsvetl05/>

5.2 Použité cudzie časti

V niektorých prípadoch som pre zjednodušenie implementácie použil už existujúce komponenty a knižnice. Jednou z týchto cudzích častí je aj Bootstrap, ktorý som uviedol už aj v predchádzajúcej podkapitole. V tejto časti si ukážeme aj ďalšie.

¹<https://code.visualstudio.com/>

²<https://www.apachefriends.org/index.html>

React Router³

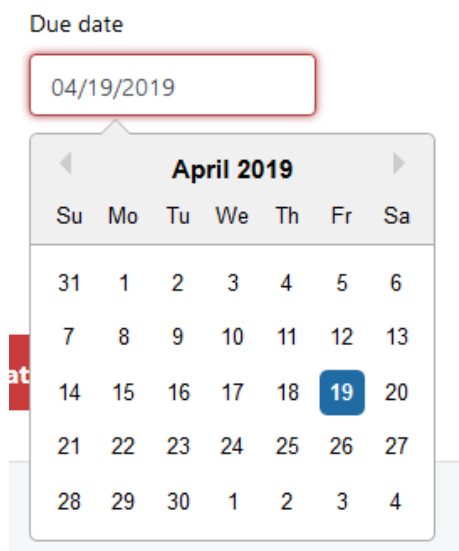
Moja aplikácia bola vytvorená ako jednostránková aplikácia. Pre zachovanie niektorých funkcií ako história prehliadača a spätného tlačidla bolo potrebné smerovanie. Knižnica React mi sama o sebe takúto funkcionálnosť neponúkala a preto som použil knižnicu React Router. React router je kolekcia navigačných komponentov, ktoré slúžia na smerovanie. Tieto navigačné komponenty sú použité miesto tradičných HTML odkazov `<a />`.

Axios

Axios je javascriptová knižnica, ktorú som aplikoval pre volanie HTTP požiadaviek. Použil som ju, pretože poskytuje jednoduchý spôsob vytvárania požiadaviek.

ReactJS Datepicker⁴

Komponent DatePicker som použil pre lepší užívateľský zážitok zo zadávania dátumu. Tento komponent poskytuje pole s dátumom. Po kliknutí na toto pole sa objaví prehľadný kalendár s výberom dátumu. Užívateľ je teda odľahčený od ručného zadávania dátumu.



Obr. 5.1: Výber dátumu ukončenia hľadania v aplikácii UNI.TEAM

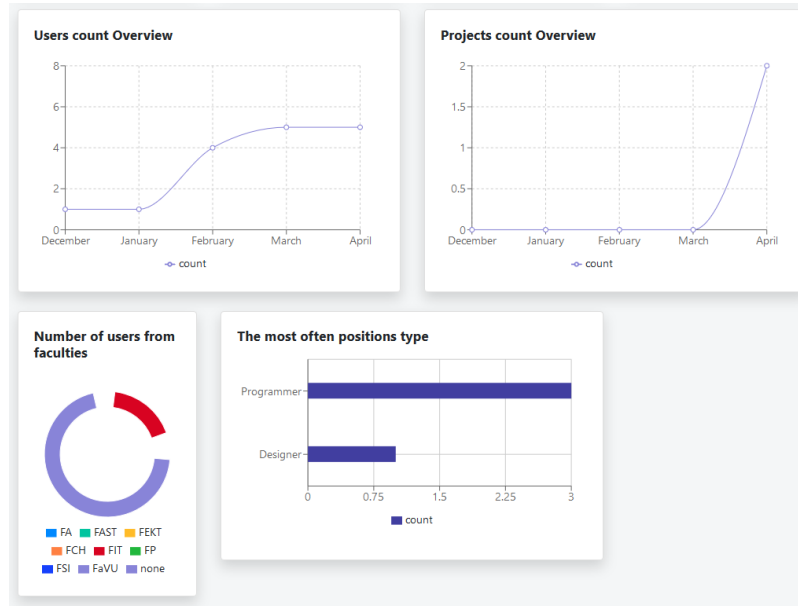
Recharts⁵

Recharts je knižnica pre tvorbu grafov. Je to ďalšia knižnica, ktorú som použil pre lepší užívateľský zážitok. Pomocou komponentov z Recharts som mohol vytvoriť rôzne grafy, ktoré som využil v administrátorskej verzii na zobrazenie štatistík.

³<https://reacttraining.com/react-router/>

⁴<https://reactdatepicker.com/>

⁵<http://recharts.org/en-US/>



Obr. 5.2: Využitie knižnice Recharts pre zobrazenie štatistík v aplikácií.

5.3 Adresárová štruktúra

Organizácia implementovaných súborov je definovaná frameworkom Laravel. Štruktúra je nasledovná:

- **app/** – Adresár obsahujúci hlavný kód aplikácie. Priamo v tejto zložke sú umiestnené modeli jednotlivých tabuliek.
- **app/Http/controllers** – Zložka obsahujúca všetky kontroléry, čiže ovládače.
- **app/notifications** – Adresár skrývajúci súbory jednotlivých notifikácií.
- **bootstrap/** – Táto zložka obsahuje súbor *app.php*, ktorý zavádza framework.
- **config/** – Tento adresár obsahuje všetky konfiguračné súbory.
- **database/** – zložka obsahuje migrácie na vytvorenie tabuliek v databáze
- **public/** – Hlavný adresár pre web. Adresár je prístupný z webu a obsahuje CSS súbory a obrázky.
- **resources/** – V tomto adresári je uložený pohľad a všetky React komponenty.
- **route/** – Definuje všetky pravidlá pre smerovanie v aplikácií.
- **storage/** – Táto zložka je určená na ukladanie súborov z aplikácie. Obsahuje tiež zkompilované súbory, ktoré generuje framework.
- **vendor/** – Adresár obsahuje knižnice, na ktorých je aplikácia závislá.

5.4 Dátové modely

Prvým krokom pri implementácii bolo vytvorenie databázy. Dátová vrstva bola vytvorená na základe návrhu v 4.7.

V MVC architektúre je dátová vrstva reprezentovaná modelmi. To znamená, že ku každej entitnej množine z navrhnutého ER diagramu som vytvoril model. Model predstavuje databázovú tabuľku v aplikačnom kóde a slúži pre interakciu s tabuľkami v databáze. Modeli sú umiestnené v adresári *app/*.

Databázové tabuľky som vytváral pomocou tzv. migrácií. Migrácie nám umožňujú efektívnu prácu so štruktúrou databázy. Pre každý model som si vytvoril migráciu, ktorá vytvorí tabuľku v databáze.

5.5 Kontroléry

Kontroléry majú na starosti komunikáciu medzi front-endom a back-endom. Komunikácia je zabezpečená službou REST a jej operáciami GET, POST, PUT a DELETE. Kontroléry obsahujú niekoľko metód, ktoré majú za úlohu ukladať, upravovať alebo mazať uložené dáta. Jednotlivé metódy sa volajú zadaním požiadavky na danú cestu. Každá metóda preto musí mať definovanú linku, cez ktorú bude komunikovať. Tieto linky sa definujú v súbore *route/api.php*.

Laravel obsahuje už preddefinované kontroléry pre autorizačné činnosti. Týmito činnosťami sú napríklad registrácia, prihlasovanie, odhlasovanie a podobne. Tieto kontroléry sú uložené v zložke *controllers/auth/*. Patrí sem aj kontrolér *HomeController*, ktorý sa stará o kontrolu prihlásenia. Pri svojej implementácii som tieto komponenty využil, ale vykonal som na nich drobné úpravy, aby zapadali do návrhu aplikácie. Ďalšie kontroléry som implementoval samostatne. Zoznam ďalších kontrolérov:

- **UserController**

Tento kontrolér zabezpečuje komunikáciu pre správu užívateľov. Metóda, ktorá nám vráti zoznam všetkých užívateľov, má názov *index()*. Táto metóda tiež zabezpečuje dátové triedenie, zoradovanie a vyhľadávanie, ktoré sú vykonávané databázovo. Pomocou parametrov je v metóde určené, ktorých užívateľov má zoznam obsahovať na výstupe. Metóda *show()* vracia dáta o užívateľovi a metóda *supervisor()* vracia zoznam supervízorov. Kontrolér zahŕňa aj metódy pre schválenie alebo neschválenie nového užívateľa. Pomocou ďalších metód sa vďaka kontroléru upravujú záznamy o užívateľoch alebo sa odstraňuje užívateľ.

- **ProjectController**

ProjectController sa sústreďuje na správu projektov. Obsahuje metódy na vytvorenie *store()* a vymazanie *delete()* projektu. Metóda *index()* má rovnakú funkcionálnosť ako rovnomená metóda v *UserController*i, akurát vracia zoznam projektov. Ďalšími metódami sú metódy, ktoré slúžia na úpravu záznamu o projekte ako *update()*, *changeState()*, *saveCover()*. Kontrolér obsahuje aj metódy na schválenie alebo neschválenie projektu.

- **PositionController**

Tento kontrolér zabezpečuje pomocou svojich metód vytvorenie, úpravu a vymazanie pozície. Okrem toho metódy *chooseUser()* a *removeChooseUser()* zaisťujú výber užívateľa na pozíciu a odstránenie užívateľa z pozície.

- **PositionCategoryController**

Ak potrebujem zoznam všetkých možných kategórií pozície, tak volám metódu *index()* v kontroléri *PositionCategoryController*. Pomocou metódy *store()* zase vytváram novú kategóriu. Kontrolér obsahuje aj metódy pre úpravu a mazanie kategórií.

- **NotificationController**

Kontrolér *NotificationController* sa stará o správu notifikácií. Pomocou metódy *getNotifications* získavam všetky notifikácie užívateľa. Ďalšia metóda *setNotificationsRead()* nastavuje notifikácie ako prečítané. Poslednou metódou je *sendRecommend()*, ktorou vytvorím odporúčenie.

- **MessageController**

Tento kontrolér pracuje s databázovou tabuľkou *Message* a má 3 metódy. Prvou metódou je *store()*, ktorej úlohou je vytvoriť správu. Ďalšími metódami sú *index()* a *show()*. Prvá menovaná získava zoznam správ užívateľa a druhá vracia dáta o konkrétnej správe definovanej na základe id.

- **InterestController**

InterestController zabezpečuje komunikáciu pre riadenie záujmov na projekte. Pomocou metód *store()* a *delete()* sa vytvára alebo odstraňuje záujem. Metóda *index()* zase vracia všetky záujmy určitého užívateľa.

- **BookmarkController**

Kontrolér *BookmarkController* pomocou svojich metód *store()* a *delete()* zaisťuje tvorbu a mazanie užívateľských záložiek na vybraný projekt. Obsahuje tiež metódu *show()*, ktorá vracia záložky užívateľa definovaného pomocou id.

5.6 Pohľad

Pohľad tvorí grafické rozhranie, ktoré bolo vytvorené na základe návrhu z časti 4.6. Základom je súbor *resources/view/index.blade.php*. Tento súbor má kostru HTML a obsahuje koreňový uzol, ktorého obsah je vykreslovaný *Reactom*. Jednotlivé komponenty sú uložené v zložke *resources/js/components/*. Hlavným koreňovým komponentom je *Index.js*. Súčasťou tohto komponentu je smerovač, ktorý na základe adresy vyberie podkomponent ktorý sa vykreslí.

Jednotlivé adresy vykresľujú nasledujúce stránky:

- / – zobrazí úvodnú stránku s prihlasovaním
- /**signup** – zobrazí stránku s registračným formulárom
- /**projects** – ukazuje stránku na ktorej sú zoznamy projektov
- /**users** – aplikácia zobrazí zoznam užívateľov
- /**create** – formulár pre vytvorenie projektu
- /**project/{id}** – profil projektu s daným id
- /**user/{id}** – profil užívateľa s daným id
- /**messages** – zoznam správ prihláseného užívateľa

- `/message/{id}` – zobrazí správu s daným id
- `/about` – stránka na úpravu vlastného profilu
- `/account` – zobrazí stránku na možnú úpravu vlastností účtu
- `/admin/dashboard` – hlavná stránka administrátora
- `/admin/users` – administrátorská stránka na úpravu užívateľov
- `/admin/positions` – administrátorská stránka pozícií
- `/admin/projects` – administrátorská stránka projektov
- `/admin/positioncategories` – administrátorská stránka na úpravu kategórií pozície

Kapitola 6

Užívateľské testovanie a vyhodnotenie testovania

„*Ak chcete skvelú stránku, musíte ju testovať*“, toto hovorí vo svojej knihe Steve Krug [7]. Testovanie má vo vývoji webových aplikácií veľký význam. Vďaka testovaniu výsledný produkt zistíme kvalitu projektu z rôznych pohľadov, ako funkčnosť, spoľahlivosť či použiteľnosť.

Moje testovanie bolo vykonávané v niekoľkých fázach. Cieľom testovania bolo overiť návrh a zistiť užívateľskú spokojnosť so systémom. Testovanie bolo vykonávané formou užívateľského testovania. Testovacími subjektami každej fázy boli 3 osoby, ktoré vychádzali z cieľovej skupiny užívateľov. Traja užívatelia by podľa Steva Kruga mal byť dostatočný počet pre jednu fázu testovania [7]. Testovanie prebiehalo na prehliadačoch Safari, Chrome a Firefox.

V tejto kapitole sú popísané jednotlivé fázy testovania, ktoré som vykonával počas vývoja webového portálu. V tretej časti tejto kapitoly sú popísané výsledky testovania a na záver kapitoly priblížim možné vylepšenia.

6.1 Prvá fáza

Prvú fázu testovania aplikácie som sa rozhodol vykonávať počas doladovania implementácie, aby mi ukázala nedostatky, ktorými ešte systém disponuje. Tieto testy sa zameriavali na jednotlivé hlavné funkcie systému. Testovacím subjektom bolo v krátkosti vysvetlené, čomu sa aplikácia venuje. Následne boli posadení, pred otvorený webový portál. Postupne im boli zadávané úlohy, ktoré majú vykonať. Úlohy boli zamerané na hlavné činnosti systému. Testovací užívatelia boli nabádaní k rozmyšľaniu nahlas a komentovaniu každého kroku. Sledované bolo či užívateľ vedel ako má úlohu vyriešiť a nedostatky funkčnosti systému. Celý priebeh vykonávania testov som pozoroval. Zaznamenával som odhalené nedostatky systému či myšlienky užívateľa. V prípade potreby im bola poskytnutá pomoc a navedenie na riešenie úlohy.

Vykonávané úlohy

- Zaregistrujte sa do aplikácie UNI.TEAM.
- Vyplňte si údaje vo vašom profile.

- Vytvorte projekt s názvom Vývoj aplikácie pre podporu športovcov. Projekt bude obsahovať 2 pozície (Programátor, Grafický dizajnér) a hľadanie spolupracovníkov bude trvať do 14.apríla.
- Nájdite projekt Webový portál pre študentov a otvorte jeho profil.
- V projekte Web KAM vyjadrite záujem o voľnú pozíciu Programátora.
- Odpovedajte na správu, ktorá vám prišla.
- Otvorte profil projektu, ktorý ste vytvorili a vyberte jedného zo záujemcov na pozíciu.
- Otvorte profil projektu, upravte jeho popis a pridajte novú pozíciu Backend programátor.
- Vymažte projekt, ktorý ste vytvorili.

Po skončení vykonávania úloh som s užívateľmi viedol rozhovor. Cieľom bolo zistiť, ako sa užívateľom riešili jednotlivé úlohy a či mali s nimi nejaký problém. Zisťoval som ako sa im pracovalo s jednotlivými prvkami systému, či im vyhovuje dizajn a rozloženie stránky a čo by na ňom zmenili prípadne vylepšili.

6.2 Druhá fáza

V druhej fáze testovania som mal dostupný už plne funkčný produkt webového portálu. Cieľom bolo sledovať prácu užívateľa so systémom. Pozoroval som, či užívateľ vie nájsť všetky činnosti, ktoré chce vykonať, či nerobí zbytočné kliky a či má nejaký problém s aplikáciou.

Súčasťou fázy bolo aj testovanie mobilnej verzie aplikácie. Túto verziu testovalo rovnaký počet testovacích subjektov ako klasickú verziu.

V tejto fáze som chcel, aby testovanie prebiehalo viacej samostatne. Cieľom bolo nasiemulovať reálnu udalosť, kvôli ktorej užívateľ portál navštívi. Pre tú príležitosť som pripravil niekoľko dlhších slovných zadaní, ktoré zahrňovali úlohy v systéme. Tieto úlohy boli testovaciemu subjektu predložené bez ďalšieho komentáru. Testovací užívatelia mali tieto úlohy vyriešiť. Každá úloha sa skladala z dvoch častí a medzi riešeniami prvej a druhej časti som v systéme nasimuloval rôzne situácie. Zadania sú dostupné v prílohe C.

Užívatelia boli pri vypracovávaní zadania sledovaní a bol stopovaný čas práce v systéme. Pri pozorovaní boli zapisované poznámky o riešení úloh, orientácií v systéme a iné. Na záver testovania som sa užívateľov pýtal na ich názor na aplikáciu. Otázky boli napríklad: „Ako sa vám pracovalo so systémom? Zmenili by ste niečo v systéme?“

6.3 Výsledky testovania

V tejto podkapitole prejdeme výsledkami testovania jednotlivých častí systému v oboch fázach testovania.

Navigačné menu

Pri testovaní bol spozorovaný jeden problém, na základe ktorého som následne upravil navigačné menu. Problémom bolo, že testovací užívatelia automaticky hľadali vytvorenie nového

projektu v sekcii „Projekty“. Nespozorovali, že vytvorenie nového projektu je priamo súčasťou menu. Výsledkom bolo teda presunutie odkazu na vytvorenie projektu do sekcie „Projekty“ a upravil som navigačné menu.

Registrácia a prihlásenie

Užívatelia bez problémov našli spôsob ako sa registrovať do systému. Proces registrácie im tiež nerobil žiadne starosti.

Vytvorenie projektu

Pri vytváraní projektu som spozoroval drobné problémy. Pri vyplňovaní častí supervízora a dátumu ukončenia vyhľadávania užívatelia trávili väčšie množstvo času, než pri ostatných položkách. Zistil som, že užívatelia dobre nevedeli k čomu slúži supervízor a dátum ukončenia vyhľadávania. Tento problém som vyriešil nápodovou pri týchto položkách, ktorá im vysvetlí k čomu slúžia. Dátum ukončenia vyhľadávania som tiež premenoval na dátum začiatku práce na projekte.

Vyhľadávanie projektu

Všetky testovacie subjekty si vedeli otvoriť panel rozšíreného vyhľadávania, aj keď v rozhovore sa mi niektorí priznali, že išli viacmenej podľa inštinktu.

Vyhľadanie a filtrovanie projektov bolo z pohľadu užívateľov bezproblémové. Vedeli si najisto vlastné projekty aj vytvoriť a následne zobrazit záložky.

Zájum na pozíciu

Vytvorenie zájmu bolo u užívateľov automatické a nezaznamenal som žiadne problémy.

Správy

Pri správach som musel viac zvýrazniť priložený životopis, keďže ho užívatelia často prehliadali.

Notifikácie

Riešenie notifikačného panelu bolo iné než to, na ktoré sú užívatelia zvyknutý zo sociálnych sietí alebo mnohých ďalších aplikácií. Bol som preto veľmi zvedavý, ako na to zareagujú. Všetcu užívatelia si vedeli bez problémov otvoriť notifikácie a najisto nové oznámenie. Z oznámenia sa automaticky preklikli do správy či profilu. Z následných rozhovorov vychádzalo, že sa im toto riešenie celkom pozdávalo.

6.4 Zhrnutie testovania

Testovanie webového portálu prebiehalo bez problémov a nepopierateľne malo svoj prínos. Pozorovaním riešenia úloh v oboch fázach testovania som nezaznamenal žiadne väčšie problémy systému. Čas realizovania úloh bol primerane dlhý a užívatelia vedeli najisto všetko, čo potrebovali. Počas testovania sa podarilo odhaliť niekoľko menších chýb a nedostatkov aplikácie v rámci funkcionality a užívateľskej skúsenosti. Tieto nedostatky som uviedol v časti 6.3. Všetky vykonané úpravy viedli k zlepšeniu celkovej užívateľskej spokojnosti.

Z rozhovorov s testovacími užívateľmi bola zaznamenaná pozitívna spätná väzba. Užívateľom sa s aplikáciou pracovalo dobre a pochopili k čomu systém slúži. Viacero z nich sa vyjadrilo, že ak by bola aplikácia zavedená do praxe, tak by túto aplikáciu v prípade potreby určite využili.

Na základe pozorovania a rozhovoru s testovacími užívateľmi je možné vyvodiť záver, že užívatelia boli s návrhom aplikácie spokojní. Návrh teda môže poslúžiť pre implementáciu finálneho riešenia webového portálu pre spoluprácu študentov.

6.5 Možné vylepšenia

V rámci vytvoreného webového portálu sa načrtávajú rôzne vylepšenia, ktoré by bolo možné dorobiť. V tejto časti si predstavíme niektoré z nich.

Prihlasovanie cez univerzitné údaje

V prípade zavedenia systému v rámci univerzity, by sa dal obísť systém registrácie. Prihlasovanie by bolo možné prostredníctvom univerzitného prihlasovacieho mena a hesla. Registrácia by ostala len pre užívateľov mimo univerzity.

Hodnotenie študentov

Do systému by sa tiež mohol pridať určitý gamifikačný prvok a to konkrétne hodnotenie študentov. Po skončení práce na projekte by autor projektu vedel ohodnotiť užívateľov, ktorý pracovali na jednotlivých pozíciách. Užívateľ by takto zbieral hodnotenia a jeho priemerný výsledok by sa zobrazoval v profile užívateľov. Autori by si mohli takto zdieľať spokojnosť so spolupracovníkmi. Hodnotenie by následne mohlo ovplyvňovať výber z kandidátov na pozíciu.

Systém správy projektu

Výsledná aplikácia sa zameriava len na proces podnietenia spolupráce na projekte. Systém by bolo možné rozšíriť aj o proces správy pracovnej činnosti na projekte.

Kapitola 7

Záver

Cieľom tejto práce bolo navrhnúť užívateľský prijateľný systém pre študentskú spoluprácu. Vytvorený návrh bolo potrebné overiť a zistiť či by bol takýto systém prijatý cieľovými užívateľmi. Pri celom vývoji aplikácie som sa snažil myslieť na užívateľa. Aby som docielil priaznivú užívateľskú spokojnosť, tak som sa v priebehu vývoja zameral na použitie rôznych UX metód.

Na začiatku práce bolo potrebné naštudovať moderné webové technológie a vybrať si tie, s ktorými budem pracovať. Ďalej bolo potrebné si spraviť prehľad v oblasti UX a zoznámiť sa s metódami UX, ktoré sa používajú pre vývoj webových aplikácií. Po získaní teoretických znalostí som začal vyvíjať aplikáciu. Ako prvé pri vývoji som si vytýčil ciele a spravil prehľad existujúcich portálov. Následne som si definoval cieľových užívateľov. Zo získaných dát som si stanovil požiadavky, ktoré musel systém spĺňať a vytvoril graf prípadov použitia. Po získaní potrebných dát som mohol pristúpiť k návrhu aplikácie. Návrh spočíval v dátovom modelovaní a vytvorení návrhu grafického užívateľského rozhrania. Aby som mohol návrh otestovať, bolo potrebné implementovať testovaciu verziu aplikácie. Záverečnú časť práce tvorilo testovanie vytvoreného prototypu webového portálu. Na záver som sa zamyslel nad možnými vylepšeniami aplikácie.

Výsledkom mojej práce je návrh a prototyp webového portálu, na ktorom som overil vytvorený návrh. Výsledky užívateľských testov mi odhalili drobné nedostatky, ktoré som opravil. Pri testovaní sa však nevyskytli žiadne väčšie problémy s používaním systému, z čoho usudzujem, že systém bol navrhnutý správne a návrh sa môže použiť pre finálnu implementáciu. Z rozhovorov vedených po testovaní som zistil, že vytvorená aplikácia mala u užívateľov priaznivé reakcie a v prípade nasadenia by ju určite používali.

Literatúra

- [1] Anonym: *Designové metody*. [Online; navštíveno 4.04.2019].
URL <http://libdesign.kisk.cz/designove-metody>
- [2] Anonym: *Javascript, jeho frameworky a ako sa v nich nestratit*. [Online; navštíveno 4.04.2019].
URL <https://www.kremsa.sk/javascript-jeho-frameworky-a-ako-sa-v-nich-nestratit/>
- [3] Anonym: *Usage of server-side programming languages for websites*. [Online; navštíveno 4.04.2019].
URL https://w3techs.com/technologies/overview/programming_language/all
- [4] Banks, A.; Porcello, E.: *Learning React: Functional Web Development with React and Redux*. O'Reilly Media, 2017, ISBN 9781491954591.
- [5] Buley, L.: *The user experience team of one : a research and design survival guide*. Rosenfeld, 2013, ISBN 978-1-933820-18-7.
- [6] Klímová, N.: *Komunikační platforma pro iniciaci studentskej spolupráce*. Vysoké učení technické v Brně. Fakulta výtvarných umění, 2016.
- [7] Krug, S.: *Don't make me think, Revisited*. New Riders, 2014, ISBN 978-0-321-96551-6.
- [8] Merholz, P.: *Peter in Conversation with Don Norman About UX Innovation*. [Online; navštíveno 27.03.2019].
URL <https://adaptivepath.org/ideas/e000862/>
- [9] Monus, A.: *10 PHP Frameworks For Developers – Best of*. [Online; navštíveno 6.04.2019].
URL <https://www.hongkiat.com/blog/best-php-frameworks/>
- [10] Morville, P.: *User Experience Design*. [Online; navštíveno 03.04.2019].
URL http://semanticstudios.com/user_experience_design/
- [11] Nielsen, J.: *How Users Read on the Web*. [Online; navštíveno 3.04.2019].
URL <https://www.nngroup.com/articles/how-users-read-on-the-web/>
- [12] Norman, D.: *The Design of Everyday Things*. Basic Books, 1988, ISBN 978-0-465-06710-7.
- [13] Norman, D.; Nielsen, J.: *Definition of User Experience (UX)*. [Online; navštíveno 27.03.2019].
URL <https://www.nngroup.com/articles/definition-user-experience/>

- [14] Pureval, S.: *Learning Web App Development*. O'Reilly Media, 2014, ISBN 978-1449370190.
- [15] Suchá, L. Z.: *100 metod*. [Online; navštíveno 4.04.2019].
URL <https://100metod.cz/>

Príloha A

Obsah priloženého pamäťového média

- **uniteam/** – Adresár so zdrojovými kódmi aplikácie
- **srcTZ/** – Adresár so zdrojovými kódmi technickej správy
- **xsvetl05_tz.pdf** – Technická správa v PDF
- **manual.txt** – Návod k inštalácií
- **data/** – Adresár obsahujúci video, plagát a mockup

Príloha B

Persony

Ján Žilinka



Vek: 22
Pohlavie: muž
Brigáda/Práca: -
Stav: zadáný
Bydlisko: Martin , Slovensko
Fakulta: FEKT
Ubytovanie: koleje

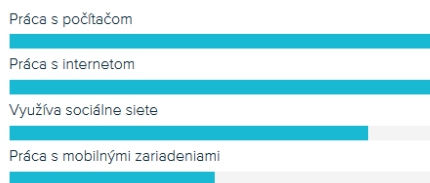
Charakter

spofahlivý, praktický, bez problémov sa učí novým veciam, sebavedomý

Informácie

Ján je študentom fakulty FEKT odbor Informačná bezpečnosť. Zaujíma sa o IT sféru hlavne z pohľadu bezpečnosti. Rád sleduje nové trendy a výskumy v oblasti IT bezpečnosti. Nadviazovanie kontaktu je pre neho problém. Najčastejšie komunikuje cez sociálne siete. Jeho záľubami je hlavne pozeranie filmov a seriálov. Veľmi rád tiež trávi čas v prírode. Vlastní smartphone, ktorý má vždy po ruke. Používa ho na surfovanie po internete, sociálne siete a hranie hier.

Schopnosti



Používanie portálu

Momentálne nemá žiadny nápad na projekt. Má ale dostatok času, a preto by sa rád pripojil k nejakému projektu, na ktorom by mohol pracovať. Mal by záujem o pozíciu v oblasti IT bezpečnosti alebo sietí. Pracovať chce popri škole a má záujem iba o pozície, ktoré mu prinesú aspoň malý zárobok.

Obr. B.1: Persona Ján Žilinský

Klára Konečná



Vek: 19
Pohlavie: žena
Brigáda/Práca: -
Stav: slobodná
Bydlisko: Bratislava, Slovensko
Fakulta: FSI
Ubytovanie: koleje

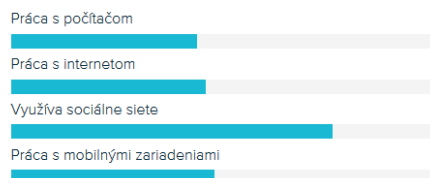
Charakter

spofahlivá, sebavedomá, dochvilná, spoločenská, usilovná

Informácie

Klára je študentkou 1. ročníka fakulty FSI odbor materiálové inžinierstvo. Často chodí von s priateľmi a rada komunikuje cez sociálne siete. Pomoc druhým jej nerobí žiadny problém. Vždy sa snaží zo seba vydať to najlepšie. Jej záľubami sú hlavne prechádzky so psom a počúvanie hudby. Veľmi rád tiež trávi čas v prírode a rada športuje. Vlastní smartphome, ktorý používa hlavne na sociálne siete.

Schopnosti



Používanie portálu

Klára má nápad na vytvorenie mobilnej aplikácie. Nepozná ale študentov, ktorý by jej vedeli pomôcť s realizáciou a preto využije portál UNI.TEAM. Snaží sa nájsť spolupracovníkov na programovanie frontendovej a backendovej časti ale aj schopného dizajnéra, ktorý by dal aplikácií priateľný vzhľad. Realizácia projektu by mala prebiehať popri škole.

Obr. B.2: Persona Klára Konečná

Príloha C

Zadania úloh 2.fázy testovania

1.úloha

- **1.časť**

Predstavte si, že máte nápad na mobilnú aplikáciu pre výučbu anglického jazyka hravou formou. Nanešťastie ale nepoznáte ľudí, ktorý by vám pomohli tento nápad zrealizovať. Potrebujete nájsť jedného frontend programátora a jedného grafického dizajnéra. Pracovať na projekte by ste chceli začať do mesiaca. Kamarát vám poradil aby ste vyskúšali UNI.TEAM, čo je portál na hľadanie spolupracovníkov. Link je <http://www.stud.fit.vutbr.cz/~xsvet105/>.

- **2.časť**

Po nejakom čase sa idete pozrieť či sa vám niekto neozval. V prípade ak áno pozriete si či už pracoval na nejakých projektoch a stiahnete si jeho životopis, ktorý vám poslal. Záujemcovi odpíšete na správu, že ste sa rozhodli ho vybrať na pozíciu. Následne ho prijmete na danú pozíciu. Ak sa vám na niektorú z pozícií nikto neprihlásil, tak si vyhľadajte v zozname užívateľov takých, ktorý by mohli mať záujem o vašu pozíciu. Záujem by mohli mať tí, ktorí majú uvedenú pozíciu v profile rovnakú ako je vaša voľná pozícia.

2.úloha

- **1.časť**

Predstavte si, že by ste radi spolupracovali na tvorbe nejakého projektu. Chceli by ste sa venovať programovaniu v Jave, s ktorou už máte skúsenosti. Od kamaráta ste počuli o webe UNI.TEAM, kde sa dá nájsť spolupráca. Idete teda na túto stránku a prihlásite sa tam. Po úspešnom prihlásení si upravíte profil, aby bolo uvedených o vás viac informácií a vyhľadáte si nejakú zaujímavú pozíciu pre vás. Na nájdenu pozíciu sa rozhodnete vyjadriť záujem. Link je <http://www.stud.fit.vutbr.cz/~xsvet105/>.

- **2.časť**

Po určitom čase sa idete pozrieť, či sa vám niekto neozval s informáciou o pozícií. V prípade, že áno odpíšete mu na správu a dohodnete si podrobnosti.