

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

SOFTVÉROVÁ INFRAŠTRUKTÚRA PRE FABLAB.SK

Bakalárska práca

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

SOFTVÉROVÁ INFRAŠTRUKTÚRA PRE FABLAB.SK

Bakalárska práca

Študijný program: Aplikovaná informatika

Študijný odbor: 2511 Aplikovaná informatika

Školiace pracovisko: Katedra aplikovanej informatiky

Školiteľ: Mgr. Pavel Petrovič, PhD.



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Ladislav Bilisics
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: aplikovaná informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Softvérová infraštruktúra pre FabLab.SK
Software Infrastructure for FabLab.SK

Cieľ: V tvorivej dielni FabLab.SK je pre verejnosť dostupných viacero zariadení pre prototypovanie digitálnou fabrikáciou, organizujú sa tam workshopy, semináre a iné podujatia, návštevníci sa organizujú do záujmových skupín. Zariadenia môžu obsluhovať iba zaškolení pracovníci alebo hostia, je potrebné si ich vopred zarezervovať. Celá prevádzka dielne je v súčasnosti organizovaná čiastkovými informačnými systémami, prostredníctvom cloudových služieb a chýba napríklad základná evidencia ľudí. Cieľom práce je v spolupráci s vedením FabLabu analyzovať potreby dielne, vyšpecifikovať webovú aplikáciu, ktorá bude centralizovať všetku evidenciu a exportovať výstupy do ostatných moderných aplikácií a sociálnych sietí s ktorými bude integrovaná.

Literatúra: Lightweight Directory Access Protocol (LDAP): The Protocol, RFC 4511, dostupné online, URL: tools.ietf.org/html/rfc4511
Google Calendar API, dostupné online: developers.google.com/google-apps/calendar/
Google Drive API, dostupné online: developers.google.com/drive/

Vedúci: Mgr. Pavel Petrovič, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.
Dátum zadania: 03.10.2016

Dátum schválenia: 17.10.2016

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Pod'akovanie

Chcel by som sa pod'akovať môjmu školiteľovi, Mgr. Pavlovi Petrovičovi, PhD. za jeho pomoc a čas venovaný vylepšovaniu mojej práce, návrhov, zmien a zhodnotenia výsledkov pri tvorbe tejto bakalárskej práce.

Taktiež by som sa chcel pod'akovať pánovi Ing. Jozefovi Vaškovi, z vedenia FabLabu, ktorý mi venoval množstvo svojho času a energie, aby sme vytvorili návrh podľa jeho predstáv.

Čestné vyhlásenie

Čestne prehlasujem, že som na mojej bakalárskej práci pracoval sám pod dozorom môjho školiteľa, Mgr. Pavla Petroviča, PhD.

Abstrakt

Táto práca je zameraná na vytvorenie informačného systému pre FabLab Bratislava. FabLab Bratislava je detašovaným oddelením Centra vedecko-technických informácií SR. Ponúka moderné služby ako napríklad používanie prístrojov, ktoré širokej verejnosti nie sú dostupné. Príkladom takýchto prístrojov sú 3D tlačiarne či CNC lasery. Taktiež organizuje mnohé udalosti, súťaže a exkurzie, aby obohatila a zaujala v najnovších technológiách verejnosť. V priebehu času a rozvoja vedenie FabLab Bratislavy sa rozhodli, že potrebujú nové funkcionality, ktoré ich aktuálny systém nedokáže podporovať.

Preto cieľom tejto bakalárskej práce bolo vytvoriť rozšírenia pre momentálny informačný systém stránok FabLab.SK, ktoré boli prevedené do systému WordPress. Tento systém bol považovaný za najlepšiu možnosť, vďaka jeho modularite a umožnil nám implementovať riešenia vo forme WordPressových rozšírení. Tieto rozšírenia implementujú automatickú prácu s Google kalendármi, ktoré tvoria prehľad udalostí a rezervácií na prístroje a udalosti, taktiež umožňujú evidenciu práce zamestnancov a automatické generovanie evidencie vo forme Excel súborov. Ďalej bol FabLab rozšírený o samotnú možnosť registrácie, prihlásenia na newsletter, určenia rolí či certifikátov, rezervácie a platenia prístrojov, projektov, logov o dianí v systéme a prípravenie systému na používanie výhradne kartami používateľov. Tieto rozšírenia po nainštalovaní a správnom nastavení poskytnú administrátorom stránok lepšiu, ľahšiu a rýchlejšiu kontrolu nad obsahom a dianiami v rámci FabLabu.

Moja práca sa skladá z najprv návrhu funkcionalít a následnej implementácie vo forme separátnych rozšírení.

Kľúčové slová: WordPress, Rozšírenie, PHP, JavaScript, HTML

Abstract

This thesis is focused on creating information system for FabLab Bratislava. FabLab Bratislava is a detached department of Slovak centre of scientific and technical information. It offers modern services such as usage of machines that are not available to the public. Example of machines like that are 3D printers or CNC lasers. FabLab Bratislava also organizes events, competitions and excursions for general public, for educational purposes and to induce interests in the newest technologies. As FabLab Bratislava progressed, its supervisors decided, that it desperately needs new functionalities, which current system can't support.

That's why the objective of this bachelor thesis is to create plugins for the current information system of webpages FabLab.SK, which were converted into WordPress system. This system is considered as the best option, thanks to its modularity as it enabled us to implement solutions in form of WordPress plugins. These plugins implement automatic jobs with Google calendars, which creates overview of events and event reservations. Also they enable evidence of employees work and automatic generation of this evidence in a form of Excel files. Another plugin extends FabLab's with abilities of registration, subscription to newsletter, selection of roles or certificates, reservations and paying for usage of machines, projects, logs of all actions in the system and preparation for usage of FabLab solely through member cards. After correct installation and setup of plugins, they provide administrators of webpages with better, easier and faster control over content and events happening within FabLab.

My thesis consists of draft of all wanted functionalities followed up by their implementation in form of separate plugins.

Keywords: WordPress, Plugin, PHP, JavaScript, HTML

Obsah

1. Úvod.....	1
2. Analýza súčasnej prevádzky a potrieb FabLabu.....	2
2.1. Problémy aktuálneho informačného systému	3
3. Prehľad existujúcich a súvisiacich riešení	4
3.1. Vývoj a implementácia informačného systému v konkrétnej organizácii	4
3.2. Aktualizácia webu katedry	4
3.3. Podobná služba pre fablaby – fabman.io	5
4. Analýza technológií	8
4.1. MediaWiki.....	8
4.2. Odoó	8
4.3. WordPress	9
4.4. Google API.....	9
4.5. Ajax	10
4.6. JSON	11
4.7. JQuery	11
4.8. WPDB	11
4.9. MySQL.....	12
5. Návrh	13
5.1. Analýza FabLab používateľov/ use-case	13
5.1.1. Princíp use-case špecifikácie systému	13
5.1.2. Naš systém	13
5.1.3. Nezaregistrovaný používateľ	13
5.1.4. Prihlásený používateľ	14
Pracovník	16
5.1.5. Administrátor	17
5.1.6. Pokladník	18
5.2. Štandardy NF databázy a jej model	19
5.2.1. Čo je štandard NF?	19
5.2.2. Model databázy	20
5.3. Sekvenčný diagram rezervácie zariadenia	22
5.3.1. Čo je a na čo slúži?	22
5.3.2. Sekvenčný rezervácie prístroja s popisom.....	22
5.4. Podrobná špecifikácia cieľného informačného systému.....	24
5.4.1. Registrácia	24

5.4.2.	Obmedzenie článkov.....	24
5.4.3.	Udalosti.....	24
5.4.4.	Newsletter.....	25
5.4.5.	Logovanie.....	26
5.4.6.	Rezervačné formuláre prístrojov.....	26
5.4.7.	Projekty.....	26
5.4.8.	Priradenie certifikátov.....	26
5.4.9.	Rezervácia prístrojov.....	27
5.4.10.	Spracovanie rezervácií.....	28
5.4.11.	Použitie prístrojov a karty používateľov FabLabu.....	28
5.4.12.	Platenie rezervácií.....	28
5.4.13.	Evidencia práce zamestnancov.....	29
5.5.	Implementácia.....	30
5.5.1.	Filtrovanie.....	30
5.5.2.	Admin a user menu.....	30
5.5.3.	Registračný formulár.....	31
5.5.4.	Formátovanie dátumov.....	31
5.5.5.	Sanitácia.....	32
5.5.6.	Shortcode.....	32
5.5.7.	publish_post.....	32
6.	Inštalačná príručka.....	34
6.1.	Požiadavky:.....	34
6.2.	Postup.....	34
7.	Záver.....	37
8.	Literatúra.....	39
9.	Prílohy.....	41

1. Úvod

Sieť FabLabov vznikla v Bostone pre dizajnérov, umelcov, vývojárov, študentov a neskôr aj pre širokú verejnosť. Čoskoro sa rozšírila do celého sveta, aby mohla poskytnúť svoje služby mnohým ďalším krajinám. Ponúka im prístup k zariadeniam ako napríklad 3D tlačiarne rôznych typov, CNC či laserom. Taktiež ponúka kurzy či už všeobecnejšieho vzdelania alebo odborné ohľadom manipulácie zariadení, aby osoby ktoré ich absolvujú boli schopné samostatne obsluhovať zariadenia[1].

Webová stránka FabLabu bola pôvodne dostatočná, keď FabLab.SK potreboval hlavne prezentačnú vrstvu. Časom však vyšli na povrch nové požiadavky, ktoré pôvodný systém nemohol podporovať a bol potrebný nový systém. Tieto okolnosti viedli ku vzniku mojej bakalárskej práce, ktorú som si vybral aby som získal skúsenosti a prehľad v tvorení informačných systémov, prípadne sa naučil niečo nové ako napríklad používať MediaWiki či WordPress.

Momentálne FabLab ktorý sídli v Bratislave využíva stránku na službe Facebook a stránku fablab.sk ako svoju prezentačnú vrstvu, kde oznamujú všetky svoje udalosti a ponúkajú spoločnostiam ale aj širokej verejnosti mnohé svoje služby. Majú zamestnancov, ktorí momentálne manuálne spracúvajú e-mailové žiadosti, pridávajú ich do kalendára a takýmto spôsobom zákazníkom priradzujú termíny na dané prístroje.

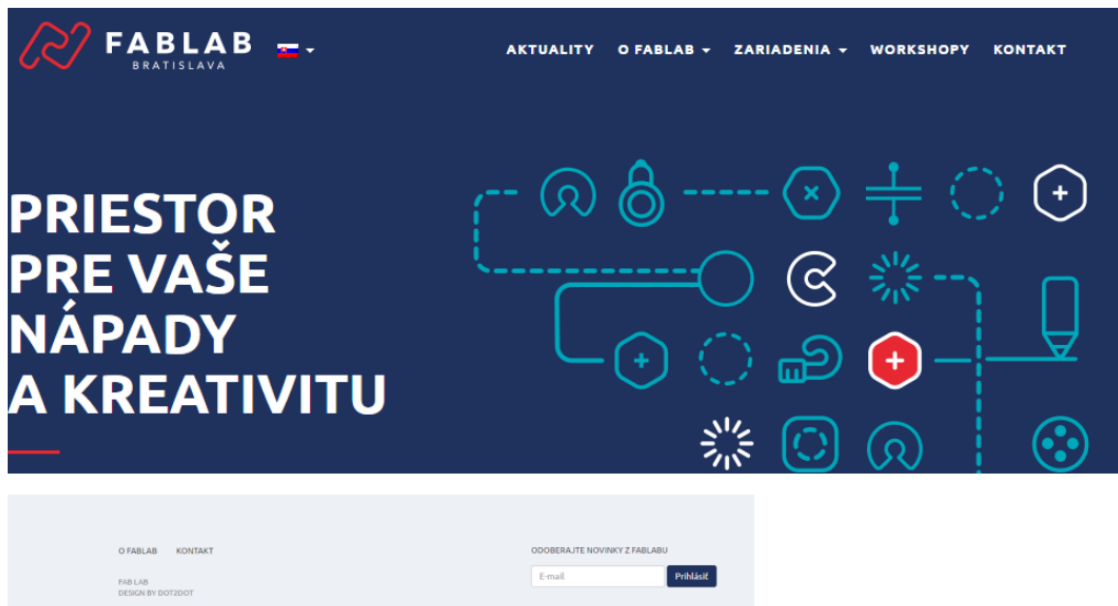
Samotná webová stránka je naprogramovaná hlavne staticky, čo sťažuje pre administrátorov FabLabu zmenu dát, ktoré sa na stránke zobrazujú. Taktiež samotný aktuálny jazyk v ktorom je webová stránka vytvorená nie je v jazyku, ktorý sú schopní samostatne spravovať zamestnanci CVTI SK/FabLab. Ďalší problém momentálneho systému je, že vyžaduje zbytočne zložitú a časovo náročnú administráciu pridávania obsahu, ako sú napríklad prednášky a semináre. Taktiež prihlasovanie používateľov na stroje a zložitejšie nemoderné používanie stránky je nepraktické pre zákazníkov a často môže viesť k tomu, že používateľ stratí záujem o služby. Martin Krupa je expertom v oblasti UI a práve preto som si zobral príkladom, keď povedal počas prednášky Webové aplikácie v praxi: „Ak sa návštevník neorientuje v priebehu 3-5 sekúnd, odíde z webu.“

Podrobností problémov sú podrobne rozanalyzované v sekcii „Porovnanie systémov“.

2. Analýza súčasnej prevádzky a potrieb FabLabu

Na presné špecifikovanie požiadaviek som najprv analyzoval aktuálny stav stránky FabLabu. Následne som zhrnul všetky problémy s týmto informačným systémom a v článku „Podrobná špecifikácia cieľného informačného systému“ v sekcii „Návrh“ navrhol všetky zmeny, ktoré je potrebné implementovať pre plynulejšie, prehľadnejšie a príjemnejšie používanie stránok FabLab.

Odkaz na aktuálny web: <https://www.fablab.sk/>



Obrázok 1 – Aktuálny web FabLabu, zdroj: www.fablab.sk

Zistenie času tlače

V softvéri MakerBot Desktop kliknete v hornom menu na „Devices“ → „Select Type of Device“ (Replicator 2X). Po kliknutí na „PREVIEW“ sa Vám zobrazí odhadovaný čas tlače (vpravo hore). Takisto tu uvidíte ako bude vyzerať samotná tlač modelu a zbadáte tam aj prípadné chyby modelu.

V prípade záujmu pracovať na 3D tlačiarňi prosíme o mailovú rezerváciu na 3dtlac@fablab.sk.

Softvér k tlačiarni

3D MakerBot 2x

Dnes 2. – 8. jan 2017

Týždeň Mesiac Agenda

	po 2.1.	ut 3.1.	st 4.1.	št 5.1.	pi 6.1.	so 7.1.	ne 8.1.
10:00							
11:00							
12:00							
13:00							
14:00							

Obrázok 2 – Aktuálna registrácia na prístroje FabLabu, zdroj:

<https://www.fablab.sk/sk/zariadenie/3d-tlaciaren-makerbot-replicator-2x/>

Aktuálny systém je webová stránka, ktorá obsahuje informácie pre návštevníkov, ako napríklad udalosti, prihlásenie na newsletter pomocou mailchimp služby, kontakt a informácie o prístrojoch. Ak sa navštevujúci používatelia chcú prihlásiť na prístroj, musia napísať správu na e-mail pod daným prístrojom, ktorý následne zamestnanci manuálne spracujú a pridajú po odsúhlasení do Google kalendára. V prípade, že už je čas zarezervovaný musia viesť konverzáciu s používateľom pomocou e-mailu. Moderátori pridávajú články taktiež manuálne zvlášť na webovú stránku ktorá je statická. Zamestnanci si zapisujú pracovné časy do Excel tabuľky, ktorú potom manuálne musí niekto na konci mesiaca spracovať do formulára evidencie práce.

2.1. Problémy aktuálneho informačného systému

Momentálne nie je dostupný žiadny registračný systém pre zákazníkov, kde by mohol administrátor systému sledovať, kedy aký prístroj použil alebo im pridelovať certifikáty aby nepotrebovali dozor. Taktiež obsah stránky sa nedá dynamicky jednoducho editovať, lebo je napísaný staticky (napevno), čo by dlhodobo mohol byť problém, ak by sa menili základné údaje o stránke. Tento systém bol vytvorený v jazyku Ruby, ktorý nikto vo FabLabe a CVTISR neovláda a nevedia web naďalej spravovať. Služby spolu nekomunikujú automaticky, čo pridáva zamestnancom na práci. Aktuálny web nevyžaduje od používateľov, aby boli členmi FabLabu a platili členský poplatok za služby, čo je momentálne cieľom FabLabu. Taktiež, pri používaní služby mailchimp. osobné údaje používateľov prihlásených na newsletter sú uložené na serveroch tretej strany. Newsletter nemá žiadnu možnosť vybrať používateľom len oblasť záujmov, o ktorých by chceli byť informovaný. Taktiež neumožňuje sa používateľom prihlásiť na udalosti, či už informačne pre FabLab alebo záväzne v prípade že udalosť je platená. Neexistuje v aktuálnom informačnom systéme žiadne obmedzenie na viditeľnosť článkov, čo zabraňuje administrácii FabLabu oznámiť informáciu len časti používateľov. Naviac FabLab chce evidovať všetky rezervácie ako súčasť projektov, ktoré spadajú do kategórie a má členov, pričom sa môžu navzájom do projektov pozívať.

3. Prehľad existujúcich a súvisiacich riešení

3.1. Vývoj a implementácia informačného systému v konkrétnej organizácii

Táto bakalárska práca bola vypracovaná Branislavom Helczom v roku 2010.

„Implementácia systému teda v sebe zahŕňa neustále opakujúci sa kolobeh analýzy požiadaviek na systém, návrhu, programovania, testovania, zavedenia do produkcie a pravidelnej údržby“ [2] výstižne poznamenáva na proces, ktorým si moja bakalárska práca musela prejsť. Najväčší dôraz som pri začiatku práce venoval analýze požiadaviek, aby som mal jasne stanovené začiatkové požiadavky a funkcionality systému, ktoré sa už počas tejto analýzy menili a neskôr sa zamerlal na zvyšné procesy. Dôležitosť tohto procesu podporuje citát z tejto práce „systém stráca svoj zmysel, ak nevykonáva tie činnosti, ktoré sa od neho očakávajú, bez ohľadu na to ako kvalitne bol naprogramovaný a aké technológie boli použité.“ [2]. Sekcia 5.2 je založená okolo témy use-case, v práci nazývanej často “prípady použitia”. Hneď v úvode sekcie [2] poznamenáva na fakt, že use-case je špecifický pre daného “actora” (slovenský ekvivalent účastníka) diagramu, na rozdiel od iných tém, ktoré sa väčšinou dotýkajú projektu globálne. Taktiež poukazuje v časti 5.2.3 “Riadenie úloh”, že dodržiavanie zobrazeného postupu na use-case diagrame [2] je veľmi prospešné pri tvorbe informačného systému. Spočíva vo vytvorení zoznamu úloh, na ktoré bol cieľový systém rozdelený, a zaznamenávaní ich jednotlivého priebehu a stavu.

3.2. Aktualizácia webu katedry

Táto bakalárska práca bola vypracovaná Dominikom Knechtom pre FMFI UK v roku 2015. Je vypracovaná v MediaWiki a poskytla mi pomoc nielen čo sa týka písomnej formy, ale aj prvkami implementácie mojej bakalárskej práce. V sekcii 4.5 sú útržky kódu, ktoré mi pomohli počas učenia sa základov programovania v PHP a MediaWiki, keď som ešte s PHP nevedel poriadne robiť a systém v ktorom mala byť moja práca vytvorená mala byť tiež MediaWiki [3]. Sekcia 1.1 a 1.2 venuje dôraz fakt, ako vytvárať prehľadné navigácie. [3] Taktiež navrhuje, aby pred tvorbou informačného systému boli všetky dáta dosiahnuteľné na stránke zozbierané. Potom ich môžeme postupne zoskupovať pod rôznymi navigáciami, pričom nechceme aby bolo navigácii príliš veľa, ani aby boli príliš hlboké. Príkladom na príliš veľa je ak má hlavné menu stránky cez desať navigácií. Príliš hlboké znamená, že máme veľa sekcií ktoré sú v iných sekciách a aj keby boli sekcie dosť prehľadné, používateľovi by trvalo dlho sa dostať ku podstránke, ktorú chce. Cieľom je štruktúru vytvoriť prirodzene, nech návštevník, nech už hľadá čokoľvek, sa vedel ku svojmu zdroju dostať čím jednoduchšie. S čím nesúhlasím v tejto práci je, že v sekcii 2.3.

„Používatelia“ [3], nedefinuje záujemcov o štúdium ako samostatnú skupinu používateľov, pričom ich činnosť na stránke je špecifická. Podľa môjho názoru by mala byť táto kategória v rámci infraštruktúry zvýraznená, keďže sa jedná o prvú interakciu medzi katedrou a jej potenciálnymi budúcimi študentami. Tiež pre niekoho, kto v živote na vysokej škole neštudoval, je pomerne ťažké sa zorientovať a vedieť, akými všetkými krokmi si musí prejsť. Jednoduchá navigácia ku záujemcovým prvým krokom štúdia je kľúčová. Nielen pre samotných maturantov, ale aj pre zamestnancov, ktorým sa pridáva práca vždy, keď študenti úplne nesplnia svoje povinnosti. Taktiež sa mi v informačnom systéme nepáči samotná navigácia. Študent sám hovoril, že navigácia by mala byť krátka a dobre zoradená, aby sa v nej tromi kliknutiami každý dostal ku informácii, ktorú potrebuje. Ale opak je na webe niekedy pravdou. Pokiaľ nepoužijeme search bar ale spomínanú navigáciu a predstavíme si, že sme začínajúci študent s otázkami, ktorý sa ešte na webstránke nevyzná, aby sme sa dostali k informáciám o študijnom oddelení, musíme prejsť týmito krokmi: Na hlavnej stránke je sekcia „Štúdium“, kde ale nenájdete tieto informácie. Namiesto toho musíte prejsť z hlavnej stránky odkazom do „Zóna pre študentov a zamestnancov“, ktoré načítajú novú stránku, veľmi podobnú hlavnej stránke. Tu už v sekcii „študenti a štúdium“ môžeme prejsť cez menu (ktoré každý krát načítava novú podstránku namiesto stromovej štruktúry) do podmenu, odtiaľ do sekcie študijné oddelenie čo nám otvorí ďalšie menu a základné informácie o študijnom oddelení, kde si môžete vybrať o kom chcete zo študijného oddelenia dostupné informácie. Ak by som mal túto skúsenosť zhrnúť, jednoznačne nie je podľa môjho názoru vhodné vytvoriť dve stránky s veľmi podobnými názvami a obsahmi, lebo sa medzi webmi používateľ ľahko stratí. Ak už chceme naozaj vytvoriť druhý web, mal by byť exkluzívny a neobsahovať podobný obsah ako predošlý.

3.3. Podobná služba pre fablaby – fabman.io

Fabman.io je služba, ktorá umožňuje zamestnancom pracujúcim v tvorivých dielnach udržiavať organizáciu nad prístrojmi a prihlásenými osobami. Prípadne ak poskytujú verejné služby, je tento systém taktiež použiteľný. Táto služba je najvhodnejšia pre fablaby práve svojimi funkciami, ktoré spomeniem neskôr. Služba práve začala svoju publikáciu a nie je príliš používaná, no má veľký potenciál a stále sa vyvíja. Miesta, kde napríklad takúto aplikáciu používajú je FabLab vo Viedni, kde je veľa jej funkcionalít využitých a naplno implementovaných. Túto službu si beriem ako príklad a motiváciu svojej bakalárskej práce, no samozrejme sú niektoré jej časti, ktoré mi nevyhovujú a sám

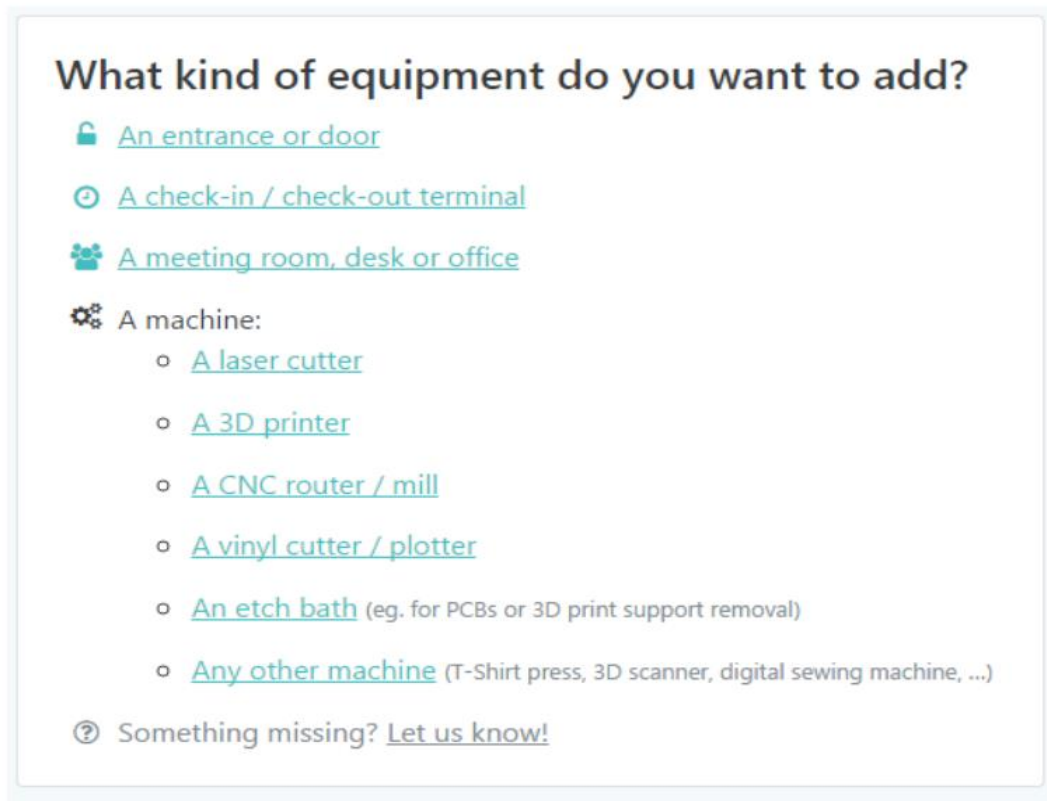
by som ich spravil pri takejto službe inak. Najprv by som poznamenal, že jediní používatelia, ktorí sú zaregistrovaní sú zamestnanci, čo je v rozpore s mojím cieľom. Mojm cieľom práce je, aby mal každý používateľ svoj vlastný účet a v rámci neho uložené svoje dáta pre zjednodušenie prihlasovania sa na prístroje a umožnenia dostávania certifikátov.

FREE	SMALL	MEDIUM	LARGE	UNLIMITED
€ 0 /month	€ 29 /month	€ 79 /month	€ 199 /month	€ 299 /month
☉ up to 5 members	☉ up to 20 members	☉ up to 70 members	☉ up to 250 members	☉ Unlimited members

Obrázok 3 – aktuálny cenník služieb fabman.io, zdroj: <https://fabman.io/#price> (7.1.2017)

Ďalej by som sa vyjadril k tomu, za čo sa pri tejto službe hlavne platí. Samotný systém je zadarmo avšak program „FREE“ (zadarmo) podporuje len päť zaregistrovaných používateľov. Ďalšie poplatky potom rastú, ak Váš systém chce používať viac ako päť zamestnancov, obrázok vyššie ich znázorňuje. Príde mi to ako nevhodný model vzhľadom k tomu, že systém nie je zaťažovaný počtom zamestnancov ale počtom návštevníkov, ktorých registráciu ešte nepodporuje. Jedinečný prvok tohto systému je, že za poplatok sa dá priamo pripojiť na prístroje a čítačku kariet. Takýmto spôsobom vie online systém priamo informovať prístroje, kto sa na aký prístroj prihlásil a k čomu má mať presne kedy a na koľko prístup. Osobám, ktoré už presiahli svoj dohodnutý čas, sa prístroj zablokuje a tak zabráňuje konfliktom, ak by boli na prístroj prihlásené v takejto situácii po sebe dve osoby. Takýto prvok každopádne stojí používateľa extra poplatok a niekedy by FabLab potreboval možnosť aby viacero používateľov zdieľalo jedno zariadenie. Tento systém automaticky udržuje štatistiky o svojich používateľoch, ich činnostiach, o využití prístrojov či rezervovaných časov. Takéto štatistiky sa dajú využiť štatistikmi napríklad na vytvorenie prehľadu ohľadom vyťaženia, ale aj popularity daného prístroja, no aj ošetriť časom chyby, ku ktorým často dochádza. Napríklad ak vieme zo štatistiky povedať, že cez 30% času keď sa niekto prihlási na 3D tlačiareň je prerušený pred dokončením svojej činnosti, znamená to asi, že by sme mali používateľov varovať o dĺžke trvania 3D tlače. Prípadne vieme aké vybavenie treba dokúpiť alebo ktoré je natoľko populárne, že by FabLab ohľadom neho zorganizoval kurzy ukončené certifikátmi. Snahou mojej bakalárskej práce bude taktiež obsiahnuť funkcionality udržiavania podobných štatistík ohľadom používateľov. Nastavenie otváracích hodín fyzickej pobočky - FabLabu sa dá nastaviť priamo v systéme, nielen pre separátne prístroje. Je zobrazené na hlavnej stránke a ľahko sa upravuje. Navyiac

sa pomocou neho ľahko oznamujú sviatky či špeciálne udalosti, ktoré by mali byť podľa administrátorovho uznania spojené s otváracími hodinami a službami FabLabu.



Obrázok 4 – zoznam ponuky vybavenia na pridanie do systému, zdroj:

<https://fabman.io/manage/14/resources/new>, požaduje registráciu a prihlásenie (7.1.2017)

System disponuje jednoduchým menu na pridávanie zariadení zo širokej variácie prístrojov, vrátane všetkých prístrojov aktuálneho FabLabu. Umožňuje nastaviť danému prístroju obmedzenie a časy kedy môže byť obsluhovaný. Taktiež sa v ňom dajú pridať miestnosti, kontrolné body či prístroje na kontrolu identity. Tieto samozrejme slúžia buď na organizovanie akcií alebo v prípade, že používateľ si zaplatil prepojenie informačného systému s kartami používateľov, na potvrdenie identity a zaznamenanie do databázy údaje o návšteve zákazníka. Napriek svojim výhradám musím uznať, že tento informačný systém je na dobrej ceste, aby nahradil jednoduché stránky s ťažkou manažérovateľnosťou elegantným systémom, ktorý zvláda základné úkony. A napriek svojim pripomienkam ohľadom registrácie každého zákazníka FabLabu, čo momentálne nie je možnosť, fabman.io už ponúkajú API a možnosť rozšírení, ktoré dovoľia používateľovi si upraviť systém podľa seba v dostatočne veľkej miere, aby bol použiteľný na rôznych systémoch. No treba myslieť na to, že tento informačný systém stále prechádza ešte vývojom a vylepšeniami.

4. Analýza technológií

Aj keď náš informačný systém používa programovací jazyk PHP a je stavaný na štandarde HTML5 s JavaScriptom, neplánujem ich príliš popisovať, vzhľadom k tomu že sa jedná o všeobecne známe a často opakované technológie. Namiesto toho som sa zamerával radšej na softvéry, ich výhody a slabiny, aby som si pri výbere základného softvéru, na ktorom bude informačný systém postavený, mal dostatočný prehľad. Taktiež som analyzoval technológie, ktoré budú pomáhať môjmu informačnému systému rozširovať jeho funkcionality.

4.1. MediaWiki

MediaWiki je softvér, ktorý pôvodne slúžil ako základ online webovej stránky Wikipédia, najväčšej online encyklopédie [4]. Momentálne je free open-source, vytvorený v programovacom jazyku PHP a je používaný na mnoho nielen podobných stránok ako Wikipédia ale aj na webových stránkach mnohých firiem, organizácií a spoločností. Na tomto softvéri funguje aj momentálne stránka Aplikovanej informatiky, fakulty matematiky, fyziky a informatiky UK. MediaWiki je známa tým, že ponúka svojim používateľom ľahké editovanie obsahu stránky a používanie mnohých rozšírení, ktoré už pre ňu existujú. Na ukladanie dát defaultne používa MySQL databázu. MediaWiki je v informačnom systéme implementovaná ako „back end“, nainštalovaný na samostatnom webe kde prebieha administrácia, registrácia a všetky novo implementované funkcie informačného systému.

Aj keď systém MediaWiki bol zaujímavý kandidát, nebol vhodný pre tento systém. nemá najlepšie vyriešenú implementáciu organizácie práv používateľov a navyše nie je dostatočne zabezpečená na ukladanie senzitívnych dát používateľov [5].

4.2. Odo

Odo je ako softvér ešte stále mladý ale nádejný systém, ktorý sa snaží vytvoriť nový pohľad na tvorenie webov a administráciu obsahu. Je štruktúrou veľmi podobný WordPressu, ibaže bol od základov skôr zameraný pre organizácie. Na prvý pohľad sa mi systém pozdával a odskúšal som si jeho základné funkcie, s ktorými som bol veľmi milo prekvapený. Systém sa pozdával aj vedeniu FabLabu, ktorý ho odporúčal ako vreľého kandidáta. Avšak po podrobnejšom používaní a snahy o vývoj rozšírenia v systéme Odo som zistil, že nemá poriadnu dokumentáciu či manuál na vývoj rozšírení a všetky už existujúce rozšírenia sú platené. Z tohto dôvodu som od tohto systému upustil, keďže

vytvorenie najmenšieho rozšírenia s jednoduchou funkcionalitou bolo zdĺhavé, ťažké a často bez chybných hlášok nefungujúce.

4.3. WordPress

WordPress je softvér, ktorý začal byť čím ďalej tým viac populárny v poslednej dobe. Najznámejšia je webová stránka WordPress.com, kde si používatelia môžu vytvoriť za pár minút blog, ktorý je založený na softvéri WordPress [6]. Aj keď pôvodne začal ako takýto blogovací systém, momentálne sa stal plnohodnotným softvérom s jednoduchým manažmentom a ponúka tisícky rozšírení a miniaplikácií. Tento softvér som nakoniec použil ako základ môjho informačného systému pre stránky www.fablab.sk. Aktuálna stránka FabLabu bola prevedená do WordPressu zamestnancami CVTI, s ktorým zamestnanci FabLabu vedia pracovať.

Tento softvér sa stal mojou voľbou nakoniec z viacerých dôvodov. Jednak má oveľa príjemnejšiu administráciu obsahu, rozsiahlejšie možnosti ako MediaWiki a je prepracovanejší systém ako Odoo, má podrobnejšiu dokumentáciu a existujú na neho mnohé rozšírenia zadarmo, ktoré uľahčili vývoj systému.

4.4. Google API

Za účelom automatického zapisovania do online Google služieb bolo pre mňa potrebné si preštudovať Google Drive API 4 a Google Calendar API 4.

Google Calendar API 4 umožňuje informačnému systému pridávať, odstraňovať a modifikovať udalosti rôznych typov v rôznych kalendároch, dokonca aj vytvoriť kalendár a uložiť si priamo jeho ID pre ďalšie editácie [7].

Použitý je na automatické zapisovanie rezervácií prístrojov a udalostí, ktoré používateľ v informačnom systéme odsúhlasí alebo pridá.

Na vytvorenie fungujúceho prepojenia medzi informačným systémom a Google kalendárom som si musel preštudovať aj Google Drive API 4 a OAuth 2.0 autentifikáciu, vzhľadom k tomu, že ho využíva Google na komunikáciu. Nakoniec sa mi po podrobnom preštudovaní dokumentácií podarilo nájsť chybu v momentálnej dokumentácii Google Calendar API 4, ktorú som opravil a chybu nahlásil. Problém bol, že OAuth 2.0 len pri prvom pripojení vygeneroval refresh token, ktorý bol uložený v jednom JSON súbore spolu s access tokenom. Keď sa pomocou refresh tokenu access token obnovil, prepísal celý JSON súbor a ten už neobsahoval refresh token.

Pre znázornenie prikladám opravený kód, ktorý slúži na obnovenie access tokenu:

```
function getClient() {
    $client = new Google_Client();
    //nastavíme klientov prístup
    // Skúšame získať tokeny zo súboru
    $credentialsPath = expandHomeDirectory(CREDENTIALS_PATH);
    if (file_exists($credentialsPath)) {
        $accessToken = json_decode(file_get_contents($credentialsPath), true);
    } else {
        //Ak uložené tokeny neexistujú, spustí sa vytvorenie prvého tokenu, potrebné vykonávať
        manuálne, nie cez systém.
    }
    //keď sme tokeny už získali, priradíme ho API klientovi a skontrolujeme, či jeho platnosť
    nevypršala
    $client->setAccessToken($accessToken);
    if ($client->isAccessTokenExpired()) {
        //Ak vypršal, uložíme si starý access token (v ktorom je aj refresh token), táto činnosť
        nebola obsiahnutá v dokumentácii Google API a získame nový access token pomocou
        refresh tokenu
        $oldAccessToken=$client->getAccessToken();
        $client->fetchAccessTokenWithRefreshToken($client->getRefreshToken());
        $accessToken=$client->getAccessToken();
        $accessToken['refresh_token']=$oldAccessToken['refresh_token'];
        //pred uložením sme novému tokenu pridali refresh token, ktorý potrebuje do budúca na
        d'alšie získania access tokenov
        file_put_contents($credentialsPath, json_encode($accessToken));
    }
    return $client;
}
//Pre plný prehľad činností referujte v prílohe súbor zdrojový kód/FabLab-  
eventy/googleCalendarFunctions.php
```

Za účelom prístupovania z informačného systému do Google API potrebujeme okrem access a refresh tokenov aj dodatočné informácie. Medzi ne patrí špecifická ID kalendára a špecifická ID udalosti. Preto v nastaveniach ukladáme informáciu ID kalendára udalostí, aby mohol používateľ určiť, kde sa budú zapisovať. Pri rezerváciách nie je potrebné každému prístroju zadávať kalendár, vzhľadom k tomu že počas vytvárania prístroja vieme získať jeho ID a zapamätať si ju.

4.5. Ajax

Asynchronous JavaScript XML, skrátene AJAX, sú žiadosti keď chceme od servera, aby spracoval nejaké informácie odoslané metódou POST a používateľovi zaslal naspäť nejakú odpoveď (response), ktorý obsahuje výsledné dáta pre používateľa. Tento typ žiadostí sme využívali počas celého vývoja, keďže je to vo WordPresse hlavný spôsob ako pracovať s dátami [8].

V kombinácii s JQuery nám umožňuje zmeniť obsah stránky dynamicky bez potreby obnovenia stránky od serveru, ktorý by mohol pri vkladaní väčšieho množstva dát (napríklad zápis plánu práce na začiatku mesiaca) spôsobiť zbytočné desiatky načítaní, zaťaženie servera a stratu času.

4.6. JSON

JSON je spôsob formátovania dát, pôvodne používaný v JavaScripte na jednoduché a čitateľné uschovanie dát objektov. Tento zápis dát je nezávislý na jazyku a stal sa rýchlo populárnym aj vo zvyšných programovacích jazykoch. [9] Dáta sú formátované dvoma spôsobmi:

- kolekcia dvojíc meno/hodnota
- zoradený zoznam dát, najčastejšie interpretovaný ako pole (array)

Príklad v práci je ukladanie Google tokenov v súbore:

```
{"access_token": "ya29.GlxUBF5KvB-N7PJcc-hpUvL7JKBZQ",  
  "token_type": "Bearer",  
  "expires_in": 3600,  
  "created": 1495658775,  
  "refresh_token": "1\u2716u-6Ve7hXJgxiPJcr"}
```

Taktiež sme využili JavaScript funkciu JSONu `JSON.stringify`, ktorá uloží dáta ako v premennej typu string a následne sa dá bez problémov parsovať späť do dát. Toto sme využili aby sme nemali príliš zahltenú databázu možnosťami účasti na udalostiach.

4.7. JQuery

Jedná sa o voľne dostupnú, open-source JavaSkriptovú knižnicu ktorá beží na strane klienta (client side). Služi na ľahšie získavanie dát z webových dokumentov, ich editáciu, nahrádzanie, mazanie, vytváranie animácií a rýchle spracovanie AJAX žiadostí.

4.8. WPDB

Jedná sa o WordPressovú vstavanú triedu, ktorá obsluhuje databázu a tabuľky v nej obsiahnuté. Pri jej použití automaticky dochádza k sanitácii textu, aby nemohol zneužiť používateľ vstupné polia pomocou útoku SQL injection [10].

Má však svoje obmedzenia, keďže napríklad funkcia DELETE, INSERT alebo UPDATE dodáva len možnosť podmienok WHERE a v prípade UPDATE funkcie nové hodnoty. Nevieme napríklad povedať WPDB aby po vykonaní INSERT nám navrátil jeho ID.

4.9. MySQL

MySQL je otvorený databázový systém, ktorú využíva WordPress na ukladanie dát a pomocou WPDB túto databázu ovláda.

Aj keď MySQL neponúka štandardný SQL príkaz FULL JOIN [11], vieme ich simulovať rôznymi spôsobmi. Toto sme využili aj pri získavaní dát, keď sme chceli vypísať všetky rezervácie na prístroje a či používateľ má certifikát na prístroj, na ktorý sa prihlásil. Čo z pohľadu systému znamenalo, že sme potrebovali získať všetky rezervácie a potom im priradiť buď to NULL ak certifikát nemajú, alebo hodnotu, ak majú. Problém som vyriešil tak, že všetky podmienky ohľadom druhej tabuľky, ktorá obsahovala informáciu o tom či tento certifikát má alebo nie, definovať nie vo WHERE časti, ale v ON časti pri LEFT JOIN príkaze.

Samotná MySQL žiadosť (pre čitateľnosť som nahradil premenné čitateľnými hodnotami):

```
'SELECT * FROM
pristroje_nepotvrdene_formulare
LEFT JOIN pristroje_certifikaty ON
pristroje_certifikaty.certificateID = pristroje_nepotvrdene_formulare.certificateID AND
pristroje_nepotvrdene_formulare.userID = pristroje_certifikaty.userid
LEFT JOIN pristroje_IDpristrojov ON
pristroje_nepotvrdene_formulare.certificateID = pristroje_IDpristrojov.id
JOIN pristroje_projekty ON pristroje_nepotvrdene_formulare.projectID = pristroje_projekty.id
JOIN pristroje_kategorieprojektov
ON pristroje_projekty.kategory = pristroje_kategorieprojektov.id
WHERE pristroje_nepotvrdene_formulare.startTime >= "VSTUP 1" AND
pristroje_nepotvrdene_formulare.startTime <= ""VSTUP 2" ';
```

5. Návrh

5.1. Analýza FabLab používateľov/ use-case

5.1.1. Princíp use-case špecifikácie systému

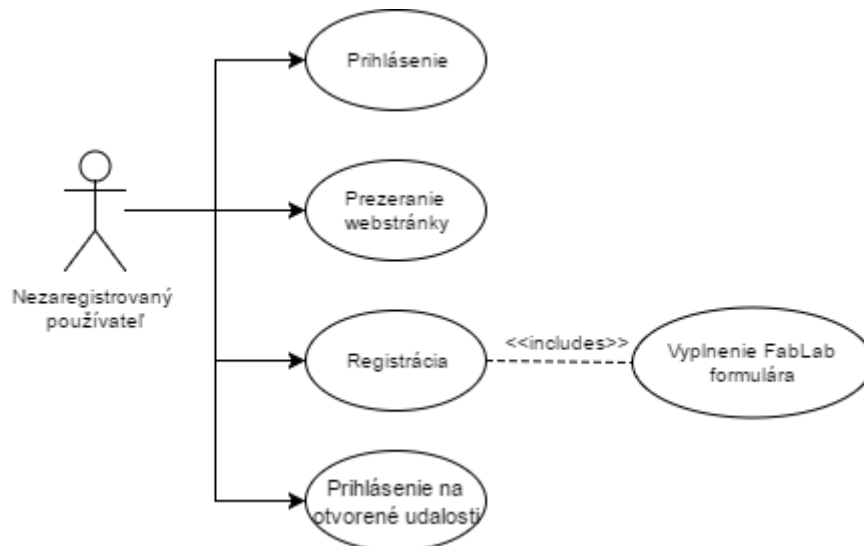
Use-case je metóda špecifikácie systému, ktorá pomáha identifikovať a zorganizovať jednotlivé požiadavky na systém. Na zachytenie týchto požiadaviek využívame priamo interakciu medzi jednotlivými typmi používateľov a plánovaným systémom, pričom znázorňujeme špecifickú akciu či požiadavku používateľa s nejakým cieľom, ktorý mu systém má pomôcť dosiahnuť [12].

5.1.2. Náš systém

V rámci nášho systému rozdeľujeme používateľov do viacerých skupín podľa ich práv a typu registrácie na stránke. Medzi faktory, podľa ktorých rozdeľujeme používateľov spadá newsletter, absolvované kurzy, členstvo vo FabLabe, samotná registrácia a práva pridelené administrátorom.

Všetky use-case diagramy boli vytvorené vo voľne dostupnej online aplikácii <https://www.draw.io/>.

5.1.3. Nezaregistrovaný používateľ

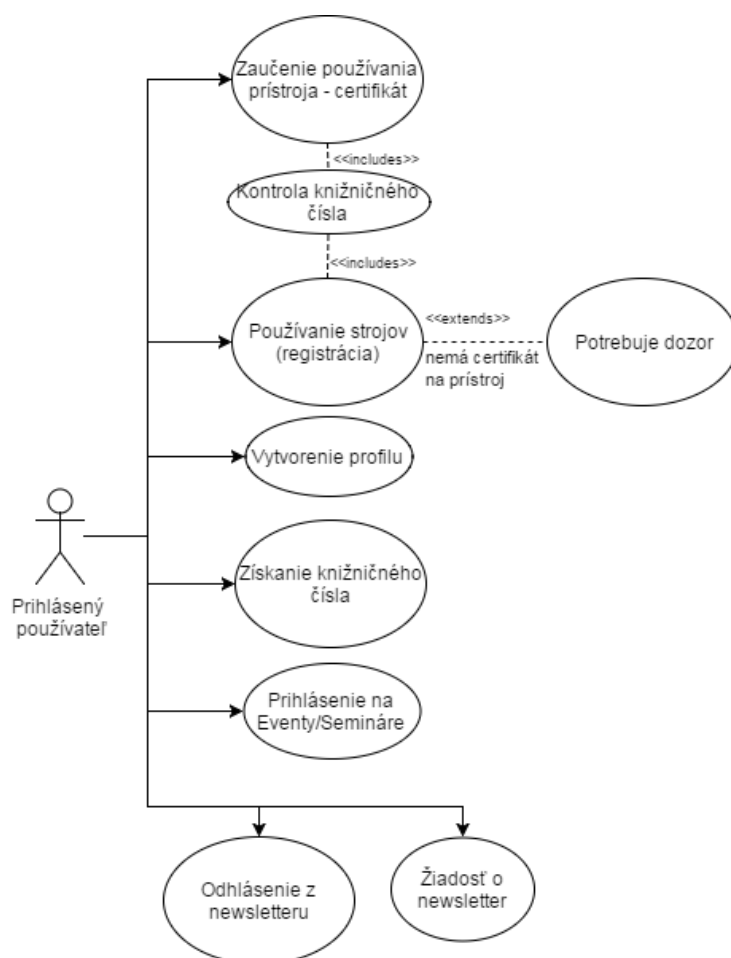


Obrázok 5 – use-case diagram nezaregistrovaného používateľa, vytvorené na stránke www.draw.io

Nezaregistrovaný používateľ má právo prezerať si prezentačnú časť stránky a časť článkov, ktoré mu *administrátor* v systéme sprístupnil. Ďalej sa smie neprihlásený používateľ prihlásiť na udalosť vytvorenú bez povinnej registrácie, pomocou svojho mena,

priezviska a e-mailovej adresy. Používateľ má právo zmeniť svoju účasť na udalosti, v prípade, ak nie je prihlásenie na udalosť *záväzná* (je platená) a používateľ nepotvrdil svoju účasť možnosťou „Áno“. Pre využitie ďalších služieb sa musí prihlásiť a v prípade že nemá účet zaregistrovať sa. Registrácia obsahuje formulár dodaný FabLabom, ktorý požaduje povinne okrem platnej e-mailovej adresy, prihlasovacieho mena a hesla aj meno, priezvisko, a súhlas so spracovaním osobných údajov. Po odoslaní registračného formulára príde používateľovi potvrdzovací e-mail, ktorý musí odsúhlasiť, aby bola registrácia úspešná.

5.1.4. Prihlásený používateľ



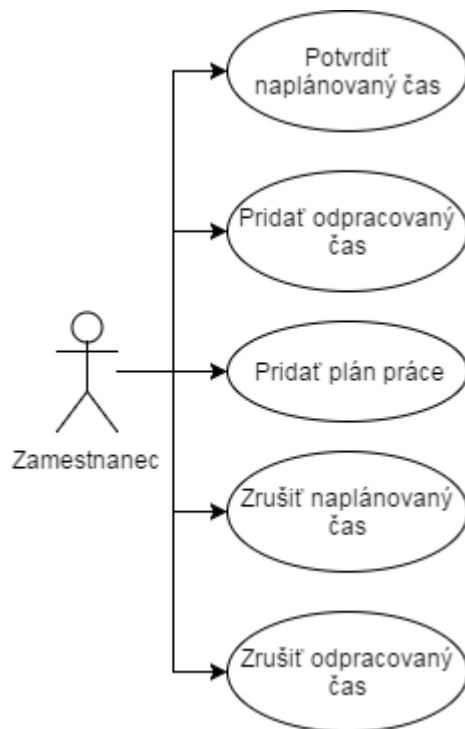
Obrázok 6 – use-case diagram prihláseného používateľ'a, vytvorené na stránke www.draw.io

Prihlásený používateľ má možnosť upravovať svoj profil, kde môže vyplniť svoju biografiu a prípadný odkaz na svoju stránku. Používateľ sa môže odhlásiť alebo prihlásiť na newsletter, pričom súhlasí so spracovaním osobných údajov a vyberá si z ponúkaných kategórií newsletteru, ktoré sú dostupné. Tiež sa môže zúčastniť udalostí (Eventov/Seminárov), ktoré sú buď prístupné verejnosti alebo aj registrovaným

používateľom. Formulár je uzamknutý pre *prihláseného používateľa* na editovanie, keďže jeho osobné údaje už máme a automaticky ich vieme doplniť. Jediné čo potrebujeme je typ účasti (Áno, Nie, Možno atď.) aby sme vedeli či sa chce zúčastniť. Tieto udalosti môžu byť buď zadarmo, alebo platené. Ak sú platené, tak prihlásenie účasti „Áno“ považujeme za záväzné. S týmto používateľ súhlasí pri odsúhlasení pravidiel prihlasovania sa na udalosti.

V prípade, že chce používateľ mať prístup k používaniu prístrojov FabLabu, musí byť členom FabLabu. Členstvo vo FabLabe si musí prísť osobne vybaviť do FabLabu. Po zaplatení poplatkov a potvrdení členstva mu *administrátor* číslo karty priradí k účtu a používateľ sa smie prihlásiť na používanie prístrojov. Na individuálne používanie prístroja potrebuje zákazník absolvovať zaučenie na prístroj a získať certifikát. Sú tri rôzne role, ktoré môže počas zaučenia dosiahnuť. *Zaučený*, keď smie sám obsluhovať prístroje, *dozor*, ak smie robiť dozor *nezaučeným osobám* a *školiteľ* v prípade, ak má dostatočné skúsenosti a vzdelanie na to, aby mohol zaškoliť ďalších používateľov. Ak má používateľ jednu z týchto rolí na daný prístroj, tak jeho rezervácia je automaticky považovaná za platnú, inak ju musí administrátor buď odsúhlasiť alebo zamietnuť. Ak je prístroj už rezervovaný, zobrazí sa varovná správa, ktorá upozorní používateľa, že dostupnosť prístroja nie je garantovaná.

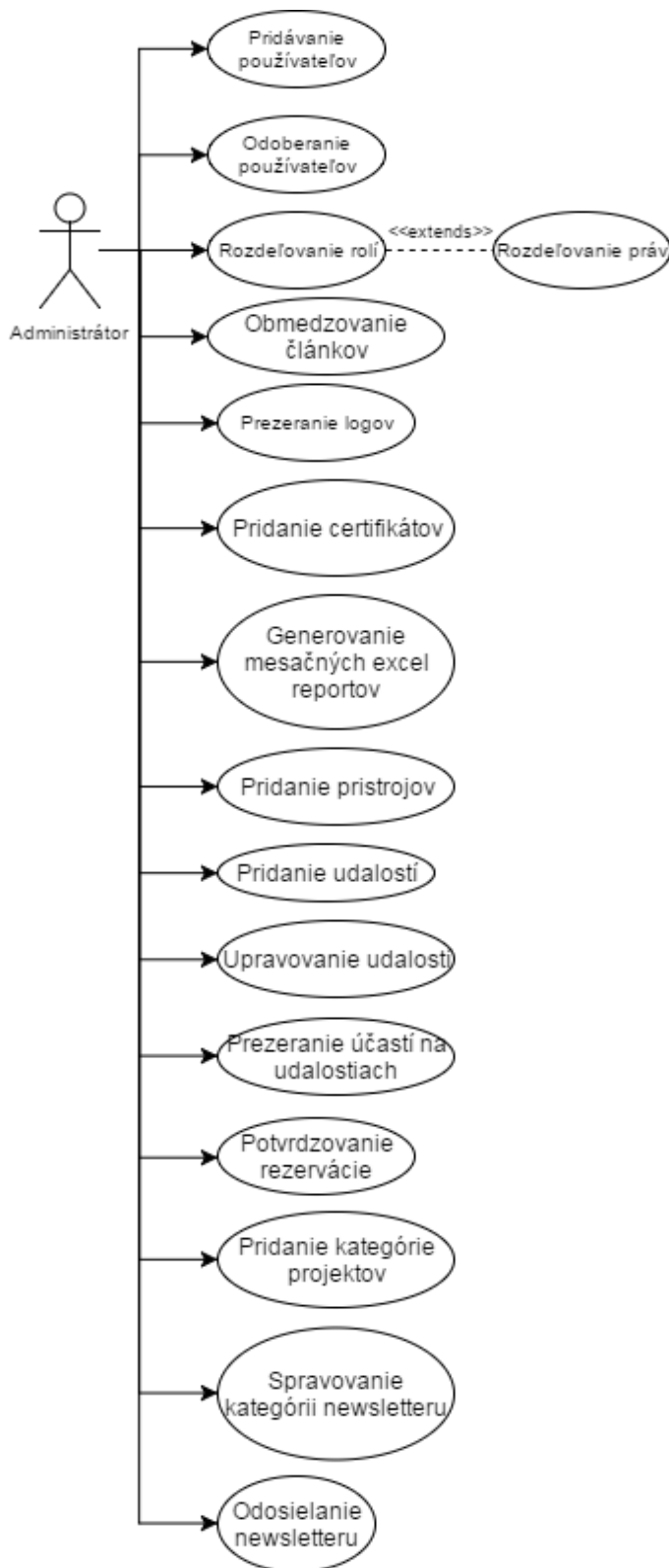
Pracovník



Obrázok 7 – use-case diagram zamestnanca, vytvorené na stránke www.draw.io

Za pracovníka považujeme každého, kto pracuje v rámci FabLabu. Každý pracovník musí evidovať svoju prácu v systéme. Tu má možnosť buď to naplánovať nejakú prácu do budúcnosti, alebo si ju sám aj priamo zapísať. Má právo editovať svoje odpracované hodiny v rámci posledného mesiaca. Naplánovaná práca môže byť buď to zrušená alebo potvrdená, v ktorom prípade sa presunie medzi odpracované hodiny.

5.1.5. Administrátor



Administrátor má práva na spravovanie celého systému a všetkých jeho vyvinutých funkcií. Medzi ne spadá aj možnosť rozdeľovať tieto práva a pridelovať používateľom role, ktoré ich rozdelia do skupín so špecifickými právami a úlohami.

Medzi jeho spravovanie spadá pridávanie či odstraňovanie zaregistrovaných používateľov, písanie článkov pre špecifických používateľov či pridávanie nových prístrojov FabLabu.

Ďalej si môže prezerat' logy ostatných používateľov, ktoré uchovávajú všetky informácie o dôležitých činnostiach používateľov, kde je vidieť dátum činnosti a podrobný popis.

Taktiež smie rozdeľovať certifikáty používateľom, potvrdzovať ich rezervácie prístrojov, pridávať kategórie projektov, do ktorých rezervácie spadajú.

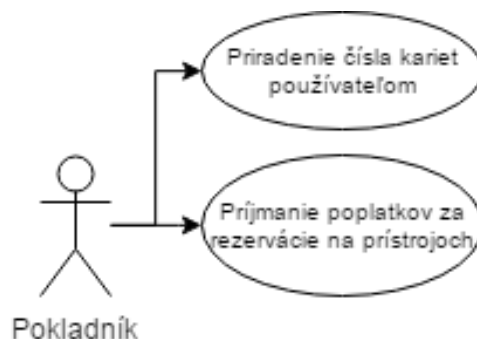
Obrázok 8 – use-case diagram zamestnanca s rolou administrátora, vytvorené na stránke www.draw.io

Naviac systém umožňuje aj rozosielanie e-mailu o aktualitách stránky (ďalej už len newsletter), pomocou SMTP e-mailového hostiteľa. Newsletter je rozdelený do kategórií, ktoré administrátor smie vytvárať a upravovať podľa svojej vôle. Odosielanie newsletteru funguje aj pre články, ktoré napíše a pomocou tagov ich priradí k jednotlivým kategóriám newsletteru a používateľom v týchto skupinách sa newsletter rozošle po publikovaní článku.

Administrátor má tiež prístup ku odpracovaným časom zamestnancov a môže si ich nielen prezerať, no aj stiahnuť vo forme Excel súboru, ktorý dodržiava všetky požiadavky FabLab a je pripravený na vytlačenie, podpísanie a archivovanie.

Posledná vlastnosť administrátora sú udalosti. Smie ich vytvárať, určiť či sú platené, obmedziť ich, neskôr ich upravovať či si prezrieť podrobné informácie o používateľoch, ktorý sa na akciu prihlásili.

5.1.6. Pokladník



Obrázok 9 – use-case diagram pokladníka, vytvorené na stránke www.draw.io

Rolu pokladníka dostáva zamestnanec, ktorý má na starosti potvrdzovanie registrácií členstva. Na základe tohto osobne príma poplatky za členstvo a pridáva používateľom ich čísla kariet. Karta používateľovi pomôže pomocou po priložení v rámci FabLabu potvrdiť svoju činnosť, rezerváciu prístroja a umožní mu prístroje používať.

Okrem prijímania poplatkov za členstvo a kartu príma poplatky za používanie prístrojov, ktoré sú evidované v rámci systému a uľahčujú mu vypočítavanie cien za služby, prezranie histórie zaplatených poplatkov a označenie nových poplatkov za zaplatené. Tieto sú potom archivované ako potvrdené spolu s cenou, ktorá bola za ne zaplatená.

5.2. Štandardy NF databázy a jej model

5.2.1. Čo je štandard NF?

Jedná sa o databázovú normálovú formu, ktorej úlohou je aby boli dáta prehľadné, zmysluplné a uľahčuje reprezentáciu kolekcii dát [13]. Rozdeľujeme ju do troch kategórií a našim cieľom je dosiahnuť objektovú normálnu formu 3NF všetkých našich tried a tým pádom aj našej schémy.

Naša trieda je v 1NF, ak zabránime zbytočnému opakovaniu dát. Toto zaistíme tak, že ak sa nám nejaké dáta opakujú, oddelíme ich do zvlášť tabuľky a pomocou špecifického identifikačného znaku (ďalej už len ID) napojíme na predošlý prvok, pod ktorý patrila. Príklad v našej databáze môže byť používateľ a jeho certifikáty. Vieme, že jeden používateľ môže mať až desiatky certifikátov a tak nemá zmysel priamo v tabuľke o používateľoch udržiavať aj dáta o jeho certifikátoch. Radšej ich premiestnime do zvlášť tabuľky, v našom prípade pristroje_certifikáty, kde udržiavame informáciu ku ktorej ID používateľa patrí ktorý certifikát.

Ďalej našu triedu preformulujeme do 2NF tak, že ak dve triedy zdieľajú nejaké dáta, tak ich premiestnime do zvlášť tabuľky. V našom prípade sa môže jednať o pozvánky do projektov. Pozvánka bola niekým odoslaná pre niekoho. Každopádne nemá zmysel tento údaj udržiavať pre každého zvlášť, keďže ho zdieľajú. Radšej sme vytvorili tabuľku pristroje_projekty_pozvanky, kde uchováваме informáciu o tom od koho a komu bola aká pozvánka odoslaná.

Tretiu a finálnu normovú formu ktorú chceme dosiahnuť, 3NF dosiahneme, ak rozdelíme dáta podľa ich významu. Každá tabuľka databázy by mala reprezentovať len jednu skupinu dát, ktoré spĺňajú len jeden účel. Takýto príklad môžeme vidieť v našej databáze, kde sme rozdelili tabuľky potvrdených (pristroje_potvrdene_formulare) a nepotvrdených rezervácií (pristroje_nepotvrdene_formulare). Mohli by sme síce použiť jednu tabuľku s pridanou hodnotou či bola alebo nebola rezervácia potvrdená. Toto by však spravilo prácu s dátami ťažšiu a prehľad samotnej databázy, keďže by bola špecifikácie len pristroje_formulare.

5.2.2. Model databázy

Našu databázu vieme rozdeliť podľa jednotlivých skupín, ktoré neskôr vytvorili samostatné rozšírenia. Máme zvlášť skupinu tabuliek na newsletter, kde evidujeme kategórie newsletteru na ktoré sa dá prihlásiť a kto už prihlásený je.

Druhou skupinou sú udalosti. Sem spadá logovanie všetkého, čo používateľ spraví a pridávanie udalostí ako napríklad semináre, stretnutia, pozvánky na stretnutie, pričom musíme udržiavať aj informáciu, kto sa na našu udalosť už prihlásil. Posledné do tejto časti spadá aj newsletter, ktorý je rozdelený na kategórie na ktoré sa môže používateľ prihlásiť a administrátor ho vie následne informovať o novinkách v tejto kategórii.

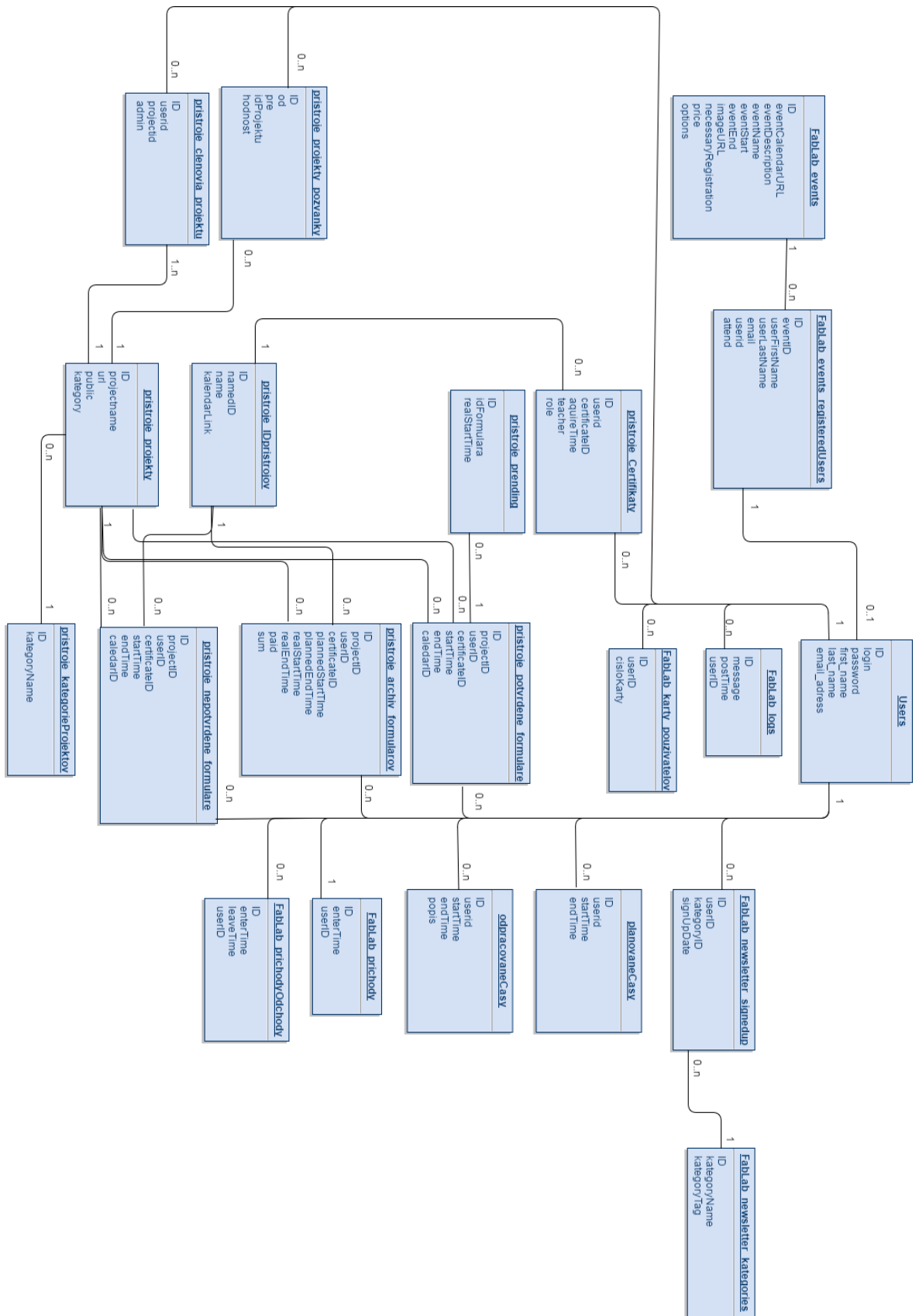
Za tretiu skupinu údajov v rámci databázy považujeme prístroje. Najprv potrebujeme mať samotnú tabuľku prístrojov (pristroje_idPristrojov), ktorá obsahuje špecifickú ID prístroja a jeho názov. Ďalej ak sa chce používateľ prihlásiť na použitie prístroja, musí mať kartu ktorou sa na neho prihlási. Taktiež potrebuje projekt, do ktorého si prácu na prístroji zapíše.

Projekt musí spadať práve do jednej z dostupných kategórií. Ak je projekt verejný, môže sa k nemu pridať hocikto. V inom prípade ak chceme s niekým spolupracovať na projekte, môže nás do neho pozvať.

Okrem projektu sa pri rezervácii kontroluje, či na danom prístroji vieme samostatne pracovať. Ak áno, tak naše rezervácie sú okamžite potvrdené, inak idú na posúdenie administrátorom.

Potom, ako si používateľ zarezervoval prístroj a vybavil kartu aby sa mal ako prihlásiť, musí prísť do FabLabu, priložiť pri vchode kartu, čím sa eviduje jeho príchod. Príde k prístroju a po priložení jeho karty vieme, ktorá rezervácia práve prebieha. Po ukončení činnosť na prístroji archivujeme a čaká na zaplatenie pokladníkom. Po zaplatení sa už rezervácia ďalej nemení a je považovaná za archivovanú. Pri odchode zo zariadenia FabLab si používateľ znova priloží kartu aby potvrdil svoj odchod z objektu pričom jeho interakcia s FabLabom bola úspešne ukončená.

Posledná skupina tabuliek slúži výlučne zamestnancom. Zapisujú si svoj plán práce, ktorý neskôr spolu s nepovinným popisom práce potvrdzujú, čím sa eviduje ich práca pre zamestnávateľa.



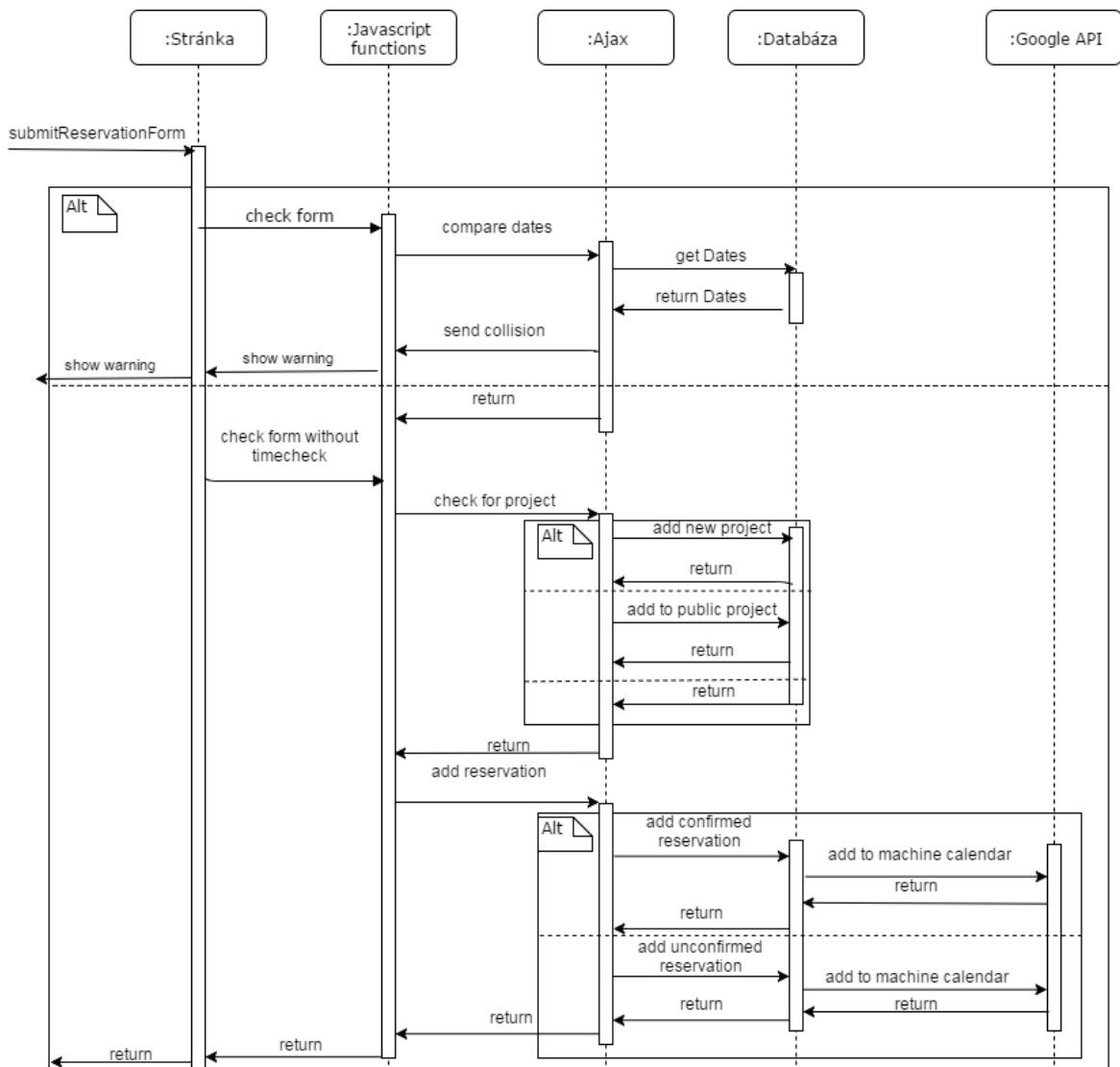
Obrázok 10 – UML diagram spojitostí tabuliek databázy, vytvorené na stránke www.draw.io

5.3. Sekvenčný diagram rezervácie zariadenia

5.3.1. Čo je a na čo slúži?

Sekvenčný diagram je jeden z UML diagramov, ktorý slúži na zobrazenie jedného konkrétneho scenára, ktorý spustí náš si používateľ. Tento scenár zobrazuje ako komunikáciu medzi objektami informačného systému v sekvenčnom poradí, ako aj alternatívy, požiadavky a odpovede systému [14].

5.3.2. Sekvenčný diagram rezervácie prístroja s popisom



Obrázok 11 – Sekvenčný UML diagram znázorňujúci činnosť rezervácie prístroja, vytvorené na stránke www.draw.io

Z následného sekvenčného diagramu môžeme vidieť, že používateľ po odoslaní formy hneď narazí na dve alternatívy kontroly. Prvá prebieha vždy prvý-krát, keď sa snaží

system zistiť, či sa náhodou neprekrývajú časy s inými rezerváciami. Ak áno, tak toto oznámi používateľovi ako varovanie, spolu s možnosťou ignorovať iné rezervácie. Pri tomto rozhodnutí používateľ ale riskuje nedostupnosť zariadenia. V prípade ak kontrola časov prebehla úspešne a nenašli sa žiadne kolízie, pokračuje tam, kde začína registrácia bez tejto kontroly.

Ďalej prebieha kontrola projektu, do ktorého má rezervácia spadať. Tu môžu nastať tri možnosti:

- Používateľ pridáva s rezerváciou aj nový projekt, ktorý je potrebné skontrolovať a pridať do databázy tak ako projekt aj členstvo používateľa.
- Používateľ sa pridáva k verejnému projektu, ktorého nie je členom. V tomto prípade potrebujeme len používateľa zapísať ako nového člena projektu.
- Používateľ pridáva rezerváciu do projektu, ktorého je členom. V tomto prípade nemusíme nič robiť, keďže projekt už existuje a používateľ je už členom.

Posledný krok je samotné pridanie rezervácie. Tu máme dve alternatívy, buď používateľ je zaučený používať prístroj, alebo nie. Ak je zaučený aspoň na úrovni „Zaučený“, tak sa rezervácia pridá rovno medzi potvrdené, keďže nepotrebuje dozor. Ak ale nemá rolu, jeho rezervácia čaká na schválenie.

V oboch prípadoch sa musí systém napojiť na Google API, aby mohol odoslať do kalendára informácie o rezervácii, kedy sa koná a či je potvrdená. Tu prebieha samotná kontrola validity access tokenu Google API a prípadné obnovenie access tokenu pomocou refresh tokenu.

Keď je toto všetko úspešne ukončené, odošle sa používateľovi správa o úspechu rezervácie na prístroji.

5.4. Podrobná špecifikácia cieleného informačného systému

Cielený informačný systém je nadstavba pre aktuálny systém. Vyžaduje prevedenie aktuálneho webu do zvoleného frameworku, kde budú nasledujúce vlastnosti pridané a implementované.

Taktiež by som k tejto sekcii poznamenal, že FabLab síce chce integrovať povinné členstvo, no zatiaľ sa tak nestalo. Práve preto sú všetky rezervácie na prístroje obmedzené len požiadavkou registrácie, aby kým bude FabLab na túto zmenu prichystaný mohli používatelia stále využívať služby FabLabu a registrácie mohol odsúhlasovať administrátor aj používateľom bez karty.

5.4.1. Registrácia

FabLab informačný systém si udržuje informácie o používateľoch, umožňuje im sa registrovať pomocou FabLab formulára, kde používateľ musí vyplniť nasledujúce polia:

- Používateľské meno
- Heslo
- Kontrolu hesla
- Captcha ochrana
- Krstné meno
- Priezvisko
- E-mailová adresa
- Súhlas so spracovaním osobných údajov

Taktiež poskytuje používateľovi voliteľné polia adresy jeho internetovej stránky (webu), ikonku používateľa a životopis.

Registrácia je potvrdzovaná pomocou dodanej e-mailovej adresy, po ktorej sa až stane používateľ registrovaným členom FabLab informačného systému.

5.4.2. Obmedzenie článkov

Administrátor môže obmedziť komu sú zobrazované ktoré články a vytvárať rôzne skupiny používateľov.

5.4.3. Udalosti

FabLab často koná mnohé udalosti, niektoré zadarmo a iné platené. Preto potrebujú jednoduchý systém, ktorým vygenerujú registračný formulár pre používateľov na prihlásenie

na tieto udalosti. Tento registračný formulár obsahuje následné údaje, ktoré môže aj po vytvorení administrátor upraviť:

- Názov udalosti
- Začiatok a koniec udalosti
- Popis
- Obrázok udalosti (voliteľné)
- Či musia byť účastníci zaregistrovaní na stránke
- Cenu za udalosť
- Možnosti účasti (výber pre používateľa, ako hodnotí svoju účasť, napríklad „Áno“, „Nie“, „Možno“ atď.)

Formulár sa používateľom zobrazí ako informácia o udalosti, pod ktorou bude prihlasovací formulár na udalosť, ktorý obsahuje následné povinné polia:

- Účasť (výber z možností účasti)
- Krstné meno
- Priezvisko
- E-mailovú adresu
- Súhlas so spracovaním osobných údajov podľa pravidiel FabLabu

V prípade, že je cena stanovená iná ako „zadarmo“, prihlásenie na udalosť možnosťou „Áno“ je považované za záväzné a nemôže si ho používateľ zmeniť. Jediná možnosť ako tento údaj zmeniť je požiadať administrátora FabLabu. Ak je cena zadarmo, na zmenu účasti používateľa mu stačí zadať e-mailovú adresu a vybrať inú možnosť účasti. Po odoslaní formulára bude jeho účasť aktualizovaná. Ak je používateľ v systéme už zaregistrovaný evidujeme aj jeho ID a automaticky za neho formulár vyplníme.

Administrátor má prístup k prehľadu informácií prihlásených používateľov a môže ich z udalosti odstrániť.

5.4.4. Newsletter

Newsletter je v cielenom informačnom systéme rozdelený do kategórií, ktoré administrátor pridáva a používateľ sa na ne môže následne prihlásiť. Administrátor môže buď posilať newsletter priamo vybraným kategóriám používateľom, s možnosťou poslať testovací e-mail na zvolenú e-mailovú adresu alebo pomocou tagu danej kategórie môže používateľom poslať článok, ktorý práve publikoval.

Pre prihlásenie na newsletter je potrebné aby bol používateľ registrovaným používateľom FabLabu. Po prihlásení má v menu na výber, na ktoré kategórie newsletteru sa chce prihlásiť. Na potvrdenie prihlásenia na newsletter požadujeme od používateľa, aby súhlasil s pravidlami newsletteru FabLabu. Až po odsúhlasení týchto pravidiel sa môže používateľ na newsletter prihlásiť, ohlásiť alebo aktualizovať svoj newsletter.

5.4.5. Logovanie

Cielený informačný systém musí zaznamenávať činnosti či už zamestnancov alebo používateľov vo forme logu. Log je uložený štýlom správa a čas, kedy sa tento log udial. Každý používateľ má právo prezerat' svoje logy, aby mohol sledovať nielen svoju činnosť ale aj interakcie medzi jeho účtom, administráciou a inými používateľmi. Administrátor má právo prezerat' logy všetkých používateľov v prípade potreby.

5.4.6. Rezervačné formuláre prístrojov

Jeden z našich cieľov je nahradiť existujúci spôsob prihlasovania na prístroje a automatizovať ho. Preto pre každý prístroj má vlastný vygenerovaný formulár, cez ktorý sa naň môžu používatelia prihlásiť. Formulár vytvoríme pomocou systému, kde napíšeme názov prístroja a jeho ID. Vygeneruje sa každému prístroju pri pridaní jeho Google kalendár, do ktorého sa všetky rezervácie zapisujú. Tento formulár pri rezervácii obsahuje zoznam ľudí, ktorí majú hodnotu „Zaučený“, „Dozor“ a „Školiteľ“ a formulár na rezerváciu prístroja pre používateľa.

5.4.7. Projekty

Každá rezervácia musí spadať do nejakého projektu. Každý projekt má svoju kategóriu, ktoré určuje administrátor. Členovia všetkých typov projektov môžu svoj projekt hocikedy opustiť.

Poznáme dva typy projektov. Prvý je verejný, v ktorom nikto nie je administrátor a hocikto sa bez žiadosti môže k nemu pridať. Taktiež prehľad všetkých verejných projektov je otvorený verejnosti. Druhé sú súkromné projekty, ktorých je administrátorom osoba, ktorá ich vytvorila. Do projektu môže následne poslať pozvánku používateľom a určiť, či smú administrovať projekt tiež.

5.4.8. Priradenie certifikátov

Ako sme už spomínali, používateľ môže na danom prístroji pomocou kurzu/zaučenia získať tri role, „Zaučený“, „Dozor“ a „Školiteľ“. Zaučený je používateľ, pokiaľ je schopný samostatne bez dohľadu pracovať na zariadení. Dozor je osoba ktorá bola nie len zaučená,

ale je zodpovedná dostatočne na to, aby robila dozor používateľom, ktorý nemajú ani hodnosť „Zaučený“. Školiteľ je osoba, ktorá má práva oboch predošlých skupín, no navyše má právo zaučiť používateľa na prístroj, podľa role ktorú sám usúdi.

5.4.9. Rezervácia prístrojov

Ak sa chce používateľ prihlásiť na prístroj, musí vyplniť formulár na rezerváciu, ktorý obsahuje nasledujúce prvky:

- Začiatok a koniec práce na prístroji
- Výber projektu
 - Výber z projektov, ktorých je používateľ členom
 - Dostupné verejné projekty ktorých nie je členom
 - Nový projekt
 - Názov projektu
 - Či je projekt verejný
 - Stránka projektu (dobrovoľné)
 - Kategória projektu

Formulár má svoje kontroly a obmedzenia a oznamuje používateľovi vždy, keď niečo vo formulári nie je dobré. Prvý prvok kontroly je, že čas rezervácie sa nesmie nachádzať v minulosti ani trvať dlhšie ako osem hodín, avšak nie je zakázané aby ak používateľ naozaj potreboval prístroj na dlhšiu dobu si vytvoril viacero rezervácií na ten istý prístroj. V prípade, ak sa používateľova rezervácia prekrýva s rezerváciou niekoho iného, tak dostane varovnú správu o potenciálnej nedostupnosti zariadenia kvôli inej rezervácii a možnosť sa rezervovať napriek tomuto varovaniu, pričom FabLab neručí za dostupnosť zariadenia.

V prípade nového projektu sa kontroluje fakt, aby osoba nemala dva súkromné projekty s rovnakým názvom, avšak verejné projekty sa v názve zhodovať môžu a taktiež dva súkromné projekty dvoch rôznych používateľov.

Všetky, aj nepotvrdené aj potvrdené rezervácie sú zapísané v kalendári prístroja, ktorý je verejne dostupný používateľom. Takto majú prehľad aj administrátori aj používatelia o vytážení daného prístroja.

5.4.10. Spracovanie rezervácií

Administrátor, alebo osoba ktorej bolo priradené právo na spracovanie rezervácií smie prezerat' všetky potvrdené a nepotvrdené časy používateľov na prístrojoch. Má k dispozícii potvrdiť rezerváciu osoby ktorá potrebuje dozor, ak vie že dozor bude dostupný. Pri potvrdení sa vymaže z kalendára nepotvrdená rezervácia a nahradí sa potvrdenou. Taktiež smie hocikoho hocikedy odhlásiť z prístroja alebo odmietnuť nepotvrdenú žiadosť v prípade potreby. Všetky takéto akcie sa však veľmi presne logujú a používateľ presne vie, kedy a kto mu rezerváciu zrušil.

Ďalšia jedinečná vlastnosť administrátora je, že môže priamo označiť prácu na prístroji ako vykonanú, v ktorom prípade sa za skutočný čas na prístroji ráta ako plánovaný, zadaný počas vytvárania rezervácie.

5.4.11. Použitie prístrojov a karty používateľov FabLabu

Administrátor smie pridelit' používateľovi kartu, ktorá znamená že má prístup k prístrojom a zaplatil členský poplatok. Bez platnej karty sa prístroje nedajú použiť a používateľove rezervácie sú zbytočné.

Pri vstupe do budovy používateľ priloží kartu, a systém ho privíta v objekte a pri odchode je potrebné priložiť kartu taktiež. Údaj o jeho príchode a odchode sa zapíše do jeho logu, kde si môže administrátor prezrieť, koľko a kedy sa nachádzal v priestoroch FabLabu.

Pri použití prístroja priloží používateľ kartu k čítačke, ktorá zistí podľa toho ku ktorému prístroju je pripojená, či má v najbližšej dobe ku prístroju osoba rezerváciu alebo či neprišiel neskoro na nejakú svoju prebiehajúcu rezerváciu. V prípade, že sa žiadna rezervácia nenašla, napíše používateľovi varovnú správu. Inak napíše správu používateľovi a potom môže začať prístroj používať. Keď na prístroji používateľ skončí, priloží opäť kartu a systém zaregistruje informáciu o konci používania prístroja. Priloženie karty zapisuje napriek plánovanému času práce čas reálny, za ktorý sa reálne platí, pričom z plánovaného času sa stáva len informácia pre administrátora.

5.4.12. Platenie rezervácií

Každý kto využije prístroje FabLab a ukončí na nich svoju činnosť musí za prácu zaplatiť. Preto pre zamestnanca FabLabu, ktorá spĺňa rolu pokladníka existuje v systéme aj možnosť platenia rezervácií. Zobrazuje všetky nezaplatené aj zaplatené činnosti na prístrojoch, ktoré používateľ vykonal, vyrátava reálny odpracovaný čas a ponúkne pokladníkovi vybrať z nezaplatených rezervácií, pripísať k nim cenu na hodinu a systém mu

spätne napíše koľko stála práca na jednotlivých prístrojoch a koľko má používateľ zaplatiť spolu. Po odsúhlasení o zaplatení sa vybrané práce na prístrojoch archivujú aj s celkovou cenou zaplattenou.

5.4.13. Evidencia práce zamestnancov

Každý zamestnanec FabLabu eviduje svoju prácu pomocou systému. Systém ponúka plánovanie práce zamestnanca, ktorú neskôr môže popísať a potvrdiť ale aj vymazať. Druhou možnosťou je, že si zamestnanec priamo zapíše odpracovaný čas. V prípade chyby má zamestnanec ešte stále právo editovať odpracované časy za daný mesiac.

Administrátor má prístup ku všetkým odpracovaným časom z ľubovoľného obdobia a pre ľahšiu administráciu si môže daný mesiac stiahnuť ako Excel súbor, ktorý spĺňa všetky štandardy FabLabu a CVTI, je prichystaný na podpísanie a archivovanie.

5.5. Implementácia

Popis implementácie obsahuje jednotlivé, zatiaľ nespomenuté prvky, s ktorými som sa stretol v rámci vývoja informačného systému a ako som vyriešil jednotlivé požiadavky na systém.

5.5.1. Filtrovanie

Počas práce so systémom často používateľ docení, že ak si chce napríklad prezrieť informácie o konkrétnom verejnom projekte, tak som využíval ako filter vlastnosť MySQL funkcie LIKE [15], ktorá skontrolovala všetky vlastnosti podľa vstupu používateľa a vďaka JavaScriptovej funkcii ktorá sa spúšťa počas akcie HTML input elementu onChange a v reálnom čase zmení výber pre používateľa podľa jeho vstupu.

V prípade filtrovania používateľov sme nepoužívali MySQL, ale funkcie dodané WordPressom WP_User_Query [16], ktorá prijme výraz, obmedzenie na polia, ktoré má prehľadávať a aké polia chceme aby vrátila. Príklad z našej aplikácie je AJAX žiadosť ktorý od servera tieto výsledky získal:

```
$args = array(
    'search' => '.*$_POST[ 'vstup' ].*'
    'search_columns' => array( 'user_login', ),
    'fields' => array( 'id', 'user_email', 'user_login' ));
$user_query = (new WP_User_Query( $args ))->get_results();
```

V tomto príklade sme filtrovali všetkých používateľov, ktorý mali svoje prihlasovacie meno podobné so vstupom od používateľa.

5.5.2. Admin a user menu

Menu je rozdelené v rámci nášho systému podľa WordPressových takzvaných capabilities (spôsobilosti). Každý používateľ môže mať vlastné capabilities pridelené administrátorom, no väčšinou je zvykom vytvoriť pomocou WordPressového systému rolu. Na tento účel som hľadal vhodné, už existujúce rozšírenie a našiel som ako najlepšieho kandidáta s názvom „Advanced Access Manager“. Medzi jeho vlastnosti patrí chcené obmedzovanie obsahu podľa rolí, vrátane obmedzenia článkov, ktoré FabLab požadoval. Taktiež sa dajú v ňom ľahko vytvoriť nové role a priradiť im nejakú z existujúcich capabilities.

Vďaka tomuto rozdeleniu vieme vytvoriť používateľa s capability level_0, ktorým bude každý práve zaregistrovaný používateľ. Takto dostane prístup k administratívnejmu menu, avšak len sekciám vytvorených mnou pre používateľov FabLabu.

Naopak, zamestnanec má capabilities rozšírené, napríklad eviduje_pracu, ktorá mu umožňuje prístup do menu na evidenciu vlastnej práce, ale nie ku extra menu ktoré umožňuje prezerat' prácu všetkých zamestnancov. Táto vlastnosť je vyhradená skupine administrátor, ktorým by mala byť len vedúca osoba ktorá preberá zodpovednosť za funkčnosť systému a FabLabu.

5.5.3. Registračný formulár

Registračný formulár bolo potrebné rozšíriť zo základného tak, aby nie len obsahoval extra polia, ale aby boli aj povinné. Preto som mal aj za úlohu nájsť rozšírenie, ktoré by bolo neplatené, bez reklám a pomohlo by pri vytvorení týchto políček v rámci registrácie a obmedzilo ich.

Ako najlepší kandidát na túto funkčnosť vyšiel „Cimy User Extra Fields“, ktorý spĺňal všetky chcené požiadavky a pridával aj nadmernú kontrolu nad obsahom profilu a registrácie používateľov. Po krátkej dobe testovanie sa mi podarilo vytvoriť pomocou tohto rozšírenia registráciu požadovanú FabLabom.

5.5.4. Formátovanie dátumov

Častý problém s ktorým som sa stretával bol formát dátumu. JavaScript interpretoval čas ako "2017-04-1T21:28:38.567Z", HTML mali vstupy typu "2017-04-01T21:28", MySQL ako „2017-04-01 23:28:56“ a používateľ chcel ideálne vidieť čas zobrazený ako 23:28 01.04.2017. Kvôli tomuto sme často museli konvertovať čas z jedného typu na druhý a použili sme na to rôzne funkcie.

Keď sme získavali dáta z databázy a chceli sme ich len zobrazit' používateľovi, využili sme MySQL funkciu DATE_FORMAT (time, "%T %d.%m.%Y"), ktorá preformátovala databázový čas do nami zvoleného formátu.

Ak sme chceli pridať dátum do databázy, museli sme použiť konverziu poskytnutú PHP date("Y-m-d H:i:s", strtotime(\$_POST['date])), ktorá skonvertovala string do času a vložila ho v správnom formáte do databázy,

Posledná konverzia dátumu prebiehala, ak sme chceli, aby sa zamestnancovi napríklad pri evidencii práce automaticky vyplnil čas podľa aktuálneho času a dátumu. Na

toto sme využili jQuery, keďže disponuje funkciou jQuery(document).ready, ktorá sa spustí práve po načítaní dokumentu. Obsahovala nasledovnú funkciu, ktorá vyrátala časový rozdiel a vložila ho do všetkých dátových vstupoch.

```
var date=new Date();
date.setTime( date.getTime() + Math.abs(date.getTimezoneOffset()*60*1000 );
jQuery('input[type=datetime-local]').val(date.toJSON().slice(0,16));
```

5.5.5. Sanitácia

Napriek vyjadreniu vedenia FabLabu, že systém a stránka sú založené na dôvere používateľom, snažil som sa dodržiavať základnú bezpečnosť, zvlášť ak vstupné dáta pochádzali od bežného používateľa. WordPressová trieda WPDB síce ako som už spomínal implementuje sanitáciu, no nechcel som vykonávanie príkazov, ktoré sú prevažne vo forme SQL forme nechať na náhode, ak by WordPress v týchto poliach sanitáciu nepodporoval. Na sanitáciu som teda využil PHP príkaz:

```
sanitize_text_field( $_POST['vstup'] )
```

5.5.6. Shortcode

V rámci WordPressu poznáme tri spôsoby ako ovplyvniť pomocou rozšírení funkčnosť stránky. Pridaním filtra, ktorý sa vykoná vždy keď sa spustí konkrétna akcia a získa jej dáta, pomocou akcie, ktorá sa vykoná vždy keď sa vykonáva niečo iné a pomocou shortcodeu.

Shortcode slúži na generovanie textov hocikde v rámci systému, kde to administrátor potrebuje. Shortcode je naviazaný na špecifický názov, ktorý keď je vložený do hranatých zátvoriek ([Názov]), tak sa pri každom načítaní dokumentu spustí shortcode a nahradí tento riadok hodnotou, ktorá je shortcodom navrátená [17].

Túto technológiu používame veľmi často, či už pri zobrazovaní verejných projektov, generovaných registračných formulárov alebo udalostiach. Dáva adminovi taktiež slobodu určiť, kde sa kód zobrazí. Či už v článku, na vlastnej stránke alebo pod už existujúcim prístrojom.

5.5.7. publish_post

Špecifický tag publish_post, používaný v našom kontexte ako add_action('publish_post', 'sendNewsletters',10,2), je jeden z hookov (háčikov), ktorým vieme získať všetky dáta o článku, počas jeho publikácie [18]. Museli sme ale vedieť, ako presne chceme tieto dáta

získať. Vytvorili sme akciu, ktorá spustí funkciu a posledné dve čísla určujú akej priority je táto akcia (poradie vykonania) a koľko premenných od nej očakávame. Priorita desať nám zaručí, že sa naša akcia vykoná ešte pred funkciou publikácie a počet argumentov dva, že získame tak ako aj ID článku, aj všetky ostatné dáta. Medzi dáta spadá pre nás hlavička, dôležitá správa a tagy, pomocou ktorých určujeme, kto z newsletteru získa túto správu.

6. Inštalačná príručka

Táto inštalačná príručka vysvetľuje prvé kroky ako úspešne začať využívať vyvinuté rozšírenia pre FabLab.

6.1. Požiadavky:

- Nainštalovaná najnovšia verzia WordPressu
- PHP 7.0 +
- Nastavená databáza pre WordPress stránku
- Nainštalovanie dvoch voľne dostupných rozšírení
 - Advanced Access Manager (www.WordPress.org/plugins/advanced-access-manager/)
 - Cimy User Extra Fields (<https://sk.WordPress.org/plugins/cimy-user-extra-fields/>)
- Funkčnú e-mailovú službu nainštalovanú na webe WordPress, ktorá podporuje a používa funkciu wp_mail

6.2. Postup

Prvý krok inštalácie je prichystať web na nové rozšírenia. Toto dosiahneme v pár krokoch. Najprv potrebujeme mať vytvorenú a povolenú cez Google účet Google API, ktorá má vygenerovaný O2Auth client_secret.json súborom. Toto dosiahneme, ak vytvoríme na stránke <https://console.developers.google.com/> projekt, aktivujeme v rámci neho Google Calendar API, vyplníme Consent Screen a následne vytvoríme O2Auth secret po kliknutí na Create credentials – O Auth Client ID – Other – Create. Po vyplnení by ste mali získať nový client_secret, ktorý by mal byť stiahnuteľný. Zmeňte názov na client_secret.json a umiestnite ho do rozšírení „FabLab-eventy“ a „FabLab-pristroje“. V konzoly spustíte quickstart.php pomocou príkazu „PHP quickstart.php“. Toto vygeneruje Váš access a refresh token, a uloží ho na vašom serveri [19].

Po tomto kroku môžete postupne rozšírenia aktivovať. Avšak pre ich plnú funkčnosť je stále potrebné ich najprv nastaviť. Zájďte do nastavení, sekcia „Nastavenia FabLabu“. Vytvorte si jeden kalendár kam sa môžu zapisovať údaje o udalostiach FabLabu a skopírujte z jeho nastavení do WordPress nastavení naňho odkaz. Vyplnenie poľa „Akceptovaná adresa“ obsahuje IP adresu, ktorá má právo využívať prihlásenie na prístroje. V prípade localhostu sa jedná vždy o IP „:1“. Vyplňte zvyšné polia ako uznáte za vhodné.

Na vytvorenie požadovanej rezervácie FabLabom použijeme vyššie spomenutý rozšírenie „Cimy User Extra Fields“.

Najprv pre základné nastavenia „Cimy User Extra Fields“ majte označené len nasledujúce polia:

- Show all fields in the welcome email
- Enable email confirmation
- Enable form confirmation
- Enable Securimage Captcha
- Hide name field
- Hide email field
- Hide website field
- Hide n. posts field
- Show username
- Show password
- Show confirmation password
- Show password strength meter
- Show first name
- Show last name
- Show website
- Show Biographical Info

Pre lepší vizuálny dojem odporúčam vymazať obsah možnosti „Extra Fields section title“. Momentálne nám chýba v registrácii ešte jedno pole a to je súhlas so spracovaním osobných údajov. Na tento účel existuje na spodnej časti nastavení Cimy časť „Add a new Field“ a nastavte nasledovné vlastnosti novému poľu:

- Meno: ľubovoľné
- Type: checkbox
- Fieldset: no fieldset
- Label: „Súhlasím so spracovaním osobných údajov podľa [FabLab podmienok](LINK)“ kde namiesto LINK je odkaz na pravidlá
- Nastaviť nasledujúce polia ako zaškrtnuté:

- Can be empty
- Should be equal TO: YES
- Show the field in the registration
- Ostatné polia nechajte neoznačené a prázdne.

System je pripravený na používanie, konzultujte „Príručka na administráciu“ pre informácie ako pracovať s informačným systémom.

7. Záver

Cieľom bakalárskej práce bolo vytvoriť nadstavbu z hľadiska architektúry a informačného systému stránok FabLab pomocou WordPress rozšírení.

Všetky rozšírenia boli testované lokálne a pravidelne prezentované zadávateľovi a školiteľovi. Nová verzia stránok FabLab bola nakoniec vyvinutá v systéme WordPress vo forme rozšírení, vytvorených pre najnovšiu verziu PHP (7+), čím sme umožnili ľahšiu administráciu obsahu samotnej stránky správcovi FabLabu, nespomínajúc samotné rozšírenia. Vytvorený informačný systém momentálne čaká na nasadenie na server, ktoré nastane po aktualizácii na najnovšiu verziu PHP 7+. Preto aktuálna verzia mojej práce ešte nie je implementovaná v systéme FabLab stránok. Počas vývoja som sa snažil dodržiavať všetky štandardy podľa najnovších štandardov ES6, PHP a WordPressu.

Aj napriek tomu že moja práca rieši všetky problémy, ktoré som spomínal v sekcii „Problémy aktuálneho systému“, keď spätne hodnotím vlastnú prácu videl by som pár nedostatkov v oblasti čistého kódu a potenciál na ďalší vývoj.

Počas vývoja systému sme dlhšie uvažovali nad tým, aký základný systém použiť na rozšírenie o požadované funkcionality. Týmito alternatívami boli MediaWiki, Odo a WordPress, vďaka čomu som si rozšíril svoj obzor o vlastnostiach, výhodách a nevýhodách týchto systémov.

Z programovacej stránky som sa naučil používať PHP, Google PHP API, WordPress a jQuery s ktorými som doteraz nepracoval a považujem ich za podstatné nové vedomosti do mojej kariéry. Taktiež som si vyskúšal ako fungujú systémy Odo, MediaWiki a VPN. Získal som aj zaujímavé skúsenosti s interakciou s rozsiahlejšou organizáciou, akou CVTI SR je, ktorá zavádza rozličné interné regulácie pre vývoj informačných systémov.

Počas tvorby práce som získal reálne skúsenosti so spoluprácou pri vývoji informačného systému pre väčšie organizácie ako sú CVTI SR a FabLab Bratislava. Napríklad pri vyjasňovaní obsahu systému a jeho návrhu na stretnutiach so školiteľom a zadávateľom. Vďaka týmto stretnutiam som vytvoril úplný zoznam požiadaviek a základný návrh systému, z ktorého bola moja práca vytvorená.

Aj keď momentálna nadstavba informačného systému implementuje veľa nových funkcií a možností pre používateľov, stále viem navrhnúť ďalšie body pre budúci rozvoj. Medzi ne patrí SSO (Single-sign on), ktorý CVTI síce ešte nemá implementovaný, no

vývoj CVTI stránok k tomu smeruje. Ďalej prihlásenie pomocou napríklad Facebook účtov alebo Google účtov by taktiež mohlo používateľom uľahčiť používanie webu, prípadne po pridaní akcií by sa mohli informácie o udalosti automaticky poslať na FabLab Facebook stránku a vytvoriť udalosť aj tam, ak by to bolo možné.

8. Literatúra

- [1] Centrum vedecko-technických informácií SR, FabLab: O nás, 2016. [online] https://www.fablab.sk/sk/o_nas/, prístupované 5.1.2017
- [2] Helcz, Branislav, Vývoj a implementácia informačného systému v konkrétnej organizácii: Bakalárska práca, Vedúci: RNDr. Martin Nehéz, Trenčín: Vysoká škola manažmentu, 2010 [online] <http://www.cutn.sk/Library/Thesis/2010/Helcz.pdf>
- [3] Knechta, Dominik, Aktualizácia webu katedry: Bakalárska práca, Školiteľ: Mgr. Ján Kľuka, PhD., Bratislava: Univerzita Komenského, 2015 [online] www.st.fmph.uniba.sk/~knechta2/bakalarka/
- [4] MediaWiki, MediaWiki : About. 2016, MediaWiki Foundation. [online] <https://www.mediawiki.org/wiki/Project:About> prístupované 5.4.2017
- [5] MediaWiki, Security issues with authorization extensions, MediaWiki Foundation. [online] https://www.mediawiki.org/wiki/Security_issues_with_authorization_extensions prístupované 5.5.2017
- [6] WordPress. org, WordPress: About, 2016, [online] <https://WordPress.org/about/> prístupované 6.1.2017
- [7] Google Developers, PHP Quickstart, 2016, [online] <https://developers.google.com/drive/v3/web/quickstart/php> prístupované 3.3.2017
- [8] Segue Technologies, What is Ajax and Where is it Used in Technology?, 2013, [online] <http://www.seguetech.com/ajax-technology/> prístupované 6.1.2017
- [9] Autor neznámy, Introducing JSON, [online] <http://www.json.org/> prístupované 17.5.2017
- [10] WordPress Codex, Class Reference/wpdb, 2016, [online] https://codex.wordpress.org/Class_Reference/wpdb prístupované 6.5.2017
- [11] Chad Russell / John Stephens, Beginning MySQL Database Design and Optimization: From Novice to Professional, 2004, [online] <https://books.google.sk/books?id=W3hkvBKjcZEC> prístupované 4.4.2017, dostupné aj v náhľade, strana 255

- [12] Margaret Rouse, definition: use case, 2007, [online] <http://searchsoftwarequality.techtarget.com/definition/use-case> pristupované 3.2.2017
- [13] Kris Wenzel, Database Normalization Explained in Simple English, 2017, [online] <https://www.essentialsql.com/get-ready-to-learn-sql-database-normalization-explained-in-simple-english/> pristupované 25.5.2017
- [14] Donald Bell, UML basics: Sequence diagram, 2004, [online] <https://www.ibm.com/developerworks/rational/library/3101.html> pristupované 10.5.2017
- [15] Oracle Corporation, Pattern Matching, 2016, [online] <https://dev.mysql.com/doc/refman/5.7/en/pattern-matching.html> pristupované 7.5.2017
- [16] WordPress Codex, Class Reference/WP User Query, 2016, [online] https://codex.wordpress.org/Class_Reference/WP_User_Query pristupované 6.5.2017
- [17] WordPress Codex, Shortcode API, 2016, [online] https://codex.wordpress.org/Shortcode_API pristupované 6.5.2017
- [18] WordPress Codex, Plugin API/Action Reference/publish post, 2016, [online] https://codex.wordpress.org/Plugin_API/Action_Reference/publish_post pristupované 6.5.2017
- [19] Google Developers, PHP Quickstart, 2016, [online] <https://developers.google.com/google-apps/calendar/quickstart/php> pristupované 3.3.2017

9. Prílohy

1. Digitálna príloha – CD obsahujúce vytvorené WordPress rozšírenia a príručku na administráciu