

Webová aplikace pro správu zákazníků a produktů

Bc. Anh Vu Pham

Diplomová práce
2023



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2022/2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Anh Vu Pham**
Osobní číslo: **A20690**
Studijní program: **N0613A140022 Informační technologie**
Specializace: **Softwarové inženýrství**
Forma studia: **Prezenční**
Téma práce: **Webová aplikace pro správu zákazníků a produktů**
Téma práce anglicky: **Web Application for Customer and Product Management**

Zásady pro vypracování

1. Proveďte rešerši existujících řešení.
2. V rámci teoretické části vypracujte stručný rozbor technologií, které budou použity k návrhu aplikace.
3. Proveďte rozbor a analýzu požadavků.
4. V rámci praktické části realizujte a otestujte navrženou aplikaci.
5. Věnujte pozornost zabezpečení aplikace.

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. NEUSTADT, Ila; ARLOW, Jim. UML 2 a unifikovaný proces vývoje aplikací. Computer Press, Albatros Media as, 2016.
2. JOHNSON, Glenn. Programming in HTML5 with JavaScript and CSS3: training guide. Redmond, Wash.: Microsoft, 2013. ISBN 978-0735674387.
3. UNHELKAR, Bhuvan. Software engineering with uml. Auerbach Publications, 2017.
4. Official CodeIgniter documentation [online], 2021. British Columbia Institute of Technology: CodeIgniter Foundation [cit. 2021-11-18]. Dostupné z: https://codeigniter.com/user_guide/intro/index.html
5. JQuery API Documentation [online], 2006. New York BarCamp: John Resig [cit. 2022-11-18]. Dostupné z: <https://api.jquery.com/>
6. JavaScript Guide [online], 1995. Netscape Communications Corporation, Mozilla Foundation: Brendan Eich [cit. 2022-11-18]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>

Vedoucí diplomové práce: **doc. Ing. Jiří Vojtěšek, Ph.D.**
Ústav řízení procesů

Datum zadání diplomové práce: **2. prosince 2022**
Termín odevzdání diplomové práce: **26. května 2023**



doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan

prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 7. prosince 2022

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 21.05.2023

Bc. Anh Vu Pham, v. r.
podpis studenta

ABSTRAKT

Tato diplomová práce je zaměřena na vytvoření webové aplikace pro správu zákazníků a produktů. V teoretické části je provedena rešerše existujících řešení softwarových platform, systémů pro správu obsahu a rozbor použitých technologií pro realizaci této aplikace. V praktické části je provedena analýza a rozbor požadavků nového systému, kde jsou vytvořeny funkční a nefunkční požadavky, model tříd, případy užití, fyzický datový model, sekvenční diagramy a drátěné modely. Další složkou praktické části je samotná realizace webové aplikace, kde jsou popsány základní a zajímavé funkce v tomto systému včetně testovacích případů a způsobů vylepšení této aplikace.

Klíčová slova: web, webová aplikace, webová stránka, zákazníci, klienti, správa klientů, správa produktů, CMS, PHP, PHP framework, CodeIgniter, UML, Javascript, jQuery

ABSTRACT

The goal of this Master's thesis is to develop a web application for customer and product management. A research of existing solutions of software platforms, content management systems, and analysis of the technologies used for the implementation of this application is carried out in the theoretical part. The practical section includes analysis of the new system, which includes the creation of functional and non-functional requirements, a class model, use cases, a physical data model, sequence diagrams, and wireframes. The actual implementation of the web application, which describes basic and interesting features of this system, including test cases and approaches to improving the application, is another aspect of the practical part.

Keywords: web, web application, website, customers, clients, products, client management, product management, CMS, PHP, PHP framework, CodeIgniter, UML, Javascript, jQuery

Touto cestou bych rád poděkoval svému vedoucímu doc. Ing. Jiřímu Vojtěškovi, Ph.D. za kvalitní vedení, velmi cenné rady, a hlavně za ochotu a čas strávený na konzultacích při realizaci této diplomové práce.

Dále bych chtěl poděkovat své rodině, přátelům a blízkým za neskutečnou podporu při studiu.

OBSAH

OBSAH	7
ÚVOD	11
ÚVOD	11
I. TEORETICKÁ ČÁST	13
1 REŠERŠE	14
1.1 CÍL REŠERŠE	14
1.2 ASANA	15
1.2.1 REGISTRACE	15
1.2.2 ZÁKLADNÍ FUNKCE A VZHLED	16
1.2.3 PŘIDÁNÍ A EDITACE KLIENTA	16
1.2.4 PŘIDÁNÍ A EDITACE PRODUKTU	16
1.2.5 PŘIDÁNÍ NOVÉHO UŽIVATELE	18
1.2.6 NASTAVENÍ ZÁKLADNÍHO VZHLEDU	18
1.2.7 APLIKACE ZKOUMANÝCH FUNKCÍ	19
1.2.8 NABÍDKA PŘEDPLATNÝCH	20
1.2.9 ZAJÍMAVÉ FUNKCE	22
1.2.10 CELKOVÝ DOJEM	23
1.3 FREELO	24
1.3.1 REGISTRACE	24
1.3.2 ZÁKLADNÍ FUNKCE A VZHLED	25
1.3.3 PŘIDÁNÍ A EDITACE KLIENTA	26
1.3.4 PŘIDÁNÍ A EDITACE PRODUKTU	27
1.3.5 PŘIDÁNÍ NOVÉHO UŽIVATELE	28
1.3.6 APLIKACE ZKOUMANÝCH FUNKCÍ	29
1.3.7 NABÍDKA PŘEDPLATNÝCH	30
1.3.8 ZAJÍMAVÉ FUNKCE	31
1.3.9 CELKOVÝ DOJEM	34
1.4 WORDPRESS	34
1.4.1 NASTAVENÍ DATABÁZE	35
1.4.2 VZHLED SYSTÉMU	35
1.4.3 VÝPIS KLIENTŮ A PRODUKTŮ	36
1.4.4 PŘIDÁNÍ A EDITACE KLIENTŮ A PRODUKTŮ	37
1.4.5 PŘIDÁNÍ NOVÉHO UŽIVATELE	40
1.4.6 PLUGINY	40
1.4.7 APLIKACE ZKOUMANÝCH FUNKCÍ	42
1.4.8 CELKOVÝ DOJEM	43
1.5 JOOMLA!	43

1.5.1	NASTAVENÍ DATABÁZE	43
1.5.2	VZHLED SYSTÉMU	44
1.5.3	PŘIDÁNÍ A EDITACE KLIENTA A PRODUKTŮ.....	45
1.5.4	PŘIDÁNÍ NOVÉHO UŽIVATELE	48
1.5.5	PLUGINY.....	49
1.5.6	APLIKACE ZKOUMANÝCH FUNKCÍ	50
1.5.7	CELKOVÝ DOJEM.....	51
1.6	CELKOVÉ SROVNÁNÍ.....	52
2	ROZBOR POUŽITÝCH TECHNOLOGIÍ.....	55
2.1	PHP	55
2.2	DATABÁZE MYSQL	55
2.3	METRONIC FRAMEWORK.....	56
2.4	BOOTSTRAP	56
2.5	JAVASCRIPT A KNIHOVNA JQUERY	56
2.6	UML.....	56
2.7	HTML.....	57
2.8	CSS.....	57
2.9	PHP FRAMEWORKY	57
2.9.1	LARAVEL.....	58
2.9.2	SYMFONY	58
2.9.3	CODEIGNITER.....	59
II.	PRAKTICKÁ ČÁST	60
3	ANALÝZA PROJEKTU	61
3.1	POŽADAVKY	61
3.1.1	FUNKČNÍ POŽADAVKY	61
3.1.2	NEFUNKČNÍ POŽADAVKY	64
3.1.3	REALIZACE POŽADAVKŮ	65
3.2	MODEL TŘÍD.....	67
3.3	FYZICKÝ DATOVÝ MODEL	68
3.4	AKTÉŘI	69
3.5	PŘÍPADY UŽITÍ.....	69
3.5.1	SPRÁVA KLIENTŮ	70
3.5.2	SPRÁVA PRODUKTŮ	71
3.5.3	SPRÁVA UŽIVATELŮ	72
3.5.4	SCÉNÁŘE PŘÍPADU UŽITÍ.....	72
3.6	SEKVENČNÍ DIAGRAMY	82

3.6.1	PŘIHLÁŠENÍ UŽIVATELE.....	82
3.6.2	VYTVOŘENÍ NOVÉHO KLIENTA	83
3.6.3	VYHLEDÁVÁNÍ FYZICKÝCH A PRÁVNICKÝCH OSOB	84
3.6.4	EDITACE PRODUKTU	85
3.6.5	VYTVOŘENÍ NOVÉHO UPOZORNĚNÍ.....	86
3.6.6	ZMĚNA PŘÍSTUPOVÉHO HESLA.....	86
3.7	DRÁTĚNÉ MODEL Y (WIREFRAMES).....	87
3.7.1	ÚVODNÍ STRANA	87
3.7.2	VÝPIS PRODUKTŮ	88
3.7.3	NASTAVENÍ UŽIVATELSKÉHO ÚČTU.....	88
3.7.4	DETAIL KLIENTA	88
3.7.5	DETAIL PRODUKTU	88
4	REALIZACE APLIKACE.....	89
4.1	PŘEDSTAVENÍ APLIKACE	89
4.2	VYTVOŘENÍ A EDITACE KLIENTA	90
4.3	ARES	91
4.4	VYTVOŘENÍ A EDITACE PRODUKTU.....	95
4.5	SYSTÉM UPOZORNĚNÍ	96
4.6	AUTOMATICKÉ ODHLÁŠENÍ UŽIVATELE	98
4.7	IONAUTH.....	99
4.8	CRON	99
4.9	MAPY	100
4.9.1	API MAPY.CZ.....	100
4.9.2	GOOGLE MAPS	102
4.9.3	LEAFLET.....	103
4.10	VYTVOŘENÍ ŠTÍTKŮ A PŘÍRAZENÍ KE KATEGORI I.....	104
4.11	NAHRÁVÁNÍ SOUBORŮ A PŘÍLOH.....	105
4.11.1	DROPZONE	106
4.11.2	LIGHTBOX2	107
4.12	CRUD OPERACE.....	107
4.13	IMPORT OKRESŮ A KRAJŮ	109
4.14	TESTOVÁNÍ APLIKACE	110

4.14.1 PŘIHLÁŠENÍ UŽIVATELE.....	111
4.14.2 VYTVOŘENÍ KLIENTA	112
4.14.3 EDITACE KLIENTA	113
4.14.4 VYTVOŘENÍ PRODUKTU.....	115
4.14.5 EDITACE PRODUKTU.....	116
4.14.6 VYTVOŘENÍ UŽIVATELE	118
4.14.7 EDITACE UŽIVATELE	119
4.14.8 VYTVOŘENÍ UPOZORNĚNÍ.....	120
4.14.9 EDITACE UPOZORNĚNÍ.....	122
4.14.10 VYHLEDÁVÁNÍ FYZICKÝCH A PRÁVNICKÝCH OSOB	123
4.14.11 ZAZNAMENÁNÍ POLOHY PRODUKTU	125
4.14.12 ZMĚNA STAVU PRODUKTU.....	127
4.15 BEZPEČNOST	128
4.15.1 AUTENTIZACE	129
4.15.2 PŘENOSOVÉ PROTOKOLY HTTP A HTTPS.....	129
4.15.3 CERTIFIKÁTY SSL A TLS.....	130
4.15.4 SQL INJECTION	130
4.15.5 CROSS-SITE REQUEST FORGERY (CSRF).....	132
4.15.6 CROSS SITE SCRIPTING.....	133
5 NÁVRHY PRO VYLEPŠENÍ APLIKACE	136
5.1 IMPORT MODULŮ	136
5.2 GENEROVÁNÍ FAKTURY PRODUKTU.....	136
5.3 MOŽNOST DEFINOVAT TYPY SOUBORŮ	137
ZÁVĚR	138
SEZNAM POUŽITÉ LITERATURY.....	140
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	151
SEZNAM OBRÁZKŮ	152
SEZNAM TABULEK.....	155
SEZNAM ZDROJOVÝCH KÓDŮ.....	156
SEZNAM PŘÍLOH.....	157

ÚVOD

V dnešní době existuje spousta řešení a možností pro každé odvětví. Spousta lidí, kteří nejsou v kontaktu nebo nepracují s informačními technologiemi, neví, jakým způsobem se vytváří webová stránka, kde je webová stránka uchována a jakým způsobem funguje. Technologie jde každým dnem kupředu a počet možností a řešení vzrůstá. Základem každé firmy je mít vlastní systém pro správu firmy, prezentaci firmy včetně popisu práce nebo zobrazení kontaktních údajů a referencí, správu některých produktů a v návaznosti jejich prodej pomocí internetových obchodů (e-shopů), vlastního interního systému pro správu firmy, zaměstnanců, objednávek apod. Výpočetní technika jde každým dnem dopředu, a ne každá firma jde s dobou. Stále existují firmy, které řídí firmu zastaralým způsobem v papírové formě, resp. všechny smlouvy, faktury, zakázky atd. řeší papírovým způsobem. Vyhledávání a následná orientace v hromadě papírů je nepřehledná a každou objednávkou náročnější. Otázkou je, proč firmy nevyužívají některé softwary pro správu firmy, řízení projektů, objednávek a produktů? Ne nadarmo se říká, že je dobré mít vše „černé na bílém“, ale v dnešní době existují digitální podpisy, datové schránky, bankovní identity, přičemž mají stejnou váhu jako podpis na papíře.

Tato diplomová práce se zabývá realizací jednoho interního systému v podobě webové aplikace pro správu zákazníků a produktů, kde produkty tohoto systému jsou úřední desky, platební moduly, software pro úřední desky, webové stránky, rezervační moduly atd. Cílem těchto produktů je usnadnit práci úředníkům, a hlavně časově umožnit obyvatelům si kdykoliv vyřídit některé úkony prostřednictvím výše zmíněných produktů.

Teoretická část obsahuje rešerši, která se zabývá softwarovými platformami Asana, FreeLo a také open source CMS redakčními systémy Wordpress a Joomla. Na začátku rešerše jsou definovány cíle rešerše v podobě definování funkcionalit, které jsou potřebné pro tento systém a následně jsou tyto technologie vyhodnoceny, zda dokážou nebo nabízejí funkcionality definované v cílech rešerše. Po rozboru jednotlivých technologií jsou tyto technologie srovnávány mezi sebou i s vlastním řešením v podobě webové aplikace. Další částí teoretické části je vypracovaný rozbor použitých technologií.

Praktická část se věnuje realizaci celé aplikace od samotné analýzy a návrhu nového systému, která obsahuje některé nejčastější metody v procesu návrhu softwaru počínaje analýzou a rozбором funkčních a nefunkčních požadavků a jejich realizace pomocí případu užití,

dále model tříd, fyzický datový model, sekvenční diagramy až po drátěné modely pro vizualizaci grafického návrhu systému. Další částí praktické části je samotná realizace webové aplikace pro správu zákazníků a produktů, ve které jsou představeny některé základní a zajímavé funkcionality systému společně se způsobem řešení a ukázkou. Následně jsou provedeny testovací případy a představení některých bezpečnostních hrozeb a způsoby obrany proti těmto hrozbám. Na závěr praktické části jsou představeny funkcionality, resp. návrhy na vylepšení aplikace, které by mohly být v budoucnu implementovány.

I. TEORETICKÁ ČÁST

1 REŠERŠE

Tématem této diplomové práce je „Webová aplikace pro správu zákazníků a produktů“ a cílem této práce je najít nejvhodnější řešení pro správu zákazníků a produktů. Hlavními produkty tohoto systému by měly být fyzické úřední desky společně s webovými stránkami, platební moduly atd. V dnešní době se minimalizuje využívání tištěné verze i vzhledem k životnímu prostředí. Častěji můžeme vidět digitalizaci, modernizaci a automatizaci všeho. Tato část se bude věnovat rešerši existujících řešení, jako například SaaS, tj. Software as service, aplikace Asana a Freeloo. Tyto zmíněné aplikace jsou softwarové platformy a online aplikace, které se zaměřují na řízení projektů, práci v týmech a řešením úkolů. Je zde možné i kooperovat s ostatními uživateli. Současně je cílem těchto aplikací řídit, organizovat a zlepšovat efektivitu práce.

Kromě SaaS aplikací, jak bylo zmíněno výše, tu jsou i řešení a možnosti mít vlastní systém pro správu obsahu, resp. CMS systémy, tj. Content Management systém, jako např. Wordpress a Joomla. Tyto systémy mají předem definované šablony, které mohou být jak zpoplatněné, tak i bezplatné. Každopádně není nutností využívat těchto předdefinovaných šablon, ale je možné si vytvořit i vlastní šablonu a vzhled systému, který nám bude vyhovovat. Kromě toho je možné i využívat pluginy a rozšíření, které mohou zefektivnit a vylepšit náš systém. Jedná se ale o pluginy, které vytvořil někdo jiný a mohou být zpoplatněny, a ne vždy budou stoprocentně přizpůsobené na náš systém a jakékoliv změny a úpravy mohou být složitější nebo časově náročnější než vlastní vývoj. Otázkou časové náročnosti u úpravy funkcí, které vytvořil někdo jiný taktéž je, jestli časová náročnost je ekvivalentní k vlastnímu vývoji. Minimální rozdíly by měly vést k vývoji vlastní funkce právě z pohledu možných budoucích změn. Taktéž je potřeba být obezřetný při úpravě vzhledem k licenčním smlouvám. Součástí rešerše bude také konečné srovnání všech aplikací a systémů, cenové srovnání a taktéž celková časová náročnost vývoje systému. Některé aplikace nebo pluginy mohou být zpoplatněny, převod je proveden podle České národní banky [6], kdy 1 € se rovná 23,275 Kč.

1.1 Cíl rešerše

Cílem rešerše je najít nejvhodnější řešení pro správu zákazníků a produktů. Produktem se rozumí úřední desky, platební moduly a webové stránky pro obce. V rámci rešerše budou představeny výše zmíněné technologie a u každé technologie bude zkoumán způsob registrace do softwarové platformy, implementace základního balíčku, základní funkce a vzhled aplikace. Kromě toho budou představeny i varianty předplatných a placených funkcí, způsob

přidávání a editace zákazníků a produktů, možnost kooperace s dalšími uživateli, zajímavé funkce a na závěr bude shrnut celkový dojem a komfort při užívání aplikace.

Mezi zkoumanými funkcemi budou patřit:

- Možnost implementace mapy pro zobrazení pozice produktu
- Možnost uchovávat data pro:
 - o Podpis smlouvy
 - o Datum dodání
 - o Datum konce záruční doby pro HW
 - o Datum konce záruční doby pro SW
- Možnost přidávat nové uživatele pro správu systému
- Možnost aktualizovat stav produktu
- Možnost kategorizace produktu
- Možnost upravovat, přiřazovat a definovat posloupnost stavů produktu ke kategoriím
- Hledání klientů podle ARES [7], tj. Administrativní registr ekonomických subjektů
- Logování změn

1.2 Asana

Asana [8] je softwarová platforma pro řešení pracovních postupů, řízení projektů nebo organizace v týmu. Jedná se o nástroj, který dokáže plánovat, organizovat a řídit projekty od začátku do konce. V této platformě je možné koordinovat týmové úkoly a každý v týmu má přehled o tom, jaké úkoly jsou aktuálně nebo budou v budoucnu ve vývoji.

1.2.1 Registrace

Prvním krokem je registrace do této softwarové platformy. Na začátku registrace je vyžadována e-mailová adresa, pod kterou bude veden vytvořený uživatelský účet. Pro vytvoření účtu je možné i propojit účet od společnosti Google. Po zadání e-mailové adresy, je potřeba ověřit e-mailovou adresu pomocí odkazu, který je odeslán na zadanou e-mailovou adresu. Po potvrzení e-mailové adresy je potřeba zadat jméno a přístupové heslo. Po potvrzení e-mailové adresy je aktivována zkušební verze na 14 dní. Tuto platformu je možné využívat prostřednictvím webové a mobilní aplikace. Po zvolení přístupového hesla je možnost zvolit si primární roli, jako např. studenti, manažeři, členové týmu apod. Následně výběr a specifikace odvětví vývoje, např. informační technologie, marketing, design, výzkum a vývoj atd. Následuje volba hlavního cíle využívání aplikace, např. projektový a procesní management,

řešení osobních úkolů, osobní portfolio a řízení pracovní zátěže. Účelem těchto výběrů je, aby Asana mohla nabídnout nejvhodnější předplatné pro využívání systému. V posledních krocích je možnost vytvořit si první projekt nebo pracovní prostředí, nastavení prvních úkolů, seznamů, vylepšení a způsob zobrazení pracovního prostředí nebo si nastavit nějaké předdefinované sloupce. Na závěr je možné pozvat další uživatele ke spolupráci a sdílení tohoto pracovního prostředí.

1.2.2 Základní funkce a vzhled

V této aplikaci bylo vytvořeno pracovní prostředí s názvem „Správa zákazníků a produktů“. Vzhled aplikace je zobrazen v příloze:

PŘÍLOHA PII: VZHLED APLIKACE ASANA

Vzhled aplikace je zobrazen jako seznam, mimo jiné je možné zobrazit pracovní prostředí tabulkovým vzhledem. Jednotlivé úkoly reprezentují zákazníky, dále jako klienty, společně s nimi jsou uchovány základní údaje, např. identifikační číslo osoby (IČO), daňové identifikační číslo (DIČ) a kontaktní údaje. Fakturační údaje, přílohy apod. mohou být přidány do detailu klienta. Klienta je možné přiřadit i k jinému uživateli, který má přístup k tomuto systému, aby bylo na první pohled jasné, kdo přivedl klienta nebo sjednal novou smlouvu. Pracovní prostředí je pro příklad rozděleno do čtyř částí – poslaná nabídka, čekání na objednávku, složeno ve skladě a nainstalováno. Každá část reprezentuje aktuální vývoj a stav produktu od poslané nabídky po nainstalování desky.

1.2.3 Přidání a editace klienta

Jak již bylo zmíněno výše, klienta v této aplikaci reprezentuje úkol. Vytvořením nového úkolu je vytvořený nový klient. V tomto systému je v podstatě možné v jakékoliv části vytvořit nového klienta a je tedy důležité, aby byl klient založen správně, aby byl správně zálohovaný postup a vývoj předání produktu klientovi. Veškeré editace a změny přímo u klienta jsou zálohovány a je tedy možné zpětně dohledat veškeré úpravy, změny a veškerý progres u klienta nebo produktu.

1.2.4 Přidání a editace produktu

Přidání produktu je v tomto systému poměrně jednoduché, ale při větším množství bude přehled zmatečný. Jelikož se bude jednat o interní systém a nebude veřejně dostupný, tak je nutné k systému přistupovat z jiného pohledu. Produkty klienta budou reprezentovat dílčí

úkoly, dále jako produkty klienta. K jednomu klientovi je možné vytvořit neomezený počet produktů. První nevýhoda tohoto systému přichází v momentě, kdy budeme chtít přesunout produkt do dalšího stavu. V případě, že klient bude mít více produktů, tak není možné přesunout produkt do dalšího stavu a je tedy nutné vytvořit duplicitně dalšího klienta, oddělit produkty a následně přesunout do dalšího stavu. Systém tedy bude mít duplicitní data. Správa produktů je zobrazena v příloze, kde jsou produkty řešeny jako dílčí úkoly:

PŘÍLOHA PIII: SPRÁVA PRODUKTŮ V APLIKACI ASANA

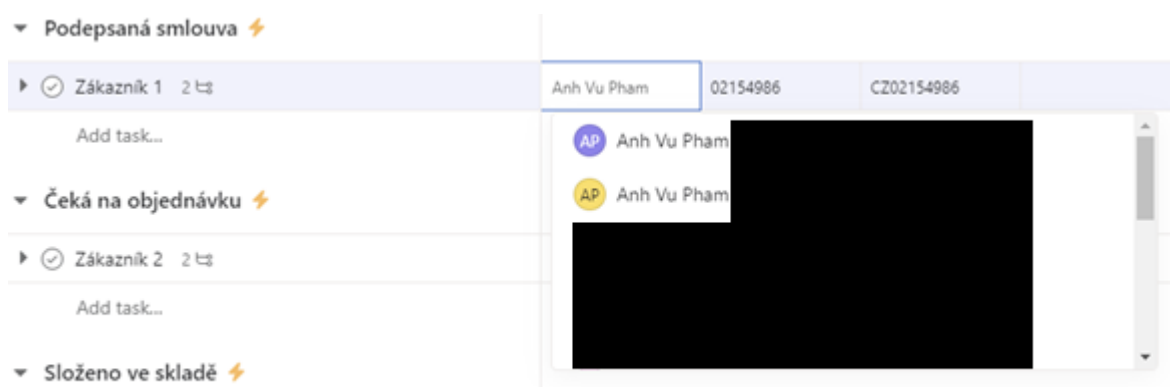
V této aplikaci je možné ukládat výdaje a výnosy, důležitá data a nastavit mezní termíny a prioritu. Detail celého produktu je znázorněn na Obr. 1. Jsou zde zobrazeny veškeré předem definované sloupce, popis produktu a také záznamy změn, které byly provedeny.

The screenshot shows the Asana task detail view for a task titled "Zákazník 7". At the top, there is a "Mark complete" button and a toolbar with icons for sharing, pinning, commenting, and deleting. The task is assigned to "Anh Vu Pham" (AP) and is due on "Thursday". It is part of the project "Správa zákazníků a produktů" (Customer and Product Management) and has a "Medium" priority. The task includes several custom fields: "# Výnos" (Revenue) and "# Náklady" (Costs) are both empty; "A ICO" is 02154965; "A DIC" is CZ02154965; "A Telefon" is +420777666555; "A E-mail" is zakaznik7@email.cz; "Datum podpisu s..." (Signature date) is 10 Apr; and "Datum instalace" (Installation date) is empty. There is a "Description" field with the placeholder text "What is this task about?". Below the task details is a "History" section showing that Anh Vu Pham created the task 4 minutes ago, assigned it to the user, changed the due date to Thursday, and moved it from "Poslaná nabídka" to "Podepsaná smlouva" in the "Správa zákazníků a produktů" project 1 minute ago. At the bottom, there is a comment input field and a "Collaborators" section showing the user and a plus sign to add more.

Obr. 1: Detail produktu – Asana

1.2.5 Přidání nového uživatele

Kooperace a spolupráce je v tomto systému poměrně jednoduchá. V rámci pracovního prostředí je možné přizvat jakéhokoliv uživatele ze stejné firmy pod stejnou organizací, kteří využívají tuto platformu. Při pozvání a využití univerzitního e-mailu jsou nabízeny ostatní studenti využívající tuto aplikaci. Kromě toho je možné sdílet toto pracovní prostředí prostřednictvím pozvánky, která je odeslaná na e-mailovou adresu, a dokonce je možnost přizvat uživatele i pomocí vygenerovaného odkazu na pracovní prostředí. Obr. 2 znázorňuje způsob přidání nových uživatelů.



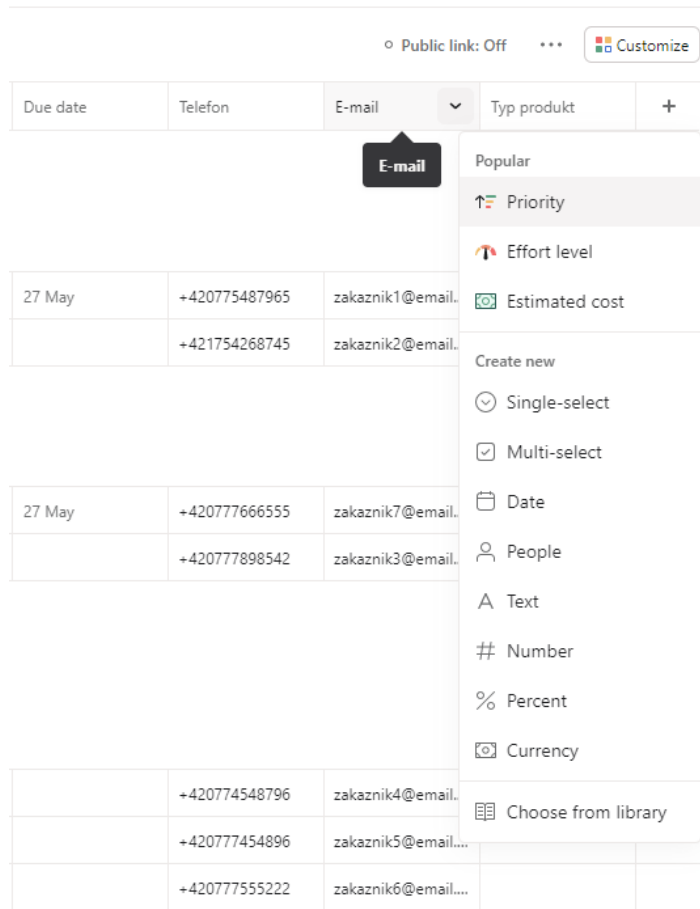
Obr. 2: Přidání uživatele – Asana

1.2.6 Nastavení základního vzhledu

Na následujícím Obr. 3 můžeme vidět přidání nového sloupce při výpisu všech klientů a produktů. Jedná se o přehled, který je znázorněn v příloze:

PŘÍLOHA PIII: SPRÁVA PRODUKTŮ V APLIKACI ASANA

Jak naznačuje následující Obr. 3, je u posledního sloupce možnost přidat nový sloupec, který se aplikuje pro všechny klienty a produkty v této aplikaci. Mezi populární sloupce jsou uvedeny předdefinované sloupce pro určení priority v podobě štítku, následně cena a úsilí. Sloupce jsou v různých formátech jako např. text, číslo, výběr z možností, procenta, měny atd. Případně je možné si přidat z vlastní knihovny v rámci organizace.



Obr. 3: Přidání nového sloupce – Asana

1.2.7 Aplikace zkoumaných funkcí

V kapitole 1.1 Cíl rešerše byly představeny cíle rešerše a zkoumané funkce pro jednotlivé systémy. Tab. 1 popisuje, které zkoumané funkce jsou splněny v aplikaci Asana.

Tab. 1: Výsledky zkoumaných funkcí aplikace – Asana

Aplikace zkoumaných funkcí	
Možnost implementace mapy pro zobrazení pozice produktu	Není možné, neexistuje plugin ani rozšíření a nelze přidat.
Možnost uchovávat data pro podpis smlouvy	Ano
Možnost uchovávat data pro datum dodání	Ano

Možnost uchovávat datum konce záruční doby HW	Ano
Možnost uchovávat datum konce záruční doby SW	Ano
Možnost přidávat nové uživatele pro správu systému	Ano
Možnost aktualizovat stav produktu	Ano
Možnost kategorizace produktu	Ne
Možnost upravovat, přiřazovat a definovat posloupnost stavů produktu ke kategoriím	Ne
Využití databáze ARES	Ne
Logování změn	Ano

Podle Tab. 1 můžeme vidět, že softwarová platforma Asana nedokáže implementovat rozšíření nebo plugin pro interakci s mapou a ukládat polohu produktů. Nemá taktéž možnost využívat databázi ARES pro vyhledávání klientů podle IČO, není možné kategorizovat produkty a s tím i spojená možnost upravovat a definovat posloupnost stavů produktu ke kategoriím. Ostatní funkce jako uchování důležitých termínů, logování změn a aktualizace stavů produktu je v tomto systému možné. Aplikace Asana tedy nabízí využívat 7 funkcí z 11.

1.2.8 Nabídka předplatných

Asana je nabízena ve čtyř variantách odběrů. Jedná se o verze Basic [9], Premium, Business a Enterprise. U tohoto systému jsou nabízeny různé cenové nabídky pro neregistrované uživatele [9]. Jsou to předplatná typu Basic, Premium a Business. Pro registrované uživatele je nabídnuta nejvyšší verze Enterprise[10]. Na Obr. 4 můžeme vidět jednotlivá předplatná.

The image shows three pricing plans for Asana, presented in a clean, modern layout. Each plan is contained within a rounded rectangular box with a distinct border color: Basic (grey), Premium (purple), and Business (teal). The Basic plan is free, while Premium and Business are paid subscriptions. Each plan includes a 'Get Started' button and a 'purchase now' link. The Premium and Business plans also feature a list of advanced features with checkmarks.

Plan	Price	Target Audience	Key Features
Basic	€0 (Free forever)	Individuals or teams just getting started with project management.	Unlimited tasks, projects, messages, activity log, file storage (100MB per file), 15 teammates, list/board/calendar views, assignee and due dates, project overview/brief, mobile apps, 100+ integrations.
Premium	€10.99 (Per user, per month billed annually; €13.49 billed monthly)	Teams that need to create project plans with confidence.	Timeline, Workflow Builder (NEW), Unlimited Dashboards, Reporting across unlimited projects, Advanced search, Custom fields, Unlimited free guests, Forms, Rules, Start dates and times, Task Templates, Milestones, Admin Console, Private teams & projects.
Business	€24.99 (Per user, per month billed annually; €30.49 billed monthly)	Teams and companies that need to manage work across initiatives.	Everything in Premium, plus: Portfolios, Goals, Workload, Custom rules builder, Forms branching & customization, Approvals, Proofing, Lock custom fields, Advanced reporting, Time tracking, Advanced integrations with Salesforce, Adobe Creative Cloud, Tableau, Power BI.

Obr. 4: Přehled verzí předplatných – Asana

Forma **Basic** je určena pro jednotlivce a týmy, kteří začínají s vytvářením projektů. Toto předplatné je dost omezené. Není možné vytvářet vlastní sloupce, formuláře, je zde omezený počet uživatelů, vytvářet milníky a cíle, pravidla atd. Tato forma je zdarma.

Předplatné typu **Premium** vychází z formy **Basic**, nabízí pokročilé vyhledávání, reportování projektů, nastavení pravidel, nastavení vlastních sloupců a formulářů a je možné mít neomezený počet uživatelů v projektu. Dokonce je zpřístupněna i administrátorská konzole, se kterou je možné spravovat členy týmu, fakturační údaje a nastavení zabezpečení pro organizaci. Současně je možné sledovat aktivitu organizace v Asaně. Tato verze nabízí i funkci Workflow builder, která má za cíl zautomatizovat procesy koordinace v týmu. Měsíční předplatné vychází na 10,99 €/měsíc, tj. zhruba 256 Kč.

Dalším typem předplatného je **Business**. Tato forma vychází z formy **Premium** a nabízí další funkce a možnosti. Nabízí možnost monitorovat a spravovat skupiny projektů pomocí

portfolií pro zobrazení stavu důležitých projektů. Dále vytvoření cílů organizace, proces schvalování, neomezené řídicí panely, sledování času, rozšíření pro sledování a monitorování času strávených na projektech a úkolech. Měsíční předplatné vychází na 24,99 €/měsíc, tj. zhruba 582 Kč.

Nejvyšší verzí předplatného, které vychází z **Business** formy, je předplatné **Enterprise**. Toto předplatné nabízí mimo jiné i vlastní obchodní značku, podporu 24/7, blokovou integraci, zálohy napříč regiony, servisní účty a hlavně **SAML** [11] (*Security Assertion Markup Language*). Jedná se o otevřený federační standard, který umožňuje poskytovateli identity ověřovat uživatele a následně předat ověřovací token jiné aplikaci známou jako poskytovatel služeb (Service provider). Měsíční předplatné není uvedeno a je sjednáno individuálně s organizací na vyžádání.

1.2.9 Zajímavé funkce

Mezi zajímavé funkce této aplikace určitě patří Rules (tj. pravidla) a Workflow. Pravidla lze předem definovat pro každý projekt a jsou spuštěny při nějaké akci, např. změna stavu produktu. Workflow je v podstatě nadstavbou a nabízí definovat tyto pravidla v rámci projektu.

Pravidla

V aplikaci Asana je možné nastavit pravidla pro jednotlivé sekce a tento projekt je rozdělen do čtyř sekcí. Pro každou sekci můžeme nastavit různá pravidla, resp. triggery, které se nám spustí po přesunutí klienta z jedné sekce do jiné. Posloupnost a akce, které jsou spuštěny jsou zobrazeny v příloze:

PŘÍLOHA PIV: VYTVOŘENÍ PRAVIDLA V APLIKACI ASANA

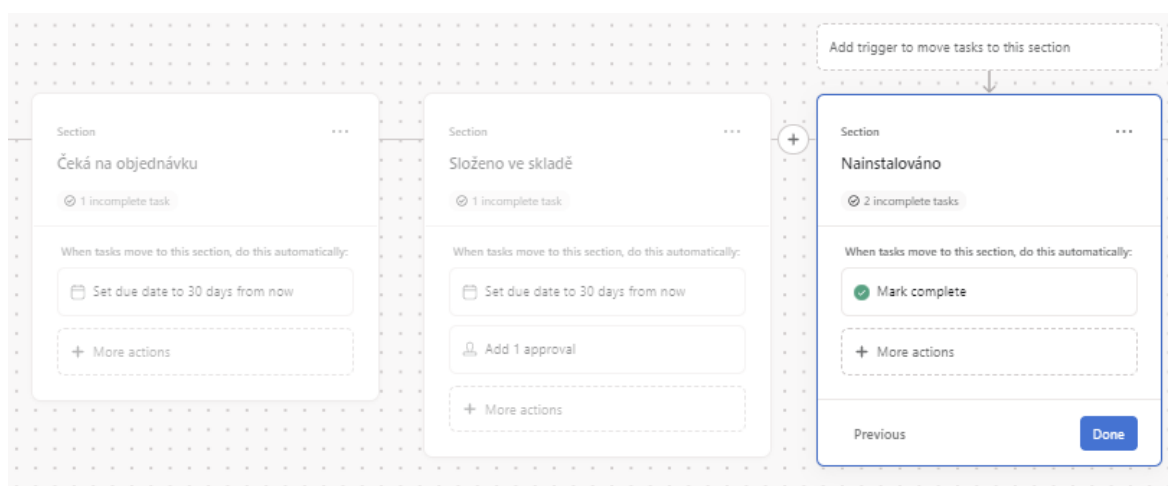
Při vytvoření pravidla je nutné zvolit o jakou sekci se jedná. V tomto případě se jedná o sekci „Nainstalováno“ a jsou zobrazeny funkce a akce, které mohou být spuštěny. Jedná se o přesunutí klienta, přesunutí nebo přidání do projektu, možnost přiřadit úkoly uživateli, nastavení mezního termínu atd. Kromě předem definovaných funkcí je možné využít i externí aplikace. Příklady externích aplikací jsou zobrazeny v příloze:

PŘÍLOHA PV: NASTAVENÍ PRAVIDLA S EXTERNÍMI APLIKACEMI

Zde je možné poslat soukromou zprávu, zprávu do vlákna v aplikaci Slack a Microsoft Teams, odeslat e-mail prostřednictvím Gmailu, vytvoření události do Google kalendáře atd.

Workflow

Jak bylo zmíněno výše v kapitole Pravidla tak tyto pravidla je možné nastavit i jiným způsobem pomocí Workflow. V této části je možné pro jednotlivé sekce nastavit pravidla, založit produkt po vytvoření klienta, nastavit si šablonu při vytváření klienta apod. Je možné mít v každé sekci více pravidel a mohou být různě kombinovány. Tato funkcionality by mohla sloužit pro nastavení mezních termínů, např. doba dodání po podpisu smlouvy, nastavení záruky po nainstalování produktu apod. Současně je možné nastavit nutný souhlas a potvrzení kompetentní osoby. Workflow v podstatě slouží pro automatizaci procesů. Příkladový proces je znázorněn na Obr. 5.



Obr. 5: Nastavení workflow – Asana

1.2.10 Celkový dojem

Aplikace na první pohled vypadá velmi přehledně, jednotlivé operace jsou velmi jednoduché a lze vše vyřešit pomocí pár kliků. Registrace je poměrně jednoduchá a výhodou je možnost kooperace v rámci organizace. Požadavky, různé pořadí a obsah sloupců lze velmi snadno měnit a je možné nastavit vlastnosti těchto sloupců. Nicméně jak bylo zmíněno v kapitole 1.2 Asana, jedná se o softwarovou platformu pro řešení pracovních postupů, řízení projektů a organizace v týmu. Tato **aplikace** svým způsobem **plní některé požadavky systému**, ale bohužel **nesplňuje všechny požadavky systému** nebo funkce zmíněné v kapitole 1.1 Cíl řešerše. Stejně tak nelze upravovat pořadí jednoho produktu na jednoho klienta. Pro každý produkt se musí vytvořit jeden klient a s ním pracovat dále. Zkrátka pokud budeme mít u jednoho klienta více produktů a každý produkt bude v jiné fázi, tak je potřeba pro každý produkt vytvořit nového klienta, přičemž bude docházet k duplicitě dat. V souvislosti s daty,

kteří je potřeba uchovávat u klienta, nelze v tomto systému rozumně ukládat fakturační údaje. Fakturační údaje mohou být uloženy ve sloupci, ale zobrazení je nepřehledné. Mohou být uchovány v popisu klienta, ale opět bude docházet k duplicitě dat. Tato aplikace splňuje zkoumané funkce v poměru 7/11 podle Tab. 1. Tato aplikace tedy nespĺňuje možnost implementace mapy pro zobrazení pozice produktu, možnost kategorizaci produktu a využití databáze ARES.

1.3 Freelo

Další aplikací v této rešerši je česká softwarová platforma Freelo [13]. Podobně jako aplikace Asana, nabízí řešení pro organizaci pracovních postupů, řízení projektů a organizace práce a v týmu. Tato softwarová platforma nabízí možnost kooperace s více uživateli, je možné přizvat a sdílet pracovní prostředí, dominuje v přehlednosti komunikaci v týmu, nabízí možnost fakturace, vytvářet vlastní šablony, integraci s dalšími aplikacemi atd. Tato softwarová platforma je alternativním řešením aplikace Asana.

1.3.1 Registrace

Registrace a začátky používání této aplikace jsou velmi podobné jako u předchozí aplikace Asana. V prvním momentě je potřeba zadat platnou e-mailovou adresu a následně je na tuto e-mailovou adresu zaslán e-mail pro potvrzení e-mailu a registrace. Na rozdíl od Asany, Freelo nenabízí možnost připojit existující účet skrze Gmail od společnosti Google. Po potvrzení e-mailové adresy je potřeba vyplnit osobní údaje jako jméno a příjmení, a následně také zvolení účelu užívání, jako např. majitelé firmy, projektoví manažeři a lídři, živnostníci nebo také pro vlastní užívání. Stejně jako u aplikace Asana je po potvrzení e-mailové adresy aktivována zkušební verze po dobu 14 dní. Freelo je možné využívat prostřednictvím webové a mobilní aplikace stejně jako u platformy Asana.

Po úspěšné registraci je zobrazeno seznamovací video [14] s platformou Freelo. V tomto videu je představena tato softwarová platforma, jsou zde zmíněny způsoby využití aplikace, charakterizace a příklady použití. Na začátku videa jsou představeny způsoby využití, jednotlivé příklady užití na existujícím projektu, způsoby komunikace, nahrávání souborů, upozornění a způsoby zobrazení pracovního prostředí, jako např. Mind mapa, tabulkové zobrazení a kalendář. V následující části jsou představeny přehledy práce a fakturace. Následně jsou představeny možnosti využití jako vlastní firemní dokumentace, rady a tipy, zápisy ze

schůzek a na závěr integraci s ostatními aplikacemi jako je Slack, Gmail, Google kalendář atd. Kromě seznamovacího videa je zobrazen i ukázkový projekt.

The screenshot displays the 'Ukázkový projekt' (Sample Project) interface in the Freeloo application. At the top, the project name is shown alongside a progress bar indicating 1:30h (0%) and a budget of 10 000 Kč, with 3 250 Kč (33%) spent and 6 750 Kč (68%) remaining. Below this, a navigation menu includes options like 'To-Do listy', 'Poznámky', 'Soubory', 'Kalendář úkolů', 'Výkazy', 'Fakturace', 'Mind mapa', and 'Timeline'. The 'To-Do listy' section is active, showing a list of tasks with checkboxes and progress indicators. The tasks include: 'Vyzkoušej si Freeloo' (checked), 'Tohle je úkol, klikni na čtvercový boxik, jako že je hotový' (checked), 'Podívej se, co všechno úkoly umí' (1 level), 'V tomhle úkolu uvidíš diskuzi se soubory' (4 levels), 'Klikni na + v horním menu a přidej svůj projekt, úkol nebo pozvi kolegu', 'Odszkroluj níže a podívej se na soubory a poznámky', and 'Přidej výkaz přes ikonku hodin na konci řádku'. Below the tasks, there is a 'Připnuté informace' section and a 'Poslední aktivity u projektu' section listing recent activities with timestamps and user names. At the bottom, there is a 'Diskuze' section.

Obr. 6: Ukázkový projekt – Freeloo

Na Obr. 6 je zobrazen výše zmíněný ukázkový projekt. V tomto ukázkovém projektu jsou představeny seznamy úkolů, časové rozpracování, naplnění rozpočtu na daný projekt, poslední aktivity, diskuse, poznámky ze schůzek, a nakonec i nahrané soubory.

1.3.2 Základní funkce a vzhled

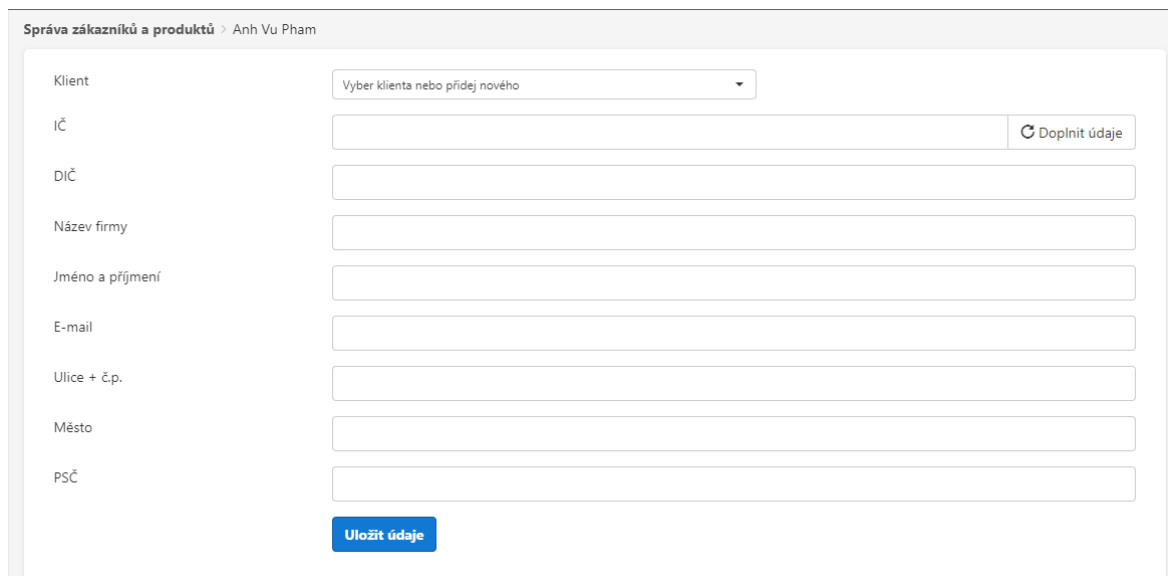
Při seznámení s touto softwarovou platformou byl představen příkladový projekt, který je zobrazen na Obr. 6. Pro systém pro správu klientů a produktů byl vytvořen nový projekt, který je zobrazen v příloze:

PŘÍLOHA PVI: VZHLED APLIKACE FREELOO

Stejně jako u Asany je tento projekt rozdělen do čtyř částí. Tyto části jsou reprezentovány jako To-Do listy. V každé části je možnost vytvořit nový produkt, resp. úkol v tomto prostředí. V softwarové platformě Asana byli klienti reprezentováni jako úkol. Ve Freeloo je možné přiřadit klienta pomocí IČO, proto není potřeba vytvářet stejnou strukturu. Stejně jako v aplikaci Asana, jsou tu k dispozici štítky k produktům, možnost spuštění sledování času vývoje a práce, je možné zadat mezní termíny, nastavovat priority úkolům, přiřadit úkoly k uživatelům atd. Na první pohled vypadá přehledněji než aplikace Asana. Součástí jsou i poslední aktivity, diskuse a poznámky k projektu. Na rozdíl od aplikace Asana zde není možné vytvářet sloupce a zobrazovat důležitá data, které potřebujeme vidět (datum podpisu smlouvy, kontaktní údaje atd.), ve zkušební verzi. Pro každý projekt je možné nastavit finanční a časový rozpočet. V přehledu projektu je zobrazeno aktuální pokrytí rozpočtu.

1.3.3 Přidání a editace klienta

Jak již bylo zmíněno v předchozí kapitole, tak v tomto systému je možné klienta přiřadit k projektu. Celkově je tento systém trochu jiný a je třeba k systému přistoupit z jiného pohledu. Projekt můžeme ve Freeloo chápat více způsoby. Projekt může reprezentovat všechny klienty a produkty v jednom. V této variantě je postup stejný jako u aplikace Asana, resp. klienta reprezentuje úkol a produkty jsou reprezentovány dílčími úkoly. Stejně jako u aplikace Asana, tak i v tomto systému bude stejná nevýhoda při přesunu více produktů jednoho klienta a při různých stavech. Druhý možný pohled projektu je takový, že projekt v podstatě definuje vše pro jednoho klienta. Jak již bylo zmíněno, tak v této softwarové platformě je možné přiřadit klienta k projektu. Postup, jak probíhá přiřazení klienta k projektu je znázorněn na Obr. 7.

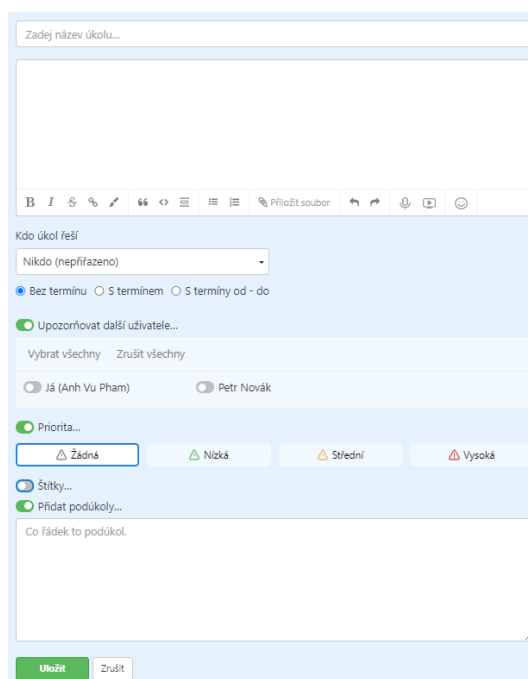


Obr. 7: Výběr nebo přidání klienta – Freelo

Na Obr. 7 je zobrazen formulář pro přiřazení klienta k projektu. Je zde nabídnuta možnost doplnit údaje pro zadání osobního identifikačního čísla (IČO), údaje jsou následně doplněny pomocí dat z databáze ARES, taktéž je možné zvolit již předem vytvořeného klienta. Přidání a editace klienta je řešena pomocí stejného formuláře na Obr. 7.

1.3.4 Přidání a editace produktu

Na následujícím Obr. 8 je zobrazen formulář pro přidání nového produktu.



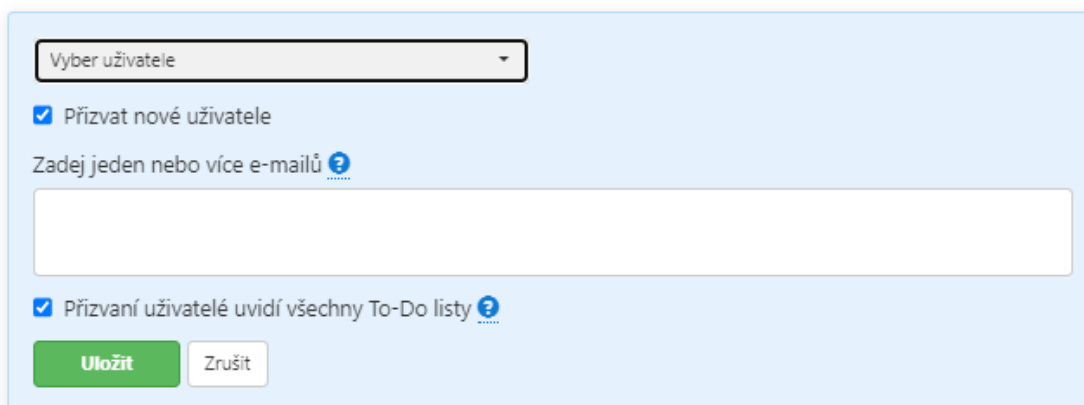
Obr. 8: Přidání nového produktu – Freelo

Přidávání produktu je taktéž poměrně jednoduché. V tomto formuláři je možné zadat název úkolu, přiřadit úkol k řešiteli (uživateli), nastavení mezního termínu, přidat příjemce upozornění při změně, nastavení priority a štítků a na závěr je možné vytvářet dílčí úkoly. Součástí je i textové pole, do kterého je možné psát detaily, poznámky k produktu a také je tímto polem zahájeno vlákno pro diskusi. Zajímavostí je, že je možné do komunikačního vlákna nahrát zvukovou zprávu, a dokonce i nahrát obrazovku.

1.3.5 Přidání nového uživatele

Taktéž i v tomto systému je kooperace a spolupráce s ostatními uživateli velmi jednoduchá. V rámci jakéhokoliv projektu je možné přizvat uživatele prostřednictvím e-mailové adresy a všech uživatelů v rámci týmu. Na Obr. 9 je zobrazen formulář pro přidání nového uživatele do systému. V rámci formuláře je možné vybrat existujícího uživatele, který spolupracuje již v jiných projektech nebo je součástí týmu a také je možné přizvat uživatele zadáním e-mailové adresy.

Uživatelé +



Vyber uživatele

Přizvat nové uživatele

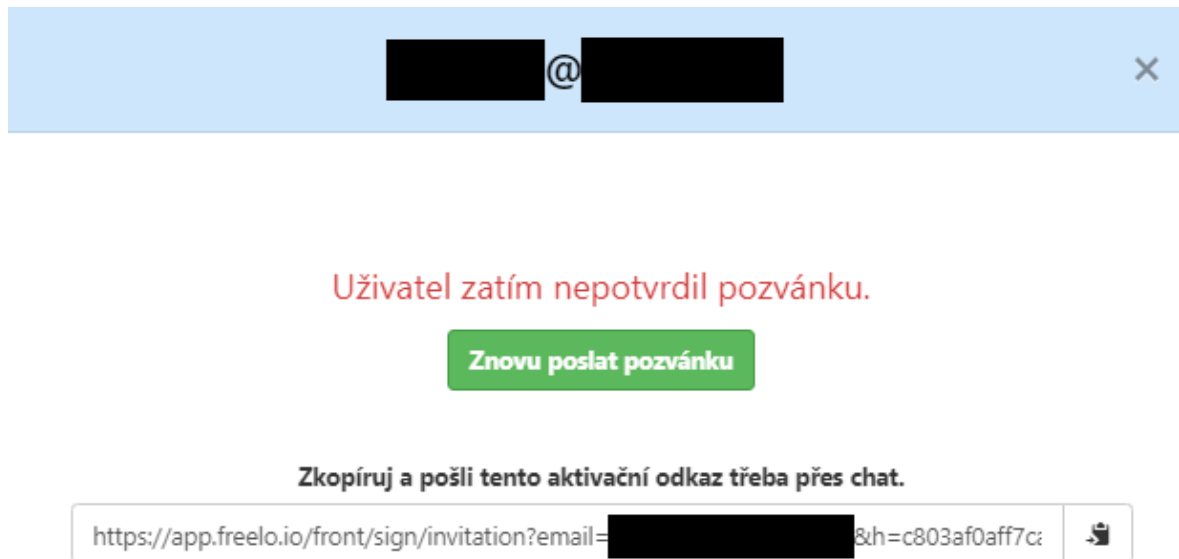
Zadej jeden nebo více e-mailů ?

Přizvání uživatelé uvidí všechny To-Do listy ?



Obr. 9: Přidání nového uživatele – Freelo

Projekty ve Freelo není možné sdílet prostřednictvím odkazu. Pozvánka ke spolupráci je možná pouze na základě zadání e-mailové adresy a až následně je možné poslat dodatečnou pozvánku pomocí odkazu, který ale nemůže využít jiný uživatel, neboť je pozvánka individuálně řešená, nikoliv globálně. Průběh je znázorněn na následujícím Obr. 10.



Obr. 10: Přidání nového uživatele odkazem – Freelo

1.3.6 Aplikace zkoumaných funkcí

V rámci zkoumaných funkcí zkoumaného systému Freelo jsou výsledky popsány v následující Tab. 2.

Tab. 2: Výsledky zkoumaných funkcí aplikace – Freelo

Aplikace zkoumaných funkcí	
Možnost implementace mapy pro zobrazení pozice produktu	Není možné, neexistuje plugin ani rozšíření a nelze přidat.
Možnost uchovávat data pro podpis smlouvy	Ne, pouze Bussiness forma
Možnost uchovávat data pro datum dodání	Ne, pouze Bussiness forma
Možnost uchovávat datum konce záruční doby HW	Ne, pouze Bussiness forma
Možnost uchovávat datum konce záruční doby SW	Ne, pouze Bussiness forma
Možnost přidávat nové uživatele pro správu systému	Ano
Možnost aktualizovat stav produktu	Ano

Možnost kategorizace produktu	Ne
Možnost upravovat, přiřazovat a definovat posloupnost stavů produktu ke kategoriím	Ne
Využití databáze ARES	Ano
Logování změn	Ano

Podle Tab. 2 můžeme vidět, že softwarová platforma Freelo také nedokáže implementovat rozšíření nebo plugin pro interakci s mapou a následné ukládání polohy produktu. Nemá možnost kategorizovat produkty a s tím i spojenou možnost upravovat a definovat posloupnost stavů produktu ke kategoriím. Na rozdíl od aplikace Asana není možné v aplikaci Freelo ve zkušební verzi uchovávat důležité termíny jako datum podpisu smlouvy, data dodání a záručních lhůt a není ověřena funkčnost ukládání těchto dat a je dostupná od Business formy předplatného. Freelo nabízí pouze možnost zadání mezního termínu. V tomto systému je ale možné využít databázi ARES pro přidání nového klienta, je k dispozici logování změn, aktualizace stavů produktu a také přidání nových uživatelů do projektu pro kooperaci s ostatními uživateli. Celkově systém Freelo nabízí možnost využívat 4 funkce z 11.

1.3.7 Nabídka předplatných

Softwarová platforma Freelo nabízí několik možností předplatných [15]. Jednotlivé předplatná jsou zobrazeny na následujícím Obr. 11.

The screenshot shows the Freelo website's pricing page. At the top, there are navigation links for 'Produkt', 'Ceník', 'Reference', 'Kariéra', and 'O nás', along with 'Přihlásit' and 'Začít zdarma' buttons. Below the navigation, there are two tabs: 'Ročně (10% sleva)' and 'Měsíčně'. The main content area displays four pricing cards:

- Freelance**: 890 Kč za neomezené uživatele měsíčně bez DPH. Pro Freelancery a jejich klienty. Features include: Různé pohledy na úkoly: Řádky, Sloupce, Tabulka, Kalendář, Mind mapa; Timetracking pro trackování času a nákladů a jednoduché fakturování; Neomezené projektů, file systém na soubory, poznámky a dokumenty a skvělá mobilní i desktop aplikace.
- Team**: 1 990 Kč za neomezené uživatele měsíčně bez DPH. Pro menší firmu. Vše, co ve Freelance, navíc. Features include: Správcové funkce pro vedoucí oddělení: Správa uživatelů na projektech, Správa a zakládání projektů pod jedním účtem, Přehledy výkazů týmu.
- Business**: 3 800 Kč za 20 uživatelů měsíčně bez DPH. NEJPRODÁVANĚJŠÍ. Pro firmu s odděleními. Vše, co v Team, navíc. Features include: Vlastní pole umožní z Freela udělat miniCRM nebo datově propojit s vaším IS/ERP; Detailní přehledy: Uživatelů, Projektů, Týmů; Rozšířené zabezpečení: Správa uživatelů v doméně, nahrazení pracovníka, vynucení 2FA; Ganttův diagram (Timeline).
- Enterprise**: od 45 900 Kč měsíčně bez DPH. Pro korporace. Vše, co v Business, navíc. Features include: SSO, SAML, SLA, Integrace na míru, automatizace, Onboarding workshop, Success manager, ...; Kontaktujte nás a domluvíme se na vašich potřebách, požadavcích a termínech implementace.

Obr. 11: Cenová nabídka předplatných – Freelo

Celkově nabízí pět variant předplatných a jedná se o varianty Free, Freelance, Team, Business a Enterprise. Ve všech verzích je možné využívat desktopovou a mobilní aplikaci. Předplatné **Freelance** nabízí různé pohledy na úkoly, resp. tabulkové zobrazení a kalendář, time tracking pro zaznamenání času při řešení úkolu, resp. produktu, možnost fakturace a vytvářet neomezený počet projektů a uživatelů, 15GB úložiště pro soubory. Měsíční předplatné vychází na 890 Kč bez DPH za měsíc.

Další formou předplatného je **Free**, které je bezplatné a na první pohled velmi omezené. Toto předplatné vychází z předplatného **Freelance** a má řadu omezení. Forma Free povoluje kooperaci pouze se třemi dalšími uživateli a mít pouze tři aktivní projekty. Nemá k dispozici Mind mapu (způsob plánování a organizace úkolu) a omezené úložiště na soubory. Je možné nahrát pouze 500 MB.

Následující verze **Team** vychází z formy **Freelance** a nabízí navíc správcovské funkce pro vedoucí jednotlivých oddělení, spravovat uživatele na projektech, přehledy výkazů týmu a 100GB úložiště pro soubory. Měsíční předplatné vychází na 1 990 Kč bez DPH za měsíc.

Čtvrtou formou předplatného je forma **Business**, která vychází z formy **Team**. Tato forma nabízí možnost vytvářet vlastní pole, detailní přehledy uživatelů, projektů a týmů. Dále nabízí rozšířené zabezpečení, živý přehled sledování času nad úkoly, šablony nastavení atd. Měsíční předplatné vychází na 3 800 Kč bez DPH za měsíc pro 20 uživatelů.

Poslední, nejvyšší verzí je verze **Enterprise**, která vychází z formy **Business**. Tato forma nabízí federační standard **SAML**, zmíněný v kapitole 1.2.8 Nabídka předplatných, integraci na míru, pokročilou automatizaci, nadstandardní zákaznickou péči atd. Měsíční předplatné začíná na 45 900 Kč bez DPH za měsíc. Toto předplatné vyžaduje vzájemnou komunikaci pro domluvení požadavků a termínech implementace.

1.3.8 Zajímavé funkce

Zajímavou funkcí je možnost vytvářet vlastní šablonu pro strukturu projektu, fakturace a na závěr velmi přehledná dokumentace.

Šablony

Při využití tohoto systému jako správu klientů a produktů jsou klienti reprezentováni celým projektem. V rámci projektu je možné předem definovat posloupnost stavů produktu. Jelikož budeme mít v systému pro správu klientů a produktů různé typy produktů, tak bude potřeba

vytvořit více druhů šablon. Nicméně je tato funkcionality velmi přijatelná a dokáže urychlit veškeré procesy a jedná se v podstatě o počáteční automatizaci. Na následujícím Obr. 12 můžeme vidět proces vytvoření šablony projektu. Pro šablonu je nutné nadefinovat název a nastavit si To-Do listy, které budeme potřebovat.

The screenshot shows a web interface for creating a project template. At the top, a purple banner contains the text: "Tento projekt je šablona. Ze šablony můžeš **Vytvořit nový projekt** nebo ji **Smazat**." Below this is the main heading "Správa zákazníků a produktů" with a "Přidat štítek" button. A sub-header reads "Kapitánem projektu jsi ty. Klient: **nastav**". A navigation bar includes "To-Do listy", "Poznámky", "Uživatelé", "Soubory", "Kalendář úkolů", and "Mind mapa". The "To-Do listy" section is expanded, showing a list of tasks with the following categories and actions:

- To-Do listy** (+ -): Nainstalováno, Info, + Přidat úkol
- Složeno ve skladě**: Info, + Přidat úkol
- Čeká na objednávku**: Info, + Přidat úkol
- Podepsaná smlouva**: Info, + Přidat úkol
- Poslaná nabídka**: Info, + Přidat úkol

Obr. 12: Vytvoření šablony projektu – Freelo

Fakturace

Na Obr. 13 je zobrazen způsob fakturace jednotlivých projektů. Fakturace může probíhat různými způsoby. Fakturovány mohou být jednotlivé To-Do listy, resp. při změně stavu našich produktů nebo celý projekt a nezávisí na tom, jestli je projekt před nebo po dokončení. Při fakturaci je potřeba pouze zvolit jaká část má být fakturována. V přehledu je vidět celková částka pro fakturaci, počet odpracovaných hodin a v jakém období je přehled zobrazen. Součástí je i možnost filtrovat podle uživatelů, týmů, projektů atd.

Reporty > Fakturace a výkazy > Fakturovat

Fakturovat Vystavené faktury Souhrny Denní Týdenní a měsíční

Fakturace - 1/2

Vyber výkazy, které chceš vyfakturovat

27 500 Kč
Suma peněz

14:00 h
Suma času

2. 4. 2023 - 2. 5. 2023
Vybrané období

Stáhnout report
CSV Tabulka

CZK

Všichni uživatelé...

Všechny projekty

Nefakturováno

Všechny štítky úkolů

Všechny úkoly

Posledních 30 dnů

Filtrovat

Uložit filtr

<input type="checkbox"/>	Univerzita Tomáše Bati ve Zlíně -	4 000 Kč	(2:00 h)
<input type="checkbox"/>	Správa zákazníků a produktů -	23 500 Kč	(12:00 h)

Celková částka k fakturaci

0,- (0:00 h)

[Pokračovat na položky faktury](#)

Obr. 13: Způsob fakturace – Freelo

Po zvolení všech položek k fakturaci je možné uložit tyto položky jako vyfakturované. V systému Freelo je možné označit faktury jako fakturované a mimo jiné nabízí možnost fakturovat i v externích aplikacích jako Fakturoid, iDoklad a Vyfakturuj, resp. aplikace pro provádění fakturací a vedení účetnictví. Pro možnost využívat tyto externí aplikace je potřeba mít vlastní API klíč pro vzájemnou komunikaci. Před uložením je zobrazeno shrnutí fakturace a je možné provést poslední změny a doplnit chybějící data.

Dokumentace

Softwarová platforma Freelo se velmi snaží poradit se vším. Při vstupu do aplikace je zobrazeno výukové video a příkladový projekt. Freelo nabízí řadu funkcí a tyto funkce jsou popsány v přehledné dokumentaci. Dokumentace je přístupná vždy a při jakékoliv operaci ve zjednodušené formě. Tato forma funguje jako vyskakovací okno při kliku na tlačítko, zobrazí nám jednotlivé operace, a dokonce je možné i slovně vyhledávat.

1.3.9 Celkový dojem

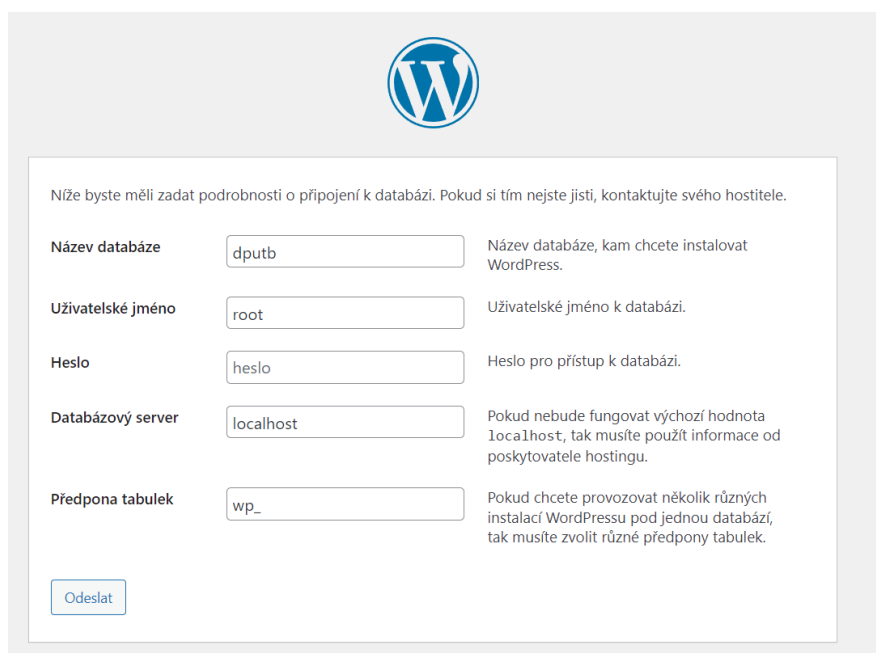
Tato softwarová platforma vypadá na první pohled velmi jednoduše a přehledně. Jednotlivé operace pro přidání nového klienta a produktu jsou velmi jednoduché. Freelo je velmi podobné aplikaci Asana. Jsou to totiž aplikace pro řešení pracovních postupů, řízení projektu a organizace v týmu. Stejně jako aplikace Asana, tak **Freelo** taktéž **nesplňuje všechny požadavky** nebo funkce **systemu** zmíněné v kapitole 1.1 Cíl rešerše. System nabízí možnost využívat databázi ARES pro vyhledávání klientů, aktualizovat stavy produktů, nastavovat rozpočet pro projekt a kooperovat s dalšími uživateli. Tato aplikace má své světlé stránky v podobě přehledné dokumentaci, způsob komunikace u jednotlivých produktů, možnost fakturace, vytváření vlastních šablon projektu atd. Nicméně nevýhodou tohoto systemu je nedostatečné logování změn u produktu. Logování není zapnuto například při změně stavu produktu, a hlavně není u jednotlivých produktů zvlášť. Přehled posledních aktivit je pro celý projekt a není přehledný. Tab. 2 ukazuje, že tato aplikace splňuje zkoumané funkce v poměru 4/11 ve zkušební verzi a v poměru 8/11 při využití **Bussiness** formy předplatného. Freelo taktéž nenabízí možnost implementace mapy pro zobrazení pozice produktu, kategorizace produktu a uchování důležitých termínů spojené s produkty.

1.4 Wordpress

V rámci rešerše byly představeny softwarové platformy Asana a Freelo. Wordpress [16] je nástroj pro tvorbu webových stránek a CMS redakčních systemů. Jedná se o jeden z nejjednodušších a nejoblíbenějších systemů pro správu obsahu. Wordpress dovoluje vytvářet vlastní šablony, rozšíření a pluginy, využívat analytické nástroje nebo si také vytvořit vlastní plně přizpůsobitelný online obchod pomocí WooCommerce. WooCommerce [17] je flexibilní softwarové řešení s otevřeným zdrojovým kódem vytvořené pro weby založené na Wordpressu a běžně se používá pro tvorbu online e-shopů. WooCommerce také umožňuje uživatelům snadno spravovat své obchody. Od nastavení způsobu zobrazení produktů a správy objednávek až po přijímání více platebních bran. Wordpress [18] je open-source software [19], tj. software se zdrojovým kódem, který může kdokoliv různě upravovat, a existuje zde velká komunita lidí, kteří spolupracují a přispívají svými funkcemi a rozšířeními. Wordpress je postavený na jazyce PHP a MySQL.

1.4.1 Nastavení databáze

Kromě samotné instalace je potřeba nastavit i připojení k databázi. Databáze *phpmyadmin* má primárně nastaveného uživatele *root* a přihlašuje se bez přístupového hesla. Před spojením s databází *phpmyadmin* je potřeba vytvořit databázi pro projekt. Následně na Obr. 14 je zobrazen způsob, jak zadat údaje pro připojení k databázi.



Niže byste měli zadat podrobnosti o připojení k databázi. Pokud si tím nejste jisti, kontaktujte svého hostitele.

Název databáze	<input type="text" value="dputb"/>	Název databáze, kam chcete instalovat WordPress.
Uživatelské jméno	<input type="text" value="root"/>	Uživatelské jméno k databázi.
Heslo	<input type="text" value="heslo"/>	Heslo pro přístup k databázi.
Databázový server	<input type="text" value="localhost"/>	Pokud nebude fungovat výchozí hodnota localhost, tak musíte použít informace od poskytovatele hostingu.
Předpona tabulek	<input type="text" value="wp_"/>	Pokud chcete provozovat několik různých instalací WordPressu pod jednou databází, tak musíte zvolit různé předpony tabulek.

Obr. 14: Nastavení připojení databáze – Wordpress

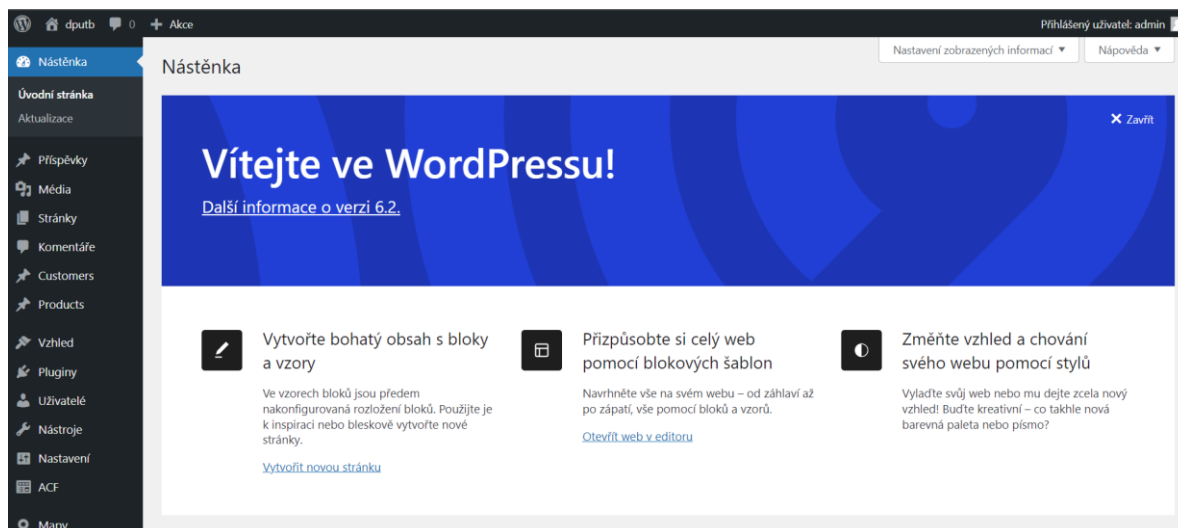
Po úspěšném nastavení databáze je potřeba nastavit název webu, vytvořit uživatele s administrátorskými právy, zadat uživatelské jméno, přístupové heslo a e-mailovou adresu. Pro heslo je možné buď využít vygenerované přímo Wordpressem, nebo si zvolit vlastní.

1.4.2 Vzhled systému

Na následujícím Obr. 15 můžeme vidět úvodní stránku po přihlášení do systému Wordpress. Přihlášení do administrace systému je možné skrz odkaz:

{navez_domeny}/navezprojektu/wp-admin

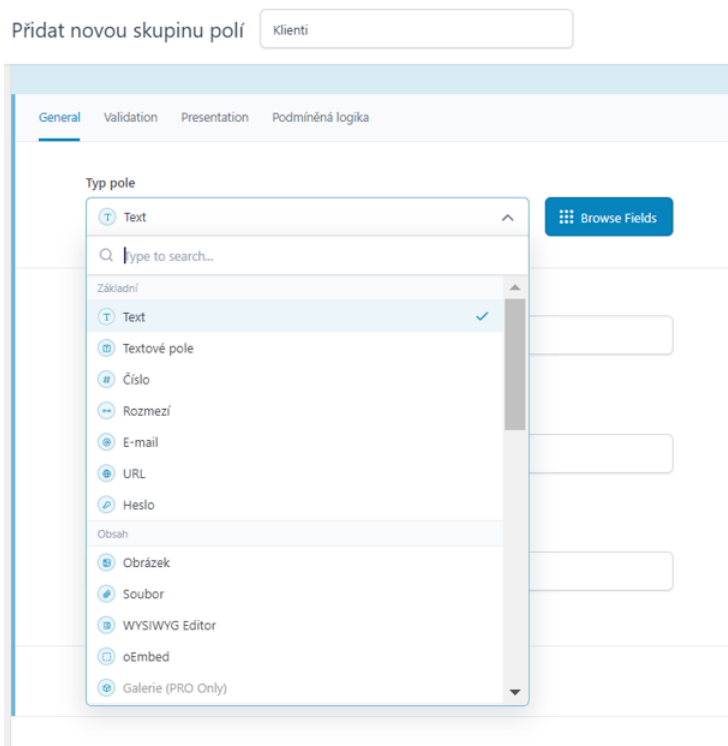
Úvodní stránka obsahuje v levé části seznam, ve kterém jsou obsaženy odkazy na nastavení vzhledu a stránek. Současně je možné měnit nastavení projektu, přidávat a editovat uživatele v systému, přidávat pluginy atd.



Obr. 15: Vzhled administrace – Wordpress

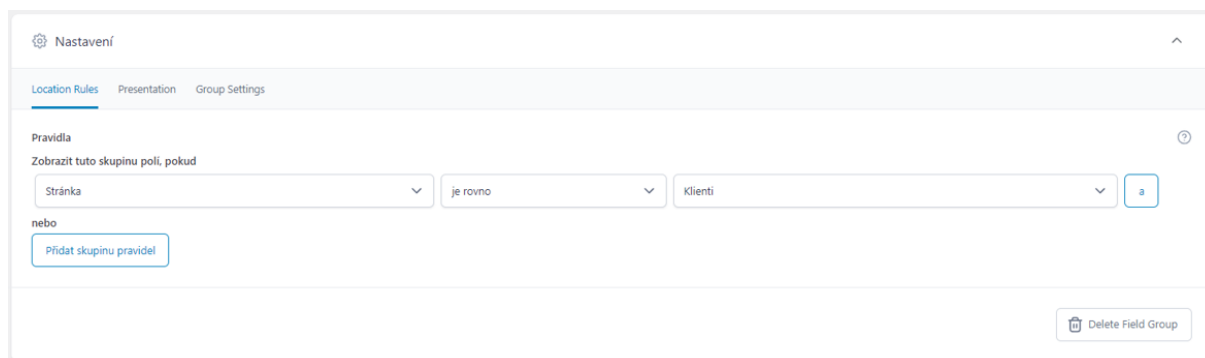
1.4.3 Výpis klientů a produktů

Celkově způsob výpisu a editace klientů a produktů bude v tomto systému zcela jiné na rozdíl od aplikací jako Asana a Freeloo. V systému Wordpress je možné využít plugin ACF, tj. Advanced Custom Fields [20]. Jedná se o rozšíření, které umožňuje vytvářet a zobrazovat vlastní pole. Tyto pole je možné předem definovat, resp. nastavit o jaký typ pole se jedná, v jakém bude formátu atd. Více na Obr. 16.



Obr. 16: Plugin ACF – typy polí

Pole jsou definována v rámci skupiny a těmto skupinám je možné nastavit, kdy se mají zobrazit, např. pouze pro určitý typ stránky, pro stránky vytvořené z určité šablony atd. Na těchto stránkách budou poté zobrazena definovaná pole. Toto nastavení je znázorněno na následujícím Obr. 17.

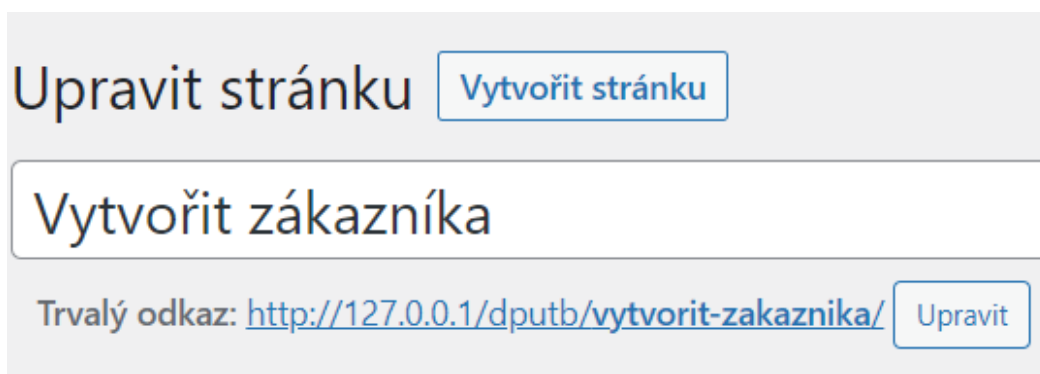


Obr. 17: Plugin ACF – Nastavení pravidla pro skupinu polí

Obsah těchto polí je potřeba implementovat pomocí jazyka PHP a pomocí SQL dotazů. Jednotlivá pole mají vlastní identifikátor a pomocí těchto identifikátorů ve skriptu zobrazíme požadovaná data, které získáme pomocí SQL dotazu.

1.4.4 Přidání a editace klientů a produktů

Přidání a editace klientů a produktů probíhá velmi podobně. Lišit se budou pouze ve struktuře textových polí formuláře. Pro každou operaci, resp. pro přidání a pro editaci je potřeba vytvořit stránku, která nám tyto operace bude obsluhovat. Vytvoření stránky a úpravy trvalého odkazu je znázorněno na následujícím Obr. 18.



Obr. 18: Vytvoření stránky pro vytvoření nového klienta

Po nastavení této stránky je dalším krokem vytvořit PHP soubor vytvořený z aktuální šablony. V tomto souboru následně vytvoříme formulář pro přidání nebo editaci klienta nebo produktu. Příkladová struktura formuláře je znázorněno na Obr. 19.

Vytvořit zákazníka

Název zákazníka

IČO

DIČ

Telefonní číslo

E-mail

Obr. 19: Formulář pro vytvoření zákazníka

Po vytvoření formuláře už stačí pouze přidat PHP kód který nám zajistí vytvoření nebo editaci klienta. Zdrojový kód 1 znázorňuje způsob vytvoření nového klienta.

```
<?php
if($_SERVER['REQUEST_METHOD'] === 'POST')
{
    $conn = db_connect();
    $data = [
        'name' => $_POST['name'],
        'ico' => $_POST['ico'],
        'dic' => $_POST['dic'],
        'phone' => $_POST['phone_number'],
        'email' => $_POST['email'],];

    $sql = "INSERT INTO client (name, ico, dic, phone, email) VALUES ";
    $sql .= "(" . "\"" . $data['name'] . "\" , " . "\"" . $data['ico'] . "\" , " . "\"" . $data['dic'] .
    "\" , " . "\"" . $data['phone'] . "\" , " . "\"" . $data['email'] . "\" )";

    if ($conn->query($sql) === TRUE)
    {
    }
    else{
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
}
else{?>
    <form method="POST">
        <div class="form-group">
            <label for="customer_name">Název klienta</label>
            <input type="text" name="name" class="form-control" id="name"
placeholder="Zadejte název klienta">
        </div>
```

```

        <div class="form-group">
            <label for="ico">IČO</label>
            <input type="number" name="ico" class="form-control" id="ico"
" placeholder="Zadejte IČO">
        </div>
        <div class="form-group">
            <label for="dic">DIČ</label>
            <input type="text" name="dic" class="form-control" id="dic"
placeholder="Zadejte DIČ">
        </div>
        <div class="form-group">
            <label for="phone_number">Telefonní číslo</label>
            <input type="text" name="phone_number" class="form-control"
id="phone_number" placeholder="Zadejte telefonní číslo">
        </div>
        <div class="form-group">
            <label for="email">E-mail</label>
            <input type="email" name="email" class="form-control"
id="email" aria-describedby="emailHelp" placeholder="Zadejte email">
        </div>

        <button type="submit" class="btn btn-primary">Vytvořit</button>
    </form>
<?php } ?>

```

Zdrojový kód 1: Ukázka formuláře pro přidání klienta

Skript je spuštěn po stisknutí tlačítka „Vytvořit“ a dochází ke zpracování zadaných dat a vytvoření spojení s databází a přípravě dat pro vložení nového řádku. Ukázka pro spojení s databází znázorňuje Zdrojový kód 2. Příložené údaje pro přístup do databáze je pouze pro vývoj a jedná se o lokální databázi. Stejně se bude chovat i produkční databáze, která bude mít jiný název serveru a jiné autentizační údaje. Příložený Zdrojový kód 1 je pouze pro ukázkou, chybí zde validace, že byly vytvořeny všechna pole, korektní formát pro IČO atd. Tento kód pouze ukládá zadaná data uživatelem a provede vložení nového řádku do databáze do tabulky *client*.

```

function db_connect()
{
    //Db init
    $servername = "localhost";
    $username = "root";
    $password = "";
    $dbname = "dputb";

    //db connection
    $conn = new mysqli($servername, $username, $password, $dbname);
    if ($conn->connect_error)
    {
        die("Connection failed: " . $conn->connect_error);
    }
    return $conn;
}

```

Zdrojový kód 2: Ukázka připojení databáze

1.4.5 Přidání nového uživatele

Způsob přidání uživatele do tohoto systému je podobně jednoduché jako u ostatních aplikací jako Asana a Freeloo. Ve Wordpressu je možné přizvat uživatele do systému pomocí formuláře zobrazený na Obr. 20. Mezi povinnými údaji je uživatelské jméno, e-mailová adresa a přístupové heslo. E-mail se zadává pro případ, že by uživatel zapomněl své přístupové heslo. Heslo je možné si nechat buď vygenerovat, nebo si zvolit vlastní. Na závěr je možné nastavit oprávnění uživatele, tj. jaké bude mít práva a kterou část systému si může zobrazit a případně editovat.

Můžete vytvořit zcela nového uživatele a registrovat ho na tomto webu.

Uživatelské jméno (vyžadováno)

E-mail (vyžadováno)

Jméno

Příjmení

Webová stránka

Jazyky

Heslo

Odeslat uživateli oznámení Odeslat novému uživateli e-mail s informacemi o jeho účtu.

Úroveň
Návštěvník
Manažer obchodu
Zákazník
Návštěvník
Spolupracovník
Redaktor
Šéfredaktor
Administrátor

Děkujeme, že používáte [WordPress](#).

Obr. 20: Přidání nového uživatele do systému – Wordpress

1.4.6 Pluginy

Wordpress využívá spousta lidí a tato obrovská komunita přispívá k vylepšení systému tvorbou nových funkcionalit pro ostatní uživatele. Největší výhodou Wordpressu je právě možnost instalovat pluginy, které stačí pouze nainstalovat a aktivovat. Část těchto pluginů většinou bývá zdarma a nadstavbové funkce bývají zpoplatněny, a navíc je obtížné, jakkoliv upravovat a měnit tyto pluginy, což je nevýhodou těchto pluginů. Zdrojové kódy, které vytvořil někdo jiný je složité číst a při snaze upravovat tento kód je občas obtížné i pochopit tento kód.

Plugin pro získání dat z databáze ARES

V systému Wordpress je dostupný plugin pro automatické přidávání IČO, DIČ a pokud se jedná o českou fakturační adresu, tak je možné automaticky přidat i název společnosti, adresu, město a PSČ z databáze ARES. Tento plugin vyžaduje minimální verzi PHP 7.1+ & PHP 8.0+. Název tohoto pluginu je **Kybernaut IČO DIČ** [21], jejím autorem je Karolína Vyskočilová. Součástí vyhledávání dat z ARES je i vyhledávání ve VIES [22], tj. systém výměny informací provozovaný evropskou komisí. Pomocí tohoto systému je možné vyhledávat přihlášené subjekty z vnitrostátních databází DPH. Inicializace aplikace **Kybernaut IČO DIČ** [23] je napsána v jazyce PHP a samotná logika aplikace využívá skriptovací jazyk Javascript s využitím knihovny jQuery. Po zadání IČO do textového pole je provedena validace vstupu od uživatele a po úspěšné validaci je spuštěno volání *AJAX* [24], tj. Asynchronous JavaScript And XML. Jedná se o použití objektu *XMLHttpRequest* pro komunikaci se serverem. Může přijímat a odesílat informace v různých formátech. Asynchronní metoda pro odesílání požadavků HTTP, která probíhá na pozadí a nevyžaduje aktualizaci stránky. Tento plugin vyžaduje mít nainstalovaný plugin **WooCommerce** pro kompletní funkčnost. Kompatibilní je s pluginy jako **WooCommerce SuperFaktura**, **WooCommerce Checkout Field Editor Pro** atd. Nevýhodou tohoto pluginu je, že jej není možné aplikovat pro systém pro správu klientů a produktů. **WooCommerce** nabízí možnost prodávat tyto produkty, což není žádané u systému pro správu klientů a produktů. Poslední aktualizace byla v roce 2022.

Plugin WP Go Maps

Pro implementaci mapového rozhraní pro zobrazení produktů je možné využít plugin **WP Go Maps** [25]. Tento plugin umožňuje vytvořit přizpůsobenou mapu, resp. mapu vytvořenou s pomocí Google Maps. Plugin je možné využívat zdarma s omezením nebo v plné verzi za poplatek. Bezplatná verze nabízí možnost vytváření responzivních map (optimalizované zobrazení stránek pro různé zařízení), přidání neomezeného počtu mapových značek, přidání animací, vyhledávání obchodů atd. Placená verze umožňuje větší počet map, přidávání popisů, odkazů a obrázků, filtrování značek podle kategorie, pokročilejší možnosti pro vyhledávání obchodů atd. Placená verze vychází na 39,99 €, tj. zhruba 931 Kč. Díky placené verzi je tento plugin pravidelně aktualizovaný. Značky je možné definovat pomocí hodnot zeměpisné délky a šířky nebo pomocí zadání vstupu do textového pole. V přílohách:

PŘÍLOHA PVII: VYTVÁŘENÍ ZNAČKY PODLE GPS

PŘÍLOHA PVIII: VYTVÁŘENÍ ZNAČKY PODLE ZEMĚPISNÉ DÉLKY A ŠÍŘKY

jsou zobrazeny způsoby zadání hodnot a způsob definování značek mapy. Nevýhodou tohoto pluginu je, že není možné automaticky inicializovat mapu na základě hodnot zeměpisné délky a šířky uloženou u produktu v databázi.

1.4.7 Aplikace zkoumaných funkcí

Po představení systému pro správu obsahu, následující Tab. 3 zobrazuje, jakým způsobem dokáže tento systém splnit důležitou část požadavků na tento systém.

Tab. 3: Výsledky zkoumaných funkcí aplikace – Wordpress

Aplikace zkoumaných funkcí	
Možnost implementace mapy pro zobrazení pozice produktu	Ano, v placené verzi/vlastní implementace
Možnost uchovávat data pro podpis smlouvy	Ano, vlastní implementace
Možnost uchovávat data pro datum dodání	Ano, vlastní implementace
Možnost uchovávat datum konce záruční doby HW	Ano, vlastní implementace
Možnost uchovávat datum konce záruční doby SW	Ano, vlastní implementace
Možnost přidávat nové uživatele pro správu systému	Ano, vlastní implementace
Možnost aktualizovat stav produktu	Ano, vlastní implementace
Možnost kategorizace produktu	Ano, vlastní implementace
Možnost upravovat, přiřazovat a definovat posloupnost stavů produktu ke kategoriím	Ano, vlastní implementace
Využití databáze ARES	Ano, vlastní implementace
Logování změn	Ano, vlastní implementace

Výsledky zkoumaného systému ukazují, že Wordpress dokáže implementovat mapu pro zobrazení polohy produktu, ovšem využívat tuto funkcionalitu je možné u zpoplatněné verze nebo pomocí vlastního vývoje. Co se týče uchovávání důležitých dat, kooperace s dalšími

uživateli, aktualizace a definování stavů produktu je v tomto systému taktéž možné. Wordpress ale nedovoluje využívat plugin pro získávání dat z databáze ARES, neboť tato funkcionality je možná pouze vlastní implementací, např. pomocí jazyku Javascript. Wordpress je postavený na jazyce PHP a je tedy možné implementovat i javascriptový kód, nicméně implementace skriptu pro získání dat z databáze ARES nebo vlastní logování změn bude vyžadovat vlastní vývoj.

1.4.8 Celkový dojem

Wordpress je velmi známý systém pro správu obsahu po celém světě. Na rozdíl od aplikací jako Asana a Freelo, je u Wordpressu **nutné mít základní znalosti pro vývoj** v tomto systému. Instalace pluginů a následné využívání těchto pluginů není zcela jednoduché. Wordpress má obrovskou komunitu, a především velmi rozsáhlou dokumentaci. V případě nejasností a problémů při vývoji je velká pravděpodobnost, že podobný problém již někdo vyřešil. Vývoj v tomto systému vyžaduje desítky hodin samostudia a následně desítky až stovky hodin vývoje, aby bylo možné splnit veškeré požadavky systému. Celkově redakční systém Wordpress nabízí možnost využívat všechny funkcionality, kdy jedna funkcionality je možná řešit zakoupením placené verze a zbytek funkcí je potřeba řešit vlastní implementací. **Vývoj a implementace všech funkcí může být stejně časově náročné** jako implementace vlastního systému.

1.5 Joomla!

Joomla [26] je bezplatný a open-source CMS redakční systém pro publikování webového obsahu. Využívá se pro tvorbu firemních webových stránek, portálů atd., webové stránky pro malé firmy, online časopisy a publikace atd. Podobně jako Wordpress také dovoluje vytvářet vlastní šablony a pluginy. Většina pluginů jsou zdarma pod licencí GPL [27], tj. General Public License. Jedná se o licenci pro libovolné používání, sdílení a úpravy bez nutnosti platit za službu.

1.5.1 Nastavení databáze

Po úspěšném nainstalování systému, nastavení administrátora je dalším krokem nastavení databáze. Proces nastavení databáze, resp. přihlašovací údaje a databáze samotná, je stejná jako v kapitole 1.4.1 Nastavení databáze. Proces nastavení databáze je znázorněn na následujícím Obr. 21.

Nastavení databáze

Vyberte typ databáze. *

MySQLi

Zadejte název hostitele, obvykle 'localhost' nebo '127.0.0.1' nebo název od vašeho hostitele. *

localhost

Buď něco jako 'root' nebo uživatelské jméno přidělené webhostingem *

root

Vytvořené heslo, nebo heslo zadané hostitelem.

Zadejte název databáze. *

dputb

Zadejte prefix tabulky nebo použijte náhodně vygenerovaný prefix tabulky. *

jml_

If you are using an existing database with tables with the same prefix, Joomla will rename those existing tables by adding the prefix "bak_".

Šifrování připojení *

Výchozí (řízeno serverem)

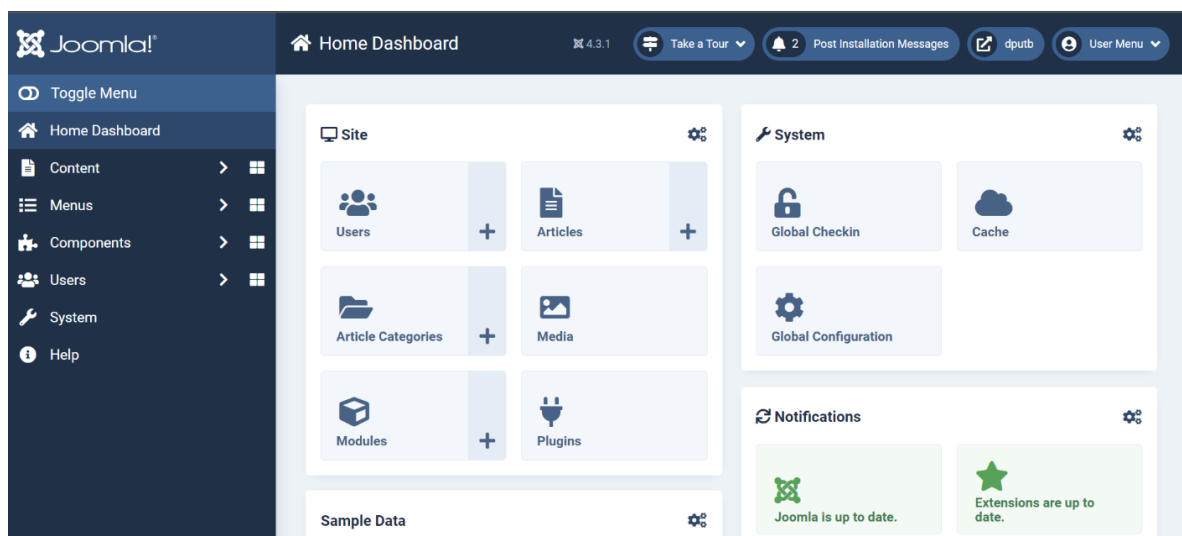
Obr. 21: Nastavení připojení databáze – Joomla

1.5.2 Vzhled systému

Administrace je dostupná po přihlášení do systému. Přihlášení do administrace je možné skrz odkaz:

localhost/nazevprojektu/administrace

Vzhled systému je zobrazen na následujícím Obr. 22.



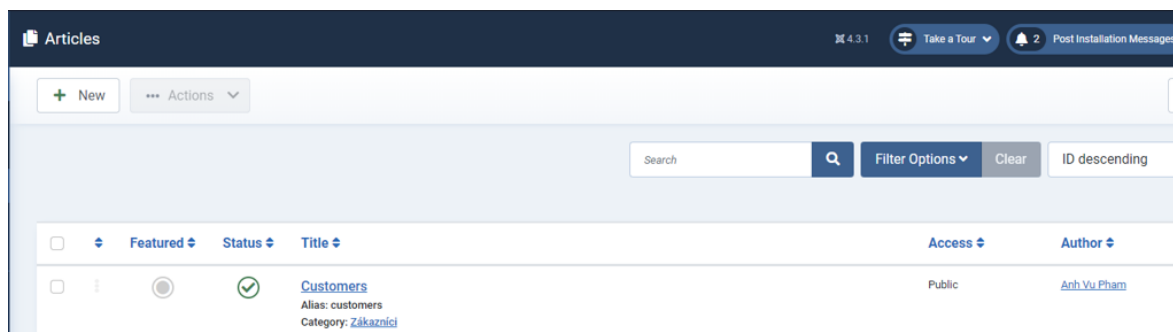
Obr. 22: Vzhled systému – Joomla

Na úvodní stránce nástěnky jsou k dispozici tlačítka pro přidání nových uživatelů, vytváření nových článků, přidání modulů a rozšíření atd.

1.5.3 Přidání a editace klienta a produktu

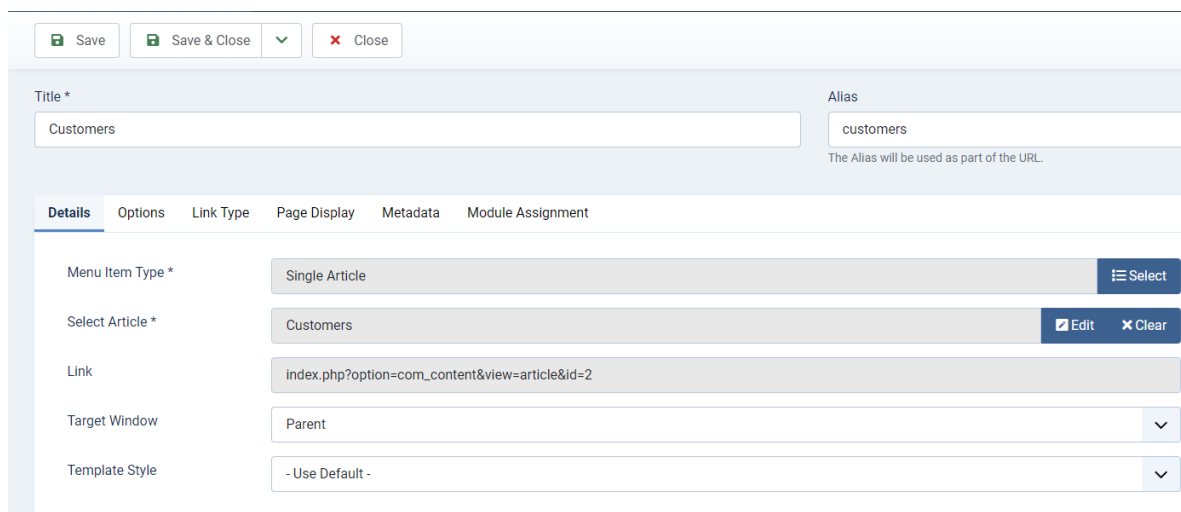
Postup pro přidání nebo editaci klienta a produktu je velmi podobný k Wordpressu. Samotné operace jsou odlišné pouze v tom, že u editace jsou předvyplněné hodnoty uložené v databázi. Zda se bude jednat o klienta nebo o produkt, tak rozdíl bude pouze ve struktuře textových polí. Joomla v základním nastavení nenabízí možnost přidávat zdrojové kódy a nejrychlejším způsobem je vytvoření vlastního formuláře. Na začátek je tedy potřeba nainstalovat si rozšíření **Sourcerer od Regular Labs** [28]. Rozšíření samotné bude zmíněno v následující části práce. Po instalaci tohoto rozšíření je možné v rámci stránek, resp. articles, přidávat zdrojové kódy nebo soubory. Soubory je možné přidávat ve formátu *.html*, *.css*, *.php* a *.js* pouze v placené verzi rozšíření, jinak je možné přidávat zdrojové kódy v rámci WYSIWYG editoru, tj. „*What You See Is What You Get*.“

Prvním krokem je vytvoření stránky, resp. *article*. Při založení je potřeba pouze vyplnit název a alias pro tuto stránku. Vytvořená stránka je potom vidět v přehledu jako na Obr. 23.



Obr. 23: Přehled vytvořených stránek – Joomla

Dalším krokem je vytvoření nového menu. U tohoto menu je potřeba nadefinovat o jaký typ menu se jedná. V tomto kroku u položky „*Menu Item Type*“ zvolíme typ „*Single Article*“, resp. jedna stránka. U položky „*Select Article*“ poté vybereme vytvořenou stránku v předchozím kroku. Způsob vytvoření menu je znázorněno na následujícím Obr. 24.

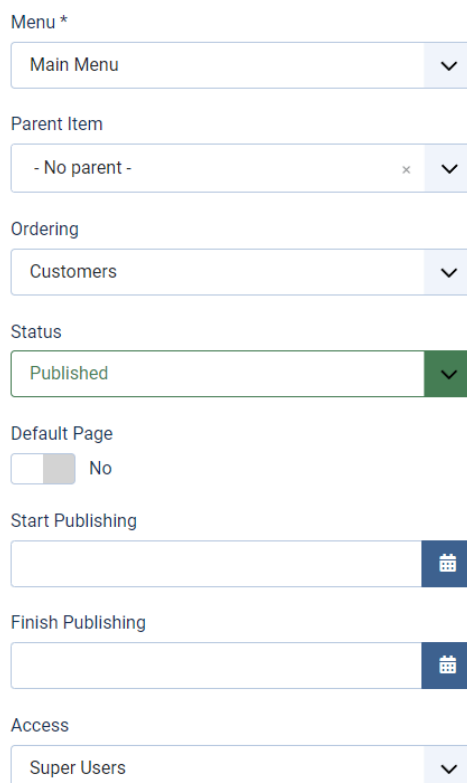


The screenshot shows the Joomla! menu creation interface. At the top, there are buttons for 'Save', 'Save & Close', and 'Close'. Below this, the 'Title *' field contains 'Customers' and the 'Alias' field contains 'customers'. A note states: 'The Alias will be used as part of the URL.' Below the title fields, there are tabs for 'Details', 'Options', 'Link Type', 'Page Display', 'Metadata', and 'Module Assignment'. The 'Details' tab is active, showing the following settings:

- Menu Item Type *: Single Article (with a 'Select' button)
- Select Article *: Customers (with 'Edit' and 'Clear' buttons)
- Link: index.php?option=com_content&view=article&id=2
- Target Window: Parent (dropdown menu)
- Template Style: - Use Default - (dropdown menu)

Obr. 24: Vytvoření menu – Joomla

Po instalaci Joomla je předem vytvořené menu „Main Menu“ a pro nově vytvořené menu nastavíme stejný typ menu. Současně je potřeba nastavit oprávnění, resp. „Access“ pro zobrazení tohoto menu a obsahu. Nastavení je zobrazeno na dalším Obr. 25. Toto provedené nastavení zaručuje, že je formulář pro přidání nového klienta zobrazen až po přihlášení uživatele s nejvyšším oprávněním.

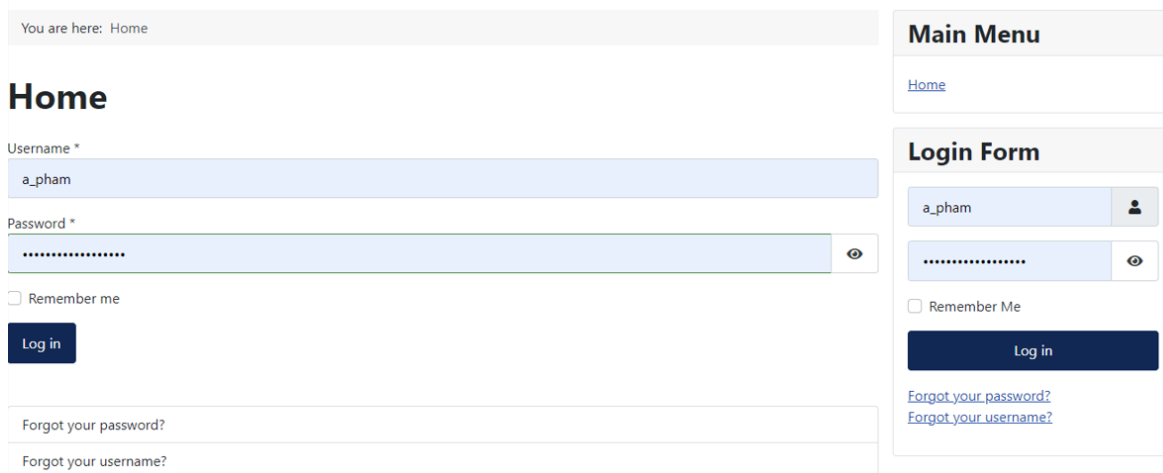


The screenshot shows the Joomla! menu settings interface. The settings are as follows:

- Menu *: Main Menu (dropdown menu)
- Parent Item: - No parent - (dropdown menu with a close button)
- Ordering: Customers (dropdown menu)
- Status: Published (checkbox)
- Default Page: No (checkbox)
- Start Publishing: (calendar icon)
- Finish Publishing: (calendar icon)
- Access: Super Users (dropdown menu)

Obr. 25: Nastavení typu menu a oprávnění – Joomla

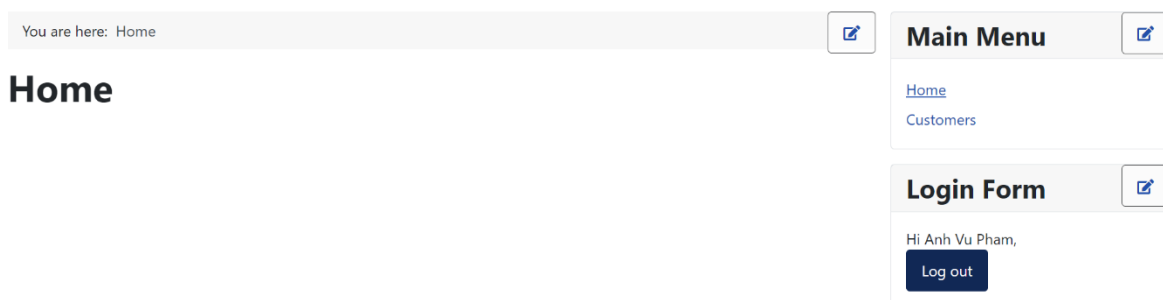
Vzhled úvodní stránky je zobrazen na následujícím Obr. 26 a je zde vidět, že není dostupné menu „Customers“.



The screenshot shows the Joomla! home page before login. At the top left, it says "You are here: Home". Below this is the "Home" heading. The main content area contains a login form with fields for "Username *" (containing "a_pham") and "Password *" (masked with dots). There is a "Remember me" checkbox and a "Log in" button. Below the login form are links for "Forgot your password?" and "Forgot your username?". On the right side, there is a "Main Menu" widget with a link for "Home" and a "Login Form" widget with the same login fields and "Log in" button as the main form. There are also links for "Forgot your password?" and "Forgot your username?" in the right-hand login form.

Obr. 26: Úvodní stránka s přihlašovacím formulářem – Joomla

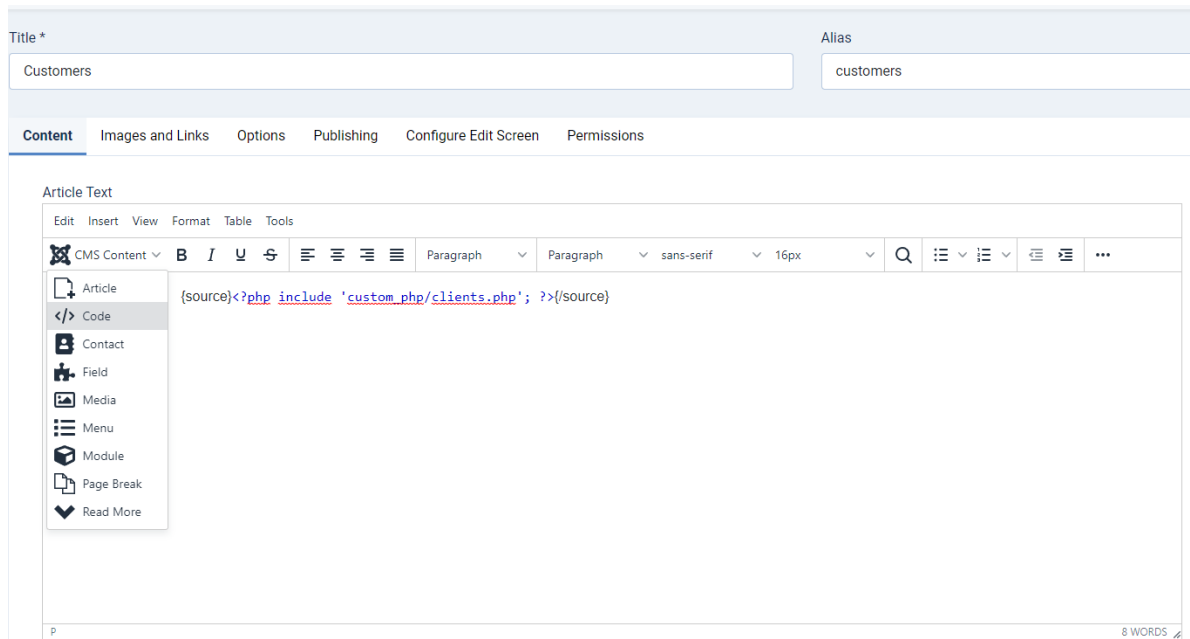
Menu „Customers“ je dostupné až po přihlášení a demonstrováno na dalším Obr. 27.



The screenshot shows the Joomla! home page after login. At the top left, it says "You are here: Home". Below this is the "Home" heading. The main content area is empty. On the right side, there is a "Main Menu" widget with links for "Home" and "Customers". Below it is a "Login Form" widget that now displays "Hi Anh Vu Pham," and a "Log out" button. There are also edit icons (pencil icons) next to the "Main Menu" and "Login Form" widgets.

Obr. 27: Vzhled menu po přihlášení do systému – Joomla

Zdrojové kódy pro spojení s databází a implementace formuláře pro vytvoření nového klienta, resp. Zdrojový kód 1 a Zdrojový kód 2, budou stejné jako v kapitole 1.4.4 Přidání a editace klientů a produktů. Způsob přidání je zobrazen na Obr. 28 pomocí WYSIWYG editoru.



Obr. 28: Způsob vložení zdrojového kódu v nastavení stránky – Joomla

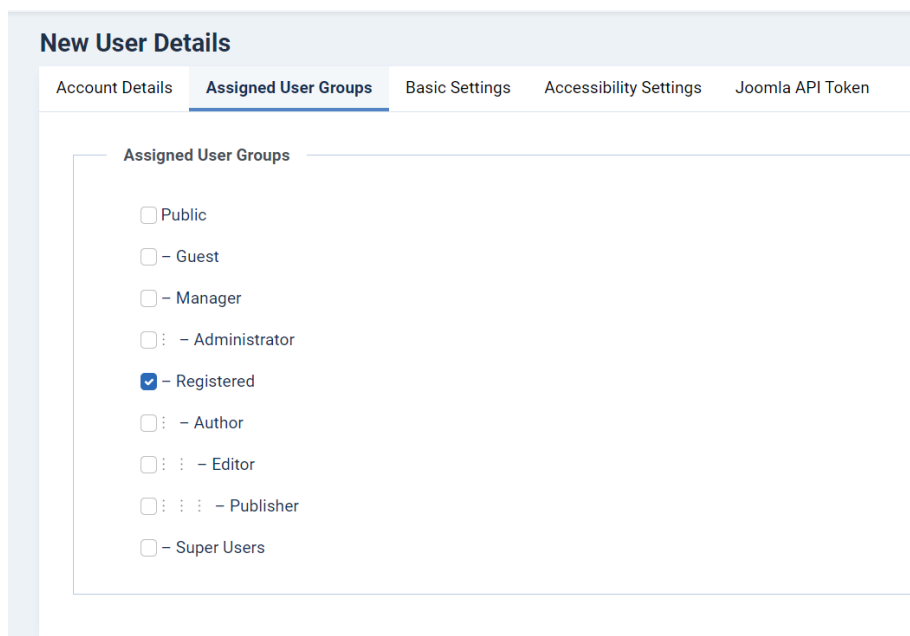
V nastavení stránky je v editoru možné pod položkou „CMS Content“ přidat zdrojový kód díky rozšíření **Sourcerer od Regular Labs**, zmíněno na začátku této kapitoly. V hlavním adresáři je tedy potřeba vytvořit vlastní adresář, např. „*custom-php*“ a do něj vkládat vlastní zdrojové kódy. Není to nutností, ale pouze pro lepší přehlednost. Pomocí PHP funkce *include* stačí pouze zadat cestu k souboru se zdrojovým kódem.

1.5.4 Přidání nového uživatele

Přidání nového uživatele do systému je stejně jednoduchý proces jako u ostatních řešení. V systému Joomla je možné přidat uživatele do systému na základě vyplněného formuláře s povinnými údaji. Formulář pro přidání nového uživatele je zobrazen na dalším Obr. 29.

Obr. 29: Formulář pro vytvoření uživatele – Joomla

Mezi povinné údaje patří jméno uživatele, uživatelské jméno pro přihlášení, přístupové heslo, které musí mít minimálně dvanáct znaků a na závěr e-mailová adresa. Kromě těchto údajů je možné i nastavit skupinu oprávnění uživatele. Toto nastavení je zobrazeno na Obr. 30.



Obr. 30: Nastavení uživatelských práv – Joomla

1.5.5 Pluginy

Joomla není tak známá jako Wordpress, ale také má nabídku spousty rozšíření. Stejně jako u Wordpressu je možné instalovat pluginy ostatních uživatelů do vlastního systému. Část těchto pluginů je zdarma a některé nadstavbové funkce mohou být zpoplatněny.

Sourcerer

Sourcerer [28] od firmy **Regular Labs** je plugin, který dokáže umožnit přidávat zdrojové kódy nebo soubory do obsahu webové stránky vytvořená prostřednictvím systému Joomla. Do obsahu je možné přidávat zdrojové kódy skriptovacích jazyků PHP a Javascript, značkovací jazyk HTML nebo kaskádové styly CSS. Tyto zdrojové kódy je možné přidávat do vytvořených článků, kategorií, modulech, komponentách, hlavičce HTML stránky atd. Jednotlivé kódy je totiž možné přidávat do **WYSIWYG** editoru. Standardně většina textových editorů v systému Joomla odstraňuje části HTML kódu, jako např. Javascript a značky pro vložení videa. **Sourcerer** v tomto nemá žádná omezení. Placená verze vychází na 49 € ročně, tj. zhruba 1140 Kč.

My Maps Location

My Maps Location [29] je placený plugin, který dokáže zaznamenávat, ukládat, kategorizovat umístění na mapě. Kromě map od společnosti **Google** je možné využívat mapy **OpenStreetMap** a **BingMaps**. **My Maps Location** nabízí responzivní design, směřování aktuální polohy, automatické vyplňování informací o poloze (zeměpisná délka a šířka), možnost zobrazovat více bodů na mapě, modul pro načtení vlastních míst atd. Jelikož se jedná o placenou verzi, tak je často aktualizovaná. Předplatné [30] vychází na 49 € ročně, tj. zhruba 1140 Kč.

GMapFP: Google Map or Openstreet Map

GMapFP [31] je bezplatný plugin pro přidávání map do systému Joomla. **GMapFP** nabízí možnost využívat mapy od společnosti **Google** a **OpenStreetMaps** pouze ve verzi Joomla 4. Jedná se o nativní komponentu kompatibilní se systémem Joomla a dostupná ve francouzském a anglickém jazyce. Součástí pluginu jsou i překlady do některých jazyků. **GMapFP** také nabízí přidání libovolných míst na mapě, kategorizaci těchto míst, zobrazování informačních bublin atd.

1.5.6 Aplikace zkoumaných funkcí

Následující Tab. 4 zobrazuje, jakým způsobem dokáže splnit důležitou část požadavků na systém zmíněné v kapitole 1.1 Cíl řešerše.

Tab. 4: Výsledky zkoumaných funkcí aplikace – Joomla

Aplikace zkoumaných funkcí	
Možnost implementace mapy pro zobrazení pozice produktu	Ano, v placené verzi/vlastní implementace
Možnost uchovávat data pro podpis smlouvy	Ano, vlastní implementace
Možnost uchovávat data pro datum dodání	Ano, vlastní implementace
Možnost uchovávat datum konce záruční doby HW	Ano, vlastní implementace
Možnost uchovávat datum konce záruční doby SW	Ano, vlastní implementace

Možnost přidávat nové uživatele pro správu systému	Ano, vlastní implementace
Možnost aktualizovat stav produktu	Ano, vlastní implementace
Možnost kategorizace produktu	Ano, vlastní implementace
Možnost upravovat, přiřazovat a definovat posloupnost stavů produktu ke kategoriím	Ano, vlastní implementace
Využití databáze ARES	Ano, vlastní implementace
Logování změn	Ano, vlastní implementace

Podle výsledků zkoumaného systému Joomla, je zřejmé že Joomla v implementaci zkoumaných funkcí je na tom dost podobně jako předchozí CMS redakční systém Wordpress. Do systému Joomla je totiž také možné implementovat mapu pro zobrazení polohy produktu v placené i neplacené verzi. Neplacená verze **GMapFP** nenabízí možnost načítat data z databáze a vytvářet nové body na mapě a tyto body poté zobrazovat. Nejprůvětivější variantou je vlastní implementace. Stejně jako u systému Wordpress, tak v systému Joomla je řešením uchování důležitých dat, resp. jednotlivé termíny pro podpis smlouvy apod., vlastní implementace. V systému Joomla neexistuje rozšíření nebo jiná externí možnost, jak využívat databázi ARES pro získávání dat o klientech, logování změn, možnost aktualizovat stavy produktu a možnost upravovat, přiřazovat a definovat posloupnost stavů ke kategoriím. Tyto funkcionality je možné přidat pomocí vlastní implementace. **I v tomto systému je vyžadována základní znalost** pro práce se systémem, počínaje vytvářením stránek, menu, přidání modulů a rozšíření atd.

1.5.7 Celkový dojem

Systém na první pohled vypadá velmi přehledně a moderně. Funkcionality na vytváření stránek, vytvoření menu, formuláře apod. je bez znalostí tohoto systému lehce obtížné. Stejně jako u redakčního systému Wordpress, tak i **Joomla vyžaduje mít základní znalost vývoje** v tomto systému a ideálně i znalosti v oblasti programování ve skriptovacím jazyce PHP. Instalace pluginů a následné využívání těchto pluginů jsou ve většině případů docela jednoduché. Přikládání zdrojových kódů je v tomto systému velmi jednoduché díky pluginu **Sourcerer**. Pro některé funkcionality zmíněné v kapitole 1.1 Cíl řešerše existují pluginy.

Podle výsledků z předchozí kapitoly je zřejmé, že **většina funkcí vyžaduje vlastní implementaci. Samostudium, vývoj a implementace** všech funkcí může být **stejně časově náročné** jako implementace vlastního systému.

1.6 Celkové srovnání

V této kapitole v rámci celkového srovnání jsou srovnány zkoumané softwarové platformy, tj. Asana a Freelo, a systémy pro správu obsahu, tj. Wordpress a Joomla, s vlastním redakčním systémem. Následující Tab. 5 popisuje, zda jednotlivé technologie dokážou implementovat zkoumané funkce zmíněné v kapitole 1.1 Cíl řešerše.

Tab. 5: Výsledky zkoumaných funkcí a metod aplikace – shrnutí

Aplikace zkoumaných funkcí napříč technologiemi					
Funkce/Technologie	Asana	Freelo	Wordpress	Joomla	Vlastní CMS
Možnost implementace mapy pro zobrazení pozice produktu	Ne	Ne	Ano ¹	Ano ¹	Ano ²
Možnost uchovávat data pro podpis smlouvy	Ano	Ne	Ano ²	Ano ²	Ano ²
Možnost uchovávat data pro datum dodání	Ano	Ne	Ano ²	Ano ²	Ano ²
Možnost uchovávat datum konce záruční doby HW	Ano	Ne	Ano ²	Ano ²	Ano ²
Možnost uchovávat datum konce záruční doby SW	Ano	Ne	Ano ²	Ano ²	Ano ²

¹ Implementace je možná pomocí placené verze pluginu nebo vlastní implementací

² Implementace je možná pomocí vlastní implementace

Možnost přidávat nové uživatele pro správu systému	Ano	Ano	Ano ²	Ano ²	Ano ²
Možnost aktualizovat stav produktu	Ano	Ano	Ano ²	Ano ²	Ano ²
Možnost kategorizace produktu	Ne	Ne	Ano ²	Ano ²	Ano ²
Možnost upravovat, přiřazovat a definovat posloupnost stavů produktu ke kategoriím	Ne	Ne	Ano ²	Ano ²	Ano ²
Využití databáze ARES	Ne	Ano	Ano ²	Ano ²	Ano ²
Logování změn	Ano	Ano	Ano ²	Ano ²	Ano ²

Podle výsledků zkoumaných metod jsou CMS redakční systémy lepší volbou než softwarové platformy, tj. Asana a Freelo, neboť redakční systémy umožňují implementaci všech zkoumaných funkcí. Nevýhodou u existujících řešení, tj. Wordpress a Joomla je, že u implementace mapy je vyžadována placená verze pro správné využití, všechny ostatní funkce je potřeba implementovat stejně jako při vývoji vlastního systému a vývoj je možný pouze se znalostí systému. Veškeré softwarové platformy a redakční systémy zmíněné v této rešerši byly vyzkoušeny. Odhad pro práci se softwarovými platformami se odhaduje na 5 MD, tj. ManDay [32], resp. doba odpovídající dnu práce jednoho pracovníka. Běžně je 1 MD roven osmi hodinám práce, ale může se lišit na základě dohody. V rešerši jsou představeny postupy pro vytvoření nového klienta a bylo zjištěno, že všechny zkoumané funkce, v systémech Wordpress a Joomla, nebylo možné správně implementovat a vyžadují dlouhou přípravu a čas pro realizaci. Implementace vlastního řešení se odhaduje na 30 MD a řešení pomocí existujících řešení systémů Wordpress a Joomla se celkově odhaduje na 40 MD.

Pronájem hostingu na celý rok je pro všechny tři druhy redakčních systémů stejný. Při objednání předplatného **Premium** formy u aplikace Asana jsou potřeba minimálně dva uživatelé. Minimální počet uživatelů, pro které je pořizováno předplatné u platformy Asana, jsou dva uživatelé.

Roční předplatné pro platformu Asana vychází na 263,76 € za dva uživatele, tj. zhruba 6 139,01 Kč. U platformy Freelo je cena za dva uživatele 3 840 Kč bez DPH ve formě **Business**. Při řešení CMS redakčních systémů je potřeba myslet i na pronájem hostingu. Ročně je u hostingu **Wedos** [33] cena 468 Kč bez DPH a u hostingu **Active24** [34] je cena 708 Kč bez DPH. Tab. 6 zobrazuje celkový přehled pro cenové a časové srovnání jednotlivých kategorií.

Tab. 6: Cenové a časové srovnání technologií

	Asana	Freelo	Wordpress	Joomla	CMS
Hosting/Cena služby za rok	263,76 € Tj. 6139 Kč 2 uživatelé	3840 Kč 2 uživatelé	468 Kč 708 Kč	468 Kč 708 Kč	468 Kč 708 Kč
Odhadovaný čas vývoje	5 MD	5 MD	30 MD	30 MD	30 MD
Samostudium	-	-	10 MD	10 MD	-

Podle výsledků rešerše je tedy nejlepším řešením vytvoření vlastní webové aplikace pro správu klientů a produktů. Časová náročnost je velmi podobná při řešení redakčních systému Wordpress a Joomla. Výhodou vytvořením vlastního systému jsou snadné budoucí úpravy systému, snadné přidání nových funkcionalit a údržba samotného systému.

2 ROZBOR POUŽITÝCH TECHNOLOGIÍ

Výsledkem rešerše a nejlepším způsobem pro vytvoření aplikace pro správu klientů a produktů je vytvoření vlastního systému. V této části jsou stručně rozebrány jednotlivé technologie použité pro analýzu a realizaci diplomové práce. Jedná se o skriptovací jazyky **PHP** a **Javascript** společně s knihovnou **jQuery**, modelovací jazyk **UML** pro analýzu a návrh systému, značkovací jazyk **HTML**, kaskádové styly **CSS** a také srovnání **PHP** frameworků jako **Symfony**, **Laravel** a **CodeIgniter**.

2.1 PHP

PHP [35] (*Hypertext Preprocessor*) je široce používaný open-source univerzální skriptovací jazyk, který je zvláště vhodný pro vývoj a tvorbu dynamických webových stránek a aplikací a lze jej vložit do statických webových stránek **HTML**. Při skriptování na straně serveru se **PHP** přidává na webovou stránku za účelem generování dynamického obsahu při přístupu prostřednictvím klientského prohlížeče [36]. Pokud je vyžádána webová stránka ze serveru, tak je interpretována část stránky v jazyce **PHP**, provedou se operace skriptů **PHP** a vygeneruje se **HTML kód**, tj. značkovací jazyk *Hypertext Markup Language*, který je výsledkem těchto operací. Tento **HTML** kód je poté odeslán do klientského prohlížeče spolu se zbytkem **HTML** kódů na stránce, čímž je zajištěno plynulé vykreslení obsahu. Webové stránky, které obsahují skript v jazyce **PHP** jsou považovány za dynamické stránky **HTML**, protože se obsah mění na základě výsledků skriptu. Nejnovější verze **PHP** je 8.2.3, která byla vydána v únoru 2023.

2.2 Databáze MySQL

Databáze [37] je organizovaný soubor strukturovaných informací nebo dat, typicky uložený elektronicky v počítačovém systému. **Databáze** je obvykle řízena systémem správy databáze. Relační databáze ukládá data do samostatných tabulek a struktury databáze jsou uspořádány do fyzických souborů optimalizovaných pro rychlost [38]. Logický model s objekty, jako jsou databáze, tabulky, pohledy, řádky a sloupce, nabízí flexibilní programovací prostředí. Jsou nastaveny pravidla upravující vztahy mezi různými datovými poli, jako jsou vztahy 1:1, 1:M nebo M:N. **MySQL** je systém pro správu databází, jsou relační a open source.

2.3 Metronic Framework

Jedná se o víceúčelovou a responzivní šablonu. **Metronic** [39] má jasný a uživatelsky přívětivý vyvážený design, který umožňuje vytvořit atraktivní a snadno ovladatelný administrativní panel pro správu webových stránek. **Metronic** framework je komerční produkt společnosti **Keenthemes**. Součástí celého balíčku jsou vytvořené šablony ve značkovacím jazyce **HTML** společně i se soubory obsahující javascriptový kód a je možné si nasimulovat chování jednotlivých tabulek, grafů, funkcionalit atd.

2.4 Bootstrap

Bootstrap je jeden z nejpoblárnějších **CSS** frameworků pro vytváření responzivních webových stránek a aplikací, a hlavně podporuje mobile-first přístup. Tento framework obsahuje navržené šablony pro různé formuláře, navigace, tlačítka atd. pomocí jazyků **HTML** a **CSS** [40]. Nejnovější verze je nyní **Bootstrap 5** [41].

2.5 Javascript a knihovna jQuery

Javascript [42] je skriptovací jazyk, který umožňuje vytvářet dynamicky aktualizovaný obsah, ovládat multimédia, animovat obrázky atd. Jedná se o především multiplatformní a objektově orientovaný jazyk. Je to jeden z nejznámějších skriptovacích jazyků pro tvorbu webových stránek.

jQuery [43] je rychlá a malá javascriptová knihovna, která umožňuje vytvářet snadné procházení a manipulaci s dokumenty, zpracování událostí, animace, volání Ajax atd. Účelem **jQuery** je výrazně usnadnit používání Javascriptu při tvorbě webové aplikace. **jQuery** [44] přebírá mnoho běžných úkolů, které vyžadují mnoho řádků kódu JavaScript, a zabaluje je do metod, které je možné volat pomocí jediného řádku kódu.

2.6 UML

UML [45] (*Unified Modeling Language*) je grafický modelovací jazyk používaný v oblasti softwarového inženýrství, který představuje součásti konceptu objektově orientovaného programování. Jedná se o obecný způsob, jak definovat celou softwarovou architekturu a strukturu. Je určen pro popis a návrh softwarových systémů a aplikací [40].

2.7 HTML

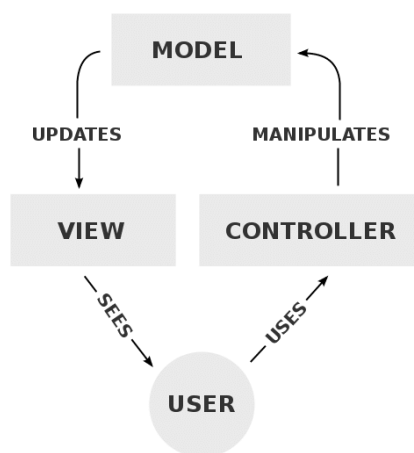
HTML (*HyperText Markup Language*) [40] je značkovací jazyk, který obsahuje základní kostru webové stránky. **HTML** dokáže pomocí odkazů vzájemně propojovat text (*hypertext*) a jednotlivé bloky označit speciálními značkami (*markup*). Webové prohlížeče poté zobrazují tento obsah obalený do značek jazyka **HTML** uživatelům. Webová stránka se v podstatě skládá z textového obsahu (prostý text), odkazů (obrázkové, zvukové atd.) a značek (značky jazyka HTML). Nejnovější verzí je nyní **HTML5**.

2.8 CSS

CSS (*Cascading Style Sheets*) [40] jsou kaskádové styly, které definují způsob zobrazení elementů (např. HTML) webové stránky (barva písma, velikost písma, rozmístění, způsob zobrazení atd.). Jednoduše řečeno, jazyk **HTML** se stará o textový obsah stránky a kaskádové styly se starají o celkový vzhled. Nejnovější verzí je nyní **CSS3**.

2.9 PHP frameworky

Jedná se o platformu pro vytváření **PHP** webových aplikací, které poskytují knihovny kódů pro běžně používané funkce a snižují množství původního kódu, který je potřeba napsat [46]. **PHP frameworky** mají vestavěné knihovny a nástroje a tím pádem je čas potřebný pro vývoj kratší a pro použití funkcí se nemusí psát tolik originálního kódu. Mnoho úkolů, které vývojáři potřebují pro tvorbu webových aplikací, je běžných jako např. validace formulářů, **CRUD** operace (Create, Read, Update, Delete) a místo psaní vlastních funkcí je možné využívat funkce, které jsou součástí frameworku. Využitím některých **PHP frameworků** [47] je potřeba znalost objektově orientovaného programování v jazyce PHP a také se řídí návrhovým vzorem zobrazení modelu **MVC** (Model View Controller)[48], resp. **Model** ukládá obchodní logiku a aplikační data, které předává do **View** (pohled). Uživatel interaguje s pohledem a může zadávat pokyny prostřednictvím **Controlleru** (ovladač), který dává tyto příkazy do **Modelu**. Na Obr. 31 [49] je znázorněna zmíněná interakce **MVC**.



Obr. 31: Interakce Model-View-Controller

Nejznámějšími PHP frameworky jsou **Laravel**, **Symfony**, **CodeIgniter** atd.

2.9.1 Laravel

Jedná se o open-source **PHP** framework [50], který se řídí vzorem **MVC** a pomáhá vývojářům snadněji a jednodušeji řídit úkoly. Původně byl koncipován jako pokročilejší alternativa frameworku **CodeIgniter** a je jedním z nejoblíbenějších frameworků **PHP**. Jako ostatní **PHP** frameworky **Laravel** taktéž umožňuje přístup k mnoha funkcím pro efektivnější práci, snižuje potřebu provádět opakující se úkoly, snadné unit testy atd. Kromě těchto funkcí nabízí **Laravel** robustní systém balíčků (**Packalyst**), který má více než 15 000 balíčků s předpřipravenými funkcemi, které pomáhají vývojářům vytvářet webové aplikace rychleji. Výhodou tohoto frameworku je slušná rychlost [51], obrovská komunita vývojářů, bezpečnost webových aplikace jako např. autentizace, ochrana proti **SQL injection**, **Cross-Site Request Forgery** (CSRF) atd. **Laravel** byl poprvé vydán v listopadu 2011 a podporuje **Composer** [52], tj. nástroj pro správu závislostí v PHP, který umožňuje deklarovat a spravovat (instalovat/aktualizovat) knihovny součástí projektu.

2.9.2 Symfony

Další z open-source **PHP** frameworků je **Symfony** [53], který je známý jako jeden z nejlepších pro vytváření komplexních webových stránek a řídí se vzorem **MVC**. Dále nabízí funkce pro přizpůsobení, které činí webové aplikace uživatelsky přívětivější, systém komponent je mimořádně výkonný a umožňuje vytvářet komplexní produkty s mnoha funkcemi a vybranými vlastnostmi. Výhodou frameworku [54] je snadná instalace a konfigurace, ob-

sahuje opakovaně použitelné komponenty a knihovny **PHP**, obsahuje velkou komunitu vývojářů a podporuje většinu databází jako **MySQL**, **SQLite**, **SQLServer**, **Oracle** atd. **Symfony** byl poprvé vydán v říjnu 2005 a také podporuje **Composer**.

2.9.3 CodeIgniter

CodeIgniter [55] byl poprvé vydán v roce 2006 a je také jedním z open-source **PHP** frameworků, který nevyžaduje stahování obrovského počtu souborů. Jedná se zhruba o 1,2 MB, kdy v podstatě i dokumentace je větší než samotný framework. **CodeIgniter** [56] je vhodný pro tvorbu dynamických webových stránek a nabízí sadu nástrojů s různými předpřipravenými moduly pro vývoj plnohodnotných webových aplikací a také se řídí vzorem **MVC**. Je vybaven cennými funkcemi jako jednoduché zabezpečení, šifrovací kroky a snadné zpracování chyb a obsahuje celou řadu knihoven. Framework zahrnuje obranu proti útokům **Cross-site Request Forgery (CSRF)**, **Cross Site Scripting (XSS)**, **SQL injection**, zásady zabezpečení obsahu atd. Výhodou frameworku je rozhodně rychlost a patří mezi ty nejrychlejší frameworky. Je vhodný pro začátečníky díky jednoduchému postupu a dokumentaci, umožňuje ukládání více metod do mezipaměti (Caching), má také slušnou komunitu vývojářů, generuje čisté URL adresy, podporuje **Composer** atd.

II. PRAKTICKÁ ČÁST

3 ANALÝZA PROJEKTU

Před realizací této aplikace je potřeba provést analýzu a návrh celého systému. V rámci analýzy celého systému budou v této části řešeny funkční a nefunkční požadavky, případy užití a realizace požadavků. Některé požadavky na systém již byly zmíněny v kapitole 1.1 Cíl rešerše. Současně bude sestaven model tříd, fyzický datový model, sekvenční diagramy a na závěr budou představeny drátěné modely.

3.1 Požadavky

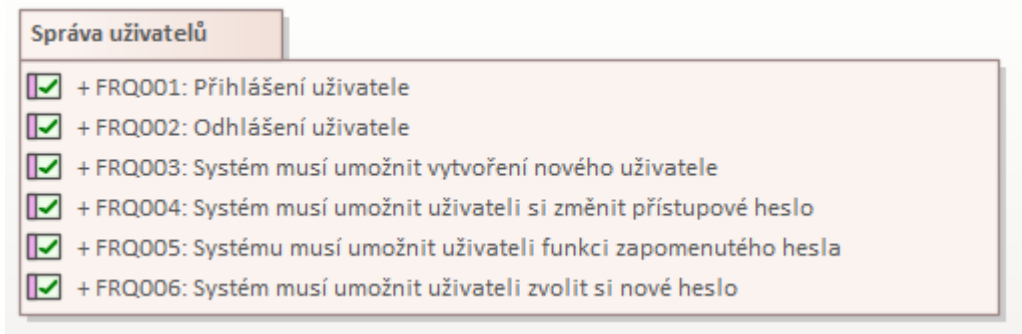
Jak již bylo zmíněno výše, součástí analýzy projektu jsou požadavky. Požadavky v podstatě mohou představovat některé funkce systému. Cílem je tedy vyhledat a zanalyzovat budoucí funkcionalitu systému. Požadavky systému by se měly odrazit na způsobu realizace projektu. Požadavky jsou rozděleny na funkční a nefunkční požadavky.

3.1.1 Funkční požadavky

Funkční požadavky [57] by měly zachycovat očekávání od cílových uživatelů, resp. jakým způsobem se má systém chovat a jakým způsobem má fungovat nebo co musí umět. Jsou to v podstatě vlastnosti produktu nebo funkce, které musí vývojáři implementovat, aby umožnil splnit jejich úkoly. Požadavky by měly být napsány jasně a srozumitelně pro vývojáře a zainteresované osoby. Může se tedy jednat o popisy operací a pracovních postupů, které musí systém umět, autentizace, úrovně oprávnění, dodržování zákonů a předpisů atd. Funkční požadavky jsou rozděleny do třech kategorií:

- Správa uživatelů
 - o Zde jsou popsány základní požadavky pro uživatele v systému, tvorba nových uživatelů a změna přístupového hesla atd.
- Správa produktů
 - o Zde jsou popsány požadavky pro produkty při přidání produktu do systému, kategorizace, logování změn, změny stavů apod.
- Správa klientů
 - o Zde jsou popsány požadavky pro klienty při přidání klienta do systému, uchování kontaktní osoby a fakturačních údajů apod.

Funkční požadavky pro správu uživatelů jsou popsány na Obr. 32.



Obr. 32: Funkční požadavky pro správu uživatelů

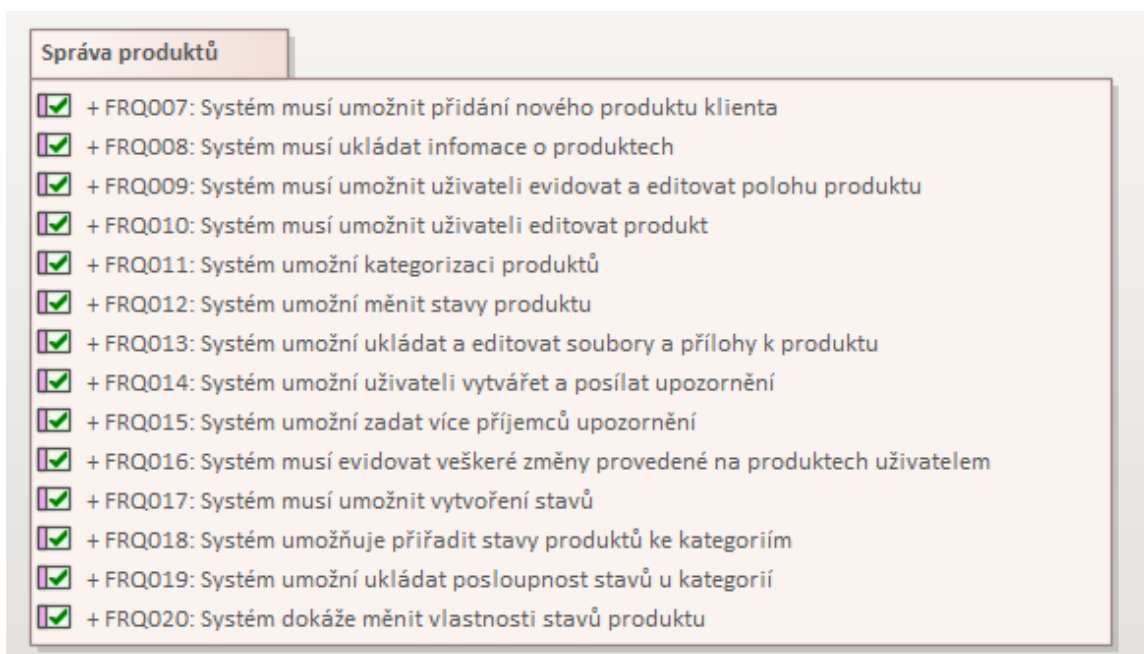
FRQ003: Systém musí umožnit vytvoření nového uživatele

- V systému musí být možné vytvořit nové uživatele. Každý uživatel musí mít aktivní e-mailovou adresu, zadané jméno a příjmení. Autentizace je pomocí e-mailové adresy a přístupového hesla.

FRQ004: Systém musí umožnit uživateli změnit si přístupové heslo

- V systému musí být uživateli umožněno změnit si přístupové heslo do systému.

Další částí požadavků jsou požadavky pro správu produktů. Tyto požadavky jsou zobrazeny na následujícím Obr. 33.



Obr. 33: Funkční požadavky pro správu produktů

FRQ007: Systém musí umožnit přidání nového produktu klienta

- V systému musí být možné přidat nový produkt klienta, který není v databázi produktů. Systém bude evidovat veškeré informace o produktech, jako např. název, cena, typ produktu, kategorie, poloha produktu, výdaje a výnosy, datum vytvoření atd.

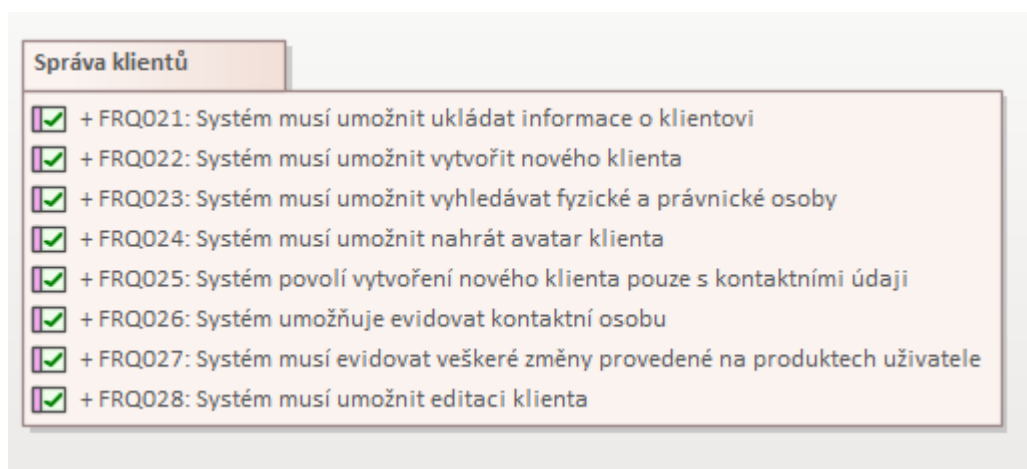
FRQ014: Systém umožní uživateli vytvářet a posílat upozornění

- Součástí tohoto systému bude možnost vytvářet upozornění. Tyto upozornění budou odesílány v pravidelných intervalech prostřednictvím e-mailové zprávy.

FRQ016: Systém musí evidovat veškeré změny provedené na produktech uživatelem

- Systém musí automaticky uchovávat veškeré provedené změny na produktu v databázi. Záznamy o provedených změnách musí obsahovat informace o jaký produkt se jedná, jaká operace byla provedena, kdo a čas poslední provedené změny.

Poslední částí funkčních požadavků jsou pro správu klientů. Obr. 34 popisuje obsah této sady požadavků.



Obr. 34: Funkční požadavky pro správu klientů

FRQ021: Systém musí umožnit ukládat informace o klientovi

- Systém musí ukládat informace o klientovi, kontaktní osobu, fakturační údaje v databázi.

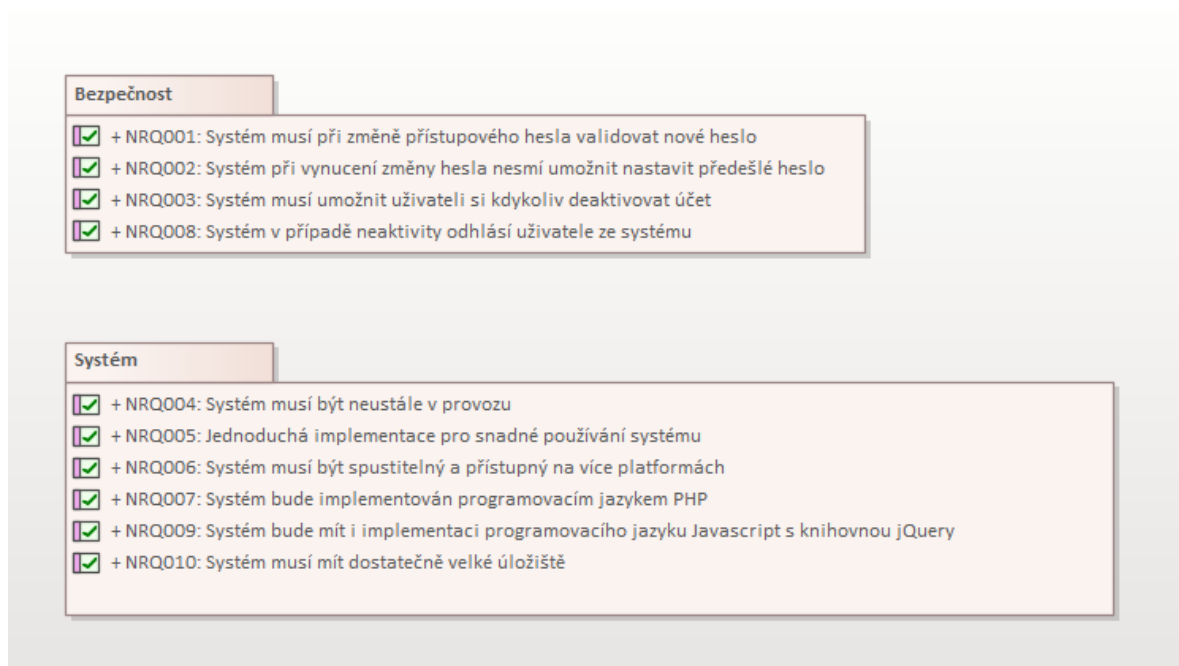
FRQ023: Systém musí umožnit vyhledávat fyzické a právnické osoby

- Systém bude mít funkci pro vyhledávání fyzických a právnických osob z databáze ARES. Tato funkce bude vyžadovat pouze zadání identifikačního čísla osoby a systém v případě nálezu vyplní prázdná pole jako název klienta a adresu sídla.

3.1.2 Nefunkční požadavky

Nefunkční požadavky [57] naopak specifikují, jakým způsobem má být zabezpečený systém, nesouvisí s funkčností systému, ale definuje, jak má fungovat, jak velká by měla být kapacita serveru nebo také jakým způsobem má být implementován. Ve své podstatě řeší požadavky na systém, výkon, spolehlivost, kapacitu, dostupnost, bezpečnost, podporu během provozu atd.

Následující Obr. 35 popisuje nefunkční požadavky systému.



Obr. 35: Nefunkční požadavky pro systém a bezpečnost

NRQ001: Systém musí při změně přístupového hesla validovat nové heslo

- Pokud si uživatel bude chtít změnit přístupové heslo, tak systém musí toto zadané heslo validovat. Přístupové heslo musí obsahovat alespoň jedno velké a malé písmeno a číslici.

NRQ006: Systém musí být spustitelný a přístupný na více platformách

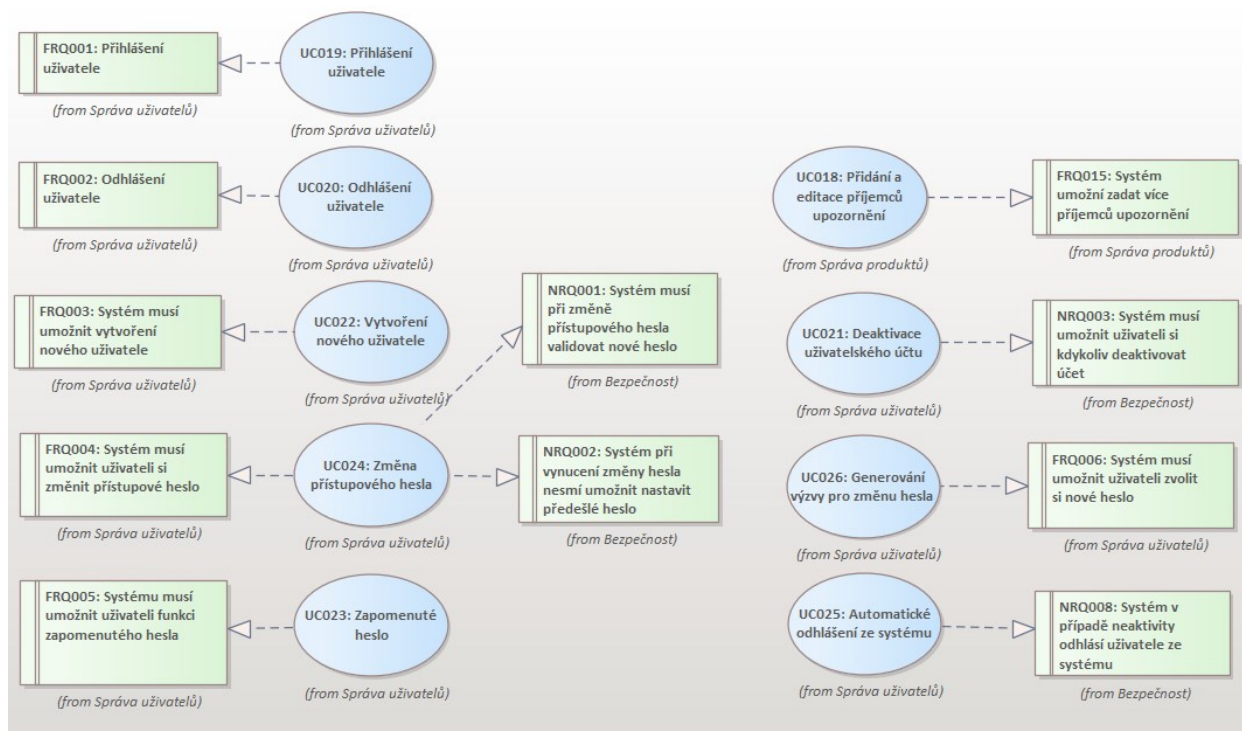
- Systém musí být spustitelný a přístupný na více platformách, jak na počítačové, tak mobilní verzi.

NRQ008: Systém v případě neaktivity odhlásí uživatele ze systému

- Systém bude sledovat aktuální neaktivitu přihlášeného uživatele a po vypršení předem definovaného časového limitu automaticky odhlásí přihlášeného uživatele.

3.1.3 Realizace požadavků

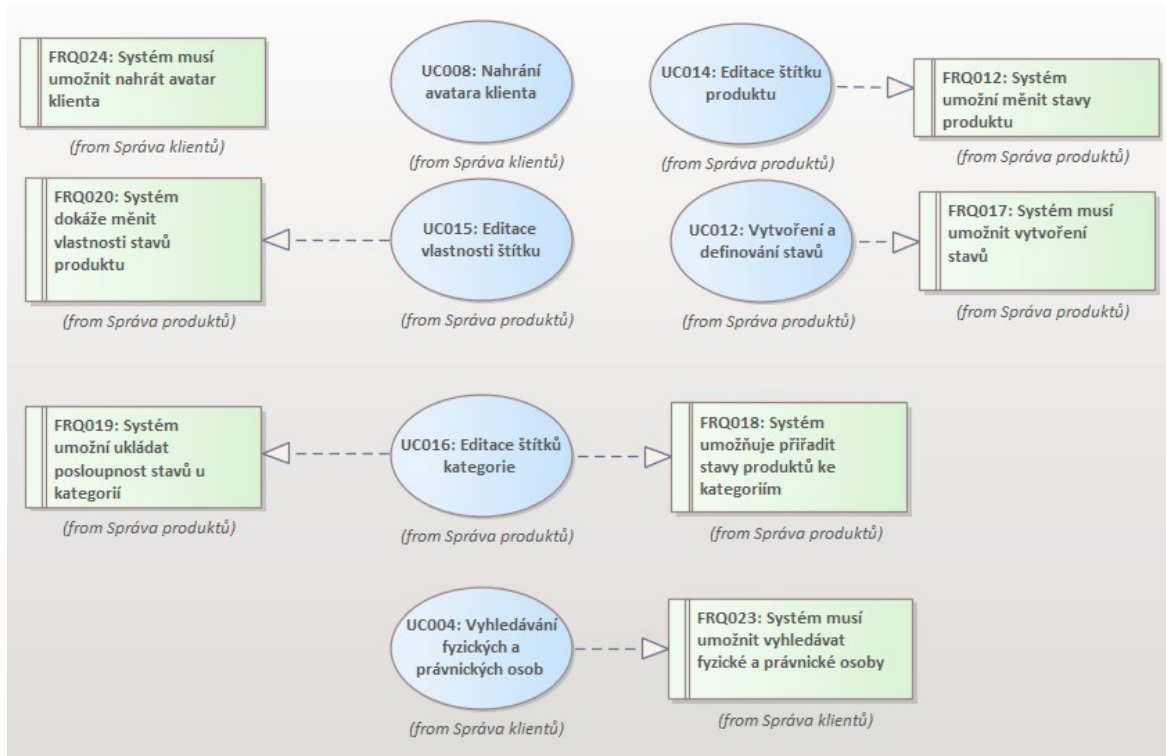
Požadavky jsou realizovány pomocí případu užití. Každý případ užití je realizací alespoň jednoho požadavku. Samotné realizace požadavků jsou znázorněny na Obr. 36, Obr. 37 a Obr. 38.



Obr. 36: Realizace požadavků č. 1

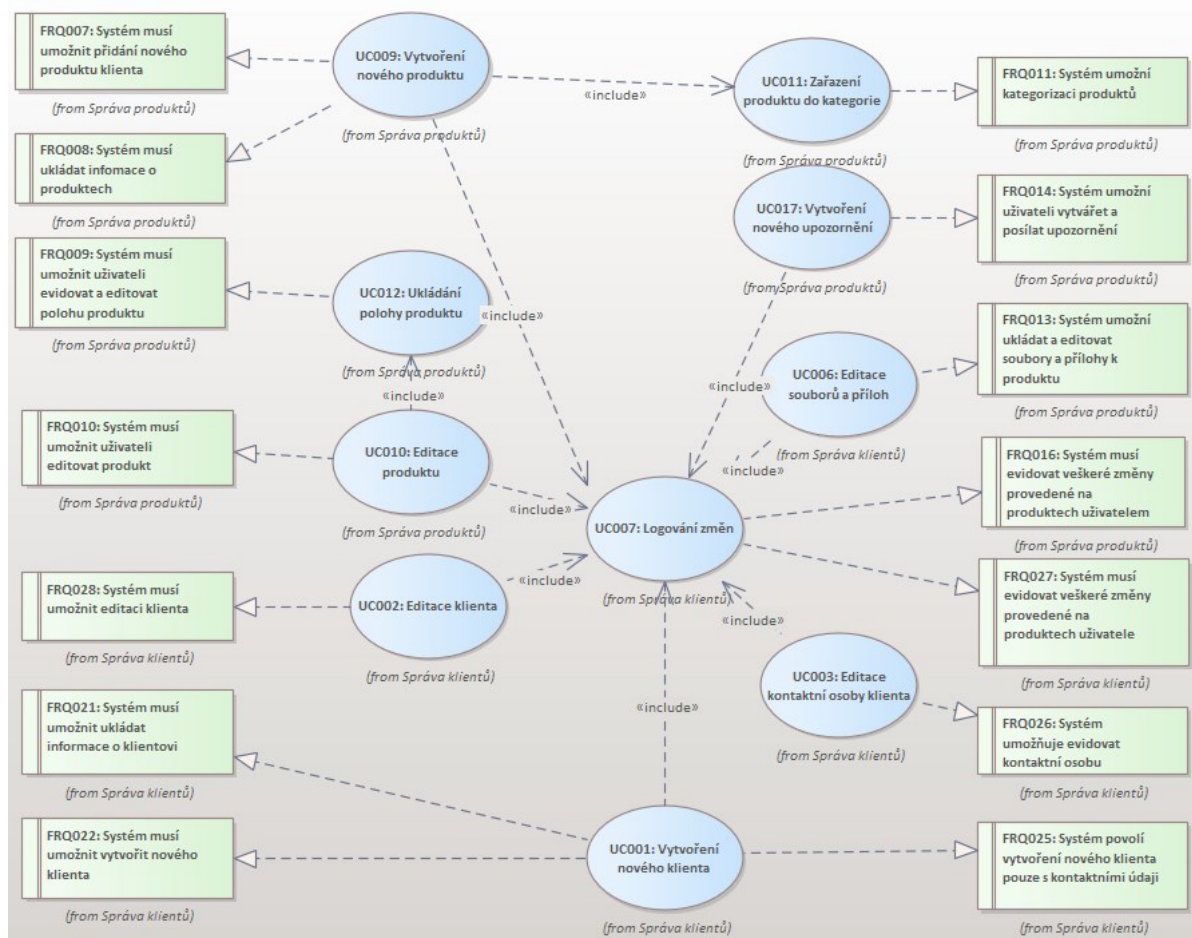
Na předchozím Obr. 36 můžeme vidět realizaci funkčních i nefunkčních požadavků. Například případ užití „Změna přístupového hesla“ je realizací funkčního požadavku **FRQ004**, kde systém musí umožnit uživateli si změnit přístupové heslo a u nefunkčních požadavků **NRQ001** a **NRQ002** se jedná o validace nového přístupového hesla, přičemž nové heslo musí obsahovat alespoň jedno velké a malé písmeno a číslici a současně nesmí být stejné jako heslo předešlé.

V další části realizace požadavků na Obr. 37 můžeme vidět případy užití „Vyhledávání fyzických a právnických osob“ a „Editace kontaktní osoby klienta“ v rámci sady funkčních požadavků pro správu klientů



Obr. 37: Realizace požadavků č. 2

Obr. 38 znázorňuje jednotlivé případy užití, které jsou spojeny s realizací jiného případu užití. Případy užití mohou obsahovat funkce jiného případu užití jako součást jejich běžného zpracování [58]. Příkladem je „Vytvoření nového klienta,“ kdy při vytvoření nového klienta musí být proveden případ užití „Logování změn.“



Obr. 38: Realizace požadavků č. 3

3.2 Model tříd

Diagram modelu tříd [59] v podstatě ukazuje stavební bloky jakéhokoliv objektově orientovaného systému, znázorňuje, jaké má atributy, jaké chování má a zachycuje všechny možné třídy potřebné k systému. Jsou nejužitečnější při ilustraci vztahů mezi třídami, zobecnění, agregace a asociace jsou cenné při definování dědičnosti. Diagram je k nahlédnutí v příloze:

PŘÍLOHA PI: CD → EA projekt ve formátu .qea → Model → Model tříd

V přiloženém diagramu můžeme vidět, že třída *Administrator* dědí od třídy *User*, která reprezentuje fyzickou osobu, resp. se může jednat o administrátora systému nebo kontaktní osobu klienta (*ContactUser*). Pro obě třídy *Administrator* a *ContactUser* jsou atributy *Jméno*, *příjmení* a *e-mail* společné a současně mají společnou funkci *editUserInfo()* pro editaci uživatelských údajů. Dalším příkladem je třída *Log*. Tato třída totiž má funkci *createLog()* a je realizována třídami *Product*, *Client* a *File* při přidávání, nahrávání nebo editaci.

Třída *Product* je jedna z těch větších a reprezentuje produkty v systému pro správu zákazníků, resp. klientů, a produktů. K produktům se totiž ukládá typ produktu, kategorie produktu, stav produktu, náklady a výnosy atd. Ke každému produktu musí být přiřazený klient, bez kterého není možné vytvořit produkt. Produkty dále mohou mít definované upozornění, např. pro záruční lhůtu, nebo přiložené soubory.

3.3 Fyzický datový model

Fyzický datový model [60] vizuálně představuje strukturu dat implementovanou jako schéma relační databáze. Kromě poskytování vizuální abstrakce databázové struktury je u fyzického datového modelu důležitou výhodou to, že z modelu můžeme automaticky odvodit schéma databáze. To je možné díky bohatosti metadat zachycených fyzickým datovým modelem a blízkému mapování na aspekty schématu databáze, jako jsou tabulky, sloupce, primární a cizí klíče. Diagram je k nahlédnutí v příloze:

PŘÍLOHA PI: CD → EA projekt ve formátu .qea → Model → Fyzický datový model

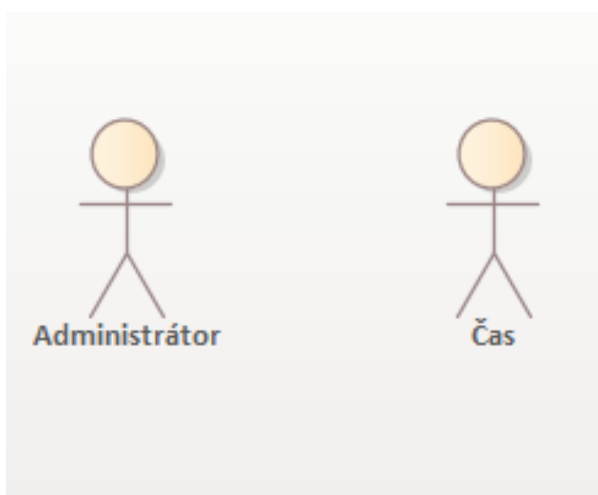
Ve vytvořeném fyzickém modelu v příloze je vytvořeno celkově 22 tabulek (*AuthUser*, *AuthUserGroup*, *AuthUser*, *Category*, *CategoryCondition*, *Client*, *Condition*, *ContactUser*, *Cost* a *CostType*, *File* a *FileType*, *Log* a *LogType*, *Product* a *ProductType*, *Reminder*, *ReminderRecipient*, *ReminderType*, *District*, *Region* a *State*). Tabulky, které obsahují slovo *User* nebo *Group*, definují uživatele v systému a je možné v budoucnu přidávat role uživatelům, resp. tabulka ***AuthGroup*** slouží pro definování nových oprávnění, tabulka ***AuthUserGroup*** slouží pro přiřazení vytvořených oprávnění k uživateli tabulka ***AuthUser*** ukládá informace o vytvořených uživateli, jako jsou e-mail, jméno a příjmení, zašifrované heslo, datum poslední změny hesla atd. Pro splnění některých zkoumaných funkcí zmíněné v rešerši v kapitole 1.1 Cíl rešerše jsou v tabulce ***Products*** vytvořeny sloupce *latitude* a *longitude* (zeměpisná šířka a délka), *signature_date* (datum podpisu smlouvy), *build_date* (datum dodání), *guarantee_date_hw* a *guarantee_date_sw* (datum konce záruční doby pro hardwarové a softwarové řešení) a *id_condition* (stav produktu). Současně jsou v tabulce ***CategoryCondition***, ve vztahu M:N³ [61], uloženy cizí klíče *id_category* a *id_condition* z tabulek ***Category*** a ***Condition*** pro přiřazení stavů ke kategoriím

³ Vztah, kdy jeden záznam nebo více záznamů v tabulce A souvisí s jedním nebo více záznamů v tabulce B a naopak

a také pro uložení posloupnosti stavů. Veškeré provedené operace, které jsou definované v tabulce **LogType**, jsou následně zaznamenány v tabulce **Log** jako např. vytvoření a editace produktů a klientů, nahrávání a editace souborů a příloh atd.

3.4 Aktéři

Diagram případu užití ukazuje interakci mezi systémem a entitami (objekty) mimo systém, které jsou reprezentovány jako aktéři [62]. V rámci funkčních požadavků není definováno, zda je potřeba mít více druhů uživatelů, a proto je v tomto systému jediný druh uživatele, tj. **Administrátor**. Aktéři tohoto systému jsou znázorněni na následujícím Obr. 39.

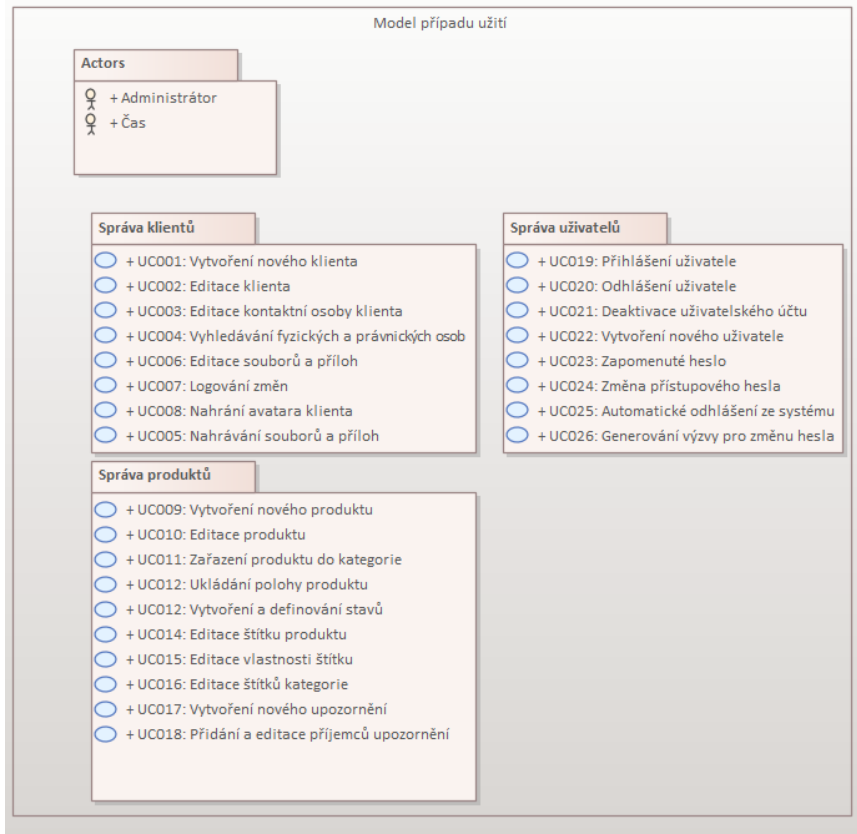


Obr. 39: Aktéři

Aktéři představují role, které mohou zahrnovat lidské uživatele, externí hardware, jiné systémy nebo např. **Čas**, který po uplynutí předem definované doby upozorní systém a odhlásí uživatele ze systému nebo může kontrolovat platnost hesla a na základě vypršení lhůty platnosti hesla generovat výzvu pro uživatele k vytvoření nového přístupového hesla.

3.5 Případy užití

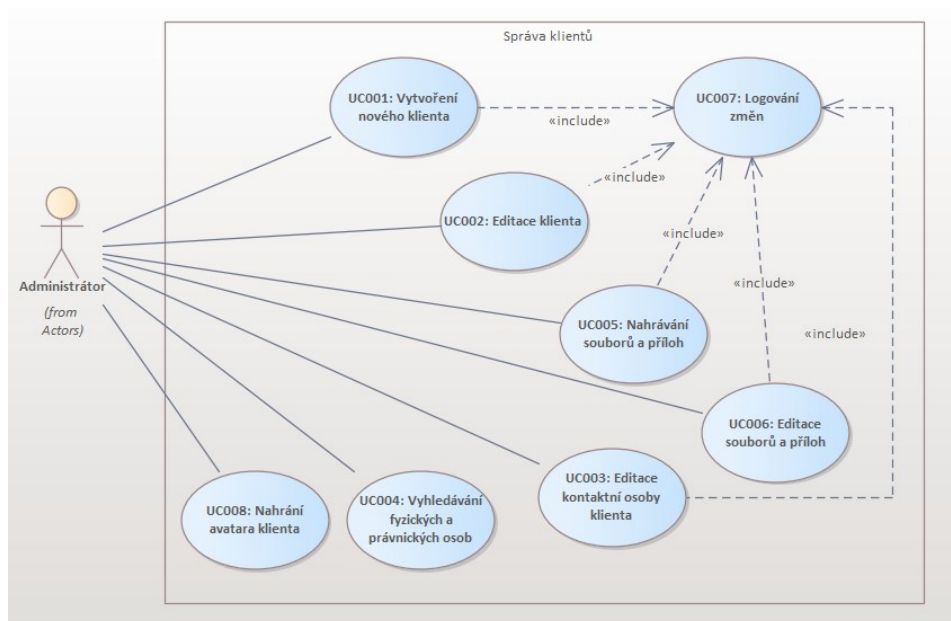
Případy užití nebo také **Use Case** [63] zachycují a realizují požadavky systému, jak bylo zmíněno v kapitole 3.1.3 Realizace požadavků, a jsou prostředkem pro komunikaci s uživateli a dalšími zainteresovanými stranami (Aktéři) o tom, co má systém dělat. Případy užití jsou také definovány jako specifikace činností a chybových posloupností, které může systém vykonávat a jsou popsány scénářem, resp. posloupnost kroků popisující interakci mezi uživatelem a systémem [40]. Na následujícím Obr. 40 můžeme vidět kompletní model případu užití, včetně aktérů v systému a začleněné do správy uživatelů, klientů a produktů.



Obr. 40: Model případu užití

3.5.1 Správa klientů

Případy užití pro správu klientů jsou zobrazeny na následujícím Obr. 41 a znázorňuje vztahy mezi jednotlivými případy užití a aktérem *Administrátor*.

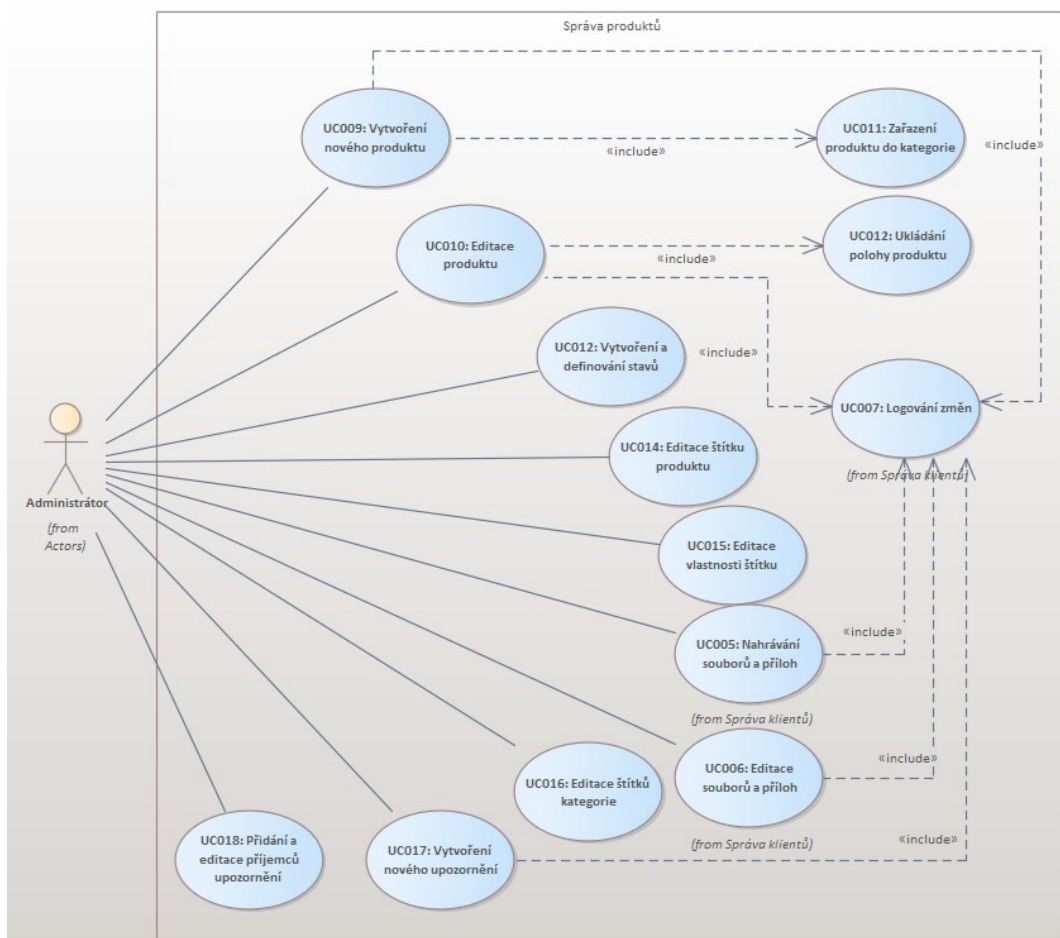


Obr. 41: Případy užití pro správu klientů

Na Obr. 41 můžeme vidět, že aktér *Administrátor* dokáže v systému pro správu klientů a produktů vytvářet a editovat klienty i kontaktní osoby a společně i s nahráváním a editací souborů a příloh musí být provedeno logování změn. Jediný případ užití *Vyhledávání fyzických a právnických osob* nevyžaduje vykonání případu užití pro logování změn.

3.5.2 Správa produktů

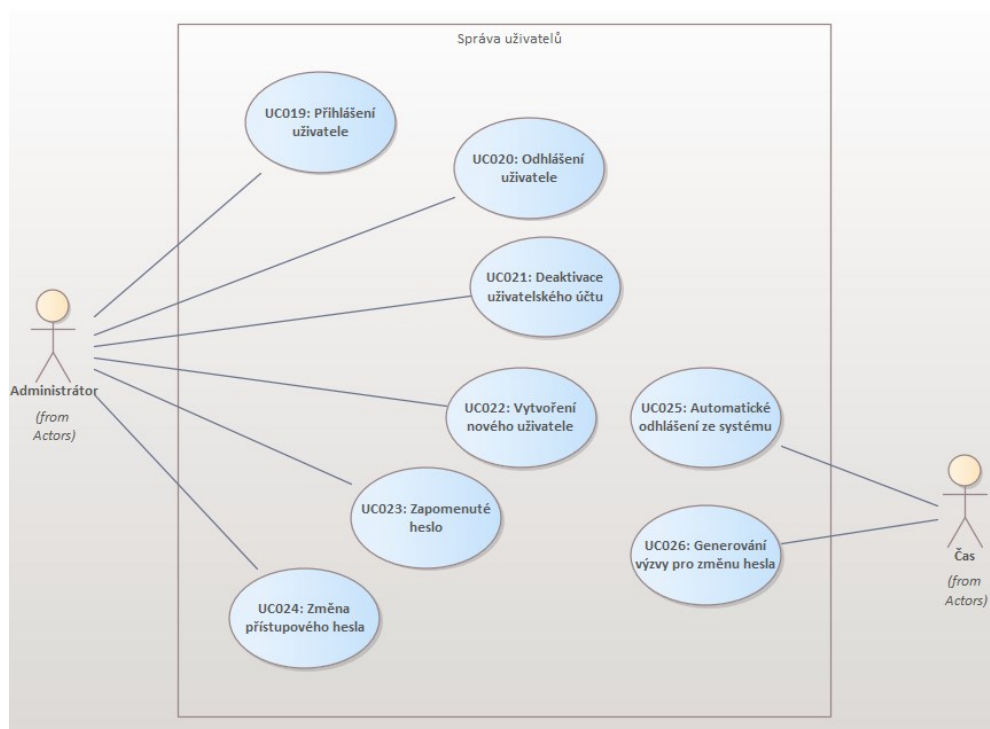
Na dalším Obr. 42 jsou zobrazeny případy užití pro správu produktů. Správa produktů obsahuje případy užití pro vytvoření a editace produktu, stavů produktu, nahrávání a editace souborů a příloh, vytváření upozornění a přidání a editace příjemců upozornění. Při vytvoření nového produktu je zároveň produkt zařazen do kategorie a při editaci produktu dochází i k uložení polohy produktu. Současně případ užití *Logování změn* musí být proveden při vykonání každého jednoho případu užití pro *Vytvoření nového produktu*, *Editace produktu*, *Nahrávání souborů a příloh*, *Editace souborů a příloh* nebo *Vytvoření nového upozornění*.



Obr. 42: Případy užití pro správu produktů

3.5.3 Správa uživatelů

Poslední částí případu užití jsou pro správu uživatelů v systému a jsou zde dva aktéři *Administrátor* a *Čas*. Aktér *Administrátor* musí mít možnost v systému vytvářet nové uživatele, musí mít možnost deaktivace vlastního uživatelského účtu, možnost si změnit přístupové heslo a současně musí být aktérovi umožněno přihlášení a odhlášení ze systému. Aktér *Čas* po uplynutí předem definované doby neaktivity uživatele automaticky odhlásí uživatele ze systému a po vypršení platnosti hesla vygeneruje výzvu na změnu přístupového hesla. Tyto případy užití jsou znázorněny na následujícím Obr. 43.



Obr. 43: Případy užití pro správu uživatelů

3.5.4 Scénáře případu užití

Součástí jednotlivých případu užití jsou i postupy, resp. scénáře případu užití, popisující interakci mezi uživatelem a systémem. Veškeré scénáře případu užití jsou definované u jednotlivých případu užití v příloze:

PŘÍLOHA PI: CD → EA projekt ve formátu .qea → Model → Případy užití

Jednotlivé scénáře jsou zobrazeny při dvojitým kliknutí na daný případ užití. Následně jsou představeny dva příklady ke každé části případu užití (Správa klientů, produktů a uživatelů).

Přihlášení uživatele

Tab. 7: Scénář případu užití: Primární scénář: Přihlášení uživatele

Název případu užití: Přihlášení uživatele		
ID: UC019		
Charakteristika: Scénář pro přihlášení uživatele		
Primární aktér: Uživatel		
Vedlejší aktéři: Nejsou		
Vstupní podmínky: System musí umožnit uživateli se přihlásit. Uživatel, který se přihlašuje, musí mít vytvořený uživatelský účet. System musí evidovat čas posledního přihlášení.		
Výstupní podmínky: Přihlášení musí být úspěšné.		
Primární scénář:		
Krok	Aktér	Popis
1	Uživatel	Odeslání požadavku pro zobrazení formuláře pro přihlášení uživatele
2	System	Zobrazení formuláře pro přihlášení uživatele
3	Uživatel	Zadání e-mailové adresy
4	Uživatel	Zadání přístupového hesla
5	Uživatel	Potvrzení údajů odesláním požadavku přihlášení
6	System	Kontrola správnosti údajů
7	System	Přihlášení uživatele
Alternativní scénáře: UC019a: Alternativní scénář pro kroky 3 a 4 → Pokud jsou špatně zadány data ve formuláři, systém vyzve uživatele k opětovnému zadání → Tab. 8		

Tab. 8: Scénář případu užití: Alternativní scénář: Přihlášení uživatele

Název – Alternativní scénář: Opakované přihlášení

ID: UC019a		
Charakteristika: Zachycení případu, kdy uživatel zadal neplatný údaj ve formuláři pro přihlášení		
Alternativní scénář:		
Krok	Aktér/systém	Popis
1	System	V případě špatně vyplněného formuláře, systém vyzve uživatele k opětovnému zadání
2	Uživatel	Zadání e-mailové adresy
3	Uživatel	Zadání přístupového hesla
4	Uživatel	Potvrzení údajů odesláním požadavku přihlášení
5	System	Kontrola správnosti údajů
6	System	Přihlášení uživatele

Vytvoření nového klienta

Tab. 9: Scénář případu užití: Primární scénář: Vytvoření nového klienta

Název případu užití: Vytvoření nového klienta
ID: UC001
Charakteristika: Scénář zachycující postup pro vytvoření nového klienta
Primární aktér: Administrátor
Vedlejší aktéři: Nejsou
Vstupní podmínky: Administrátor , který chce vytvořit nového klienta, musí být přihlášen. System musí umožnit administrátorovi vytvořit nového klienta. Pro vytvoření klienta musí být vyplněné kontaktní údaje.
Výstupní podmínky: Vytvoření nového klienta
Primární scénář:

Krok	Aktér	Popis
1	Administrátor	Přihlášení uživatele s oprávněním <i>Administrátor</i>
2	Administrátor	Odeslání požadavku pro vytvoření nového klienta
3	System	Zobrazení formuláře pro vytvoření nového klienta
4	Administrátor	Zadání všech povinných a nepovinných polí
5	Administrátor	Potvrzení zadaných hodnot
6	System	Kontrola zadaných hodnot
7	System	Vytvoření nového klienta

Alternativní scénáře:

UC001a: Alternativní scénář pro krok 4 → Pokud v systému existuje klient se stejně zadaným identifikačním číslem osoby (IČO), tak systém nevytvoří nového uživatele → Tab. 10

Primární a alternativní scénáře pro přihlášení uživatele jsou popsány v Tab. 7 a Tab. 8

Tab. 10: Scénář případu užití: Alternativní scénář: Vytvoření nového klienta

Název – Alternativní scénář: Vytvoření nového klienta		
ID: UC001a		
Charakteristika:		
Zachycení případu, kdy se <i>Administrátor</i> pokusil o vytvoření klienta s existujícím IČO.		
Alternativní scénář:		
Krok	Aktér/system	Popis
1	System	V případě špatně vyplněného formuláře, systém vyzve uživatele k opětovnému zadání
2	Administrátor	Zadání všech povinných a nepovinných polí
3	Administrátor	Zadání unikátního IČO
4	System	Kontrola zadaných hodnot
5	System	Vytvoření nového klienta

Vyhledávání fyzických a právnických osob

Tab. 11: Scénář případu užití: Primární scénář: Vyhledávání fyzických a právnických osob

Název případu užití: Vyhledávání fyzických a právnických osob		
ID: UC004		
Charakteristika: Scénář pro vyhledávání fyzických a právnických osob		
Primární aktér: Administrátor		
Vedlejší aktéři: Nejsou		
<p>Vstupní podmínky:</p> <p><i>Administrátor</i>, který chce vytvořit nebo editovat klienta, musí být přihlášen.</p> <p>System musí umožnit administrátorovi vyhledat klienta podle IČO.</p> <p>Pro správné vyhledávání musí být zadané validní IČO.</p>		
Výstupní podmínky: Vyplnění údajů z databáze ARES		
Primární scénář:		
Krok	Aktér	Popis
1	Administrátor	Přihlášení uživatele s oprávněním <i>Administrátor</i>
2	Administrátor	Odeslání požadavku pro vytvoření nebo editaci klienta
3	System	Zobrazení formuláře pro vytvoření nebo editaci klienta
4	Administrátor	Zadání IČO
5	Administrátor	Žádost o vyhledání klienta pomocí IČO
6	System	Vyhledávání klienta pomocí IČO v databázi ARES
7	System	Klient nalezen
8	System	Vyplnění jména klienta
9	System	Vyplnění adresy nebo sídla klienta

<p>Alternativní scénáře:</p> <p>UC004a: Alternativní scénář pro krok 4 → Pokud uživatel zadá špatné IČO a nebude možné najít klienta, tak systém nevyplní žádné pole a vrátí chybovou hlášku → Tab. 12</p> <p>Primární a alternativní scénáře pro přihlášení uživatele jsou popsány v Tab. 7 a Tab. 8</p>

Tab. 12: Scénář případu užití: Alternativní scénář: Vyhledávání fyzických a právnických osob

Název – Alternativní scénář: Vyhledávání fyzických a právnických osob		
ID: UC004a		
Charakteristika: Zachycení případu, kdy bylo zadáno nesprávné IČO a nebylo možné najít klienta		
Alternativní scénář:		
Krok	Aktér/systém	Popis
1	System	V případě špatně vyplněného IČO, systém vrátí chybovou hlášku a může se pokusit o opětovné zadání čísla IČO
2	Administrátor	Zadání IČO
3	Administrátor	Žádost o vyhledání klienta pomocí IČO
4	System	Vyhledávání klienta pomocí IČO v databázi ARES
5	System	Klient nalezen
6	System	Vyplnění jména klienta
7	System	Vyplnění adresy nebo sídla klienta

Editace produktu

Tab. 13: Scénář případu užití: Primární scénář: Editace produktu

Název případu užití: Editace produktu
ID: UC010
Charakteristika: Scénář pro zachycení postupu při editaci klienta

Primární aktér: Administrátor		
Vedlejší aktéři: Nejsou		
Vstupní podmínky: <i>Administrátor</i> , který chce editovat produkt, musí být přihlášen. Systém musí umožnit administrátorovi editovat produkt.		
Výstupní podmínky: Úspěšná editace produktu		
Primární scénář:		
Krok	Aktér	Popis
1	Administrátor	Přihlášení uživatele s oprávněním <i>Administrátor</i>
2	Administrátor	Odeslání požadavku pro editaci produktu
3	System	Zobrazení formuláře pro editaci produktu
4	Administrátor	Úprava požadovaných dat produktu
5	Administrátor	Potvrzení editace produktu
6	System	Kontrola všech provedených změn
7	System	Editace informací o produktu v systému
8	System	Vytvoření záznamu změn o editaci produktu
Alternativní scénáře: UC010a: Alternativní scénář pro krok 4 → Pokud uživatel zadá nesmyslné údaje, např. Datum dodání je dřív než datum podpisu smlouvy, tak systém neprovede úpravy a vrátí chybovou hlášku → Tab. 14 Primární a alternativní scénáře pro přihlášení uživatele jsou popsány v Tab. 7 a Tab. 8		

Tab. 14: Scénář případu užití: Alternativní scénář: Editace produktu

Název – Alternativní scénář: Editace produktu
ID: UC10a
Charakteristika: Zachycení případu, kdy byly zadány špatné údaje.

Alternativní scénář:		
Krok	Aktér/systém	Popis
1	System	V případě špatně vyplněného vyplněných údajů, systém vrátí chybovou hlášku a <i>Administrátor</i> se může pokusit o opětovnou editaci produktu
2	Administrátor	Úprava požadovaných dat produktu
3	Administrátor	Potvrzení editace produktu
4	System	Kontrola všech provedených změn
5	System	Editace informací o produktu v systému
6	System	Vytvoření záznamu změn o editaci produktu

Vytvoření nového upozornění

Tab. 15: Scénář případu užití: Primární scénář: Vytvoření nového upozornění

Název případu užití: Vytvoření nového upozornění
ID: UC017
Charakteristika: Scénář postupu při vytvoření nového upozornění
Primární aktér: Administrátor
Vedlejší aktéři: Nejsou
Vstupní podmínky: <i>Administrátor</i> , který chce vytvořit nové upozornění, musí být přihlášen. Vytvoření upozornění je možné, pokud v systému existuje produkt a jsou vyplněny všechny termíny (datum podpis smlouvy, datum dodání, datum konce záruční doby HW nebo SW) Systém musí umožnit administrátorovi editovat produkt.
Výstupní podmínky: Vytvoření nového upozornění
Primární scénář:

Krok	Aktér	Popis
1	Administrátor	Přihlášení uživatele s oprávněním <i>Administrátor</i>
2	Administrátor	Odeslání požadavku pro vytvoření upozornění
3	System	Zobrazení formuláře pro vytvoření upozornění
4	Administrátor	Zadání povinných a nepovinných hodnot
5	Administrátor	Výběr typu upozornění
6	Administrátor	Potvrzení zadaných hodnot
7	System	Uložení nového upozornění do systému
Alternativní scénáře:		
Primární a alternativní scénáře pro přihlášení uživatele jsou popsány v Tab. 7 a Tab. 8		

Změna přístupového hesla

Tab. 16: Scénář případu užití: Primární scénář: Změna přístupového hesla

Název případu užití: Změna přístupového hesla		
ID: UC024		
Charakteristika: Scénář pro změnu přístupového hesla administrátora.		
Primární aktér: Administrátor		
Vedlejší aktéři: Nejsou		
Vstupní podmínky:		
Administrátor , který si chce změnit heslo, musí být přihlášen.		
System musí umožnit uživateli si kdykoliv změnit své přístupové heslo		
Heslo musí obsahovat alespoň 8 znaků, jedno malé a velké písmeno a číslici.		
Výstupní podmínky: Změna přístupového hesla		
Primární scénář:		
Krok	Aktér	Popis
1	Administrátor	Přihlášení uživatele do systému

2	Administrátor	Odeslání požadavku pro změnu přístupového hesla
3	System	Zobrazení formuláře pro změnu přístupového hesla
4	Administrátor	Zadání aktuálního přístupového hesla
5	Administrátor	Zadání nového přístupového hesla
6	Administrátor	Opětovné zadání přístupového hesla
7	Administrátor	Potvrzení zadaných údajů
8	System	Kontrola zadaných dat
9	System	Provedení změny přístupového hesla
Alternativní scénáře:		
UC024a: Alternativní scénář pro kroky 4-6 → Pokud jsou špatně zadány data ve formuláři, systém vyzve uživatele k opětovnému zadání → Tab. 17		
Primární a alternativní scénáře pro přihlášení uživatele jsou popsány v Tab. 7 a Tab. 8		

Tab. 17: Scénář případu užití: Alternativní scénář: Změna přístupového hesla

Název – Alternativní scénář: Změna přístupového hesla		
ID: UC024a		
Charakteristika:		
K alternativnímu scénáři může dojít v případě, že buď uživatel zadá špatné aktuální přístupové heslo, nebo se nová zadaná hesla neshodují. Kvůli bezpečnosti musí uživatel zadat všechny údaje znovu, i přes fakt, že ostatní údaje zadal správně.		
Alternativní scénář:		
Krok	Aktér/systém	Popis
1	System	System vyzve uživatele k opětovnému zadání údajů
2	Administrátor	Zadání aktuálního přístupového hesla
3	Administrátor	Zadání nového přístupového hesla
4	Administrátor	Opětovné zadání nového přístupového hesla
5	Administrátor	Potvrzení údajů

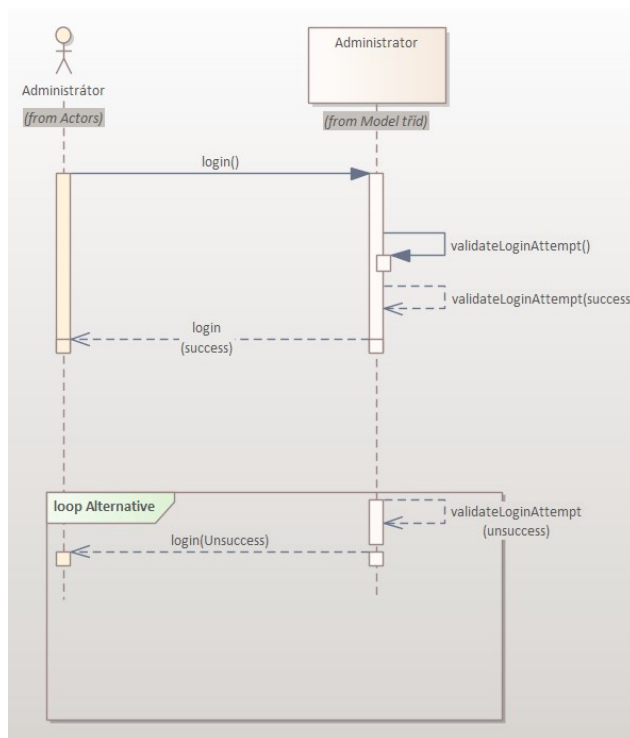
6	System	Kontrola správnosti údajů
7	System	Změna přístupového hesla

3.6 Sekvenční diagramy

Sekvenční diagramy [64] jsou formou interakčního diagramu, který zobrazuje objekty jako lifeline, tj. linie života, probíhající po stránce, přičemž jejich interakce v čase jsou reprezentovány jako zprávy znázorněné pomocí šipek od zdrojové lifeline k cílové lifeline. Sekvenční diagramy popisují, jakým způsobem objekty mezi sebou komunikují, jaké zprávy vyvolávají a nejsou určeny pro zobrazení složitých procedurálních logik. Sekvenční diagramy mimo jiné zachycují chování jednoho scénáře a ukazuje několik vzorových zpráv a objektů, které jsou předávány mezi sebou v rámci jednoho případu užití [65]. Pomocí sekvenčních diagramů se snažíme zobrazit potřebné třídy k vykonání daného případu užití. Pro každý scénář případu užití v předchozí kapitole je vytvořen sekvenční diagram.

3.6.1 Přihlášení uživatele

Přihlašování uživatele do systému je možné pouze v případě, že byl vytvořen administrátorem v systému. Tato funkcionalita je popsána na následujícím Obr. 44, primární a alternativní scénář byly popsány v Tab. 7 a Tab. 8.

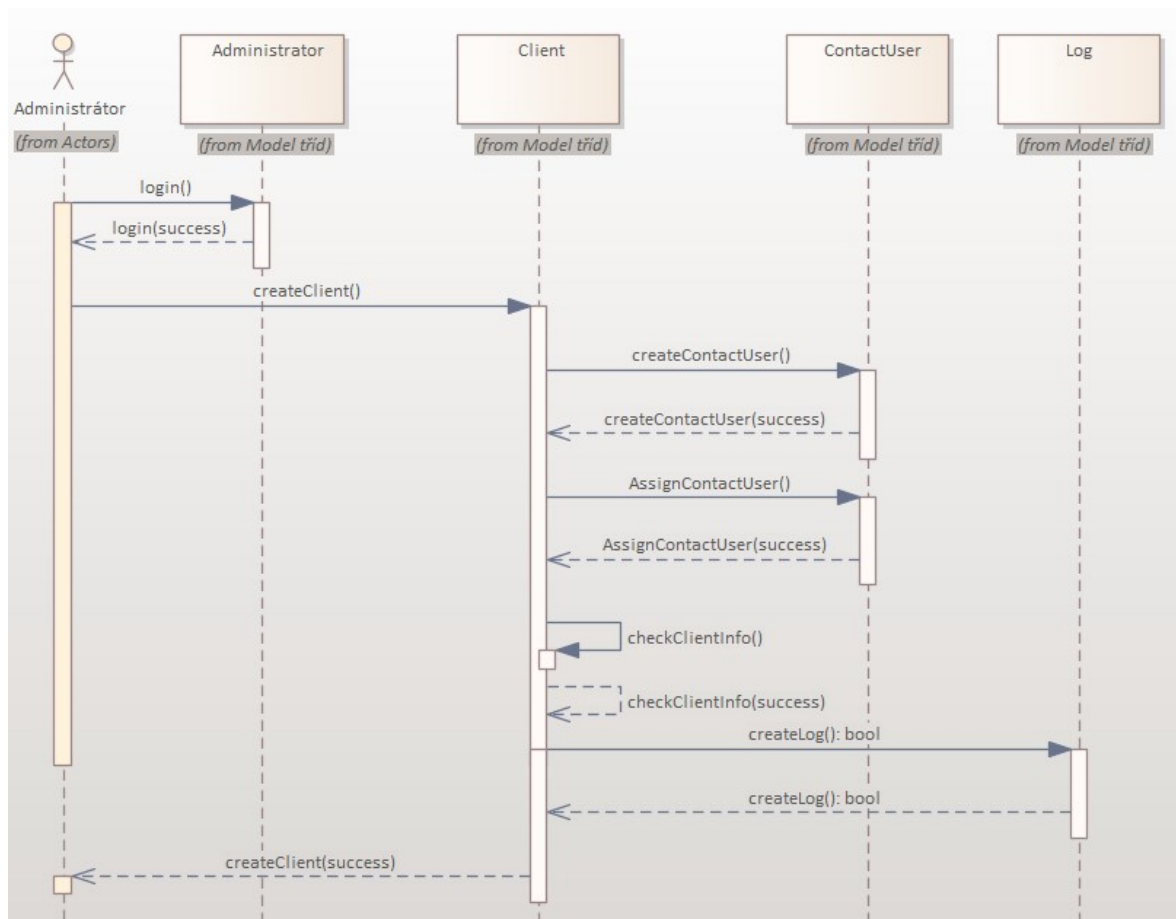


Obr. 44: Sekvenční diagram: Přihlášení uživatele

Pokud se chce *Administrátor* přihlásit tak musí vyplnit přihlašovací formulář, do kterých zadá e-mailovou adresu a přístupové heslo. Systém poté zkontroluje a spáruje zadaná data uživatelem s databází uživatelů. V případě shodných údajů je uživatel přihlášen do systému.

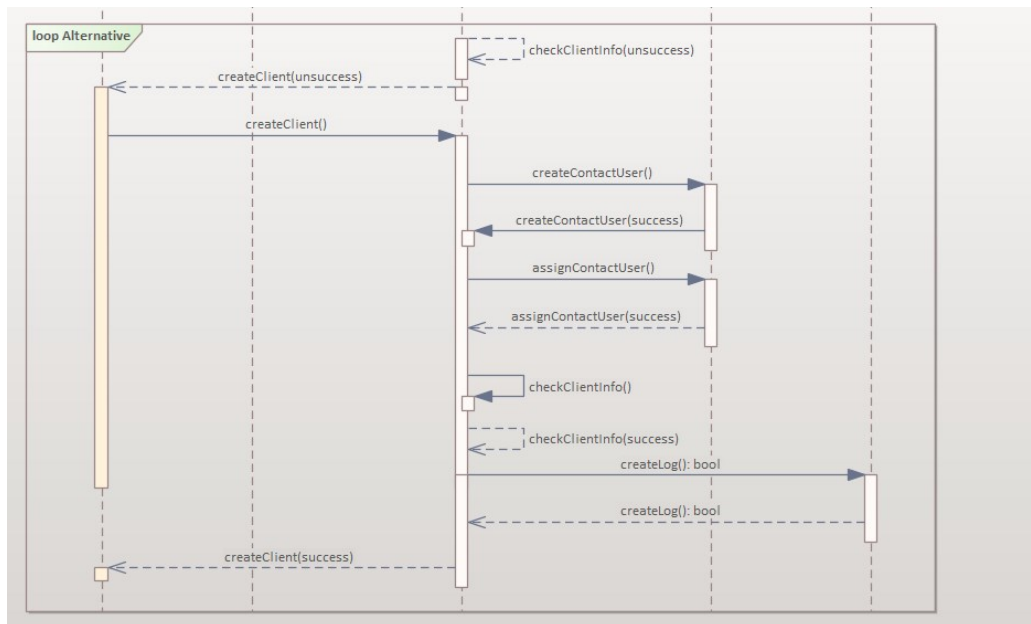
3.6.2 Vytvoření nového klienta

Hlavním aktérem je *Administrátor*, který chce vytvořit nového klienta. Po přihlášení si administrátor může otevřít formulář a vyplnit pole potřebné k vytvoření nového klienta. **Součástí musí být kontaktní údaje** (e-mailová adresa a telefonní číslo), bez kterých nelze vytvořit nového uživatele. V případě zadání čísla IČO musí být toto číslo unikátní a nesmí v systému existovat již existující klient se zadaným číslem a po vyplnění všech povinných a nepovinných polí a potvrzení zadaných dat dochází ke kontrole údajů systémem. Po úspěšné validaci zadaných dat je vytvořen nový klient. Sekvenční diagram je zobrazen na Obr. 45. Primární a alternativní scénáře jsou popsány v Tab. 9 a Tab. 10.



Obr. 45: Sekvenční diagram: Vytvoření nového klienta

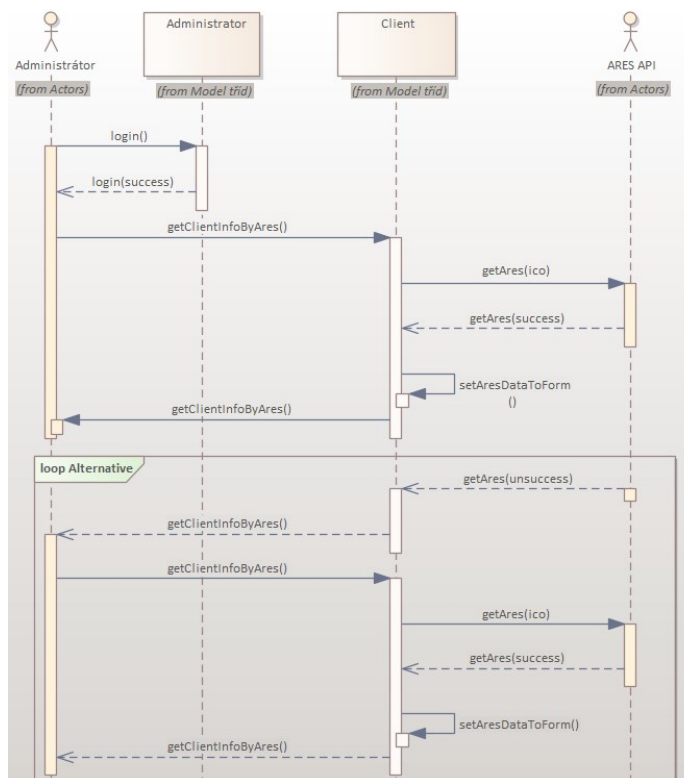
Pokračování sekvenčního diagramu pro alternativní scénář je popsán v následujícím Obr. 46.



Obr. 46: Sekvenční diagram: Vytvoření nového klienta – alternativní scénář

3.6.3 Vyhledávání fyzických a právnických osob

Pro usnadnění práce vyplňováním polí jako fakturační adresa, název klienta atd., tak vyhledávání fyzických a právnických osob může sloužit i pro ověření pravosti klienta. Sekvenční diagram je zobrazen na Obr. 47.

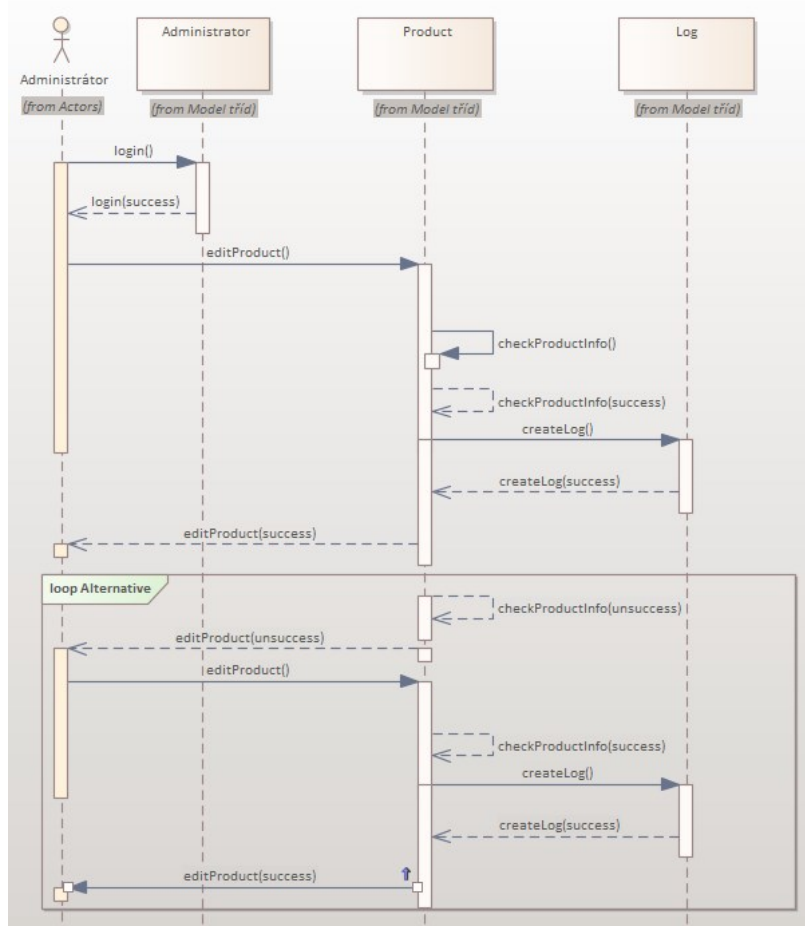


Obr. 47: Sekvenční diagram: Vyhledávání fyzických a právnických osob

Pro vyhledávání fyzických a právnických osob je potřeba zadat platné IČO, které má standardně osm čísel. Po vyplnění čísla IČO může **Administrátor** vyvolat funkci pro získání dat z databáze ARES a po úspěšném nálezů vyplní pole pro název klienta, fakturační adresu, PSČ atd. Primární a alternativní scénáře jsou popsány v Tab. 11 a Tab. 12.

3.6.4 Editace produktu

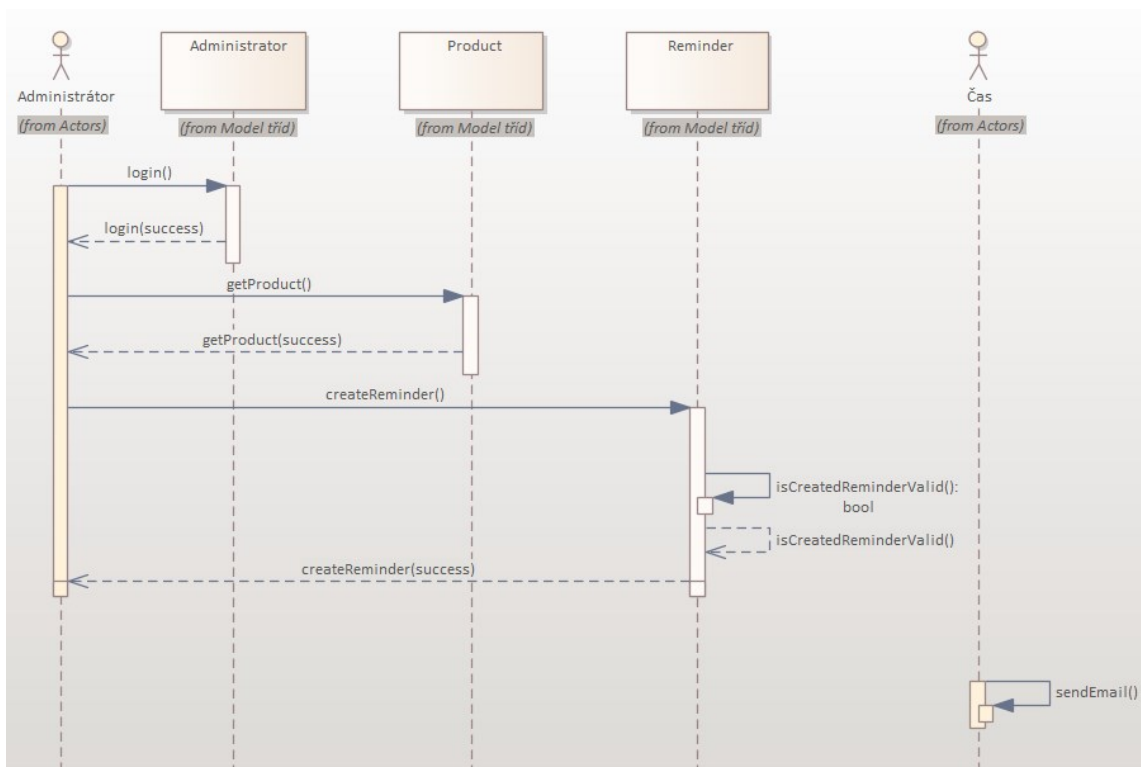
Po přihlášení **Administrátora** do systému je potřeba najít produkt, který je potřeba editovat a po zvolení produktu k editaci je administrátorovi zobrazen formulář pro editaci produktu. Po provedených změnách jsou data zkontrolovány, aby nedocházelo k ukládání chybných dat, jako např. datum dodání produktu je dřív než datum podpisu smlouvy apod. Sekvenční diagram pro editaci produktu je znázorněn na Obr. 48. Tab. 13 a Tab. 14 popisují primární a alternativní scénáře případu užití **Editace produktu**.



Obr. 48: Sekvenční diagram: Editace produktu

3.6.5 Vytvoření nového upozornění

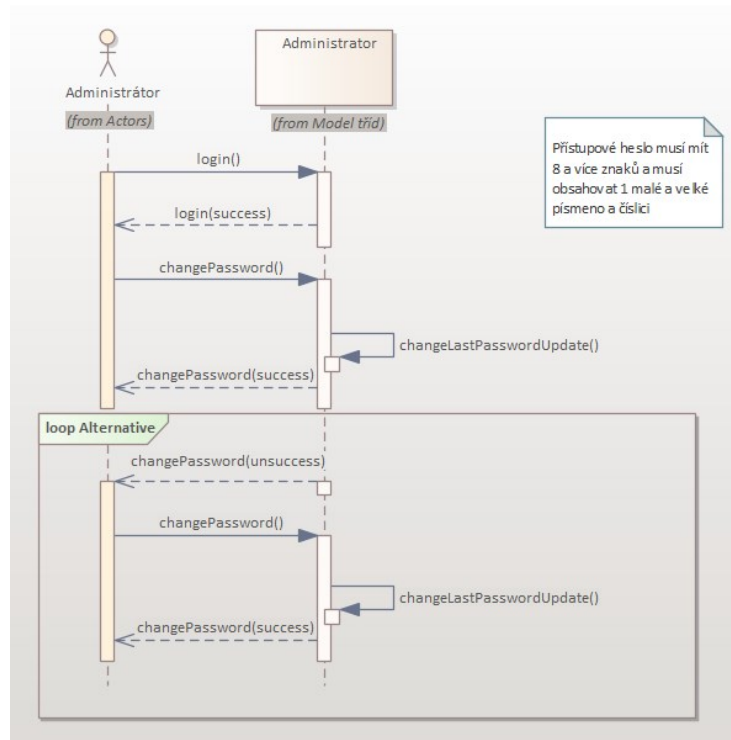
Dalším popisovaným sekvenčním diagramem je vytvoření nového upozornění. Primární scénář je popsán v Tab. 15 a nemá alternativní scénář. Aktéři tohoto sekvenčního diagramu jsou *Administrátor* a *Čas*, kdy *Administrátor* vytváří nové upozornění nad existujícím produktem a aktér *Čas* dohlíží na správné odesílání vytvořených upozornění. Tento diagram je zobrazen na Obr. 49.



Obr. 49: Sekvenční diagram: Vytvoření nového upozornění

3.6.6 Změna přístupového hesla

Poslední sekvenční diagram je pro změnu přístupového hesla. Každý přihlášený *Administrátor* si může kdykoliv změnit své přístupové heslo a nastavit si toto heslo jako trvalé, aby nedocházelo ke kontrole, zda vypršela platnost přístupového hesla. Každé přístupové heslo musí mít osm a více znaků a současně musí obsahovat alespoň jedno malé a velké písmeno a číslici. Bez splnění těchto kritérií vůbec nedojde k validaci zadaných hesel a nebude provedena žádná změna. Primární a alternativní scénáře jsou popsány v Tab. 16 a Tab. 17. Náhorná ukázka sekvenčního diagramu je na Obr. 50.



Obr. 50: Sekvenční diagram: Změna přístupového hesla

3.7 Drátěné modely (Wireframes)

Drátěné modely jsou vytvořeny pomocí nástroje Moqups [66]. Jedná se o online nástroj pro vytvoření drátěných modelů, diagramů, prototypů aplikace atd. Pro tvorbu je nutná registrace. Pro export je nutné mít zakoupenou licenci. V rámci analýzy aplikace jsou vytvořeny drátěné modely pro hlavní stránku, výpis produktů, detail klienta, detail produktu a také nastavení uživatelského účtu.

3.7.1 Úvodní strana

Prvním drátěným modelem je úvodní stránka administrace systému. V administraci je obsažena mapa, která zobrazuje pozici všech produktů v systému a na levé části obrazovky můžeme vidět menu společně s logem firmy. Součástí jsou veškerá vytvořená upozornění a také tabulka rychlých akcí, ve kterých je možné se rychle přepnout k jakémukoliv klientovi, produktu nebo kontaktní osobě. Drátěný model je zobrazen v příloze:

PŘÍLOHA PIX: DRÁTĚNÝ MODEL PRO ÚVODNÍ NÁSTĚNKU APLIKACE

3.7.2 Výpis produktů

Dalším drátěným modelem je výpis produktů a obsahuje jednoduchou tabulku, kde jsou zobrazeny všechny produkty v systému. Součástí této tabulky je vyhledávací pole, tlačítko pro přidání nového produktu a tlačítko pro filtrování produktů. Drátěný model je součástí přílohy:

PŘÍLOHA PX: DRÁTĚNÝ MODEL PRO VÝPIS PRODUKTŮ

3.7.3 Nastavení uživatelského účtu

Drátěný model pro nastavení uživatelského účtu obsahuje formulář, kde je možné editovat jméno a příjmení uživatele, nastavit si nový avatar nebo také možnost pro změnu přístupového hesla a deaktivace účtu. Tento drátěný model je součástí přílohy této práce:

PŘÍLOHA PXI: DRÁTĚNÝ MODEL PRO NASTAVENÍ UŽIVATELSKÉHO ÚČTU

3.7.4 Detail klienta

Následující drátěný model je zaměřený na detail klienta. Detail klienta obsahuje základní informace o klientovi, počínaje názvem, fakturační adresou, avatarem apod. Součástí tohoto detailu je i kontaktní osoba, vytvořené upozornění na produktech klienta, záznamy veškerých změn klienta a také výpis všech produktů klienta. Drátěný model je uveden v příloze:

PŘÍLOHA PXII: DRÁTĚNÝ MODEL PRO DETAIL KLIENTA

3.7.5 Detail produktu

Posledním drátěným modelem je detail produktu a součástí tohoto detailu jsou základní informace o produktu, např. název produktu, kategorie a typ produktu, poloha produktu, důležité termíny produktu, výdaje a výnosy, zaznamenané změny na produktech, nahrané soubory a přílohy a také mapa, která zobrazuje aktuální polohu produktu. Vytvořený drátěný model je v příloze:

PŘÍLOHA PXIII: DRÁTĚNÝ MODEL PRO DETAIL PRODUKTU

4 REALIZACE APLIKACE

Cílem této diplomové práce je vytvoření webové aplikace pro správu zákazníků a produktů. Hlavními produkty tohoto systému by měly být fyzické úřední desky, webové stránky pro zobrazení těchto desek, platební moduly atd. V rámci teoretické části jsou srovnány existující řešení v podobě softwarových platforem Asana a Freeloo a také redakční systémy Wordpress a Joomla. Výsledkem rešerše je vlastní realizace pomocí webové aplikace společně s analýzou vyvíjeného systému. Pro realizaci webové aplikace jsou použity technologie PHP a je využit framework CodeIgniter 4 díky širší znalosti tohoto frameworku na rozdíl od ostatních zmíněných frameworků a z pohledu budoucího vývoje a údržbě systému. Dále je použita technologie Javascript s využitím knihovny jQuery pro backendovou část aplikace a pro frontendovou část je použit framework Metronic a Bootstrap, zmíněné v kapitole 2 Rozbor použitých technologií.

4.1 Představení aplikace

Při implementaci interního systému pro správu zákazníků, resp. klientů a produktů je prvním krokem přihlašovací formulář, pomocí kterého je možné se dostat do správy systému. Vizualní vzhled a domovská stránka s přihlašovacím formulářem začíná na webové stránce:

{navez_domeny}/auth

Po otevření webové stránky výše je zobrazen přihlašovací formulář a po úspěšném přihlášení je zobrazena úvodní nástěnka systému. Vzhled této úvodní nástěnky je zobrazen v příloze:

PŘÍLOHA PXIV: VZHLED WEBOVÉ APLIKACE

Nástěnka této webové aplikace je inspirována drátěným modelem zmíněný v kapitole 3.7.1 Úvodní strana. Na nástěnce je vidět mapa zobrazující aktuální polohu všech produktů v systému, všechny vytvořené upozornění a také rychlé akce, které obsahují výpis všech produktů, klientů, kontaktních osob a současně lze v této části aplikace měnit stavy produktů. V levé části obrazovky je k dispozici menu, které obsahuje položky pro klienty, produkty, uživatele a nástroje. Položky **klienti** a **produkty** slouží pro zobrazení výpisu a možnost vytvoření nového klienta nebo produktu. V nástrojích jsou možnosti (výpis, vytvoření a editace) pro typy produktů, kategorie, upozornění, typy souborů a štítky produktů. Poslední částí jsou uživatelé, kde je možné zobrazit výpis všech uživatelů v systému a také vytvoření nových uživatelů, resp. administrátorů.

4.2 Vytvoření a editace klienta

Základem celého systému jsou klienti, neboť bez nich nelze vytvořit produkt. Vytvoření nového klienta je možné dvěma způsoby, kdy prvním způsobem je přes vyskakovací okno u výpisu klientů a druhým způsobem přes položku v menu **Vytvořit klienta**. Jedná se v podstatě o stejný typ formuláře, zobrazený na Obr. 51, akorát v jiném provedení a na jiných místech.

The image shows a web form for creating a new user. The form is organized into several sections:

- IČO:** A text input field labeled "Zadejte IČO". To its right is a toggle switch labeled "Aktivní" which is currently turned on, and a blue button labeled "ARES".
- Název:** A text input field labeled "Zadejte obchodní název klienta".
- E-mail:** A text input field labeled "Zadejte e-mail".
- Adresa / sídlo:** A section header followed by a text input field labeled "Zadejte ulici a číslo popisné".
- Město:** A text input field labeled "Zadejte město".
- Stát:** A dropdown menu labeled "Vyberte stát...".
- PSČ:** A text input field labeled "Zadejte poštovní směrovací číslo (PSČ)".
- Kraj:** A dropdown menu labeled "Vyberte okres...".
- Okres:** A dropdown menu labeled "Vyberte okres...".

At the bottom of the form, there are two buttons: "Zrušit" and "Uložit".

Obr. 51: Formulář pro vytvoření nového uživatele

Na Obr. 51 je znázorněn formulář pro vytvoření nového uživatele, který obsahuje textové pole pro zadání základních informací o klientovi, resp. obchodní název klienta, adresa nebo sídlo klienta a kontakt v podobě e-mailové adresy. Součástí tohoto formuláře je i textové pole pro zadání identifikačního čísla osoby (IČO) a tlačítko pro získání dat fyzických a právnických osob z databáze ARES, které je dostupné po vyplnění posledního čísla IČO, které má standardně osm čísel. Po úspěšném získání dat jsou vyplněny pole **DIČ** (pokud existuje), **Název**, **Ulice a č. p.**, **Město**, **Stát**, **PSČ**, **Kraj** a **Okres** a zbývá pouze doplnit e-mailovou adresu. Editace produktu probíhá pomocí stejného formuláře, kde jsou dynamicky předem vyplněná data z databáze a druhou variantou je kliknutím na tlačítko pro editaci u výpisu klientů u konkrétního klienta.

4.3 ARES

Jak bylo zmíněno v předchozí kapitole, systém obsahuje funkcionalitu pro získávání dat fyzických a právnických osob z databáze ARES. Prvním krokem jsou jednoduché validace pro zadání čísla IČO, kde jsou provedeny validace na délku celého vstupu a nahrazením všech nečíselných znaků za prázdný řetězec, díky čemuž docílíme, že nebude možné do textového pole vkládat písmena a symboly. S tímto řešením se nabízí pro zadání čísel využití textového pole typu *number*, ale může dojít k problému při ukládání identifikačního čísla osoby do databáze v případě, že bude toto číslo začínat číslicí 0. Tyto validace jsou znázorněny ve zdrojovém kódu, tj. Zdrojový kód 3, a po dosažení osmi číslic v textovém poli je zpřístupněno tlačítko pro získání dat z databáze ARES. Získávání dat je omezené na 10 000 dotazů od 10:00-18:00 nebo na 50 000 dotazů od 18:00-08:00 následujícího dne [67].

```
$(document).on("input", ".ico", function () {
    let icoInput = $('.ico');
    var ico = icoInput.val().trim();
    ico = ico.replace(/[^\d-9]/g, '');

    if(ico.length === 8)
    {
        $('#kt_ares_fill').removeAttr("disabled");
    }
    else{
        $('#kt_ares_fill').attr("disabled", true);
    }
    icoInput.val(ico);
});
```

Zdrojový kód 3: Validace vstupu uživatele – ARES

Po odeslání požadavku pro získání dat, resp. stiskem tlačítka, je provedena příprava dat a následné odeslání AJAX požadavku na url adresu *{nazev_domeny}/admin/klient/get_ares*, Zdrojový kód 4 níže popisuje odeslání požadavku a vyčkává na odpověď.

```
$('#kt_ares_fill').click(function()
{
    var ic = $('.ico').val();
    var base_url = $('#base_url').val();
    var state_select = $('#select_state');
    var region_select = $('#select_region');
    var district_select = $('#select_district');
    $('.ares_clicked').val("1");
    $.ajax(
    {
        url:base_url+'/admin/klient/get_Ares',
        method: 'post',
        data:{ic:ic},
        success: function(res, text)
        {
            var jsres = JSON.parse(res);
            if(jsres.stav !== "ok")
```

```

        {
            alert(jsres.stav);
            return;
        }
        else
        {
            jsres.ulice = jsres.ulice.trim();
            $(".client_name").val(jsres.firma);
            $(".address").val(jsres.ulice);
            $(".city").val(jsres.mesto);
            $(".postcode").val(jsres.psc);
            $(".dic").val(jsres.dic);

            changeSelectedOption(state_select, jsres.position.id_state);
            changeSelectedOption(region_select, jsres.position.id_region);
            changeSelectedOption(district_select, jsres.position.id_district);
        }
    },
    error: function(req, status, err)
    {
        alert('Došlo k chybě, kontaktujte prosím správce');
    }
});
});
function changeSelectedOption(element, value)
{
    element.find('option:selected').removeAttr("selected", "selected");
    element.find('option[value="'+value+'"]').attr("selected", "selected");
    element.trigger('change');
    element.attr("disabled", "disabled");
}

```

Zdrojový kód 4: Odeslání a zpracování AJAX požadavku

V případě získání validních, resp. korektních dat dochází k doplnění a aktualizaci jednotlivých polí zmíněné v předchozí kapitole a současně jsou zakázány jakékoliv úpravy select boxů pro výběr státu, kraje a regionu. Získání dat z databáze ARES [68] je řešeno pomocí skriptovacího jazyku PHP v následujícím zdrojovém kódu, resp. Zdrojový kód 5, pomocí cURL [69].

```

public function get_ares_ajax()
{
    $ico = $this->request->getPost('ico');
    $path = 'http://wwinfo.mfcr.cz/cgi-bin/ares/darv_bas.cgi?ico='.$ico;

    $xml = trim($this->sendXmlOverPost($path));
    if($xml) $xml = simplexml_load_string($xml);

    $a = array();
    if (isset($xml))
    {
        $ns = $xml->getDocNamespaces();
    }
}

```

```

$data = $xml->children($ns['are']);

$sel = $data->children($ns['D'])->VBAS;
if ((string)$sel->ICO == $ico)
{
    $a['ico'] = $this->request->getPost('ic');
    $a['dic'] = (string)$sel->DIC;
    $a['firma'] = (string)$sel->OF;

    $a['jmeno'] = "";
    $a['prijmeni'] = "";
    // detekce jména a firmy ..
    $firma = $a['firma'];
    $roz = explode(" ", $firma);
    $match = preg_match("/(s\.r\.o\|s\. r\. o\|spol\
s r\.o\|a\.s\.|a\. s\.|v\.o\|s|v\. o\ s\| o\ s\|k\s\|kom\ spol\.)/",
    $firma);
    if (count($roz) == 2 AND !$match)
    {
        // nenašli jsme shodu s firmou, pravděpodobně se jedná o
        živnostníka se jménem ..
        $a['jmeno'] = $roz[0];
        $a['prijmeni'] = $roz[1];
    }

    $a['ulice'] = (string)$sel->AA->NU;
    if (!empty($sel->AA->CO) OR !empty($sel->AA->CD))
    {
        // detekování čísla popisného a orientačního
        $a['ulice'] .= " ";
        if (!empty($sel->AA->CD)) $a['ulice'] .= $sel->AA->CD;
        if (!empty($sel->AA->CO) AND !empty($sel->AA->CD)) $a['ulic-
ce'] .= "/";
        if (!empty($sel->AA->CO)) $a['ulice'] .= $sel->AA->CO;
    }

    $a['mesto'] = (string)$sel->AA->N;
    $a['psc'] = (string)$sel->AA->PSC;
    $a['position'] = $this->clientModel->get_position_by_psc($sel-
>AA->PSC);
    $a['stav'] = 'ok';
}
else
{
    $a['stav'] = 'IČ firmy nebylo v databázi ARES nalezeno';
}
}
else
{
    $a['stav'] = 'Databáze ARES není dostupná';
}
echo json_encode($a);
}
function sendXmlOverPost($url)
{
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);

    // For xml, change the content-type.

```

```
curl_setopt ($ch, CURLOPT_HTTPHEADER, Array("Content-Type: applica-
tion/xml"));

curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$result = curl_exec($ch);
curl_close($ch);
return $result;
}
```

Zdrojový kód 5: Odeslání a zpracování požadavku cURL

Výsledek po úspěšném zpracování požadavku je znázorněn na následujícím Obr. 52.

IČO DIČ

Aktivní

Název *

E-mail *

Adresa / Sídlo

Ulice a č. p. *

Město

Stát * PSČ *

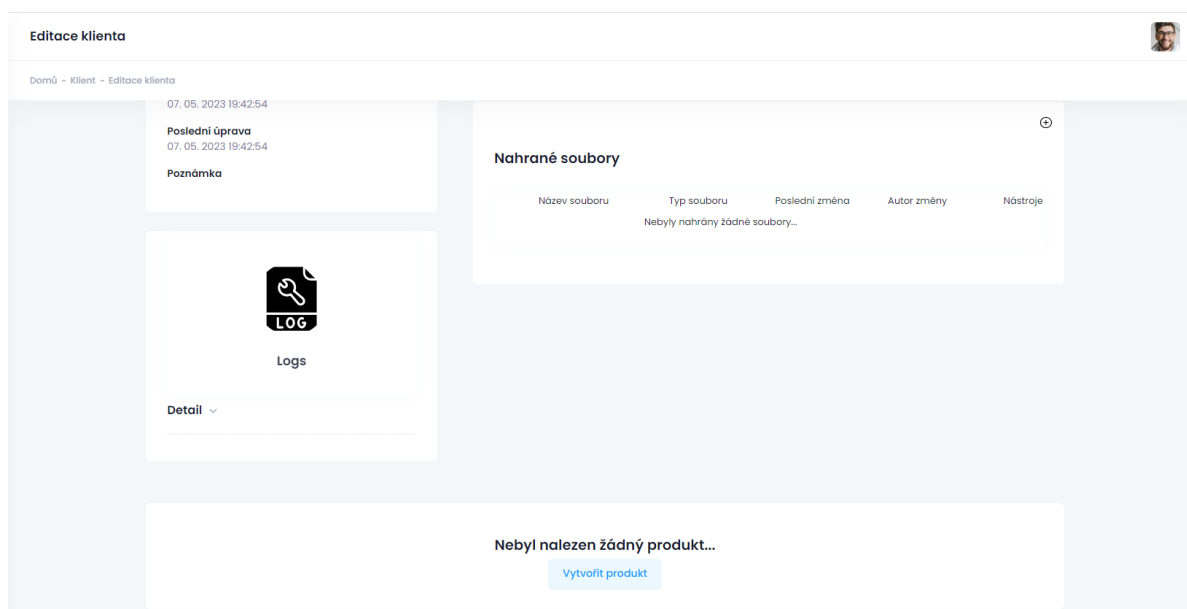
Kraj * Okres *

Obr. 52: Výsledek po úspěšném zpracování ARES požadavku

XML služba ARES [70] má vlastní dokumentaci a popis pro získávání dat z odpovědi typu XML (*eXtensible Markup Language*) [71], tj. obecně rozšiřitelný značkovací jazyk, standard pro výměnu dat. Po zpracování cURL požadavku, pomocí funkce **SendXmlOverPost** a získání odpovědi je potřeba tyto XML data rozklíčovat a vytáhnout z nich požadovaná data, které jsou následně poslány zpět do asynchronního volání AJAX k dokončení procesu vyplňování jednotlivých polí.

4.4 Vytvoření a editace produktu

Další důležitou součástí systému jsou právě produkty jednotlivých klientů, neboť produkty nelze zakládat bez předem vytvořeného klienta. Celkově jsou tři možnosti, jak vytvořit nový produkt do systému, kde dva způsoby jsou stejné jako u vytvoření klienta, prostřednictvím vyskakovacího formuláře u výpisu všech produktů v systému nebo přesměrováním na formulář z položky v menu. Dalším způsobem vytvoření produktu je přímo v detailu klienta, kde jsou zobrazeny základní informace klienta, kontaktní osoby, produkty klienta atd. a pod posledním produktem je možnost vytvořit nový produkt znázorněný na následujícím Obr. 53.



Obr. 53: Způsob přidání produktu do systému

Pro vytvoření nového produktu do systému je potřeba vyplnit všechny povinné pole, kterými jsou název produktu, typ produktu, klient, datum podpisu smlouvy a lhůta k dodání. Ostatní nepovinné pole je možné kdykoliv doplnit v editaci produktu a celkový vzhled formuláře je zobrazen na Obr. 54.

Nový produkt

Domů - Produkt - Nový produkt

Informace o produktu

Název produktu *	Aktivní
<input type="text" value="Zadejte název produktu"/>	<input checked="" type="checkbox"/>
Vyberte typ produktu *	Vyberte klienta *
<input type="text" value="Vyberte typ produktu..."/>	<input type="text" value="Vyberte klienta..."/>
Adresa HW	Adresa SW
<input type="text" value="Zadejte adresu HW"/>	<input type="text" value="Zadejte adresu SW"/>
Popis produktu	
<input type="text" value="Zadejte popis produktu"/>	

Termíny

Datum podpisu smlouvy *	Lhůta k dodání *
<input type="text" value="dd.mm.rrrr"/>	<input type="text" value="Zadejte lhůtu k dodání (počet dní)"/>
Datum dodání	
<input type="text" value="dd.mm.rrrr"/>	
Délka záruky SW	Délka záruky HW
<input type="text" value="Vyberte délku záruky SW..."/>	<input type="text" value="Vyberte délku záruky HW..."/>

Zrušit

Obr. 54: Formulář pro přidání nového produktu

Při vytvoření produktu není k dispozici zadání polohy produktu a je možné tuto polohu doplnit až při editaci produktu, neboť se počítá s tím, že nový produkt při podepsání smlouvy ještě není fyzicky nainstalovaný nebo předaný.

4.5 Systém upozornění

V tomto systému je implementován jednorázový způsob upozornění na jednotlivé typy upozornění jako např. přesné datum upozornění, záruka hardwarového a podpora softwarového řešení. Upozornění lze vytvářet pouze u konkrétního klienta ke konkrétnímu produktu jako na Obr. 55, neboť může v systému docházet k ukládání stejných typů produktů pro různé klienty a tímto způsobem nedojde k záměně.

Editace klienta

Domů - Klient - Editace klienta

Informace o produktu Upravit produkt Smazat produkt

Název produktu: Outdoor 56 Typ produktu: OUTDOOR

Kategorie: Digitální úřední deska Adresa HW

Latitude: Adresa SW

Longitude: Poznámka

Status: **Aktivní** Stav: Vyzvednuto v PT

Termíny

DATUM PODPISU SMLOUVY	LHŮTA K DODÁNÍ	DATUM DODÁNÍ	ZÁRUKA HW	ZÁRUKA SW	NÁSTROJE
15. 03. 2023	60 - 14. 05. 2023	15. 04. 2023	14. 04. 2024 / 342 dní	14. 04. 2024 / 342 dní	✎ ✖

Přidat upozornění

Náklady a výnos

Typ	Produkt	Kotva	Doprava	Dodání	Ostatní	Celkem	Nástroje
Náklady	10.00 Kč	10.00 Kč	10.00 Kč	10.00 Kč	10.00 Kč	50.00 Kč	✎ ✖
Zisk	20.00 Kč	20.00 Kč	20.00 Kč	50.00 Kč	15.00 Kč	125.00 Kč	✎ ✖
Rozdíl	10.00 Kč	10.00 Kč	10.00 Kč	40.00 Kč	5.00 Kč	75.00 Kč	Smazat vše

Obr. 55: Editace klienta – možnost přidání upozornění

Upozornění pro hardwarová a softwarová řešení lze různě modifikovat, pomocí vytvoření nových typů upozornění, kde je možné si různě měnit dobu před vypršením záruky pro HW a SW řešení, neboť je možné zadávat počet dní a systém vypočítá plánované datum upozornění pomocí javascriptové knihovny MomentJs [67] jako na Obr. 56.

Přidat upozornění ✕

Druh upozornění *

30 dní do konce HW záruky ▼

Možnosti

Přesné datum

Záruka HW

Podpora SW

Počet dní do konce podpory HW

90 ▾

90 dní do konce záruky: 15. 01. 2024

Poznámka k upozornění

Zadejte poznámku k upozornění ✎

Další příjemci

██████████

██████████

Zrušit Uložit

Obr. 56: Vytvoření nového upozornění

Podle Obr. 56 je vidět, že k upozornění dojde dne 15.01.2024 prostřednictvím e-mailové zprávy vybraným příjemcům (administrátorům), že zbývá 90 dní do konce hardwarové podpory. E-mailové zprávy [73] jsou rozesílány pomocí SMTP [74] (*Single Mail Transfer Protocol*), tj. protokol pro posílání a přijímání e-mailů přes internet.

4.6 Automatické odhlášení uživatele

Z pohledu bezpečnosti je v systému implementována funkce pro kontrolu neaktivity přihlášeného uživatele. Tato doba neaktivity je nastavena na 1 200 000 ms, tj. 20 minut a upozornění na případné odhlášení je zobrazeno po pěti minutách neaktivity. Toto upozornění je zobrazeno pomocí tzv. toasterů [75] a během zbývajících 15 minutách probíhá odpočet, po kterém dojde k odhlášení uživatele. Zdrojový kód 6 [76], zobrazuje způsob nastavení intervalu pro výpočet zbývajících času a přepočítání na minuty a sekundy.

```
setInterval(function(){
    var remain = (parseFloat($(document).idleTimer("getRemainingTime")/1000)).toFixed(0);
    var minutes = parseInt((parseFloat($(document).idleTimer("getRemainingTime")/1000)).toFixed(0)/60);
    var difference = ($(document).idleTimer("getRemainingTime").toFixed(0)) - (minutes * 60000);
    difference = parseInt(difference / 1000);

    if($(document).idleTimer("getRemainingTime") <= showRemainTime){
        if(remain < 60){
            $('#elapsed').html("<p class='alert alert-danger'>Byla zaznamenána dlouhá nečinnost! pokud neprojevíte aktivitu budete automaticky odhlášen!. Odhlášení za: <span class='timesystemcas' style='font-weight: 600;width: 114px;'>"+remain+"s</span>");
        }
        else{
            $('#elapsed').html("<p class='alert alert-danger'>Byla zaznamenána dlouhá nečinnost! pokud neprojevíte aktivitu budete automaticky odhlášen!. Odhlášení za: <span class='timesystemcas' style='font-weight: 600;width: 114px;'>"+minutes+"m "+ difference+" s</span>");
        }
    }
    if($(document).idleTimer("getRemainingTime")==0){
        location.href = HOST_URL + "/admin/logout";
    }
},
);
```

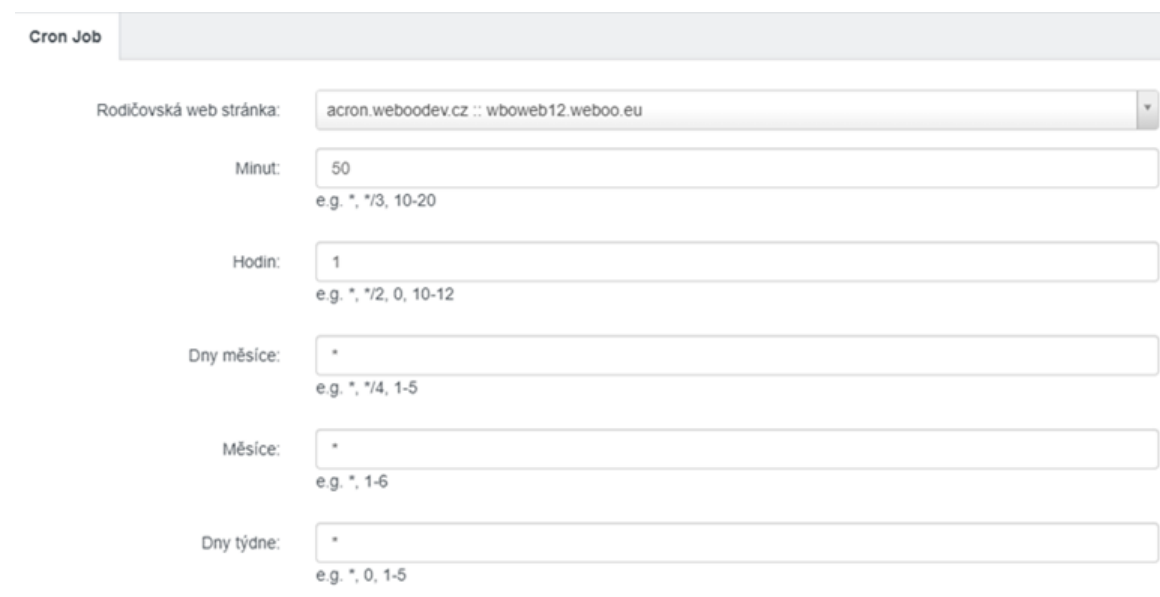
Zdrojový kód 6: Nastavení intervalu nečinnosti uživatele

4.7 IonAuth

Součástí tohoto systému je knihovna IonAuth od autora Ben Edmunds [77] pod licencí MIT, která slouží pro autentizaci uživatelů např. při přihlašování uživatelů do systému a je kompatibilní s frameworkem CodeIgniter4. MIT licence [78] nemá žádná omezení na úpravy, používání, sdílení atd. IonAuth slouží pro autentizaci uživatelů při přihlášení, kontrolu oprávnění, vytváření hash hesel, resp. zašifrování hesel. Pro šifrování hesel je možné využít šifrovací metody jako bcrypt, argon2 a argon2id.

4.8 Cron

Pravidelné úkoly, volání funkcí jsou v tomto systému řešeny pomocí Cronů [79]. Cron (*softwareový démon*) je v podstatě plánovač procesů, který umožňuje spouštět příkazy, skripty a programy podle předem zadaných plánů. Samotné nastavení je zobrazeno na Obr. 57.



Obr. 57: Nastavení Cronu pro kontrolu a odeslání upozornění

V tomto systému jsou Crony využívány pro automatické kontroly upozornění plánované na daný den. Obr. 57 výše definuje způsob definování cronu a je nastaven na každý den v roce vždy v 01:50 h, tzn. že se vždy v tento čas spustí vyhledávání všech upozornění plánované na daný den a spustí se odesílání hromadného upozornění prostřednictvím e-mailové zprávy. Pro odesílání e-mailových zpráv je využita knihovna **PHPMailer** [80].

4.9 Mapy

Nejzajímavější funkcionalitou tohoto systému jsou právě operace s mapou. V tomto systému jsou využité **Mapy API** [81] od společnosti **Seznam**, **mapy** od **společnosti Google** [82] a využití javascriptové knihovny **Leaflet** [83].

4.9.1 API Mapy.cz

Mapy API jsou dostupné zdarma pro komerční účely, má velmi přehlednou a detailní dokumentaci s příklady použití, nemají žádná omezení pro počet požadavků a je využívána pro vyhledávání polohy produktu pomocí našeptávače [84] nebo pomocí kliknutí do mapy, přičemž jsou získány data o zeměpisné šířce a délce. Mapa je volána asynchronním způsobem [85] a samotná inicializace reprezentuje následující Zdrojový kód 7.

```
$(document).ready(function() {

    Loader.async = true;
    Loader.load(null, {suggest: true}, initMapApiCZ);
});
function initMapApiCZ() {

    var latitude = parseFloat($("#input[name=lat]").val());
    var longitude = parseFloat($("#input[name=lot]").val());
    var center = '';
    var title = $('#address_hw').val();

    /**
     * Nastavení středu mapy => pokud není, tak střed Prahy
     */

    if(longitude > 0 && latitude > 0)
    {
        center = SMap.Coords.fromWGS84(longitude, latitude);
    }
    else{
        title = "Střed České republiky";
        center = SMap.Coords.fromWGS84(14.4341412988, 50.0835493857);
    }
    var m = new SMap(JAK.gel("m"), center, 15); // vygeneruj mapu na da-
ných souřadnicích

    m.addDefaultLayer(SMap.DEF_BASE).enable();
    m.addDefaultControls();
```

Zdrojový kód 7: Asynchronní načtení mapy a nastavení středu mapy

Zdrojový kód 7 popisuje způsob inicializace načtení asynchronní mapy a současně je zde nastaven střed mapy, který se orientuje podle uložených hodnot z databáze. V případě chybějících hodnot je střed mapy nastaven na střed města Prahy. **API Mapy.cz** je využit pro zadání přesné polohy produktu a je celkově možné tuto polohu zvolit třemi způsoby a prvním z nich je kliknutím do libovolného místa na mapě. Systém po detekci kliknutí uživatelem

zjistí polohu, resp. zeměpisnou šířku a délku společně s adresou polohy. Další možností je přetáhnutí značky na libovolné místo (pouze Česko a Slovensko) a poslední možností je zadáním adresy pomocí vyhledávacího pole (našeptávače), ve kterém je možné zadávat název města, ulice nebo také zastávky autobusu. Zdrojový kód 8 zobrazuje nastavení parametrů našeptávače [86].

```
var suggest = new SMap.Suggest(inputEl, {
  provider: new SMap.SuggestProvider({
    updateParams: params => {
      params.enableCategories = 1,
      params.lang = "cs",
      params.limit = 50,
      params.count = 50,
      params.locality = "cz|sk",
      params.pubtBoost = 1, //zapni i zastávky
      params.zoom = 20,
      params.langFilter = "cs",
      params.category = "municipa-
lity_cz,address_cz,street_cz,traffic_cz,trafficpoi_cz,municipa-
lity_sk,address_sk,street_sk,traffic_sk,trafficpoi_sk"
    })});
```

Zdrojový kód 8: Nastavení parametrů našeptávače

Dalším zmíněným způsobem je kliknutí uživatelem do mapy a **využití funkce reverzního geokódování**, kdy nám po kliku **API Mapy.cz** asynchronně vrátí odpověď, co přesně je na daném bodu [87]. Odpověď nám vrátí zeměpisnou šířku a délku a také celou adresu, kterou potom pouze vyplníme do textových polí ve formuláři. Následující Zdrojový kód 9 popisuje implementaci detekci kliku do mapy, získání dat a vyplnění příslušných polí.

```
m.getSignals().addListener(window, "map-click", click);
/**
 * Po kliknutí => získání hodnoty
 */
function click(e, elm) {
  var coords = SMap.Coords.fromEvent(e.data.event, m);
  new SMap.Geocoder.Reverse(coords, result);
  m.setCenter(coords, true);
}
var result = function(geocoder) {

  var results = geocoder.getResults();

  var address=results.label;

  l.removeAll();

  var options = {
    title: address,
  };
  var marker = new SMap.Marker(SMap.Coords.fromWGS84(re-
sults.coords.x,results.coords.y), "myMarker", options);
  marker.decorate(SMap.Marker.Feature.Draggable);
  l.addMarker(marker);
```

```
$( 'input[name=address]' ).val (address);
$( '#mapyapisearch' ).val (address);
$( '.address_hw' ).val (address);
$( 'input[name=lat]' ).val (parseFloat (results.coords.y));
$( 'input[name=lot]' ).val (parseFloat (results.coords.x));
}
```

Zdrojový kód 9: Detekce kliku a získání dat o poloze

Poslední variantou získání přesné polohy je přetáhnutí značky v mapě pomocí myši, resp. Drag and drop [88]. Tato funkce je inicializována pomocí následujícího kódu:

```
marker.decorate (SMap.Marker.Feature.Draggable);
```

V této funkcionalitě je potřeba využít signálů, resp. události pro začátek a konec tažení, kdy jsou na konci tažení získány hodnoty zeměpisné šířky a délky. Zdrojový kód 10 zobrazuje způsob implementace pro přidávání a zachycení události tažení značky a po konci tohoto tažení jsou hodnoty zeměpisné šířky a délky využity pro získání adresy této polohy využitím reverzního kódování.

```
var signals = m.getSignals();
signals.addListener(window, "marker-drag-stop", stop);
signals.addListener(window, "marker-drag-start", start);
function start(e) { /* Začátek tažení */

    var node = e.target.getContainer();
    node[SMap.LAYER_MARKER].style.cursor = "move";
}
function stop(e) { /* Konec tažení */
    var node = e.target.getContainer();
    node[SMap.LAYER_MARKER].style.cursor = "";
    var coords = e.target.getCoords();
    new SMap.Geocoder.Reverse(coords, result);
    m.setCenter(coords, true);
}
```

Zdrojový kód 10: Nastavení signálů pro detekci začátku a konce tažení značky

4.9.2 Google Maps

Další využívanou mapou v systému je od společnosti **Google** [89] a je využita u detailu produktu, protože může být využita funkce *StreetView*, tj. zobrazení místa podle skutečnosti, pro lepší představu místa, kde má být umístěn produkt. **Pro využívání map od společnosti Google je potřeba mít vytvořený API klíč pro autorizaci a je zde omezený počet požadavků.** Při zobrazení mapy pomocí jazyku Javascript je tento počet stanoven zhruba na 18 000 požadavků měsíčně a může se lišit vzhledem k požadavkům ke *StreetView* [90] nebo jiných možnostech (s využitím stejného API klíče) a celková cena za tyto požadavky nesmí

přesáhnout 200 € měsíčně, tj. 4 655 Kč. K zobrazení mapy od společnosti Google jsou potřeba pouze hodnoty zeměpisné šířky a délky, pomocí kterých je nastaven střed mapy. Zdrojový kód 11 popisuje způsob inicializace mapy, nastavení středu mapy a vytváření značek.

```
var latitude = document.getElementById("latitude-hw").value;
var longitude = document.getElementById("longitude-hw").value;
var product_name = document.getElementById('product_name').value;
function initMap() {
    var latlng = new google.maps.LatLng(latitude, longitude);
    var myOptions = {
        zoom: 12,
        center: latlng
    };
    var map = new google.maps.Map(document.getElementById("map"), myOptions);
    var myMarker = new google.maps.Marker({
        position: latlng,
        map: map,
        title: product_name
    });
}
```

Zdrojový kód 11: Inicializace mapy od společnosti Google

4.9.3 Leaflet

Poslední variantou mapy je využití javascriptové knihovny **Leaflet** [91], která nabízí možnost přidávat více značek do mapy, přidávat popisky apod. Tato mapa je zobrazena na úvodní nástěnce administrace jako v příloze:

PŘÍLOHA PXIV: VZHLED WEBOVÉ APLIKACE

A jak již bylo zmíněno, tak zobrazuje polohy všech produktů v systému, resp. ty, které mají definovanou zeměpisnou délku a šířku. Následující Zdrojový kód 12 popisuje implementaci mapy, přidání panelu pro zvětšení mapy a vytváření značek.

```
var map = L.map('map', {
    center: [49.7437989, 15.3386383],
    zoom: 7,
    zoomControl: false,
    scrollWheelZoom: false,
});
L.control.zoom({
    position: 'topright'
}).addTo(map);

for(var i=0; i< markers.length; i++){
    marker = new L.marker([markers[i].latitude, markers[i].longitude])
        .bindPopup("<p1>" +
            "<a target=\"_blank\" href=\""+base_url+"/admin/klient/upravit_klienta/"+markers[i].id_client+"\">"+markers[i].client_name+"</a><br>" +
```

```

    "<a target=\"_blank\" href=\""+base_url+"/admin/pro-
dukt/detail/"+ markers[i].id_product+"\">"+markers[i].name+"</a><br>Po-
loha: " +markers[i].latitude +", "+ markers[i].longi-
tude+"</p1>").addTo(map);
}

```

Zdrojový kód 12: Inicializace mapy pomocí Leaflet

4.10 Vytvoření štítků a přiřazení ke kategorii

Další zajímavou funkcionalitou je vytvoření štítků, resp. stavů, a následné přiřazení k příslušné kategorii, kde je možné upravovat posloupnost těchto stavů. V následujícím Obr. 58 jsou zobrazeny všechny možné stavy pro kategorii **Digitální úřední deska**.

#	NÁZEV	ŠTÍTEK	POŘADÍ	NÁSTROJE
1	Poslaná nabídka	Náhled	1 +	■
3	Podepsána SOD	Náhled	– 2 +	■
4	Zaplacena záloha výrobcí	Náhled	– 3 +	■
5	Vyzvednuto v PT	Náhled	– 4 +	■
6	Složeno ve skladě	Náhled	– 5 +	■
7	Nainstalováno	Náhled	– 6	■

[Přidat štítek](#)

Obr. 58: Výpis štítku kategorie – Digitální úřední deska

Každý stav má své vlastní pořadí, které je možné měnit, vlastní vzhled štítku pro pozadí a písmo štítku, které lze měnit pomocí barevné palety. Současně je možné přidávat nové štítky ke kategorii, které již byly vytvořeny a jsou ve výpisu všech štítků, definovat pořadí u dané kategorie, kdy systém dopočítá rozdílovou posloupnost a automaticky přepíše celou posloupnost stavů. Na následujícím Obr. 59 je zobrazen způsob editace štítku, kde je možné měnit název štítku, upravovat barvy pozadí a písma.

Editace štítku

Text štítku *

Zaplacena záloha výrobci

Vyberte barvu štítku *

Vyberte barvu písma *

Náhled štítku

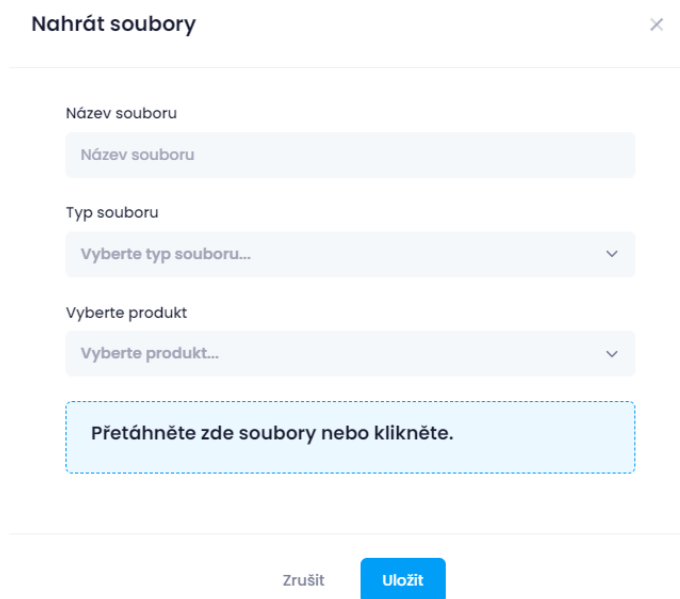
Náhled

Zrušit Uložit

Obr. 59: Editace vytvořeného štítku

4.11 Nahrávání souborů a příloh

Před instalací nebo vydáním nějakého produktu může dojít k podpisu smlouvy, souhlasu o zpracování osobních údajů atd. Pro uchování těchto souborů a nalezení na správném místě je možné tyto soubory najít hromadně u klienta anebo také přímo u detailu produktu. Přidávání a editace těchto souborů je možné na obou místech, kdy při přidání u klienta je možné zvolit k jakému produktu má být přiřazené a o jaký typ souboru se jedná. Tyto typy souborů je možné přidávat a upravovat. Nahrávání souborů je možné pomocí přetáhnutí souboru do pole pomocí **Dropzone** [92] (pomocí Drag&Drop) nebo pomocí výběru z adresáře počítače. Pro náhled obrázků je využita javascriptová knihovna **Lightbox2** a pro ostatní soubory je využit HTML element **iframe**. Pro nahrávání souboru je vytvořen formulář, kde je potřeba zadat název souboru, zvolit o jaký typ souboru se jedná a ke kterému produktu má být přiřazen, jako na Obr. 60.



Nahrát soubory

Název souboru

Název souboru

Typ souboru

Vyberte typ souboru...

Vyberte produkt

Vyberte produkt...

Přetáhněte zde soubory nebo klikněte.

Zrušit Uložit

Obr. 60: Formulář pro nahrávání souboru

Při nahrávání avatarů klienta, vytváření nových typů souborů jsou vytvořeny kopie menšího obrázku [93] pro zobrazení ve zmenšené verzi, aby se nemusely stahovat obrovské soubory. Obrázky mohou být zmenšeny ručně, ale velikost staženého souboru se nemění. Pro vytvoření menších kopií je využita PHP knihovna GD [94].

4.11.1 Dropzone

V předchozí kapitole byla zmíněna funkcionální **Dropzone** [92], která slouží pro nahrávání souboru pomocí přetažení souboru do určitého pole. Následující Zdrojový kód 13 demonstruje způsob nastavení funkcionality **Dropzone**.

```
var dropzone_1 = function () {
    $('#product_files_dropzone').dropzone({
        autoProcessQueue: false,
        url: $('#kt_modal_add_files_form').attr("action"),
        paramName: "product_image",
        maxFiles: 10,
        maxFileSize: 10,
        parallelUploads: 10,
        addRemoveLinks: true,
        dictRemoveFile: "Odstranit",
        dictDefaultMessage: "Klikněte nebo přetáhněte soubory zde.",
        acceptedFiles: '.jpeg, .jpg, .png, JPEG, .JPG, .PNG, .pdf',
        init: function() {
            var myDropzone = this;

            $('#kt_modal_add_files_submit').click(function (e) {
                e.preventDefault();
                myDropzone.processQueue();
            });
            this.on('sending', function(file, xhr, formData) {
```

```

        var data = $('form').serializeArray();
        $.each(data, function(key, el) {
            formData.append(el.name, el.value);
        });
    });
}
});
};

```

Zdrojový kód 13: Nastavení funkcionality Dropzone

Zdrojový kód 13 popisuje způsob nastavení základního nastavení, resp. ukazatel na vstupní pole při nahrávání obrázku, koncovou adresu pro vykonání funkce nahrávání, jednotlivé texty, jaké typy a počet povolených souborů atd. Současně jsou zachyceny události při kliku na potvrzení formuláře a spuštění fronty pro nahrávání obrázků.

4.11.2 Lightbox2

V kapitole 4.11 Nahrávání souborů a příloh byly zmíněny způsoby pro zobrazení obrázků a jiných typů souborů. Pro obrázky je využita javascriptová knihovna **Lightbox2** [95], která je využívána [96] pro vytváření galerií, zobrazení obrázků, prezentace atd. Pro správnou funkcionality této knihovny je potřeba mít funkční knihovnu jQuery a vytvoření odkazů pomocí elementu **a**, kde je cílem odkazu cesta k větší variantě obrázku, která bude zobrazena po rozkliknutí a uvnitř elementu **a** bude menší varianta obrázku definovaná pomocí elementu **img**, která je zobrazena před rozkliknutím. Následující Zdrojový kód 14 popisuje způsob nastavení odkazů pro zobrazení obrázku pomocí knihovny **Lightbox**.

```

<a title="Zobrazit náhled" href="<?=( !empty($file->path)) ?
base_url($file->path) : base_url($file->path_thumb);?>" data-light-
box="<?=$file->id_file_type;?>" data-title="<?=$file->name." - ".$file-
>file_name;?>" class="text-dark fw-bolder text-hover-primary mb-1 fs-6
btn_file_preview">
    
</a>

```

Zdrojový kód 14: Způsob nastavení odkazů pro zobrazení obrázku pomocí knihovny Lightbox

4.12 CRUD operace

V této webové aplikaci jsou nejvíce využívány **CRUD** operace (*Create, Read, Update, Delete*). Tyto operace jsou použity pro vytváření, čtení, editaci a mazání záznamů z databáze. Vkládání záznamu do databáze reprezentuje **SQL** příkaz *INSERT*, pro čtení se využívá *SELECT*, pro editaci *UPDATE* a pro mazání záznamů *DELETE*. Tyto operace jsou provedeny pomocí třídy **Query Builder**, která bude popsána v následující části práce.

Všechny SQL dotazy jsou psány v **Modelech**, které dědí od třídy **CoreModel**, kde je v konstruktoru navázáno spojení s databází. V konstruktoru těchto modelů jsou definovány názvy tabulek společně s aliasem. Zdrojový kód 15 popisuje způsob definování tabulek, které budou využity v tomto modelu.

```
public function __construct()
{
    parent::__construct();
    $this->builder = $this->db->table("file as f");
    $this->builder_type = $this->db->table("file_type as ft");
    $this->builder_client = $this->db->table("client as c");
}
```

Zdrojový kód 15: Nastavení tabulek v konstruktoru třídy FileModel

Výsledek dotazovacího příkazu *SELECT* je možné získat v různých formátech, jako např. pole objektů *getResultObject()*, jediný řádek objektu *getRowObject()*, získání prvního nebo posledního řádku *getFirstRow()* a *getLastRow()* atd. Následující Zdrojový kód 16 popisuje způsob získání hodnot sloupce *path* pro konkrétního klienta.

```
public function get_client_files($id)
{
    $this->builder->select("path")
        ->where("id_client", $id);
    return $this->builder->get()->getResultObject();
}
```

Zdrojový kód 16: SQL příkaz SELECT pro získání cesty k souborům konkrétního klienta

Dalším příkladem je vkládání nových záznamů do databáze. Před vložením záznamu je potřeba mít nachystané pole, které bude mít jako *index* název sloupce a *value* hodnotu, která bude uložena v databázi. Zdrojový kód 17 popisuje způsob přípravy pole před vložením, kde jsou připraveny hodnoty pro název souboru, který je validován před vložením, cesta k souboru, typ souboru atd. Funkce *cleanSpecialChars()* je popsána v následující části práce.

```
$data = array(
    "name"=>$this->cleanSpecialChars($name_file),
    "path"=>$directory,
    "created_date"=>date("Y-m-d H:i:s"),
    "created_by" =>$user_id,
    "id_file_type"=>$file_type,
    "id_client"=>$id_client,
    "extension" => $extension,
);
```

Zdrojový kód 17: Příprava pole pro vložení záznamu do databáze

Po přípravě dat je proveden následující Zdrojový kód 18, který provede vložení záznamu do databáze a vrátí *ID* řádku, který byl právě vložen a v případě neúspěšného vložení se vrací **bool** hodnota *false*.

```
public function insert_file_path($data)
{
    $this->builder->from("file", true);
    $this->builder->insert($data);

    $id = $this->db->insertID();

    return !empty($id) ? $id : false;
}
```

Zdrojový kód 18: Příklad vložení záznamu do databáze

Pro editaci je potřeba provést úplně stejný postup jako u vložení, resp. příprava pole hodnot, které má být změněno, a navíc je potřeba mít unikátní data, např. primární klíč tabulky, podle kterého bude vyhledaný konkrétní záznam a následně se provede změna tohoto řádku. Zdrojový kód 19 zobrazuje způsob editace, kde parametry funkce jsou **pole** a **primární klíč** v tabulce *file*.

```
public function update_file($data, $id_file)
{
    $this->builder->from("file", true);
    $this->builder->where("id_file", $id_file);
    $this->builder->update($data);
}
```

Zdrojový kód 19: Příklad pro editaci konkrétního záznamu v databázi

Posledním příkazem je mazání záznamu v databázi a je podobný příkazu pro editaci. Rozdíl je v tom, že tento příkaz nepotřebuje přípravu pole hodnot a liší se v názvu funkce, kde je funkce *update()* nahrazena *delete()*, kde je parametrem pole, které obsahuje název sloupce primárního klíče a hodnotu primárního klíče jako v následujícím kódu, tj. Zdrojový kód 20.

```
$this->builder->delete(['id_file'=>$id_file]);
```

Zdrojový kód 20: Příklad odstranění záznamu z databáze

4.13 Import okresů a krajů

Ve webové aplikaci pro správu klientů a produktů je při tvorbě nového klienta nutnost uvádět adresu a číslo popisné, resp. sídlo klienta nebo doručovací adresu. Bez zadání této adresy není možné klienta vytvořit. Při využití funkcionality pro vyhledávání fyzických a právnických osob jsou tyto pole vyplněna automaticky a v opačném případě je nutnost vybrat z výběru možností o jaký stát se jedná a v jakém kraji a okrese je. Kraje a okresy České [97] a Slovenské republiky [98] jsou vloženy pomocí **SQL příkazu** *INSERT*. Import je pomocí PHP skriptu, který popisuje Zdrojový kód 21, kde dochází k přečtení celého souboru a je uložen do pole podle definovaného oddělovače ve funkci *fgetcsv()*. Následně se toto pole prochází a je vytvořeno nové pole, kde jsou definovány sloupce tabulky *district* jako *index*

pole a ukládá se uložená hodnota. Při importu krajů je postup stejný a je potřeba před spuštěním tohoto skriptu připravit pole podle tabulky *region*. Funkce *import_districts()* vykonává příkaz *INSERT*, který byl popsán v předchozí kapitole.

```
public function parseCsv($path)
{
    $path = "app/okresy_cr.csv"
    $row = 1;
    $okresy = array();
    if (($handle = fopen($path, "r")) !== FALSE) {
        while (($data = fgetcsv($handle, 1000, ",", "")) !== FALSE) {
            $num = count($data);
            $okresy[] = $data;
            $row++;
        }
        fclose($handle);
    }
    if(!empty($okresy))
    {
        foreach($okresy as &$okres)
        {
            $okres["name"] = $okres[0];
            $okres['code'] = "SK".$okres[1];
            $okres["id_region"] = $okres[2];
            unset($okres[0]);
            unset($okres[1]);
            unset($okres[2]);
        }
        unset($okres);
    }
    foreach($okresy as $district)
    {
        $pole = array(
            "name"=>$district['name'],
            "id_region"=>$district['id_region'],
            "code"=>$district['code']
        );
        $this->dashboardModel->import_districts($pole);
    }
}
```

Zdrojový kód 21: Způsob čtení souboru CSV a import okresů do databáze

4.14 Testování aplikace

Po dokončení vývoje aplikace pro tuto diplomovou práci byly realizovány manuální testy. Aplikace bude doplněna novými funkcionalitami a po úspěšné implementaci těchto funkcionalit budou sestaveny automatické testy. Manuální testy jsou zaměřeny na základní funkcionalitu systému, tj. vytvoření a editace klientů, produktů, uživatelů, upozornění, vyhledávání fyzických a právnických osob z databáze **ARES**, operace s **API Mapy.cz** a změna stavů produktu.

4.14.1 Přihlášení uživatele

Manuální testovací případ pro přihlášení uživatele je zobrazen v Tab. 18 a součástí je popis testovacího případu, základní požadavky na systém, prohlížeč a vstupní podmínky pro zahájení testu. V následující části jsou zobrazeny kroky a operace, které jsou provedeny, očekávaný výsledek a také operace před ukončením testovacího případu jako např. odhlášení uživatele ze systému a zavření prohlížeče.

Tab. 18: Test case pro přihlášení uživatele – manuální test

TC_001 – Přihlášení uživatele	
Verze: 1.0.1	
Datum: 10.05.2023	
Autor: Anh Vu Pham	
<p>Popis:</p> <p>Testovací případ ověřuje, zda došlo ke korektnímu přihlášení uživatele a po úspěšném přihlášení, zda byl uživatel přesměrován na správnou URL adresu a zda zadaný element byl nalezen.</p>	
<p>Pre-conditions</p> <ol style="list-style-type: none"> 1) MS Windows 11 Home – Verze: 22H2 2) Prohlížeč Google Chrome – Verze: 113.0.5672.93 3) Vytvořený uživatel s administrátorským oprávněním 4) Myš, klávesnice, monitor 	
Kroky	Operace
1	Uživatel spustí prohlížeč Google Chrome a zadá do prohlížeče URL: {navez_domeny}/auth → Systém načte stránku s přihlašovacím formulářem
2	Uživatel zadá e-mail a přístupové heslo
3	Uživatel klikne na tlačítko „Přihlásit“ → systém provede autentizaci uživatele a v případě zadaných správných údajů přihlásí uživatele do systému
<p>Očekávaný výsledek:</p> <ul style="list-style-type: none"> - Systém přihlásí uživatele a bude přesměrován na URL: {navez_domeny}/admin/dashboard - Hledaný element je nalezen 	

Post-conditions:

Odhlášení uživatele → Zavření prohlížeče

4.14.2 Vytvoření klienta

Tab. 19 popisuje testovací případ pro vytvoření nového klienta do systému. Pro testování tohoto případu je vyžadován přihlášený uživatel s administrátorským oprávněním. Po vyplnění všech povinných polí a potvrzení formuláře, systém validuje všechny vstupy a zkontroluje duplicitu klienta na základě čísla IČO.

Tab. 19: Test case pro vytvoření klienta – manuální test

TC_002 – Vytvoření klienta	
Verze: 1.0.1	
Datum: 10.05.2023	
Autor: Anh Vu Pham	
Popis:	
Testovací případ ověřuje, zda může přihlášený administrátor vytvořit nového klienta.	
Pre-conditions	
<ol style="list-style-type: none"> 1) MS Windows 11 Home – Verze: 22H2 2) Prohlížeč Google Chrome – Verze: 113.0.5672.93 3) Přihlášený uživatel s administrátorským oprávněním 4) Myš, klávesnice, monitor 	
Kroky	Operace
1	Uživatel spustí prohlížeč Google Chrome a zadá do prohlížeče URL: {navez_domeny}/admin/klient/novy_klient → Systém načte stránku s formulářem pro vytvoření nového uživatele
2	Uživatel klikne na tlačítko „Uložit“ → Systém vypíše chybové hlášky
3	Uživatel vyplní všechna povinná a nepovinná pole
4	Uživatel klikne na tlačítko „Uložit“ → Systém provede ověření zadaných údajů, provede validaci dat, zkontroluje duplicitu klienta a následně vytvoří nového uživatele a přesměruje uživatele na url: {navez_domeny}/admin/klient/vypis

Očekávaný výsledek: <ul style="list-style-type: none"> - Systém vytvoří nového klienta a bude přesměrován na URL: {nazev_domeny}/admin/ klient/vypis - Systém vytvoří záznam procesu „Vytvoření klienta“ → je zobrazeno v záložce Logs v detailu klienta - Hledaný element je nalezen
Post-conditions: Odhlášení uživatele → Zavření prohlížeče

4.14.3 Editace klienta

V Tab. 20 je popsán testovací případ pro editaci klienta, kdy je vybrána jedna hodnota, která bude změněna a následně jsou zkontrolovány provedené změny a zda došlo k zaznamenání akce pro editaci klienta. Tyto akce jsou očekávaným výsledkem testovacího případu. Vstupními podmínkami pro vykonání testovacího případu je úspěšné přihlášení administrátora a existující klient v systému.

Tab. 20: Test case pro editaci klienta – manuální test

TC_003 – Editace klienta	
Verze: 1.0.1	
Datum: 10.05.2023	
Autor: Anh Vu Pham	
Popis: Testovací případ ověřuje, zda může přihlášený administrátor vytvořit nového klienta.	
Pre-conditions <ol style="list-style-type: none"> 1) MS Windows 11 Home – Verze: 22H2 2) Prohlížeč Google Chrome – Verze: 113.0.5672.93 3) Přihlášený uživatel s administrátorským oprávněním 4) V systému existuje klient 5) Myš, klávesnice, monitor 	
Kroky	Operace

1	Uživatel spustí prohlížeč Google Chrome a zadá do prohlížeče URL: {navez_omeny}/admin/klient/vypis → Systém načte stránku, kde je zobrazen seznam všech klientů
2	Uživatel klikne na název klienta „Univerzita Tomáše Bati ve Zlíně“ → Systém přesměruje uživatele na detail klienta, kde je možné provádět úpravy.
3	Uživatel klikne na tlačítko „Upravit“ → Systém zobrazí vyskakovací okno s formulářem pro editaci základních údajů klienta
4	Uživatel provede změnu v položce „Číslo klienta“ na jiné číslo než ve formuláři.
5	Uživatel klikne na tlačítko „Uložit“ → Systém provede ověření zadaných údajů, provede validaci dat → při úspěšné editaci zobrazí hlášku o úspěšné editaci
6	Uživatel klikne na tlačítko „Pokračovat“ → systém skryje formulář pro editaci a obnoví stránku
7	Uživatel klikne na šipku u nadpisu Detail v části Logs → Systém zobrazí všechny záznamy o úpravě
7	Uživatel klikne na tlačítko „Upravit“ → Systém zobrazí vyskakovací okno s formulářem pro editaci základních údajů klienta a jsou zobrazeny provedené změny
<p>Očekávaný výsledek:</p> <ul style="list-style-type: none"> - Systém uloží provedené změny a zobrazí je uživateli při rozkliknutí editace základních informací klienta - Systém vytvoří záznam procesu „Editace klienta“ → je zobrazeno v záložce Logs v detailu klienta - Hledaný element je nalezen 	

Post-conditions:

Odhlášení uživatele → Zavření prohlížeče

4.14.4 Vytvoření produktu

Tab. 21 popisuje testovací případ pro vytvoření produktu. Vstupními podmínkami pro zahájení testovacího případu jsou existující klient a přihlášený uživatel s administrátorským oprávněním. V prvním kroku je přesměrování na stránku s formulářem pro vytvoření nového produktu a zkusit vytvořit produkt bez zadání dat. Systém vrátí chybovou hlášku o chybějících datech a neprovede žádné operace. **Po vyplnění všech povinných polí jako jsou název produktu, výběr typu produktu, výběr klienta a zadání data podpisu smlouvy a lhůtu k dodání, je možné vytvořit nový produkt.** Ostatní informace o produktu mohou být doplněny později a není nutné znát všechny data. Po odeslání požadavku k vytvoření produktu, systém zkontroluje zadaná data a následně je vytvořen nový produkt a přesměruje uživatele na stránku s výpisem všech produktů. S vytvořením nového produktu je založen záznam o provedení operace „Vytvoření produktu“.

Tab. 21: Test case pro vytvoření produktu – manuální test

TC_004 – Vytvoření produktu	
Verze: 1.0.1	
Datum: 10.05.2023	
Autor: Anh Vu Pham	
Popis:	
Testovací případ ověřuje, zda může přihlášený administrátor vytvořit nový produkt.	
Pre-conditions	
<ol style="list-style-type: none"> 1) MS Windows 11 Home – Verze: 22H2 2) Prohlížeč Google Chrome – Verze: 113.0.5672.93 3) Přihlášený uživatel s administrátorským oprávněním 4) V systému existuje klient 5) Myš, klávesnice, monitor 	
Kroky	Operace

1	Uživatel spustí prohlížeč Google Chrome a zadá do prohlížeče URL: {navez_domeny}/admin/produkt/novy_produkt → Systém načte stránku s formulářem pro vytvoření nového produktu
2	Uživatel klikne na tlačítko „Uložit“ → Systém vypíše chybovou hlášku
3	Uživatel vyplní všechna povinná a nepovinná pole
4	Uživatel klikne na tlačítko „Uložit“ → Systém provede ověření zadaných údajů, provede validaci dat a následně vytvoří nový produkt a přesměruje uživatele na url: {navez_domeny}/admin/produkt/vypis
Očekávaný výsledek:	
<ul style="list-style-type: none"> - Systém vytvoří nový produkt a bude přesměrován na URL: {navez_domeny}/admin/produkt/vypis - Systém vytvoří záznam procesu „Vytvoření produktu“ → je zobrazeno v záložce Logs v detailu klienta a produktu - Hledaný element je nalezen 	
Post-conditions:	
Odhlášení uživatele → Zavření prohlížeče	

4.14.5 Editace produktu

Dalším testovacím případem je editace produktu a je popsán v Tab. 22. Pro tento testovací případ musí být uživatel přihlášený s administrátorskými právy a v systému musí být již existující produkt. Po editaci produktu se očekává, že systém uložil provedené změny a zaznamenal editaci produktu.

Tab. 22: Test case pro editaci produktu – manuální test

TC_005 – Editace produktu
Verze: 1.0.1
Datum: 10.05.2023
Autor: Anh Vu Pham
Popis:

Testovací případ ověřuje, zda může přihlášený administrátor editovat existující produkt.	
Pre-conditions	
<ol style="list-style-type: none"> 1) MS Windows 11 Home – Verze: 22H2 2) Prohlížeč Google Chrome – Verze: 113.0.5672.93 3) Přihlášený uživatel s administrátorským oprávněním 4) V systému existuje produkt 5) Myš, klávesnice, monitor 	
Kroky	Operace
1	Uživatel spustí prohlížeč Google Chrome a zadá do prohlížeče URL: {navez_domeny}/admin/produkt/vypis → Systém načte stránku, kde je zobrazen seznam všech produktů
2	Uživatel klikne na název libovolného produktu → Systém přesměruje uživatele na detail produktu → url: {navez_domeny}/admin/produkt/detail/{id_produktu}
3	Uživatel klikne na tlačítko „Upravit produktu“ → Systém přesměruje uživatele na url: {navez_domeny}/admin/produkt/upravit_produkut/{id_produktu}
4	Uživatel doplní text „Test“ do pole pro popis produktu
5	Uživatel kline na tlačítko „Uložit“ → Systém provede validaci zadaných údajů, provede validaci dat → Systém přesměruje uživatele na url: {navez_domeny}/admin/produkt/detail/{id_produktu}
6	Uživatel klikne na šipku u nadpisu Logs → Systém zobrazí všechny záznamy o úpravě
Očekávaný výsledek:	
<ul style="list-style-type: none"> - Systém uloží provedené změny a jsou zobrazeny v detailu produktu - Systém vytvoří záznam procesu „Editace produktu“ → je zobrazeno v záložce Logs v detailu klienta a produktu - Hledaný element je nalezen 	

Post-conditions:

Odhlášení uživatele → Zavření prohlížeče

4.14.6 Vytvoření uživatele

Pro vytvoření nového uživatele musí být přihlášen uživatel s administrátorským oprávněním. Systém nepovolí vytvoření uživatele v případě, že je v systému jiný uživatel se stejnou zadanou e-mailovou adresou. Po úspěšném vytvoření nového uživatele je uživatel přesměrován na výpis všech uživatelů s administrátorským oprávněním. Průběh testovacího případu je popsán v Tab. 23.

Tab. 23: Test case pro vytvoření uživatele – manuální test

TC_006 – Vytvoření uživatele	
Verze: 1.0.1	
Datum: 10.05.2023	
Autor: Anh Vu Pham	
<p>Popis:</p> <p>Testovací případ ověřuje, zda může přihlášený administrátor vytvořit nového uživatele s administrátorským oprávněním.</p>	
<p>Pre-conditions</p> <ol style="list-style-type: none"> 1) MS Windows 11 Home – Verze: 22H2 2) Prohlížeč Google Chrome – Verze: 113.0.5672.93 3) Přihlášený uživatel s administrátorským oprávněním 4) Myš, klávesnice, monitor 	
Kroky	Operace
1	Uživatel spustí prohlížeč Google Chrome a zadá do prohlížeče URL: {navez_domeny}/admin/user/novy_uzivatel → Systém načte stránku s formulářem pro vytvoření nového uživatele
2	Uživatel klikne na tlačítko „Uložit“ → Systém nepovolí kliknutí na tlačítko
3	Uživatel vyplní jméno a příjmení a e-mailovou adresu
4	Uživatel vyplní přístupové heslo, které musí mít alespoň 8 znaků a obsahovat alespoň jedno malé a velké písmeno a jednu číslici.

5	Uživatel znovu vyplní přístupové heslo → Po splnění podmínek vytvoření hesla a shodě zadaných hesel → Systém zpřístupní tlačítko „Uložit“
6	Uživatel klikne na tlačítko „Uložit“ → Systém provede ověření zadaných údajů, provede validaci dat a následně vytvoří nového uživatele a přesměruje uživatele na url: {nazev_domeny}/admin/user/vypis
Očekávaný výsledek:	
<ul style="list-style-type: none"> - Systém vytvoří nového uživatele a bude přesměrován na URL: {nazev_domeny}/admin/user/vypis - Hledaný element je nalezen 	
Post-conditions:	
Odhlášení uživatele → Zavření prohlížeče	

4.14.7 Editace uživatele

Pro editaci uživatele musí být přihlášený uživatel s administrátorským oprávněním. Tab. 24 popisuje testovací případ pro editaci uživatele, kdy je prvním krokem přesměrování uživatele na stránku se seznamem všech uživatelů. Následně po kliknutí na jméno a příjmení uživatele je uživatel přesměrován na stránku, kde je zobrazen formulář pro editaci uživatele. Pro tento testovací případ je nastavena změna jména uživatele na „Test“ a po uložení jsou systémem zkontrolována data. Po provedení změny je uživatel přesměrován zpět na stránku s výpisem všech uživatelů v systému.

Tab. 24: Test case pro editaci uživatele – manuální test

TC_007 – Editace uživatele
Verze: 1.0.1
Datum: 10.05.2023
Autor: Anh Vu Pham
Popis:
Testovací případ ověřuje, zda může přihlášený administrátor editovat uživatelský účet.
Pre-conditions

1) MS Windows 11 Home – Verze: 22H2 2) Prohlížeč Google Chrome – Verze: 113.0.5672.93 3) Přihlášený uživatel s administrátorským oprávněním (Anh Vu Pham) 4) Přihlášený uživatel může editovat pouze svůj účet 5) Myš, klávesnice, monitor	
Kroky	Operace
1	Uživatel spustí prohlížeč Google Chrome a zadá do prohlížeče URL: {nazev_domeny}/admin/user/vypis → Systém načte stránku se seznamem všech uživatelů
2	Uživatel klikne na jméno a příjmení uživatele „Anh Vu Pham“ → Systém přesměruje uživatele na stránku pro editaci uživatele
3	Uživatel změní jméno na „Test“
4	Uživatel klikne na tlačítko „Uložit“ → Systém validací zadaných dat a následně provede změny a přesměruje uživatele na url: {nazev_domeny}/admin/user/vypis
Očekávaný výsledek: <ul style="list-style-type: none"> - Systém provede úpravy na existujícím uživateli a bude přesměrován na URL: {nazev_domeny}/admin/user/vypis - Hledaný element je nalezen 	
Post-conditions: Odhlášení uživatele → Zavření prohlížeče	

4.14.8 Vytvoření upozornění

Pro vytvoření nového upozornění je potřeba mít vytvořený produkt, ke kterému má být upozornění vázané. Jelikož nelze přímo v detailu produktu vytvářet upozornění a nejsou zde zobrazeny všechna upozornění, tak je možné tyto upozornění vytvářet pouze v detailu klienta, aby nedocházelo k záměně, neboť mohou být produkty k sobě podobné a jednoduše může dojít k omylu. Jak popisuje Tab. 25, tak prvním krokem je přesměrovat se na stránku s výpisem všech klientů a následně se přesměrovat na detail konkrétního klienta, který má produkt, pro který má být vytvořeno upozornění. Kliknutím na tlačítko „Přidat upomínku“ je uživateli zobrazen formulář pro vytvoření nového upozornění, kde je potřeba zvolit druh upomínky, který je možný přidávat a editovat v nástrojích systému. Po vyplnění mezního

termínu a zvolení všech příjemců upozornění systém vytvoří nové upozornění a následně zaeviduje záznam o vytvoření.

Tab. 25: Test case pro vytvoření upozornění – manuální test

TC_008 – Vytvoření upozornění	
Verze: 1.0.1	
Datum: 10.05.2023	
Autor: Anh Vu Pham	
Popis:	
Testovací případ ověřuje, zda může přihlášený administrátor vytvářet nové upozornění	
Pre-conditions	
<ol style="list-style-type: none"> 1) MS Windows 11 Home – Verze: 22H2 2) Prohlížeč Google Chrome – Verze: 113.0.5672.93 3) Přihlášený uživatel s administrátorským oprávněním 4) V systému musí být založený produkt 5) Produkt musí mít známé datum podpisu smlouvy a datum dodání 6) Myš, klávesnice, monitor 	
Kroky	Operace
1	Uživatel spustí prohlížeč Google Chrome a zadá do prohlížeče URL: {navez_domeny}/admin/klient/vypis → Systém načte stránku se seznamem všech klientů
2	Uživatel klikne na klienta „Univerzita Tomáše Bati ve Zlíně“ → Systém přeměruje uživatele na stránku pro editaci klienta
3	Uživatel vyhledá produkt a položku „Termíny“
4	Uživatel klikne na tlačítko „Přidat upomínku“ → Systém zobrazí formulář pro vytvoření upozornění
5	Uživatel vybere druh upomínky, vybere způsob upozornění, vybere datum upozornění a zvolí příjemce.
6	Uživatel klikne na tlačítko „Uložit“ → Systém provede ověření zadaných údajů, provede validaci dat a následně vytvoří nové upozornění a obnoví stránku → Systém zobrazí vytvořené upozornění v části „Upozornění“

7	Uživatel klikne na šipku u nadpisu Detail v části Logs → Systém zobrazí všechny záznamy o úpravě
Očekávaný výsledek:	
<ul style="list-style-type: none"> - Systém vytvoří nové platné upozornění - Systém vytvoří záznam procesu „Nové upozornění“ → je zobrazeno v záložce Logs v detailu klienta - Hledaný element je nalezen 	
Post-conditions:	
Odhlášení uživatele → Zavření prohlížeče	

4.14.9 Editace upozornění

Postup editace upozornění je velmi podobný procesu vytvoření a vstupní podmínkou je, že upozornění, které chceme editovat, musí být vytvořené. Tab. 26 popisuje postup editace upozornění a po úspěšné editaci je vytvořen záznam o provedené operaci.

Tab. 26: Test case pro editaci upozornění – manuální test

TC_009 – Editace upozornění	
Verze: 1.0.1	
Datum: 10.05.2023	
Autor: Anh Vu Pham	
Popis:	
Testovací případ ověřuje, zda může přihlášený administrátor editovat existující upozornění.	
Pre-conditions	
<ol style="list-style-type: none"> 1) MS Windows 11 Home – Verze: 22H2 2) Prohlížeč Google Chrome – Verze: 113.0.5672.93 3) Přihlášený uživatel s administrátorským oprávněním 4) V systému musí být založený produkt 5) Produkt musí mít známé datum podpisu smlouvy a datum dodání 6) Produkt již má vytvořené upozornění 7) Myš, klávesnice, monitor 	
Kroky	Operace

1	Uživatel spustí prohlížeč Google Chrome a zadá do prohlížeče URL: {navez_domeny}/admin/klient/vypis → Systém načte stránku se seznamem všech klientů
2	Uživatel klikne na klienta „Univerzita Tomáše Bati ve Zlíně“ → Systém přeměruje uživatele na stránku pro editaci klienta
3	Systém klikne na tlačítko pro editaci upozornění v části Upozornění → Systém zobrazí formulář pro editaci existujícího upozornění
4	Uživatel zadá „Testování editace“ do pole Poznámka k upomínce
5	Uživatel klikne na tlačítko „Uložit“ → Systém provede ověření zadaných údajů, provede validaci dat a následně provede změny upozornění a obnoví stránku
7	Uživatel klikne na šipku u nadpisu Detail v části Logs → Systém zobrazí všechny záznamy o úpravě → po kontrole je vytvořený záznam o editaci nového upozornění
Očekávaný výsledek:	
<ul style="list-style-type: none"> - Systém provede změny existujícího upozornění - Systém zobrazí poznámku „Testování editace“ u upozornění, které bylo editováno - Systém vytvoří záznam procesu „Editace upozornění“ → je zobrazeno v záložce Logs v detailu klienta - Hledaný element je nalezen 	
Post-conditions:	
Odhlášení uživatele → Zavření prohlížeče	

4.14.10 Vyhledávání fyzických a právnických osob

Testovací případ v Tab. 27 popisuje způsob vyhledávání fyzických a právnických osob z databáze ARES. Vyhledávání probíhá u vytváření nebo editace klienta, kdy je potřeba do vstupního pole pro *IČO* vložit osm číslic. Ve vstupním poli jsou provedeny validace vstupních znaků, kde není možné vkládat jiné znaky než číslice. Po zadání osmimístného čísla je uživateli zpřístupněno tlačítko pro získávání dat z databáze **ARES** a v případě, že je nalezen

klient, tak systém automaticky vyplní pole *DIC*, *Název*, *Ulice a č. p.*, *Město*, *Stát*, *PSC*, *Kraj* a *Okres*, kde jediné pole *DIC* nemusí být vyplněno kvůli chybějícím datům z požadavku.

Tab. 27: Test case pro vyhledávání fyzických a právnických osob – manuální test

TC_010 – Vyhledávání fyzických a právnických osob	
Verze: 1.0.1	
Datum: 10.05.2023	
Autor: Anh Vu Pham	
<p>Popis:</p> <p>Testovací případ ověřuje, zda je možné při vytvoření nebo editaci klienta najít fyzickou nebo právnickou osobu pomocí čísla IČO.</p>	
<p>Pre-conditions</p> <ol style="list-style-type: none"> 1) MS Windows 11 Home – Verze: 22H2 2) Prohlížeč Google Chrome – Verze: 113.0.5672.93 3) Přihlášený uživatel s administrátorským oprávněním 4) Myš, klávesnice, monitor 	
Kroky	Operace
1	Uživatel spustí prohlížeč Google Chrome a zadá do prohlížeče URL: {navez_omeny}/admin/klient/vypis → Systém načte stránku, kde je zobrazen seznam všech klientů
2	Uživatel klikne na tlačítko „Vytvořit klienta“ → Systém zobrazí vyskakovací okno s formulářem pro vytvoření nového klienta
3	Uživatel klikne na tlačítko „ARES“ → Systém nepovolí kliknout na toto tlačítko → uživatel musí zadat posloupnost 8 čísel
4	Uživatel zadá do pole IČO hodnotu „UTB“ → Systém nepovolí vkládat písmena a speciální znaky
5	Uživatel zadá do pole IČO hodnotu „70883521“ → Systém zpřístupní tlačítko „ARES“
Očekávaný výsledek:	

- Systém po úspěšném hledání vyplní pole <i>DIC</i> , <i>Název</i> , <i>Ulice a č.p.</i> , <i>Město</i> , <i>Stát</i> , <i>PSČ</i> , <i>Kraj</i> , <i>Okres</i> → jediné pole <i>DIC</i> nemusí existovat a může být prázdné
Post-conditions: Odhlášení uživatele → Zavření prohlížeče

4.14.11 Zaznamenání polohy produktu

Tento testovací případ vychází z testovacího případu pro editaci produktu v Tab. 22. Ve formuláři pro editaci produktu je v dolní části formuláře vstupní pole pro našeptávač společně s mapou, která zobrazuje střed Prahy v případě chybějících dat zeměpisné šířky a délky, jinak je zobrazena poslední uložená poloha podle hodnot zeměpisné šířky a délky. V rámci testovacího případu popsany v Tab. 28 je testována funkcionality našeptávače, kde je zadána hodnota „Nad Stráněmi 4511“ a po kliknutí na první zobrazenou hodnotu je mapa překreslena, zobrazí přesnou polohu zvolené hodnoty a následně jsou vyplněny vstupní pole pro zeměpisnou šířku a délku. Po potvrzení formuláře dojde k uložení a zaznamenání operace „Editace produktu“.

Tab. 28: Test case pro zaznamenání polohy produktu – manuální test

TC_011 – Zaznamenání polohy produktu	
Verze: 1.0.1	
Datum: 10.05.2023	
Autor: Anh Vu Pham	
Popis: Testovací případ ověřuje, zda je možné zaznamenat polohu u existujícího produktu.	
Pre-conditions <ol style="list-style-type: none"> 1) MS Windows 11 Home – Verze: 22H2 2) Prohlížeč Google Chrome – Verze: 113.0.5672.93 3) Přihlášený uživatel s administrátorským oprávněním 4) V systému existuje produkt 5) Myš, klávesnice, monitor 	
Kroky	Operace

1	Uživatel spustí prohlížeč Google Chrome a zadá do prohlížeče URL: {navez_domeny}/admin/produkt/vypis → Systém načte stránku, kde je zobrazen seznam všech produktů
2	Uživatel klikne na název libovolného produktu → Systém přesměruje uživatele na detail produktu → url: {navez_domeny}/admin/produkt/detail / {id_produkту}
3	Uživatel klikne na tlačítko „Upravit produktu“ → Systém přesměruje uživatele na url: {navez_domeny}/admin/produkt/upravit_produkту/{id_produkту}
4	Uživatel doplní text „Nad Stráněmi 4511“ do vstupního pole pro našeptávače
5	Uživatel klikne na 1. zobrazenou možnost v našeptávači → Systém obnoví mapu, zobrazí polohu zvolené možnosti a vyplní hodnoty zeměpisné délky (longitude) a šířky (latitude)
5	Uživatel kline na tlačítko „Uložit“ → Systém provede ověření zadaných údajů, provede validaci dat → Systém přesměruje uživatele na url: {navez_domeny}/admin/produkt/detail / {id_produkту}
6	Uživatel klikne na šipku u nadpisu Logs → Systém zobrazí všechny záznamy o úpravě
<p>Očekávaný výsledek:</p> <ul style="list-style-type: none"> - Systém uloží provedené změny a jsou zobrazeny v detailu produktu - Systém automaticky vyplní a uloží hodnoty zeměpisné délky a šířky (longitude, latitude) - Systém automaticky vyplní a uloží adresu podle výstupu z našeptávače - Po vyplnění těchto dat je zobrazena mapa a značka na zvoleném místě - Systém vytvoří záznam procesu „Editace produktu“ → je zobrazeno v záložce Logs v detailu klienta a produktu - Hledaný element je nalezen 	

Post-conditions:

Odhlášení uživatele → Zavření prohlížeče

4.14.12 Změna stavu produktu

Testovací případ ověřuje funkcionalitu pro změnu stavu produktu. Před změnou stavu produktu musí produkt existovat a zároveň musí kategorie, přiřazená k produktu, mít definovanou posloupnost stavů, jinak není možné stavy produktu měnit. Způsob testování je popsáno v Tab. 29, kdy je na úvodní stránce panel „Rychlé akce“, ve kterém jsou zobrazeni klienti, produkty a kontaktní osoby. Kliknutím na stav produktu je otevřeno vyskakovací okno, kde je zobrazen formulář s nabídkou stavů pro danou kategorii produktu. Po potvrzení dojde k uložení stavu a následně k obnově stránky. Součástí je i vytvoření záznamu o provedené změně stavu produktu.

Tab. 29: Test case pro změnu stavu produktu – manuální test

TC_012 – Změna stavu produktu	
Verze: 1.0.1	
Datum: 10.05.2023	
Autor: Anh Vu Pham	
Popis:	
Testovací případ ověřuje, zda může přihlášený administrátor změnit stav produktu.	
Pre-conditions	
<ol style="list-style-type: none"> 1) MS Windows 11 Home – Verze: 22H2 2) Prohlížeč Google Chrome – Verze: 113.0.5672.93 3) Přihlášený uživatel s administrátorským oprávněním 4) V systému existuje produkt 5) Kategorie u existujícího produktu má nedefinovanou posloupnost stavů 6) Myš, klávesnice, monitor 	
Kroky	Operace
1	Uživatel spustí prohlížeč Google Chrome a zadá do prohlížeče URL: {navez_domeny}/admin/dashboard → Systém načte úvodní stránku
2	Uživatel klikne na stav u produktu v panelu „Rychlé akce“ → Systém zobrazí formulář pro editaci stavu

3	Uživatel vybere hodnotu „Složeno ve skladě“ (případně jinou hodnotu)
4	Uživatel klikne na tlačítko „Uložit“ → Systém provede změnu a zobrazí hlášku úspěšné editace stavu
5	Uživatel klikne na tlačítko „Pokračovat“ → Systém zavře formulář a obnoví stránku
6	Uživatel klikne na název produktu „Outdoor 55“ → Systém přesměruje uživatele na detail produktu v novém okně
Očekávaný výsledek:	
<ul style="list-style-type: none"> - Systém změní předešlý stav produktu na nový „Složeno ve skladě“ (případně jiná hodnota) - Systém vytvoří záznam procesu „Změna stavu produktu“ → je zobrazeno v záložce Logs v detailu produktu společně s informací o předešlém stavu a novém stavu - Hledaný element je nalezen 	
Post-conditions:	
Odhlášení uživatele → Zavření prohlížeče	

4.15 Bezpečnost

Při vývoji jakéhokoliv systému nebo aplikace, resp. webové, mobilní nebo desktopové atd. je základní otázkou dostatečná bezpečnost systému. Při vývoji se většinou klade na důraz na dostatečné validace, způsoby ošetření některých vstupů od uživatele, šifrování některých dat atd. Bez dostatečného zabezpečení systému může totiž docházet k zneužití slabých stránek aplikací, odposlouchávání prostřednictvím nešifrované síťové komunikace, neoprávněné vstupy do systému kvůli slabé nebo chybějící autentizaci nebo pomocí metodiky sociálního inženýrství atd. Pro obranu proti jedné z metod sociálního inženýrství je v aplikaci realizována funkce pro automatické odhlášení uživatele po vypršení předem definované doby uvedené v kapitole 4.6 Automatické odhlášení uživatele. Tato práce je vyvíjena na vývojovém serveru, který využívá přenosový protokol **HTTP** a produkční verze využívá **HTTPS**. V rámci bezpečnosti aplikace jsou v následujících kapitolách zmíněny některé známé útoky jako **Cross-site Scripting**, **SQL injection** apod. společně i se způsobem obrany proti těmto útokům.

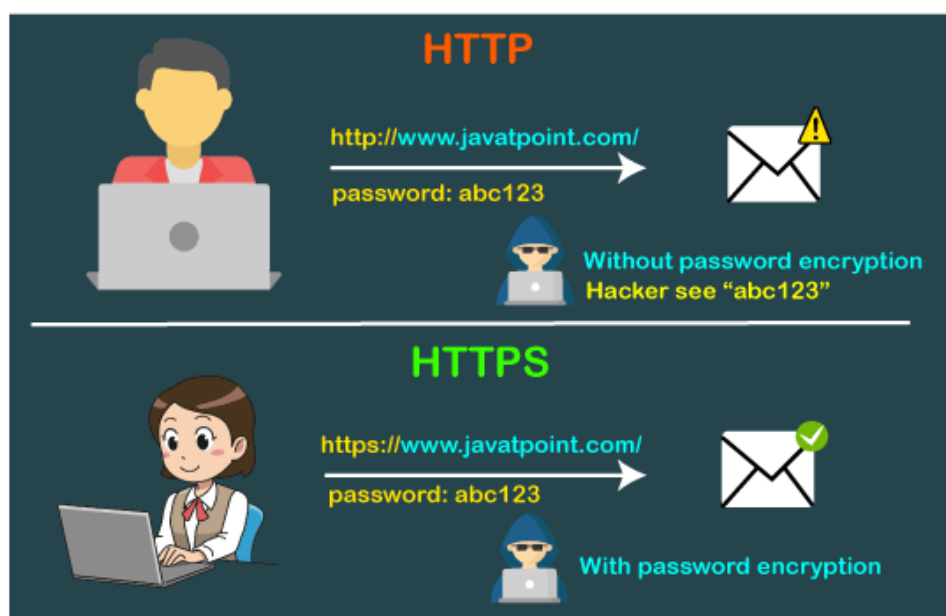
4.15.1 Autentizace

Autentizace [99] je proces, kdy se zjišťuje, zda je někdo skutečně tím, za koho se vydává. Tato technologie ověřování zajišťuje řízení přístupu k systému tím, že kontroluje, zda se údaje shodují s údaji v databázi oprávněných uživatelů. Autentizace tedy umožňuje udržovat systém a data v bezpečí tím, že povoluje přístup k chráněným zdrojům pouze ověřeným uživatelům nebo procesům. To může zahrnovat počítačové systémy, sítě, databáze webových stránky atd. Při realizaci webové aplikace pro správu zákazníků a produktů je pro ověření autentizace použita knihovna **IonAuth**, která je zmíněna v kapitole 4.7 IonAuth.

4.15.2 Přenosové protokoly HTTP a HTTPS

HTTP (*Hypertext Transfer Protocol*) [100] je protokol zajišťující komunikaci mezi různými komunikačními systémy. Když uživatel v prohlížeči zadá požadavek **HTTP**, webový server odešle požadovaná data uživateli ve formě webových stránek, resp. protokol **HTTP** umožňuje přenášet data ze serveru klientovi. Protokol **HTTP** nezajišťuje bezpečnost dat.

HTTPS (*Hypertext Transfer Protocol Secure*) [100] je v podstatě zabezpečená verze protokolu **HTTP**, neboť umožňuje přenášet data v šifrované podobě. Tento protokol je žádoucí u systémů, kde je potřeba zadat údaje o bankovním účtu, přihlašovací údaje atd. Protokol **HTTPS** pro zajištění šifrování využívá protokol známý jako **Secure Sockets Layer (SSL)** a aktualizovaná a bezpečnější verze **Transport Layer Security (TLS)**. Na následujícím Obr. 61 [101] je popsán rozdíl mezi protokoly **HTTP** a **HTTPS**.



Obr. 61: Rozdíl mezi protokoly HTTP a HTTPS

Vývoj této aplikace je nasazen na vývojový server **HTTP** a produkční aplikace je na serveru **HTTPS** jak již bylo zmíněno v kapitole 4.15 Bezpečnost.

4.15.3 Certifikáty SSL a TLS

Certifikáty **SSL** (*Secure Socket Layer*) a **TLS** (*Transfer Layer Security*) [102] jsou v podstatě technologie pro zabezpečení internetového připojení a ochranu všech citlivých dat, která jsou přenášena mezi dvěma systémy, což zabraňuje třetí osobě číst a upravovat přenášené informace. **TLS** je v podstatě modernější a bezpečnější náhrada certifikátu **SSL**. Pro ochranu a zabezpečení dat je využito asymetrické a symetrické šifrování. Asymetrické šifrování se používá k vytvoření zabezpečené relace mezi klientem a serverem a symetrické šifrování se používá k výměně dat v rámci zabezpečené relace. Webové stránky musí mít certifikát **SSL/TLS** pro svůj název domény nebo webový server, aby bylo možné používat šifrování. Bezpečný přenos je zaručen po nainstalování certifikátu a komunikace mezi klientem a serverem probíhá podle následujících kroků [103]:

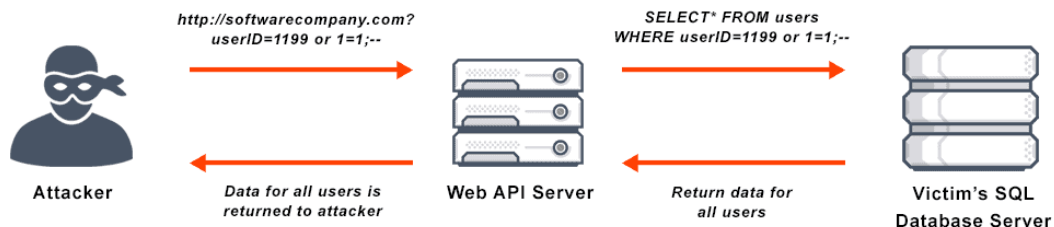
1. Klient kontaktuje server pomocí zabezpečené adresy URL (**HTTPS**)
2. Server pošle klientovi svůj certifikát a veřejný klíč
3. Klient vše ověří u **důvěryhodné kořenové certifikační autority**, zda se jedná legitimní certifikát
4. Klient a server se dohodnou na nejsilnějším typu šifrování, který mohou podporovat
5. Klient zašifruje „**Session keys**“ veřejným klíčem a odešle jej zpět serveru
6. Server dešifruje komunikaci klienta svým soukromým klíčem → navázána relace
7. „**Session keys**“ se používá k šifrování a dešifrování dat přenášených mezi klientem a serverem

Mezi serverem a klientem dochází k vytvoření „**Session Keys**“ pro vzájemnou komunikaci a po opuštění webové stránky jsou tyto klíče zlikvidovány. Při další návštěvě se vygeneruje nová sada klíčů. Počáteční komunikace mezi klientem a serverem se nazývá **Handshake**.

4.15.4 SQL injection

SQL injection [104] je technika, kterou útočníci používají k získání neoprávněného přístupu do databáze webové aplikace přidáním škodlivého kódu do dotazu na databázi. **SQL injection** manipuluje s kódem **strukturovaného dotazovacího jazyka** (*SQL*) tak, aby umožnil přístup k chráněným datům a při správném provedení je možné odhalit jakékoliv údaje

z databáze. Tyto útoky je možné použít u jakékoliv aplikace, která využívá databázi **SQL**. Na následujícím Obr. 62 [105] je popsán princip **SQL injection**.



Obr. 62: Princip *SQL injection*

Příkladem může být aplikace, která uchovává informace o uživatelích, kteří jsou vyhledávání pomocí identifikátoru *userID* a místo vyhledávání konkrétního *userID* může útočník do vstupního pole vložit kód: “*userID = 1199 OR 1=1*“. Příkaz *1=1* je vždy pravdivý a místo zobrazení uživatele, který má *userID* 1199, tak by **SQL** dotaz vrátil všechny dostupné zákazníky i s daty o zákaznících, což umožní útočníkovi obejít ověření a získat přístup na úrovni správce.

CodeIgniter [106] umožňuje přístup ke třídě **Query Builder** [107], která umožňuje vyhledávat, vkládat a aktualizovat informace v databázi. Hlavní výhodou **Query Builderu** je to, že umožňuje vytváření bezpečnějších dotazů. **Query Builder** není navržen tak, aby zabránil všem útokům a součástí by měla být řádná validace vstupních dat, což bude vést k minimalizaci rizik. Při realizaci této webové aplikace jsou využity funkce od **Query Builder** pro všechny **CRUD** operace (Create, Read, Update, Delete), kde dochází k **doplnění zpětného lomítka** \ před znaky uvozovek, resp. **jednoduché ' a dvojité "**. Dosazení zpětného lomítka zabraňuje předčasnému ukončení řetězce a nedovolí útočníkovi doplnit škodlivý kód. Příkladem může být doplnění řetězce `OR 1 = 1` do url odkazu jako např.

```
{navez_domeny}/admin/klient/upravit_klienta/65'%22+or+1=1
```

, kdy příkaz `1=1` je vždy pravdivý a mohlo by dojít k zobrazení všech záznamů z tabulky *client*. Následující Zdrojový kód 22 zobrazuje vygenerovaný **SQL dotaz** pomocí **Query Builderu**, kde jsou doplněné zpětná lomítka před uvozovkami.

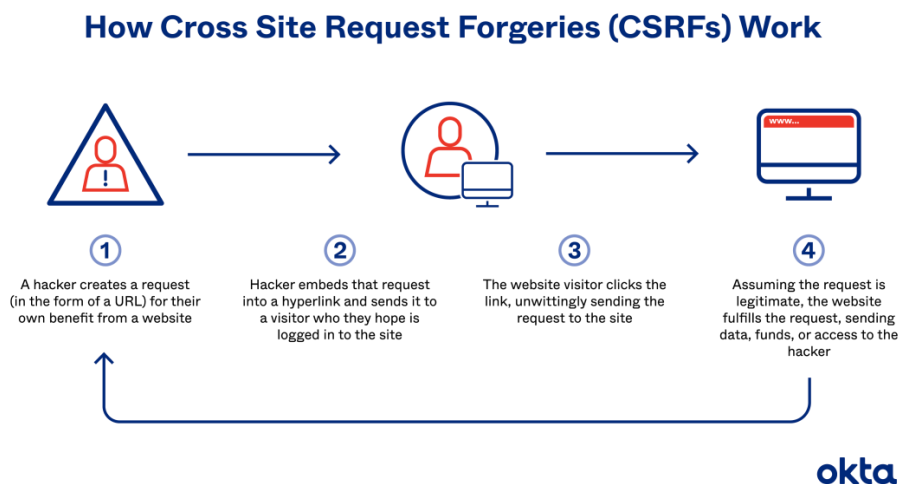
```
SELECT `c`.* FROM `client` as `c` WHERE `c`.`id_client` = '65\'\' or 1=1'
```

Zdrojový kód 22: Ukázka SQL dotazu při doplnění škodlivého kódu

Následující **SQL dotaz** tedy zobrazí klienta s *ID 65* a nedojde k zobrazení všech klientů. Před provedením jedné z **CRUD** operace jsou data v této aplikaci validována a jsou odstraněny nepovolené znaky.

4.15.5 Cross-Site Request Forgery (CSRF)

Cross-Site Request Forgery (CSRF) [108] je útok, který nutí koncového uživatele provádět nežádoucí akce na webové aplikaci, ve které je aktuálně autentizován. Pomocí sociálního inženýrství, kdy útočník pomocí e-mailu může přimět uživatele ke kliknutí na odkaz, který obsahuje neautorizovaný požadavek na konkrétní webovou aplikaci. Úspěšný útok **CSRF** může donutit uživatele provést požadavky na změnu stavu jako např. převod financí, změna e-mailové adresy, změna hesla atd. **Sociální inženýrství** [109] manipuluje s lidmi ke sdílení informací, stahování softwarů, navštěvování webových stránek bez vědomí uživatele nebo posílání peněz nebo dělání jiných chyb, které by mohly ohrozit osobní nebo organizační majetek nebo bezpečnost. Útoky **CSRF** se zaměřuje na webové aplikace, které nedokážou rozlišit mezi platnými a podvrženými požadavky. Na Obr. 63 [110] je popsán princip fungování **CSRF**.



Obr. 63: Princip fungování **CSRF**

V prvním kroku je vytvořen požadavek útočníkem, který bývá umístěn uvnitř formuláře nebo URL, následně je vložen požadavek do hypertextového odkazu a dále je odeslán uživateli. Následně dochází k čekání na uživatele, kdy dojde ke kliknutí na odeslaný odkaz a dojde k odeslání požadavku na cílovou stránku. V případě legitimního požadavku cílová stránka vykoná požadavek a odešle data, finanční prostředky nebo přístup.

Nejlepším způsobem pro zabránění k útokům **CSRF** je použití tokenu **CSRF** [111]. Jedná se o bezpečný náhodný token, který je pro každou relaci uživatele jedinečný a měl by mít velkou náhodnou hodnotu, aby bylo obtížné jej uhodnout. Aplikace zabezpečená proti **CSRF** přiřazuje jedinečný **CSRF** token pro každou relaci uživatele a jsou vkládány do skrytých parametrů formulářů **HTML**.

Framework **CodeIgniter** má knihovnu *Security Class* [112], která obsahuje metody pro vytváření bezpečnějších aplikací a obsahuje funkce, které by měly zabezpečovat aplikaci proti útokům **CSRF**. Knihovna umožňuje taktéž vytvářet **CSRF** tokeny, které při validaci potvrzují pravost dat zadané uživatelem. Pro začátek je potřeba inicializovat **CSRF** protekci pomocí úprav souborů *Filters.php* a *Security.php*, které jsou v adresáři *app/Config/*, kde v souboru *Filters.php* nastavíme **CSRF** filtr. Zdrojový kód 23 popisuje způsob zapnutí filtru.

```
public $globals = [  
    'before' => ['csrf', ],  
];
```

Zdrojový kód 23: Zapnutí CSRF filtru

Dalším krokem může být nastavení **náhodného generování CSRF tokenu** v souboru *Security.php*, kde stačí přidat globální proměnou *\$tokenRandomize* a nastavit hodnotu na *true*. Zdrojový kód 24 popisuje způsob definování pro generování náhodného **CSRF** tokenu a automatické regenerování tokenu.

```
public $tokenRandomize = true;  
public $regenerate = true;
```

Zdrojový kód 24: Nastavení náhodného generování CSRF tokenu

Generování tokenů je možné pomocí následného kódu:

```
<input type="hidden" name="<?= csrf_token() ?>" value="<?= csrf_hash()  
?>" />
```

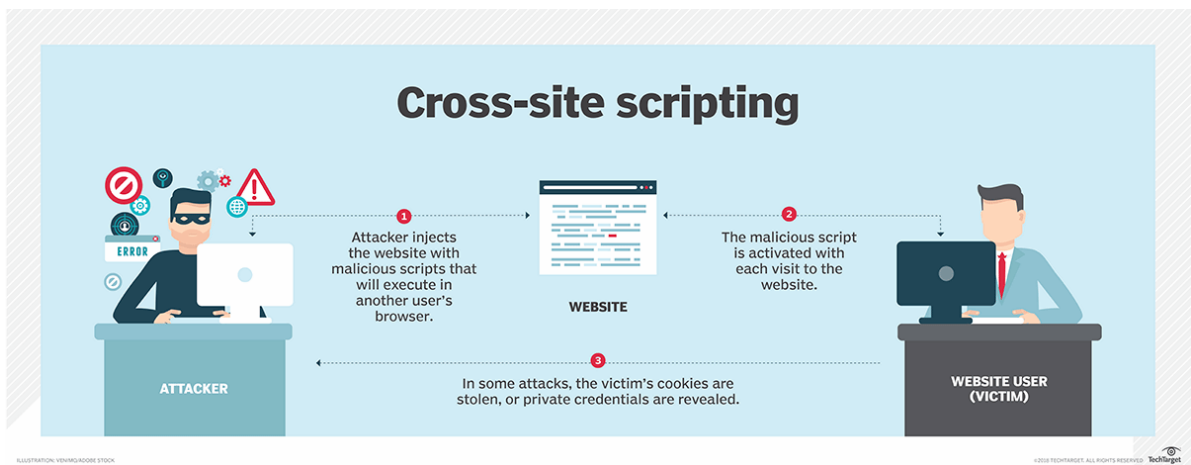
Zdrojový kód 25: Generování CSRF tokenu

Zdrojový kód 25 popisuje způsob přidání **CSRF** tokenu přímo do formuláře, který je následně po odeslání formuláře zkontrolován, zda se jedná o správný token. V případě shodného tokenu dojde k pokračování skriptu, tedy k validaci a provedení požadované operace.

4.15.6 Cross Site Scripting

Cross-Site Scripting (XSS) [113] je útok, který je založen na vložení skriptů do neošetřených vstupních polí, jako jsou např. pole pro vyhledávání, pole ve formulářích apod. Jedná se o jeden z nejčastějších typů kybernetických útoků a mohou obsahovat skripty, které jsou

spuštěny v prohlížeči oběti [113]. Nejčastěji bývají tyto skripty v jazyce Javascript, Java atd. Na Obr. 64 [114] je popsán princip XSS.



Obr. 64: Princip Cross-Site scripting

XSS je podobný jiným útokům typu *injection*, např. **SQL injection** a využívá slabiny prohlížečů, které nedokážou rozlišit legitimní operace a spustí všechny požadavky. Principem je, že útočník vytvoří škodlivý kód, který podstrčí uživateli [115]. Tento skript je spuštěný při každé návštěvě uživatele a dovoluje útočníkovi získávat data, jako např. přihlašovací údaje, soubory relace cookies atd. **Cookies** [116] jsou informace, které jsou uloženy webovou stránkou do počítače uživatele a ukládají omezené informace z relace webového prohlížeče na dané webové stránce, které lze načíst v budoucí relaci. Obranou proti tomuto útoku je ošetření a validace vstupních polí zadané uživatelem, kdy dojde k odstranění všech znaků mimo písmena abecedy a čísel.

Zdrojový kód 26 a Zdrojový kód 27 popisují funkce `cleanInputString()` a `cleanSpecialChars()`, které provádí validaci vstupních dat od uživatele. Zdrojový kód 26 odstraňuje všechny znaky kromě znaků abecedy (velká i malá písmena), čísel a mezery. Zdrojový kód 27 nahradí všechny znaky, které jsou v poli `$symbols`, prázdným řetězcem.

```
public function cleanInputString(string $string) : string
{
    return preg_replace('/[^A-Za-z0-9 ]/', '', $string);
}
```

Zdrojový kód 26: Způsob validace řetězce

```
public function cleanSpecialChars(string $string) : string
{
    $symbols = [
        ";", "<", ">", "%", "{", "}", "[", "]",
        "=", "\\ ", "(,)", "\", "'",
    ];
```

```

];
return str_replace($symbols, '', $string);
}

```

Zdrojový kód 27: Odstranění nepovolených symbolů – validace

Na následujícím Obr. 65 je zobrazen příklad přidání javascriptového kódu přímo do vstupního pole formuláře. Tento kód může být nahrazen právě škodlivým kódem.

Adresa / Sídlo ^

Ulice a č. p. *

<script> alert("ahoj"); </script>@+ěščřžýáíé;úů{}()[]\$"

Město

<script> alert("ahoj"); </script>

Obr. 65: Příklad vložení kódu pomocí jazyku Javascript do vstupního pole

Po uložení těchto hodnot vstupních polí na Obr. 65 dojde k validaci vstupních dat. Vstupní pole pro *Ulice a č. p.* je očištěno pomocí funkce `cleanInputString()` a vstupní pole *Město* pomocí `cleanSpecialChars()`. Zdrojový kód 26 popisuje funkci `cleanInputString()` a je použita u vstupních polí, ve kterých by se neměly objevovat speciální znaky jako například u poznámek a popisů. Zdrojový kód 27 popisuje funkci `cleanSpecialChars()` odstraňuje znaky, které by mohly umožnit přidání škodlivého kódu. Výstup po validaci zadaných znaků je znázorněn na následujícím Obr. 66.

Adresa
 script alertahoj /script@+ěščřžýáíéúů\$
 76502, script alertahoj /script
 Zlín, Zlínský kraj
 Česká republika

Obr. 66: Výsledek uložení javascriptového kódu

5 NÁVRHY PRO VYLEPŠENÍ APLIKACE

Realizace webové aplikace pro správu a zákazníků je aktuálně přizpůsobená na požadavky systému pro správu konkrétních produktů, jako např. úřední desky, platební moduly, softwarová řešení atd. Aplikace jako taková by mohla být obecnější za účelem rozšíření možnosti správy dalších produktů. Prvním krokem vylepšení aplikace je upravit veškerou funkcionalitu jako moduly, které by administrátor mohl různě zapínat a vypínat na základě toho, jestli jsou potřeba pro daný typ produktu. Dalším nápadem je generování faktur, kdy by ke každému produktu bylo možné vystavit fakturu, vybrat položky k fakturaci, automaticky doplňovat údaje o klientovi atd. Kromě nahrávání souborů *pdf* a obrázků je možné, že bude v budoucnu potřeba přidávat další typy souborů a nabízí se možnost definovat v typech souborů přípony a typy souborů, které bude možné nahrát.

5.1 Import modulů

Jak bylo na úvod této kapitoly zmíněno, tak některé typy produktů jako např. softwarová řešení nebudou potřebovat zaznamenávat polohu produktů a nabízí se možnost **vytvořit moduly pro jednotlivé funkcionality** jako je **zaznamenání polohy a zobrazení podle mapy**. Podobně na tom mohou být stavy k produktům, které taky nejsou potřebné u všech druhů produktů, ale naopak mohou fungovat jako štítky, které mohou být vytvořeny nebo vybrány z existujících štítků. Nabízí se tedy možnost, zda **pro daný typ produktu budou ukládány stavy produktu nebo štítky produktů** (*tags*). Dalším příkladem může být definování jednotlivých položek, které budou u daného produktu fakturovány a ukládány, kdy u softwarových řešení se může jednat o pronájem hostingu, práce za vývoj atd. U úředních desek se může jednat o cestovní náklady, materiál pro sestavení a složení atd.

5.2 Generování faktury produktu

V rámci webové aplikace je možné nahrávat smlouvy, dodatky, obrázky atd. a každý prodaný produkt nebo služby mohou být fakturovány. Při rešerši existujících řešení v platformě **Wordpress** bylo zmíněno rozšíření **WooCommerce**, v rámci rozšíření **Kybernaut IČO DIČ** v kapitole 1.4.6 Pluginy, které nabízí možnost prodávat produkty. Požadavky tohoto systému nevyžadují nabízet produkty k prodeji. Tato funkcionalita by měla pouze dokázat **vygenerovat fakturu na základě vybraných položek**, které mohou být fakturovány. U úředních desek to mohou být náklady za dopravu, materiál na sestavení a samotná práce za sestavení, u softwarových řešení to může být částka za vývoj apod. Na základě budoucí

funkcionality, která by nabízela možnost vybrat nebo vytvořit nové druhy položek k fakturaci nebo ukládání jednotlivých cen, by bylo možné zvolit si jednotlivé položky a zadat požadavek na vygenerování faktury, která by automaticky vygenerovala fakturační údaje klienta. Pro úplnost dat by bylo potřeba doplnit bankovní spojení a kompletní fakturační údaje dodavatele i odběratele. **Generování PDF souborů** by mohlo být řešeno pomocí knihoven **FPDF**, **mPDF** apod. Společně s generováním faktury by bylo možné **generování QR kódů** pro platbu, tj. **Quick response code** [117], typ čárového kódu, který se používá k zajištění snadného přístupu k online informacím skrz digitální fotoaparáty na mobilních zařízeních nebo tabletech. Tyto kódy mohou mimo jiné obsahovat odkaz na webovou stránku, odeslání textových zpráv atd. V systému je připravena funkcionality pro odesílání upozornění prostřednictvím e-mailové zprávy, kterou lze využít pro odesílání e-mailu se zprávou o vygenerované faktuře společně s přílohou, která bude obsahovat vygenerovanou fakturu. Při využití stejné knihovny **PHPMailer** [80] jako u kapitoly 4.8 Cron je možné přidávat i přílohy a je potřeba znát pouze cestu k příloze pomocí funkce *addAttachment()*.

5.3 Možnost definovat typy souborů (přípony)

V systému je možné definovat typy souborů jako např. typy obrázků, typy dokumentů apod. **Možnost definovat typy (přípony) souborů** je další funkcionalitou, která by měla umožnit při vytváření nových typů souborů definovat jaké přípony souborů budou akceptovány a následně budou tyto soubory filtrovány při nahrávání (výběru) souboru, kde budou nastavené filtry přípon podle nastavených hodnot a následně se dají validovat i po potvrzení formuláře. Validace po potvrzení bude provedena tak, že se získají všechny akceptovatelné přípony z databáze u daného typu souboru a porovná se, zda nahraný soubor má tuto příponu.

ZÁVĚR

V teoretické části je zpracovaná rešerše existující řešení softwarových platform Asana a Freeloo a systémů pro správu obsahu Wordpress a Joomla. Výsledkem rešerše je, že komplexní řešení pro správu zákazníků a produktů existují, ale tyto softwarové platformy nedokážou aplikovat všechny funkcionality, které jsou potřebné ke správě produktů a zákazníků a jsou navrženy spíše pro řízení projektů, řízení vývojových týmů a organizace práce. Důležitým bodem, proč některé firmy nevyužívají tyto komplexní a hotová řešení je např. příliš vysoká cena, nadměrný počet nepotřebných funkcionalit a chybějící funkcionality, které jsou důležité pro daný systém. Problémem totiž je, že tyto softwarové platformy nabízejí své služby ve formě předplatných, a ne vždy jsou využity. Ideálním řešením pro firmy je možný výběr funkcionalit, které potřebují pro svůj systém, neboť se může stát, že si mohou pořídit nejlevnější předplatné, ale chybí jim některé zásadní funkcionality, které jsou nabízeny pouze v nejdražší variantě a jsou tedy nuceni přistoupit k nejdražší variantě předplatného.

V oblasti systémů pro správu obsahu může být problémem chybějící znalost potřebná k implementaci tohoto systému. Systémy pro správu obsahu nabízejí již hotová řešení v podobě rozšíření, pluginů a modulů, ale také chybí některé funkcionality, které jsou potřeba implementovat vlastním řešením. V rámci budoucího vývoje a možných změn je jakákoliv úprava a změna existujících funkcí náročná, může být omezena pod licenční smlouvou nebo již nebudou podporované. Výsledkem srovnání existujících řešení s implementací vlastního systému je takový, že odhadovaný čas potřebný pro implementaci systému pro správu zákazníků a produktů je velmi podobný, a to buď v případě řešení pomocí existujícího systému pro správu obsahu, nebo implementací vlastního systému.

V rámci teoretické části je zpracován rozbor použitých technologií a srovnání nejznámějších PHP frameworků. Vybraným PHP frameworkem pro realizaci webové aplikace pro správu zákazníků a produktů je CodeIgniter 4, který má velmi slušnou rychlost, stabilitu, široký výběr knihoven, knihovny pro ochranu vyvíjeného systému, a především je velmi jednoduchý pro začínající programátory.

V praktické části je provedena analýza nového systému pro webovou aplikaci pro správu zákazníků a produktů, která znázorňuje systém v teoretickém pohledu a obsahuje rozbor funkčních a nefunkčních požadavků a způsob realizace pomocí případu užití. Dále je vypracován model tříd, který zachycuje všechny potřebné třídy v systému včetně vazeb mezi sebou a fyzický datový model, který představuje strukturu dat implementovanou jako schéma

relační databáze. Následují sekvenční diagramy, které znázorňují jednotlivé interakce a způsob komunikace mezi jednotlivými třídami a drátěné modely, které zobrazují vizualizaci grafického návrhu pro tento systém.

Další část praktické části se věnuje samotné realizaci, kde je představena samotná aplikace a jsou zde představeny některé základní funkcionality jako např. vytvoření a editace zákazníků (klientů), produktů atd. Dále je představena funkcionality pro vyhledávání fyzických a právnických osob, zaznamenání polohy pomocí našeptávače a interaktivní mapy, systém pro upozornění apod. Kromě některých funkcionalit systému jsou představeny i testovací případy a také je věnována bezpečnosti aplikace, kde jsou představeny některé nejznámější útoky a způsoby zabezpečení komunikace. Jsou popsány principy jednotlivých útoků a způsob obrany v této webové aplikaci.

Tato diplomová práce je vytvořena tak, aby dokázala co nejlépe zjednodušit správu zákazníků a produktů pro firmu zabývající se instalací zmíněných produktů. Webová aplikace je aktuálně ve funkční verzi a v rámci budoucího vývoje se nabízí implementace návrhů pro vylepšení aplikace, vytvoření a rozšíření testovacích případů pomocí automatických testů.

SEZNAM POUŽITÉ LITERATURY

- [1] ARLOW, Jim a Ila NEUSTADT, 2007. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 2., aktualiz. A dopl. Vyd. Brno: Computer Press. ISBN 9788025115039.
- [2] UNHELKAR, Bhuvan, [2018]. *Software engineering with UML*. Boca Raton: CRC Press, Taylor & Francis Group, CRC Press, is an imprint of the Taylor & Francis Group an informa business. ISBN 9781138297432.
- [3] *Official CodeIgniter documentation* [online], 2021. British Columbia Institute of Technology: CodeIgniter Foundation [cit. 2022-02-14]. Dostupné z: https://codeigniter.com/user_guide/intro/index.html
- [4] *Jquery API Documentation* [online], 2006. New York BarCamp: John Resig [cit. 2022-02-14]. Dostupné z: <https://api.jquery.com/>
- [5] *JavaScript Guide* [online], 1995. Netscape Communications Corporation, Mozilla Foundation: Brendan Eich [cit. 2022-02-14]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
- [6] *KURZY DEVIZOVÉHO TRHU – ČESKÁ NÁRODNÍ BANKA: Kurzy devizového trhu* [online]. [cit. 2023-04-13]. Dostupné z: <https://www.cnb.cz/cs/financni-trhy/devizovy-trh/kurzy-devizoveho-trhu/kurzy-devizoveho-trhu/index.html?date=13.04.2023>
- [7] *Ares.cz – ekonomické subjekty: Co je to ARES?* [online]. [cit. 2023-04-24]. Dostupné z: <https://ares.cz/>
- [8] *Asana: Explore ways to use Asana* [online]. [cit. 2023-03-02]. Dostupné z: <https://asana.com/uses>
- [9] *Asana: Pricing* [online]. [cit. 2023-03-05]. Dostupné z: <https://asana.com/pricing>
- [10] *Asana pricing: Select your plan* [online]. [cit. 2023-03-05]. Dostupné z: <https://app.asana.com/0/billing/pricing/1204324155581389>
- [11] *What is Security Assertion Markup Language (SAML): Understanding SAML* [online]. [cit. 2023-04-24]. Dostupné z: <https://www.oracle.com/cz/security/cloud-security/what-is-saml/>
- [12] *Asana: Explore ways to use Asana* [online]. [cit. 2023-03-05]. Dostupné z: <https://asana.com/uses>

- [13] *Freelo: Projektové řízení, správa úkolů a týmová komunikace online – Freelo* [online]. [cit. 2023-04-03]. Dostupné z: <https://www.freelo.io/cs>
- [14] In: *Youtube: Freelo: jak začít pracovat opravdu efektivně* [online]. [cit. 2023-04-03]. Dostupné z: <https://www.youtube.com/watch?v=GQckCgcjs8w>
- [15] *Ceník & tarify – Freelo* [online]. [cit. 2023-04-03]. Dostupné z: <https://www.freelo.io/cs/cenik>
- [16] *What Is WordPress & What Is It Used For: What exactly is WordPress?* [online]. [cit. 2023-04-12]. Dostupné z: https://www.bluehost.com/blog/what-is-wordpress/?psafe_param=1&utm_campaign=affiliate-link_searchgenericpromo_PPC&utm_source=direct&utm_medium=affiliate&utm_affiliate=searchgenericpromo&irpid=101&clickid=P61C101S570N0B5578A2D4499E0000V261&pb=signup_searchgenericpromo&siteid=101&channelid=P61C101S570N0B5578A2D4499E0000V261&gclid=Cj0KCQJw6cKiBhD5ARIsAKXUdyZDvQiyMGAWrtCscJ3CIzDmzLQGf4Sssqrbq8GagNYft-BwPSxs_eDoaAs6gEALw_wcB&gclidsrc=aw.ds
- [17] *What is WooCommerce? A Guide to WordPress e-Commerce: What is WooCommerce?* [online]. [cit. 2023-04-12]. Dostupné z: <https://multilingualpress.org/what-is-woocommerce/>
- [18] *About: our mission* [online]. [cit. 2023-04-12]. Dostupné z: <https://wordpress.org/about/>
- [19] *What is open source software? | Opensource.com: What is open source software?* [online]. [cit. 2023-04-13]. Dostupné z: <https://opensource.com/resources/what-open-source>
- [20] *ACF | Advanced Custom Fields Plugin for WordPress: Content editing made easy* [online]. [cit. 2023-04-12]. Dostupné z: <https://www.advancedcustomfields.com/>
- [21] *Kybernaut IČO DIČ – Plugin Wordpressu* [online]. [cit. 2023-04-12]. Dostupné z: <https://cs.wordpress.org/plugins/woolab-ic-dic/>
- [22] *Systém výměny informací o DPH (VIES) – Your Europe: Systém výměny informací o DPH (VIES)* [online]. [cit. 2023-04-12]. Dostupné z: https://europa.eu/youreurope/business/taxation/vat/check-vat-number-vies/index_cs.htm

- [23] *GitHub – vyskoczilova/kybernavut-ic-dic: Kybernavut IC DIC (WordPress plugin): Přidá IČO a DIČ do formuláře s fakturační adresou ve WooCommerce a rovnou ověří, jestli jsou zadané hodnoty skutečné.: Kybernavut IC DIC (WordPress plugin)* [online]. [cit. 2023-04-12]. Dostupné z: <https://github.com/vyskoczilova/kybernavut-ic-dic>
- [24] *Getting started: What’s AJAX?* [online]. [cit. 2023-04-12]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX/Getting_Started
- [25] *WP Go Maps (formerly WP Google Maps) – Plugin WordPressu | WordPress.org Česko: WP Go Maps (formerly WP Google Maps)* [online]. [cit. 2023-05-21]. Dostupné z: <https://cs.wordpress.org/plugins/wp-google-maps/>
- [26] *Joomla! – Content Management http to build websites & apps: About Joomla!* [online]. [cit. 2023-04-13]. Dostupné z: <https://www.joomla.org/about-joomla.html>
- [27] *GNU General Public License (GPL) | SUSE Defines: GNU General Public License (GPL)* [online]. [cit. 2023-04-13]. Dostupné z: <https://www.suse.com/suse-defines/definition/gnu-general-public-license-gpl/>
- [28] *Sourcerer – Place any code in Joomla! - Regular Labs* [online]. [cit. 2023-04-13]. Dostupné z: <https://regularlabs.com/sourcerer>
- [29] *My Maps location, by JoomUnited – Joomla Extension Directory: Introduction* [online]. [cit. 2023-04-13]. Dostupné z: <https://extensions.joomla.org/extension/maps-a-weather/maps-a-locations/my-maps-location/>
- [30] *My Maps location, Joomla map search extension: My Maps Location for Joomla Extension Prices* [online]. [cit. 2023-04-13]. Dostupné z: <https://www.joomunited.com/products/my-maps-location>
- [31] *GMapFP: Google Map or Openstreet Map, by Fabrice4821 – Joomla Extension Directory: Introduction* [online]. [cit. 2023-04-13]. Dostupné z: <https://extensions.joomla.org/extension/maps-a-weather/maps-a-locations/gmapfp-google-map/>
- [32] *Co je to Man-day?* [online]. [cit. 2023-04-13]. Dostupné z: <https://it-slovník.cz/pojem/man-day>
- [33] *Webhosting - WEDOS.cz: NoLimit* [online]. [cit. 2023-04-13]. Dostupné z: <https://www.wedos.cz/webhosting>
- [34] *Neomezený webhosting | Active24: hosting s WordPressem* [online]. [cit. 2023-04-13]. Dostupné z: <https://www.active24.cz/webhosting>

- [35] *What is PHP?* [online]. [cit. 2023-04-03]. Dostupné z: <https://www.php.net/manual/en/intro-what-is.php>
- [36] *What is PHP (Hypertext Preprocessor)? - TechTarget Definition: PHP (Hypertext Preprocessor)* [online]. [cit. 2023-05-13]. Dostupné z: <https://www.techtarget.com/whatis/definition/PHP-Hypertext-Preprocessor>
- [37] *What Is a Database: Database defined* [online]. [cit. 2023-04-03]. Dostupné z: <https://www.oracle.com/in/database/what-is-database/>
- [38] *MySQL: MySQL 8.0 Reference Manual: 1.2.1 What is MySQL?* [online]. [cit. 2023-05-13]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>
- [39] *METRONIC ADMIN DASHBOARD TEMPLATE: FEATURES AND ADVANTAGES: What is Metronic and what are its features* [online]. [cit. 2023-04-03]. Dostupné z: <https://lvivivity.com/metronic-admin-template>
- [40] PHAM, Anh Vu. *Návrh aplikace pro hodnocení výrobků a služeb* [online]. Zlín, 2020 [cit. 2023-05-13]. Dostupné z: <https://theses.cz/id/qukpdf/>. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky. Vedoucí práce doc. Ing. Petr Šilhavý, Ph.D.
- [41] *Get started with Bootstrap · Bootstrap v5.3: Get started with Bootstrap* [online]. [cit. 2023-04-03]. Dostupné z: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- [42] *What is JavaScript: A high-level definition* [online]. [cit. 2023-04-03]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript#a_high-level_definition
- [43] *What is jQuery?* [online]. [cit. 2023-04-03]. Dostupné z: <https://jquery.com/>
- [44] *What is jQuery?* [online]. [cit. 2023-04-03]. Dostupné z: https://www.w3schools.com/jquery/jquery_intro.asp
- [45] *Unified Modeling Language: What is UML?* [online]. [cit. 2023-04-03]. Dostupné z: https://www.edrawsoft.com/what-is-uml-diagram.html?psafe_param=1&gad=1&gclid=CjwKCAjwuqiiBhBtEiwATgvixDGG25nwsDDczhfOCVY-jce3rqIco_zjAtaBLedwj_kQGxj9RfuJ_DhoC5wgQAvD_BwE
- [46] *The Most Popular PHP Frameworks to Use in 2023: Why Use a PHP Framework?* [online]. [cit. 2023-05-13]. Dostupné z: <https://kinsta.com/blog/php-frameworks/>

- [47] The Most Popular PHP Frameworks to Use in 2023: What You Need to Know Before Using a PHP Framework [online]. [cit. 2023-05-13]. Dostupné z: <https://kinsta.com/blog/php-frameworks/>
- [48] The Most Popular PHP Frameworks to Use in 2023: Model View Controller architecture [online]. [cit. 2023-05-13]. Dostupné z: <https://kinsta.com/blog/php-frameworks/>
- [49] In: The Most Popular PHP Frameworks to Use in 2023: Model View Controller architecture [online]. [cit. 2023-05-13]. Dostupné z: <https://kinsta.com/blog/php-frameworks/>
- [50] The 10 Best PHP Frameworks for Web Development in 2023 - DistantJob - Remote Recruitment Agency: Laravel [online]. [cit. 2023-05-13]. Dostupné z: <https://distantjob.com/blog/best-php-frameworks-web-development/>
- [51] The Most Popular PHP Frameworks to Use in 2023: Laravel [online]. [cit. 2023-05-13]. Dostupné z: <https://kinsta.com/blog/php-frameworks/>
- [52] Introduction – Composer [online]. [cit. 2023-05-13]. Dostupné z: <https://getcomposer.org/doc/00-intro.md>
- [53] The 10 Best PHP Frameworks for Web Development in 2023 - DistantJob - Remote Recruitment Agency: Symfony [online]. [cit. 2023-05-13]. Dostupné z: <https://distantjob.com/blog/best-php-frameworks-web-development/>
- [54] The Most Popular PHP Frameworks to Use in 2023: Symfony [online]. [cit. 2023-05-13]. Dostupné z: <https://kinsta.com/blog/php-frameworks/>
- [55] The 10 Best PHP Frameworks for Web Development in 2023 - DistantJob - Remote Recruitment Agency: CodeIgniter [online]. [cit. 2023-05-13]. Dostupné z: <https://distantjob.com/blog/best-php-frameworks-web-development/>
- [56] The Most Popular PHP Frameworks to Use in 2023: CodeIgniter [online]. [cit. 2023-05-13]. Dostupné z: <https://kinsta.com/blog/php-frameworks/>
- [57] *Functional and Nonfunctional Requirements: Specification and Types: Functional vs nonfunctional requirements* [online]. [cit. 2023-04-03]. Dostupné z: <https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/>

- [58] *Use Case Diagram – UML 2 Tutorial | Sparx Systems: Including Use Cases* [online]. [cit. 2023-04-15]. Dostupné z: <https://sparxsystems.com/resources/tutorials/uml2/use-case-diagram.html>
- [59] *Class Diagram – UML 2 Tutorial | Sparx Systems: Class Diagrams* [online]. [cit. 2023-04-15]. Dostupné z: <https://sparxsystems.com/resources/tutorials/uml2/class-diagram.html>
- [60] *Physical Data Models | Enterprise Architect User Guide: Physical Data Models* [online]. [cit. 2023-04-15]. Dostupné z: https://sparxsystems.com/enterprise_architect_user_guide/14.0/model_domains/physical_data_model.html
- [61] *Three Table Types Relationship (1:1, 1:n, m:n): many-to-many relationship* [online]. [cit. 2023-04-15]. Dostupné z: <https://www.relationaldbdesign.com/database-design/module6/three-relationship-types.php>
- [62] *Use Case Diagram – UML 2 Tutorial | Sparx Systems: Actors* [online]. [cit. 2023-04-15]. Dostupné z: <https://sparxsystems.com/resources/tutorials/uml2/use-case-diagram.html>
- [63] *Use Case Diagram – UML 2 Tutorial | Sparx Systems: Use Cases* [online]. [cit. 2023-04-15]. Dostupné z: <https://sparxsystems.com/resources/tutorials/uml2/use-case-diagram.html>
- [64] *Sequence Diagram – UML 2 Tutorial | Sparx Systems: Sequence Diagrams* [online]. [cit. 2023-04-15]. Dostupné z: <https://sparxsystems.com/resources/tutorials/uml2/sequence-diagram.html>
- [65] FOWLER, Martin, 2009. *Destilované UML*. Praha: Grada. Knihovna programátora (Grada). ISBN 9788024720623.
- [66] *Online Mockup, Wireframe & UI Prototyping Tool · Moqups: Wireframe, Diagram & Whiteboard Online* [online]. [cit. 2023-05-06]. Dostupné z: <https://moqups.com/>
- [67] *Moment.js | Home: Format Dates* [online]. [cit. 2023-05-08]. Dostupné z: <https://momentjs.com/>
- [68] *ARES – Podmínky provozu: Podmínky provozu* [online]. [cit. 2023-05-06]. Dostupné z: https://www.info.mfcr.cz/ares/ares_podminky.html
- [69] *Php – How to get page content using –URL? - Stack Overflow: How to get page content using cURL?* [online]. [cit. 2023-05-06]. Dostupné z: <https://stackoverflow.com/questions/14953867/how-to-get-page-content-using-curl>

- [70] *ARES – XML služby: XML služby ARES* [online]. [cit. 2023-05-06]. Dostupné z: https://www.info.mfcr.cz/ares/ares_xml.html.cz
- [71] *Co je to–XML? - IT Slovník: Co znamená zkratka XML?* [online]. [cit. 2023-05-06]. Dostupné z: <https://it-slovník.cz/pojem/xml>
- [72] *Doplňující informace k službě Basic* [online]. [cit. 2023-05-08]. Dostupné z: https://www.info.mfcr.cz/ares/ares_xml_basic.html.cz
- [73] *Send Email in CodeIgniter 4 With SMTP* [online], 2021. Web: Tuts Make [cit. 2023-02-14]. Dostupné z: <https://www.tutsmake.com/send-email-in-codeigniter-4-with-smtp/>
- [74] *SMTP (Simple Mail Transfer Protocol) Definition: SMTP* [online]. [cit. 2023-05-06]. Dostupné z: <https://techterms.com/definition/smtp>
- [75] *GitHub – CodeSeven/toastr: Simple javascript toast notifications: CodeSeven / toastr* [online]. [cit. 2023-05-06]. Dostupné z: <https://github.com/CodeSeven/toastr>
- [76] *GitHub – thorst/jquery-idletimer: provides you a way to monitor user activity with a page.: jQuery – idletimer* [online]. [cit. 2023-05-15]. Dostupné z: <https://github.com/thorst/jquery-idletimer>
- [77] *CodeIgniter-Ion-Auth/Auth.php at 4 · benedmunds/CodeIgniter-Ion-Auth · GitHub: benedmunds / CodeIgniter-Ion-Auth* [online]. [cit. 2023-05-06]. Dostupné z: <https://github.com/benedmunds/CodeIgniter-Ion-Auth/blob/4/Controllers/Auth.php>
- [78] *CodeIgniter-Ion-Auth/LICENSE at 4 · benedmunds/CodeIgniter-Ion-Auth · GitHub: CodeIgniter-Ion-Auth/LICENSE* [online]. [cit. 2023-05-15]. Dostupné z: <https://github.com/benedmunds/CodeIgniter-Ion-Auth/blob/4/LICENSE>
- [79] *Cron, cron jobs and crontab – explained: What is cron?* [online]. [cit. 2023-05-06]. Dostupné z: <https://crontogo.com/blog/cron-cron-jobs-and-crontab-explained/>
- [80] *GitHub – PHPMailer/PHPMailer: The classic email sending library for PHP: PHP-Mailer* [online]. [cit. 2023-05-15]. Dostupné z: <https://github.com/PHPMailer/PHP-Mailer>
- [81] *API Mapy.cz: Mapy API verze 4.13 – Neil Armstrong* [online]. [cit. 2023-05-06]. Dostupné z: <https://api.mapy.cz/>
- [82] *Google Maps Platform Documentation | Maps JavaScript API | Google Developers: Maps JavaScript API* [online]. [cit. 2023-05-08]. Dostupné z: <https://developers.google.com/maps/documentation/javascript>

- [83] *Quick Start Guide – Leaflet – a JavaScript library for interactive maps: Leaflet Quick Start Guide* [online]. [cit. 2023-05-08]. Dostupné z: <https://leafletjs.com/examples/quick-start/>
- [84] *API Mapy.cz – Ukázky práce s API: Pokročilý našeptávač* [online]. [cit. 2023-05-08]. Dostupné z: <https://api.mapy.cz/view?page=suggestadv>
- [85] *API Mapy.cz – Ukázky práce s API: Asynchronní načtení mapy* [online]. [cit. 2023-05-08]. Dostupné z: <https://api.mapy.cz/view?page=async>
- [86] *JSdoc: SMap.Suggest: Třída SMap.Suggest* [online]. [cit. 2023-05-08]. Dostupné z: <https://api.mapy.cz/doc/SMap.Suggest.html>
- [87] *API Mapy.cz – Ukázky práce s API: Reverzní geokódování* [online]. [cit. 2023-05-08]. Dostupné z: <https://api.mapy.cz/view?page=rgeocoding>
- [88] *API Mapy.cz – Ukázky práce s API: Přesouvateľná značka* [online]. [cit. 2023-05-08]. Dostupné z: <https://api.mapy.cz/view?page=moving-marker>
- [89] *Google Maps Platform Documentation | Maps JavaScript API | Google Developers: Maps JavaScript API* [online]. [cit. 2023-05-08]. Dostupné z: <https://developers.google.com/maps/documentation/javascript>
- [90] *Platform Pricing & API Costs – Google Maps Platform: Dynamic Maps+* [online]. [cit. 2023-05-08]. Dostupné z: <https://mapsplatform.google.com/pricing/>
- [91] *Leaflet – a JavaScript library for interactive maps* [online]. [cit. 2023-05-08]. Dostupné z: <https://leafletjs.com/index.html>
- [92] *DropzoneJS & Bootstrap Integration by Keenthemes* [online]. [cit. 2023-05-10]. Dostupné z: <https://preview.keenthemes.com/html/metronic/docs/forms/dropzonejs>
- [93] *Resize PNG image in PHP. Stackoverflow* [online]. [cit. 2023-02-14]. Dostupné z: <https://stackoverflow.com/questions/5064839/resize-png-image-in-php>
- [94] *PHP: GD – Manual* [online]. [cit. 2023-05-10]. Dostupné z: <https://www.php.net/manual/en/book.image.php>
- [95] *Lightbox2* [online]. [cit. 2023-05-10]. Dostupné z: <https://lokeshdhakar.com/projects/lightbox2/#getting-started>
- [96] *Image Lightbox & Gallery For jQuery – Lightbox2 | Free jQuery Plugins* [online]. [cit. 2023-05-10]. Dostupné z: <https://www.jqueryscript.net/lightbox/lightbox2.html>

- [97] Eagri – Data ke stažení [online]. [cit. 2023-05-23]. Dostupné z: <https://eagri.cz/ssl/nosso-app/DataKeStazeni/Okresy>
- [98] GitHub – gunsoft/obce-okresy-kraje-slovenska: Obce, okresy a kraje v SR [online]. [cit. 2023-05-23]. Dostupné z: <https://github.com/gunsoft/obce-okresy-kraje-slovenska>
- [99] What is Authentication? [online]. [cit. 2023-05-13]. Dostupné z: <https://www.techtarget.com/searchsecurity/definition/authentication>
- [100] HTTP vs HTTPS‘ What's the Difference? - javatpoint: HTTP vs HTTPS [online]. [cit. 2023-05-13]. Dostupné z: <https://www.javatpoint.com/http-vs-https>
- [101] In: HTTP vs HTTPS‘ What's the Difference? - javatpoint: Main difference between the HTTP and HTTPS [online]. [cit. 2023-05-14]. Dostupné z: <https://www.javatpoint.com/http-vs-https>
- [102] What is SSL, TLS and HTTPS? | DigiCert: What is an SSL Certificate? [online]. [cit. 2023-05-14]. Dostupné z: <https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https>
- [103] What is SSL/TLS Encryption? | F5: How Does SSL/TLS Encryption Work? [online]. [cit. 2023-05-14]. Dostupné z: <https://www.f5.com/glossary/ssl-tls-encryption>
- [104] What is a SQL Injection? | Definition from TechTarget: SQL injection (SQLi) [online]. [cit. 2023-05-14]. Dostupné z: <https://www.techtarget.com/searchsoftwarequality/definition/SQL-injection>
- [105] In: What is SQL Injection Attack? Definition & FAQs | Avi Networks: SQL Injection Attack Definition [online]. [cit. 2023-05-14]. Dostupné z: <https://avinetworks.com/glossary/sql-injection-attack/>
- [106] Function – How to avoid SQL injection in CodeIgniter? - Stack Overflow: How to avoid SQL injection in CodeIgniter? [online]. [cit. 2023-05-14]. Dostupné z: <https://stackoverflow.com/questions/5857386/how-to-avoid-sql-injection-in-codeigniter/5857481#5857481>
- [107] Query Builder Class — CodeIgniter 4.3.4 documentation: SQL Injection Protection [online]. [cit. 2023-05-14]. Dostupné z: https://codeigniter.com/user_guide/database/query_builder.html

- [108] Cross Site Request Forgery (CSRF): Overview [online]. [cit. 2023-05-14]. Dostupné z: <https://owasp.org/www-community/attacks/csrf>
- [109] What is Social Engineering? | IBM: What is social engineering? [online]. [cit. 2023-05-14]. Dostupné z: <https://www.ibm.com/topics/social-engineering>
- [110] In: CSRF Attack: Cross-Site Request Forgery Definition & Defense | Okta: How Do CSRF Attacks Work? [online]. [cit. 2023-05-14]. Dostupné z: <https://www.okta.com/identity-101/csrf-attack/>
- [111] What Is Cross-Site Request Forgery (CSRF) and How Does It Work? | Synopsys: How can an application prevent a Cross-Site Request Forgery attack? [online]. [cit. 2023-05-14]. Dostupné z: <https://www.synopsys.com/glossary/what-is-csrf.html>
- [112] Security — CodeIgniter 4.3.4 documentation: Cross-Site Request Forgery (CSRF) [online]. [cit. 2023-05-14]. Dostupné z: https://www.codeigniter.com/user_guide/libraries/security.html#cross-site-request-forgery-csrf
- [113] What is Cross-Site Scripting (XSS)? How to Prevent and Fix It: What is cross-site scripting (XSS)? [online]. [cit. 2023-05-15]. Dostupné z: <https://www.techtarget.com/searchsecurity/definition/cross-site-scripting>
- [114] In: What is Cross-Site Scripting (XSS)? How to Prevent and Fix It: How does cross-site scripting work? [online]. [cit. 2023-05-15]. Dostupné z: <https://www.techtarget.com/searchsecurity/definition/cross-site-scripting>
- [115] What is Cross-Site Scripting (XSS)? How to Prevent and Fix It: How does cross-site scripting work? [online]. [cit. 2023-05-15]. Dostupné z: <https://www.techtarget.com/searchsecurity/definition/cross-site-scripting>
- [116] What is a Cookie? [online]. [cit. 2023-05-15]. Dostupné z: <https://www.techtarget.com/searchsoftwarequality/definition/cookie>
- [117] What is QR code (quick response code)? | Definition from TechTarget: QR code (quick response code) [online]. [cit. 2023-05-15]. Dostupné z: <https://www.techtarget.com/whatis/definition/QR-code-quick-response-code>
- [118] VIES (VAT Information Exchange System): Taxation and Customs Union [online]. [cit. 2023-05-15]. Dostupné z: https://taxation-customs.ec.europa.eu/vies-vat-information-exchange-system_en

- [119] What Is an API (Application Programming Interface): DEFINITION application programming interface (API) [online]. [cit. 2023-05-15]. Dostupné z: <https://www.techtarget.com/searchapparchitecture/definition/application-program-interface-API>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

SAAS	Software as Service
CMS	Content Management System
ARES	Administrativní registr ekonomických subjektů
IČO	Identifikační číslo osoby
DIČ	Daňové identifikační číslo
SAML	Security Assertion Markup Language
PSC	Poštovní směrovací číslo
VIIES	VAT Information Exchange System [118]
XML	eXtensible Markup Language
URL	Uniform Resource Locator
PHP	Hypertext Preprocessor
HTML	Hypertext Markup Language
CSS	Cascade Style Sheets
UML	Unified Modelling Language
API	Application Programming Interface [119]
AJAX	Asynchronous JavaScript And XML
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
SSL	Secure Socket Layer
TLS	Transfer Layer Security
SQL	Structured Query Language
CSRF	Cross-Site Request Forgery
XSS	Cross-Site Scripting
QR	Quick Response code

SEZNAM OBRÁZKŮ

Obr. 1: Detail produktu – Asana	17
Obr. 2: Přidání uživatele – Asana	18
Obr. 3: Přidání nového sloupce – Asana.....	19
Obr. 4: Přehled verzí předplatných – Asana	21
Obr. 5: Nastavení workflow – Asana.....	23
Obr. 6: Ukázkový projekt – Freelo	25
Obr. 7: Výběr nebo přidání klienta – Freelo	27
Obr. 8: Přidání nového produktu – Freelo	27
Obr. 9: Přidání nového uživatele – Freelo	28
Obr. 10: Přidání nového uživatele odkazem – Freelo.....	29
Obr. 11: Cenová nabídka předplatných – Freelo	30
Obr. 12: Vytvoření šablony projektu – Freelo	32
Obr. 13: Způsob fakturace – Freelo	33
Obr. 14: Nastavení připojení databáze – Wordpress	35
Obr. 15: Vzhled administrace – Wordpress.....	36
Obr. 16: Plugin ACF – typy polí.....	36
Obr. 17: Plugin ACF – Nastavení pravidla pro skupinu polí.....	37
Obr. 18: Vytvoření stránky pro vytvoření nového klienta.....	37
Obr. 19: Formulář pro vytvoření zákazníka.....	38
Obr. 20: Přidání nového uživatele do systému – Wordpress	40
Obr. 21: Nastavení připojení databáze – Joomla	44
Obr. 22: Vzhled systému – Joomla	44
Obr. 23: Přehled vytvořených stránek – Joomla.....	45
Obr. 24: Vytvoření menu – Joomla	46
Obr. 25: Nastavení typu menu a oprávnění – Joomla.....	46
Obr. 26: Úvodní stránka s přihlašovacím formulářem – Joomla	47
Obr. 27: Vzhled menu po přihlášení do systému – Joomla	47
Obr. 28: Způsob vložení zdrojového kódu v nastavení stránky – Joomla.....	48
Obr. 29: Formulář pro vytvoření uživatele – Joomla.....	48
Obr. 30: Nastavení uživatelských práv – Joomla	49
Obr. 31: Interakce Model-View-Controller	58
Obr. 32: Funkční požadavky pro správu uživatelů	62

Obr. 33: Funkční požadavky pro správu produktů	62
Obr. 34: Funkční požadavky pro správu klientů.....	63
Obr. 35: Nefunkční požadavky pro systém a bezpečnost	64
Obr. 36: Realizace požadavků č. 1	65
Obr. 37: Realizace požadavků č. 2	66
Obr. 38: Realizace požadavků č. 3	67
Obr. 39: Aktéři.....	69
Obr. 40: Model případu užití	70
Obr. 41: Případy užití pro správu klientů	70
Obr. 42: Případy užití pro správu produktů	71
Obr. 43: Případy užití pro správu uživatelů	72
Obr. 44: Sekvenční diagram: Přihlášení uživatele.....	82
Obr. 45: Sekvenční diagram: Vytvoření nového klienta	83
Obr. 46: Sekvenční diagram: Vytvoření nového klienta – alternativní scénář.....	84
Obr. 47: Sekvenční diagram: Vyhledávání fyzických a právnických osob	84
Obr. 48: Sekvenční diagram: Editace produktu.....	85
Obr. 49: Sekvenční diagram: Vytvoření nového upozornění	86
Obr. 50: Sekvenční diagram: Změna přístupového hesla	87
Obr. 51: Formulář pro vytvoření nového uživatele	90
Obr. 52: Výsledek po úspěšném zpracování ARES požadavku	94
Obr. 53: Způsob přidání produktu do systému	95
Obr. 54: Formulář pro přidání nového produktu	96
Obr. 55: Editace klienta – možnost přidání upozornění	97
Obr. 56: Vytvoření nového upozornění	97
Obr. 57: Nastavení Cronu pro kontrolu a odeslání upozornění	99
Obr. 58: Výpis štítku kategorie – Digitální úřední deska	104
Obr. 59: Editace vytvořeného štítku	105
Obr. 60: Formulář pro nahrávání souboru	106
Obr. 61: Rozdíl mezi protokoly HTTP a HTTPS	129
Obr. 62: Princip SQL injection	131
Obr. 63: Princip fungování CSRF.....	132
Obr. 64: Princip Cross-Site scripting.....	134
Obr. 65: Příklad vložení kódu pomocí jazyku Javascript do vstupního pole	135

Obr. 66: Výsledek uložení javascriptového kódu135

SEZNAM TABULEK

Tab. 1: Výsledky zkoumaných funkcí aplikace – Asana.....	19
Tab. 2: Výsledky zkoumaných funkcí aplikace – Freelo.....	29
Tab. 3: Výsledky zkoumaných funkcí aplikace – Wordpress	42
Tab. 4: Výsledky zkoumaných funkcí aplikace – Joomla	50
Tab. 5: Výsledky zkoumaných funkcí a metod aplikace – shrnutí.....	52
Tab. 6: Cenové a časové srovnání technologií	54
Tab. 7: Scénář případu užití: Primární scénář: Přihlášení uživatele	73
Tab. 8: Scénář případu užití: Alternativní scénář: Přihlášení uživatele.....	73
Tab. 9: Scénář případu užití: Primární scénář: Vytvoření nového klienta.....	74
Tab. 10: Scénář případu užití: Alternativní scénář: Vytvoření nového klienta	75
Tab. 11: Scénář případu užití: Primární scénář: Vyhledávání fyzických a právnických osob	76
Tab. 12: Scénář případu užití: Alternativní scénář: Vyhledávání fyzických a právnických osob.....	77
Tab. 13: Scénář případu užití: Primární scénář: Editace produktu	77
Tab. 14: Scénář případu užití: Alternativní scénář: Editace produktu.....	78
Tab. 15: Scénář případu užití: Primární scénář: Vytvoření nového upozornění	79
Tab. 16: Scénář případu užití: Primární scénář: Změna přístupového hesla	80
Tab. 17: Scénář případu užití: Alternativní scénář: Změna přístupového hesla.....	81
Tab. 18: Test case pro přihlášení uživatele – manuální test	111
Tab. 19: Test case pro vytvoření klienta – manuální test	112
Tab. 20: Test case pro editaci klienta – manuální test	113
Tab. 21: Test case pro vytvoření produktu – manuální test.....	115
Tab. 22: Test case pro editaci produktu – manuální test	116
Tab. 23: Test case pro vytvoření uživatele – manuální test.....	118
Tab. 24: Test case pro editaci uživatele – manuální test	119
Tab. 25: Test case pro vytvoření upozornění – manuální test	121
Tab. 26: Test case pro editaci upozornění – manuální test.....	122
Tab. 27: Test case pro vyhledávání fyzických a právnických osob – manuální test	124
Tab. 28: Test case pro zaznamenání polohy produktu – manuální test	125
Tab. 29: Test case pro změnu stavu produktu – manuální test.....	127

SEZNAM ZDROJOVÝCH KÓDŮ

Zdrojový kód 1: Ukázka formuláře pro přidání klienta	39
Zdrojový kód 2: Ukázka připojení databáze	39
Zdrojový kód 3: Validace vstupu uživatele – ARES	91
Zdrojový kód 4: Odeslání a zpracování AJAX požadavku	92
Zdrojový kód 5: Odeslání a zpracování požadavku cURL	94
Zdrojový kód 6: Nastavení intervalu nečinnosti uživatele	98
Zdrojový kód 7: Asynchronní načtení mapy a nastavení středu mapy	100
Zdrojový kód 8: Nastavení parametrů našeptávače	101
Zdrojový kód 9: Detekce kliku a získání dat o poloze	102
Zdrojový kód 10: Nastavení signálů pro detekci začátku a konce tažení značky.....	102
Zdrojový kód 11: Inicializace mapy od společnosti Google	103
Zdrojový kód 12: Inicializace mapy pomocí Leaflet	104
Zdrojový kód 13: Nastavení funkcionality Dropzone	107
Zdrojový kód 14: Způsob nastavení odkazů pro zobrazení obrázku pomocí knihovny Lightbox	107
Zdrojový kód 15: Nastavení tabulek v konstruktoru třídy FileModel	108
Zdrojový kód 16: SQL příkaz SELECT pro získání cesty k souborům konkrétního klienta	108
Zdrojový kód 17: Příprava pole pro vložení záznamu do databáze.....	108
Zdrojový kód 18: Příklad vložení záznamu do databáze	109
Zdrojový kód 19: Příklad pro editaci konkrétního záznamu v databázi	109
Zdrojový kód 20: Příklad odstranění záznamu z databáze	109
Zdrojový kód 21: Způsob čtení souboru CSV a import okresů do databáze	110
Zdrojový kód 22: Ukázka SQL dotazu při doplnění škodlivého kódu	131
Zdrojový kód 23: Zapnutí CSRF filtru	133
Zdrojový kód 24: Nastavení náhodného generování CSRF tokenu	133
Zdrojový kód 25: Generování CSRF tokenu	133
Zdrojový kód 26: Způsob validace řetězce	134
Zdrojový kód 27: Odstranění nepovolených symbolů – validace	135

SEZNAM PŘÍLOH

PI	CD
PII	Vzhled aplikace Asana
PIII	Správa produktů v aplikace Asana
PIV	Vytvoření pravidla v aplikaci Asana
PV	Nastavení pravidla s externími aplikacemi
PVI	Vzhled aplikace Freelo
PVII	Vytváření značky podle GPS
PVIII	Vytváření podle zeměpisné délky a šířky
PIX	Drátěný model pro úvodní stránku aplikace
PX	Drátěný model pro výpis produktů
PXI	Drátěný model pro nastavení uživatelského účtu
PXII	Drátěný model pro detail klienta
PXIII	Drátěný model pro detail produktu
PXIV	Vzhled webové aplikace

PŘÍLOHA PI: CD

Příložené CD obsahuje tyto soubory:

- Ilustrační obrázky → komprimovane_obrazky.zip
- EA projekt ve formátu qea → Sprava-zakazniku-produktu.qea
- Diplomová práce ve formátu PDF/A → fulltext.pdf
- Zdrojové kódy aplikace → dputb_zdrojove_kody.zip

PŘÍLOHA PII: VZHLED APLIKACE ASANA

The screenshot displays the Asana interface for a project titled "Správa zákazníků a produktů". The interface includes a left sidebar with navigation options like Home, My Tasks, and various team management tools. The main content area shows a list view of tasks, organized into sections: "Poslaná nabídka", "Podepsaná smlouva", "Čeká na objednávku", "Složeno ve skladě", and "Nainstalováno". Each task row contains columns for task name, assignee, and various identifiers. The "Složeno ve skladě" section includes summary rows with numerical values.

Project: Správa zákazníků a produktů On track

Navigation: Overview, List, Board, Timeline, Calendar, Workflow, Dashboard, Messages, Files

Task List:

Task name	Assignee	ICO	DIC	Náklady	Výnos	Datum podp...	Datum instal...	Priority	Due date	Telefon	E-mail	
▼ Poslaná nabídka												
✓ Zákazník 1	Anh Vu Pham	02154986	CZ02154986							+420775487965	zakaznik1@email....	
Add task...												
▼ Podepsaná smlouva												
✓ Zákazník 2	Anh Vu Pham	05643128	CZ05643128							+421754268745	zakaznik2@email....	
✓ Zákazník 3	Anh Vu Pham	03215487	CZ03215487							+420777698542	zakaznik3@email....	
Add task...												
▼ Čeká na objednávku												
✓ Zákazník 4	Anh Vu Pham	02156489	CZ02156489							+420774548796	zakaznik4@email....	
Add task...												
▼ Složeno ve skladě												
✓ Zákazník 5	Anh Vu Pham	03251584	CZ03251584							+420777454896	zakaznik5@email....	
Add task...												
SUM 90 000,00 ... SUM 175 000,00...												
▼ Nainstalováno												
✓ Zákazník 6	Anh Vu Pham	06584875	CZ06584875							+420777555222	zakaznik6@email....	
Add task...												
SUM 75 000,00 ... SUM 150 000,00...												
+ Add section												

PŘÍLOHA PIII: SPRÁVA PRODUKTŮ V APLIKACI ASANA

Správa zákazníků a produktů ▼ ☆
On track

AP AP
Share

Search

+

Trial: 10 days left
Add billing info

AP

Overview **List** Board Timeline Calendar Workflow Dashboard Messages Files

+ Add task
Filter
Sort
Show fields

Public link: Off
Customize

Task name	Assignee	ICO	DIC	Náklady	Výnos	Datum podp...	Datum instal...	Priority	Due date	Telefon	E-mail
▶ ✔ Zákazník 1 2 🗨️	AP Anh Vu Pham	02154986	CZ02154986						Today – 28 Apr	+420775487965	zakazni
Add task...											
▼ Podepsaná smlouva											
▶ ✔ Zákazník 2 2 🗨️	AP Anh Vu Pham	05643128	CZ05643128							+421754268745	zakazni
▶ ✔ Zákazník 3 2 🗨️	AP Anh Vu Pham	03215487	CZ03215487							+420777898542	zakazni
Add task...											
▼ Čeká na objednávku											
▶ ✔ Zákazník 4 2 🗨️	AP Anh Vu Pham	02156489	CZ02156489							+420774548796	zakazni
Add task...											
▼ Složeno ve skladě											
▶ ✔ Zákazník 5 2 🗨️	AP Anh Vu Pham	03251584	CZ03251584							+420777454896	zakazni
Add task...											
SUM 90 000,00 ... SUM 175 000,00...											
▼ Nainstalováno											
▼ ✔ Zákazník 6 2 🗨️	AP Anh Vu Pham	06584875	CZ06584875							+420777555222	zakazni
✔ Produkt 11											
				45 000,00 CZK	85 000,00 CZK	20 Mar	5 Apr	Medium	10 Apr		
✔ Produkt 12											
				30 000,00 CZK	65 000,00 CZK	28 Feb	3 Apr	High	7 Apr		
Add task...											
SUM 75 000,00 ... SUM 150 000,00...											

PŘÍLOHA PIV: VYTVOŘENÍ PRAVIDLA V APLIKACI ASANA

Add rule

• Active Runs on tasks

Moved to section

Nainstalováno

→ Task moved to a section
Nainstalováno

+ Add action

Add action

Add an action that occurs as a result of the rule.

Search actions

Actions External actions

Move task

- Move to a section
- + Move or add to a project

Set task field to...

- Set assignee to...
- Set due date to...

Set custom field to...

- Set Priority to...

Create new

- Create task
- Create subtasks
- Create approvals

Convert task to...

- Convert task to project

Set task status to...

- Set completion status to...

Add to task

- Add comment
- Add collaborator

Back Create rule

PŘÍLOHA PV: NASTAVENÍ PRAVIDLA S EXTERNÍMI APLIKACEMI

Add rule

• Active Runs on tasks

Moved to section

Nainstalováno

→ Task moved to a section
Nainstalováno

+ Add action

Add action

Add an action that occurs as a result of the rule.

Search actions

Actions External actions

- Slack
 - Send channel message
 - Send direct message
- Microsoft Teams
 - Send channel message
- Gmail Rules
 - Send email
- Google Calendar
 - Create a calendar event
- Dropbox
 - Upload attachments
- Notion
 - Create page
- Confluence
 - Create page
- OneDrive
 - Upload attachments
- Mailchimp
 - Add campaign report to task

Back Create rule

PŘÍLOHA PVI: VZHLED APLIKACE FREELO

Správa zákazníků a produktů

Termin

Přidat štítek

Kapitánem projektu jsi ty. Klient: Univerzita Tomáše Bati ve Zlíně (Univerzita Tomáše Bati ve Zlíně)

Home To-Do listy Poznámky Uživatelé Soubory Kalendář úkolů Výkazy Fakturace Mind mapa Timeline



12:00h (40%) ▼ 18:00h (60%) 30:00h

23 500 Kč (47%) ▼ 26 500 Kč (53%) 50 000 Kč

To-Do listy +

Poslaná nabídka

+ Přidat úkol

Podepsaná smlouva

1 500 Kč 1:00h

Digitální deska 55 PN čvn 1.

Podepsáno - 20.04.2023 1 500 Kč 1:00 h

+ Přidat úkol

Čeká na objednávku

+ Přidat úkol

Složeno ve skladě

+ Přidat úkol

Nainstalováno

22 000 Kč 11:00h

Webová stránka - Digitální deska 1 PN

22 000 Kč 11:00 h

+ Přidat úkol

Připnuté informace +

Poslední aktivity u projektu

- před 3 hod Jsi upravil štítky úkolu: Digitální deska 55 i
- před 3 hod Jsi změnil termín úkolu: Digitální deska 55 i
- před 3 hod Jsi upravil prioritu úkolu: Digitální deska 55 i
- před 3 hod Jsi upravil prioritu úkolu: Webová stránka - Digitální deska i

PŘÍLOHA PVII: VYTVÁŘENÍ ZNAČKY PODLE GPS

< Editor map

PRO Přidání pokročilých značek podporováno ve verzi PRO

Upravit existující značky

- K úpravě informací o této značce použijte pole v editoru značek
- Kliknutím a přetažením upravte polohu značek.

Adresa/GPS
Zlín

Animace
Žádný

Infookno otevřít ve výchozím nastavení
Ne

Nadpis
Nadpis

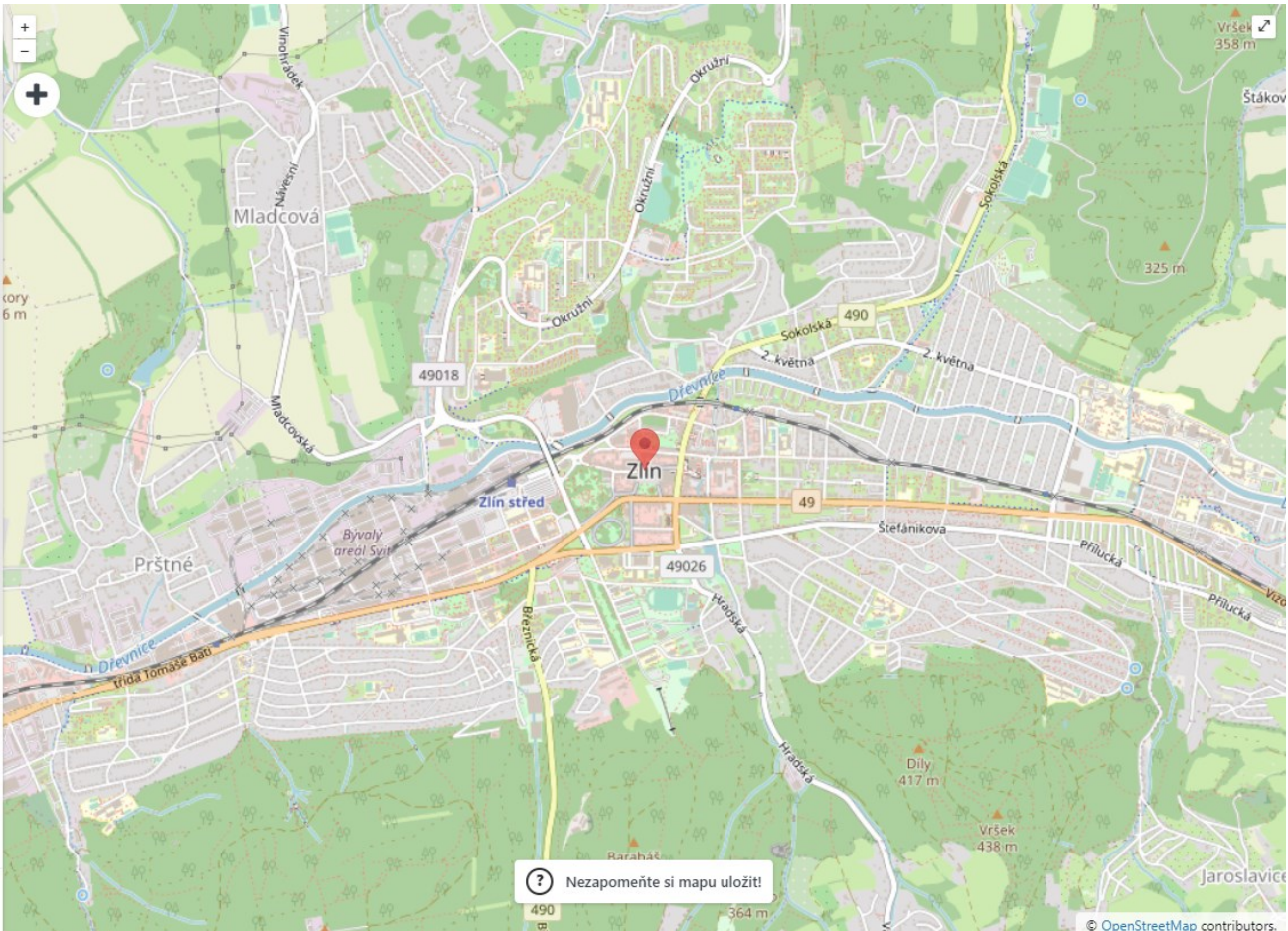
Popis

Galerie

URL odkazu

Get Shortcode

Uložit značku



© OpenStreetMap contributors.

PŘÍLOHA PVIII: VYTVÁŘENÍ ZNAČKY PODLE ZEMĚPISNÉ DÉLKY A ŠÍŘKY

Editor map

PRO Přidání pokročilých značek podporováno ve verzi PRO

! Upravit existující značky

- K úpravě informací o této značce použijte pole v editoru značek
- Kliknutím a přetažením upravte polohu značek.

Nastavení polohy

Nyní můžete přetažením značky upravit polohu bez aktualizace pole adresy. Úprava polí je v tomto režimu zakázána

Pozice

Nezapomeňte si mapu uložit!

© OpenStreetMap contributors.

Get Shortcode

Uložit značku

PŘÍLOHA PIX: DRÁTĚNÝ MODEL PRO ÚVODNÍ NÁSTĚNKU APLIKACE

The wireframe shows a dashboard application with a dark blue sidebar on the left and a main content area on the right. The sidebar contains a menu with the following items:

- Dashboard
- Klienti
 - Výpis klientů
 - Vytvořit klienta
- Produkty
 - Výpis produktů
 - Vytvořit produkt
- Nástroje
 - Typy produktů
 - Kategorie
 - Upozornění
 - Typy souborů
 - Výpis typů souboru
 - Vytvořit typ souboru
 - Štítky produktů

The main content area is titled "Dashboard" and "Domů - Dashboard". It features a large map placeholder labeled "MAP". Below the map are two panels:

Upozornění

Typ upozornění
Název klienta
Název produktu
Datum upozornění
Kdo vytvořil upozornění

Rychlé akce

Klienti Kontaktní osoby Produkty

Název	Kategorie produktu	Klient	Stav
Název produktu	Kategorie produktu	Název klienta	Stav produktu
Název produktu	Kategorie produktu	Název klienta	Stav produktu
Název produktu	Kategorie produktu	Název klienta	Stav produktu
Název produktu	Kategorie produktu	Název klienta	Stav produktu
Název produktu	Kategorie produktu	Název klienta	Stav produktu

PŘÍLOHA PX: DRÁTĚNÝ MODEL PRO VÝPIS PRODUKTŮ

Editace klienta


Domů - Produkty Seřadit dle:

[+ Vytvořit produkt](#) [Filtr](#)

Zobrazit záznamů na stránce


#	Název produktu	Aktivní	Typ produktu	Kategorie	Klient	Datum vytvoření	Nástroje
1	Název produktu	Aktivní	Název typu	Název kategorie	Název klienta	Datum vytvoření	
2	Název produktu	Aktivní	Název typu	Název kategorie	Název klienta	Datum vytvoření	
3	Název produktu	Aktivní	Název typu	Název kategorie	Název klienta	Datum vytvoření	
4	Název produktu	Aktivní	Název typu	Název kategorie	Název klienta	Datum vytvoření	
5	Název produktu	Aktivní	Název typu	Název kategorie	Název klienta	Datum vytvoření	
6	Název produktu	Aktivní	Název typu	Název kategorie	Název klienta	Datum vytvoření	
7	Název produktu	Aktivní	Název typu	Název kategorie	Název klienta	Datum vytvoření	

PŘÍLOHA PXI: DRÁTĚNÝ MODEL PRO NASTAVENÍ UŽIVATELSKÉHO ÚČTU



Nastavení uživatele

Domů - Uživatel - Nastavení uživatele





Nastavení uživatele

Jméno [Ⓜ] Jméno [Ⓜ]

E-mail [Ⓜ]

Avatar



Staré heslo

Nové heslo

Zopakujte nové heslo

Permanentní heslo

PŘÍLOHA PXII: DRÁTĚNÝ MODEL PRO DETAIL KLIENTA

Editace klienta

Domů - Klient - Editace klienta

Dashboard

- Klienti
 - Výpis klientů
 - Vytvořit klienta
- Produkty
 - Výpis produktů
 - Vytvořit produkt
- Nástroje
 - Typy produktů
 - Kategorie
 - Upozornění
- Typy souborů
 - Výpis typů souboru
 - Vytvořit typ souboru
- Štítky produktů

Název klienta [Upravit](#)

Detail ^

Adresa
ulice
město
kraj
IČO
00000000
DIČ
CZ00000000
Datum založení
xx.xx.xxxx xx:xx
Poslední úprava
xx.xx.xxxx xx:xx
Poznámka

Logs

Operace
Čas: xxxxxxxx xxxx jméno a příjmení

Jméno kontaktní osoby
funkce

Kontaktní informace

E-mail
xxxxx@xxxx.xx
Telefon
+420 000 000 000
Poznámka

Upozornění

Typ upozornění
název
Datum upozornění:
xx.xx.xxxx
Stav
Název produktu
Založil


Nahrané soubory

	Název souboru	Typ souboru	Poslední	Autor změn	Nástroje
	Název souboru	Tvo souboru	xx.xx.xxxx	Jméno	
	Název souboru	Tvo souboru	xx.xx.xxxx	Jméno	

Informace o produktu ^ [Upravit produkt](#) [Smazal produkt](#)

Název produktu	Název produktu	Typ produktu	Typ produktu
Kategorie	Kategorie	Adresa HW	Adresa HW
Latitude	0.00000000	Adresa SW	Adresa SW
Longitude	0.00000000	Poznámka	
Status	Aktivní	Stav	Poslaná nabídka

PŘÍLOHA PXIII: DRÁTĚNÝ MODEL PRO DETAIL PRODUKTU



Dashboard

- Klienti
 - Výpis klientů
 - Vytvořit klienta
- Produkty
 - Výpis produktů
 - Vytvořit produkt
- Nástroje
 - Typy produktů
 - Kategorie
 - Upozornění
 - Typy souborů
 - Výpis typů souboru
 - Vytvořit typ souboru
 - Štítky produktů

Editace klienta

Domů - Produkt - Detail produktu

Informace o produktu ^
Upravit produkt
Smazat produkt

Název produktu	Název produktu	Typ produktu	Typ produktu
Kategorie	Kategorie	Adresa HW	Adresa HW
Latitude	0.00000000	Adresa SW	Adresa SW
Longitude	0.00000000	Poznámka	
Status	Aktivní	Stav	Poslaná nabídka

Sockety ^

ID	Aktivní	E-maily	Telefonní čísla
X	Neaktivní	xxxxxxx@xxxxxx.xx	+420 000 000 000

Termíny ^

Datum podpisu smlouvy	Lhůta k dodání	Datum dodání	Záruka HW	Záruka SW
xx.xx.xxxx	X - xx.xx.xxxx	xx.xx.xxxx	xx.xx.xxxx / 10 dní	xx.xx.xxxx / -5 dní

Náklady a výnos ^







Typ	Produkt	Kotva	Doprava	Dodání	Ostatní	Celkem
Náklady	10	15	0	0	0	0
Zisk	20	10	0	0	0	0
Rozdíl	10	-5	0	0	0	0

Logs ^

Operace

Čas: xx.xx.xxxx xxxx jméno a příjmení


Soubory

Název souboru	Typ souboru	Poslední změna	Autor změny	Nástroje
 Název souboru	Typ souboru	xx.xx.xxxx xx:xx	Jméno	 
 Název souboru	Typ souboru	xx.xx.xxxx xx:xx	Jméno	 

Přidat

Název produktu
název klienta
Latitude, longitude

Mapa - pozice



PŘÍLOHA PXIV: VZHLED WEBOVÉ APLIKACE

OBEOS4

Dashboard

Domů - Dashboard

Dashboard

KLIENTI

Klienti

PRODUKTY

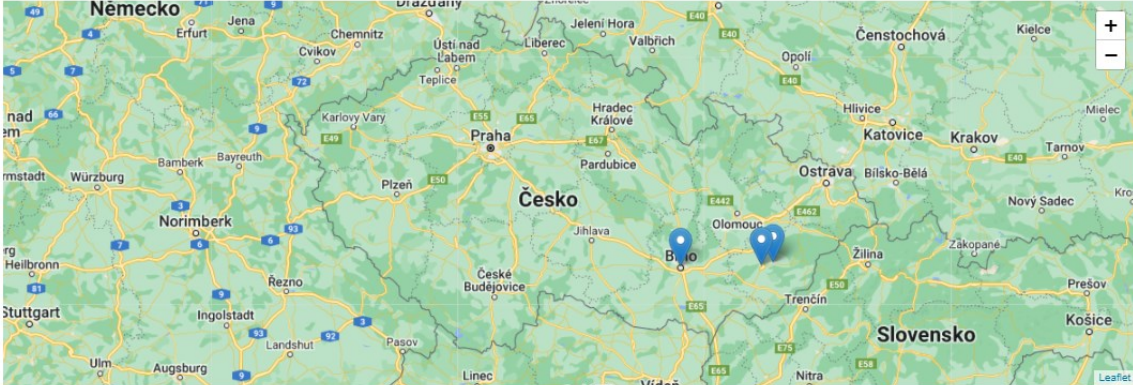
Produkty

Nástroje

UŽIVATELE

Uživatelé

Přehled produktů



Map showing the Czech Republic and surrounding areas (Germany, Slovakia). Key cities labeled include Praha, Brno, Olomouc, Ostrava, and others. The map includes zoom controls and a Leaflet logo.

Upozornění

Lhůta k dodání

Univerzita Tomáše Bati ve Zlíně

Outdoor 55

Lhůta k dodání

15. 05. 2023

Anh Vu Pham

Končí zítra

90 dní do konce SW podpory

Rychlé akce

Klienti Kontaktní osoby **Produkty**

Název	Kategorie produktu	Klient	Stav	Stav socketu
Outdoor 55 OUTDOOR	Digitální úřední deska	Univerzita Tomáše Bati ve Zlíně	Vyzvednuto v PT	Neaktivní
Outdoor FAI OUTDOOR	Digitální úřední deska	Univerzita Tomáše Bati ve Zlíně	Poslané nabídka	Neaktivní
Outdoor 55	Digitální úřední deska	Vysoké učení technické v Brně	Poslané nabídka	Neaktivní