

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Tvorba webové aplikace pro inventarizaci zařízení
Bakalářská práce

Autor: Ondřej Beran
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Martina Husáková, Ph.D.

Hradec Králové

duben 2024

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 20.4.2024

.....
Ondřej Beran

Poděkování:

Děkuji vedoucí bakalářské práce Ing. Martině Husákové, Ph.D. za metodické vedení práce a veškeré rady. Dále bych chtěl poděkovat i všem pedagogům, kteří mě naučili všechny znalosti, které jsem mohl použít pro tvorbu této práce.

Abstrakt

Hlavním cílem této bakalářské práce bylo vytvoření funkční webové aplikace, která slouží jak k prezentaci informací o firmě, tak i k inventarizaci zařízení spravovaných firem. Pro vývoj aplikace byly využity programovací jazyky HTML, PHP, JavaScript, CSS a dotazovací jazyk SQL. Tato kombinace byla zvolena na základě předchozí zkušenosti s těmito jazyky a jejich schopnosti poskytnout robustní a efektivní prostředky pro tvorbu webových aplikací. Výsledná aplikace byla úspěšně dokončena a nasazena do praxe. Nicméně, jak je obvyklé u vývoje softwaru, i tato aplikace bude podléhat dalším úpravám a vylepšením v následujících verzích. Kromě samotné implementace aplikace je dalším cílem této práce detailně zdokumentovat celý proces vytváření aplikace a popsat všechny použité metody a postupy. Tato závěrečná práce poskytne ucelený přehled o vývoji aplikace a bude sloužit jako cenný zdroj informací pro budoucí úpravy aplikace.

Klíčová slova: webové aplikace, databáze, programování, PHP, optimalizace

Abstract

Title: Development of web pages for equipment inventory

The main objective of this bachelor thesis was to develop a functional web application serving both as a platform for presenting company information and for inventory management of client-managed devices. For the development of the application, programming languages HTML, PHP, JavaScript, CSS, and the querying language SQL were utilized. This selection was made based on prior experience with these languages and their ability to provide robust and efficient tools for web application development. The resulting application was successfully completed and deployed into practice. However, as is customary in software development, this application will undergo further modifications and enhancements in subsequent versions. In addition to the implementation of the application itself, another aim of this thesis is to comprehensively document the entire process of application development and describe all the methods and procedures employed. This documentation will offer a comprehensive overview of the application's

development and serve as a valuable source of information for future application modifications.

Key words: web applications, databases, programming, PHP, optimization

Obsah

1	Úvod.....	1
2	Cíl a metodika práce.....	2
2.1	Cíl práce.....	2
2.2	Metodika.....	2
3	Teoretická část.....	4
3.1	Popis problematiky a její řešení.....	4
3.1	Vývojové prostředky a jazyky pro tvorbu webových aplikací.....	4
3.1.1	Vývojové prostředky.....	4
3.1.2	Programovací jazyky.....	9
3.2	Databázové systémy, jejich analýza a návrh.....	16
3.3	Existující aplikace pro správu zařízení.....	19
3.3.1	Microsoft 365 admin center.....	19
3.3.2	TeamViewer Management Console.....	21
3.3.3	Avast Business Hub.....	22
3.3.4	AVG Business.....	23
3.4	Optimalizace webových aplikací.....	24
4	Praktická část.....	26
4.1	Použité programovací jazyky.....	26
4.2	Připojení k databázi.....	26
4.3	PHP.....	28
4.4	HTML.....	31
4.5	CSS.....	33
4.6	JavaScript.....	41
4.7	Databáze.....	43
4.7.1	tblFirma:.....	46

4.7.2	tblNastaveni:.....	46
4.7.3	tblStredisko:.....	47
4.7.4	tblTypVlast:.....	47
4.7.5	tblTypZar:	48
4.7.6	tblUmistení:	48
4.7.7	tblUzivatel:	49
4.7.8	tblVlastnosti:	49
4.7.9	tblZarizeni:	50
4.7.10	tblZurnal:.....	50
4.7.11	tblPopisZurnal:.....	51
5	Shrnutí a diskuse výsledků.....	52
6	Závěry a doporučení	53
7	Seznam použité literatury.....	54
8	Seznam obrázků.....	57
9	Přílohy	59
10	Zadání práce z IS (eVŠKP)	60

1 Úvod

Webové aplikace jsou v dnešní době velmi populární, a proto vzniklo i mnoho možností, jak webové stránky vytvářet pro osobní uživatelské potřeby, a to i s minimální znalostí programovacích jazyků. Jedná se o různé aplikace, kde vývoj probíhá čistě jen graficky. Ve většině případů lze ovšem na takto vytvořených webech najít mnoho nepotřebného kódu, který stránky dělá zbytečně náročnými, proto je projekt pro tuto bakalářskou práci zpracováván pouze s pomocí programů pro psaní kódu.

Jako téma pro tuto bakalářskou práci byl zvolen vývoj webové aplikace pro inventarizaci zařízení. Aplikace je psaná na míru IT firmě, ve které je inventarizace řešena velmi neefektivně.

Webová aplikace má následující záměry a části. Jedním z důležitých cílů projektu je publikace firmy a informování nových zákazníků. K tomuto účelu slouží první část stránky, na které si každý zákazník může přečíst veškeré informace o firmě. Nejprve je zobrazen stručný popis firmy jak ve formě textu, nebo stručně pomocí seznamu. Poté lze na této stránce nalézt i popis jednotlivých služeb, a nakonec i kontaktní údaje.

Druhá část služeb, dostupná pouze dlouhodobým klientům společnosti, se zaměřuje na správu a inventarizaci zařízení. Zde jsou pečlivě evidována všechna zařízení společnosti spolu se všemi relevantními parametry. Každé zařízení má svou historii změn vlastností, což umožňuje provádět zpětnou kontrolu. Tento záznam je automaticky generován, což zajišťuje efektivní a přesnou dokumentaci.

V následujícím textu jsou nejprve teoreticky popsány veškeré nástroje a metody, které byly použity při zpracovávání tohoto projektu. Po teoretické části následuje praktická sekce, kde je celý projekt detailně zdokumentován a popsán, včetně konkrétních užití metod. Tyto popisy jsou doplněny o obrázky z projektu, aby lépe ilustrovaly postupy a výsledky.

2 Cíl a metodika práce

2.1 Cíl práce

Cílem této práce je vytvořit funkční webovou aplikaci pro firmu, která bude sloužit jak pro publikaci informací o firmě, tak pro inventarizaci zařízení, dále také analyzovat a zdokumentovat celý vývojový proces a zkušenosti získané během zpracovávání tohoto projektu. V rámci této bakalářské práce jsou přiblíženy detaily implementace, použité technologie a přínosy, které aplikace přináší jak pro firmu, tak pro uživatele.

2.2 Metodika

V úvodní fázi tvorby této práce byly podrobně prozkoumány a analyzovány existující aplikace s podobným zaměřením. Cílem bylo získat vhled do současných řešení, pochopit jejich silné stránky, ale také identifikovat možná omezení či nedostatky. Tyto zkoumané aplikace jsou následně srovnávány v teoretické části práce, což poskytuje kontext a porovnání pro nově navrhovanou webovou aplikaci.

V dalším kroku byl vytvořen analytický model webové aplikace. Tento model byl navržen s cílem zajistit jasný a strukturovaný přehled o klíčových funkcích, komponentách a interakcích aplikace. Tato struktura a pochopení přispěla k lepšímu návrhu a implementaci aplikace v pozdějších fázích.

Pro implementaci webové aplikace byla zvolena kombinace programovacích jazyků HTML, CSS, JavaScript, PHP a pro práci s databází jazyk SQL. Tato volba vycházela z předchozích zkušeností s těmito technologiemi a z jejich schopnosti efektivně spolupracovat při vytváření dynamických webových stránek. Hlavní část aplikace je napsána v jazyce PHP. Ten umožňuje interakci s databází, která je klíčovou součástí celého projektu. Pro ukládání a správu dat byla zvolena relační databáze MySQL, vyvinutá firmou Microsoft. Pro pohodlnou manipulaci s touto databází bylo využito uživatelsky přívětivého prostředí phpMyAdmin. Tato kombinace technologií byla vybrána s ohledem na potřeby projektu a s cílem zajistit robustní a efektivní fungování webové aplikace.

Veřejná stránka představuje firmu skrze detailní a srozumitelný popis, který zdůrazňuje klíčové výhody a nabízené služby. Pro lepší orientaci uživatelů jsou využívány vizuální prvky a grafika. Tato část je systematicky rozdělena do několika sekcí, včetně kontaktních informací, kde je k dispozici formulář pro rychlou komunikaci s firmou a interaktivní mapa pro snadnější lokalizaci. Dále se zde nachází sekce naše služby, kde jsou zákazníci informováni o dostupných službách, včetně možnosti stáhnutí vzdálené podpory pro rychlé řešení problémů.

Inventarizační část je přístupná pouze po ověření uživatelského jména a hesla, což zajišťuje ochranu citlivých informací. Správce firmy má možnost spravovat všechny uživatelské účty. Klienti v této sekci mohou přidávat, upravovat a vyhledávat zařízení ve své přiřazené firmě.

3 Teoretická část

3.1 Popis problematiky a její řešení

Bakalářská práce se zaměřuje na problematiku inventarizace zařízení. Při tvorbě této bakalářské práce byla navázána spolupráce s firmou, která měla problém s inventarizací zařízení. Předmětná firma se zabývá IT službami, respektive správou počítačových sítí, poradenstvím, servisem a podporou. Při správě všech zařízení celé klientské firmy je potřeba udržovat aktuální stav všech zařízení ve firmách. V každé firmě je pověřená osoba, která přerozděluje pracovníkům nová zařízení nebo vyřazuje stávající. Před vývojem tohoto projektu si správce i pověřená osoba ve spravované firmě vedli vlastní inventarizaci. Ani jedna strana nedokázala napsat vše, protože ne vždy všechno řešila právě ona, tak si po určitém čase soubory vyměnili a doplnili rozdíly. Toto řešení není ideální. Přináší spoustu možností chyb a také je zcela neefektivní a časově velmi náročné.

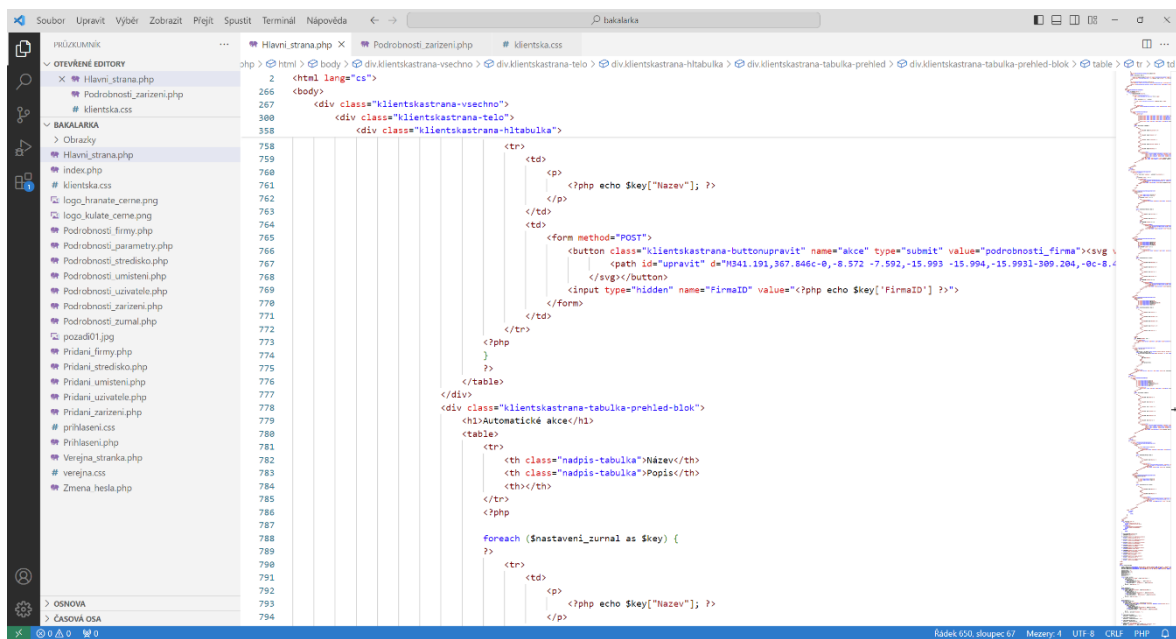
Řešením této problematiky bylo vytvoření webové aplikace s databázovým backendem, která slouží nejen pro inventarizaci, ale i pro prezentaci firmy. Vytvořený systém umožňuje oběma stranám jak správcům, tak pověřeným osobám ve spravovaných firmách editovat a aktualizovat data v reálném čase. Toto řešení eliminuje možnost chyb spojených s ručním procesem a zvyšuje efektivitu celého inventarizačního procesu. Implementace aplikace zahrnuje i bezpečný přístup ke správě zařízení s možností přihlášení, a to s využitím programovacího jazyka PHP a ukládáním dat v MySQL databázi. Vytvořená webová aplikace přináší nejen řešení konkrétních výzev firmy ohledně inventarizace, ale také zlepšuje celkovou správu a prezentaci firemních informací.

3.1 Vývojové prostředky a jazyky pro tvorbu webových aplikací

3.1.1 Vývojové prostředky

Mezi vývojové prostředky, které byly využity při tvorbě webové inventarizační aplikace, patří program s názvem Visual Studio Code viz obr. 1. Tento bezplatný kódovací editor, vyvinutý společností Microsoft, byl navržen s důrazem na

jednoduchost, rychlost a rozšiřitelnost, aby vyhovoval širokému spektru vývojářů. Program vyniká svou flexibilitou a rozsáhlou nabídkou rozšíření, která nezahrnují pouze další programovací jazyky, ale také motivy, ladící nástroje a další užitečné funkce. Microsoft při vývoji nezapomněl ani na podporu Gitových repozitářů a s nimi související práci. Pro psaní a ladění spustitelných aplikací je k dispozici také nástroj pro debugging (Visual Studio Code 2024a).



Obr. 1 Visual Studio Code; Zdroj: Vlastní zpracování

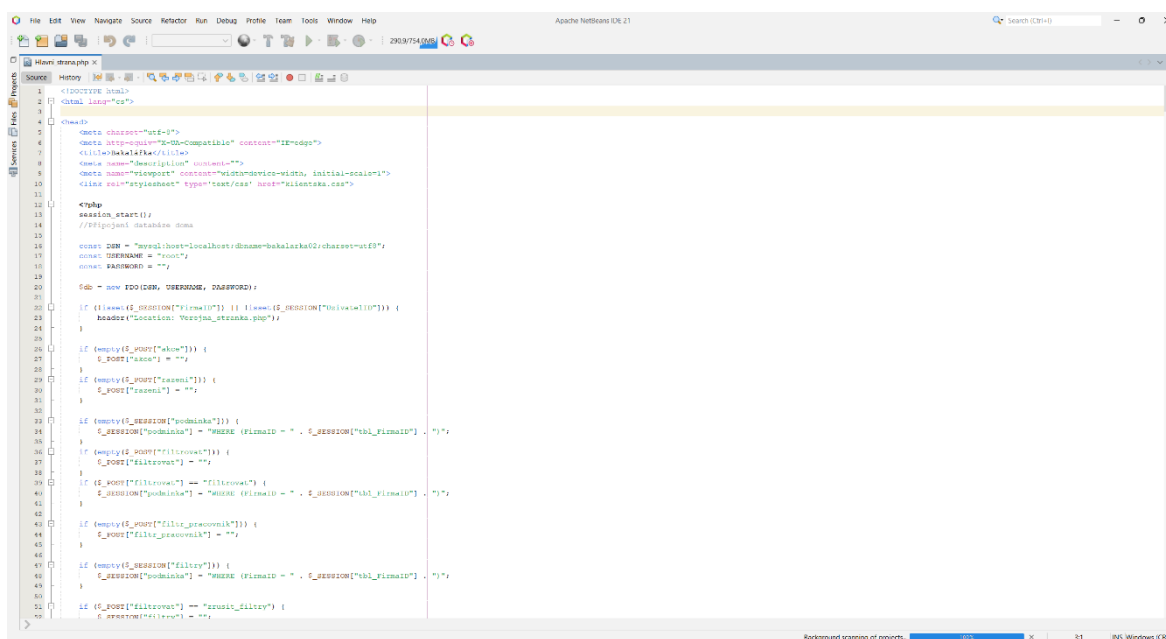
Jednou z pozitivních vlastností editoru je jeho rozsáhlá uživatelská komunita. Pokud uživatel narazí na problém, pravděpodobně není první. Samotný program je navržen s důrazem na jednoduchost a přehlednost, což znamená, že není náročný na systémové prostředky. Lze ho spouštět na nejrozšířenějších operačních systémech, včetně Windows, Linux a MacOS (Visual Studio Code 2024b).

Pro vývoj inventarizační aplikace byly použity jazyky HTML, PHP, JavaScript a CSS viz. podkapitola 3.1.2. Pro pohodlnější práci s těmito jazyky byla využita následující rozšíření, která jsou zmíněna níže v textu. Rozšíření pomáhají vývojářům s psaním kódu pomocí nápověd. Také umí formátovat kód tak, aby byl co nejlépe čitelný a přehledný. HTML CSS Support je rozšíření, které se zaměřuje na formátování

jazyků HTML a CSS. Při tvorbě webových stránek je zcela nezbytným pro čitelnost a přehlednost kódu (Visual Studio Marketplace 2024a).

PHP Intelephense je vysoce výkonným PHP jazykovým serverem plným nezbytných funkcí pro produktivní vývoj. Nabízí rychlé doplňování kódu, předdefinované symboly a klíčová slova s automatickým přidáním deklarácí. Poskytuje také podrobnou nápovědu k parametrům, metodám a funkcím (Visual Studio Marketplace 2024b).

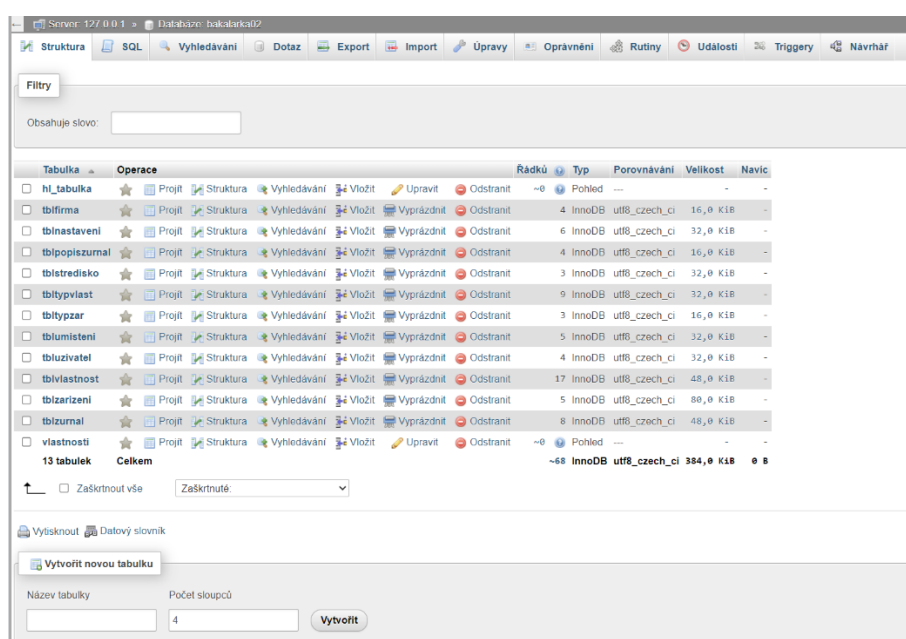
V rámci alternativního výběru vývojového nástroje pro psaní kódu se nabízí populární a uživatelsky přívětivý program NetBeans, vyvíjený společností Apache viz obr. 2. NetBeans je open-source prostředí určené pro programování v různých jazycích, včetně Java, JavaScript, PHP, HTML, CSS a dalších. Jeho univerzálnost umožňuje uživatelům pracovat na širokém spektru projektů napříč operačními systémy, a to včetně Windows, Linux a MacOS (Apathe Netbeans 2017).



Obr. 2 NetBeans; Zdroj: Vlastní zpracování

Během vytváření aplikace byl nezbytný další nástroj, konkrétně webový nástroj pro správu MySQL databáze nebo MariaDB, známý jako PHPMysqlAdmin viz obr. 3

(www.phpmyadmin.net). Jedná se o open-source nástroj navržený pro správu databáze prostřednictvím webového prohlížeče. Tento nástroj je implementován v programovacím jazyce PHP a umožňuje uživatelům snadný přístup a manipulaci s MySQL databázemi bez nutnosti používání příkazového řádku. Správa databáze probíhá prostřednictvím grafického uživatelského rozhraní, které umožňuje provádět operace, jako jsou vytváření tabulek, definice atributů a následně i navazování vztahů mezi tabulkami. Tato funkcionality je dostupná i prostřednictvím jazyka SQL (contributors 2003).



Obr. 3 PHPMyAdmin; Zdroj: Vlastní zpracování

Další alternativou pro správu databáze MySQL je aplikace přímo od společnosti Oracle, a to MySQL Workbench viz obr. 4. Tato aplikace poskytuje podobné funkce jako již zmiňovaný webový nástroj PHPMyAdmin. MySQL Workbench nabízí možnost vizuálního návrhu databází, spouštění a optimalizaci SQL dotazů, správu uživatelů a export/import dat. V porovnání s PHPMyAdmin je MySQL Workbench více vhodná pro pokročilé uživatele, neboť poskytuje složitější a podrobnější možnosti nastavení (MySQL 2024).

MySQL Model Report

Schema Overview

Schema sakila →
Columns, Indices and Triggers →

Schema Tables (16)

- actor
- address
- category
- city
- country
- customer
- film
- film_actor
- film_category
- film_text
- inventory
- language
- payment
- rental
- staff
- store

Schema Views (7)

- customer_list
- film_list
- nicer_but_slower_film_list
- staff_list
- sales_by_store
- sales_by_film_category
- actor_info

Routines (6)

- rewards_report
- get_customer_balance
- film_in_stock
- film_not_in_stock
- inventory_held_by_customer
- inventory_in_stock

Table customer (6/16)

Table Properties

Average Row Length	n/a	Use Check Sum	no
Connection String	n/a	Default Character Set	utf8
Default Collation	n/a	Delay Key Updates	no
Minimal Row Count	n/a	Maximum Row Count	n/a
Union Tables	n/a	Merge Method	n/a
Pack Keys	n/a	Has Password	no
Data Directory	n/a	Index Directory	n/a
Engine	InnoDB	Row Format	n/a

Columns

Key	Column Name	Datatype	Not Null	Default	Comment
PK	customer_id	SMALLINT	Yes		
	store_id	TINYINT	Yes		
	first_name	VARCHAR(45)	Yes		
	last_name	VARCHAR(45)	Yes		
	email	VARCHAR(50)	No	NULL	
	address_id	SMALLINT	Yes		
	active	BOOLEAN	Yes	TRUE	
	create_date	DATETIME	Yes		
	last_update	TIMESTAMP	No	CURRENT_TIMESTAMP	

Indices

Index Name	Columns	Primary	Unique	Type	Kind	Comment
PRIMARY		Yes	No	PRIMARY		
idx_fk_store_id		No	No	INDEX		
idx_fk_address_id		No	No	INDEX		
idx_last_name		No	No	INDEX		

Relationships

Relationship Name	Relationship Type	Parent Table	Child Table	Card.
fk_customer_address	Non-Identifying	address	customer	1:n
fk_customer_store	Non-Identifying	store	customer	1:n
fk_payment_customer	Non-Identifying	customer	payment	1:n

Obr. 4 MySQL Workbench; Zdroj: www.mysql.com

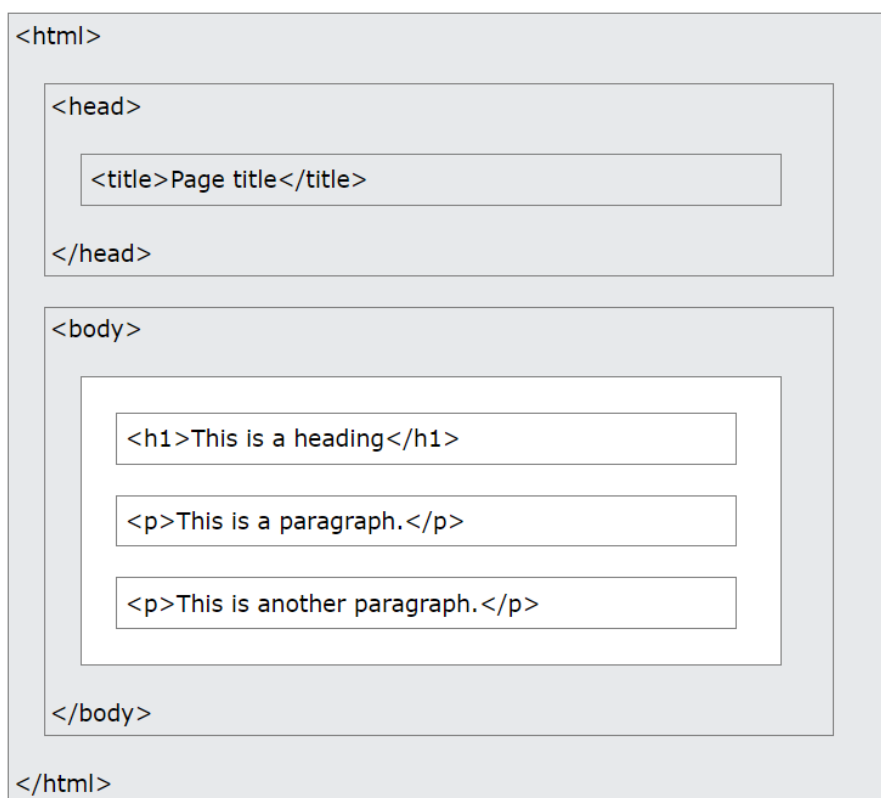
Pro běh a testování aplikací je nezbytný webový server, a to buď pronajatý nebo lokálně spuštěný na libovolném zařízení. Pro tento účel existují aplikační balíčky, jako jsou WAMP a XAMPP. Ty zajišťují jednoduchou instalaci a konfiguraci potřebných serverových technologií. Oba balíčky obsahují distribuci Apache serveru spolu s PHP, MariaDB a MySQL databázovými systémy. Oba programy jsou zdarma a open-source. WAMP a XAMPP jsou si velmi podobné, a proto je výběr mezi nimi často záležitostí osobní preference uživatele. WAMP je optimalizován především pro uživatele operačního systému Windows, zatímco XAMPP je multiplatformní a podporuje Windows, Linux a MacOS. Oba tyto balíčky umožňují rychlé nasazení webového serveru s minimálním úsilím, což je ideální pro vývoj a testování webových aplikací. Po instalaci těchto balíčků mohou uživatelé jednoduše spustit svůj lokální server a začít pracovat na svých projektech bez nutnosti složité konfigurace (XAMP 2023; Bourdon 2024).

3.1.2 Programovací jazyky

3.1.2.1 HTML

HTML představuje standardní značkovací jazyk, jehož hlavním účelem je vytváření a prezentace webových stránek. Zkratka "HTML" označuje HyperText Markup Language. Tento jazyk slouží k definování struktury dokumentu a propojení s různými zdroji, včetně odkazů, obrázků a multimédií (W3schools 1999a).

HTML využívá značky, nazývané též tagy viz obr. 5, pro identifikaci prvků na stránce. Každá značka má specifický význam a přispívá k celkové struktuře dokumentu. Struktura HTML dokumentu se skládá z deklarace (<!DOCTYPE html>), hlavičky (<head>) a těla (<body>). Hlavička obsahuje meta-informace a odkazy na externí soubory, tělo obsahuje samotný obsah stránky (W3schools 1999a).



Obr. 5 Struktura HTML; Zdroj: www.w3schools.com

V HTML je možné vytvářet rozmanité prvky, jako jsou nadpisy, odstavce, odkazy, obrázky a formulářové položky. Každý z těchto elementů může obsahovat atributy, které poskytují informace o daném prvku, například informace o viditelnosti, identifikátor (ID), název, třídu, typ a další vlastnosti. Většina prvků v HTML jsou párové, což znamená, že musí mít odpovídající začáteční a koncový tag. Koncový tag je vždy označen lomítkem, které se nachází před názvem tagu. Tato struktura umožňuje jasnou definici každého prvku a zajišťuje korektní zobrazení obsahu webové stránky (W3schools 1999b).

Pro vkládání obrázků a multimediálního obsahu poskytuje HTML příslušné tagy. Formulářové prvky, obsažené v tagu `<form>`, slouží k interaktivní komunikaci s uživatelem a mohou obsahovat textová pole pro vyplnění, tlačítka a zaškrťovací pole (W3schools 1999c).

Po správně vytvořeném HTML souboru je následným cílem webového prohlížeče korektně zobrazit obsah. Webový prohlížeč nezobrazuje jednotlivé HTML tagy, ale využívá je k definici způsobu prezentace obsahu. Například, pokud je text uzavřen mezi tagy `<h1>` (označující nadpis první úrovně), prohlížeč zobrazí tuto část jako nadpis (W3schools 1999d).

Tento jazyk byl klíčovým faktorem pro vývoj World Wide Webu. Jeho historie sahá až do počátků internetu v 90. letech 20. století. První specifikace HTML byla jednoduchá a umožňovala formátování textu pomocí značek, `<h1>` pro nadpisy a `<p>` pro odstavce. S rozvojem internetu a rostoucím počtem webových stránek se HTML vyvíjel a byl aktualizován několika verzemi. V průběhu let bylo přidáno mnoho nových prvků a atributů, což umožnilo vytvářet interaktivní a komplexní webové stránky. HTML se stalo standardem pro vývoj webových aplikací a stále je klíčovým nástrojem pro tvorbu obsahu na internetu (W3schools 1999a).

Celkově lze říci, že HTML je klíčovým nástrojem pro vývoj webových stránek. V kombinaci s dalšími technologiemi, jako jsou například CSS a JavaScript, umožňuje vytvářet poutavé a interaktivní webové stránky.

3.1.2.2 CSS

Cascading Style Sheets neboli CSS představuje stylizační jazyk používaný ve vývoji webových stránek k definici vizuálního vzhledu dokumentů napsaných v jazyce HTML. Hlavním účelem CSS je oddělení prezentace obsahu od jeho struktury a usnadnění kontroly nad vizuálním designem webových stránek. Rozdělení také zajišťuje celkovou přehlednost (W3schools 1999e).

V CSS lze definovat různé vizuální vlastnosti elementů, jako je barva, velikost písma, odsazení, zarovnání a mnoho dalších. Tato pravidla mohou být aplikována na různé typy prvků na stránce, což umožňuje dosažení konzistentního a estetického vzhledu. Rozdělení elementů může být provedeno pomocí typu elementu, jejich identifikátoru (ID) nebo třídy (W3schools 1999f).

Používání selektorů umožňuje cílení na specifické části HTML stránky a aplikování stylů pouze na vybrané prvky. Kromě toho umožňuje CSS také tvorbu mediálních dotazů, což usnadňuje vytváření responzivních designů přizpůsobujících se různým velikostem obrazovek (W3schools 1999f).

Historie kaskádových stylů (CSS) začala v roce 1994 v CERNu, kde Håkon Wium Lie navrhl jazyk pro stylování webových stránek. První návrh byl publikován v roce 1996 a podpořil ho Dave Raggett. Internet Explorer 3 se stal prvním komerčním prohlížečem s podporou CSS v roce 1996, s následnou podporou od Netscape a Operry. Druhá verze, CSS2, byla vydána kolem roku 2000 s dalšími vlastnostmi pro formátování písma a praktické věci. V dnešní době se používá poslední verze CSS3 (Lie a Bos 1999).

Díky flexibilitě a síle CSS mohou vývojáři vytvářet atraktivní a responzivní webové stránky, přičemž oddělení obsahu od jeho prezentace zvyšuje údržbu a znovupoužitelnost kódu.

Psaní veškerého kódu ručně je velmi náročné a zdlouhavé, a proto existuje několik programů které toto usnadňují. Jeden z těchto programů byl využit i pro tvorbu této práce. Program nese jméno TeleportHQ. Jedná se o kolaborativní front-endovou platformu, která kombinuje nástroje pro vývoj uživatelského rozhraní a modelování obsahu. Jeho klíčové prvky zahrnují vizuální editor, který umožňuje uživatelům rychle vytvářet a publikovat statické webové stránky bez nutnosti psaní rozsáhlého kódu. Platforma také umožňuje integraci s designovým nástrojem Figma (www.figma.com) a poskytuje možnost přizpůsobení pomocí vlastního kódu a vizuálních úprav. Další významnou funkcí je podpora spolupráce mezi týmy a integrace s GitHubem (github.com) pro sledování změn v kódu (TeleportHQ 2023).

3.1.2.3 PHP

PHP (Hypertext Preprocessor) je serverový skriptovací jazyk, který se stal jedním z nejpobulárnějších nástrojů v oblasti webového vývoje.

Jednou z klíčových vlastností PHP je jeho schopnost integrace s HTML. Kód napsaný v PHP může být přímo vložen do HTML souborů, což umožňuje kombinovat statické a dynamické prvky. Tato integrace usnadňuje vytváření interaktivních webových stránek s možností generování obsahu na základě uživatelských vstupů nebo dat z databází (PHP 2001a).

Významným aspektem PHP je podpora pro práci s databází. PHP umožňuje připojení k různým databázovým systémům, což poskytuje možnost ukládání a získávání dat. Tato schopnost je klíčová pro vývoj dynamických webových aplikací, které vyžadují efektivní správu informací (PHP 2001b).

V rámci jazyka PHP je implementována podpora pro objektově orientované programování (OOP). To přináší výhody, jako je modularita a lepší organizace projektu. PHP podporuje vytváření tříd a objektů, což přispívá k efektivnějšímu programování (PHP 2001b).

Vedle toho PHP poskytuje nástroje pro manipulaci s formuláři, zpracování uživatelských vstupů a správu session. Tyto funkce jsou klíčové pro interaktivní a personalizované uživatelské rozhraní webových stránek. PHP také poskytuje mnoho tříd a metod. Některé z nich jsou blíže představeny (PHP 2001b).

Jak už bylo zmíněno jazyk PHP podporuje práci s databází. K tomu slouží následující třída zvaná PDO, která poskytuje jednotné rozhraní pro práci. Používá se pro vytváření připojení k databázi a provádění dotazů. Jedná se o vrstvu abstrakce nad konkrétními databázovými ovladači. To znamená, že je možno pracovat s různými typy databází pomocí jednotného rozhraní, což usnadňuje práci s více databázovými systémy. PDO ovšem neposkytuje databázovou abstrakci, a tudíž ani nepřepisuje SQL funkce ani neemuluje chybějící funkce (PHP 2001c).

Jednou z klíčových vlastností PDO je používání připravených dotazů (prepared statements), což je technika, která odděluje SQL kód od datových hodnot. To zabraňuje SQL injection útokům a zvyšuje bezpečnost aplikace (PHP 2001d).

Připojení k databázi se vytváří pomocí instance základní třídy PDO, která přijímá parametry pro určení zdroje databáze a volitelně pro uživatelské jméno a heslo. Chyby připojení jsou zachycovány a zpracovávány prostřednictvím výjimky PDOException. Po úspěšném připojení se instance PDO vrátí a připojení zůstane aktivní po dobu životnosti tohoto objektu. Trvalá připojení, uložená v mezipaměti, jsou často využívána ke snížení režie spojené s navazováním nového připojení. Při použití trvalých připojení je nutné zajistit, aby všechny odkazy na instanci PDO byly odstraněny pro zachování integrity připojení (PHP 2001e).

Třída PDO nabízí i spoustu metod pro práci s databází. Metoda PDO::exec() umožňuje provádět příkazy SQL v jediném volání a vrací počet změněných nebo odstraněných řádků. Pokud žádné řádky nebyly ovlivněny, vrátí hodnotu 0. Tato metoda nevrací výsledky ze SELECT příkazů. Pro SELECT příkazy je vhodnější použití metody PDO::query() (PHP 2001f).

Metoda `PDO::query()` umožňuje připravit a provést SQL příkaz v jednom volání a vrací objekt `PDOStatement`. Pokud dotaz obsahuje zástupné symboly, je lepší použít metody `PDO::prepare()` a `PDOStatement::execute()` (PHP 2001g).

Metoda `PDO::prepare()` připravuje příkaz SQL k provedení a vrací objekt `PDOStatement` při úspěšném připravení příkazu nebo `false` v případě neúspěchu. Tato metoda umožňuje definovat šablonu příkazu s možnými značkami parametrů, které budou nahrazeny hodnotami při provádění příkazu. Každý parametr musí být zahrnut do příkazu před voláním metody `PDOStatement::execute()`. Použití `PDO::prepare()` a `PDOStatement::execute()` optimalizuje výkon aplikace a chrání před útoky SQL injection (PHP 2001h).

Pro další práci s databází jsou využity i metody ze třídy `PDOStatement`. Metoda `PDOStatement::execute()` provádí připravený příkaz. Pokud příkaz obsahuje značky parametrů, ty musí být předány jako pole hodnot parametrů. Pokud je doplněno více hodnot, než je uvedeno ve zpracovávaném SQL příkazu, způsobí selhání operace s chybou. Metoda vrací hodnotu `true` při úspěšném provedení a `false` při selhání (PHP 2001ch).

Metoda `PDOStatement::fetchAll()` načte všechny řádky z výsledku dotazu a vrátí je jako pole. Může být nastavena různými režimy načítání, jako je `FETCH_COLUMN` pro získání hodnot jednoho sloupce, `FETCH_CLASS` pro vytvoření instancí určité třídy nebo `FETCH_FUNC` pro použití uživatelské funkce k manipulaci s daty. Tato metoda vrací prázdné pole, pokud nejsou k dispozici žádné výsledky (PHP 2001i).

Metoda `PDOStatement::fetchColumn()` umožňuje vrátit hodnotu jednoho sloupce z výsledků dotazu. Je-li řádek k dispozici, vrátí hodnotu sloupce, v opačném případě vrátí `false`. Parametr `column` určuje index sloupce, který má být načten, přičemž výchozí hodnota je 0 pro první sloupec (PHP 2001j).

3.1.2.4 SQL

SQL (Structured Query Language) neboli (strukturovaný dotazovací jazyk) je standardizovaný programovací jazyk, který se používá k manipulaci s databázovými daty. Jeho hlavním účelem je umožnit uživatelům provádět různé operace s daty, jako je vytváření, čtení, aktualizace a mazání dat v databázích. SQL umožňuje uživatelům definovat, jakým způsobem budou data ukládána, jak budou organizována a získávána z databázového systému (W3schools 1999g).

SQL poskytuje možnosti pro vytváření a modifikaci struktury databáze. Uživatelé mohou pomocí příkazů `CREATE TABLE` a `ALTER TABLE` definovat strukturu tabulek. Tyto příkazy umožňují definovat datové typy, primární a cizí klíče a další vlastnosti tabulek. Pro zrušení tabulky existuje příkaz `DROP TABLE`, který z praktického hlediska není vhodný a dokáže napáchat velké škody ve struktuře databáze (W3schools 1999h).

Tento jazyk se skládá z různých typů příkazů, které jsou používány pro manipulaci s daty. Mezi nejčastější příkazy SQL patří `SELECT`, `INSERT`, `UPDATE` a `DELETE`. Příkaz `SELECT` se používá k načítání dat z databáze, `INSERT` umožňuje vkládat nová data do tabulek, příkaz `UPDATE` je používán k aktualizaci existujících dat v tabulkách a `DELETE` je používán k odstraňování dat z tabulek (W3schools 1999h).

Důležitou součástí SQL jsou dotazovací operátory, které umožňují filtrování, řazení a seskupování dat v rámci dotazů. Používají se dohromady s příkazem `SELECT`. Operátor `DISTINCT` zaručuje vrácení pouze různých hodnot. Hodnoty, které jsou stejné, se zobrazí jako jedna (W3schools 1999ch).

Pro seřazení výběru existuje operátor `ORDER BY`, který výsledná data seřadí sestupně přidáním klíčového slova `DESC` nebo vzestupně pomocí slova `ASC`. Filtrování je prováděno pomocí podmínek v klauzuli `WHERE` (W3schools 1999i).

SQL je základním nástrojem pro práci s relačními databázemi a je nezbytný pro vývoj a správu datových aplikací.

3.1.2.5 JavaScript

JavaScript je interpretovaný programovací jazyk, který se používá především pro vývoj webových aplikací. Jedná se o skriptovací jazyk s možností vykonávání kódu na straně klienta, což přináší interaktivitu do webových stránek (W3schools 1999j).

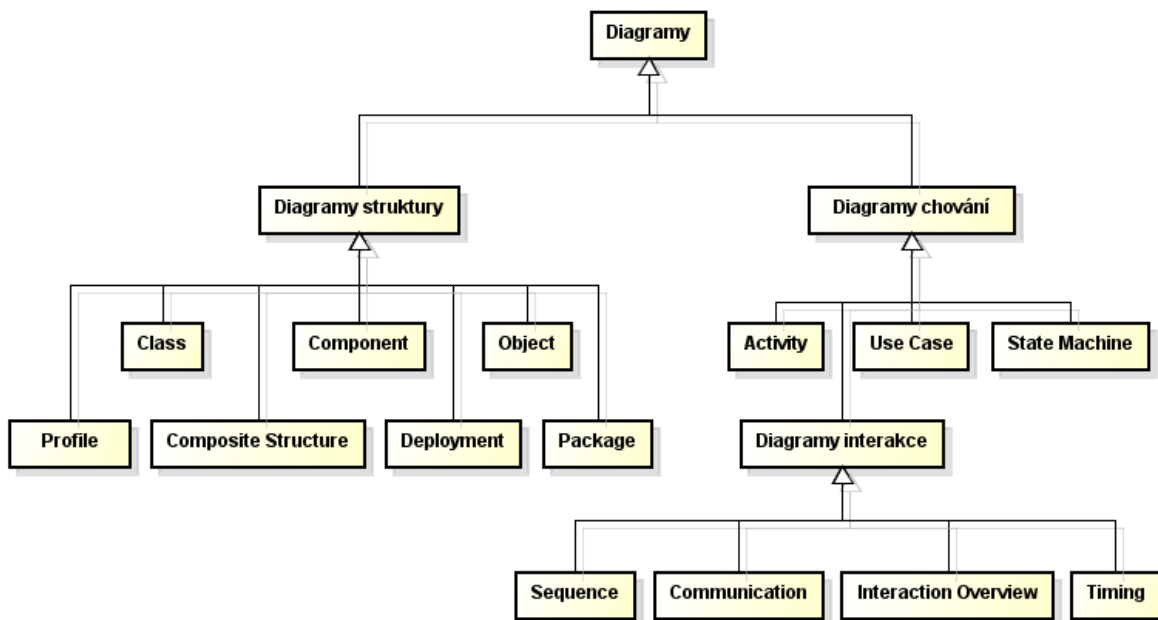
JavaScript nabízí několik významných vlastností. První z nich je schopnost asynchronního programování, což znamená, že může provádět operace neblokujícím způsobem, a tím je zvýšen výkon webových aplikací. Další klíčovou vlastností je reagování na uživatelské interakce nebo události a následné provádění akce. JavaScript je objektově orientovaný jazyk, který umožňuje vytvářet objekty a manipulovat s nimi. Podporuje také dynamické typování, což znamená, že datové typy proměnných nemusí být explicitně určeny při deklaraci (W3schools 1999k).

Další oblastí využití JavaScriptu je práce s DOM (Document Object Model), což představuje strukturu HTML nebo XML dokumentu. To umožňuje vývojářům dynamicky měnit obsah a styl webových stránek. Vyhledávání HTML elementu, u kterého chceme provést změnu, probíhá pomocí identifikátoru (ID) (W3schools 1999l).

3.2 Databázové systémy, jejich analýza a návrh

Databázové systémy představují klíčový pilíř v moderním informačním prostředí, které poskytuje strukturované uložení, organizaci a správu dat. Používají se pro efektivní uchovávání rozmanitých typů informací, od jednoduchých textových položek až po složité relace mezi daty. Tyto systémy se využívají v širokém spektru aplikací, od webových a mobilních aplikací, přes podnikové informační systémy, až po vědecké projekty. Díky nim je možné rychle a spolehlivě manipulovat s daty, zajišťovat jejich bezpečnost a integritu a poskytovat uživatelům rychlý a efektivní přístup k informacím.

UML (Unified Modeling Language) je standardizovaný soubor grafických notací používaných při vývoji softwaru, který se stal klíčovým nástrojem pro analýzu a návrh informačních systémů. Vznikl v 90. letech sjednocením metodik a specifikací a dnes je široce využíván v dokumentacích a materiálech. Jeho hlavní význam spočívá v usnadnění komunikace v týmu a lepším porozumění klientovým požadavkům. UML lze využít jako náčrt pro jednoduchou vizualizaci, plán pro detailní návrh a dokumentaci. Skládá se ze 14 diagramů, které lze rozdělit do dvou základních skupin: diagramy struktury, popisující strukturu systému, a diagramy chování, popisující jeho chování, viz obr. 6 (Hartinger 2024a).



Obr. 6 Seznam diagramů UML; Zdroj: www.itnetwork.cz

Následující výklad je zaměřen na bližší charakterizaci dvou UML diagramů. Jsou přímo prakticky využity pro analýzu/návrh samotné webové aplikace. Existují dva základní diagramy ze souboru UML, které slouží pro lepší pochopení databáze a její struktury. První graf nazývaný Use Case Diagram, známý také jako diagram případů užití, slouží k popisu chování systému z pohledu uživatele. Jeho hlavním účelem je zachytit funkcionalitu systému a to, co od něj uživatelé očekávají, aniž by se zabýval konkrétní implementací. Jedná se často o první diagram, který se vytváří při návrhu

a pomáhá definovat požadavky a funkcionalitu systému. Diagram se skládá z případů užití, aktérů a vztahů mezi nimi.

Případ užití je sada akcí, které vedou k dosažení určitého cíle, a popisuje jednu funkcionalitu, kterou by měl systém umět. Aktéři jsou role, které komunikují s případy užití a mohou být uživateli nebo externími systémy. Diagram je vizualizován pomocí elips pro případy užití a postav pro aktéry, propojených šipkami označujícími vztahy (Hartinger 2024b).

V Use Case diagramu jsou vztahy mezi aktéry a případy užití kritickým prvkem, který definuje, jakým způsobem aktéři interagují se systémem. Vazby v diagramu jsou reprezentovány různými symboly a indikují specifické interakce mezi aktéry a případy užití. Základní vazbou je asociace, která popisuje, jakým způsobem může aktér využívat danou funkcionalitu systému. Dále je zde generalizace, která znázorňuje dědění vlastností a funkcí mezi aktéry. Speciální vazbou je <<include>>. Ta indikuje, že daná funkcionalita je důležitá a spustí se vždy, když je spuštěn případ užití, na který je napojena. Další důležitou vazbou je <<extend>>. Tento vztah indikuje, že daný případ užití může rozšiřovat jiný případ užití o dodatečnou funkcionalitu. Rozšířený případ užití se spustí pouze v případě splnění určité podmínky nebo události v rozšiřujícím případě. Tyto vazby jsou klíčové pro pochopení dynamiky a chování systému. Zároveň umožňují lepší návrh a analýzu systému z pohledu uživatelských interakcí (Hartinger 2024b).

Dalším důležitým diagramem, který ukazuje strukturu databáze, je diagram tříd. Diagram tříd, známý jako Class diagram, je nástrojem implementace. Slouží k detailnímu popisu struktury a funkcí systému. Na rozdíl od doménového modelu, který poskytuje spíše obecný přehled, je Class diagram konkrétní a obsahuje všechny třídy, atributy a metody, které budou součástí aplikace. Při jeho tvorbě se používají datové typy příslušného jazyka. Class diagram slouží jako návod pro programátory, kteří na základě něj převádějí návrh do kódu. Vytvoření Class diagramu je důležité i pro individuální projekty, protože pomáhá ujasnit si požadavky a přemýšlet o celkové struktuře systému před samotným

programování. Dokumentace vytvořená na základě tohoto diagramu poskytuje cenný zdroj informací pro budoucí úpravy a rozšíření aplikace. V Class diagramu jsou třídy, atributy a metody znázorněny graficky pomocí symbolů. U atributů se používají různé modifikátory přístupu: privátní, veřejné, chráněné nebo viditelné v rámci balíku (Hartinger 2024c).

Třídy jsou propojeny různými typy vztahů, které definují strukturu, chování a interakce mezi objekty v softwarovém systému. Asociace reprezentuje vztah mezi dvěma třídami, kde objekty jedné třídy mohou odkazovat na objekty druhé třídy. Speciální formy asociace zahrnují agregaci, kde jedna třída obsahuje objekty druhé třídy, a kompozici, kde je existence agregovaných objektů úzce svázána s životností kontejneru. Generalizace umožňuje dědění atributů a metod mezi třídami, ve které podtřída dědí vlastnosti od nadřazené třídy. Dále, vazba závislost ukazuje, jak změny v jedné třídě mohou ovlivnit druhou třídu, a vztah realizace indikuje, že jedna třída implementuje rozhraní nebo kontrakt definovaný jinou třídou nebo rozhraním. Tyto vztahy jsou zásadní pro efektivní modelování, návrh a analýzu objektově orientovaných systémů, umožňují lepší porozumění systémových struktur (Hartinger 2024d).

3.3 Existující aplikace pro správu zařízení

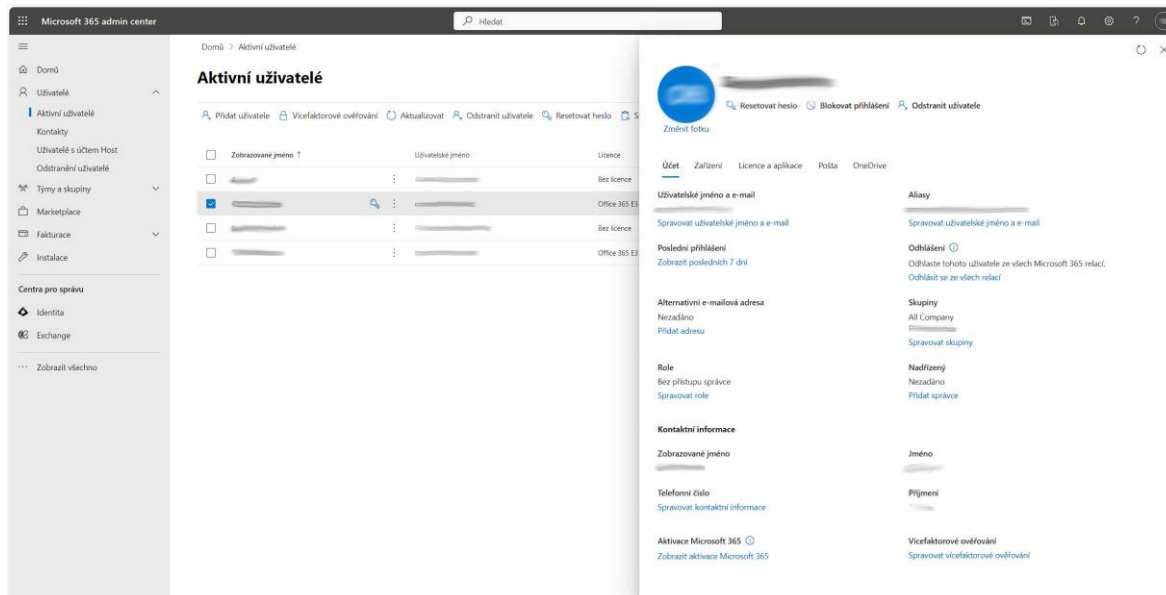
Při vývoji aplikace pro správu zařízení byl proveden průzkum existujících aplikací. Inspirace byla hledána u aplikací s podobným cílem. Na aplikace bylo nahlíženo z hlediska rozložení, vizuálního provedení a funkčnosti. Následující kapitoly se zaměřují na konkrétní aplikace a představují některé z podobností, které lze vysledovat mezi těmito existujícími projekty a touto prací.

3.3.1 Microsoft 365 admin center

První webová aplikace, kterou je tento projekt inspirován, je aplikace od společnosti Microsoft. Webové prostředí Microsoft 365 Admin Center viz obr. 6 umožňuje komplexní správu aplikací, služeb, dat, zařízení a uživatelů v rámci Microsoftu 365, primárně určeno pro IT firmy. Centrum pro správu nabízí přizpůsobitelné prostředí odpovídající unikátním potřebám uživatelů podle jejich rolí a organizací. Kromě

toho zahrnuje specializované pracovní prostory, které umožňují hlubší analýzu a správu. Aplikace zdůrazňuje důležitá data, která vyžadují nálehavé řešení, a poskytuje doporučení pro optimalizaci výsledků pro koncové uživatele i organizaci. Microsoft reaguje na aktuální potřeby, a proto vytvořil mobilní aplikaci Microsoft 365 Admin. Ta umožňuje uživatelům zobrazovat nastavení a provádět základní úkoly i na cestách. Aplikace poskytuje možnost přijímat oznámení, přidávat uživatele, resetovat hesla, spravovat zařízení a vytvářet žádosti o podporu – vše dostupné kdykoli a odkudkoli. Celkově je Microsoft 365 Admin Center komplexní platformou, která usnadňuje a zefektivňuje správu digitálních pracovních prostorů pro různé typy organizací (Microsoft 2024).

Z této aplikace bylo převzato uzpůsobení zobrazení možností, které se zobrazují jako menší okno v pravé části obrazovky. Dále pak i uspořádání hlavní strany, a to hlavičku ve které nalezneme vlevo logo, hned vedle názvu aplikace. V pravé části hlavičky pak můžeme vidět možnost odhlášení a účet.

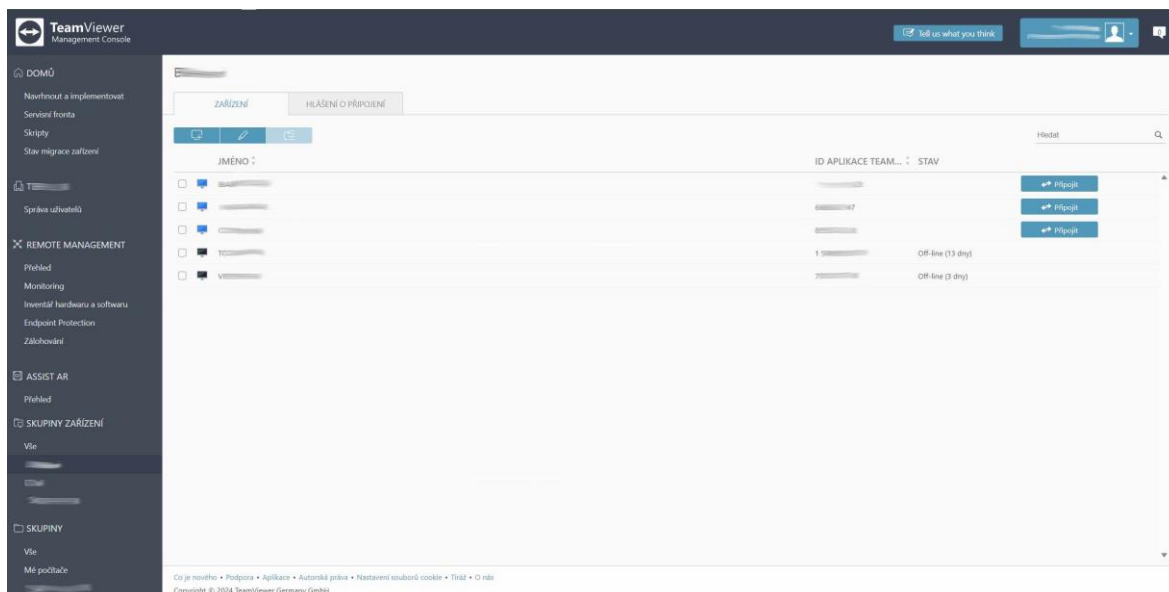


Obr. 7 Microsoft 365 admin center; Zdroj: Vlastní zpracování

3.3.2 TeamViewer Management Console

TeamViewer je komplexní aplikace navržená pro efektivní vzdálený přístup, sdílení pracovní plochy a online setkání. Jeho rozmanité funkce slouží k usnadnění vzdálené spolupráce a komunikace. Jednou z klíčových možností je vzdálený přístup, který umožňuje uživatelům ovládat vzdálené zařízení, což se ukazuje jako cenné při poskytování technické podpory nebo řešení problémů na dálku. Další důležitou funkcí je možnost sdílení pracovní plochy, což je užitečné pro prezentace, spolupráci na projektech a další formy interaktivní komunikace. TeamViewer též poskytuje prostředky pro bezpečný přenos souborů mezi připojenými zařízeními. Díky multiplatformní podpoře může TeamViewer být využíván na různých operačních systémech, což zvyšuje jeho univerzálnost. Bezpečnostní opatření, jako je šifrování přenosu dat a možnost nastavení hesel pro přístup, zajišťuje bezpečné prostředí pro vzdálenou spolupráci. TeamViewer je především využíván v profesionální sféře pro firmy, může ho ovšem využívat kdokoli. Program je placený a dělí se na několik variant podle možného počtu spravovaných zařízení a současně spuštěných relací vzdáleného připojení (TeamViewer 2024).

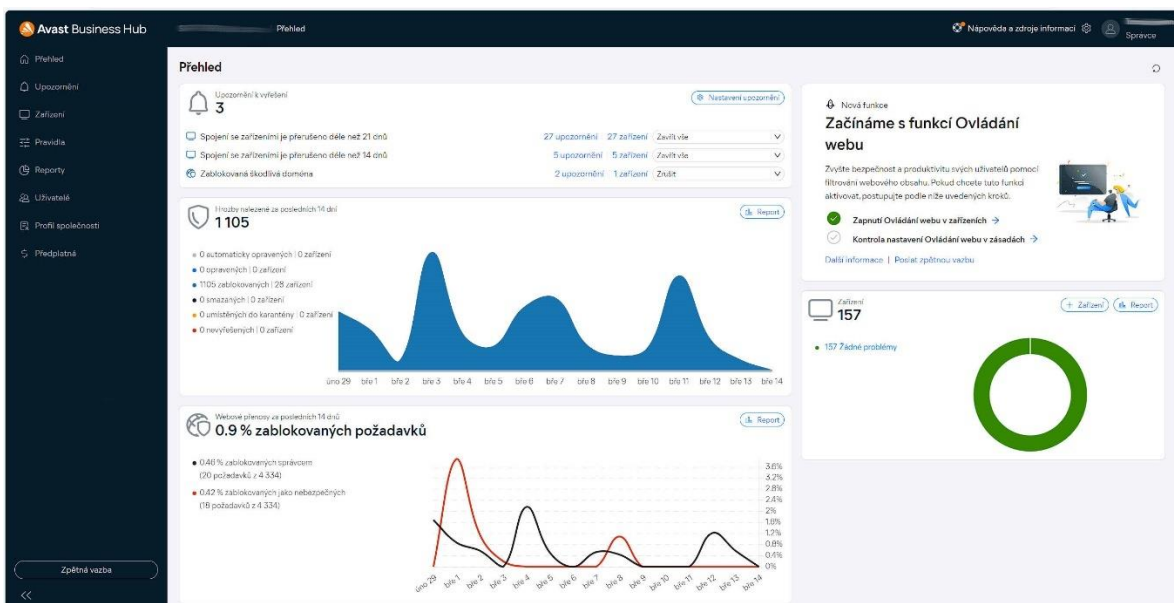
Tato aplikace je asi nejvíce podobná vyvíjené aplikaci. Můžeme zde najít podobnosti v designu. Další inspirací bylo celkové rozložení aplikace. V horní části hlavička, kde vlevo můžeme vidět název a logo a v pravé části přihlášeného uživatele s možnostmi. Poté v levém sloupečku nalezneme menu a hlavní část se nachází vpravo. To, kde se už aplikace liší, je po rozkliknutí podrobností. Dalším velkým rozdílem jsou data, která jsou shromažďována o každém zařízení a uživateli. Aplikace mají zcela odlišný hlavní cíl, a tak se pro firmu vyplatí používat obě.



Obr. 8 TeamViewer Management Console; Zdroj: Vlastní zpracování

3.3.3 Avast Business Hub

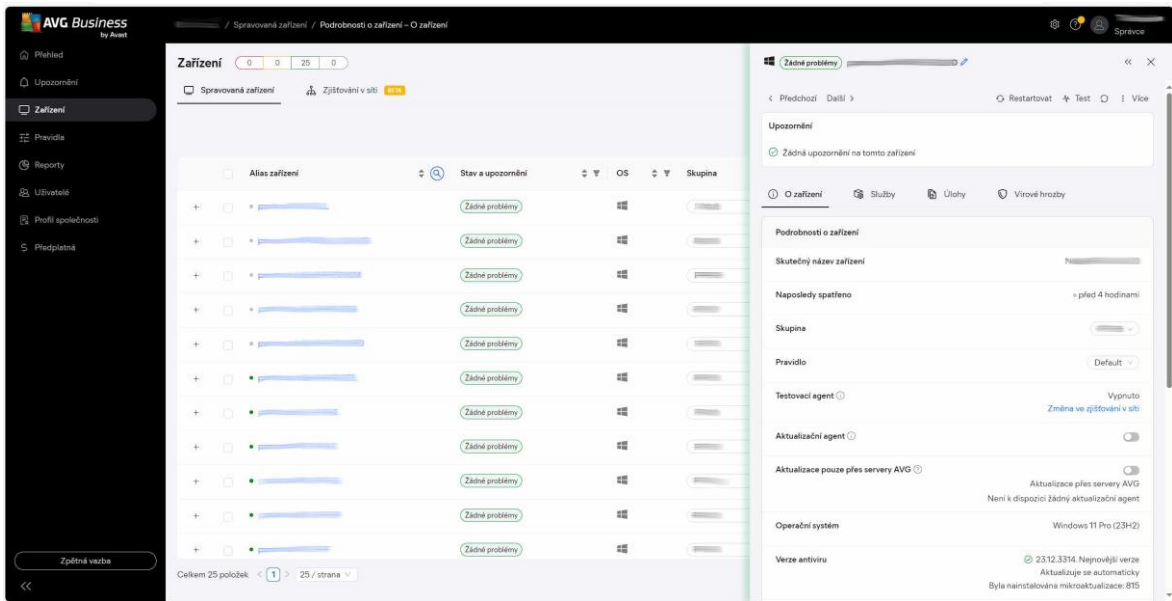
Avast Business Hub viz obr. 8 je komplexní aplikace vyvinutá společností Avast, která slouží k efektivní správě bezpečnostních aspektů ve firmě. Tato platforma umožňuje centralizovanou kontrolu a monitorování bezpečnostních opatření na všech připojených zařízeních v podniku. Jedná se o nástroj, který umožňuje správcům řídit antivirové funkce, distribuovat aktualizace, upravovat bezpečnostní politiku a také poskytuje sledování stavu bezpečnosti v reálném čase. Aplikace Avast Business Hub také podporuje generování zpráv a analýz bezpečnostních událostí. Uživatelé tak nabízejí komplexní nástroje pro zajištění kybernetické bezpečnosti v podnikovém prostředí (Avast 2024).



Obr. 9 Avast Bussiness Hub; Zdroj: Vlastní zpracování

3.3.4 AVG Business

AVG Cloud Management Console viz obr. 9 představuje sofistikovaný nástroj vyvinutý společností AVG Technologies, navržený pro centralizovanou správu bezpečnostních opatření v podnikových sítích. Tato aplikace nabízí uživatelsky přívětivé rozhraní pro efektivní správu bezpečnosti počítačů a dalších zařízení v rámci organizace. Klíčové funkce zahrnují správu antiviru a antimalwaru, centralizovanou aktualizaci, definici bezpečnostních politik, sledování bezpečnostního stavu, vzdálenou správu a generování zpráv i analýz. AVG Cloud Management Console se stává klíčovým prvkem pro efektivní řízení a zabezpečení informačních technologií ve firemním prostředí (AVG 2024).



Obr. 10 AVG Bussiness; Zdroj: Vlastní zpracování

3.4 Optimalizace webových aplikací

Optimalizace webových aplikací je nezbytný proces, který má za cíl zlepšit výkon a efektivitu aplikace, čímž se zajišťuje lepší uživatelská zkušenost a snižuje se náročnost na zdroje. Tento proces se opírá o několik klíčových principů, včetně minimalizace HTTP požadavků, optimalizace obrázků a multimediálního obsahu, komprese CSS a JavaScript kódů. Existují různé typy optimalizace, z nichž některé zahrnují optimalizaci na straně klienta nebo serveru. Optimalizace na straně klienta se zaměřuje na snižování velikosti souborů a minimalizaci HTTP požadavků, zatímco optimalizace na straně serveru zahrnuje zlepšení serverového hardwaru a softwaru. Optimalizace databáze se zaměřuje na optimalizaci struktury databáze a výkonných databázových dotazů. Tyto typy optimalizace spolupracují na zlepšení výkonu a efektivity webových aplikací, čímž se zajišťuje lepší uživatelská zkušenost a efektivnější využití zdrojů.

Optimalizace nemusí být pouze o výkonu, ale také o optimalizaci pro vyhledávače. Tedy, aby se webová stránka zobrazovala ve vyhledávačích na předních příčkách. Tato optimalizace se nazývá SEO neboli optimalizace pro vyhledávače. Jedná se o proces zlepšování viditelnosti webových stránek ve výsledcích vyhledávání. Cílem

je pomoci vyhledávačům lépe porozumět obsahu stránek a umožnit uživatelům snadněji je nalézt. Klíčovými prvky SEO jsou správné použití klíčových slov, kvalitní obsah a technická optimalizace webu. Zlepšení SEO může trvat nějaký čas a neexistuje žádná záruka okamžitého úspěchu. Je důležité, sledovat trendy a osvědčené postupy, pro zlepšení optimalizace stránky pro vyhledávače (Google 2024).

4 Praktická část

4.1 Použité programovací jazyky

Pro implementaci webové aplikace byly strategicky zvoleny programovací jazyky HTML, CSS, PHP a JavaScript. Tyto jazyky byly vybrány z důvodu jejich schopnosti poskytovat robustní a efektivní nástroje pro vývoj webových stránek s interaktivními a dynamickými prvky. Zároveň bylo rozhodnuto využít databázi pro ukládání, správu dat a autorizaci uživatelů. Pro ukládání dat byl zvolen MySQL, relační databázový systém. Tento systém umožňuje efektivní organizaci a správu dat, což je klíčový prvek pro správný chod webové aplikace. PhpMyAdmin, jako nástroj pro správu MySQL databáze, poskytl přehledné a uživatelsky přívětivé prostředí pro manipulaci s databází. Všechny tyto komponenty jsou jednotlivě představeny v další části práce.

Obecně lze konstatovat, že kombinace HTML, CSS, PHP, JavaScript, MySQL tvoří robustní a komplexní základ pro vytvoření funkční webové aplikace, která splňuje požadavky pro publikaci informací o firmě a zároveň umožňuje efektivní správu inventáře zařízení.

4.2 Připojení k databázi

Pro připojení databáze k webové aplikaci byla využita funkce PDO (PHP Data Objects) viz obr. 10. Toto rozšíření představuje odlehčené a konzistentní rozhraní pro efektivní přístup k databázím v PHP. Každý databázový ovladač, který implementuje rozhraní PDO, má možnost vystavovat specifické funkce pro konkrétní databáze jako běžné rozšiřující funkce. Tento typ připojení byl zvolen zejména kvůli jeho univerzálnosti, která umožňuje přístup k různým typům databází. Díky této volbě není nutné v případě přesunu systému na jinou databázi provádět rozsáhlé úpravy v kódu.

Dalším klíčovým důvodem pro volbu PDO byl jeho objektový přístup. Všechna data vrácená z databáze jsou strukturována jako objekt, to zajišťuje efektivnější manipulaci s daty. Tímto způsobem je dosaženo vyšší srozumitelnosti kódu, což

přispívá k celkové údržbě a rozšiřitelnosti aplikace. Objektový přístup usnadňuje práci s daty a zvyšuje přehlednost kódu. Toto řešení je zásadní pro efektivní vývoj a správu webové aplikace.

Celkově lze konstatovat, že volba PDO přináší do procesu práce s databází větší flexibilitu, bezpečnost a univerzálnost ve srovnání s tradičními rozhraními. Tato moderní technologie je obecně doporučována jako přednostní volba díky svým výhodám a schopnosti usnadnit a zefektivnit práci s databázovými operacemi v rámci webové aplikace.

Při připojení k databázi může dojít k chybě. Pro zajištění robustnosti aplikace je využita konstrukce try-catch, která umožňuje zachytit výjimky během vykonávání kódu viz obr. 10. Tímto způsobem je možné elegantně zobrazit chybovou stránku uživateli, aniž by aplikace skončila nekontrolovatelným selháním.

```
const DSN = "mysql:host=localhost;dbname=skolni_dbs;charset=utf8";
const USERNAME = "root";
const PASSWORD = "";

try {
    $db = new PDO(DSN, USERNAME, PASSWORD);
} catch (PDOException $e) {
    echo "<h1>Připojení k databázi se nezdařilo!</h1>";
    echo $e->getMessage();
    return;
}
```

Obr. 11 Připojení databáze; Zdroj: Vlastní zpracování

Pro práci s databází je využíváno PDOStatement:prepare k přípravě SQL kódu a PDOStatement:exec pro následné provedení kódu a vrácení počtu ovlivněných řádků. Pokud není počet ovlivněných řádků vyžadován, je použito PDOStatement:execute. Dále je používáno PDOStatement:query pro přípravu i provedení kódu a vrácení objektu PDOStatement. Tyto metody poskytují efektivní nástroje pro bezpečné a spolehlivé provádění SQL operací v rámci webové aplikace.

4.3 PHP

Skriptovací jazyk PHP je využíván pro tvorbu dynamických stránek, přičemž tento jazyk byl zvolen pro implementaci většiny back-endu na základě předchozích zkušeností.

Z důvodu komplexnosti a rozsahu webové aplikace by bylo nepraktické zahrnout veškerý kód do jednoho souboru. Namísto toho byla struktura aplikace rozdělena do několika samostatných souborů PHP, což umožňuje lepší organizaci a správu kódu. Každý z těchto souborů obsahuje ucelenou část funkcionality aplikace, což usnadňuje údržbu a rozšiřování aplikace v budoucnu. Hlavními soubory aplikace jsou veřejná a klientská stránka. Kromě toho byly vytvořeny další malé soubory pro části aplikace jako je přihlášení a změna údajů uživatele. Pro každou tuto část je vytvořen externí soubor se styly pro ucelený a konzistentní vzhled. Tyto soubory obsahují kompletní kód jednotlivých stránek ve formátu HTML.

Nicméně, i přes tuto rozdělenou strukturu by klientská stránka stále představovala příliš dlouhý a nepřehledný kód, což by ztížilo údržbu a další vývoj aplikace. Proto

byly jednotlivé sekce rozděleny do vedlejších souborů a následně vkládány do hlavního souboru pomocí PHP funkce include viz obr. 11.

```
if ($_SESSION["podokno"] == "podrobnosti_umisteni")
    include("Podrobnosti_umisteni.php");

if ($_SESSION["podokno"] == "podrobnosti_uzivatele")
    include("Podrobnosti_uzivatele.php");

if ($_SESSION["podokno"] == "podrobnosti_stredisko")
    include("Podrobnosti_stredisko.php");

if ($_SESSION["podokno"] == "podrobnosti_firma")
    include("Podrobnosti_firmy.php");

if ($_SESSION["podokno"] == "pridat_uzivatele")
    include("Pridani_uzivatele.php");

if ($_SESSION["podokno"] == "pridat_firmu")
    include("Pridani_firmy.php");

if ($_SESSION["podokno"] == "pridat_umisteni")
    include("Pridani_umisteni.php");

if ($_SESSION["podokno"] == "pridat_stredisko")
    include("Pridani_stredisko.php");
```

Obr. 12 Struktura PHP; Zdroj: Vlastní zpracování

Tento postup nejenže zlepšuje čitelnost a údržbu kódu, ale také umožňuje jednodušší správu a aktualizaci jednotlivých částí aplikace nezávisle na sobě.

Na klientské stránce lze identifikovat několik složitějších případů užití PHP, z nichž některé jsou dále přiblíženy. Prvním komplexním blokem programu jsou filtry. Filtraci dat tvoří mnoho podmínek, přičemž podmínka je postupně sestavována a následně použita jako kritérium v SQL dotazu směřujícím do databáze. V prvním kroku je nezbytně vybíráno, co má být filtrováno z databáze, aby ve filtru byla zobrazena pouze relevantní data pro filtrování. Ve druhém kroku jsou ověřována data označená ve filtrech a z nich je vytvářena podmínka. Důležité je, aby text podmínky byl srozumitelný a použitelný v SQL dotazu, což vyžaduje správné zacházení se závorkami a spojkami OR a AND. Na přiloženém obrázku viz obr. 12 lze vidět jeden z filtrů, specificky určený pro filtrování typu zařízení.

```

$typZar = $db->query("SELECT DISTINCT tbltypzar.Nazev, tbltypzar.TypZarID
                    FROM `tblzarizeni`, `tbltypzar`
                    WHERE tblzarizeni.FirmaID = " . $_SESSION["tbl_FirmaID"] . "
                    AND tblzarizeni.TypZarID = tbltypzar.TypZarID;");
$podminka_typZar = "";
$pocet_podminek = 0;

foreach ($typZar as $key) {
    if (!empty($_POST["filtr_typZar" . $key["TypZarID"]])) {
        $pocet_podminek++;
        if ($pocet_podminek == 1) {
            $podminka_typZar = " TypZarID = " . $key["TypZarID"];
        } elseif ($pocet_podminek > 1) {
            $podminka_typZar .= " OR TypZarID = " . $key["TypZarID"];
        }
        $filtry .= $key["Nazev"] . ":";
    }
}

```

Obr. 13 Filtry; Zdroj: Vlastní zpracování

Druhý problém, který byl řešen, se týká řazení záznamů v hlavní tabulce viz obr. 13. Řazení je v některých ohledech stejné jako filtrování, přičemž v obou případech se pracuje s databází. V tomto případě je do procesu přidán řetězec obsahující SQL příkaz ORDER BY, která slouží k uspořádání záznamů v databázi. Kvůli požadavku na změnu řazení při kliknutí na stejnou ikonu je nezbytné porovnávat aktuální stav pomocí podmínek a následně ji upravit.

```

if ($_POST["razeni"] == "typZar") {
    if ($_SESSION["razeni"] == "ORDER BY Typ_zarizeni") {
        $_SESSION["razeni"] = "ORDER BY Typ_zarizeni DESC";
        $_SESSION["razeni_ikona"] = "Z-A"
    } elseif ($_SESSION["razeni"] == "ORDER BY Typ_zarizeni DESC") {
        $_SESSION["razeni"] = "";
        $_SESSION["razeni_ikona"] = "";
    } else {
        $_SESSION["razeni"] = "ORDER BY Typ_zarizeni";
        $_SESSION["razeni_ikona"] = "A-Z"
    }
}

```

Obr. 14 Řazení; Zdroj: Vlastní zpracování

Další důležitou částí projektu je žurnál, který slouží k automatickému zaznamenávání událostí a změn spojených se zařízeními viz obr. 14. Tyto změny mohou zahrnovat například aktualizaci pracovníka přiřazeného k zařízení, změny umístění zařízení nebo přesuny na jiná střediska. Žurnál systematicky eviduje datum, popis provedených změn a nové hodnoty jednotlivých parametrů. V případě

změny událostí, které nejsou součástí standardních parametrů, je možné záznam provést i manuálně. Administrátor má také možnost upravit automaticky generované popisy.

```
if ($_POST["Pracovnik"] != $vlastnosti["Pracovnik"]) {
    $update_vlastnosti = $update_vlastnosti . " `Pracovnik`='" . $_POST["Pracovnik"] . "',";
    $popis = $db->query("SELECT Popis FROM `tblpopiszurnal` WHERE Nazev = 'Pracovnik';")->fetchColumn();
    $db->query("INSERT INTO `tblzurnal` (`ZarizeniID`, `Datum`, `Popis`)
VALUES ( '" . $_SESSION["ZarizeniID"] . "', '" . date("Y-m-d") . "', '" . preg_replace('/%.*?%/','
$_POST["Pracovnik"], $popis) . "'");
}
if ($_POST["Model"] != $vlastnosti["Model"]) {
    $update_vlastnosti = $update_vlastnosti . " `Model`='" . $_POST["Model"] . "',";
    $popis = $db->query("SELECT Popis FROM `tblpopiszurnal` WHERE Nazev = 'Model';")->fetchColumn();
    $db->query("INSERT INTO `tblzurnal` (`ZarizeniID`, `Datum`, `Popis`)
VALUES ( '" . $_SESSION["ZarizeniID"] . "', '" . date("Y-m-d") . "', '" . preg_replace('/%.*?%/','
$_POST["Model"], $popis) . "'");
}
```

Obr. 15 Zápis do žurnálu; Zdroj: Vlastní zpracování

Pro zajištění bezpečnosti ukládání hesel do databáze na stránce s přihlašováním byla implementována účinná ochrana. Byl zvolen jednoduchý, ale efektivní přístup k hashování neboli šifrování hesel, což znamená, že hesla nejsou uložena v databázi v čitelné podobě. I v případě, že by byl útočník úspěšný v zachycení komunikace se serverem, nebude schopen přečíst heslo díky použití soli. Bezpečnostní opatření spočívají v přidání soli ke každému heslu, což zvyšuje komplexnost hesla a ztěžuje jeho odhadnutí. Pro hashování byla využita PHP metoda crypt() s konstantou soli DEFAULT_PASSWORD. I když toto zabezpečení není na úrovni náhodně generované soli, lze jej považovat za dostatečné, zejména vzhledem k tomu, že webová aplikace běží v zašifrovaném protokolu HTTPS, což poskytuje další vrstvu ochrany.

Při odolávání dalším typům útoků, jako je přetížení databáze dotazy, účinně přispívá reCAPTCHA od společnosti Google, na kterou je více zaměřena následující část věnovaná JavaScriptu.

4.4 HTML

V procesu vytváření front-endu byl využit jazyk HTML ke strukturování obsahu webových stránek. Pro organizaci a rozložení obsahu bylo rozhodnuto využít blokové elementy, které jsou v HTML značeny tagem <div>. Tento tag je párový, což

znamená, že každý otevírací tag <div> musí být následován uzavíracím tagem </div>. Bloky byly využity kvůli jejich flexibilitě při rozložení stránky.

V rámci struktury stránky byly také intenzivně využity nadpisy, které jsou značeny tagy <h1> až <h6>. Tyto nadpisy jsou párové tagy a jejich použití může být klíčové pro hierarchii a strukturu obsahu na stránce. Každý nadpis byl zakončen uzavíracím tagem s příslušným číslem, například </h1>.

V rámci inventarizační stránky zaujímá klíčové místo využití tagů <table> pro vytvoření tabulky, která systematicky prezentuje informace o zařízeních. Současně jsou integrovány další klíčové tagy, jako <tr> pro přidání nového řádku, <th> pro definování buňky hlavičky a <td> pro vytvoření buňky v těle tabulky. Všechny tyto značky jsou párové, což vyžaduje pečlivé uzavírání, aby byla zachována správná struktura HTML. Tabulka významně přispívá k systematickému zobrazení informací a umožňuje uživatelům získat komplexní přehled o spravovaných zařízeních a jejich základních údajích. Správné uspořádání tabulek a řádků je klíčové pro efektivní prezentaci a zlepšení uživatelského prostředí.

Celkově lze říci, že proces vytváření front-endu byl úspěšně zrealizován s důrazem na efektivní využití jazyka HTML a jeho klíčových prvků. Blokované elementy, zejména tag <div>, byly široce aplikovány pro strukturování a flexibilní rozložení obsahu webových stránek. Nadpisy <h1> až <h6>, posloužily jako prostředek pro logické uspořádání obsahu.

```
<div class="klientskastrana-tabulka-prehled-blok">
  <h1>Uživatelé</h1>
  <table>
    <tr>
      <th class="nadpis-tabulka">Jméno</th>
      <th class="nadpis-tabulka">Příjmení</th>
      <th class="nadpis-tabulka">Uživatelské jméno</th>
      <th class="nadpis-tabulka">Telefon</th>
      <th class="nadpis-tabulka">Email</th>
    </tr>
  </table>
</div>
```

Obr. 16 Ukázka HTML kódu; Zdroj: Vlastní zpracování

4.5 CSS

Ve stylizačním jazyce CSS byla využita technika nazvaná Flexbox, která umožňuje jednoduché a efektivní uspořádání prvků na stránce. Flexbox přináší responzivitu a flexibilitu rozložení stránky, což je důležité pro různé typy zařízení a velikosti obrazovek. Tato technika výrazně zjednodušuje design a umožňuje snadnou manipulaci s umístěním a rozmístěním prvků.

Flex-container se stává blokem, na který je aplikována vlastnost `display: flex`. Všechny elementy uvnitř tohoto bloku jsou automaticky považovány za flex-items, což znamená, že mohou být flexibilně uspořádány.

Další důležitou vlastností Flexboxu je `flex-direction`, která ovlivňuje směr, ve kterém se flex-items uspořádávají. Tato vlastnost může být nastavena na hodnoty jako `row` (vodorovné uspořádání), `column` (svislé uspořádání), `row-reverse` a `column-reverse` (obrácené uspořádání).

Pro horizontální zarovnání flex-items byla využita vlastnost `justify-content`, která může nabývat hodnot jako `flex-start`, `flex-end`, `center`, `space-between` a `space-around`. Pro vertikální zarovnání byla využita vlastnost `align-items` s podobnými hodnotami jako `justify-content`.

Pro dosažení jednotného a esteticky působícího vizuálního dojmu byly zvoleny čtyři barvy, přičemž dvě z nich jsou v odstínu modré, aby tvořily harmonický celek. Tyto barvy byly pečlivě vybrány s ohledem na stávající grafiku firmy, konkrétně se jedná o dva odstíny modré: #001957 a #16479E. Pro zachování čistoty designu byly ponechány standardní barvy pro text a pozadí, konkrétně bílá a černá. Pro textový obsah na všech stránkách, ať už ve veřejné nebo klientské části, byl zvolen osvědčený a jednoduchý font Arial. Logo, které provází celou aplikaci, je upravenou

verzí stávajícího firemního loga. Byla změněna pouze barva, pokud byla tato úprava nezbytná.

V dnešní době, kdy většina uživatelů prochází webové stránky pomocí mobilních zařízení, nabývá responzivní design značného významu. Responzivní design umožňuje webovým stránkám přizpůsobit se různým rozměrům obrazovek, což zahrnuje jak změnu rozložení, tak i přizpůsobení obsahu v závislosti na orientaci zařízení. V tomto projektu je kladen důraz na implementaci responzivního designu pouze pro veřejnou část webových stránek. Pro dosažení tohoto cíle byla použita metoda CSS nazývaná @media s definovanou maximální šířkou okna pro zobrazení viz obr. 17. Tato technika umožňuje definovat specifické styly, které se aplikují v případě, že šířka okna prohlížeče nepřekročí stanovenou hodnotu.

```
@media only screen and (max-width: 1200px) {  
  
  .verejnastranka-hlavicka {  
    height: 80px;  
  }  
  
  .logo-hlavicka {  
    width: 127px;  
    height: 127px;  
    top: 12px;  
  }  
}
```

Obr. 17 Ukázka kódu responzivity; Zdroj: Vlastní zpracování

První část veřejné stránky, kterou návštěvník zaznamená, představuje úvodní sekci viz obr. 15. Zde se nachází logo v levém horním rohu, které se částečně prolíná s obrázkem v pozadí. Vedle loga je umístěn důležitý telefonní kontakt, následovaný jednoduchým menu obsahujícím tři body odkazující na ostatní části stránky. Vpravo se nachází ikona na přihlášení do klientské sekce. Obrázek v pozadí na pravé straně

částečně zakrývá sloupec s hlavním nadpisem, který charakterizuje zaměření firmy. Tento sloupec se dále rozprostírá i do dalších částí stránky.



Obr. 18 Úvodní stránka; Zdroj: Vlastní zpracování

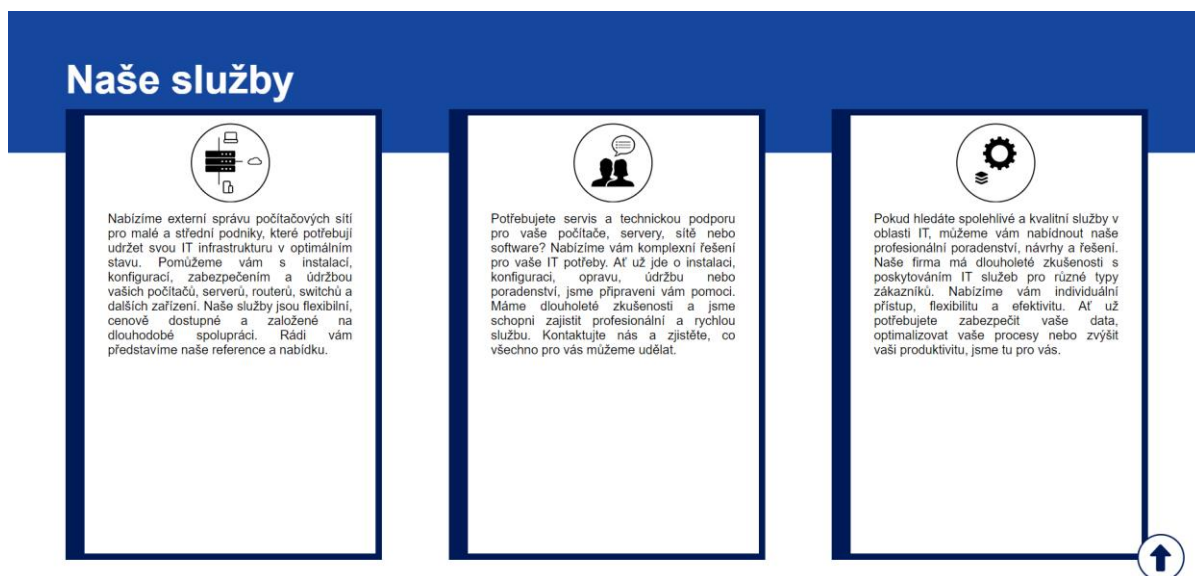
Druhou částí je stručný popis činností firmy viz obr. 16. Tato část je pomyslně rozdělena do dvou sloupců. V levém sloupci je uveden bodový výčet služeb poskytovaných firmou, následovaný rychlým odkazem pro stažení aplikace pro vzdálenou podporu. V pravém sloupci je umístěn textový blok, který detailněji

charakterizuje společnost. Tento text je umístěn ve stejném sloupci jako hlavní nadpis na úvodní stránce.



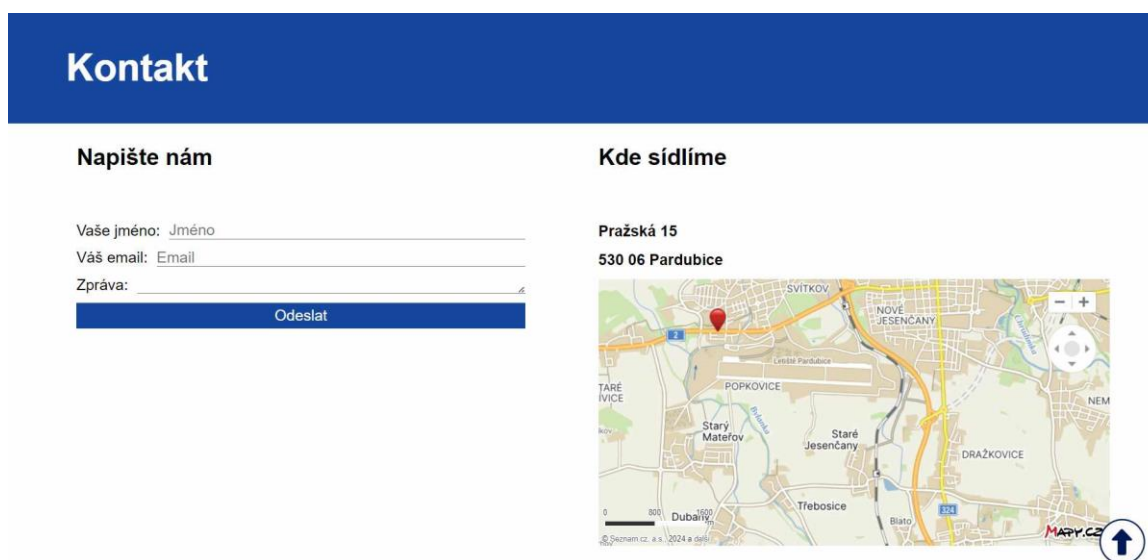
Obr. 19 Stránka o nás; Zdroj: Vlastní zpracování

Předposlední částí stránky je detailní popis poskytovaných služeb viz obr. 17. Tyto služby jsou uspořádány do samostatných buněk, přičemž každá buňka obsahuje ikonu charakterizující příslušnou službu umístěnou nahoře, následovanou popisem této služby. Každá buňka je ohraničena, aby vynikla od okolního pozadí.



Obr. 20 Naše služby; Zdroj: Vlastní zpracování

Poslední sekce ve veřejné části webové stránky je věnována kontaktním informacím viz obr. 18. Tato část je rozdělena do dvou sloupců, přičemž v levém sloupci se nachází kontaktní formulář pro kontaktování firmy. Pro odeslání zprávy stačí vyplnit jméno, email a textovou zprávu. V pravém sloupci je uvedena adresa společnosti a interaktivní mapa, která umožňuje snadnější orientaci.



Obr. 21 Kontaktní strana; Zdroj: Vlastní zpracování

V části klientského rozhraní webové aplikace hraje klíčovou roli tabulka, která prezentuje spravovaná zařízení s jejich důležitými informacemi. Pro úpravu vzhledu tabulky byly aplikovány specifické styly, které slouží k jejímu formátování a estetickému zpracování.

Nejprve byla tabulka zbavena vnějších ohraničení pomocí vlastnosti border-collapse. Nadpisy jednotlivých sloupců v prvním řádku tabulky byly dále stylovány pomocí vlastností border-bottom pro podtržení řádku, padding pro vnitřní odsazení a text-align pro zarovnání textu. Stejně styly byly aplikovány i na ostatní buňky tabulky, avšak s jinými hodnotami pro vytvoření jednotného vizuálního dojmu. Každý lichý řádek tabulky byl zvýrazněn pomocí pseudotřídy nth-child() s odlišným pozadím, což usnadňuje vizuální orientaci a čtení obsahu. Dále byla

implementována pseudotřída hover, která zajišťuje zvýraznění řádku, na který je kurzor myši zaměřen.

Tímto způsobem byla tabulka graficky upravena a zdokonalena pomocí základních prvků, což přispívá k lepší přehlednosti a uživatelskému komfortu při práci s webovou aplikací. Všechny stránky klientské části sdílejí jednotnou strukturu, která zahrnuje stejnou hlavičku, levý sloupec a hlavní obsahovou část. Hlavička se nachází v horní části a obsahuje firemní logo v levém rohu a vpravo od něj pak rozbalovací menu umožňující výběr firmy. V pravém rohu je zobrazeno uživatelské jméno přihlášeného uživatele. Pod hlavičkou je obsah rozdělen do dvou sloupců. V menším levém sloupci jsou umístěny odkazy na ostatní stránky, zatímco v pravém sloupci se nachází hlavní obsah.

Úvodní stránka poskytuje přehledové informace viz obr. 19. Hlavní část této stránky obsahuje tabulky, které zobrazují uživatele, střediska, umístění a počty zařízení podle jejich typu.

The screenshot shows a web application interface with a blue header and a light blue sidebar. The header contains a logo on the left and the text 'Administrátor' on the right. The sidebar has four buttons: 'Přehled', 'Zařízení', 'Uživatelé', and 'Nastavení'. The main content area is divided into four sections:

- Střediska**: A table with columns 'Zkratka', 'Název', and 'Přidat středisko'. It contains one row with a blurred entry.
- Umístění**: A table with columns 'Zkratka', 'Adresa', 'Wan IP', 'Síť LAN', and 'Přidat umístění'. It contains three rows with blurred entries.
- Uživatelé**: A table with columns 'Jméno', 'Příjmení', 'Uživatelské jméno', 'Telefon', and 'Email'. It contains one row with blurred entries and a 'Zobrazit všechny uživatele' link below.
- Počty zařízení**: A table with two columns: device type and count. It contains three rows: 'Počítač' (2), 'Notebook' (1), and 'Switch' (1). A 'Zobrazit všechny zařízení' link is at the bottom.

Obr. 22 Přehled; Zdroj: Vlastní zpracování

Druhou a nejdůležitější stránkou z klientské části je stránka zobrazující soupis zařízení zvolené firmy viz obr. 20. V levém sloupci přibyly oproti zbylým stránkám filtry pro lepší a rychlejší výběr zařízení.

Typ zařízení	Označení	Pracovník	Datum zařazení	Umístění	Přidat zařízení
PC			06 2021	NL	☰
PC			02 2024	NC	☰
NB			01 2024	NC	☰
PC			11 2023	NE	☰
SW			01 2021	NL	☰

Obr. 23 Výpis zařízení; Zdroj: Vlastní zpracování

Další částí systému, která je dostupná pouze pro administrátora, je seznam uživatelů viz obr. 21. Zde má administrátor možnost upravovat informace existujících uživatelů, přiřazovat je k jednotlivým firmám nebo přidávat úplně nové uživatele.

Firma	Jméno	Příjmení	Uživatelské jméno	Telefon	Email	Přidat uživatele
						☰
				000 213 987		☰
				000 010 111		☰
				+420 278 234 342		☰

Obr. 24 Výpis uživatelů; Zdroj: Vlastní zpracování

Poslední stránka, dostupná opět pouze pro administrátora nese název nastavení viz obr. 22. Na této stránce lze spravovat firmy a také upravovat automatické akce vypisované do žurnálu při změně údajů u zařízení.

Název	Přidat firmu
[redacted]	[edit icon]
[redacted]	[edit icon]
[redacted]	[edit icon]

Název	Popis
Umístění	Umístění bylo změněno na [edit icon]
Středisko	Bylo změněno středisko na [edit icon]
Pracovník	Byl změněn pracovník na [edit icon]
Evcislo	Evidenční číslo bylo změněno na [edit icon]

Obr. 25 Nastavení; Zdroj: Vlastní zpracování

V aplikaci jsou k dispozici další tabulky, které slouží k zobrazení podrobnějších údajů o zařízeních viz obr. 23. Tyto tabulky se zobrazí po kliknutí na ikonu s podrobnostmi, umístěnou na konci řádku každého zařízení. Po stisknutí této ikony se otevře okno v pravé části stránky, kde jsou zobrazeny detaily týkající se daného

zařízení. V tomto okně naleznete tabulku obsahující všechny vlastnosti zařízení a po kliknutí na tlačítko parametry se zobrazí rovněž tabulka s jeho parametry.

The screenshot shows a web application interface with a blue header and a sidebar on the left. The main content area displays a table with the following data:

Typ zařízení	Označení	Pracovník	Datum zařazení
PC	1		08 2021
PC			02 2024
NB			01 2024
PC			11 2023
SW			01 2021

The sidebar on the left contains several filter sections: 'Typ zařízení' (with radio buttons for 'Položka', 'Notebook', and 'Switch'), 'Umístění' (with a dropdown menu), 'Rok zařazení' (with radio buttons for '2021', '2024', and '2023'), and 'Pracovník' (with a text input field and a 'Filtrovat' button). The right side of the interface shows a 'Vlastnosti' (Properties) section with a 'Znamení' (Label) and 'Parametry' (Parameters) section. The 'Vlastnosti' section includes fields for 'Evidenční číslo', 'Pořizeno', 'Umístění', 'Středisko', 'Pracovník', 'Model', and 'Poznámka'.

Obr. 26 Podrobnosti zařízení; Zdroj: Vlastní zpracování

4.6 JavaScript

JavaScript je interpretovaným programovacím jazykem, který se využívá převážně při vývoji webových aplikací. Jeho významné výhody a důvody, proč je preferován při tvorbě webových stránek, spočívají zejména ve schopnosti skriptování na straně klienta. Tato vlastnost umožňuje provádění procesů na straně klienta, což eliminuje potřebu obnovování celé stránky. Kromě toho jsou nabízeny další přínosy, jako jsou objektově orientovaný přístup a nestriktní typování. Díky nestriktnímu typování není nutné explicitně specifikovat datové typy proměnných, což zjednodušuje a urychluje proces programování.

V projektu bylo k tomuto jazyku přistupováno zřídka. První využití zahrnovalo vytvoření krátkého skriptu na veřejné stránce, který měl za úkol zobrazit tlačítko sloužící k návratu na úvodní stránku viz obr. 24. V přiloženém kódu může být identifikována jedna proměnná, která je deklarována klíčovým slovem "let". Tato proměnná nese název "topButton" a slouží k uchování objektu tlačítka. Dále se v kódu nachází funkce nazvaná "scrollFunction", kde je prováděna jednoduchá podmínka. Tato podmínka zjišťuje, zda se uživatel nachází úplně nahoře na stránce,

nebo se nachází 20 bodů pod horním okrajem. Pokud je podmínka splněna, nastaví se styl tlačítka na "block", což znamená, že je viditelné. V opačném případě, když podmínka není splněna, je styl nastaven na "none", což značí, že tlačítko není viditelné.

```
<script>
  let topButton = document.getElementById("topButton");

  window.onscroll = function() {
    scrollFunction()
  };

  function scrollFunction() {
    if (document.body.scrollTop > 20 || document.documentElement.scrollTop > 20) {
      topButton.style.display = "block";
    } else {
      topButton.style.display = "none";
    }
  }
</script>
```

Obr. 27 TopButton; Zdroj: Vlastní zpracování

V dalším scénáři byl JavaScript využit k implementaci interaktivní mapy viz obr. 25 na úvodní stránce, konkrétně se jedná o mapu od společnosti Mapy.cz. Na prvním řádku je HTML div element, který slouží jako kontejner pro implementaci mapy následným skriptem. Při detailnějším zkoumání skriptu je identifikováno pět proměnných s názvy "center", "m", "layer", "options" a "marker". Proměnná "center" obsahuje objekt typu Coords, který reprezentuje zeměpisné souřadnice. Proměnná "m" vytváří instanci objektu SMap, což představuje mapové zobrazení. "Layer" vytváří novou vrstvu pro zobrazení značek na mapě. Proměnná "options" obsahuje

prázdný objekt, který slouží pro případnou konfiguraci značek. Poslední proměnná "marker" vytváří instanci objektu Marker, což znamená značky umístěné na mapě.

```
<div id="mapa" class="verejnastranka-mapa"></div>
<script type="text/javascript">
    var center = SMap.Coords.fromWGS84(15.7213686, 50.0206314);
    var m = new SMap(JAK.gel("mapa"), center, 17);
    m.addDefaultLayer(SMap.DEF_BASE).enable();
    m.addDefaultControls();

    var layer = new SMap.Layer.Marker();
    m.addLayer(layer);
    layer.enable();

    var options = {};
    var marker = new SMap.Marker(center, "myMarker", options);
    layer.addMarker(marker);
</script>
```

Obr. 28 Implementace mapy; Zdroj: Vlastní zpracování

Dalším příkladem využití tohoto programovacího jazyka je jeho role v oblasti zabezpečení. Při procesu přihlášení na klientskou stránku byla implementována reCAPTCHA od společnosti Google viz obr. 26. Tato bezpečnostní funkce slouží k ochraně před spamovými útoky a útoky automatizovaných botů. Hlavním účelem reCAPTCHA je rozlišit lidské uživatele a automatizované boty. Při její implementaci může být uživatel vyzván k provedení konkrétních akcí nebo vyplnění testu, který je pro lidi relativně snadný, zatímco pro boty představuje výzvu. Implementace této funkce na webové stránky je velmi jednoduchá a provádí se načítáním skriptu ze zadaného odkazu, jak je vidět v níže uvedeném ukázkovém kódu. Podobně jako při implementaci mapy je třeba vytvořit HTML div element, do kterého bude tato funkce implementována.

```
<div class="g-recaptcha" data-sitekey="your_site_key"></div>
<script src="https://www.google.com/recaptcha/api.js" async defer></script>
```

Obr. 29 Implementace recaptchy; Zdroj: Vlastní zpracování

4.7 Databáze

Databáze slouží k ukládání dat o zařízeních, uživateli a firmách. Informace zahrnují umístění, pracovníka, rok nákupu, označení, model a další relevantní údaje.

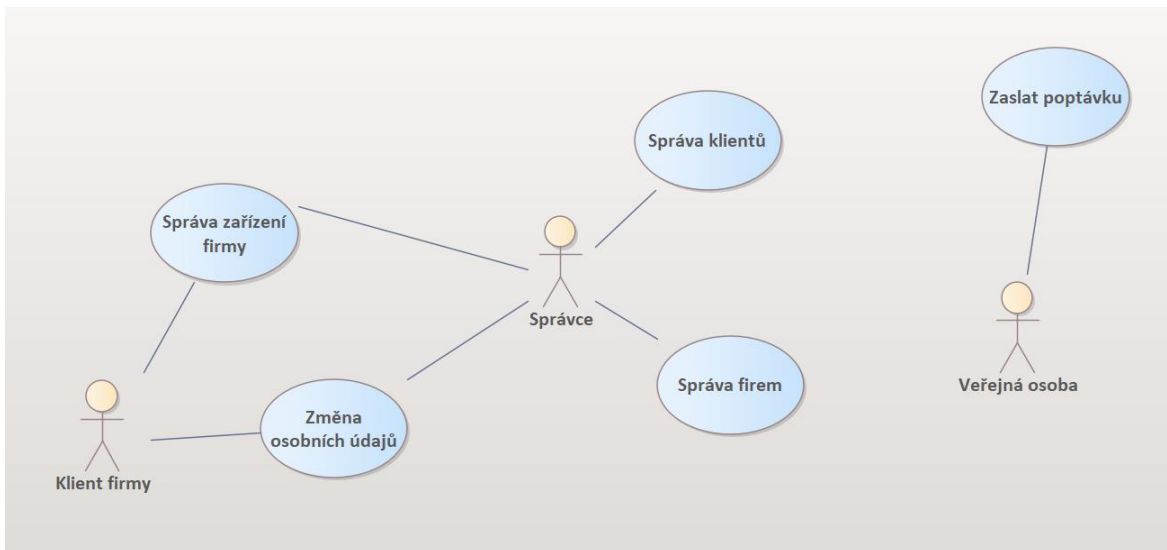
Struktura databáze je navržena tak, aby efektivně organizovala a uchovávala tato data.

Další fáze se zaměřuje na funkcionality navrhované databáze pomocí diagramu případu užití viz obr. 27, který poskytuje komplexní pohled na interakce mezi různými aktéry systému. V tomto případě jsou identifikováni tři hlavní aktéři: Klient firmy, Správce a Veřejná osoba.

Klient firmy reprezentuje pověřeného uživatele spravované firmy, s možností upravovat své údaje, včetně hesla, e-mailu a telefonního čísla. Má také právo spravovat zařízení své firmy, včetně přidání nových zařízení, úpravy existujících nebo jejich odstranění.

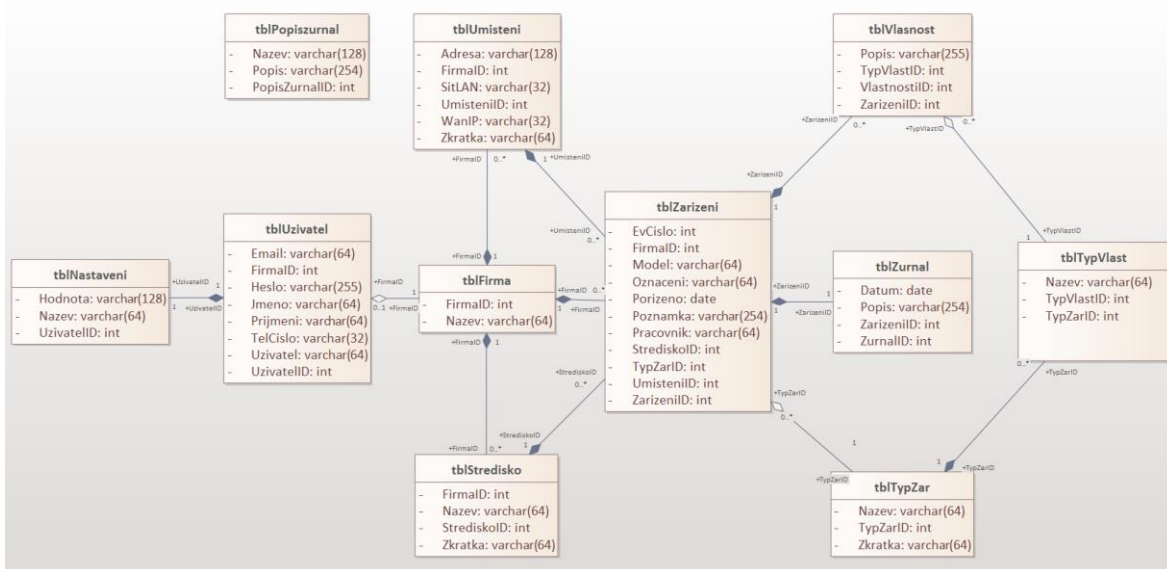
Správce je hlavním aktérem s rozsáhlými možnostmi. Má schopnost spravovat všechny firmy v systému, včetně uživatelů a zařízení. Podobně jako Klient firmy může správce provádět změny ve vlastních osobních údajích.

Veřejná osoba představuje třetího aktéra, který má pouze pozorovatelskou roli. Má přístup pouze k veřejné stránce, kde získá informace o firmě, ale nemá oprávnění k administrativním funkcím, včetně správy zařízení. Přesto má možnost projevit zájem o spolupráci zadáním nezávazné poptávky o budoucí spolupráci, což poskytuje potenciální cestu k navázání nových obchodních vztahů.



Obr. 30 Diagram případu užití; Zdroj: Vlastní zpracování

Za účelem podrobnějšího grafického zobrazení struktury databáze je dále zobrazen diagram tříd viz obr. 28, který pečlivě analyzuje jednotlivé tabulky, jejich vzájemné vazby a atributy. V tomto schématu jsou jasně identifikovatelné následující tabulky: tblFirma, tblNastaveni, tblStredisko, tblTypVlast, tblTypZar, tblUmisteni, tblUzivatel, tblVlastnost, tblZarizeni a tblZurnal. Pro správnou strukturu a provázanost databáze jsou klíčové vztahy mezi těmito tabulkami.



Obr. 31 Diagram tříd; Zdroj: Vlastní zpracování

Na diagramu se objevují dva hlavní typy vazeb: agregace a kompozice. Agregáčn  vazby jsou p řítomny mezi tabulkami tblFirma a tblUzivatel, tblTypZar a tblZarizeni, tblTypVlast a tblVlastnost. Tento typ vazby se osv dčuje v p řípadech, kdy pot řebujeme flexibiln  propojen  mezi celkem a jeho částmi, p řičemž část m že existovat nezávisle na celku. Nap řıklad, Firma m že existovat i bez Uživatele.

Druhým použitým typem vazby je kompozice, coř představuje siln jší formu agregace, kde existence části závis  na existenci celku. Tato vazba je mezi tabulkami tblNastaveni a tblUzivatel, tblUmisteni a tblFirma, tblStredisko a tblFirma, tblZarizeni a tblFirma, tblVlastnost a tblZarizeni, tblZurnal a tblZarizeni, tblTypVlast a tblTypZar.

Takov  strukturov n  vazeb poskytuje kl čov  vhlad do komplexn  organizace datab ze, coř je nezbytn  pro efektivn  spr vu a manipulaci s daty v r mci vyv jen  webové aplikace. D ky tomuto p řistupu lze zajistit korektnost dat a optim ln  fungov n  syst mu.

N že uveden  kapitoly (kap. 4.7.1 – kap. 4.7.11) poskytuj  podrobn  popis jednotliv ch v še zmiňovan ch tabulek.

4.7.1 tblFirma:

Slouř pro z znamy firem, které jsou spravov ny. Obsahuje pouze dva atributy FirmaID (id firmy, PK) a Nazev.

#	N�zev	Typ	Porovn�v�n�	Vlastnosti	Pr�zdn�	V�choz�	Koment�ře	Dalř
1	FirmaID	int(11)			Ne	ř�dn�		AUTO_INCREMENT
2	Nazev	varchar(64)	utf8_czech_ci		Ano	NULL		

Obr. 32 tblFirma; Zdroj: Vlastn  zpracov n 

4.7.2 tblNastaveni:

Tabulka, která n m zajist , aby uřivatel  i po odhl řen  a n sledn m op tovn m p řihl řen  m li vybr ny stejn  filtry a dalřn  nastaven  zobrazen . Obsahuje t ř



atributy, a to **Nazev** (pro název session, kterou potřebujeme uložit), **Hodnota** (hodnota dané session) a **UzivateliID** (id uživatele, FK).

#	Název	Typ	Porovnávání	Vlastnosti	Prázdný	Výchozí	Komentáře	Další
1	Nazev	varchar(64)	utf8_czech_ci		Ne	Žádná		
2	Hodnota	varchar(128)	utf8_czech_ci		Ano	NULL		
3	UzivateliID	int(11)			Ne	Žádná		

Obr. 33 tblNastaveni; Zdroj: Vlastní zpracování

4.7.3 tblStredisko:

Střediska jsou dalším důležitým údajem. Protože jedna firma může mít více středisek je potřeba další tabulka. **TblStredisko** má následující atributy **StrediskoID** (id střediska, PK), **FirmaID** (id firmy pro středisko, FK), **Zkratka** a **Nazev**.



#	Název	Typ	Porovnávání	Vlastnosti	Prázdný	Výchozí	Komentáře	Další
1	StrediskoID 	int(11)			Ne	Žádná		AUTO_INCREMENT
2	FirmaID 	int(11)			Ano	NULL		
3	Zkratka	varchar(64)	utf8_czech_ci		Ano	NULL		
4	Nazev	varchar(64)	utf8_czech_ci		Ano	NULL		

Obr. 34 tblStredisko; Zdroj: Vlastní zpracování

4.7.4 tblTypVlast:

Existuje více druhů zařízení, které je spravováno a každé má jiné vlastnosti. Pro určení vlastností k zařízení slouží tato tabulka s následujícími atributy. **TypVlastID** (id typu vlastnosti, PK), **TypZarID** (id typu zařízení, FK), **Nazev** **Poradi** (určuje,


v jakém pořadí se budou vlastnosti zobrazovat), TypHodn (některé hodnoty budou vypisovány jako text a jiné pouze jako ano/ne).

#	Název	Typ	Porovnávání	Vlastnosti	Prázdný	Výchozí	Komentáře	Další
1	TypVlastID 	int(11)			Ne	Žádná		AUTO_INCREMENT
2	TypZarID 	int(11)			Ano	NULL		
3	Nazev	varchar(64)	utf8_czech_ci		Ano	NULL		
4	Poradi	int(11)			Ne	1		
5	TypHodn	varchar(16)	utf8_czech_ci		Ne	text		

Obr. 35 tblTypVlast; Zdroj: Vlastní zpracování

4.7.5 tblTypZar:

Tabulka typ zařízení byla zmíněna již v tabulce typ vlastnosti. Obsahuje TypZarID (id tyou zařízení, PK), Zkratka a Nazev.



#	Název	Typ	Porovnávání	Vlastnosti	Prázdný	Výchozí	Komentáře	Další
1	TypZarID 	int(11)			Ne	Žádná		AUTO_INCREMENT
2	Zkratka	varchar(64)	utf8_czech_ci		Ano	NULL		
3	Nazev	varchar(64)	utf8_czech_ci		Ano	NULL		

Obr. 36 tblTypZar; Zdroj: Vlastní zpracování

4.7.6 tblUmisteni:

Tabulky umístění a středisko se mohou zdát stejné, ovšem obě jsou potřeba a mají jiný význam. Středisko je označení pro oddělení, kde více středisek může být na stejném místě a středisek se stejným názvem může být více na různých místech, proto potřebujeme znát i umístění střediska, které nám určí jeho polohu a další informace podle následujících hodnot. UmisteniID (id umístění, PK), FirmaID (id

firmy daného umístění, FK), Zkratka, Adresa (geografická adresa), WanIP (Wanovou adresu pro vzdálený přístup) a SitLan (místní rozsah adres).

#	Název	Typ	Porovnávání	Vlastnosti	Prázdný	Výchozí	Komentáře	Další
1	UmistenID 	int(11)			Ne	Žádná		AUTO_INCREMENT
2	FirmaID 	int(11)			Ne	Žádná		
3	Zkratka	varchar(64)	utf8_czech_ci		Ano	NULL		
4	Adresa	varchar(128)	utf8_czech_ci		Ano	NULL		
5	WanIP	varchar(32)	utf8_czech_ci		Ano	NULL		
6	SitLAN	varchar(32)	utf8_czech_ci		Ano	NULL		

Obr. 37 tblUmistení; Zdroj: Vlastní zpracování

4.7.7 tblUzivatel:

Jak bylo již zmíněno do webové aplikace bude mít přístup více uživatelů, proto musí být uchovány jejich následující údaje. UzivatelID (id uzivatele, PK), FirmaID (id firmy, pro kterou uživatel pracuje, FK), Jmeno, Prijmeni, TelCislo (telefonní číslo), Email, Uzivatel (uživatelské jméno), Heslo.

#	Název	Typ	Porovnávání	Vlastnosti	Prázdný	Výchozí	Komentáře	Další
1	UzivatelID 	int(11)			Ne	Žádná		AUTO_INCREMENT
2	FirmaID 	int(11)			Ano	NULL		
3	Jmeno	varchar(64)	utf8_czech_ci		Ano	NULL		
4	Prijmeni	varchar(64)	utf8_czech_ci		Ano	NULL		
5	TelCislo	varchar(32)	utf8_czech_ci		Ano	NULL		
6	Email	varchar(64)	utf8_czech_ci		Ano	NULL		
7	Uzivatel	varchar(64)	utf8_czech_ci		Ne	Žádná		
8	Heslo	varchar(255)	utf8_czech_ci		Ne	Žádná		

Obr. 38 tblUzivatel; Zdroj: Vlastní zpracování

4.7.8 tblVlastnosti:

Tato tabulka slouží pro přiřazení hodnot danému parametru pro zařízení. Tedy například, že u konkrétního počítače se nachází procesor typu Intel i5. VlastnostID (id vlastnosti, PK), ZarizeniID (id zařízení pro tuto vlastnost, FK), TypVlastID (id vlastnosti neboli parametru, FK), Popis (hodnota parametru).

#	Název	Typ	Porovnávání	Vlastnosti	Prázdný	Výchozí	Komentáře	Další
1	VlastnostID	int(11)			Ne	Žádná		AUTO_INCREMENT
2	ZarizeniID	int(11)			Ano	NULL		
3	TypVlastiD	int(11)			Ano	NULL		
4	Popis	varchar(254)	utf8_czech_ci		Ano	NULL		

Obr. 39 tblVlastnosti; Zdroj: Vlastní zpracování

4.7.9 tblZarizeni:



Jedná se o nejkompexnější tabulku v databázi. Nalezneme zde nejvíce atributů, ale složitost zde žádná není. Jedná se o tabulku, která popisuje veškerá zařízení v aplikaci. ZarizeniID (id zařízení, PK), FirmaID (id firmy, do které zařízení spadá, FK), EvCislo (evidenční číslo zařízení), Porizeno (datum, kdy bylo zařízení pořízeno, abychom znali jeho stáří), TypZarID (id typu zařízení, FK), Oznaceni (označení zařízení ve firmě), UmisteniID (id umístění, kde se zařízení nachází, FK), StrediskoID (id střediska, do kterého zařízení spadá, FK), Pracovnik (jméno pracovníka), Model, Poznámka.

#	Název	Typ	Porovnávání	Vlastnosti	Prázdný	Výchozí	Komentáře	Další
1	ZarizeniID	int(11)			Ne	Žádná		AUTO_INCREMENT
2	FirmaID	int(11)			Ano	NULL		
3	EvCislo	int(11)			Ano	NULL		
4	Porizeno	date			Ano	NULL		
5	TypZarID	int(11)			Ano	NULL		
6	Oznaceni	varchar(64)	utf8_czech_ci		Ano	NULL		
7	UmisteniID	int(11)			Ano	NULL		
8	StrediskoID	int(11)			Ano	NULL		
9	Pracovnik	varchar(64)	utf8_czech_ci		Ano	NULL		
10	Model	varchar(64)	utf8_czech_ci		Ano	NULL		
11	Poznamka	varchar(255)	utf8_czech_ci		Ano	NULL		

Obr. 40 tblZarizeni; Zdroj: Vlastní zpracování

4.7.10 tblZurnal:


Předposlední tabulka databáze slouží pro záznam historie, přesněji co bylo u zařízení změněno nebo přepsáno. ZurnalID (id žurnálu, PK), ZarizeniID (id zařízení, FK), Datum (kdy byla změna provedena), Popis (co bylo změněno).

#	Název	Typ	Porovnávání	Vlastnosti	Prázdný	Výchozí	Komentáře	Další
1	ZurnalID 	int(11)			Ne	Žádná		AUTO_INCREMENT
2	ZarizeniID 	int(11)			Ano	NULL		
3	Datum	date			Ano	NULL		
4	Popis	varchar(254)	utf8_czech_ci		Ano	NULL		

Obr. 41 tblZurnal; Zdroj: Vlastní zpracování

4.7.11 tblPopisZurnal:

Poslední tabulka je velmi jednoduchá a slouží pro správu automatických popisků v žurnálu. Tabulka obsahuje tři atributy, a to PopisZurnalID (id popisu, PK), Nazev a Popis.

#	Název	Typ	Porovnávání	Vlastnosti	Prázdný	Výchozí	Komentáře	Další
1	PopisZurnalID 	int(11)			Ne	Žádná		AUTO_INCREMENT
2	Nazev	varchar(128)	utf8_czech_ci		Ne	Žádná		
3	Popis	varchar(256)	utf8_czech_ci		Ne	Žádná		

Obr. 42 tblPopisZurnal; Zdroj: Vlastní zpracování

5 Shrnutí a diskuse výsledků

V rámci této práce se povedlo vytvořit fungující webovou aplikaci pro inventarizaci zařízení. Aplikace se dělí na dvě sekce. Publicistická stránka slouží pouze pro publikaci firmy. Hlavní částí webové aplikace je pak část druhá, která je zaměřena na inventarizaci a správu zařízení. Na začátku byly stanoveny následující cíle, a to přehled zařízení podle jednotlivých firem, úprava vlastností a parametrů u zařízení, automatický žurnál pro záznam změn u zařízení. Tyto služby mohou využívat všichni uživatelé, kteří mají přístup do systému. Ovšem to nebyly jediné požadované funkce stránky. Další možnosti se týkají pouze administrátorů, kteří navíc mohou spravovat uživatele, střediska a umístění u jednotlivých firem, dále pak i měnit zprávy pro automatický žurnál. Všechny tyto podmínky jsou v konečném projektu splněny, a to i přes drobné problémy s lokálním hostingem, které v průběhu vypracovávání nastaly.

6 Závěry a doporučení

Závěrem lze říci, že aplikaci se povedlo vytvořit podle prvotních představ. Webová aplikace je plně funkční, ale jako každý jiný vývoj webové aplikace nebo softwaru prochází několika opravami. Některé chyby se projeví až po chvíli používání aplikace v praxi. Lze tedy říci, že se aplikaci povedlo zpracovat, ovšem i po dokončení bude potřeba odstranit případné nedostatky a chyby pro správné fungování aplikace.

Aplikaci by bylo možné rozšířit o další funkcionality jako je například online fakturace nebo kalkulace ceny za provedené služby. Tyto služby by byly přístupné odkudkoli přes jakýkoli internetový prohlížeč hned po přihlášení. Dalším možným rozšířením by byla možnost zadání požadavku klienta přímo u daného zařízení v systému. Pracovníkovi by poté přišel email s vygenerovaným textem o daném požadavku.

Aplikace byla před nasazením do provozu podrobena lokálnímu testování firmou. Po nasazení do provozu bude aplikace ve zkušebním režimu, během kterého ji budou testovat vybraní klienti. V případě zjištění jakýchkoli problémů se bude co nejdříve pracovat na jejich odstranění. Stejně tak budou zohledněny a zpracovány veškeré návrhy na zlepšení, které se během testování vyskytnou.

Zpětná vazba od firmy je zatím velmi pozitivní. Aplikace splňuje veškeré požadavky a je možné, že v následující fázi dojde k rozšíření aplikace o další funkcionality.

7 Seznam použité literatury

APATHE NETBEANS, 2017. *Welcome to Apache NetBeans* [online] [vid. 2024-03-23]. Dostupné z: <https://netbeans.apache.org/front/main/index.html>

AVAST, 2024. *Avast Business Hub | Vícevrstvá platforma IT zabezpečení* [online] [vid. 2024-03-24]. Dostupné z: <https://www.avast.com/cs-cz/business/business-hub>

AVG, 2024. *Cloud Management Console – antivirové zabezpečení z cloudu | AVG* [online] [vid. 2024-03-24]. Dostupné z: <https://www.avg.com/cs-cz/management-console>

BOURDON, Romain, 2024. WampServer. *WampServer* [online] [vid. 2024-03-23]. Dostupné z: <https://www.wampserver.com/en/>

CONTRIBUTORS, phpMyAdmin, 2003. phpMyAdmin. *phpMyAdmin* [online] [vid. 2024-03-23]. Dostupné z: <https://www.phpmyadmin.net/>

GOOGLE, 2024. SEO Starter Guide: The Basics | Google Search Central | Documentation. *Google for Developers* [online] [vid. 2024-03-24]. Dostupné z: <https://developers.google.com/search/docs/fundamentals/seo-starter-guide>

HARTINGER, David, 2024a. *Lekce 1 - Úvod do UML* [online] [vid. 2024-04-03]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-uvod-historie-vyznam-a-diagramy>

HARTINGER, David, 2024b. *Lekce 2 - UML - Use Case Diagram* [online] [vid. 2024-03-24]. Dostupné z: <https://www.itnetwork.cz/uml-use-case-diagram>

HARTINGER, David, 2024d. *Lekce 4 - UML - Doménový model* [online] [vid. 2024-04-13]. Dostupné z: <https://www.itnetwork.cz/uml-domenovy-model-diagram>

HARTINGER, David, 2024c. *Lekce 5 - UML - Class diagram* [online] [vid. 2024-03-24]. Dostupné z: <https://www.itnetwork.cz/uml-class-diagram-tridni-model>

LIE, Håkon Wium a Bert BOS, 1999. *Cascading style sheets: designing for the Web*. 2. ed. Harlow Bonn: Addison Wesley. ISBN 978-0-201-59625-0.

MICROSOFT, 2024. *Správa Microsoftu 365* [online] [vid. 2024-03-24]. Dostupné z: <https://www.microsoft.com/cs-cz/microsoft-365/business/office-365-administration>

MYSQL, 2024. *MySQL :: MySQL Workbench* [online] [vid. 2024-03-23]. Dostupné z: <https://www.mysql.com/products/workbench/>

PHP, 2001e. *PHP: Connections and Connection management - Manual* [online] [vid. 2024-03-23]. Dostupné z: <https://www.php.net/manual/en/pdo.connections.php>

PHP, 2001c. *PHP: Introduction - Manual* [online] [vid. 2024-03-23]. Dostupné z: <https://www.php.net/manual/en/intro.pdo.php>

PHP, 2001f. *PHP: PDO::exec - Manual* [online] [vid. 2024-03-23]. Dostupné z: <https://www.php.net/manual/en/pdo.exec.php>

PHP, 2001h. *PHP: PDO::prepare - Manual* [online] [vid. 2024-03-23]. Dostupné z: <https://www.php.net/manual/en/pdo.prepare.php>

PHP, 2001g. *PHP: PDO::query - Manual* [online] [vid. 2024-03-23]. Dostupné z: <https://www.php.net/manual/en/pdo.query.php>

PHP, 2001ch. *PHP: PDOStatement::execute - Manual* [online] [vid. 2024-03-23]. Dostupné z: <https://www.php.net/manual/en/pdostatement.execute.php>

PHP, 2001i. *PHP: PDOStatement::fetchAll - Manual* [online] [vid. 2024-03-23]. Dostupné z: <https://www.php.net/manual/en/pdostatement.fetchall.php>

PHP, 2001j. *PHP: PDOStatement::fetchColumn - Manual* [online] [vid. 2024-03-23]. Dostupné z: <https://www.php.net/manual/en/pdostatement.fetchcolumn.php>

PHP, 2001d. *PHP: Prepared statements and stored procedures - Manual* [online] [vid. 2024-03-23]. Dostupné z: <https://www.php.net/manual/en/pdo.prepared-statements.php>

PHP, 2001b. *PHP: What can PHP do? - Manual* [online] [vid. 2024-03-23]. Dostupné z: <https://www.php.net/manual/en/intro-whatcando.php>

PHP, 2001a. *PHP: What is PHP? - Manual* [online] [vid. 2024-03-23]. Dostupné z: <https://www.php.net/manual/en/intro-whatis.php>

TEAMVIEWER, 2024. TeamViewer – Software pro vzdálené připojení. *TeamViewer* [online] [vid. 2024-03-24]. Dostupné z: <https://www.teamviewer.com/cs/>

TELEPORTHQ, 2023. Low-code Front-end Design & Development Platform | Teleporthq. *teleporthq.io* [online] [vid. 2024-04-03]. Dostupné z: <https://teleporthq.io/>

VISUAL STUDIO CODE, 2024b. *Documentation for Visual Studio Code* [online] [vid. 2024-03-23]. Dostupné z: <https://code.visualstudio.com/docs>

VISUAL STUDIO CODE, 2024a. *Visual Studio Code - Code Editing. Redefined* [online] [vid. 2024-03-23]. Dostupné z: <https://code.visualstudio.com/>

VISUAL STUDIO MARKETPLACE, 2024a. *HTML CSS Support - Visual Studio Marketplace* [online] [vid. 2024-03-23]. Dostupné z: <https://marketplace.visualstudio.com/items?itemName=ecmel.vscode-html-css>

VISUAL STUDIO MARKETPLACE, 2024b. *PHP Intelephense - Visual Studio Marketplace* [online] [vid. 2024-03-23]. Dostupné z: <https://marketplace.visualstudio.com/items?itemName=bmewburn.vscode-intelephense-client>

W3SCHOOLS, 1999e. *CSS Introduction* [online] [vid. 2024-03-23]. Dostupné z: https://www.w3schools.com/css/css_intro.asp

W3SCHOOLS, 1999b. *HTML Basic* [online] [vid. 2024-03-23]. Dostupné z: https://www.w3schools.com/html/html_basic.asp

W3SCHOOLS, 1999c. *HTML Forms* [online] [vid. 2024-03-24]. Dostupné z: https://www.w3schools.com/html/html_forms.asp

W3SCHOOLS, 1999a. *Introduction to HTML* [online] [vid. 2024-03-23]. Dostupné z: https://www.w3schools.com/html/html_intro.asp

W3SCHOOLS, 1999l. *JavaScript DOM Methods* [online] [vid. 2024-03-24]. Dostupné z: https://www.w3schools.com/js/js_htmlDOM_methods.asp

W3SCHOOLS, 1999k. *JavaScript Objects* [online] [vid. 2024-03-24]. Dostupné z: https://www.w3schools.com/js/js_objects.asp

W3SCHOOLS, 1999j. *JavaScript Where To* [online] [vid. 2024-03-24]. Dostupné z: https://www.w3schools.com/js/js_whereTo.asp

W3SCHOOLS, 1999f. *Selektory CSS* [online] [vid. 2024-03-23]. Dostupné z: https://www.w3schools.com/css/css_selectors.asp

W3SCHOOLS, 1999g. *SQL Introduction* [online] [vid. 2024-03-23]. Dostupné z: https://www.w3schools.com/sql/sql_intro.asp

W3SCHOOLS, 1999i. *SQL ORDER BY Keyword* [online] [vid. 2024-03-24]. Dostupné z: https://www.w3schools.com/sql/sql_orderby.asp

W3SCHOOLS, 1999ch. *SQL SELECT DISTINCT Statement* [online] [vid. 2024-03-24]. Dostupné z: https://www.w3schools.com/sql/sql_distinct.asp

W3SCHOOLS, 1999h. *SQL Syntax* [online] [vid. 2024-03-24]. Dostupné z: https://www.w3schools.com/sql/sql_syntax.asp

W3SCHOOLS, 1999d. *Úvod do HTML* [online] [vid. 2024-03-24]. Dostupné z: https://www.w3schools.com/html/html_intro.asp

XAMP, 2023. *XAMPP Installers and Downloads for Apache Friends* [online] [vid. 2024-03-23]. Dostupné z: <https://www.apachefriends.org/>

8 Seznam obrázků

Obr. 1 Visual Studio Code; Zdroj: Vlastní zpracování	5
Obr. 2 NetBeans; Zdroj: Vlastní zpracování.....	6
Obr. 3 PHPMyAdmin; Zdroj: Vlastní zpracování.....	7
Obr. 4 MySQL Workbench; Zdroj: www.mysql.com	8
Obr. 5 Struktura HTML; Zdroj: www.w3schools.com	9
Obr. 6 Seznam diagramů UML; Zdroj: www.itnetwork.cz	17
Obr. 7 Microsoft 365 admin center; Zdroj: Vlastní zpracování.....	20
Obr. 8 TeamViewer Management Console; Zdroj: Vlastní zpracování.....	22
Obr. 9 Avast Bussiness Hub; Zdroj: Vlastní zpracování	23
Obr. 10 AVG Bussiness; Zdroj: Vlastní zpracování.....	24
Obr. 11 Připojení databáze; Zdroj: Vlastní zpracování	27
Obr. 12 Struktura PHP; Zdroj: Vlastní zpracování	29
Obr. 13 Filtry; Zdroj: Vlastní zpracování	30
Obr. 14 Řazení; Zdroj: Vlastní zpracování.....	30
Obr. 15 Zápis do žurnálu; Zdroj: Vlastní zpracování.....	31
Obr. 16 Ukázka HTML kódu; Zdroj: Vlastní zpracování.....	32
Obr. 17 Ukázka kódu responzivity; Zdroj: Vlastní zpracování.....	34
Obr. 18 Úvodní stránka; Zdroj: Vlastní zpracování	35
Obr. 19 Stránka o nás; Zdroj: Vlastní zpracování	36
Obr. 20 Naše služby; Zdroj: Vlastní zpracování	36
Obr. 21 Kontaktní strana; Zdroj: Vlastní zpracování	37
Obr. 22 Přehled; Zdroj: Vlastní zpracování.....	38
Obr. 23 Výpis zařízení; Zdroj: Vlastní zpracování.....	39
Obr. 24 Výpis uživatelů; Zdroj: Vlastní zpracování	39
Obr. 25 Nastavení; Zdroj: Vlastní zpracování	40
Obr. 26 Podrobnosti zařízení; Zdroj: Vlastní zpracování.....	41
Obr. 27 TopButton; Zdroj: Vlastní zpracování.....	42
Obr. 28 Implementace mapy; Zdroj: Vlastní zpracování	43
Obr. 29 Implementace recaptchy; Zdroj: Vlastní zpracování	43
Obr. 30 Diagram případu užití; Zdroj: Vlastní zpracování.....	45

Obr. 31 Diagram tříd; Zdroj: Vlastní zpracování.....	45
Obr. 32 tblFirma; Zdroj: Vlastní zpracování.....	46
Obr. 33 tblNastaveni; Zdroj: Vlastní zpracování.....	47
Obr. 34 tblStredisko; Zdroj: Vlastní zpracování.....	47
Obr. 35 tblTypVlast; Zdroj: Vlastní zpracování.....	48
Obr. 36 tblTypZar; Zdroj: Vlastní zpracování.....	48
Obr. 37 tblUmisteni; Zdroj: Vlastní zpracování.....	49
Obr. 38 tblUzivatel; Zdroj: Vlastní zpracování.....	49
Obr. 39 tblVlastnosti; Zdroj: Vlastní zpracování.....	50
Obr. 40 tblZarizeni; Zdroj: Vlastní zpracování.....	50
Obr. 41 tblZurnal; Zdroj: Vlastní zpracování.....	51
Obr. 42 tblPopisZurnal; Zdroj: Vlastní zpracování.....	51

9 Přílohy

Veškeré soubory projektu jsou dostupné v elektronickém odevzdání bakalářské práce v systému eVSKP.

Projekt je dostupný k nahlédnutí na webových stránkách: [TB servis \(tb-servis.cz\)](https://tb-servis.cz).

Pro přihlášení do klientské části lze využít tyto údaje:

Login: uhk

Heslo: UHK.2024

10 Zadání práce z IS (eVŠKP)



Univerzita Hradec Králové
Fakulta informatiky a managementu

Zadání bakalářské práce

Autor: Ondřej Beran
Studium: I2100180
Studijní program: B1802 Aplikovaná informatika
Studijní obor: Aplikovaná informatika
Název bakalářské práce: **Tvorba webové aplikace pro inventarizaci zařízení**
Název bakalářské práce AJ: Development of web pages for equipment inventory

Cíl, metody, literatura, předpoklady:

Osnova:

1. Úvod
2. Popis problematiky a její řešení
3. Vývojové prostředky webových aplikací
4. Existující aplikace pro inventarizaci
5. Optimalizace webových aplikací
6. Závěrečné shrnutí
7. Doporučení

Cíl práce:

Cílem práce je vytvořit webovou aplikaci pro IT firmu, která usnadní inventarizaci výpočetní techniky.

SEO : optimalizace pro vyhledávače profesionálně / Danny Dover a Erik Dafforn ; [překlad David Gregor]

Vyd. 1.; Brno : Zoner Press, 2012; 978-80-7413-172-1

Web ostrý jako břitva : návrh fungujícího webu pro webdesignery a zadavatele projektů / Jan Řezáč

Vyd. 2.;[Brno] : House of Řezáč, 2016; 978-80-270-0644-1

HTML5 a CSS3 pro webové designéry / Alexis Goldsteinová, Louis Lazaris a Estelle Weylová ; [překlad Jan Pokorný]

Vyd. 1.; Brno : Zoner Press, 2011; 978-80-7413-166-0

PHP & MySQL in easy steps / Mike McGrath

Vyd. 2.; Leamington Spa, Warwickshire : In Easy Steps ; [2018]; 978-1-84078-827-3

PHP 5 a MySQL 5 : průvodce webového programátora / Michael Kofler, Bernd Öggl ; [překlad David Čepička]

Vyd. 1.; Brno : Computer Press, 2007; 978-80-251-1813-9

JavaScript okamžitě / Ara Pehlivanian, Don Nguyen ; [překlad Ondřej Baše]