Střední průmyslová škola a Vyšší odborná škola, Liberec, příspěvková organizace

Návrh a tvorba firemní aplikace pro výkazy

Maturitní práce

Autor **Ivan Kraus**

Obor **Informační technologie**

Vedoucí práce **Ing. Tomáš Kazda, DiS.**

Školní rok **2024/2025**

Počet stran **85**

Počet slov **10352**



Obsah obrázku text, snímek obrazovky, Písmo, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obsah obrázku text, snímek obrazovky, dopis, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Anotace

Tato maturitní práce se zabývá modernizací firemní aplikace Výkazovník, která slouží ke správě výkazů práce ve společnosti SMWORKS. Vychází z analýzy stávajícího systému, identifikace jeho nedostatků a následného návrhu a implementace nové aplikace s důrazem na lepší uživatelský komfort a moderní technologické standardy. Práce přináší komplexní redesign zahrnující přepracovanou databázovou strukturu, pro ni optimalizované API endpointy, nový návrh frontend a jeho implementaci pomocí frameworku React. Vývoj probíhal od návrhu wireframů přes grafický design až po implementaci funkcionalit, jako jsou responsivní design, progresivní webová aplikace (PWA), pokročilé filtrování výkazů, či integrace s Microsoft účtem. Vývoj probíhal od návrhu wireframů přes grafický design až po implementaci funkcionalit, jako jsou responsivní design, progresivní webová aplikace (PWA), pokročilé filtrování výkazů, či integrace s Microsoft účtem. Výsledkem práce je plně funkční aplikace, která byla testována zaměstnanci společnosti a jejich zpětná vazba byla následně zapracována. Projekt splnil nejen své cíle, ale zároveň vytvořil základ pro další rozvoj aplikace.

Summary

This graduation thesis focuses on modernizing the corporate application “Výkazovník,” which is used for managing work logs at the company SMWORKS. It is based on an analysis of the existing system, identification of its shortcomings, and the subsequent design and implementation of a new application with an emphasis on enhanced user comfort and modern technological standards. The thesis presents a comprehensive redesign, which includes overhauling the database structure, creating optimized API endpoints, and designing and implementing a new frontend using the React framework. The development process covered everything from wireframe creation and graphic design to the implementation of features such as responsive design, a progressive web application (PWA), advanced timesheet filtering, and integration with Microsoft accounts. The result is a fully functional application that was tested by company employees, whose feedback was then incorporated. The project not only met its objectives but also laid the foundation for further development of the application.

Čestné prohlášení

Prohlašuji, že jsem předkládanou maturitní práci vypracoval sám a uvedl jsem veškerou použitou literaturu a bibliografické citace.

V Liberci dne

Ivan Kraus

Obsah

[Úvod 1](#_Toc192756915)

[1 Analýza stávající aplikace 2](#_Toc192756916)

[1.1 Pozadí aplikace Výkazovník – aneb kde se berou data? 3](#_Toc192756917)

[1.2 Zhodnocení stavu stávající aplikace 4](#_Toc192756918)

[1.3 Responzivita webové aplikace 4](#_Toc192756919)

[1.4 Nevyhovující UI a UX aplikace 6](#_Toc192756920)

[1.4.1 Filtrace obsahu 10](#_Toc192756921)

[1.4.2 Sledování vykázaného času a další nedostatky aplikace 10](#_Toc192756922)

[1.5 Katalog požadavků 10](#_Toc192756923)

[2 Návrh nové aplikace 14](#_Toc192756924)

[2.1 Databáze 14](#_Toc192756925)

[2.1.1 Diagram nové databáze 14](#_Toc192756926)

[2.1.2 Výsledek optimalizace a normalizace databázové struktury 15](#_Toc192756927)

[2.2 Struktura uživatelů a jejich oprávnění 15](#_Toc192756928)

[2.3 Návrh nových API endpointů 16](#_Toc192756929)

[2.4 Sitemapy 18](#_Toc192756930)

[2.5 Grafický návrh 20](#_Toc192756931)

[2.5.1 Figma a komponenty 20](#_Toc192756932)

[2.5.2 Vizuální styl 21](#_Toc192756933)

[2.5.3 Wireframy 21](#_Toc192756934)

[2.5.4 Finální grafický design 28](#_Toc192756935)

[3 Vývoj a implementace nové aplikace 47](#_Toc192756936)

[3.1 Zvolené technologie a struktura aplikace 47](#_Toc192756937)

[3.1.1 Backend 47](#_Toc192756938)

[3.1.2 Frontend 47](#_Toc192756939)

[3.2 Spojení frontend–backend a databáze 48](#_Toc192756940)

[3.3 React knihovny a vlastní komponenty 51](#_Toc192756941)

[3.4 Routování 56](#_Toc192756942)

[3.4.1 Implementace 57](#_Toc192756943)

[3.4.2 Chráněné stránky pomocí ProtectedRoute 57](#_Toc192756944)

[3.5 Jednotlivé stránky aplikace 60](#_Toc192756945)

[3.6 Progresivní webová aplikace 62](#_Toc192756946)

[3.6.1 Implementace PWA 63](#_Toc192756947)

[4 Nasazení, testování a zpětná vazba 66](#_Toc192756948)

[4.1 Nasazení pro testování aplikace 66](#_Toc192756949)

[4.2 Testování aplikace 66](#_Toc192756950)

[4.3 Zpětná vazby a její implementace 66](#_Toc192756951)

[4.4 Možná budoucí vylepšení 66](#_Toc192756952)

[4.5 Nasazení na produkci 67](#_Toc192756953)

[Závěr 68](#_Toc192756954)

[Seznam zkratek a odborných výrazů 69](#_Toc192756955)

[Seznam obrázků 73](#_Toc192756956)

[Použité zdroje 75](#_Toc192756957)

[A. Seznam přiložených souborů I](#_Toc192756958)

# Úvod

**Představení tématu**

Vykazování práce představuje klíčový prvek řízení firem, které pracují současně na více projektech. Společnosti poskytující služby zákazníkům vyžadují efektivní nástroje pro evidenci času stráveného na jednotlivých úkolech, analýzu efektivity práce a získávání podkladů pro obchodní a finanční rozhodování. (1) (2) (3)

Tato práce se zaměřuje na modernizaci firemní aplikace Výkazovník, která již několik let slouží ke správě výkazů práce ve společnosti SMWORKS. Společnost SMWORKS vyvíjí a provozuje vlastní interní řešení Výkazovník na adrese [https://vykazovnik.cz](https://vykazovnik.cz/app). Vlastní vývoj aplikace umožňuje firmě plnou kontrolu nad daty, minimalizuje závislost na třetích stranách a eliminuje potenciální problémy s licenčními podmínkami externích řešení.

**Důvod výběru tématu**

Autor maturitní práce navázal spolupráci se společností SMWORKS již ve třetím ročníku v rámci firemních praxí. Na konci této praxe byla s vedoucím praxe, Ondřejem Vackem, dohodnuta možnost realizace dlouhodobé maturitní práce ve čtvrtém ročníku.

Zadáním této práce je redesign a modernizace firemní aplikace Výkazovník, která je dlouhodobě vyvíjena a přizpůsobována potřebám společnosti SMWORKS. Aktuálně se jedná již o třetí interní aplikaci používanou pro správu výkazů. Přestože aplikace nadále plní svůj primární účel – vykazování práce – její technologické zpracování a uživatelský komfort již neodpovídají moderním standardům webových aplikací. Vedení firmy proto identifikovalo potřebu redesignu a modernizace aplikace, což bylo hlavním impulzem pro výběr tohoto tématu maturitní práce.

**Cíle práce**

Práce si kladla za cíl analyzovat současný stav aplikace Výkazovník, ať už z pohledu autora práce jako uživatele aplikace, tak z pohledu ostatních běžných uživatelů, ale i z pohledu manažerů, vedoucích rolí a administrátorů aplikace ve firmě, kteří aplikaci aktivně využívají. Na základě zpětné vazby je nutné navrhnout a vytvořit novou aplikaci, která bude odrážet moderní požadavky webových aplikací, ať už co se týče zvolených technologických postupů, které budou umožňovat aplikaci vyvíjet i do budoucna, ale i požadavků na uživatelský komfort, které zefektivní firemní procesy při vykazování práce.

# Analýza stávající aplikace

Je nutné si ujasnit, co znamená termín „**výkaz**“. V aplikaci Výkazovník je termín „**výkaz**“ používán pro položku tvořenou uživatelem obsahující informace o práci, kterou uživatel vykonal. Výkaz obsahuje následující informace:

* **Úkol** – Klienti digitální agentury SMWORKS mají rozjednané s SMWORKS obchodní případy, což jsou vlastně úkoly, na kterých poté zaměstnanci, či týmy pracují (např. „Firma XYZ – Nový web“).
* **Popis úkolu** – Textové shrnutí v několika slovech, co uživatel udělal (např. „Grafický návrh nového webu ve Figmě“.)
* **Role** – Pracovní role, pod kterou výkaz spadá (např. „Programátor, grafik…“).
* **Datum** – Datum, kdy byla práce vykonána.
* **Doba trvání** (v hodinách) – Určuje délku práce (např. „1.25 hodiny“).

Existuje speciální případ, kdy uživatel, který výkaz vytváří, buď nezná úkol, ke kterému má práci zařadit, či ještě daný úkol v systému neexistuje. V tomto případě uživatel zaškrtne checkbox neznám úkol a místo úkolu zadává uživatel klienta, ke kterému výkaz přiřadí. Poté automaticky přijde jeho nadřízenému email s tím, že byl vytvořen „**neznámý výkaz**“ a ten tento výkaz opraví tím, že zpětně přiřadí výkazu správný úkol.

Pro plné pochopení firemních procesů při práci se stávající aplikací bylo nezbytné získat administrátorská práva a detailně nastudovat kompletní logiku aplikace od vytváření výkazů a následnou práci s nimi. Pro sběr zpětné vazby, identifikaci konkrétních problémů, nedostatků a návrhů na zlepšení nové aplikace byly uskutečněnyosobníschůzky s vedoucími pracovníky, během nichž bylo analyzováno jejich dosavadní využívání aplikace a pracovní postupy s ní spojené.

Na základě těchto schůzek byly shromážděny podněty na zlepšení a připomínky k tvorbě nové aplikace z pohledu osob, které řídí týmy zaměstnanců, schvalují výkazy nebo kontrolují jejich správnost. Pro zbytek zaměstnanců, tedy běžné uživatele aplikace, byl vytvořen a rozeslán jednoduchý dotazník, který jim poskytl prostor pro zhodnocení stávající funkcionality a umožnil identifikovat nedostatky a slabé stránky stávající aplikace, které by byly podklady pro vývoj nové aplikace.

Obsah obrázku text, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Jednoduchý dotazník pro zaměstnance firmy k získání zpětné vazby

Veškerá získaná zpětná vazba byla následně prodiskutována a vyhodnocena s **CTO** firmy **Ondřejem Vackem**. Některé nápady nebyly vhodné pro realizaci, některé nápady byly vhodné pro zapracované, ale ne v této fázi aplikace (odkláněly se od původních cílů aplikace – redesignu a modernizace – a zaměřovaly se více na analytickou část – tvorba grafů z dat apod.), byly však uchovány pro možnou budoucí implementaci – těmto dalším možným vylepšením aplikace je věnována pozdější kapitola 4.4 Možná budoucí vylepšení.

## Pozadí aplikace Výkazovník – aneb kde se berou data?

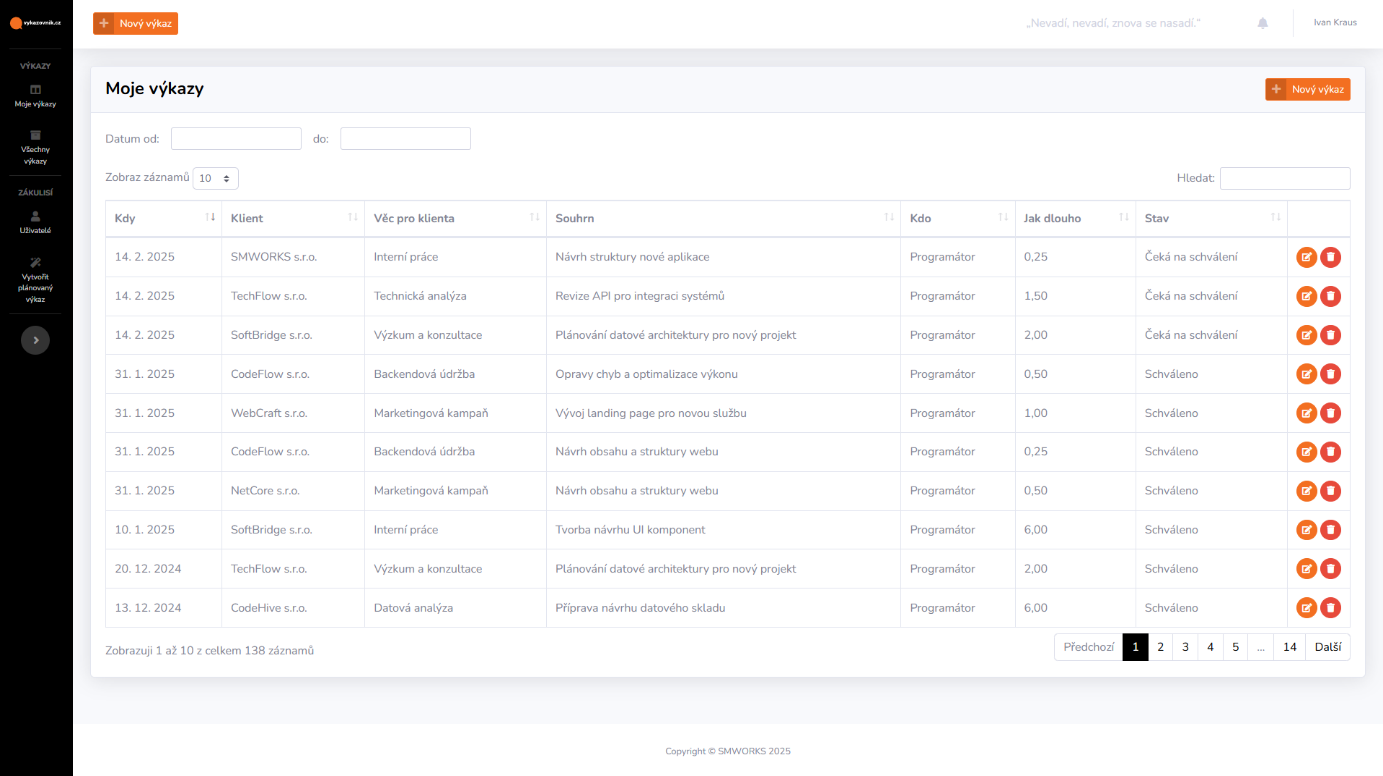
Pro návrh nové aplikace je zásadní pochopit, odkud pocházejí data a jak funguje současná infrastruktura Výkazovníku. Firma SMWORKS využívá **CRM** (Customer Relationship Management) systém **Raynet**, který slouží k evidenci klientů, obchodních případů, cen, stavů a dalších obchodních informací. Pro účetnictví poté firma používá software **Pohoda**. (4)

Aby byla zajištěna automatizovaná výměna dat mezi těmito systémy, vyvinula firma interní integrační software s názvem **P.R.D.E.L.** (Pohoda-Raynet Data Exchange Layer). Tento systém předává obchodní informace z Raynetu do ekonomického programu Pohoda pro fakturaci a následně zpětně aktualizuje informace o zaplacení těchto faktur v CRM systému.

Vzhledem k tomu, že P.R.D.E.L. obsahuje data o všech obchodních případech (úkolech – tasks) a klientech, slouží zároveň jako datový zdroj pro Výkazovník. Jeho **API** poskytuje relevantní informace, které aplikace Výkazovník využívá k práci s klienty a úkoly, což umožňuje jednotnou správu a synchronizaci dat napříč celým ekosystémem firmy.

## Zhodnocení stavu stávající aplikace

Z analýzy a nashromážděné zpětné vazby jsou zde uvedeny **klíčové nedostatky** stávající aplikace Výkazovník, na které byl brán potaz při tvorbě nové aplikace.



Obrázek – Přehled výkazů uživatele ze stávající aplikace

## Responzivita webové aplikace

Jedním z nejvýraznějších nedostatků stávající aplikace je její pouze **částečná responzivita**. Responzivní design je klíčový princip návrhu webových aplikací, který **umožňuje automatické přizpůsobení rozložení a obsahu různým velikostem obrazovek**, zejména mobilním zařízením a tabletům. V současné aplikaci dochází k problémům s přizpůsobením obsahu menším displejům, což je například vidět na stránce s uživatelskými výkazy, kde tabulky přesahují šířku obrazovky a vyžadují nepohodlné horizontální posouvání. Tento nedostatek je například viditelný na níže uvedených screenshotech a týká se celé aplikace. (5)

Aplikace sice využívá populární CSS framework Bootstrap, který poskytuje předpřipravené komponenty a grid systém pro responzivní rozložení, avšak ne všechny prvky aplikace byly při předchozím vývoji optimalizovány pro mobilní zařízení. (6)

Nedostatečná responzivita ztěžuje uživatelům práci zejména při procházení a vyhledávání dat na mobilních zařízeních. Typickým příkladem tohoto problému je každodenní ranní schůzka „standup“, při které zaměstnanci firmy shrnují svou pracovní náplň z předchozího dne a plánují úkoly na den aktuální. Kvůli „vytékajícímu obsahu“ stránek je uživatel nucen horizontálně scrollovat a je těžké a neefektivní dohledávat výkazy z minulého dne. Tento problém se týká celkové práce s Výkazovníkem na mobilních zařízeních.

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Demostrance pouze částečné reponzivity webové aplikace



Obrázek – Demostrance pouze částečné reponzivity tabulky webové aplikace

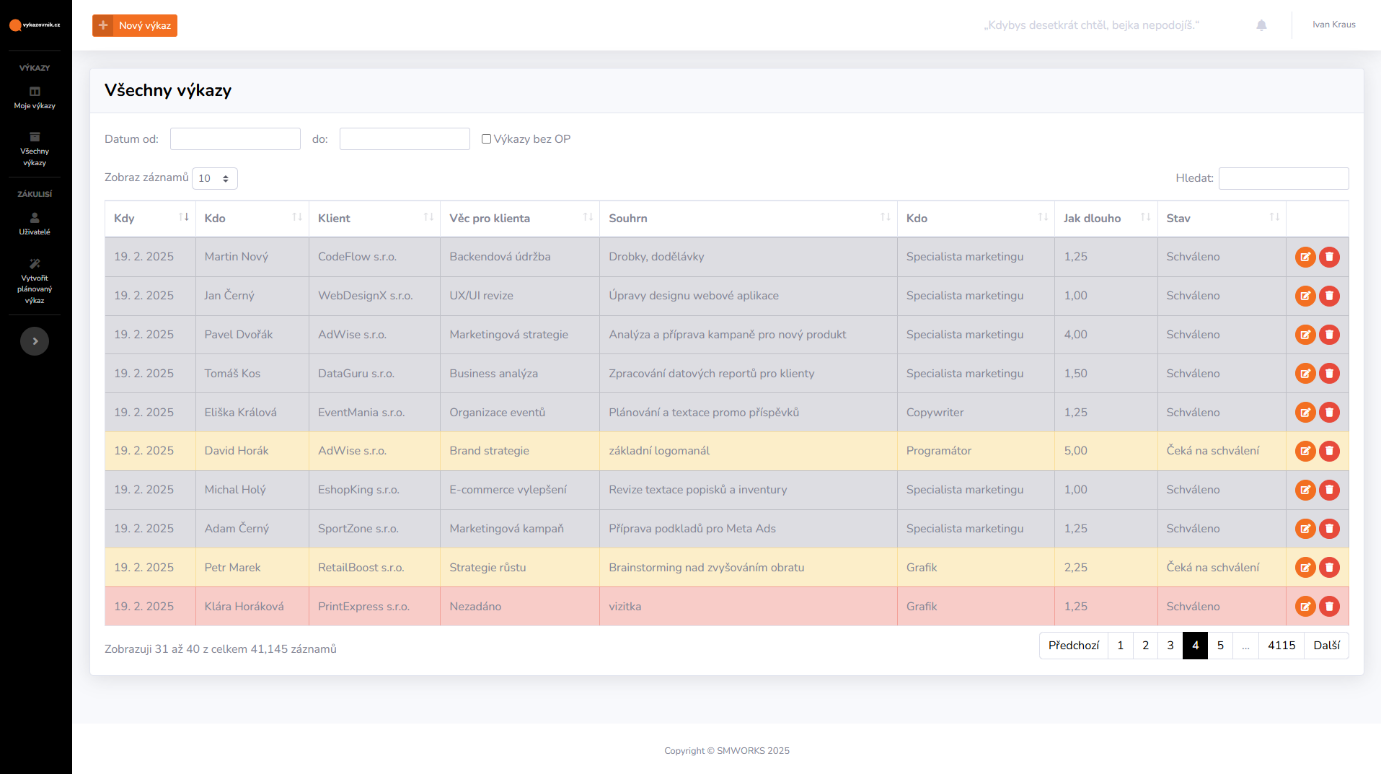
## Nevyhovující UI a UX aplikace

Stávající aplikaci sužuje nejen částečná responzivita, ale také **zastaralý UI a UX**, které neodpovídají moderním standardům webových aplikací.

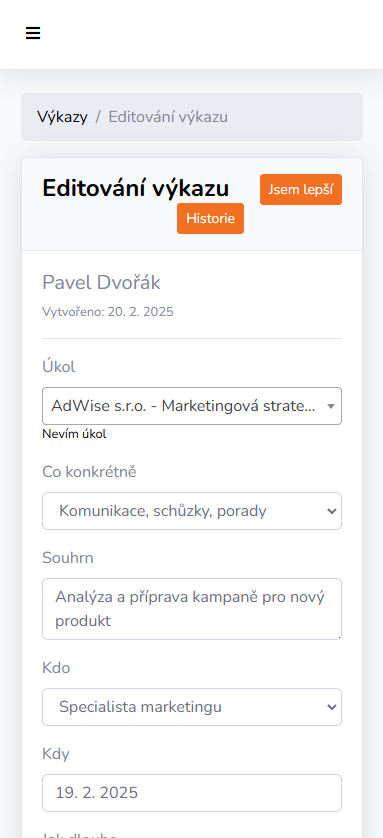
UI (User Interface) neboli uživatelské rozhraní představuje vizuální podobu aplikace, tedy její vzhled, rozložení prvků, barevnou paletu, typografii a interaktivní prvky jako tlačítka, formuláře či navigace. Dobře navržené UI usnadňuje uživateli orientaci v aplikaci a zajišťuje intuitivní ovládání. (6)

UX (User Experience) neboli uživatelský prožitek se zaměřuje na celkový pocit a efektivitu při používání aplikace. Zahrnuje například logiku uspořádání funkcí, rychlost vykonávání operací, přehlednost obsahu či snadnost dokončení určitého úkolu v aplikaci. Cílem UX designu je, aby uživatel mohl aplikaci používat efektivně. (7)

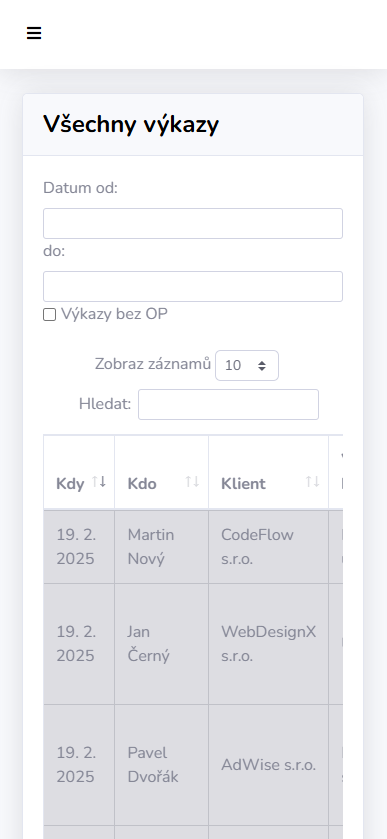
Stávající aplikace nevhodně pracuje s dostupným prostorem – v některých sekcích je obsah nevyváženě zarovnaný, zatímco jinde přetéká mimo své mantinely. Tabulky, které tvoří klíčovou součást aplikace jsou nepřehledné kvůli absenci vizuální hierarchie. Aplikace je celkově nekonzistentní ve vzhledu jednotlivých prvků jako jsou tlačítka a další komponenty. Používá místo ikon, či vizuálních indikátorů repetitivní textové popisy nebo je zvýrazněný obsah díky špatnému kontrastu popisu pole a podkladu špatně čitelný.



Obrázek – Sekce „Všechny výkazy“ se špatně čitelným textem



Obrázek – Sekce „Editování výkazu“ na mobilním zařízení



Obrázek – Sekce „Všechny výkazy“ na mobilním zařízení

### Filtrace obsahu

Na základě analýzy používání aplikace a zpětné vazby uživatelů bylo zjištěno, že aplikace postrádá několik klíčových funkcionalit, které negativně ovlivňují efektivitu práce. Jedním z hlavních problémů je **omezená možnost filtrování obsahu** v tabulkách. Aplikace aktuálně umožňuje pouze základní řazení sloupců, filtrování podle data a textové vyhledávání. Tento přístup je pro uživatele občas nedostačující, protože chybí možnost detailnějšího filtrování podle dalších relevantních kritérií, jako jsou konkrétní uživatelé, klienti, úkoly nebo stav schválení výkazu.

### Sledování vykázaného času a další nedostatky aplikace

Další nedostatky se týkají rovněž funkce **sledování vykázaných hodin**. Pro uživatele je velmi klíčové mít přehled o tom, kolik hodin průměrně či celkově vykázali za různá období. Stávající aplikace poskytuje tuto informaci pouze omezeně – zobrazí časové údaje pouze jednotlivým uživatelům, a to pouze na základě vyfiltrování konkrétního data. Manažerským rolím a administrátorům aplikace také chybí komplexní možnost sledovat hodiny zaměstnanců a analyzovat vykázaný čas pomocí filtrování v širším kontextu parametrů. Aplikace zahrnuje další problémy jako

* **Repetitivní schvalování výkazů.**
* **Neefektivní kontrola dní**, kdy si uživatel zapomněl vykázat.
* Obsahuje **funkční chyby** jako například automatické schválení výkazu po jeho editaci bez vědomí uživatele, který výkaz editoval a další problémy.

Nedostatky aplikace vedou ke snížení efektivity práce s aplikací, ztěžují uživatelům dohled nad vykázanými hodinami a celkově snižují uživatelskou přívětivost při jejím používáním. Modernizace aplikace, která proběhla v rámci této maturitní práce, zahrnuje nejen vizuální a technologické vylepšení, ale také rozšíření funkcionalit odpovídající reálným potřebám uživatelů.

## Katalog požadavků

Na základě analýzy stavu stávající aplikace a zpětné vazby byly stanoveny **požadavky pro novou aplikaci**, která má opravit nedostatky stávající aplikace a rozšířit ji o funkcionality odpovídající reálným potřebám uživatelů. Níže jsou definované jednotlivé požadavky:

* **Responzivita** – aplikace má být navržená jak pro desktop, tak i pro mobilní zařízení, na kterých bude možné efektivně pracovat s aplikací.
* **PWA** – Jednou z klíčových modernizací aplikace je její zpřístupnění jako progresivní webová aplikace (PWA). Jedná se o hybridní přístup ke kombinaci webových a mobilních aplikací. Na rozdíl od běžných webových aplikací umožňuje fungovat jako plnohodnotná aplikace, kterou lze instalovat přímo na mobilní zařízení či počítače. Mezi hlavní výhody použití PWA technologie v novém Výkazovníku je cachování statických souborů (**HTML**, CSS, **JavaScript**), dík čemuž se bude načítat aplikace rychleji než běžné webové aplikace. Aplikace bude mít offline režim, ve kterém upozorní uživatele na fakt, že nemá přístup k internetu a aplikace tak na rozdíl od běžného webu bude omezeně fungovat. Na rozdíl od nativních aplikací pro **Android** nebo **iOS** není nutná distribuce přes App Store nebo Google Play – aplikace tedy bude dostupná všem na všech zařízeních s moderním webovým prohlížečem, ze kterého si ji budou moct instalovat. PWA aplikace umožňuje integrovat do budoucna nové funkcionality jako push notifikace, čehož by se dalo v budoucnu využít například pro připomínky vykazování práce. (7)
* **Zlepšení UI a UX** – Jedním z cílů nové aplikace je přinést ucelený moderní design s důrazem na lepší přehlednost a intuitivní ovládání.
* **Autentifikace pomocí Microsoft účtu** – Jedním ze standardů moderních aplikací je možnost přihlášení se prostřednictvím externích poskytovatelů identity jako je například **Microsoft Account** (**MSA**). Firma SMWORKS právě využívá Microsoft office aplikace, zaměstnanci mají své pracovní účty pod vlastní firemní doménou **@smworks.cz**. Po přihlášení se do účtu pomocí klasického email a hesla budou moct zaměstnanci v nastavení připojit svůj pracovní Microsoft účet. Tato funkcionalita velmi usnadní a urychlí repetitivní přihlašování se bez nutnosti zadání emailu a hesla – zlepší tak uživatelský komfort. (8)
* **Návrh nové databáze** – v rámci modernizace aplikace se nabízí i optimalizace a normalizace stávající databáze. Databáze slouží jako centrální úložiště informací, ve kterém se uchovávají klíčová data, například uživatelské účty, výkazy, úkoly nebo oprávnění. Důkladně navržená databázová struktura usnadňuje správu těchto dat, umožňuje rychlejší dotazy a zajišťuje efektivnější integraci s dalšími částmi aplikace. Zlepší se konzistence dat, sníží se redundance a celkově se zvýší flexibilita v případě budoucího škálování aplikace. (9)
* **Návrh nových API endpointů** – S optimalizací databáze úzce souvisí návrh nových API endpointů. API (Application Programming Interface) představuje rozhraní mezi frontendem (uživatelským rozhraním) a backendem (serverovou částí aplikace), které umožňuje bezpečnou a efektivní výměnu dat. Každý API endpoint je konkrétní adresa, na kterou může frontend posílat požadavky, například pro načtení seznamu výkazů, vytvoření nového uživatele nebo úpravu existujících dat. Nově navržené API endpointy budou reflektovat rozšířenou funkcionalitu aplikace, jako je právě například pokročilé filtrování dat. (10) (11)
* **Pokročilé filtrování** – Uživatelé budou schopni v aplikaci filtrovat data pomocí parametrů jako jsou například konkrétní uživatelé, klienti, úkoly, stav schválení výkazů, textové vyhledávání nebo specifické období.
* **Export dat** – Současně s filtry bude možné exportovat vyfiltrované výkazy do formátu Excel, přičemž exportovaná data budou odpovídat aplikovaným filtrům. To umožní manažerům hlubší práci s daty.
* **Hromadné schvalování výkazů** – V aplikaci funguje proces schvalování výkazů, kdy nadřízení kontrolují svým podřízeným jejich výkazy a po kontrole je schvalují. Tento proces byl ve staré aplikaci repetitivní – každý výkaz se musel schválit samostatně. Nová aplikace umožní navíc výkazy jednotlivě i hromadně označovat a poté je po kontrole hromadně schválit.
* **Předvyplněné odkazy** – Nová aplikace umožní uživatelům generovat odkazy, které uživatelům automaticky předvyplní nový výkaz s předem zadanými údaji v **URL** linku. V praxi to znamená, že manažeři a administrátoři budou moct do plánovacích aplikací typu Planner vkládat k úkolům odkazy na již správně předvyplněný výkaz obsahující správný úkol a klienta. Tato funkcionalita zvýší efektivitu a uživatelský komfort, jelikož uživatelé nebudou muset řešit, zda vykazují ke správnému úkolu od klienta, tudíž ani manažeři poté nebudou mít tolik práce při opravách špatně vyplněných výkazů.
* **Přehledné statistiky o vykázaném čase** – Aplikace bude staticky ukazovat souhrnné informace o průměru a součtu vykázaných hodin jak jednotlivým uživatelům v sekci „Moje výkazy“, stejně tak manažerské role a administrátoři aplikace budou moct kontrolovat vykazování svých podřízených, či analyzovat čas strávený pro určité klienty a jejich úkoly v sekci „Všechny výkazy“. Tyto časové hodnoty budou tedy zároveň odrážet aplikované filtry.
* **Standup page** – Pro rychlé a efektivní přehledy, které budou moct číst uživatelé na ranních standupech ve svém mobilním zařízení, vznikne zcela nová stránka standup.
* **Toggle timer** – Pro měření si času stráveného při plnění úkolů budou moct uživatelé využít zcela novou stránku Toggle. Na stránce si budou moct zvolit z aktivních úkolů určitého klienta a po dokončení práce budou moct potvrzením přejít na již předvyplněný výkaz s úkolem a zaokrouhleným časem právě z togglu.
* **Automatické zaokrouhlování času** – Ve staré aplikaci nebylo ošetřené zaokrouhlování vykázaného času na čtvrthodiny, mohlo tedy nastat, že uživatel zadal například 0.33 hodin. V nové aplikaci se po zadání délky výkazu automaticky délka zaokrouhlí na nejbližší čtvrthodinu.
* **Duplikace výkazů** – při dlouhodobých projektech dochází občas k repetitivnímu vykazování stejné práce pod stejný úkol, proto budou moct uživatelé své výkazy duplikovat, kdy se pouze automaticky navolí aktuální den a uživatel případně upraví jen čas strávený nad úkolem.
* **Trvalé zobrazení všech úkolů** – Ve staré aplikaci si mohli manažeři a administrátoři zobrazovat nejen aktuálně aktivní úkoly, ale i historické úkoly, které již nejsou aktuální. Slouží to například ke zpětnému vykazování k historickým úkolům. Tato funkcionalita musela být explicitně pokaždé povolena, když chtěl manažer nebo administrátor vidět již neaktivní úkol. Nyní tato funkce bude moct být globálně trvale povolena v nastavení profilu uživatele aplikace.
* **Nápověda časových údajů** – Při vyplňování času ve výkazu dochází neustále k vyplňování podobných hodnot, bude tedy možnost navíc zakliknout pouze jednu z přednastavených hodnot a nebude potřeba psát pokaždé hodnotu manuálně.
* **Přehlednost výkazů dle dní** – Pro přehledné vizuální oddělení jednotlivých výkazů v tabulkách budou seskupeny podle dní a mezi skupinami budou mezery, což bude sloužit k lepší orientaci v záznamech.

# Návrh nové aplikace

## Databáze

Před návrhem frontend nové aplikace byla zanalyzována původní databázová struktura, při níž byly zjištěny problémy s redundancí dat, nedostatečnou normalizací a integritou a celkovou škálovatelností do budoucna. Cílem bylo tedy toto zlepšit a vznikl tak návrhový diagram nové databázové struktury, dle kterého byla kolegy ve firmě vytvořena a nasazena nová **MySQL** databáze.

MySQL je relační databázový systém, který efektivně zpracovává data pomocí strukturovaných tabulek propojených relačními vazbami. Využívá jazyk **SQL** pro manipulaci s daty a jejich dotazování. Tato databáze byla zvolena především pro kompatibilitu s již existující infrastrukturou společnosti. (12)

### Diagram nové databáze

Pro jednoduchost je v této práci vložen diagram návrhu nové databáze. Diagram databáze představuje strukturu databáze, kde jednotlivé bloky znázorňují tabulky databáze a jejich atributy. Každý řádek v tabulce odpovídá konkrétnímu sloupci v databázi s různými datovými typy. Mezi tabulkami jsou definovány vazby, jako jsou primární klíče a cizí klíče, které, které určují vztahy a závislosti mezi daty a zajišťují integritu databázové struktury. Podrobný popis a vysvětlení jednotlivých tabulek lze najít v souboru „Dokumentace aplikace Výkazovník“ v přiložených souborech maturitní práce. (13)

Obsah obrázku text, snímek obrazovky, diagram, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek –Diagram návrhu nové databáze

### Výsledek optimalizace a normalizace databázové struktury

V nové databázi došlo k optimalizaci a normalizaci, což vedlo ke zlepšení integrity dat, snížení redundance a lepší škálovatelnosti:

* **Eliminace redundance** – Ve staré databázi docházelo k duplicitnímu ukládání určitých dat. To bylo odstraněno rozdělením dat do specializovaných tabulek, jako jsou TaskType, JobTitle a Role. Díky tomu už není nutné například ukládat jméno pracovní pozice u každého uživatele nebo typ úkolu u každého reportu –místo toho se používají cizí klíče.
* **Normalizace na třetí normální formu (3NF)** – Všechny atributy v tabulkách nyní závisí pouze na primárním klíči, čímž se eliminovaly tranzitivní závislosti. To znamená, že například pracovní pozice uživatele (JobTitle) není uložena přímo v tabulce User, ale odkazuje se na samostatnou tabulku JobTitle, kde jsou tyto informace centralizovány. (9)
* **Referenční integrita** – Použití cizích klíčů mezi tabulkami zaručuje, že nelze například vytvořit výkaz (Report) s neexistujícím uživatelem nebo úkolem. To brání inkonzistencím a zajišťuje správnou vazbu mezi entitami.
* **Flexibilita** – Struktura databáze nyní umožňuje snadnější rozšíření. Například při přidání nových typů úkolů (TaskType) nebo pracovních pozic (JobTitle) není nutné měnit hlavní tabulky (User, Task, Report), protože stačí přidat nový záznam do odpovídající tabulky.
* **Výkon** – Optimalizovaná struktura znamená rychlejší dotazy na databázi. Díky správným indexům a normalizované struktuře se minimalizuje objem dat načítaných při každém dotazu. Například filtrování výkazů (Report) podle typu úkolu (TaskType) je nyní efektivnější, protože typ úkolu není opakovaně uložen u každého záznamu, ale je uložen pouze jednou v samostatné tabulce.

## Struktura uživatelů a jejich oprávnění

V aplikaci existují nové tři úrovně uživatelských rolí s odlišnými oprávněními: administrátor (admin), manažer a běžný uživatel. Každá role má specifické pravomoci, které určují, jaké akce může uživatel provádět v systému. Podle uživatelské role byl přizpůsoben i vzhled a možnosti v aplikaci.

**Uživatel (User)**

Uživatelé představují základní úroveň přístupu v aplikaci. Mohou pracovat pouze se svými vlastními výkazy a nemají práva spravovat jiné uživatele. Tato uživatelské role dále:

* **Vidí, vytváří, upravuje a maže pouze své vlastní výkazy.**
* **Nemůže schvalovat výkazy.**
* **Vidí pouze aktivní úkoly.**
* **Nemá přístup ke správě jiných uživatelů.**

**Manažer (Supervisor)**

Manažer má rozšířená oprávnění oproti běžnému uživateli. Kromě vlastních výkazů má přístup i k výkazům svých podřízených, které může spravovat. Stejně tak může spravovat své vlastní podřízené. Ve své roli:

* Vidí veškeré úkoly.
* Má přidělené podřízené uživatele (ti ho mají nastaveného v systému jako svého nadřízeného). Nemá přístup k uživatelům a jejich výkazům, kteří nejsou jeho podřízení.
* Vidí výkazy svých podřízených a může je upravovat, schvalovat a mazat.
* Je schopen upravovat uživatelské účty svých podřízených nebo je mazat.

**Administrátor (Admin)**

Administrátor má nejvyšší úroveň oprávnění v aplikaci a může provádět jakékoli změny. Především tak:

* Vidí a spravuje všechny uživatele.
* Může upravovat, mazat a schvalovat jakékoli výkazy.

## Návrh nových API endpointů

V rámci vývoje nové aplikace bylo potřeba navrhnout nové API endpointy, které umožní komunikaci mezi frontendem a databází. Tyto endpointy byly navrženy tak, aby efektivně pokryly všechny požadavky nové aplikace.

API bylo navrženo s ohledem na **RESTful** architekturu, která zajišťuje efektivní a škálovatelnou komunikaci mezi frontendem a backendem. V rámci návrhu byla autorem práce definována struktura API, endpointy a pravidla pro autentifikaci a oprávnění uživatelů. API bylo rozděleno do jednotlivých kontrolerů (např. UserController, ReportController, TaskController atd.), přičemž každý kontroler pokrývá konkrétní část aplikace respektující **HTTP** metody (GET, POST, PUT, DELETE), což zajišťuje konzistentní a snadno rozšířitelné rozhraní. (14)

Bezpečnost API je zajištěna pomocí autentizace na základě **JWT** tokenů, kde mají uživatelé již výše zmíněné role (admin, supervisor, user) a přístup k datům je řízen na základě jejich oprávnění. API podporuje pokročilé filtrování a stránkování, díky čemuž je aplikace výkonná a efektivně pracuje s velkým množstvím dat, což je u aplikace jako Výkazovník nezbytné. (15)

Implementace API byla provedena kolegy ve firmě pomocí **PHP** frameworku **Laravel**, který poskytuje robustní nástroje pro správu databáze, autentizaci a bezpečnou komunikaci

Obsah obrázku text, dokument, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka návrhu API – ReportController (1. část)

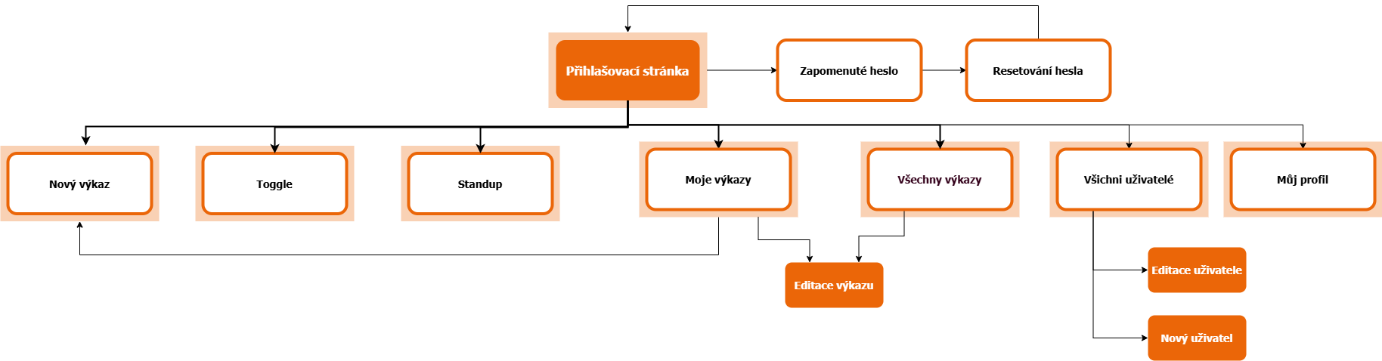
Obsah obrázku text, dokument, Písmo, snímek obrazovky

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka návrhu API – ReportController (2. část)

## Sitemapy

Pro znázornění struktury stránek aplikace byly vytvořeny sitemapy, což je vizuální znázornění struktury aplikace nebo webové stránky. Pomáhá pochopit hierarchii jednotlivých stránek, jejich vzájemné propojení a navigační logiku. Nová aplikace vesměs recykluje strukturu stránek předchozí aplikace kromě přidání nových dvou stránek „Standup“ a „Toggle“. (16)



Obrázek – Ukázka sitemap pro desktopové zařízení

V této sekci jsou uvedeny dvě varianty rozložení a obsahu – jedna pro desktopovou verzi, která nabízí více prostoru pro navigační prvky – obsahuje v navigaci přímý link na stránky:

* Nový výkaz
* Toggle
* Standup
* Moje výkazy
* Všechny výkazy
* Všichni uživatelé
* Můj profil

Obsah obrázku text, snímek obrazovky, diagram, řada/pruh

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka sitemap pro mobilní zařízení

A druhá verze pro mobilní zařízení, kde bývá méně prostoru pro obsah v navigaci, a proto bylo rozhodnuto, že v přímých odkazech v navigaci pro mobilní zařízení budou stránky:

* Nový výkazy
* Toggle
* Standup
* Stránka „Více“

Stránka „Více“ slouží výhradně pro mobilní zařízení jako takový „rozcestník“ pro další stránky jako:

* Můj profil
* Moje výkazy
* Všechny výkazy
* Všichni uživatelé

V aplikaci jsou zohledněna práva přihlášeného uživatele. Běžný uživatel proto neuvidí v navigaci stránky položku „Všechny výkazy“ a „Všichni uživatelé“. Manažeři, jak už bylo zmíněno, mají přístup pouze ke svému týmu, proto se pro ně stránka „Všechny výkazy“ jmenuje „Výkazy týmu“ a stránka „Všichni uživatelé“ nese název „Můj tým“. Administrátor vidí veškeré stránky včetně „Všechny výkazy“ a „Všichni uživatelé“.

## Grafický návrh

Grafický návrh je klíčovou součástí procesu vývoje aplikace, jelikož předem definuje vizuální podobu a uspořádání jednotlivých prvků rozhraní. Díky pečlivě připravenému návrhu lze předejít mnoha nejasnostem při samotné implementaci a výrazně tak usnadnit vývoj. Předem definované rozložení obsahu, typografie, barvy a interakční prvky umožňují efektivní vývoj bez nutnosti opakovaných úprav a zásadních změn v průběhu práce.

### Figma a komponenty

Pro návrh grafického rozhraní frontend aplikace byl zvolen moderní cloudový designový nástroj Figma, který poskytuje optimální prostředí pro tvorbu UI. Tento nástroj je ve firmě již dlouhodobě využíván. Figma podporuje spolupráci na projektech v reálném čase, což umožňuje konzultaci návrhu nové aplikace s kolegy „za chodu“. Figma také podporuje tvorbu responzivních rozhraní, využívá systém komponent a stylů, čímž se minimalizují redundantní operace a zjednodušuje se údržba návrhu. (17)

V rámci návrhu grafického rozraní bylo rozhodnuto nevyužít žádné předpřipravené CSS frameworky, jako je například Bootstrap. Hlavním cílem bylo vytvořit unikátní plně přizpůsobitelný design, který nebude omezen předdefinovanými komponentami nebo styly. Použití Figmy umožnilo navrhnout vizuální podobu aplikace od nuly, což přináší větší flexibilitu v designu a vyšší míru kontroly nad výsledným vzhledem.

Dalším důvodem pro tento přístup je výkon a efektivita aplikace. CSS frameworky často obsahují velké množství tříd a stylů, které aplikace nemusí využít, což může vést ke zbytečnému zatížení aplikace. Vytvořením vlastních stylů umožní lehčí a optimalizovaný kód.

Digitální agentura SMWORKS nemá žádnou interní knihovnu komponent a stylů, ze které bylo možné vycházet při návrhu této aplikace. Z tohoto důvodu se při novém grafickém návrhu vychází z původního stylu aplikace, kdy je optimální zachovat základní rozvržení obsahu, aby adopce uživateli byla přirozená, ale zároveň byly odebrány nedostatky aplikace a implementovány nové funkcionality.

### Vizuální styl

Rovněž bylo nutné stanovit základní vizuální styl aplikace. Pro typografii byl zvolen font **Inter**, který je moderní, dobře čitelný a je běžně využívaný v digitálních aplikacích. Barevná paleta byla zvolena s ohledem na předchozí aplikaci a vizuální prezentaci společnosti SMWORKS, přičemž klade důraz na kontrast a čitelnost prvků. (18)

Pro sadu ikon byla zvolena **FontAwesome Classic Solid**, jenž poskytuje širokou škálu vizuálně sjednocených ikon vhodných pro různé akce v aplikaci. Celkový design aplikace se zaměřuje na moderní, přehledné a intuitivní uživatelské rozhraní. Pro inspiraci byly využity různé návrhy dashboardů a komponent dostupných na platformě <https://dribbble.com/>. (19)

### Wireframy

Před návrhem grafického designu aplikace byly započaty návrhy wireframů, které představují základní vizuální koncept aplikace, který slouží jako strukturovaný náčrt rozložení prvků na stránce často bez použití finálních barev, typografie nebo grafických prvků. Wireframy se vytvářejí primárně proto, aby definovaly rozvržení obsahu, navigaci a celkovou hierarchii informací na webové stránce, což umožňuje rychlou iteraci a zpětnou vazbu ještě předtím, než se začne pracovat na detailním grafickém designu. (20)

Během zpracovávání maturitní práce byly wireframy prezentovány a konzultovány s vývojovým týmem a odborníky na UI/UX, kteří poskytli cennou zpětnou vazbu k logice rozmístění prvků, přístupnosti a intuitivnosti rozraní. Na základě těchto konzultací byly provedeny úpravy, aby byla zajištěna, co nejlepší použitelnost aplikace.

Výsledné wireframy představují již detailní návrh aplikace, což usnadnilo následnou tvorbu finálního grafického designu. Pokrývají jak desktopové, tak mobilní rozvržení a kladou důraz na efektivní využití prostoru i přes implementaci nových funkčních prvků.

Celkový návrh se vyznačuje minimalistickým, čistým a jednotným stylem s oblými hranami prvků, které se doplňují s oblými plnými ikonami, čímž působí vizuálně přívětivěji než předchozí aplikace. Pro ušetření místa byly repetitivní textové popisky jako oprávnění a pracovní role nahrazeny ikonami v tabulce.

Největší výzvou při tvorbě wireframů byla responzivita – rozsáhlé tabulky a velké množství nových filtrů bylo nutné transformovat do kompaktnější podoby a zároveň stále zachovat kompletní funkčnost. To vedlo k použití rozbalovacích filtrů a karet pro jednotlivé výkazy, které šetří místo a zároveň stále poskytují uživateli přehled o nejdůležitějších informacích.

Obsah obrázku text, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka zvolených ikon pro oprávnění a pracovních rolí

Obsah obrázku snímek obrazovky, Multimediální software, Grafický software, text

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka wireframů z pohledu návrhu ve Figmě

Obsah obrázku text, snímek obrazovky, číslo, software

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka wireframu stránky Moje výkazy v desktopové verzi

Obsah obrázku text, účtenka, snímek obrazovky, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka wireframu stránky Moje Výkazy v mobilní verzi

Obsah obrázku text, účtenka, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka wireframu stránky Moje Výkazy v mobilní verzi s otevřenými filtry a kartou výkazu

Obsah obrázku text, snímek obrazovky, software, Počítačová ikona

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka wireframu stránky Můj profil v desktopové verzi

Obsah obrázku text, snímek obrazovky, číslo, software

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka wireframu stránky Všichni uživatelé v desktopové verzi

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka wireframu stránky Můj profil v mobilní verzi

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek –Ukázka wireframu stránky Nový výkaz v mobilní verzi

### Finální grafický design

Díky strategii vytvoření podrobných wireframů předem a jejich následnému schválení kolegy nebyla tvorba finálního grafického designu nové aplikace příliš náročná. Zjednodušeně řečeno, stačilo na existující wireframy aplikovat zvolenou barevnou paletu a doladit drobné detaily.

Obsah obrázku kruh, snímek obrazovky, text, design

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka zvolené barevné palety pro grafický design nové aplikace

Klíčové bylo navrhnout design efektivně s využitím postupů jako jsou opakovaně použitelné komponenty, proměnné pro text, barvy nebo stíny, ze kterých nástroj Figma těží. Tento přístup umožnil sestavit jednotlivé stránky aplikace z již předem definovaných prvků, čímž se eliminovala potřeba opakované tvorby prvků. Výhodou tohoto přístupu je snadná udržitelnost – jakákoli úprava rodičovské komponenty se automaticky propíše do všech jejích odvozených variant, což usnadňuje správu designu a zajišťuje konzistentní vzhled napříč celou aplikací. (21)

Obsah obrázku text, snímek obrazovky, software, Multimediální software

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka vytvořených opakovatelně použitelných komponent pro grafický design

#### Přihlašovací stránka

Design stránky obsahuje klasické dashboard rozvržení, kromě formuláře pro přihlášení pod emailem a heslem je zde nově tlačítko pro přihlášení prostřednictvím Microsoft účtu. Stránka obsahuje také odkaz pro obnovu zapomenutého hesla.

Obsah obrázku text, snímek obrazovky, design

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka designu stránky pro přihlášení v mobilní i desktopové verzi vedle sebe

#### Nový výkaz

Design stránky recykluje původní formulář polí pro zadání obsahu nového výkazu. Checkbox „zobrazit všechny úkoly“ zobrazí v select výběru úkolu i neaktivní úkoly – slouží pro manažery a administrátory v přístupu k historickým úkolům. Checkbox „neznám úkol“ přepne výběr možností úkolu na výběr možností klienta.

Obsah obrázku text, snímek obrazovky, software, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek –Ukázka designu stránky Nový výkaz v desktopové verzi

Obsah obrázku snímek obrazovky, řada/pruh, text

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka přepnutého selectu pro výběr klienta

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka designu stránky Nový výkaz v mobilní verzi

#### Moje výkazy

Design stránky „Moje výkazy“ obsahuje vizuálně přehlednou tabulku, předdefinovaná tlačítka pro vyfiltrování nejčastěji volených hodnot datumu (dnes, tento týden, tento měsíc), textové vyhledávání a tlačítko pro resetování filtrů. Jednotlivé výkazy jsou seskupeny vizuálně dle data – mezi jednotlivými dny jsou mezery. Tlačítko filtr, které otevírá pokročilé filtrování dle data, výběru klienta a jeho úkolu. Také je v pokročilém filtru zahrnuto tlačítko pro export dat do aplikace Microsoft Excel. Na stránce je také statický přehled průměru a součtu času vykázané práce, které reagují na všechny aplikovatelné filtry. Tlačítka v posledním sloupci umožňují proklik na editaci výkazu, jeho duplikaci, či smazání.

Tabulky v celé aplikaci obsahují i jednotný způsob stránkování, možnost volby počtu zobrazených záznamů a informaci o aktuálně zobrazených záznamech.

Obsah obrázku text, snímek obrazovky, software, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka stránky moje výkazy v desktopové verzi

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka otevřeného pokročilého filtru na stránce Moje výkazy v desktopové verzi

Jak již bylo zmíněno, výzvou bylo optimalizovat obsah i pro mobilní zařízení. To bylo umožněno díky kartám místo řádků tabulky, které se rozbalují. Stejně tak filtry se dělí na základní a pokročilé – rozbalí se po kliknutí na tlačítko filtr. Vše je vidět na ukázkách níže.

Obsah obrázku text, snímek obrazovky, číslo, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka designu stránky Moje výkazy v mobilní verzi

Obsah obrázku text, snímek obrazovky, číslo, účtenka

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka otevřených filtrů a karty výkazu na stránce Moje výkazy v mobilní verzi

#### Všechny výkazy

Design stránky obsahuje tabulku s výkazy všech uživatelů, pro vizuální oddělení stavů výkazů jsou schválené výkazy zešedlé, výkazy ke schválení jsou kontrastní a obsahují zelené tlačítko ke schválení daného výkazu. Neznámé výkazy s nezadaným úkolem, pouze se zadáním názvu klienta, jsou červeně zvýrazněné. Opět jsou výkazy sloučené vizuálně dle dní.

Pro rychlé vyfiltrování výkazů bez úkolu (OP – obchodního případu) slouží checkbox a vedle něj je umístěn checkbox pro vyfiltrování výkazu pouze svého týmu (pro administrátory, kteří vidí všechny uživatele). Je zde také výběr stavu schválení a reset tlačítko.

Tlačítko „hromadně schválit vybrané“ umožňuje schválit hromadně označené výkazy ke schválení, které byly pomocí checkboxu před jejich datumem označeny.

Tlačítko „Filtr“ stejně jako na stránce „Moje výkazy“ otevírá pokročilé filtrování podle datumu, textového vyhledávání, výběru uživatele, klienta a jeho úkolu. Je zde také tlačítko pro export do excelu.

Obsah obrázku text, snímek obrazovky, software, Počítačová ikona

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka stránky Všechny výkazy v desktopové verzi

Obsah obrázku text, snímek obrazovky, číslo, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka zapnutého pokročilého filtru na stránce Všechny výkazy v desktopové verzi

Obsah obrázku text, číslo, snímek obrazovky

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka stránky Všechny výkazy v mobilní verzi

Obsah obrázku text, číslo, účtenka, snímek obrazovky

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka otevřených filtrů na stránce Všechny výkazy v mobilní verzi

Obsah obrázku text, snímek obrazovky, software, Počítačová ikona

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka stránky Editace výkazu v mobilní a desktopové verzi

#### Můj profil

Stránku „Můj profil“ vyplňuje formulář pro změnu údajů přihlášeného uživatele. Umožňuje spárovat uživatelský účet nástroje Výkazovník s Microsoft Office uživatelským účtem. Při zaškrtnutí checkboxu „Zobrazit všechny úkoly“ se nastaví globálně skrz celou aplikací možnost vidět neaktivní úkoly.

Obsah obrázku text, snímek obrazovky, software, Počítačová ikona

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka stránky Můj profil v mobilní a desktopové verzi

#### Všichni uživatelé

Stránka umožňuje filtrovat uživatele aplikace pomocí výběru nadřízeného uživatele, vybraného oprávnění, zda uživatel vyžaduje kontrolu a dle textového vyhledávání. Také obsahuje tlačítko pro resetování filtrů. Mobilní design využívá opět rozbalovacích karet.

Obsah obrázku text, snímek obrazovky, software, Počítačová ikona

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka stránky Všichni uživatelé v mobilní a desktopové verzi

Obsah obrázku text, snímek obrazovky, software, Multimediální software

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka stránky Nový uživatel v mobilní a desktopové verzi

Obsah obrázku text, snímek obrazovky, software, Počítačová ikona

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka stránky Editace uživatele v mobilní a desktopové verzi

#### Toggle

Stránka „Toggle“ implementuje novou funkcionalitu aplikace, a to měření času stráveného nad jednotlivými úkoly. Uživatel vybírá v nabídce dostupné úkoly a zahájí měření času. Měření času v podobě stopek lze stopovat, či ukončit. Po ukončení je uživatel přesměrován na tvorbu nového výkazu s předvyplněným úkolem a časem.

Obsah obrázku snímek obrazovky, software, text, Multimediální software

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka stránky Toggle v mobilní a desktopové verzi

#### Standup

Design nové stránky „Standup“ pomůže díky rychlému a jednoduchému přehledu výkazů z předchozího dne zaměstnancům při ranních standupech ve firmě.

Obsah obrázku text, snímek obrazovky, software, Počítačová ikona

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka stránky standup v mobilní a desktopové verzi

#### Mobilní stránka „Více“

Design stránky „Více“ představuje v mobilní verzi aplikace takový rozcestník pro přechod na méně využívané stránky.

Obsah obrázku text, snímek obrazovky, Lidská tvář, Mobilní telefon

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka stránky „Více“ na mobilních zařízeních

# Vývoj a implementace nové aplikace

Po vytvoření návrhu aplikace a jeho schválení ze strany firmy SMWORKS byl započat vývoj nové aplikace.

## Zvolené technologie a struktura aplikace

Struktura webových aplikací se dělí technologiemi na takzvaný backend a frontend.

### Backend

Backend je serverová část aplikace, která běží na serveru a zajišťuje logiku aplikace, zpracování dat a komunikaci s databází. Umožňuje, aby frontend (uživatelské rozhraní) měl přístup k datům, mohl si je číst, zapisovat, upravovat a mazat. Z pohledu uživatele je backend neviditelný, ale je zásadní pro fungování celé aplikace. (22)

Firma SMWORKS používá pro tvorbu backend při tvorbě webových aplikací a stránek robustní framework Laravel pro PHP, který klade důraz především na MVC (Model-view-controller) architekturu. Kolegy ve firmě proto byl výše zmíněný návrh API implementován pomocí Laravel, který poskytuje REST API pro frontend, řeší autentizaci uživatelů a další backendové operace. (23) (24) (25)

### Frontend

Frontend aplikace představuje uživatelské rozhraní, se kterým uživatel přímo interaguje. Zahrnuje všechny vizuální a interaktivní prvky, jako jsou tlačítka, formuláře, tabulky, navigace a další komponenty. Na rozdíl od backendu běží frontend přímo v prohlížeči a je tvořen technologiemi, jako jsou HTML, CSS a JavaScript. (22)

Společnost SMWORKS dosud pro vývoj webových aplikací nejčastěji využívala framework Vue.js. Pro novou aplikaci Výkazovník však byl zvolen React, jedná se o nejrozšířenější frontendovou knihovnu pro tvorbu uživatelských rozhraní. React nabízí širší ekosystém knihoven a nástrojů, které usnadňují integraci různých funkcionalit, než Vue.js. Díky virtuálnímu DOM poskytuje rychlé a efektivní vykreslování uživatelského rozhraní, což je klíčové zejména pro větší aplikace, jako je Výkazovník. Další výhodou je lepší kompatibilita s backendovým frameworkem Laravel, což usnadňuje komunikaci mezi frontendem a backendem. (26) (27) (28) (29)

Pro psaní frontendového kódu byl zvolen **TypeScript**, což je nadstavba jazyka JavaScript s podporou statického typování, které zvyšuje bezpečnost a udržitelnost kódu. (30)

Pro efektivnější správu stylů aplikace byla použita technologie **SCSS**, což je preprocesor CSS umožňující lepší organizaci a znovupoužitelnost stylů, čímž se zlepšuje udržovatelnost kódu. (31)

## Spojení frontend–backend a databáze

Aplikace Výkazovník je postavena na architektuře Client-Server, kde frontend, backend a databáze funguje jako samostatné, ale úzce propojené komponenty. (32)

#### REST API

Jak již bylo zmíněno Laravel poskytuje REST API, které zajišťuje takového prostředníka mezi databází a frontendem prostřednictvím HTTP požadavků.

Obsah obrázku text, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka složky v backend s controllery pro Rest API

#### Správa dat v Reactu

React poskytuje vestavěnou funkci Context API, která umožňuje jednoduché sdílení globálních dat napříč celou aplikací. Ve spojení s Provider komponentou, která uchovává data a poskytuje je pomocí Contextu, lze efektivně řídit stav aplikace bez nutnosti propisování dat přes jednotlivé komponenty – props drilling. (33) (27)

Obsah obrázku text, Písmo, snímek obrazovky, bílé

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka vytvoření jednoduchého kontextu v React

Zde je exemplární ukázka vytvoření komponenty ClientsProvider, která uchovává data o klientech. Pomocí komponenty useFetch si pro data „sahá“ přes API do backend, který Provideru vrátí klienty z databáze. Také obsahuje stav načítání a případných errorů. To vše poté předává dalším komponentám, které jsou komponentou ClientsProvider obaleny, právě pomocí vytvořeného ClientContextu.

Obsah obrázku text, snímek obrazovky, Písmo, dokument

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka vytvoření jednoduchého Provideru React

Ostatní komponenty aplikace, které jsou ClientsProvider komponentou obaleny, mohou tak jednoduše přistoupit k datům klientů, či je znovu načíst a pracovat s nimi.

Obsah obrázku text, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka využití Provideru s kontextem v ostatních komponentách aplikace

Tento obecný přístup použitý v nové aplikaci umožňuje jednoduchou správu všech dat z různých úrovní komponent aplikace.

Obsah obrázku text, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka složky ve frontend aplikace se všemi providery aplikace

#### Výsledná ukázka toku dat v aplikaci

Pro shrnutí je zde rozepsán proces komunikace mezi databází, backendem a frontend aplikace:

1. Uživatel **otevře stránku s výkazy**.
2. Provider na **frontend zavolá** **fetchReports()**, což je **fetch** **request na API**.
3. **REST API v Laravelu přijme požadavek na /reports**, **zavolá databázi a vrátí JSON odpověď.**
4. **Provider na frontend uloží data** do svého globálně sdíleného stavu.
5. **Komponenta s výpisem výkazů vezme data z Provideru** poskytující kontext a zobrazí je uživateli.



Obrázek – Diagram toku dat v aplikaci

## React knihovny a vlastní komponenty

Jak již bylo zmíněno, React disponuje rozsáhlým ekosystémem knihoven, které usnadňují vývoj webových aplikací tím, že poskytují hotová řešení běžně používaných komponent a funkcionalit. Tyto knihovny jsou dostupné přes **NPM** (Node Package Manager) a umožňují vývojářům zrychlit svoji práci. (29) (26)

V nové aplikaci hraje klíčovou roli zobrazování dat v tabulkách na desktopových zařízeních, kde uživatelé potřebují efektivně pracovat s velkým množstvím informací. Proto byla zvolena knihovna React Table, která poskytuje pokročilé funkce jako řazení, filtrování, stránkování, změny sloupců a další možnosti přizpůsobení tabulek. Díky této knihovně bylo možné implementovat jednoduše tabulky plné funkcionalit. (34)

Obsah obrázku text, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka inicializace React Table se základní konfigurací

Kromě tabulek obsahuje aplikace také různé formulářové prvky, jako jsou výběry možností (selecty), textové vstupy (inputy) a výběry datumu (datepickery). Pro usnadnění práce s těmito prvky byly využity knihovny React Select a React Datepicker, které poskytují bohatou funkcionalitu pro práci s uživatelskými vstupy. Aby bylo možné tyto knihovny lépe přizpůsobit potřebám aplikace, byly vytvořeny vlastní komponenty MyDatePicker.tsx a InputGroup.tsx. Tyto komponenty umožňují jednotné stylování, přizpůsobení chování a zjednodušují správu uživatelských vstupů napříč celou aplikací. (35) (36)

Obsah obrázku text, snímek obrazovky, číslo, dokument

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka univerzální komponenty InputGroup.tsx, která umožňuje použít více typů vstupů

Obsah obrázku text, snímek obrazovky, Písmo, dokument

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka vlastní komponenty MyDatePicker.tsx, které rozšiřuje základní funkcionalitu knihovny React-datepicker

Aplikace také pracuje s různými formáty datumů a časových údajů, proto využívá knihovny Dayjs a Date-fns za účelem lehčí manipulace s datem a časem. (37) (38)

Aby byl kód aplikace přehlednější a nedocházelo k opakování stejných částí, byly vytvořeny různé další znovupoužitelné komponenty, které se starají o opakující se prvky na stránkách. Mezi ně patří například:

* **Menu** – navigace vyskytující se skrz celou aplikací.
* **Header** – hlavička stránky, která zobrazuje název sekce, případně tlačítko pro návrat zpět.
* **Alert** – komponenta pro zobrazování upozornění a notifikací uživateli.
* **Loader** – vizuální indikátor načítání dat nebo obsahu.
* **Button** – univerzální komponenta pro tlačítka, která nad nimi umožňuje větší kontrolu.
* **Pagination** – univerzální komponenta pro přepínání mezi stránkami v tabulkách.

Díky kombinaci knihoven a vlastních komponent se podařilo vytvořit modulární a snadno rozšířitelný kód pro možný budoucí rozvoj aplikace.

Obsah obrázku text, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka komponenty Alert umožňující zobrazit různé typy upozornění uživatele skrz aplikací

## Routování

Nová aplikace využívá pro routování (směrování mezi stránkami aplikace) knihovnu React Router, která umožňuje dynamické přesměrování mezi stránkami a efektivní práci s parametry URL. (39)

### Implementace

React Router poskytuje mechanismus pro definování jednotlivých cest (routes) v aplikaci a přiřazení odpovídajících komponent, které se mají vykreslit při návštěvě dané stránky. V aplikaci byly použity funkce CreateBrowserRouter() a createRoutesFromElements(), které umožňují definovat celý routovací systém v jediném přehledném objektu. (39) (40)

### Chráněné stránky pomocí ProtectedRoute

Některé části aplikace jsou dostupné pouze přihlášeným uživatelům nebo specifickým uživatelským rolím – viz oddíl 2.4 Sitemapy. K tomuto účelu byla vytvořena komponenta ProtectedRoute, která kontroluje, zda má uživatel dostatečná práva ke vstupu na danou stránku. Tato komponenta se vkládá do každé routy, která má být chráněná v definici routování v komponentě App.tsx.

Obsah obrázku text, snímek obrazovky, dokument, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka komponenty ProtectedRoute



Obrázek – Ukázka tvorby routování aplikací v komponentě App.tsx

Obsah obrázku text, snímek obrazovky

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka implementace ProtectedRoute v definici routování React Routeru

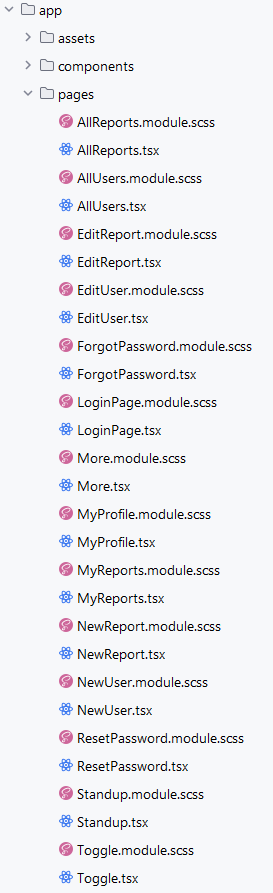
## Jednotlivé stránky aplikace

Každá stránka v aplikaci je sestavena kombinací znovupoužitelných komponent a samostatné logiky, která obstarává specifické funkce dané stránky. Tento modulární přístup umožňuje snadnou údržbu a rozšiřování aplikace bez nadbytečného opakování kódu.

Například stránka Nový výkaz (NewReport.tsx) se skládá z:

* **Menu** – navigační komponenta dostupná na všech stránkách.
* **Header** – komponenta zobrazující název stránky a možnost návratu na předchozí stránku.
* **Formulářové vstupy** – zahrnují různé typy polí, jako jsou textová pole, selecty, datumové vstupy a číselná pole – tvořené komponentami InputGroup.tsx a MyDatePicker.tsx.
* **Správa stavu a kontext** – stránka využívá kontexty (useReportContext, useTasksContext, useClientContext atd.) z providerů pro efektivní práci s daty z API.
* **Dynamická logika stránky** – zahrnuje funkce pro uložení výkazu, validaci vstupních údajů, automatické doplňování hodnot a logiku pro výběr úkolu či klienta na základě dříve zadaných údajů.
* **Interaktivní prvky** – například kopírování odkazu na výkaz, automatické vyplňování dat na základě parametrů v URL nebo dynamické načítání souvisejících úkolů a klientů.
* **Zlepšení uživatelského zážitku** – například indikátor načítání (komponenta Loader.tsx), zobrazení chybových hlášek a systémových notifikací pro uživatele prostřednictvím komponenty MessageProvider.

Podobná struktura je aplikována i na ostatní stránky aplikace, což umožňuje jednotnou architekturu a efektivní správu kódu.



Obrázek – Ukázka složky s jednotlivými stránkami aplikace

Obsah obrázku text, snímek obrazovky

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka komponenty NewReport.tsx využívající před vytvořené komponenty Header, Loader a InputGroup

## Progresivní webová aplikace

Jedním z klíčových vylepšení nového Výkazovníku jak již bylo zmíněno v katalogu požadavků je jeho transformace na progresivní webovou aplikaci (PWA). Tento přístup kombinuje výhody webových a nativních aplikací, což umožňuje uživatelům aplikaci instalovat přímo na svá zařízení z webového prohlížeče. Díky tomu je aplikace nezávislá z hlediska distribuce, jelikož ji není nutné publikovat na obchody s aplikacemi.

Cachování statických souborů jako je HTML, CSS a JavaScript – ukládání souborů přímo do úložiště zařízení uživatele optimalizuje rychlost aplikace. PWA může být také do budoucna rozšířena o různé funkce jako například push notifikace, které mohou například upozorňovat na nutnost vykázání práce. (41) (42)

### Implementace PWA

Základem každé PWA aplikace je tzv. Service Worker – speciální skript běžící na pozadí, který zachytává síťové požadavky a umožňuje cachování souborů pro rychlejší načítání i offline režim. Pro Výkazovník byl vytvořen Service Worker, který se stará o:

* **Cachování** statických souborů.
* Základní obsluhu **offline** režimu.
* **Správu verzí** – automaticky aktualizuje cache při vydání nové verze aplikace. (43)

Dalším důležitým prvkem PWA aplikace je Web App Manifest – JSON soubor, který definuje metadata aplikace, jako jsou její název, ikony, barvy a způsob zobrazení.

Obsah obrázku text, rukopis, dokument, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka JSON souboru Manifest definující metadata aplikace

Obsah obrázku text, snímek obrazovky, dokument, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka použitého skriptu Service Worker pro možnost PWA nového Výkazovníku

Přidáním Service Worker a Manifestu skriptu do složky /public aplikace se Výkazovník automaticky stává PWA, kterou lze stáhnout z prohlížeče a spustit jako samostatnou aplikaci.

Obsah obrázku text, software, Multimediální software, Počítačová ikona

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek – Ukázka instalace aplikace v prohlížeči Chrome

# Nasazení, testování a zpětná vazba

## Nasazení pro testování aplikace

Při finalizacích vývoje aplikace byla aplikace Výkazovník nasazena na firemní server pod doménou [new.vykazovnik.cz](https://new.vykazovnik.cz/). Pro zajištění hladkého přechodu mezi starou a novou aplikací byl kolegy ve firmě vytvořen skript pro převod dat ze staré databáze do nové. Díky tomu byla do nové aplikace nahrána testovací data z původní aplikace, což umožnilo reálné testování, aniž by byla ovlivněna produkční data dosud používané aplikace.

## Testování aplikace

Po nasazení byla aplikace zpřístupněna vybraným zaměstnancům firmy, kteří ji postupně testovali a hodnotili její funkčnost a uživatelský komfort při simulaci běžného používání aplikace v reálném provozu.

## Zpětná vazby a její implementace

Po několika týdnech testování byla shromážděna zpětná vazba od uživatelů. Během testování se neobjevily žádné kritické chyby narušující funkčnost aplikace, většina podnětů se týkala zlepšení uživatelského prostředí (UX/UI).

Hlavní připomínky se zaměřovaly na zlepšení práce s uživatelskými vstupy. Například u výběru data byla požadována možnost jednoduchého smazání tlačítkem s křížkem. Selecty pro výběr hodnot by měly intuitivněji dávat najevo uživateli možnost vyhledávání pomocí psaní do selectu, díky vymazání placeholderu. Další návrhy zahrnovaly omezenou dobu uchování filtrů v local storage, aby si aplikace nepamatovala staré filtry po dlouhém čase neaktivity. Po uložení nového výkazu by měl být uživatel automaticky přesměrován na přehled výkazů. Při hromadném označování výkazů by měl být viditelný jejich celkový počet, aby uživatel věděl, kolik položek vybral.

Všechny tyto uvedené a další drobné úpravy týkající se uživatelského prostředí byly postupně zapracovány do nové aplikace.

## Možná budoucí vylepšení

V průběhu testování nové aplikace vzniklo několik návrhů pro budoucí rozšíření funkcionality aplikace. Jedním z nápadů byla implementace push notifikací uživatelům, kteří zapomněli zastavit měření, například po několika hodinách běhu. Také vzniknul nápad na propojení toggle měření s backendem, což by umožnilo sledovat a spravovat měření na více zařízeních současně, nejen tedy lokálně jako je tomu teď přes local storage. Zajímavý návrh vznikl z hlediska vytvoření extension do prohlížečů, které by plnilo práci toggle stránky, ale rovnou v prohlížeči bez nutnosti otevírat Výkazovník.

Jak již bylo zmíněno v kapitole 1 Analýza stávající aplikace, při sbírání nových nápadů pro tvorbu nového Výkazovníku, byly navrhnuty funkce, které se odkláněly od původní myšlenky aplikace – jednalo se spíše o analytické funkce. Z tohoto důvodu nebyly implementovány v první verzi nové aplikace, avšak zůstávají jako potenciální rozšíření do budoucna.

## Nasazení na produkci

Ke dni 12.03.2025 je aplikace stále v testovací fázi na doméně <https://new.vykazovnik.cz/> s implementovanou zpětnou vazbou od uživatelů, kteří testovali její funkčnost. V nejbližších týdnech od termínu uveřejnění této maturitní práce dojde k aktualizaci její databáze pro synchronizaci dat a aplikace bude spuštěna pro všechny zaměstnanci firmy na doméně <https://vykazovnik.cz/>.

Závěr

Cíl vytvořit novou aplikaci, která odstraní nedostatky původního systému a poskytne moderní uživatelskou zkušenost odpovídající současným technologickým standardům, byl úspěšně splněn. Aplikace je navržena tak, aby byla snadno rozšiřitelná a umožňovala další vývoj v budoucnosti.

Po důkladné analýze stávající aplikace byly definovány požadavky na nový systém. Na jejich základě vznikl návrh přepracované databázové struktury a odpovídajících API endpointů, které zajišťují efektivní komunikaci mezi frontendem a backendem. Frontend aplikace prošel celým procesem návrhu – od wireframů až po finální grafický design, což zajistilo vizuální konzistenci a optimalizaci pro mobilní zařízení. Backend a databáze byly vyvinuty zaměstnanci firmy na základě těchto návrhů, zatímco frontend byl postaven na moderním JavaScriptovém frameworku React s využitím aktuálních vývojových postupů, což zajišťuje jeho dlouhodobou udržitelnost. Po dokončení vývoje byla aplikace důkladně testována zaměstnanci a zpětná vazba byla následně zapracována do finální verze.

Celý proces představoval komplexní výzvu zahrnující analytickou, návrhovou a vývojovou fázi tvorby nového softwaru. Výsledkem této práce je plně funkční, modernizovaná aplikace Výkazovník, která výrazně zvyšuje efektivitu a zlepšuje uživatelský komfort při vykazování práce. Projekt nejen splnil stanovené cíle, ale zároveň vytvořil pevný základ pro budoucí rozvoj. Hotová aplikace implementuje všechny klíčové požadavky definované v úvodní analýze a reflektuje reálné potřeby uživatelů, což ji činí plně připravenou pro nasazení do produkčního prostředí.

Seznam zkratek a odborných výrazů

CTO

Chief Technology Officier – Výkonná pozice ve firmě zodpovědná za technologický směr společnosti.

CRM

Customer Relationship Management – Software určený ke správě interakcí se zákazníky, poskytující často automatizaci obchodních procesů a zlepšení zákaznické zkušenosti.

P.R.D.E.L

Pohoda-Raynet Data Exchange Layer – Interní systém firmy SMWORKS zajišťující automatizovanou výměnu dat mezi programy Pohoda a Raynet.

API

Application Programming Interface – Sada pravidel a nástrojů, které umožňují aplikacím mezi sebou komunikovat. API se často využívá v aplikacích pro komunikaci mezi backendem a frontendem.

CSS

Cascading Style Sheets – Kaskádové styly využívající se ke stylování HTML dokumentů, tedy k úpravě vzhledu webových stránek. Umožňují definovat barvy, písma, rozložení prvků, animace a další vizuální aspekty.

UI

User Interface – Uživatelské rozhraní představuje vizuální podobu aplikace, včetně jejího vzhledu, rozložení prvků, barevné palety, typografie a interaktivních prvků jako tlačítka, formuláře či navigace. Dobře navržené UI usnadňuje orientaci v aplikaci a zajišťuje intuitivní ovládání.

UX

User Experience – Uživatelský prožitek se zaměřuje na celkovou použitelnost a efektivitu interakce s aplikací. Zahrnuje například logiku uspořádání funkcí, rychlost operací, přehlednost obsahu či snadnost dokončení určitého úkolu. Cílem UX designu je, aby uživatel mohl aplikaci používat pohodlně a efektivně.

PWA

Progressive Web App – Progresivní webová aplikace je typ webové aplikace, která kombinuje výhody webu a nativních mobilních aplikací. Díky moderním technologiím umožňuje rychlé načítání, offline režim, push notifikace a instalaci na zařízení, čímž poskytuje lepší uživatelský zážitek.

HTML

HyperText Markup Language – Značkovací jazyk, který se používá k tvorbě struktury webových stránek. Umožňuje definovat různé prvky jako nadpisy, odstavce, obrázky, odkazy nebo formuláře. Samotné HTML neřeší vzhled (to dělá CSS) ani interaktivitu (to řeší JavaScript).

JavaScript

Je skriptovací programovací jazyk, který se používá ke vytváření dynamických a interaktivních prvků na webových stránkách. Pomocí něj lze například animovat prvky, načítat data z API, validovat formuláře nebo spravovat uživatelské interakce.

Dnes se JavaScript využívá nejen pro web, ale i pro backend, mobilní aplikace nebo desktopové aplikace.

Android

Je operační systém vyvíjený společností Google, který běží na většině mobilních zařízení, tabletů, chytrých televizí nebo hodinek.

iOS

Je operační systém vyvíjený společností Apple, který běží na zařízeních jako iPhone, iPad nebo iPod Touch. Na rozdíl od Androidu je iOS uzavřený systém, což znamená větší bezpečnost, ale i větší omezení pro vývojáře a uživatele.

MSA

Microsoft Account – Je jednotný uživatelský účet, který umožňuje přístup ke službám a produktům společnosti Microsoft. MSA podporuje OAuth 2.0, což umožňuje jeho integraci jako přihlašovací metodu třetích stran v aplikacích. Vývojáři mohou využít Microsoft Identity Platform k implementaci "Přihlášení pomocí Microsoftu", což zjednodušuje autentizaci a zvyšuje bezpečnost.

MySQL

Je open-source relační databázový systém, který slouží k ukládání, správě a manipulaci s daty pomocí strukturovaného jazyky SQL (Structured Query Language)

SQL

Je jazyk určený pro práci s relačními databázemi. Umožňuje vkládat, upravovat, mazat a dotazovat se na data uložená v databázových tabulkách.

RESTful

Representational State Transfer (REST) Architecture – RESTful označuje webové API, které je navrženo podle architektonických principů REST (Representational State Transfer). REST je standardní způsob, jak komunikují klienti (frontend, mobilní aplikace) s backendovými servery pomocí HTTP požadavků.

API Endpoint

Je konkrétní URL adresa, na kterou klient (např. frontend aplikace) posílá požadavky na server. Každý endpoint odpovídá určité operaci, kterou API poskytuje.

HTTP

HyperText Transfer Protocol – Protokol, který se používá k přenosu dat mezi klientem (např. prohlížečem nebo mobilní aplikací) a serverem na internetu.

HTTP Metoda

HTTP metoda určuje, jaký typ operace klient provádí nad API. Například GET pro získání dat, POST pro vytvoření nového záznamu nebo PUT pro aktualizaci celého záznamu.

JWT Token

JSON Web Token – Standard pro bezpečnou autentizaci a přenos informací mezi dvěma stranami pomocí digitálně podepsaného tokenu.

PHP

Je skriptovací jazyk na straně serveru, určený pro vývoj webových aplikací a dynamických stránek.

Laravel

Je open-source PHP framework pro vývoj moderních webových aplikací. Je založen na MVC architektuře (Model-View-Controller) a poskytuje nástroje pro autentizaci, routování, práci s databází a API.

Figma

Je cloudový designový nástroj, který se používá k tvorbě UI/UX návrhů, wireframů, prototypů a spolupráci na grafických projektech. Umožňuje více lidem pracovat současně na jednom projektu přímo v prohlížeči, bez nutnosti instalace.

Backend

Je serverová část aplikace, která zajišťuje zpracování dat, logiku aplikace a komunikaci s databází. Běží na serveru a poskytuje data frontendu prostřednictvím API. Backend zahrnuje autentizaci uživatelů, správu databází a business logiku aplikace.

Frontend

Je uživatelská část aplikace, která zajišťuje vizuální rozhraní a interakci s uživatelem. Běží přímo v prohlížeči a je vytvořen nejčastěji pomocí HTML, CSS a JavaScriptu. Obsahuje všechny vizuální a interaktivní prvky, jako jsou tlačítka, formuláře, tabulky či navigace. Frontend komunikuje s backendem pomocí API a zobrazuje uživateli získaná data.

TypeScript

Je programovací jazyk vyvinutý Microsoftem, který rozšiřuje JavaScript (JS) o statické typování a další pokročilé funkce. TypeScript překládá (kompiluje) do čistého JavaScriptu, což znamená, že ho lze použít ve všech prostředích, kde běží JavaScript, například ve webových prohlížečích, Node.js nebo v Reactu a Vue.js.

SCSS

SCSS je rozšíření CSS, které přidává pokročilé funkce pro efektivnější a přehlednější stylování webových aplikací. SCSS je součástí Sass (Syntactically Awesome Stylesheets), jednoho z nejpopulárnějších preprocesorů CSS.

URL

Uniform Resource Locator – Je adresa, která slouží k jednoznačné identifikaci zdroje na internetu. Pomocí URL lze přistupovat k webovým stránkám, souborům, API endpointům nebo jiným síťovým zdrojům.

JSON

JavaScript Object Notation – Je lehce čitelný formát pro výměnu dat, který se často používá v API komunikaci mezi frontendem a backendem.

Seznam obrázků

[Obrázek 1 – Jednoduchý dotazník pro zaměstnance firmy k získání zpětné vazby 3](#_Toc192756853)

[Obrázek 2 – Přehled výkazů uživatele ze stávající aplikace 4](#_Toc192756854)

[Obrázek 3 – Demostrance pouze částečné reponzivity webové aplikace 5](#_Toc192756855)

[Obrázek 4 – Demostrance pouze částečné reponzivity tabulky webové aplikace 6](#_Toc192756856)

[Obrázek 5 – Sekce „Všechny výkazy“ se špatně čitelným textem 7](#_Toc192756857)

[Obrázek 6 – Sekce „Editování výkazu“ na mobilním zařízení 8](#_Toc192756858)

[Obrázek 7 – Sekce „Všechny výkazy“ na mobilním zařízení 9](#_Toc192756859)

[Obrázek 8 –Diagram návrhu nové databáze 14](#_Toc192756860)

[Obrázek 9 – Ukázka návrhu API – ReportController (1. část) 17](#_Toc192756861)

[Obrázek 10 – Ukázka návrhu API – ReportController (2. část) 18](#_Toc192756862)

[Obrázek 11 – Ukázka sitemap pro desktopové zařízení 19](#_Toc192756863)

[Obrázek 12 – Ukázka sitemap pro mobilní zařízení 19](#_Toc192756864)

[Obrázek 13 – Ukázka zvolených ikon pro oprávnění a pracovních rolí 22](#_Toc192756865)

[Obrázek 14 – Ukázka wireframů z pohledu návrhu ve Figmě 22](#_Toc192756866)

[Obrázek 15 – Ukázka wireframu stránky Moje výkazy v desktopové verzi 23](#_Toc192756867)

[Obrázek 16 – Ukázka wireframu stránky Moje Výkazy v mobilní verzi 24](#_Toc192756868)

[Obrázek 17 – Ukázka wireframu stránky Moje Výkazy v mobilní verzi s otevřenými filtry a kartou výkazu 25](#_Toc192756869)

[Obrázek 18 – Ukázka wireframu stránky Můj profil v desktopové verzi 26](#_Toc192756870)

[Obrázek 19 – Ukázka wireframu stránky Všichni uživatelé v desktopové verzi 26](#_Toc192756871)

[Obrázek 20 – Ukázka wireframu stránky Můj profil v mobilní verzi 27](#_Toc192756872)

[Obrázek 21 –Ukázka wireframu stránky Nový výkaz v mobilní verzi 28](#_Toc192756873)

[Obrázek 22 – Ukázka zvolené barevné palety pro grafický design nové aplikace 29](#_Toc192756874)

[Obrázek 23 – Ukázka vytvořených opakovatelně použitelných komponent pro grafický design 30](#_Toc192756875)

[Obrázek 24 – Ukázka designu stránky pro přihlášení v mobilní i desktopové verzi vedle sebe 30](#_Toc192756876)

[Obrázek 25 –Ukázka designu stránky Nový výkaz v desktopové verzi 31](#_Toc192756877)

[Obrázek 26 – Ukázka přepnutého selectu pro výběr klienta 31](#_Toc192756878)

[Obrázek 27 – Ukázka designu stránky Nový výkaz v mobilní verzi 32](#_Toc192756879)

[Obrázek 28 – Ukázka stránky moje výkazy v desktopové verzi 33](#_Toc192756880)

[Obrázek 29 – Ukázka otevřeného pokročilého filtru na stránce Moje výkazy v desktopové verzi 34](#_Toc192756881)

[Obrázek 30 – Ukázka designu stránky Moje výkazy v mobilní verzi 35](#_Toc192756882)

[Obrázek 31 – Ukázka otevřených filtrů a karty výkazu na stránce Moje výkazy v mobilní verzi 36](#_Toc192756883)

[Obrázek 32 – Ukázka stránky Všechny výkazy v desktopové verzi 37](#_Toc192756884)

[Obrázek 33 – Ukázka zapnutého pokročilého filtru na stránce Všechny výkazy v desktopové verzi 38](#_Toc192756885)

[Obrázek 34 – Ukázka stránky Všechny výkazy v mobilní verzi 39](#_Toc192756886)

[Obrázek 35 – Ukázka otevřených filtrů na stránce Všechny výkazy v mobilní verzi 40](#_Toc192756887)

[Obrázek 36 – Ukázka stránky Editace výkazu v mobilní a desktopové verzi 41](#_Toc192756888)

[Obrázek 37 – Ukázka stránky Můj profil v mobilní a desktopové verzi 41](#_Toc192756889)

[Obrázek 38 – Ukázka stránky Všichni uživatelé v mobilní a desktopové verzi 42](#_Toc192756890)

[Obrázek 39 – Ukázka stránky Nový uživatel v mobilní a desktopové verzi 43](#_Toc192756891)

[Obrázek 40 – Ukázka stránky Editace uživatele v mobilní a desktopové verzi 43](#_Toc192756892)

[Obrázek 41 – Ukázka stránky Toggle v mobilní a desktopové verzi 44](#_Toc192756893)

[Obrázek 42 – Ukázka stránky standup v mobilní a desktopové verzi 45](#_Toc192756894)

[Obrázek 43 – Ukázka stránky „Více“ na mobilních zařízeních 46](#_Toc192756895)

[Obrázek 44 – Ukázka složky v backend s controllery pro Rest API 48](#_Toc192756896)

[Obrázek 45 – Ukázka vytvoření jednoduchého kontextu v React 49](#_Toc192756897)

[Obrázek 46 – Ukázka vytvoření jednoduchého Provideru React 49](#_Toc192756898)

[Obrázek 47 – Ukázka využití Provideru s kontextem v ostatních komponentách aplikace 50](#_Toc192756899)

[Obrázek 48 – Ukázka složky ve frontend aplikace se všemi providery aplikace 50](#_Toc192756900)

[Obrázek 49 – Diagram toku dat v aplikaci 51](#_Toc192756901)

[Obrázek 50 – Ukázka inicializace React Table se základní konfigurací 51](#_Toc192756902)

[Obrázek 51 – Ukázka univerzální komponenty InputGroup.tsx, která umožňuje použít více typů vstupů 53](#_Toc192756903)

[Obrázek 52 – Ukázka vlastní komponenty MyDatePicker.tsx, které rozšiřuje základní funkcionalitu knihovny React-datepicker 54](#_Toc192756904)

[Obrázek 53 – Ukázka komponenty Alert umožňující zobrazit různé typy upozornění uživatele skrz aplikací 56](#_Toc192756905)

[Obrázek 54 – Ukázka komponenty ProtectedRoute 57](#_Toc192756906)

[Obrázek 55 – Ukázka tvorby routování aplikací v komponentě App.tsx 58](#_Toc192756907)

[Obrázek 56 – Ukázka implementace ProtectedRoute v definici routování React Routeru 59](#_Toc192756908)

[Obrázek 57 – Ukázka složky s jednotlivými stránkami aplikace 61](#_Toc192756909)

[Obrázek 58 – Ukázka komponenty NewReport.tsx využívající před vytvořené komponenty Header, Loader a InputGroup 62](#_Toc192756910)

[Obrázek 59 – Ukázka JSON souboru Manifest definující metadata aplikace 63](#_Toc192756911)

[Obrázek 60 – Ukázka použitého skriptu Service Worker pro možnost PWA nového Výkazovníku 64](#_Toc192756912)

[Obrázek 61 – Ukázka instalace aplikace v prohlížeči Chrome 65](#_Toc192756913)

[Obrázek 62 – QR kód vedoucí na Github repozitář MP I](#_Toc192756914)

Použité zdroje

1. **Kuboš, Jaroslav.** Projektové řízení. *Navigo3.* [Online] 11. září 2019. [Citace: 3. únor 2025.] https://navigo3.com/cs/projektove-rizeni/.

2. **Caflou.** ERP systém pro malé firmy – co, kdo a proč. *Caflou.* [Online] [Citace: 3. únor 2025.] https://www.caflou.cz/erp-system-pro-male-firmy-co-kdo-a-proc.

3. **Sonika, Malviya.** Výhody timesheetu, které musí znát každý majitel firmy. *Rank Tracer.* [Online] 31. červenec 2024. [Citace: 3. únor 2025.] https://www.ranktracker.com/cs/blog/timesheet-benefits-every-business-owner-must-know/.

4. **Raynet.** Nejob­líbenější cloudové CRM v ČR. *Raynet.* [Online] [Citace: 2025. únor 4.] https://raynet.cz/cloud-crm/.

5. **Michálek, Martin.** *Vzhůru do (responzivního) webdesignu.* místo neznámé : Martin Michálek - Vzhůru dolů, 2018. 9788088253006.

6. **Bootstrap.** *Bootstrap.* [Online] 2025. [Citace: 2. únor 2025.] https://getbootstrap.com/docs/3.4/.

7. **Lamprecht, Emil.** The Difference Between UX and UI Design: A Beginner’s Guide. *Careerfoundry.* [Online] 19. únor 2025. [Citace: 23. únor 2025.] https://careerfoundry.com/en/blog/ux-design/the-difference-between-ux-and-ui-design-a-laymans-guide/.

8. **Mozilla Foundation.** Progressive web apps. *MDN Web Docs.* [Online] Mozilla Foundation. [Citace: 12. únor 2025.] https://developer.mozilla.org/en-US/docs/Web/Progressive\_web\_apps.

9. **Microsoft.** Tok přihlašování aplikací pomocí platformy Microsoft Identity Platform. *Microsoft Learn.* [Online] 24. říjen 2023. [Citace: 18. únor 2025.] https://learn.microsoft.com/cs-cz/entra/identity-platform/app-sign-in-flow.

10. **Wikipedia foundation.** Normalizace databáze. *Wikipedie otevřená encyklopedie.* [Online] 16. prosinec 2024. [Citace: 2025. únor 20.] https://cs.wikipedia.org/wiki/Normalizace\_datab%C3%A1ze.

11. **Dan Nosowitz, Michael Goodwin.** What is an API endpoint? *IBM.* [Online] [Citace: 21. únor 2025.] https://www.ibm.com/think/topics/api-endpoint.

12. **The Postman Team.** What is an API endpoint? *Postman.* [Online] 20. červen 2023. [Citace: 22. únor 2025.] https://blog.postman.com/what-is-an-api-endpoint/.

13. **Wikipedia foundation.** MySQL. *Wikipedie otevřená encyklopedie.* [Online] 7. leden 2025. [Citace: únor. 23 2025.] https://cs.wikipedia.org/wiki/MySQL.

14. **Wikipedia Foundation.** Entity-relationship model. *Wikipedie otevřená encyklopedie.* [Online] 9. červenec 2024. [Citace: 24. únor 2025.] https://cs.wikipedia.org/wiki/Entity-relationship\_model.

15. **Laravel.** Controllers. *Laravel.* [Online] [Citace: 25. únor 2025.] https://laravel.com/docs/12.x/controllers.

16. **Doglio, Fernando.** How to secure a REST API using JWT authentication. *Logrocket.* [Online] 12. srpen 2022. [Citace: 25. únor 2025.] https://blog.logrocket.com/secure-rest-api-jwt-authentication/.

17. **Claritee.** Visual Sitemaps: Structured Diagrams for Website Planning. *Claritee.* [Online] 8. květen 2024. [Citace: 26. únor 2025.] https://claritee.io/blog/visual-sitemaps-structured-diagrams-for-website-planning/.

18. **Stevens, Emily.** What Is Figma? Uses, Benefits, and Key Features. *Andacademy.* [Online] 23. srpen 2024. [Citace: 26. únor 2025.] https://www.andacademy.com/resources/blog/graphic-design/what-is-figma/.

19. **RSMS.** The Inter typeface family. *rsms.me.* [Online] RSMS, 2017. [Citace: 27. únor 2025.] https://rsms.me/inter/.

20. **Wikipedia Foundation.** Font Awesome. *Wikipedie otevřená encyklopedie.* [Online] 25. leden 2025. [Citace: 27. únor 2025.] https://en.wikipedia.org/wiki/Font\_Awesome.

21. **Figma.** What is wireframing? *Figma.* [Online] [Citace: 28. únor 2025.] https://www.figma.com/resource-library/what-is-wireframing/.

22. —. Guide to components in Figma. *Figma.* [Online] [Citace: 1. březen 2025.] https://help.figma.com/hc/en-us/articles/360038662654-Guide-to-components-in-Figma.

23. **Geeksforgeeks.** Frontend vs Backend. *Geeksforgeeks.* [Online] 10. září 2024. [Citace: 2. březen 2025.] https://www.geeksforgeeks.org/frontend-vs-backend/.

24. **Laravel.** Laravel Fortify. *Laravel.* [Online] [Citace: 4. březen 2025.] https://laravel.com/docs/12.x/fortify.

25. **Hanif, Fikri.** Understanding the MVC Architecture in Laravel: A Comprehensive Guide. *Fkrihnif.* [Online] 26. červen 2023. [Citace: 3. březen 2025.] https://fkrihnif.medium.com/understanding-the-mvc-architecture-in-laravel-a-comprehensive-guide-8f620cc139b6.

26. **Laravel News.** Generate Apis with ease with Laravel Rest Api. *Laravel News.* [Online] 8. září 2023. [Citace: 4. březen 2025.] https://laravel-news.com/laravel-rest-api.

27. **Wikipedia Foundation.** React (webový framework). *Wikipedie otevřená encyklopedie.* [Online] 3. únor 2025. [Citace: 5. březen 2025.] https://cs.wikipedia.org/wiki/React\_(webov%C3%BD\_framework).

28. **Bhavsar, Dipal.** How to Use Laravel with React for Building Modern Web Applications. *Bacancytechnology.* [Online] 11. září 2024. [Citace: 6. březen 2025.] https://www.bacancytechnology.com/blog/laravel-with-react.

29. **Geeksforgeeks.** React JS ReactDOM. *Geeksforgeeks.* [Online] 9. leden 2025. [Citace: 5. březen 2025.] https://www.geeksforgeeks.org/reactjs-reactdom/.

30. **Modan, Shahil.** React.js benefits: Why It’s perfect for your business applications. *Peerbits.* [Online] 23. leden 2025. [Citace: 7. březen 2025.] https://www.peerbits.com/blog/reasons-to-choose-reactjs-for-your-web-development-project.html.

31. **Gupta, Karnika.** Mastering React with Typescript: It’s Benefits and Importance. *Medium.* [Online] 4. říjen 2023. [Citace: 7. březen 2025.] https://medium.com/womenintechnology/mastering-react-with-typescript-its-benefits-and-importance-85cbc783a85a.

32. **Shiotsu, Yoshitaka.** What Is SCSS? A Beginner's Guide for Developers. *Upwork.* [Online] 20. červenec 2024. [Citace: 7. březen 2025.] https://www.upwork.com/resources/what-is-scss.

33. **Geeksforgeeks.** Client-Server Model. *Geeksforgeeks.* [Online] 3. leden 2025. [Citace: 7. březen 2025.] https://www.geeksforgeeks.org/client-server-model/.

34. **React.** createContext. *React.* [Online] [Citace: 8. březen 2025.] https://react.dev/reference/react/createContext.

35. **TanStack.** TanStack table. *TanStack.* [Online] [Citace: 7. březen 2025.] https://tanstack.com/table/latest.

36. **JedWatson.** React Select. *React Select.* [Online] [Citace: 7. březen 2025.] https://react-select.com/home.

37. **HackerOne Inc. & Contributors.** react-datepicker. *NPM JS.* [Online] 7. březen 2025. [Citace: 7. březen 2025.] https://www.npmjs.com/package/react-datepicker.

38. **Koss., Sasha.** Getting Started. *Date FNS.* [Online] [Citace: 7. březen 2025.] https://date-fns.org/docs/Getting-Started.

39. **iamkun.** Day.js. *Day JS.* [Online] [Citace: 7. březen 2025.] https://day.js.org/.

40. **Ryan Florence, Michael Jackson.** React Router Home. *React Router.* [Online] [Citace: 7. březen 2025.] https://reactrouter.com/home.

41. —. CreateRoutesFromElements. *React Router.* [Online] [Citace: 7. březen 2025.] https://reactrouter.com/6.30.0/utils/create-routes-from-elements.

42. **Koďousková, Barbora.** Co jsou progresivní webové aplikace (PWA) a jaké mají výhody. *Rascasone.* [Online] 1. prosinec 2024. [Citace: 7. březen 2025.] https://www.rascasone.com/cs/blog/progresivni-webova-aplikace-vyhody.

43. **Wikipedia Foundation.** Progresivní webové aplikace. *Wikipedie otevřená encyklopedie.* [Online] 8. květen 2025. [Citace: 7. březen 2025.] https://cs.wikipedia.org/wiki/Progresivn%C3%AD\_webov%C3%A9\_aplikace.

44. **Web.dev.** Service workers . *Web.dev.* [Online] [Citace: 8. březen 2025.] https://web.dev/learn/pwa/service-workers.

45. **Das, Arunangshu.** State Management in React: Props Drilling vs. Context API. *Arunangshudas.* [Online] 16. květen 2024. [Citace: 7. březen 2025.] https://article.arunangshudas.com/state-management-in-react-props-drilling-vs-context-api-24caaa78bfde.

46. **Ryan Florence, Michael Jackson.** createBrowserRouter. *React router.* [Online] [Citace: 7. březen 2025.] https://reactrouter.com/6.30.0/routers/create-browser-router.

1. Seznam přiložených souborů

Na přiloženém datovém nosiči se nacházejí následující soubory a složky:

* **MP2024-25\_Kraus-Ivan\_Navrh-a-tvorba-firemni-aplikace-pro-vykazy.docx** – editovatelná verze dokumentu maturitní práce.
* **MP2024-25\_Kraus-Ivan\_Navrh-a-tvorba-firemni-aplikace-pro-vykazy.pdf** – tisknutelná verze dokumentu maturitní práce.
* **frontend-aplikace** – složka se zdrojovými kódy aplikace.
* **Dokumentace aplikace Výkazovník.docx** – dokumentace návrhu databáze a API endpointů aplikace.
* **Výkazovník V4.0.fig –** exportované wireframy a grafický design aplikace ve Figmě.
* **vykazovnik\_db\_scheme.drawio** – diagram návrhu databáze v .drawio formátu.
* **vykazovnik\_db\_scheme.drawio.png** – diagram návrhu databáze v .png formátu.
* **Výkazovník-V4.0-sitemap-desktop.drawio.png** – diagram sitemapy aplikace pro desktopová zařízení.
* **Výkazovník-V4.0-sitemap-mobile.drawio.png** – diagram sitemapy aplikace pro mobilní zařízení.

Výše zmíněné soubory a složky se také nacházejí v online Github repozitáři MP:<https://github.com/pslib-cz/MP2024-25_Kraus-Ivan_Navrh-a-tvorba-firemni-aplikace-pro-vykazy/>



Obrázek – QR kód vedoucí na Github repozitář MP